

UNIVERSIDADE FEDERAL FLUMINENSE

MARCELLO WILLIANS MESSINA RIBEIRO

**PROCESSAMENTO PARALELO DE CONSULTAS  
OLAP EM NUUVENS DE COMPUTADORES COM  
C-PARGRES**

NITERÓI

2018

UNIVERSIDADE FEDERAL FLUMINENSE

MARCELLO WILLIANS MESSINA RIBEIRO

**PROCESSAMENTO PARALELO DE CONSULTAS  
OLAP EM NUVENS DE COMPUTADORES COM  
C-PARGRES**

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Computação da Universidade Federal Fluminense como requisito parcial para a obtenção do Grau de Mestre em Computação. Área de concentração: Engenharia de Sistemas de Informação

Orientador:

DANIEL CARDOSO MORAES DE OLIVEIRA

NITERÓI

2018

Ficha catalográfica automática - SDC/BEE  
Gerada com informações fornecidas pelo autor

R484p    Ribeiro, Marcello Willians Messina  
          Processamento Paralelo de Consultas OLAP em Nuvens de  
          Computadores com C-ParGRES : / Marcello Willians Messina  
          Ribeiro ; Daniel Cardoso Moraes de Oliveira, orientador.  
          Niterói, 2018.  
          136 f. : il.

          Dissertação (mestrado)-Universidade Federal Fluminense,  
          Niterói, 2018.

          DOI: <http://dx.doi.org/10.22409/PGC.2018.m.05777338771>

          1. Computação em nuvem. 2. Data Warehousing. 3. Base de  
          dados distribuída. 4. Paralelismo. 5. Produção intelectual.  
          I. Cardoso Moraes de Oliveira, Daniel, orientador. II.  
          Universidade Federal Fluminense. Escola de Engenharia. III.  
          Título.

CDD -

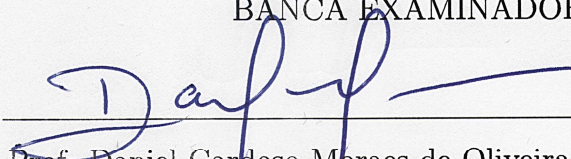
MARCELLO WILLIANS MESSINA RIBEIRO

PROCESSAMENTO PARALELO DE CONSULTAS OLAP EM NUVENS DE  
COMPUTADORES COM C-PARGRES

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Computação da Universidade Federal Fluminense como requisito parcial para a obtenção do Grau de Mestre em Computação. Área de concentração: Engenharia de Sistemas de Informação

Aprovada em Dezembro de 2018.

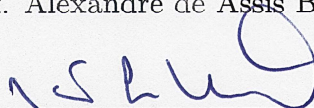
BANCA EXAMINADORA



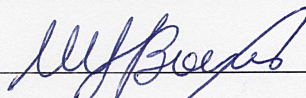
Prof. Daniel Cardoso Moraes de Oliveira - Orientador, UFF



Prof. Alexandre de Assis Bento Lima, COPPE/UFRJ



Prof. Marta Lima de Queirós Mattoso, COPPE/UFRJ



Prof. Maria Cristina Silva Boeres, UFF

Niterói

2018

*Dedico esta dissertação à minha esposa Danielle, pelo apoio incondicional e constante  
inventivo em todos os momentos deste longo caminho.*

# Agradecimentos

Início os agradecimentos por meus pais, que sempre zelaram pela minha Educação e sempre me incentivaram a seguir em frente. Obrigado meu pai João e minha mãe Vera por, além de me oferecerem a oportunidade de estudar, terem sido meus exemplos de integridade, respeito e amor ao próximo. Sou muito Feliz por isso.

Uma vez dentro da Universidade, durante a graduação tive contato com muitos professores, mas alguns ficaram marcados para sempre na minha memória. Obrigado Professor Dante Corbucci Filho por me apresentar, de maneira tão intensa e bem humorada, os primeiros conceitos de algoritmos e programação. Suas aulas nunca serão esquecidas! Agradeço também ao Professor Maurício Kischinhevsky pelo acolhimento e direcionamento dado a mim e a meus colegas nos primeiros semestres da graduação. Obrigado por ter me incentivado a fazer esse curso e pela amizade que permanece mesmo após todos esses anos.

Agradeço aos meus colegas de trabalho do IBGE Cláudio Mariano Fernandes, Luiz Antônio Vivacqua Corrêa Meyer, Bianca Fernandes Sotelo e José Sant'Anna Bevilaqua pelo apoio e pela compreensão nas minhas ausências imprevistas e saídas antecipadas para assistir às aulas do mestrado. Sem o apoio de vocês eu não teria conseguido chegar até aqui.

Agradeço também ao aluno de mestrado Thaylon Santos pelo incentivo e pelas dicas preciosas dos mais diversos assuntos.

Desde o início do curso sempre pude contar com meu amigo André Luis da Costa Nascimento. Fizemos a graduação na UFF juntos e tivemos a sorte de poder repetir essa experiência como colegas de turma no mestrado. Agradeço muito pelo seu apoio e incentivo durante todos esses anos!

Não posso deixar de agradecer (muito) a ajuda do meu Orientador Daniel Cardoso Moraes de Oliveira. O direcionamento durante esses anos foi primordial para que eu chegasse até aqui. Sua paciência e compreensão com todas as extensões de prazo também!

Obrigado Daniel! Espero poder contar com sua amizade mesmo após a conclusão deste curso.

### **AGRADECIMENTO ESPECIAL**

Meu agradecimento mais profundo e sincero só poderia ser dedicado a uma pessoa: minha Esposa Danielle. Durante todo o tempo (e não foi pouco!) ela ficou ao meu lado, me dando força e me fazendo acreditar que eu conseguiria. Os finais de semana perdidos e as noites mal dormidas, não foram tão difíceis pois você estava ao meu lado. Sou grato por cada gesto carinhoso, cada sorriso, cada palavra de incentivo, cada brincadeira. Você foi fundamental para que eu pudesse concluir esta importante etapa da minha vida. Essa conquista é nossa! Meu amor, te agradeço do fundo do meu coração!

# Resumo

O advento das tecnologias de *Big Data* mudou a forma como muitas empresas gerenciam seus dados. Várias empresas transferiram seus dados para a nuvem adotando o conceito de *Database-as-a-Service* (DBaaS). Mover bancos de dados para a nuvem apresenta vários desafios relacionados ao gerenciamento flexível e escalável de dados. Embora algumas dessas empresas tenham migrado para bancos de dados *noSQL*, a maioria ainda depende de bancos de dados relacionais na nuvem para gerenciar seus dados, especialmente os dados essenciais para o processo de tomada de decisão. As consultas de processamento analítico *on-line* (OLAP) demoram muito tempo para serem processadas, exigindo recursos de computação de alto desempenho de seus sistemas de banco de dados para obter resultados em um tempo viável. Neste trabalho, propomos uma solução de *middleware* que pode ser implantada em qualquer provedor de nuvem, **C-ParGRES**, que explora a replicação de banco de dados e o paralelismo inter- e intra-consultas para oferecer apoio eficiente a consultas OLAP na nuvem. **C-ParGRES** é uma extensão do ParGRES, um *middleware* de *cluster* de banco de dados de código aberto para processamento de consulta OLAP de alto desempenho. **C-ParGRES** explora recursos de nuvem, como provisionamento e elasticidade de recursos sob demanda. Além disso, o **C-ParGRES** pode criar *clusters* virtuais múltiplos e independentes para diferentes bancos de dados e usuários. Nós avaliamos **C-ParGRES** com duas aplicações OLAP reais, ambas do Instituto Brasileiro de Geografia e Estatística - IBGE. Os resultados mostram que o **C-ParGRES** é uma solução econômica e efetiva para o processamento de consultas OLAP na nuvem.

**Palavras-chave:** computação na nuvem; bancos de dados autônomos distribuídos; processamento paralelo de consultas OLAP; paralelismo.



# Abstract

The advent of Big Data technologies has changed the way many companies manage their data. Several companies moved their data to the cloud using the concept of Database-as-a-service (DBaaS). Moving databases to the cloud presents several challenges related to flexible and scalable management of data. Although some of these companies migrated to noSQL databases, most still rely on relational databases in the cloud to manage data, especially data that is critical to the decision making process. On-line analytical processing (OLAP) queries take a long time to be processed, thus demanding high-performance capabilities from their associated database systems to get results in a feasible time. In this article, we propose a middleware solution that can be deployed in any cloud provider, **C-ParGRES**, which explores database replication and inter- and intra-query parallelism to efficiently support OLAP queries in the cloud. **C-ParGRES** is an extension of ParGRES, an open-source database cluster middleware for high performance OLAP query processing in clusters. **C-ParGRES** exploits cloud capabilities such as on demand resource provisioning and elasticity. In addition, **C-ParGRES** can create multiple and independent virtual clusters for different database and users. We evaluate **C-ParGRES** with two real-world OLAP applications, both from the Brazilian Institute of Geography and Statistics – IBGE. Results show that **C-ParGRES** is a cost-effective solution for OLAP query processing in the cloud.

**Keywords:** compute cloud; distributed autonomous databases; parallel OLAP query processing; parallelism.

# Lista de Figuras

2.1	Fluxograma das etapas de divulgação de uma pesquisa . . . . .	6
2.2	SQL de uma consulta do sistema BME . . . . .	11
2.3	Relacionamento entre as tabelas de fatos CD2010ADOMI, CD2010APESS, CD2010AFAMI, CD2010AMORT e CD2010AEMIGR . . . . .	13
2.4	Relacionamento entre as tabelas de fatos CD2010ADOMI e CD2010APESS e suas respectivas dimensões geográficas e temporal, com indicativo da quantidade de registros presentes em cada tabela . . . . .	19
3.1	Arquitetura do ParGRES adaptada de [51] . . . . .	21
3.2	Componentes do ParGRES . . . . .	22
3.3	Paralelismo Intra-Consulta . . . . .	24
3.4	Paralelismo Inter-Consulta . . . . .	25
4.1	Arquitetura do C-ParGRES . . . . .	27
5.1	Tabela SIDRA 3459. . . . .	48
5.2	Tabela SIDRA 3525. . . . .	49
5.3	Tabela SIDRA 3534. . . . .	50
5.4	Experimento 1: Tempos de execução normalizados das consultas BME Q01- Q07, <b>sem concorrência</b> . . . . .	54
5.5	Experimento 1: Tempos de execução normalizados das consultas BME Q08- Q14, <b>sem concorrência</b> . . . . .	55
5.6	Experimento 1: Tempos de execução normalizados das consultas BME- TABULADOR, <b>sem concorrência</b> . . . . .	55
5.7	Experimento 1: <i>Speedup</i> das consultas BME Q01-Q07, <b>sem concorrência</b> . . . . .	56
5.8	Experimento 1: <i>Speedup</i> das consultas BME Q08-Q14, <b>sem concorrência</b> . . . . .	56

5.9	Experimento 1: <i>Speedup</i> das consultas BME-TABULADOR, <b>sem concorrência</b> . . . . .	57
5.10	Experimento 2: Tempos de execução normalizados das consultas BME Q01-Q07, grupos de consultas <b>sem concorrência</b> . . . . .	60
5.11	Experimento 2: Tempos de execução normalizados das consultas BME Q08-Q14, grupos de consultas <b>sem concorrência</b> . . . . .	60
5.12	Experimento 2: Tempos de execução normalizados dos grupos de consultas BME G1-G4, <b>sem concorrência</b> . . . . .	61
5.13	Experimento 2: <i>Speedup</i> das consultas BME Q01-Q07, grupos de consultas <b>sem concorrência</b> . . . . .	61
5.14	Experimento 2: <i>Speedup</i> das consultas BME Q08-Q14, grupos de consultas <b>sem concorrência</b> . . . . .	62
5.15	Experimento 2: <i>Speedup</i> dos grupos de consultas BME G1-G4, <b>sem concorrência</b> . . . . .	62
5.16	Experimento 3: Tempos de execução normalizados das consultas BME Q01-Q07, grupos de consultas <b>com concorrência</b> . . . . .	65
5.17	Experimento 3: Tempos de execução normalizados das consultas BME Q08-Q14, grupos de consultas <b>com concorrência</b> . . . . .	65
5.18	Experimento 3: Tempos de execução normalizados dos grupos de consultas BME G1-G4, <b>com concorrência</b> . . . . .	66
5.19	Experimento 3: <i>Speedup</i> das consultas BME Q01-Q07, grupos de consultas <b>com concorrência</b> . . . . .	66
5.20	Experimento 3: <i>Speedup</i> das consultas BME Q08-Q14, grupos de consultas <b>com concorrência</b> . . . . .	67
5.21	Experimento 3: <i>Speedup</i> dos grupos de consultas BME G1-G4, <b>com concorrência</b> . . . . .	67
5.22	Experimento 4: Tempos de execução normalizados das consultas BME Q01-Q07, <b>com concorrência</b> em um ambiente com carga mista. . . . .	71
5.23	Experimento 4: Tempos de execução normalizados das consultas BME Q08-Q14, <b>com concorrência</b> em um ambiente com carga mista. . . . .	71

5.24	Experimento 4: Tempos de execução normalizados das consultas BME-TABULADOR, <b>com concorrência</b> em um ambiente com carga mista. . .	72
5.25	Experimento 4: Tempos de execução normalizados dos grupos de consultas BME G1-G4 e BME-TABULADOR, <b>com concorrência</b> em um ambiente com carga mista. . . . .	72
5.26	Experimento 4: <i>Speedup</i> das consultas BME Q01-Q07, <b>com concorrência</b> em um ambiente com carga mista. . . . .	73
5.27	Experimento 4: <i>Speedup</i> das consultas BME Q08-Q14, <b>com concorrência</b> em um ambiente com carga mista. . . . .	73
5.28	Experimento 4: <i>Speedup</i> das consultas BME-TABULADOR, <b>com concorrência</b> em um ambiente com carga mista. . . . .	74
5.29	Experimento 4: <i>Speedup</i> dos grupos de consultas BME G1-G4 e BME-TABULADOR, <b>com concorrência</b> em um ambiente com carga mista. . .	74

# Lista de Tabelas

2.1	Tamanho das tabelas de fatos da base CD2010A . . . . .	14
5.1	Características detalhadas das instâncias de máquinas virtuais. . . . .	37
5.2	Principais características das consultas do BME. . . . .	46
5.3	Principais características das consultas do BME-TABULADOR. . . . .	51
5.4	Experimento 1: Tempo médio de execução (T) em segundos e Desvio Padrão do tempo (DP) das consultas BME, <b>sem concorrência</b> . . . . .	53
5.5	Experimento 1: Tempo médio de execução (T) em segundos e Desvio Padrão do tempo (DP) das consultas BME-TABULADOR, <b>sem concorrência</b> . . . . .	54
5.6	Experimento 2: Tempo médio de execução (T) em segundos e Desvio Padrão do tempo (DP) das consultas BME, grupos de consultas <b>sem concorrência</b> . . . . .	58
5.7	Experimento 2: Tempo médio de execução (T) em segundos e Desvio Padrão do tempo (DP) dos grupos de consultas BME, <b>sem concorrência</b> . . . . .	59
5.8	Experimento 3: Tempo médio de execução (T) em segundos e Desvio Padrão do tempo (DP) das consultas BME, grupos de consultas <b>com concorrência</b> . . . . .	64
5.9	Experimento 3: Tempo médio de execução (T) em segundos e Desvio Padrão do tempo (DP) dos grupos de consultas BME, <b>com concorrência</b> . . . . .	64
5.10	Experimento 4: Tempo médio de execução (T) em segundos e Desvio Padrão do tempo (DP) das consultas BME, <b>com concorrência</b> em um ambiente com carga mista. . . . .	69
5.11	Experimento 4: Tempo médio de execução (T) em segundos e Desvio Padrão do tempo (DP) das consultas BME-TABULADOR, <b>com concorrência</b> em um ambiente com carga mista. . . . .	70

---

5.12	Experimento 4: Tempo médio de execução (T) em segundos e Desvio Padrão do tempo (DP) dos grupos de consultas BME e BME-TABULADOR, <b>com concorrência</b> em um ambiente com carga mista. . . . .	70
A.1	Tabela CD2010ADOMI e seus atributos . . . . .	88
A.2	Tabela CD2010APESS e seus atributos . . . . .	92

# Sumário

<b>1</b>	<b>Introdução</b>	<b>1</b>
<b>2</b>	<b>Referencial Teórico: Censo Demográfico 2010 e as atividades desenvolvidas pelo IBGE</b>	<b>4</b>
2.1	O IBGE e sua Função Constitucional . . . . .	4
2.1.1	Sigilo da informação estatística . . . . .	5
2.1.2	Disseminação de dados . . . . .	5
2.2	O Censo Demográfico 2010 e os sistemas de disseminação do IBGE . . . .	6
2.2.1	Censo Demográfico 2010 . . . . .	6
2.2.1.1	Data de referência . . . . .	7
2.2.1.2	Base territorial . . . . .	7
2.2.1.3	Dados do Universo e dados da Amostra . . . . .	8
2.2.2	O BME - Banco Multidimensional de Estatísticas . . . . .	9
2.2.2.1	Consultas <i>ad-hoc</i> e cálculos em tempo real . . . . .	10
2.2.2.2	A base do Censo Demográfico 2010 no BME . . . . .	12
2.2.3	O SIDRA - Sistema IBGE de Recuperação Automática . . . . .	14
2.2.4	O sistema BME-TABULADOR e o processo de Tabulação de Dados	16
2.2.4.1	O <i>case</i> do Censo 2010 - Dados da Amostra . . . . .	16
2.2.4.2	Comparativo entre consultas do BME e BME-TABULADOR	17
<b>3</b>	<b>ParGRES</b>	<b>20</b>
3.1	Componentes do ParGRES . . . . .	22

3.1.1	Reescrita de SQL . . . . .	23
3.1.2	Processamento de Consultas com paralelismo Inter- ou Intra-consultas	23
3.1.3	Composição/agregação de resultados . . . . .	24
3.1.4	Processamentos de atualizações . . . . .	25
<b>4</b>	<b>C-ParGRES: Um middleware para Processamento de Consultas em Ambientes de Nuvem</b>	<b>26</b>
4.1	C-ParGRES <i>Starter</i> . . . . .	26
4.1.1	Módulo <i>Dimensioner</i> do C-ParGRES <i>Starter</i> . . . . .	28
4.1.2	Módulo de <i>Deployment</i> do C-ParGRES <i>Starter</i> . . . . .	29
4.1.2.1	Atividade #1: Verificação e validação de Permissões de Acesso . . . . .	30
4.1.2.2	Atividade #2: Configuração de Grupos de Segurança . . .	30
4.1.2.3	Atividade #3: Criação, inicialização e monitoramento das Máquinas Virtuais . . . . .	31
4.1.2.4	Atividade #4: Desligamento e encerramento das Máquinas Virtuais . . . . .	32
4.1.3	Scripts para gerenciamento do C-ParGRES . . . . .	32
4.1.3.1	nqp.sh . . . . .	33
4.1.4	cqp.sh . . . . .	33
4.1.5	add_nodes.sh . . . . .	34
4.1.6	list_nodes.sh . . . . .	35
<b>5</b>	<b>Avaliação Experimental</b>	<b>36</b>
5.1	Ambiente de execução . . . . .	36
5.2	Base de dados . . . . .	37
5.3	Consultas selecionadas para o experimento . . . . .	40
5.3.1	Consultas da aplicação BME . . . . .	40



5.3.2	As consultas da aplicação BME-TABULADOR . . . . .	45
5.4	Experimentos realizados e resultados . . . . .	51
5.4.1	Experimento 1: Consultas BME e BME-TABULADOR, sem Concorrência . . . . .	52
5.4.2	Experimento 2: Grupos de Consultas BME, sem Concorrência . . .	53
5.4.3	Experimento 3: Grupos de Consultas BME, com Concorrência . . .	59
5.4.4	Experimento 4: Consultas BME e BME-TABULADOR, com Concorrência em um Ambiente com Carga Mista . . . . .	68
<b>6</b>	<b>Trabalhos Relacionados</b>	<b>75</b>
6.1	HBaseSQL . . . . .	75
6.2	CloudTPS . . . . .	75
6.3	MPI Cloud Platform . . . . .	76
6.4	OLAP* . . . . .	77
<b>7</b>	<b>Conclusão</b>	<b>78</b>
7.1	Contribuições . . . . .	78
7.2	Limitações . . . . .	79
7.3	Trabalhos Futuros . . . . .	80
	<b>Referências</b>	<b>82</b>
	<b>Apêndice A - TABELAS DE FATOS</b>	<b>88</b>
	<b>Apêndice B - TABELAS DE FATOS</b>	<b>104</b>
B.1	Consultas da aplicação BME . . . . .	104
B.2	Consultas da aplicação BME-TABULADOR . . . . .	121

# Capítulo 1

## Introdução

Existe um grande número de aplicativos e sistemas comerciais que armazenam seus dados em bancos de dados relacionais [67, 40]. Uma classe importante de consultas realizadas nestes bancos de dados é a OLAP (*On-line Analytical Processing*) que, geralmente, acessam grandes quantidades de dados e demoram muito tempo para serem concluídas. Na maioria dos casos, as consultas OLAP estão associadas a processos de tomadas de decisão e exploração de dados. Essas consultas geralmente são executadas por muitos minutos ou até horas, dependendo dos parâmetros escolhidos.

Um dos desafios para o processamento de consultas OLAP consiste na redução do tempo total de processamento. Para alcançar esse objetivo, é possível usar bancos de dados paralelos [63] e Sistemas de Gerência de Banco de Dados Paralelos - SGBDPs. SGBDPs exploram o paralelismo inter-consulta (cada consulta é executada sequencialmente, mas várias consultas são executadas em paralelo) e o paralelismo intra-consulta (a execução paralela das consultas é obrigatória) [63, 54].

Em muitas empresas que fazem uso de SGBDPs, existe uma máquina multiprocessada dedicada para o servidor de banco de dados, ou até mesmo um *cluster*, dependendo da demanda de processamento [40]. No entanto, essas máquinas e *clusters* podem ser bastante caras e, geralmente, superdimensionadas para atender cargas de pico, tornando-se subutilizadas na maior parte do tempo. Além disso, o *hardware* se deprecia ao longo do tempo, logo, ele deve ser substituído e/ou atualizado, aumentando os custos financeiros. Dessa forma, adquirir um SGBDP e o *hardware* e *software* necessários, pode ter um custo proibitivo para muitos usuários e pequenas empresas.

Desta forma, para reduzir os custos financeiros globais, os usuários podem utilizar *middlewares* de banco de dados de código aberto [20, 19, 14, 44, 45, 43, 17] que permitem

aos usuários executarem consultas em paralelo. O ParGRES [44, 45, 43, 17] é um dos mais proeminentes *middlewares* de banco de dados livre e de código aberto que visa fornecer processamento paralelo em consultas em *clusters*. O ParGRES intercepta as consultas SQL para explorar o paralelismo intra-consulta antes de submetê-las ao SGBD sequencial, cada um sendo executado em um nó em do *cluster*. Para usar SGBDs sequenciais, o ParGRES depende do particionamento virtual e um banco de dados replicado [1]. ParGRES é compatível com qualquer aplicativo que acesse um SGBD com apoio ao SQL-99.

Embora o uso de um *middleware* de banco de dados de código aberto represente um avanço, pode não ser suficiente para reduzir os custos financeiros. Outra maneira de reduzir custos é migrar o processamento de consultas de *clusters* locais para ambientes de computação em nuvem [64]. As nuvens podem ser usadas para evitar o excesso de provisionamento de servidores e os custos financeiros associados aos servidores de banco de dados. As nuvens fornecem computação como um serviço que pode ser comprado sob demanda por diferentes tipos de clientes, seguindo um modelo de pagamento por uso [13]. Diferentemente dos usuários de *grids* [16] e *clusters*, os usuários da nuvem se beneficiam do conceito de elasticidade, ou seja, eles podem escalar recursos facilmente de acordo com sua demanda de processamento, de forma praticamente ilimitada [64, 13, 52]. Uma vantagem no uso de nuvens é que o usuário não é obrigado a adquirir uma infraestrutura computacional dispendiosa para armazenar seus dados. Bancos de dados em nuvem são oferecidos por provedores de nuvem usando máquinas virtuais pré-configuradas <sup>1 2</sup>. Neste trabalho, assumimos este modelo de banco de dados como serviço (DBaaS).

Existem várias abordagens que já fornecem processamento de consultas paralelas na nuvem [68, 3, 52]. No entanto, a maioria delas apoia apenas operações SQL básicas no banco de dados, ou seja, nenhuma delas fornece recursos mais complexos, como particionamento virtual e balanceamento de carga. Isso acontece porque a migração de *clusters* e *grids* para nuvens apresenta muitos desafios para o processamento de consultas paralelas OLAP, pois o *middleware* deve considerar mais de um nível de processamento: (i) nível de nuvem, que requer a distribuição de tarefas em diferentes sites de nuvem (como as regiões no Amazon AWS), (ii) nível do *site*, que requer a distribuição de tarefas em diferentes *clusters* virtuais, e (iii) nível do *cluster* virtual, que requer a redistribuição dessas tarefas em várias máquinas virtuais. Esses níveis adicionais exigem atenção especial por parte do *middleware*.

O trabalho proposto nesta dissertação é um novo *middleware* baseado em nuvem

---

<sup>1</sup>Amazon RDS - <http://aws.amazon.com/>

<sup>2</sup>Google Cloud SQL - <http://cloud.google.com/>

chamado **C-ParGRES** que visa realizar consultas OLAP em paralelo, fornecendo recursos de balanceamento de carga e particionamento virtual, aproveitando as vantagens que a computação em nuvem oferece. O **C-ParGRES** estende o *middleware* ParGRES para fornecer uma solução baseada em nuvem para processamento com paralelismo inter- e intra-consulta. Consideramos um ambiente de nuvem típico com vários *clusters* virtuais em *sites* diferentes (*multi-site cloud* [46, 47]). Comparado com a abordagem de *cluster* de banco de dados, em que o banco de dados é replicado em um único *cluster*, o **C-ParGRES** permite que o banco de dados seja replicado em vários *clusters* virtuais no ambiente de nuvem, aumentando assim a disponibilidade de dados e a qualidade do serviço. Por exemplo, se um *cluster* virtual estiver indisponível, ainda é possível executar consultas OLAP usando os demais.

O **C-ParGRES** foi implementado como *DBaaS* no Amazon AWS, um dos provedores de nuvem mais importantes. Nesta dissertação, avaliamos o desempenho do **C-ParGRES** com consultas SQL provenientes de duas aplicações OLAP reais, ambas do Instituto Brasileiro de Geografia e Estatística - IBGE. Os resultados mostram uma aceleração linear ou quase linear na execução de consultas, à medida que mais máquinas virtuais são adicionadas, em todas as configurações testadas.

Esta dissertação está organizada em sete capítulos. O Capítulo 2 apresenta o IBGE e suas atribuições e os conceitos relativos ao Censo Demográfico 2010. O Capítulo 3 apresenta o *middleware* ParGRES e sua arquitetura. O Capítulo 4 apresenta o **C-ParGRES**, que é a implementação de uma extensão do ParGRES para ambientes de nuvem. O Capítulo 5 apresenta a avaliação experimental da abordagem proposta através dos resultados das execuções das consultas utilizando o **C-ParGRES**. O Capítulo 6 apresenta os trabalhos relacionados. Finalmente, o Capítulo 7 apresenta as conclusões desta dissertação ressaltando as contribuições, limitações e trabalhos futuros provenientes deste trabalho.

## Capítulo 2

# Referencial Teórico: Censo Demográfico 2010 e as atividades desenvolvidas pelo IBGE

### 2.1 O IBGE e sua Função Constitucional

As estatísticas oficiais constituem um elemento indispensável no sistema de informação de uma sociedade democrática, pois fornecem, ao governo e à sociedade civil, dados fundamentais sobre a situação econômica, demográfica, social e ambiental do país [23, 62]. Nesse contexto, o Instituto Brasileiro de Geografia e Estatística - IBGE [32] se constitui no principal provedor de dados e informações do Brasil. Suas pesquisas atendem às necessidades dos mais diversos segmentos da sociedade civil, como acadêmicos, pesquisadores e empresários, bem como dos órgãos das esferas governamentais federal, estadual e municipal.

Segundo a Constituição Federal [9], em seu artigo 21°, inciso XV, compete à União organizar e manter os serviços oficiais de estatística, geografia, geologia e cartografia de âmbito nacional. O IBGE é o órgão responsável por todos esses serviços e as principais funções que desempenha são [36]:

- Produção e análise de informações estatísticas
- Coordenação e consolidação das informações estatísticas
- Produção e análise de informações geográficas
- Coordenação e consolidação das informações geográficas

- Estruturação e implantação de um sistema de informações ambientais
- Documentação e disseminação de informações
- Coordenação dos sistemas estatístico e cartográfico nacionais

### 2.1.1 Sigilo da informação estatística

A maior parte das informações que o IBGE divulga são obtidas através das operações estatísticas (pesquisas) de âmbito social, demográfico, econômico e geográfico. Uma parcela menor das informações levantadas é proveniente de registros administrativos obtidos por meio de convênios específicos com outras entidades, de governo ou não. Todos os cidadãos, órgãos e empresas brasileiras são obrigados por lei a prestar informações estatísticas solicitadas pelo IBGE (Lei nº 5.534 de 14 de novembro de 1968) [58]. As informações prestadas têm caráter sigiloso e são usadas exclusivamente para fins estatísticos não podendo servir de prova em processo administrativo, fiscal ou judicial [2].

A identidade do informante também é preservada, permanecendo protegida durante todo o processo de disseminação de dados. Apenas dados agregados de um grupo significativo de informantes são divulgados, impossibilitando a identificação dos mesmos.

### 2.1.2 Disseminação de dados

O IBGE possui um dos maiores acervos especializados em informações estatísticas e geográficas do país. Para que essa informação possa ser difundida de forma efetiva e transparente para toda a população, o Instituto possui diversos canais de disseminação: publicações impressas e eletrônicas; tabelas agregadas e microdados publicados em seu site; *releases* e apresentações específicos para a imprensa; agência de notícias própria; atendimento presencial em todas as capitais e nas principais cidades do Brasil; e bases de dados e sistemas *on-line*.

A internet é hoje, sem dúvida, o principal canal de comunicação do IBGE com a população. Dentre os sistemas disponíveis podemos citar como principais o SIDRA - Sistema IBGE de Recuperação Automática [37] e o BME - Banco Multidimensional de Estatísticas [33], ambos gratuitos e disponíveis para qualquer cidadão acessar. Cada um deles será detalhado mais adiante.

A Figura 2.1 a seguir mostra de forma resumida o fluxograma de uma pesquisa desde a fase de planejamento até sua disseminação. Fica clara a integração e o papel de cada

etapa e de cada sistema no processo. É possível observar também que a disseminação da informação ocorre por diversos meios.

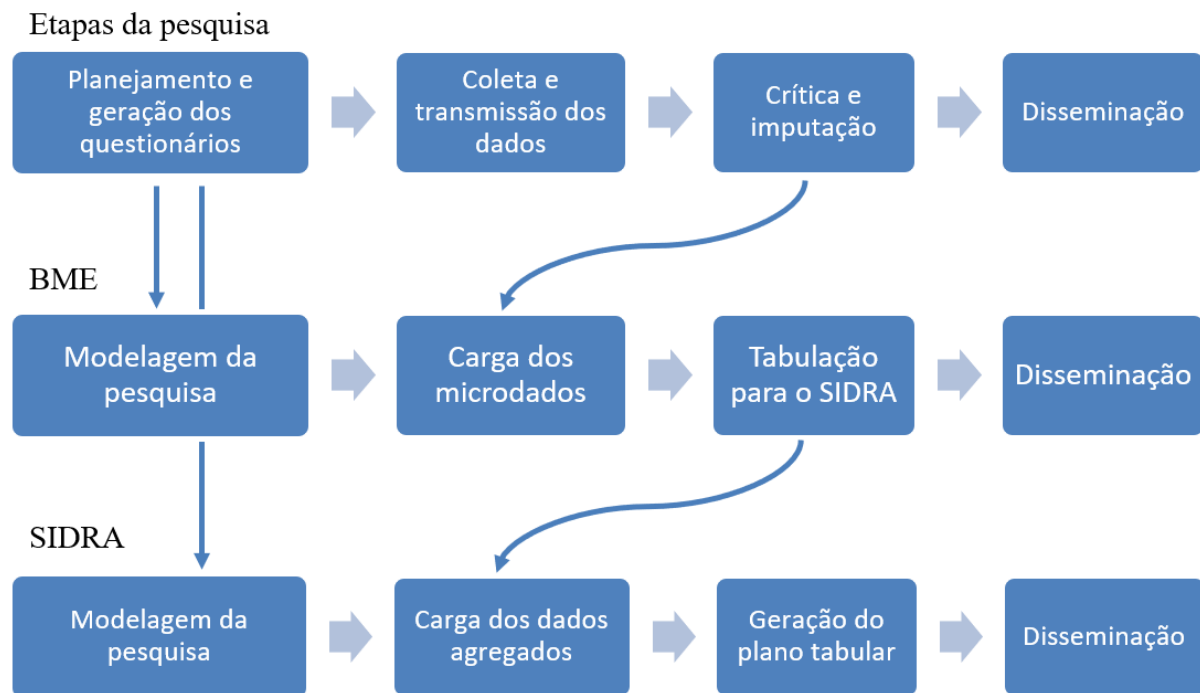


Figura 2.1: Fluxograma das etapas de divulgação de uma pesquisa

## 2.2 O Censo Demográfico 2010 e os sistemas de disseminação do IBGE

Nesta seção descrevemos os conceitos relativos ao Censo Demográfico 2010 e os desafios inerentes a uma pesquisa desse porte. A seção 2.2.1 detalha todas essas questões. Abordamos também dois dos principais sistemas responsáveis pela disseminação de informações do IBGE: o BME e o SIDRA, nas seções 2.2.2 e 2.2.3 respectivamente. Na seção 2.2.4 apresentamos o sistema BME-TABULADOR que serve de apoio para parte do processo de disseminação, integrando as bases de dados dos dois sistemas citados.

### 2.2.1 Censo Demográfico 2010

O Censo Demográfico é a mais complexa operação estatística realizada por um país, quando são investigadas as características de toda a população e dos domicílios do território nacional [31]. Todos os domicílios do país são visitados e dados sobre todos os seus moradores são levantados, representando o maior e mais preciso retrato de toda a

população em todos os municípios (e suas subdivisões - distritos, subdistritos e bairros) do Brasil. Um grande volume de informações essenciais é produzido para servir de base para a tomada de decisões nos mais diversos segmentos e esferas do governo, bem como para organizações e empresários nortear suas atividades e investimentos. Além disso, o Censo Demográfico também tem um importante papel dentro do próprio IBGE, pois serve de base para o planejamento das demais pesquisas populacionais, como por exemplo a PNAD - Pesquisa Nacional por Amostra de Domicílios e, sua sucessora, a PNAD-C - Pesquisa Nacional por Amostra de Domicílios Contínua.

### **2.2.1.1 Data de referência**

A periodicidade dos Censos Demográficos é regulamentada pela Lei no 8.184, de 10 de maio de 1991 [59], que estabelece um máximo de dez anos para o intervalo intercensitário. A coleta do Censo Demográfico 2010 foi realizada no período de 1º de agosto a 30 de outubro de 2010, contudo, para fins de apuração e divulgação dos dados, a data de referência da pesquisa foi 31 de julho de 2010.

### **2.2.1.2 Base territorial**

Base territorial é a denominação dada ao conjunto de mapas, cadastros e banco de dados, que formam um sistema integrado capaz de dar sustentação espacial às atividades de planejamento operacional, coleta e apuração de dados e divulgação de resultados do Censo Demográfico 2010. Sua função é possibilitar a cobertura integrada de todo o território nacional e ampliar as possibilidades de disseminação de informações à sociedade [31].

A organização político-administrativa do território brasileiro compreende a União, o Distrito Federal, os estados e os municípios. O setor censitário é a unidade territorial utilizada pelo IBGE para o controle dos dados cadastrais da coleta, respeitando a divisão político-administrativa citada, a separação de áreas urbanas e rurais e outros aspectos territoriais de interesse.

A subdivisão espacial em setores censitários é essencial para dar a organização necessária ao processo de coleta, visto que a realização de um levantamento como o Censo Demográfico 2010 é um desafio para qualquer instituto de estatística, dadas as dimensões continentais do nosso país. O Brasil é o maior país da América do Sul e ocupa a quinta posição no mundo com relação à área territorial. Em 2010 o Brasil teve sua área calculada em 8.515.692,27  $km^2$ , distribuídos em um território heterogêneo, composto por 27 Uni-



dades da Federação e o Distrito Federal, 5.565 municípios e 316.574 setores censitários, abrangendo cerca de 67,5 milhões de domicílios [24].

### 2.2.1.3 Dados do Universo e dados da Amostra

Atualmente, os Censos Demográficos do Brasil utilizam dois tipos de questionários para a realização da coleta dos dados: o questionário básico e o questionário da amostra [24].

O questionário básico é aplicado a todos os domicílios do país, com exceção daqueles selecionados para compor a amostra do Censo Demográfico, que respondem ao questionário da amostra. Além das perguntas contidas no questionário básico, o questionário da amostra investiga diversas outras características das unidades habitacionais, bem como informações sociais, econômicas e demográficas de seus moradores, dados de emigração, religião, tipos de trabalho e nupcialidade.

A técnica de amostragem utilizada nos Censos Demográficos brasileiros é uma amostra estratificada [31], considerando como estrato o setor censitário. Essa técnica permite a seleção de uma determinada fração do total de domicílios de forma aleatória e independente em cada setor censitário, de forma que a seleção seja espalhada geograficamente por toda a extensão dos setores censitários e consiga representar todo o universo de domicílios e pessoas do País.

A partir da aplicação dos dois questionários citados, são montadas duas bases de dados distintas: a base de dados do Universo e a base dados da Amostra. A base de dados do Universo compreende as informações que são comuns a ambos os questionários, compreendendo o total de domicílios do país. A base de dados da Amostra contém as informações presentes apenas nos questionários da amostra, com dados bem mais detalhados sobre a parcela investigada dos domicílios e população.

Em termos de quantidade de registros, a base do Universo tem 190.755.799 de registros de pessoas e 67.569.688 de registros de domicílios [24]. Já a base da Amostra contabiliza 20.635.472 registros de pessoas e 6.192.332 registros de domicílios [25]. Cada um dos registros da base da Amostra tem associado um valor de peso amostral, devidamente calculado (após a fase de coleta ter sido encerrada) e balanceado para que a Amostra represente o total da população do país com alto grau de confiabilidade.

Os microdados<sup>1</sup> da base de dados do Universo não são disponibilizados ao público por

---

<sup>1</sup>Microdados são os dados coletados nos questionários das pesquisas. Cada questionário corresponde

questões de sigilo, apenas sendo liberados na forma de agregados por setor censitário [2]. Os microdados da base de dados da Amostra estão disponíveis para qualquer pessoa na área de downloads do Censo Demográfico 2010 [35] e o formato de disponibilização é através de arquivo texto posicional. O arquivo de layout dos dados pode ser obtido no mesmo local e através dele é possível extrair e trabalhar com os dados na ferramenta que melhor convier. O detalhes da extração e transformação da base de dados em questão são explicados na seção 5.2.

### 2.2.2 O BME - Banco Multidimensional de Estatísticas

O BME - Banco Multidimensional de Estatísticas é um sistema web que permite aos usuários cadastrados efetuarem consultas sobre a base de microdados de algumas pesquisas divulgadas pelo IBGE, inclusive o Censo Demográfico 2010. Nesse sistema são oferecidas ferramentas voltadas à busca, recuperação e manuseio das informações estatísticas, possibilitando que os usuários efetuem cruzamentos personalizados entre as diversas variáveis de uma pesquisa, obtendo resultados agregados para um ou mais níveis territoriais [33]. O usuário não fica limitado ao que é publicado pelo IBGE, ele pode produzir informações que atendam às suas necessidades individualizadas de análise.

A interface do sistema BME conta ainda com a metainformação<sup>2</sup> referente às variáveis presentes nas pesquisas. Conforme o usuário efetua a navegação pelas variáveis disponíveis, a metainformação associada é exibida em um painel. Isso permite que o usuário entenda os conceitos daquela variável e realize as consultas com maior segurança e agilidade.

É importante ressaltar que o usuário não tem acesso direto à base de microdados. A interação ocorre apenas por meio do sistema, que possui ferramentas capazes de garantir o sigilo da informação estatística, conforme abordado na seção 2.1.1. A base de dados do BME possui cerca 2 bilhões de registros de informações sobre o Brasil, distribuídos entre as diversas pesquisas realizadas pelo IBGE. Deste quantitativo, o Censo Demográfico 2010 contribui com aproximadamente 285 milhões de registros.

O público alvo da aplicação é constituído pelos próprios analistas do IBGE, pesquisadores e cientistas de universidades, institutos, jornalistas, além de órgãos dos governos

---

a um ou mais registros de informações, sendo a informação mais desagregada possível em uma pesquisa estatística.

<sup>2</sup>Metainformação pode ser entendida como a “informação sobre a informação”. Pode descrever dados, processos e ferramentas envolvidas na produção e uso dos dados estatísticos, tendo fundamental importância para que haja um correto entendimento das variáveis da pesquisa.

federal, estaduais e municipais, como forma de subsidiar as ações de profissionais em atividades de planejamento e tomadas de decisão.

Até fevereiro de 2018, o BME não era um sistema aberto ao público em geral. Seu acesso era permitido mediante a contratação de uma assinatura paga ou através de um convênio com instituições de ensino ou órgãos de governo. Entretanto, o IBGE sempre almejou a liberação completa e gratuita desse acervo e, para isso, o ampliou substancialmente seu parque computacional e realizou investimentos em infraestrutura de rede e segurança para suportar o aumento da quantidade de acessos. Hoje qualquer cidadão pode obter acesso imediato ao BME mediante o preenchimento de um simples cadastro online.

O BME possui, até o momento, 2.922 usuários cadastrados no sistema, sendo que 1.386 usuários se cadastraram desde que o sistema foi liberado de forma gratuita. Esse número representa 47% do quantitativo total de usuários do sistema e demonstra o interesse da sociedade pelos dados disponibilizados pelo BME. No período entre Novembro de 2017 e Novembro de 2018 o sistema foi acessado 81.723 vezes, sendo 69.700 acessos de usuários públicos (que não realizam consultas, apenas visualizam a metainformação e variáveis das pesquisas) e 12.023 acessos de usuários cadastrados. Um total de 28.940 consultas foram feitas no período por 705 usuários distintos. O usuário que mais executou consultas nesse período, executou 1.461 consultas e o máximo de consultas executadas por um usuário em um dia foi de 198. Os registros de *log* do sistema mostram picos de acesso, em geral, na época dos treinamentos oferecidos ao público nas unidades do IBGE em todas as capitais brasileiras e nos dias em que ocorrem as divulgações de pesquisas e subsequentes.

#### 2.2.2.1 Consultas *ad-hoc* e cálculos em tempo real

As consultas realizadas sobre a base do BME são de características imprevisíveis (*ad-hoc*), pois o usuário tem à disposição todas as variáveis da pesquisa. Analisando os *logs* de consultas dos últimos dois anos é possível perceber que, durante uma sessão de acesso, o usuário executa diversas consultas consecutivas até conseguir obter a informação desejada. A análise de um resultado de uma consulta ajuda o usuário a identificar que outros filtros e operações devem ser feitas, quais variáveis devem ser incluídas ou retiradas e qual o nível de desagregação do recorte territorial é o que melhor atende às suas necessidades. O usuário, aos poucos, refina sua análise dos dados até chegar ao resultado desejado.

Conforme já dito, o usuário não tem acesso direto a base de microdados. Ele também não é responsável por criar (diretamente) as consultas SQL necessárias ao atendimento

das suas análises. O sistema transforma toda a seleção de variáveis, filtros, definição de cálculos, variáveis derivadas e classificações dinâmicas realizados no ambiente gráfico em código SQL, de forma transparente e sem acesso por parte do usuário. Em seguida, as consultas necessárias são executadas e o resultado apresentado de forma amigável na interface do navegador. O usuário também pode optar por visualizar o resultado em forma de cartograma ou baixar os dados em formato *.csv* para ser trabalhado por outro programa.

As consultas do BME são executadas em dois passos conforme pode ser visto na Figura 2.2. O Passo 1 da consulta é utilizado pelo sistema para realizar a seleção das variáveis escolhidas pelo usuário, aplicar os filtros sobre as variáveis (quando necessário), realizar *joins* entre tabelas de fatos e definir a cláusula de *GROUP BY*. O Passo 2 é responsável por realizar *joins* da tabela intermediária obtida no Passo 1 com as dimensões das variáveis categorizadas, dimensões geográficas e temporal presentes na consulta para obter e apresentar seus respectivos descritores ao usuário.

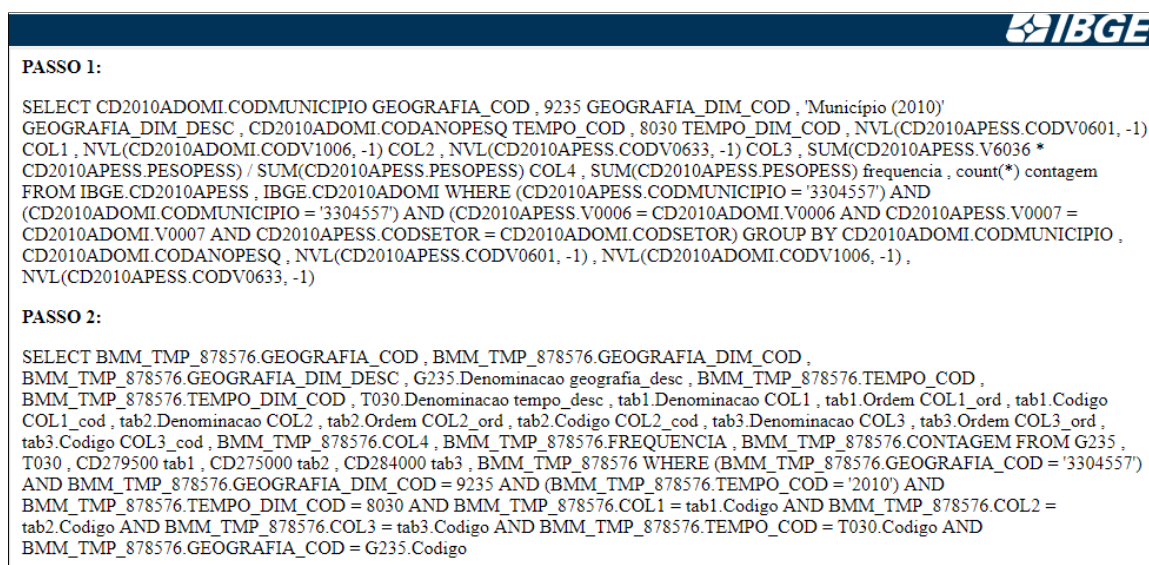


Figura 2.2: SQL de uma consulta do sistema BME

Para os experimentos desta dissertação, as consultas foram modificadas para serem realizadas em um único passo, mesclando o passo 1 com o passo 2. Algumas funções específicas do SGBD comercial utilizado pelo BME foram suprimidas ou trocadas por funções equivalentes presentes no PostgreSQL.

Em situações com maior demanda pelo acesso ou em casos de consultas extremamente complexas (como determinadas consultas sobre as bases dos Censos Demográficos), o sistema pode sugerir, ou mesmo impor, dependendo do caso, o agendamento da consulta. Dessa forma, consultas menos complexas não sofrem degradação de tempo no processa-

mento e podem ser prontamente atendidas. As consultas agendadas são executadas em lote durante a noite, quando a demanda pelo uso do sistema é mais baixa e são encaminhadas em formato *.csv* para o e-mail cadastrado do usuário.

Fica claro com este cenário, que, dependendo da pesquisa escolhida e do nível geográfico da consulta, milhões de registros terão que ser analisados, combinados e agregados para produzir o resultado demandado pelo usuário. Por exemplo, somente no Censo Demográfico 2010, contabilizando as bases do Universo e da Amostra, são mais de 285 milhões de registros de dados presentes nas tabelas de fatos.

#### 2.2.2.2 A base do Censo Demográfico 2010 no BME

O modelo do Censo Demográfico 2010 está estruturado na base do BME seguindo o esquema estrela [41] do *data warehouse*. É composto pelas tabelas de fatos, dimensões físicas específicas de cada tabela de fato, uma dimensão temporal e um conjunto de dimensões geográficas. A dimensão temporal e as dimensões geográficas são comuns a todas as tabelas de fatos. Os dados das bases do Universo - CD2010U e da Amostra - CD2010A estão alocados em tabelas de fato distintas e possuem dimensões independentes. Para efeito dessa dissertação vamos descrever apenas a estrutura da base da Amostra.

As tabelas de fato que compõem a base da Amostra são: CD2010ADOMI (dados de Domicílios), CD2010APESS (dados de Pessoas), CD2010AMORT (dados de Mortalidade), CD2010AEMIGR (dados de Emigração) e CD2010AFAMI (dados de Famílias). Todas elas se relacionam com a dimensão Temporal T030, que contém a data de referência da pesquisa e com as dimensões Geográficas G000, G031, G032, G233, G234, G235, G236, G237, G238, G239, G242, G246, cada uma contendo as unidades territoriais do nível geográfico que representam. Embora exista relacionamento entre as tabelas de fatos e todas as tabelas de dimensões geográficas e temporal, não existe relacionamento entre as tabelas de fatos e todas as demais dimensões. A relação das tabelas de fato com as suas respectivas tabelas de dimensões físicas, assim como o detalhamento das dimensões geográficas e temporal, estão descritos no Apêndice A.

As tabelas de dimensão são nomeadas seguindo o seguinte padrão para facilitar sua identificação no modelo: dimensões geográficas têm seu nome iniciado pela letra “G”; dimensões temporais pela letra “T”; e as demais dimensões pelas letras “CD”. Todas as iniciais são seguidas por um código sequencial determinado no momento de sua criação.

A tabela de fatos CD2010ADOMI tem relacionamento 1:N com as tabelas de fatos

CD2010APESS, CD2010AFAMI, CD2010AMORT e CD2010AEMIGR. A tabela de fatos CD2010AFAMI tem relacionamento 1:N com a tabela de fatos CD2010APESS e 1:N com CD2010ADOMI. A tabela de fatos CD2010APESS tem relacionamento N:1 com as tabelas de fatos CD2010AFAMI e CD2010ADOMI. As tabelas de fatos CD2010AMORT e CD2010AEMIGR têm relacionamento N:1 com a tabela de fatos CD2010ADOMI e não se relacionam com as demais tabelas de fatos.

A tabela de fatos CD2010ADOMI se relaciona com as tabelas CD2010APESS, CD2010AFAMI, CD2010AMORT e CD2010AEMIGR através do atributo de controle **V0300**. A tabela de fatos CD2010APESS se relaciona com a tabela CD2010AFAMI através dos atributos **V0300** e **V5020**, e vice-versa. A Figura 2.3 mostra o relacionamento entre as tabelas citadas. As dimensões geográficas e temporal têm relacionamento 1:N com cada tabela de fatos. A junção das tabelas de fatos com suas dimensões e com as dimensões temporal e geográficas se dá pelo atributo **CODIGO** de cada uma delas. A Figura 2.4 mostra o relacionamento entre as tabelas de fatos e as dimensões geográficas e temporal, bem como a cardinalidade de cada tabela.

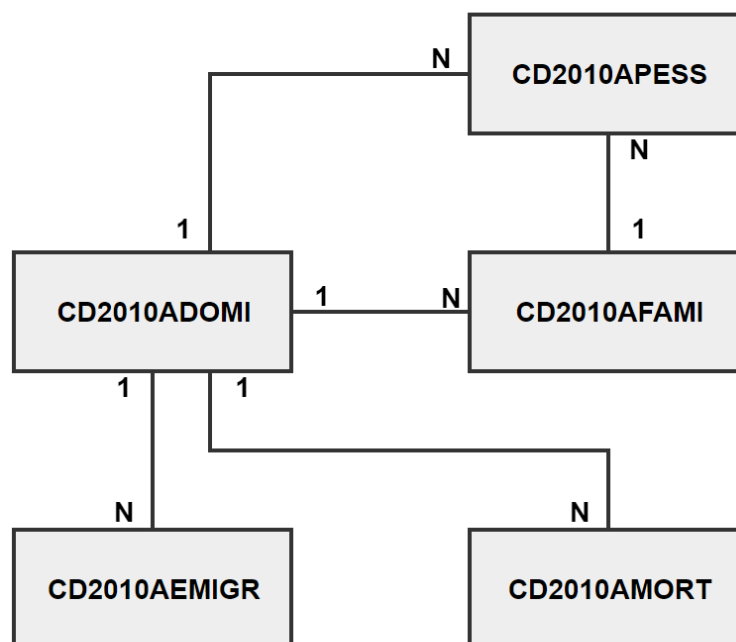


Figura 2.3: Relacionamento entre as tabelas de fatos CD2010ADOMI, CD2010APESS, CD2010AFAMI, CD2010AMORT e CD2010AEMIGR

As tabelas de dimensão geográficas relacionam-se entre si formando a hierarquia geográfica da pesquisa. O relacionamento entre as dimensões geográficas do maior nível de agregação para o menor ocorre da seguinte forma: a dimensão que representa o maior

nível de agregação (País) tem relacionamento 1:N com a dimensão que representa um nível de agregação abaixo (Regiões Geográficas); essa dimensão tem relacionamento 1:N com a dimensão que representa um nível de agregação abaixo (Unidades da Federação), e assim sucessivamente, até o relacionamento 1:N entre as dimensões que representam Área de Ponderação e Setor Censitário (que é o menor nível de agregação).

O tamanho da base de dados CD2010A é de aproximadamente 32,8GB, levando em conta o espaço ocupado por dados e índices definidos sobre os atributos. A tabela 2.1 exhibe, de forma detalhada, o espaço ocupado por cada uma das tabelas de fatos relacionadas.

Tabela 2.1: Tamanho das tabelas de fatos da base CD2010A

<b>Tabela</b>	<b>Número de linhas</b>	<b>Tamanho dos dados (MB)</b>	<b>Tamanho dos índices (MB)</b>	<b>Tamanho total (MB)</b>
CD2010ADOMI	6.192.332	2.359,13	729,13	<b>3.088,26</b>
CD2010AEMIGR	53.777	13,98	17,25	<b>31,23</b>
CD2010AFAMI	5.813.134	1.208,92	609,63	<b>1.818,55</b>
CD2010AMORT	111.555	22,95	22,25	<b>45,20</b>
CD2010APESS	20.635.472	23.072,44	4.599,56	<b>27.672,00</b>
<b>Total</b>	<b>32.806.270</b>	<b>26.677,42</b>	<b>5.977,82</b>	<b>32.655,24</b>

### 2.2.3 O SIDRA - Sistema IBGE de Recuperação Automática

O SIDRA descreve e divulga os planos tabulares contendo os dados agregados das pesquisas realizadas pelo IBGE. Os dados disponíveis na base do SIDRA são previamente calculados e combinados por diversos processos de tabulação (o BME-TABULADOR é um deles) a partir dos microdados coletados por cada pesquisa do IBGE. Esse processo será detalhado na seção 2.2.4.

Uma consulta à base de dados do SIDRA, consolidada ao longo de mais de três décadas, permite ao usuário, de forma personalizada, a seleção e a visualização dos dados tabulares, a geração de gráficos e rankings, além de mapas e cartogramas.

No SIDRA os dados estão organizados em tabelas e, em cada uma delas, são apresentados dados agregados associados a diversos recortes territoriais, ocorrências temporais e, na maioria das vezes, a um conjunto de classificações e suas categorias. Podemos pensar em cada tabela do SIDRA como um cubo de dados, no qual o usuário tem a liberdade de trabalhar da maneira que desejar, realizando cruzamentos de determinadas características

com os valores apurados na pesquisa e detalhando a informação até o limite disponível.

Para um melhor entendimento, vamos conceituar alguns termos utilizados no SIDRA [38]:

- Pesquisa: representa a fonte dos dados, ou seja, de onde provêm os dados das diversas tabelas armazenadas na base de dados;
- Plano Tabular: um conjunto das tabelas de uma pesquisa agrupadas por afinidade ou tema específico;
- Tabela: é a unidade de consulta do SIDRA. Cada tabela é identificada por um número único e é composta de variáveis e classificações;
- Variável: descreve o tipo de dado que a tabela possui. Uma mesma tabela pode possuir mais de uma variável. Ex: “Valor do rendimento médio domiciliar”, “Número de pessoas com 10 anos ou mais de idade”, etc.;
- Classificação: utilizada para qualificar as variáveis de uma tabela, tornando o seu significado mais específico. Ex: a variável “Valor do rendimento médio domiciliar” pode ser classificada por “Situação do domicílio”, pela “Existência ou não de energia elétrica” e pela “Forma de abastecimento de água no domicílio”;
- Categoria: cada um dos possíveis valores que uma classificação pode assumir. Ex.: a classificação “Situação do domicílio” possui as categorias “urbano” e “rural”;
- Ocorrência temporal: a unidade de tempo a qual os dados da tabela se referem. Ex. “ano de 2010”, “1º trimestre de 2015”, etc.;
- Nível Territorial: cada um dos diferentes tipos de divisão territorial espacialmente associados aos dados de uma tabela. Ex: “unidade da federação”, “região geográfica”, “município”, etc.;
- Unidade Territorial: cada um dos componentes de um nível territorial. Ex: para o nível territorial “unidade da federação”, temos Pernambuco, Rio de Janeiro, São Paulo, etc.;

As tabelas são concebidas de modo a agrupar certas variáveis da pesquisa, proporcionando aos usuários uma visão específica de análise dos dados. Como os dados são previamente calculados e tratados, os usuários dispõem de um acesso rápido e gratuito ao sistema, que é livre para qualquer pessoa através da internet.



### 2.2.4 O sistema BME-TABULADOR e o processo de Tabulação de Dados

O processo de tabulação de dados tem uma enorme importância dentro do IBGE, e para a população como um todo, visto que as divulgações das pesquisas do Instituto somente são possíveis através dele. O processo consiste em agregar os dados mais básicos de uma pesquisa, os microdados, sob diferentes visões temáticas de acordo com tabelas previamente definidas pelos especialistas do IBGE. Os dados agregados são enviados ao SIDRA, para serem carregados e divulgados.

O sistema BME-TABULADOR é a ferramenta responsável pelo processo de tabulação e integração da base do BME com o SIDRA. A partir do repositório de microdados do BME, os analistas do IBGE são capazes de suprir a demanda por dados de cada tabela definida no SIDRA, para uma determinada pesquisa. Dependendo dos tipos de cálculos que necessitem ser realizados, mais de uma consulta SQL precisa ser gerada para atender aos diversos níveis territoriais presentes na tabela do SIDRA. As consultas necessárias são geradas pelo sistema de forma automática, após uma etapa inicial de preenchimento de dados para correlacionar as variáveis das duas bases.

O resultado desse processo é um conjunto de arquivos texto, em um formato pré-definido pelo SIDRA, com as informações necessárias para o carregamento completo, ou parcial, de uma tabela de dados agregados, com suas classificações, categorias e variáveis quantitativas (cálculos). O SIDRA trabalha com um padrão próprio de armazenamento de dados que pode fazer com que uma única tabela necessite de arquivos com tamanho superior a 2GB para ser carregada.

A tabulação dos dados para o SIDRA pode ser realizada com o auxílio de diversas ferramentas: consultas SQL, programas SAS, scripts REDATAM ou qualquer outra que melhor atender às demandas da pesquisa. Para fins dessa dissertação, vamos tratar especificamente processo executado através de consultas SQL geradas pelo sistema BME-TABULADOR.

#### 2.2.4.1 O *case* do Censo 2010 - Dados da Amostra

O processo de tabulação do Censo Demográfico 2010 para o SIDRA foi a primeira utilização real do sistema BME-TABULADOR. O volume de dados envolvido foi de cerca 285 milhões de registros de domicílios, pessoas, famílias, mortalidade e emigração, nas bases de dados do Universo e da Amostra. Um volume de dados dessa ordem de grandeza

torna o processo de tabulação um trabalho extremamente complexo, visto que envolve a análise de cada um dos milhões de registros armazenados no banco de dados. Além disso, como dito anteriormente, cada tabela apresenta uma visão particular sobre os dados, então, é necessária uma, ou até mais, operações desse tipo por tabela. Isso multiplica a complexidade da tarefa em algumas dezenas de vezes.

O tempo de execução das consultas da tabulação pode variar de alguns minutos, a algumas horas, dependendo da complexidade da tabela do SIDRA em questão e dos níveis territoriais definidos. Tabelas que possuem somente cálculos de frequência ou somatórios, em geral, são mais rápidas de serem processadas e só precisam ser geradas para o nível territorial mais desagregado, pois o SIDRA consegue agregar os dados para os níveis superiores (por exemplo a partir do nível município, é possível agregar os dados para unidades da federação, regiões geográficas e Brasil). Por outro lado, tabelas que possuem cálculos de médias, medianas, intervalos de confiança, etc., devem ser processadas uma vez para cada nível territorial, pois nesse caso a agregação posterior pelo SIDRA não é possível. Os cálculos têm necessariamente que serem feitos a partir do microdado. Este segundo cenário é claramente mais oneroso em termos de quantidade de consultas e tempo de processamento que o primeiro.

#### **2.2.4.2 Comparativo entre consultas do BME e BME-TABULADOR**

Tanto o sistema BME quanto o BME-TABULADOR utilizam o banco de dados do BME para executar as consultas, porém elas possuem naturezas distintas. O processo de tabulação trabalha com consultas muito mais custosas do que as executadas rotineiramente pelos usuários do BME. Uma consulta de tabulação, em geral, requer muito mais combinações dos atributos da consulta, e não possui predicados altamente restritivos.

Na maioria das vezes, as consultas do processo de tabulação não possuem filtros. Não é aplicado, por exemplo, um filtro para um município específico, ou para uma característica da pessoa que restrinja significativamente o espaço de busca na base (por exemplo, pessoas entre 20 e 35 anos, do sexo feminino e que sejam solteiras). Os filtros das consultas da tabulação, quando existentes, são sempre pouco restritivos (por exemplo, pessoas de 10 anos ou mais de idade).

Outro ponto, e talvez o mais importante, é que no BME o usuário seleciona as variáveis de interesse e executa a consulta com a combinação de todas elas, logo essas consultas possuem uma única cláusula *GROUP BY*. No BME-TABULADOR, por outro lado, as consultas possuem de 1 a *n* conjuntos de grupos, cada um com uma determinada combi-

nação dos atributos selecionados em uma única consulta SQL. Isso é obtido através das cláusulas *GROUP BY* e *GROUPING SETS*, de maneira que as combinações demandadas por cada tabela do SIDRA sejam precisamente descritas na consulta.

Por exemplo, a tabela SIDRA 200 [26] possui informações sobre a população residente, agrupadas pelas classificações de sexo, situação do domicílio e grupos de idade. Para alimentar essa tabela de maneira correta, deve-se executar uma consulta com todas as possíveis combinações de *clusters* dos atributos selecionados:

1. somente por sexo;
2. somente por situação do domicílio;
3. somente por grupos de idade;
4. por sexo e situação do domicílio, sem considerar o atributo grupos de idade;
5. por sexo e grupos de idade, sem considerar o atributo situação do domicílio;
6. por situação do domicílio e grupos de idade, sem considerar o atributo o sexo;
7. por sexo, situação do domicílio e grupos de idade; e
8. pela geografia da tabela, sem considerar nenhum atributo (grande total da tabela).

O cenário pode ficar ainda mais complexo caso algumas dessas classificações possuam categorias nas quais ocorram sobreposição de dados. Por exemplo se a classificação de grupos de idade contiver as categorias “10 anos”, “11 anos”, “12 anos”, “13 anos”, “14 anos” e “10 a 14 anos”, ela terá que ser dividida em duas “subclassificações” no momento da execução da consulta. Com isso o número de combinações de *clusters* possíveis aumenta de forma significativa, conforme será observado na descrição das consultas selecionadas.

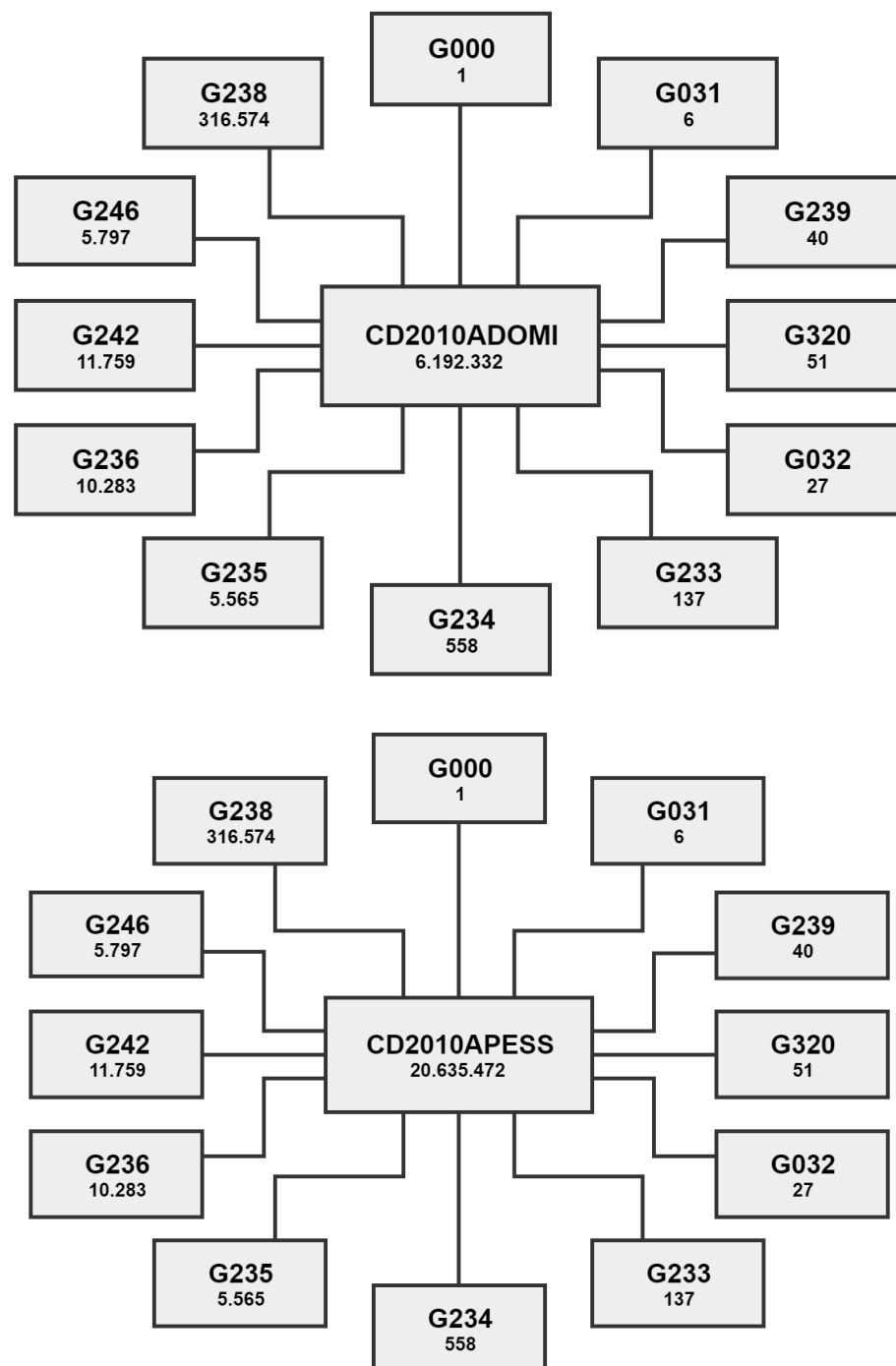


Figura 2.4: Relacionamento entre as tabelas de fatos CD2010ADOMI e CD2010APESS e suas respectivas dimensões geográficas e temporal, com indicativo da quantidade de registros presentes em cada tabela

# Capítulo 3

## ParGRES

Esta seção apresenta o ParGRES [44, 45, 17], o *middleware* que foi estendido neste trabalho. O ParGRES é uma solução de *cluster* de banco de dados que atua como um *middleware* entre a aplicação cliente e um *cluster* de SGBDs sequenciais, sendo capaz de coordenar o acesso a cada instância de SGBD presente no *cluster* e proporcionar paralelismo nas consultas [51]. O ParGRES foi desenvolvido para atender o processamento de consultas pesadas, típicas de aplicações OLAP [50]. O paralelismo é obtido através da replicação completa da base de dados em cada instância presente no *cluster*, em conjunto com a técnica de particionamento virtual adaptativo (*Adaptive Virtual Partitioning* - AVP) [42, 18].

A arquitetura do ParGRES é distribuída e foi implementada de forma que seu uso seja transparente para as aplicações clientes, que não necessitam alterar a forma de executar as consultas, tampouco conhecer a organização física do *cluster* de banco de dados acessado. A Figura 3.1 apresenta uma visão geral dessa arquitetura.

A Figura 3.1 mostra a aplicação cliente se conectando a interface do ParGRES para submeter uma consulta. Após a análise, o ParGRES determina qual método de processamento será aplicado com fins de acelerar a execução da consulta. Pode ocorrer o processamento através do paralelismo inter- ou intra-consulta [51]. Se o método escolhido for o paralelismo inter-consulta, o ParGRES envia a consulta (em sua totalidade) para processamento em um dos nós do *cluster*; se for intra-consulta, o ParGRES divide a consulta em várias sub-consultas, através da técnica de particionamento virtual (*Virtual Partitioning* - VP), e submete cada uma delas para um nó diferente [55]. Todos os nós do *cluster* são envolvidos nesse processo. Ao final dessa etapa, o ParGRES agrega os resultados obtidos por cada nó e devolve para a aplicação.

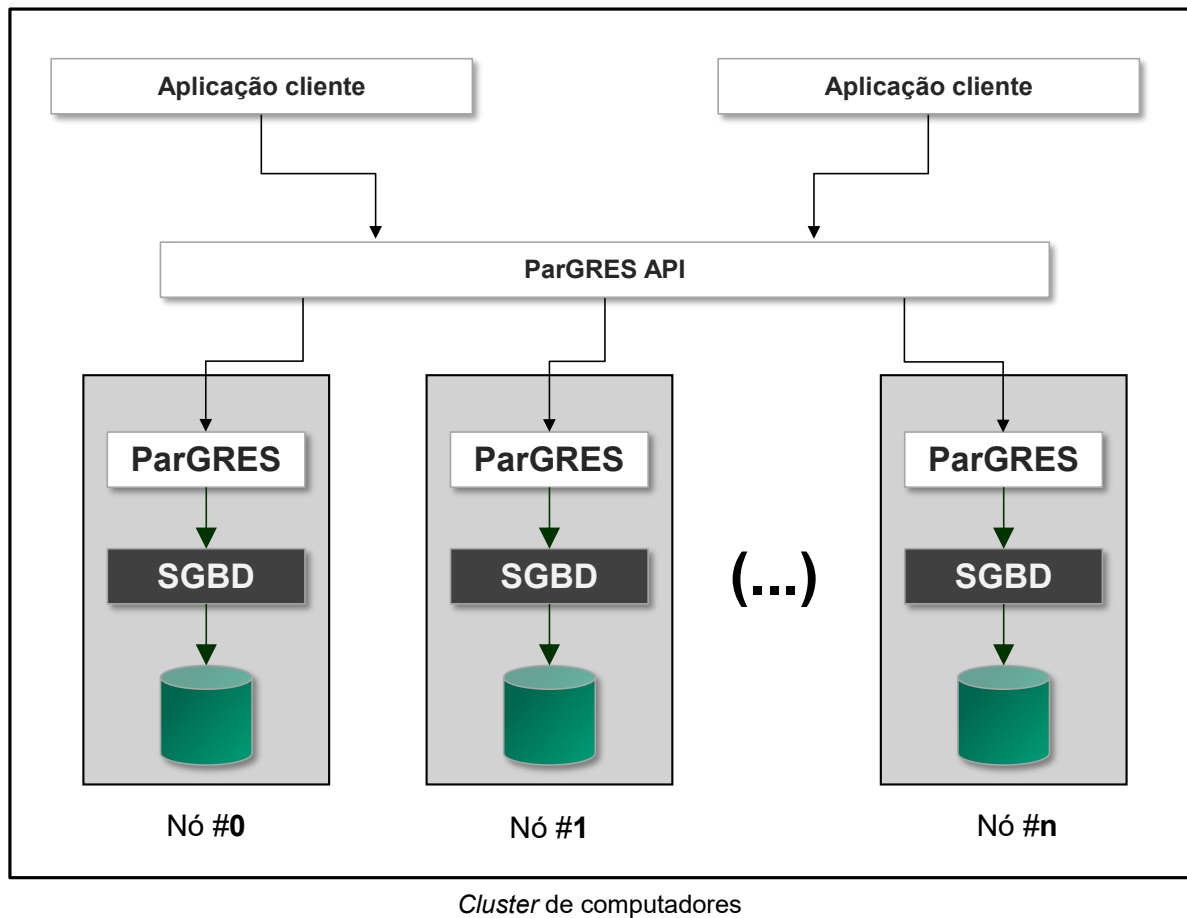


Figura 3.1: Arquitetura do ParGRES adaptada de [51]

A técnica de VP consiste em reescrever a consulta original adicionando predicados referentes a intervalos de valores sobre o atributo de particionamento virtual (*virtual partitioning attribute* - VPA) definido. O número de faixas (e sub-consultas) geradas é determinado pela quantidade de nós do *cluster*. Cada nó recebe uma sub-consulta diferente e, como o banco de dados é completamente replicado, a alocação pode ocorrer para qualquer nó [55]. Abaixo apresentamos um exemplo de uma consulta  $Q$  sobre uma tabela “tab\_1” e em seguida a consulta  $Q1$  reescrita pelo ParGRES:

```
Q: SELECT col_a, col_b, sum(col_c)
FROM tab_1
WHERE col_a = 'S'
```

```
Q1: SELECT col_a, col_b, sum(col_c)
FROM tab_1
WHERE col_a = 'S'
AND col_d >=: v1 AND col_d <: v2
```

Neste exemplo, “col\_d” foi definida com o atributo de particionamento virtual para a tabela “tab\_1”. O par de valores “v1” e “v2” é definido pelo ParGRES em função da quantidade de nós presentes no *cluster* e da faixa de domínio do VPA. Por exemplo, se o intervalo de “col\_d” for [1, 4000] e a quantidade de nós presentes no *cluster* do ParGRES for 8, o intervalo atribuído a cada nó pode ser representado da seguinte forma: n1: v1=1 e v2=501; n2: v1=501 e v2=1001; n3: v1=1001 e v2=1501; (...) n8: v1=3501 e v2=4001. Em cada nó ocorrerá o processamento sequencial da sub-consulta responsável pelo intervalo de tuplas designado pelo ParGRES. Dessa forma um SGBD convencional e não particionado fisicamente consegue proporcionar um processamento de consultas com paralelismo intra-consulta.

### 3.1 Componentes do ParGRES

Cada instância do ParGRES (representada pelos retângulos branco na Figura 3.1) é distribuída em todos os nós que compõem o *cluster* de banco de dados. A Figura 3.2 mostra os componentes do ParGRES com mais detalhes. O *Cluster Query Processor* - CPQ só é instanciado em um nó, enquanto os componentes locais presentes em cada nó são o *Node Query Processor* - NQP e o SGBD [50].

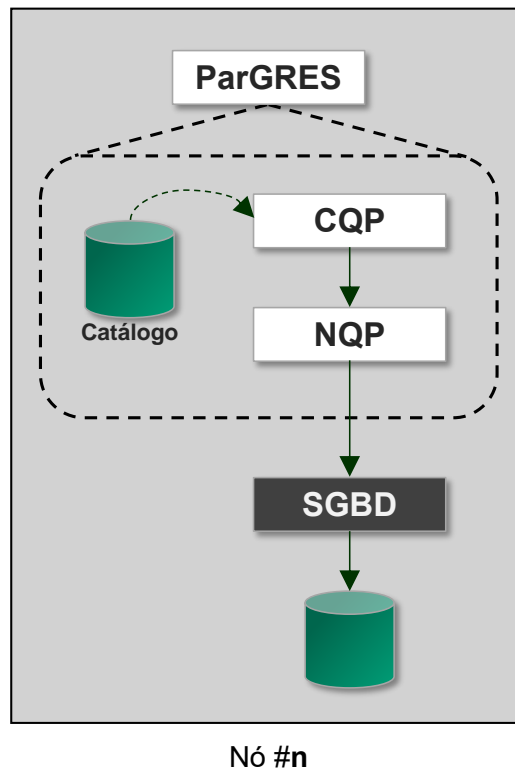


Figura 3.2: Componentes do ParGRES

O CQP é o componente mais importante do ParGRES, pois é responsável por realizar a divisão inicial da consulta em sub-consultas, alocar e coordenar o trabalho de todas as instâncias NQPs em execução [51, 55] no nós. Em seguida ele deve agregar os resultados individuais obtidos por cada NQP antes de devolvê-los à aplicação cliente. O CQP é executado por apenas um nó do *cluster* e deve ter acesso a todos os demais nós. O NQP é responsável pela execução local das consultas atribuídas a cada nó junto ao SGBD e por ajudar o CQP com o balanceamento de carga entre os nós do *cluster* [51].

O ParGRES executa basicamente 4 tipos de tarefas: reescrita de consultas SQL para proporcionar o paralelismo; processamento de consultas com paralelismo inter/intra-consultas; composição/agregação de resultados; e processamentos de atualizações (*updates*).

### 3.1.1 Reescrita de SQL

A reescrita de SQL é feita pelo Tradutor, componente presente no CQP e que consiste em um analisador sintático da gramática do SQL-99, porém nem todos os comandos são suportados. A informação gerada pelo Tradutor inclui a identificação das tabelas, operações e atributos presentes na consulta original, de forma que seja possível determinar quais transformações devem ser aplicadas na consulta para que o paralelismo possa ser alcançado e a composição final do resultado ocorra de forma correta [51].

### 3.1.2 Processamento de Consultas com paralelismo Inter- ou Intra-consultas

Após a etapa de interpretação da consulta, o CQP decide que estratégia de paralelismo utilizar: inter ou intra-consulta. A decisão é tomada com base em informações presentes no seu Catálogo, formado pelas configurações iniciais feitas pelo mantenedor do *cluster*, acrescidas de outras coletadas automaticamente no momento de inicialização dos serviços do ParGRES. Essas informações permitem ao CQP definir a melhor forma de executar uma consulta em um dado momento, levando em conta o custo da consulta e as demais em execução no ambiente [51].

A estratégia de paralelismo intra-consulta é implementada através da decomposição de consultas complexas em sub-consultas para serem executadas em paralelo em diferentes nós do *cluster*, conforme pode ser visto na Figura 3.3. Já na estratégia de paralelismo inter-consulta, representada na Figura 3.4, consultas distintas são executadas (de forma



completa) simultaneamente no *cluster* de SGBDs, uma em cada nó. A implementação desta forma de paralelismo é bastante simples: O CQP identifica o processo NQP que está com o menor número de tarefas pendentes no momento e envia a consulta para o respectivo nó. O NQP submete a consulta para o SGBD, aguarda a final do processamento, retorna o resultado para o CQP que, por sua vez, devolve o dado para a aplicação solicitante.

O paralelismo intra-consulta reduz o tempo individual de processamento de cada consulta, sendo ideal para ambientes OLAP de alto desempenho. Por outro lado, o paralelismo inter-consulta melhora o desempenho global do sistema, sendo ideal para aplicações OLTP e não adequado para consultas OLAP.

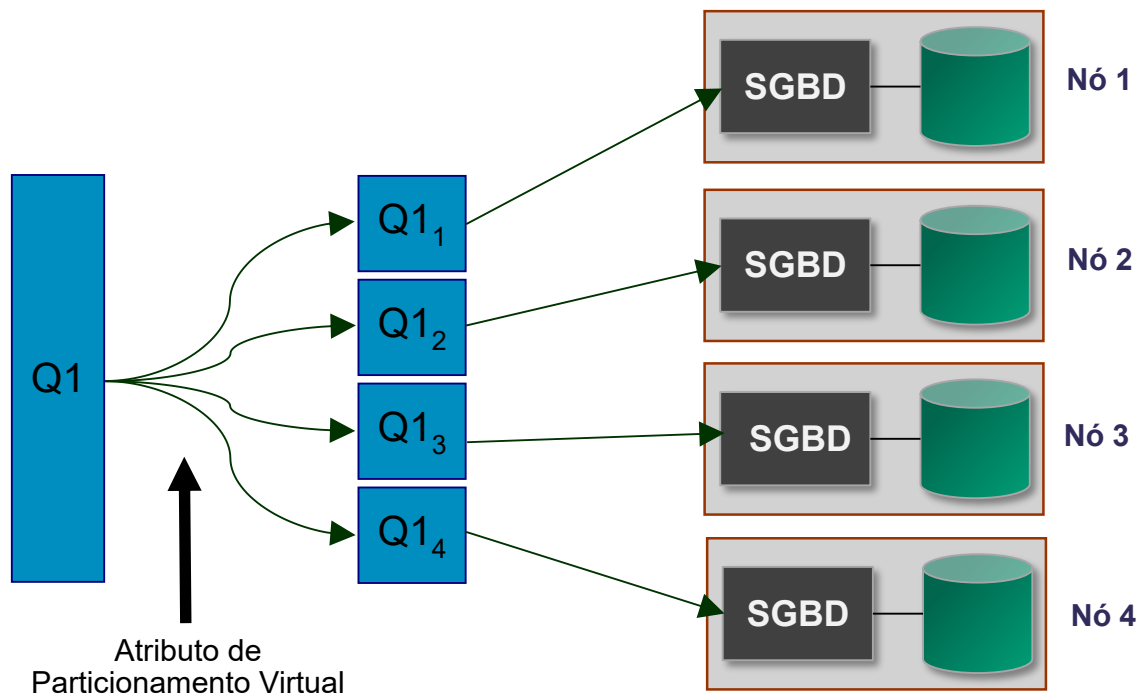


Figura 3.3: Paralelismo Intra-Consulta

### 3.1.3 Composição/agregação de resultados

A composição de resultados implementada no ParGRES é uma adaptação do algoritmo de agregação em duas fases proposto por Shatdal e Naughton [60]. Na primeira fase, os nós do *cluster* agregam os resultados das sub-consultas locais, enquanto na segunda fase, os resultados são distribuídos aos demais nós através de uma função *hash* [51]. Por fim, cada nó envia seu subconjunto de resultado para o CQP, que executa as tarefas necessárias para a obtenção do resultado unificado.

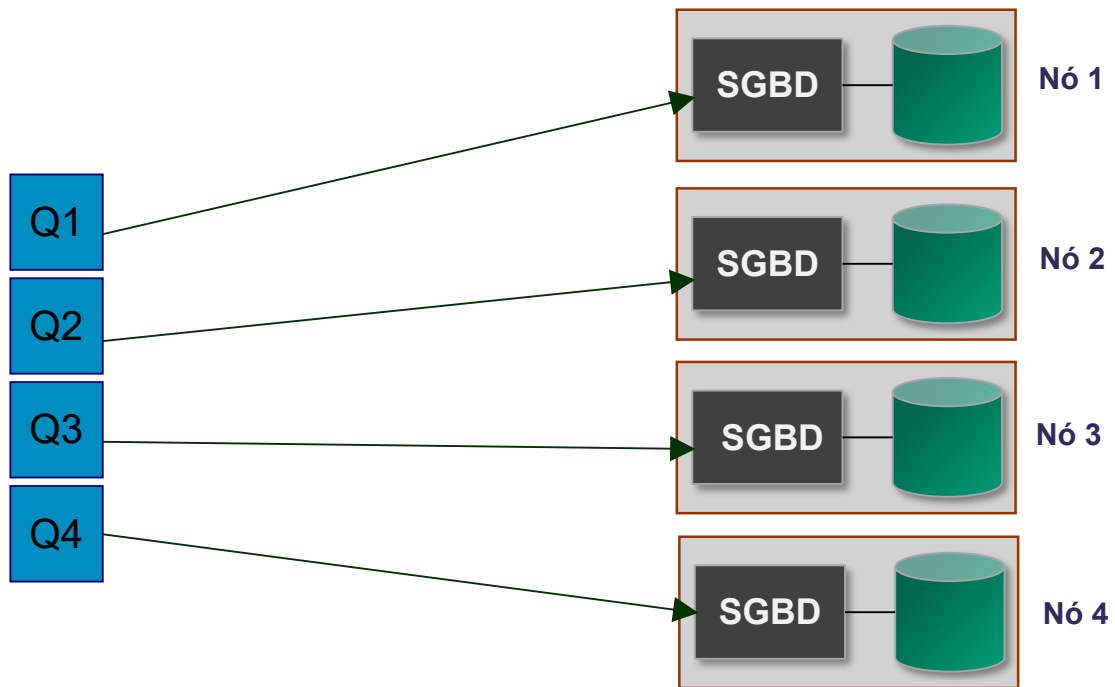


Figura 3.4: Paralelismo Inter-Consulta

### 3.1.4 Processamentos de atualizações

O foco principal do ParGRES é a execução de consultas para a leitura de dados da base, porém comandos de atualização também são apoiados. Operações de atualizações em ambiente OLAP não são frequentes e, quando necessárias, são realizadas em horários específicos para não impactar a performance global do sistema. Em um cenário de ambiente distribuído e com replicação de dados, um cuidado extra é necessário para que as consultas não sejam executadas sobre uma base de dados inconsistente [50]. Para atender a essa situação, o ParGRES adota uma política de consistência forte, não permitindo a execução concorrente de atualizações e consultas de leitura. Quando uma operação de atualização é submetida ao ParGRES, o CQP, através de seu escalonador, impede que novas consultas de leitura sejam postas em execução e aguarda que todas as consultas já iniciadas terminem. Somente então o CQP envia a atualização para cada nó NQP que compõe o ambiente, garantindo assim a consistência da base. Após todos os nós terem confirmado a conclusão da atualização, o CQP permite a execução de novas consultas no *cluster*. A partir do momento que só existam consultas de leitura presentes no escalonador do CQP, elas são executadas em paralelo [42].

## Capítulo 4

# C-ParGRES: Um middleware para Processamento de Consultas em Ambientes de Nuvem

O ParGRES foi desenvolvido como um *middleware* para ser executado em *clusters* locais [51]. Paes *et al.* [55] demonstraram o sucesso desta implementação utilizando o ambiente do GRID 5000 com os microdados do Censo Demográfico 2000 - Dados da Amostra. Em 2008 Kotowski *et al.* [42] fizeram uma proposta de implementação do ParGRES especializada para o ambiente de GRID. Nossa proposta foi dar prosseguimento a essa evolução do ParGRES fazendo a adaptação para o funcionamento em ambiente de nuvem. Embora, em uma primeira análise, as nuvens fossem consideradas inadequadas para aplicações que exigem alto desempenho, como o processamento paralelo de consultas OLAP [15, 39], outros estudos, como He *et al.* [21] fornecem evidências concretas de que a infraestrutura da nuvem está melhorando significativamente a escalabilidade e o desempenho.

A arquitetura do C-ParGRES é apresentada na Figura 4.1 e, como pode ser observado, foi desenhada como uma camada externa à própria arquitetura do ParGRES. O principal componente do C-ParGRES é o C-ParGRES *Starter* que atua como um gerenciador dos *clusters* virtuais no ambiente de nuvem da Amazon.

### 4.1 C-ParGRES *Starter*

C-ParGRES *Starter* é composto por dois módulos principais: o módulo *Deployment* e o módulo *Dimensioner*. O primeiro é responsável por tarefas relacionadas à criação, configuração, inicialização e finalização de máquinas virtuais. Na implementação atual do

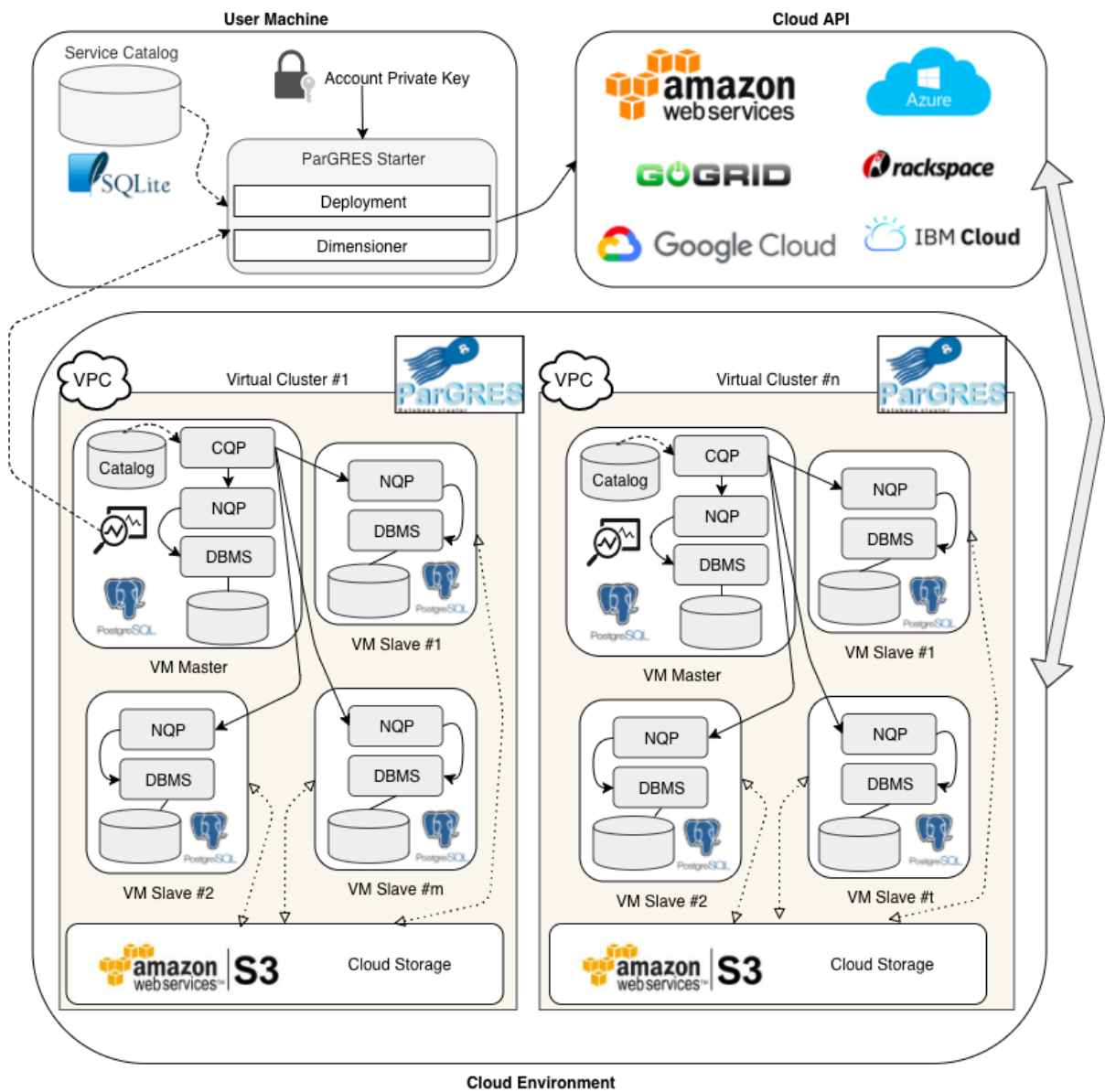


Figura 4.1: Arquitetura do C-ParGRES

C-ParGRES, ele configura o *cluster* virtual usando a API do Amazon AWS. No entanto, outros provedores de nuvem podem ser usados sem muito esforço.

O segundo é responsável por estimar a quantidade necessária de máquinas virtuais para implantar e monitorar as atividades do *cluster* e o tempo de conclusão das consultas para decidir se a criação de novas máquinas virtuais é necessária ou não. Ou, até mesmo, encerrar máquinas virtuais quando a carga de trabalho é baixa, ajudando a reduzir os custos financeiros com impacto mínimo no tempo de resposta das consultas. O *Dimensioner* é uma extensão da abordagem proposta por Coutinho *et al.* [12]. Ele analisa os *logs* das consultas e estima a quantidade necessária de máquinas virtuais (e os tipos) que devem ser implantados. A seguir, apresentamos com mais detalhes os módulos *Deployment* e *Dimensioner* do C-ParGRES *Starter*.

#### 4.1.1 Módulo *Dimensioner* do C-ParGRES *Starter*

O *Dimensioner* é um componente importante, pois, se a quantidade de máquinas virtuais alocadas no *cluster* virtual estiver sub ou superdimensionada, certamente causará impactos negativos no desempenho das consultas. Essa tarefa de alocação é muito complexa pois, em geral, os provedores de nuvem têm uma grande variedade de tipos de máquinas virtuais, cada uma associada a um custo financeiro e características de desempenho, projetados para atender às diferentes necessidades dos usuários.

O *Dimensioner* estende a abordagem proposta por Coutinho *et al.* [12], que é uma heurística baseada em GRASP [48] para executar aplicações independentes, e foi adaptado para o contexto do C-ParGRES. O *Dimensioner* recebe como parâmetro de entrada a quantidade estimada de Gflops para as consultas submetidas (que pode ser estimado consultando os *logs* das consultas), os tipos de máquinas virtuais que podem ser instanciadas para criar o *cluster* virtual e o tempo máximo de execução permitido. Escolhemos usar o GRASP, já que os procedimentos exatos e de força bruta muitas vezes se mostraram incapazes de encontrar soluções, pois são extremamente demorados, principalmente para problemas do mundo real como o processamento de consultas paralelas OLAP.

Similarmente à abordagem proposta por Coutinho *et al.* [12], o *Dimensioner* é composto por duas fases: uma fase de construção e uma fase de pesquisa local. A fase de construção encontra uma solução inicial (quantidade de máquinas virtuais a serem implantadas no *cluster* virtual) que pode ser posteriormente aprimorada pela fase de pesquisa local. Para descrever o *Dimensioner*, precisamos de alguma notação adicional. Definimos o conjunto de requisitos do usuário como o custo máximo  $CM$ , tempo máximo  $TM$ , arma-

zenamento em disco  $DS$ , capacidade de memória  $MC$  e uma demanda de processamento de  $Gf$  Gflops. Da mesma forma, cada tipo de máquina virtual  $p \in P$  tem um custo associado  $cp$  (isto é, o custo de implantação da máquina virtual para um quantum temporal) e recursos de computação como armazenamento em disco  $dp$ , capacidade de memória  $mp$  e um poder de processamento de  $gp$  Gflops por período de tempo (Gflopt). Agora definimos uma variável binária  $x_{pit}$  para cada  $p \in P$ ,  $i \in 1, \dots, NM$  e  $t \in T = 1, \dots, TM$ , tal que  $x_{pit} = 1$  se e somente se a máquina virtual  $i$  do tipo  $p$  for comprada no momento  $t$ ; caso contrário,  $x_{pit} = 0$ . Além disso, definimos a variável  $tm$  como o último período de tempo em que uma máquina virtual é instanciada pelo usuário.

Também definimos uma solução  $\{(p_1, i_1, t_1), (p_2, i_2, t_2), \dots\}$  como um conjunto de 3-tuplas  $(p, i, t)$  representando que a máquina virtual  $i$  do tipo  $p$  foi comprada no período  $t$ . Seja  $S$  o conjunto de todas as soluções possíveis. Uma solução  $s \in S$  é considerada viável se respeitar os requisitos definidos pelo usuário, isto é, restrições de tempo e de custo. Também denotamos  $tm(s)$  como o último período em que uma máquina virtual foi usada. Além disso, definimos uma função de custo  $F : S \rightarrow R$ , que mede a qualidade da solução. A função  $F$  tenta minimizar custos e tempo enquanto penaliza a inviabilidade em relação ao tempo máximo. Os detalhes sobre o processo de otimização usado no *Dimensioner* estão descritos em Coutinho *et al.* [12].

#### 4.1.2 Módulo de *Deployment* do C-ParGRES Starter

O módulo de *Deployment* de C-ParGRES Starter foi desenvolvido na linguagem Java e com o uso do conjunto de bibliotecas AWS SDK for Java disponibilizado pela Amazon. A API (*Application Programming Interface*) permite que os desenvolvedores consigam acessar os serviços disponíveis na nuvem da Amazon de forma integrada em suas aplicações. A inclusão dessas bibliotecas nos projetos pode ocorrer de duas formas: através do download diretamente no site da Amazon AWS [5] e importação manual no projeto Java; ou através da configuração de uma dependência no arquivo *pom.xml* da ferramenta de gerenciamento de projetos Apache Maven [8]. No caso da aplicação C-ParGRES Starter, a segunda opção foi escolhida já que o projeto havia sido iniciado usando a plataforma desta ferramenta.

O módulo de *Deployment* do C-ParGRES Starter tem como objetivo tornar mais simples o gerenciamento das atividades necessárias ao correto funcionamento dos serviços do C-ParGRES no ambiente de nuvem da Amazon. Uma vez que a quantidade de máquinas virtuais tenha sido definida pelo *Dimensioner*, esse dado é utilizado como parâmetro de

configuração para o módulo de *Deployment*. Podemos classificar as atividades realizadas pelo módulo de *Deployment* em 4 tipos:

1. Verificação e validação de permissões de acesso;
2. Configuração de grupos de segurança;
3. Criação, inicialização e monitoramento das máquinas virtuais; e
4. Desligamento e encerramento das máquinas virtuais.

A seguir apresentamos como o C-ParGRES implementa cada uma dessas atividades.

#### 4.1.2.1 Atividade #1: Verificação e validação de Permissões de Acesso

Esta atividade compreende a validação das credenciais de segurança do usuário cadastrado no serviço de nuvem da Amazon. Devem ser fornecidos dois códigos de segurança ao C-ParGRES: o *access\_key\_id* e o *secret\_access\_key*. Esses códigos são criados na página de gerenciamento de perfis de acesso do usuário do *site* da Amazon AWS e compreendem um conjunto de políticas de permissões configuráveis de acordo com a necessidade de cada usuário.

Após estabelecer a conexão com a nuvem da Amazon, é instanciado o objeto *AmazonEC2Client*. Este objeto provê os acessos de cliente aos recursos da nuvem Amazon EC2, sendo utilizado pelos demais módulos da aplicação como base para que suas ações sejam propagadas no ambiente de nuvem.

#### 4.1.2.2 Atividade #2: Configuração de Grupos de Segurança

Esta atividade consiste em etapas para configurar e criar o grupo de segurança, na nuvem da Amazon, com as permissões necessárias para que as máquinas virtuais e o *cluster* virtual sejam corretamente acessados pelos serviços do C-ParGRES. O grupo de segurança consiste em um conjunto de permissões de acesso a conexões de rede aplicado a uma ou mais máquinas virtuais, atuando como um *firewall* virtual. É possível aplicar regras a um grupo de segurança para permitir o tráfego de rede a partir de uma determinada máquina virtual para uma faixa de endereçamento específica, assim como, para permitir que conexões remotas sejam aceitas por máquinas virtuais associadas. Se mais de um grupo de segurança for aplicado a uma instância de máquina virtual, suas regras serão

combinadas e todas serão verificadas para decidir se o tráfego de rede será permitido ou não para cada instância [6].

#### 4.1.2.3 Atividade #3: Criação, inicialização e monitoramento das Máquinas Virtuais

As atividades #1 e #2 consistem em um conjunto de etapas destinadas a preparação do ambiente da nuvem da Amazon para executar corretamente o ParGRES em um *cluster* virtual. A atividade #3 é responsável pelas etapas relativas à criação das máquinas virtuais que irão compor o *cluster* virtual e iniciar o serviço do ParGRES nessas máquinas. É importante ressaltar que uma *Amazon Machine Image* - AMI, previamente configurada com os executáveis do ParGRES e com o banco de dados replicado, deve estar disponível para ser utilizada como fonte para a criação das máquinas virtuais pelo C-ParGRES Starter. Realizar a transferência, carga e transformações na base de dados no momento de criação do *cluster* na nuvem acarretaria um alto custo de tempo e tráfego de rede (que no caso de serviços em nuvem, traduz-se diretamente em custos financeiro).

As configurações acerca dos tipos e quantidade de instâncias são definidas no arquivo *PargresConfig.xml*. Essas informações são utilizadas para criar duas classes de máquinas virtuais na nuvem da Amazon: NQP-VM e CQP-VM. A primeira classe é destinada às máquinas virtuais que executam apenas o processo do NQP e não se comunicam com interfaces de rede externas ao *cluster*. A segunda classe é atribuída apenas a uma máquina virtual especial que executa os processos do NQP e do CQP e que, além de se comunicar com as NQP-VMs do *cluster*, também aceita conexões de computadores remotos.

Após todas NQP-VMs terem sido criadas e terem seus grupos de segurança associados, a aplicação envia solicitações, através da Amazon Cloud API, para que cada uma delas seja inicializada e garante que todas instâncias estejam em execução antes de prosseguir. Durante esse processo são obtidos os IDs e endereços IPs de todas as NQP-VMs. Esses dados são usados posteriormente, por tarefas de configuração do *cluster* no serviço do ParGRES. Assim que o Amazon Cloud API informa que todas as NQP-VMs foram instanciadas e devidamente iniciadas, é feita a inicialização da CQP-VM. A CQP-VM é responsável pela comunicação do *cluster* com as aplicações clientes. Após a CQP-VM ser iniciada e ter seu grupo de segurança definido de forma adequada, os IPs das NQP-VMs são adicionados à lista de nós do *cluster* no serviço do ParGRES. Em um ambiente estático e previamente conhecido (como um *cluster* físico local), esse processo poderia ser feito através de uma configuração no arquivo *Pargres.xml*. Porém, como estamos trabalhando



em um ambiente dinâmico, a alocação dos IPs na nuvem da Amazon ocorre somente no momento em que cada máquina virtual é instanciada.

Após todas as máquinas virtuais do *cluster* terem sido iniciadas, a aplicação cliente pode iniciar a submissão de consultas à CQP-VM. Nos experimentos realizados todas as consultas foram enviadas através de uma aplicação console executada em uma máquina virtual da Amazon criada especificamente para esta finalidade. Entretanto esta não é uma exigência, pois qualquer terminal remoto pode realizar a submissão das consultas, desde que o endereçamento de sua rede esteja configurado com permissão de acesso no grupo de segurança atribuído à CQP-VM.

#### 4.1.2.4 Atividade #4: Desligamento e encerramento das Máquinas Virtuais

Após o termino das execuções das consultas e à critério do usuário, a aplicação C-ParGRES *Starter* pode iniciar o processo de remoção das NQP-VMs e CQP-VM do *cluster* virtual. O primeiro passo é desligar as máquinas virtuais e sinalizá-las com status *Terminated*, para que a nuvem da Amazon possa dar início ao processo de deleção das máquinas virtuais. Esse processo ocorre de forma assíncrona e o C-ParGRES *Starter* aguarda até receber a confirmação de que todas as máquinas virtuais foram de fato excluídas. É importante ressaltar que as AMIs que deram origem a cada uma das NQP-VMs e a CQP-VM não são excluídas nesse processo, para que possam ser reutilizadas quando o *cluster* virtual for recriado. Os grupos de segurança associados às máquinas virtuais também são removidos ao final deste processo.

### 4.1.3 Scripts para gerenciamento do C-ParGRES

Os scripts listados a seguir foram criados na linguagem *shell script* e têm o intuito de facilitar a inicialização dos serviços do ParGRES, simplificando o desenvolvimento da aplicação C-ParGRES *Starter*. Cada um dos scripts encapsula comandos comuns entre si, como por exemplo os caminhos dos executáveis e das bibliotecas do ParGRES; o local da instalação da biblioteca Java; a quantidade máxima de memória que a máquina virtual Java pode utilizar; e as portas padronizadas para comunicação dos serviços do C-ParGRES. A seguir descrevemos e apresentamos o código de cada um dos scripts para gerenciamento do C-ParGRES.

### 4.1.3.1 nqp.sh

O *script* `nqp.sh` é responsável pela inicialização do serviço do Node Query Processor - NQP do ParGRES. Esse *script* deve ser executado em cada uma das NQP-VMs que compõem o *cluster* virtual. Além das configurações comuns, conforme descrito anteriormente, este *script* define o nome do banco de dados utilizado no experimento; o caminho e nomes do arquivo de *log* gerado pelas execuções das consultas; e recupera o ID atribuído à máquina virtual corrente na nuvem, através de consulta a um serviço específico da nuvem da Amazon.

```
basepath=/home/ubuntu/pargres/git/ParGRES
outfile=/home/ubuntu/saidas_pargres/nqp.out
libpath=$basepath/lib
binpath=$basepath/bin
configpath=$basepath/config

port=8001
nodeport=3001
maxmemory=2048m
database=censo2010a

die() { status=$1; shift; echo "FATAL: $*"; exit $status; }
EC2_INSTANCE_ID=$(wget -q -O - http://169.254.169.254/latest/meta-data/instance-id || die \
    ↳ id has failed: $? \ "")

echo "Starting NQP:" > $outfile
echo "instance_id=\"$EC2_INSTANCE_ID | tee -a $outfile

cd $binpath

java -Xdebug -Xrunjdwp:transport=dt_socket,address=$port,server=y,suspend=n -Xmx$maxmemory -cp $basepath:
    ↳ $binpath:$libpath/hsqldb.jar:$libpath/log4j-1.2.9.jar:$libpath/pargres-server.jar:$libpath/postgresql-42.0.0.
    ↳ jre6.jar org.pargres.nodequeryprocessor.NodeQueryProcessorEngine $nodeport org.postgresql.Driver jdbc:
    ↳ postgresql://localhost:5432/$database postgres postgres 0

echo "Finishing NQP:" >> $outfile
```

### 4.1.4 cqp.sh

O *script* `cqp.sh` é responsável pela inicialização do serviço do *Cluster Query Processor* - CQP. Esse *script* deve ser executado somente na máquina virtual designada como CQP-VM. Além das configurações comuns, conforme descrito no início da seção, este *script* define o caminho dos arquivos de configuração do ParGRES; e o caminho e nomes do arquivo de *log* gerado pelas execuções das consultas.

```
basepath=/home/ubuntu/pargres/git/ParGRES
outfile=/home/ubuntu/saidas_pargres/cqp.out
```

```

libpath=$basepath/lib
binpath=$basepath/bin
configpath=$basepath/config

port=8000
maxmemory=3072m

echo "Starting CQP:" > $outfile

cd $binpath

java -Xdebug -Xrunjdwp:transport=dt_socket,address=$port,server=y,suspend=n -Xmx$maxmemory -Dlog4j.debug
    ↪ -cp $basepath:$binpath:$libpath/hsqldb.jar:$libpath/log4j-1.2.9.jar:$libpath/pargres-server.jar:$libpath/
    ↪ postgresql-42.0.0.jre6.jar org.pargres.cqp.connection.ConnectionManagerImpl 8050 $configpath/PargresConfig.
    ↪ xml

echo "Finishing CQP:" >> $outfile

```

### 4.1.5 add\_nodes.sh

O *script* `add_nodes.sh` realiza a adição de um ou mais NQP-VMs ao *cluster* virtual do C-ParGRES. Esse *script* deve ser executado somente na CQP-VM e receber como parâmetro a lista de endereços IPs das NQP-VMs, conforme descrito na atividade #3.

```

basepath=/home/ubuntu/pargres/git/ParGRES
libpath=$basepath/lib
binpath=$basepath/bin
configpath=$basepath/config

port=8050
maxmemory=256m

cd $binpath

if [ "$#" -lt 1 ]
then
    echo "Please insert at least one argument"
    exit
fi

for host in "$@"
do
    java -Xmx$maxmemory -cp $basepath:$binpath:$libpath/log4j-1.2.9.jar:$libpath/pargres-jdbc-client.jar:$libpath/
        ↪ junit.jar:$libpath/commons-cli-1.0.jar org.pargres.console.Console -addnode $host:3001
done

```

### 4.1.6 `list_nodes.sh`

O *script* `list_nodes.sh` imprime no console todas as NQP-VMs que compõe o *cluster* virtual do C-ParGRES no momento.

```
basepath=/home/ubuntu/pargres/git/ParGRES
libpath=$basepath/lib
binpath=$basepath/bin
configpath=$basepath/config

port=8050
maxmemory=2048m

cd $binpath

java -Xmx$maxmemory -cp $basepath:$binpath:$libpath/log4j-1.2.9.jar:$libpath/pargres-jdbc-client.jar:$libpath/
↪ junit.jar:$libpath/commons-cli-1.0.jar org.pargres.console.Console -listnodes
```

# Capítulo 5

## Avaliação Experimental

Neste capítulo descrevemos os experimentos realizados com o *middleware* C-ParGRES em um ambiente de nuvem e gerenciando as consultas sobre a base de dados do CD2010A. Nas seções 5.1, 5.2 e 5.3 apresentamos o ambiente computacional de execução dos experimentos, as especificações da base de dados e as consultas utilizadas. Na seção 5.4 descrevemos como os experimentos foram realizados e os resultados obtidos.

### 5.1 Ambiente de execução

O provedor de serviços na nuvem escolhido para a execução do experimento foi o Amazon Web Services (AWS) [7]. Nele utilizamos o serviço Amazon Elastic Compute Cloud (Amazon EC2) [4] que disponibiliza capacidade computacional totalmente configurável, permitindo a execução dos serviços necessários ao experimento. Durante a criação das máquinas virtuais no ambiente de nuvem, não houve a preocupação de determinar se todas elas seriam alocadas em uma mesma máquina física. O serviço Amazon EC2 fornece essa possibilidade através da opção Instâncias Dedicadas, na qual o cliente define que todas as suas instâncias virtuais serão alocadas em uma mesma máquina física. O custo dessa modalidade é maior e, portanto, optamos por não a utilizar para a realização dos experimentos.

O ambiente configurado em cada máquina virtual está detalhado na Tabela 5.1 e contou com o Open JDK 8 [53] e com o PostgreSQL versão 9.5 [61] instalado com sua configuração padrão, mas dimensionado para utilizar até 8GB da memória disponível da máquina. Além disso a versão mais recente do ParGRES também foi utilizada em todas as máquinas virtuais.

Os experimentos com processamento sequencial de consultas sem concorrência foram executados com 1, 2, 4, 8 e 16 instâncias. Os processamentos com concorrência foram executados com 2, 4, 8 e 16 instâncias. Optamos pela não execução com apenas uma instância nesse caso, pois testes iniciais demonstraram a inviabilidade deste cenário em termos de tempo de execução das consultas. Cada consulta foi executada pelo menos 5 vezes para a obtenção da média do tempo de execução. Todos os experimentos encontram-se detalhados na seção 5.4.

Durante os experimentos, a instância de máquina virtual número 1 sempre executou os processos do CQP, NQP e o(s) console(s) de submissão das consultas ao ParGRES. Por esse motivo, esta instância foi criada com o tipo t2.2xlarge, enquanto as demais que executaram somente o NQP, foram criadas com o tipo t2.xlarge. A Tabela 5.1 mostra a configuração de cada um destes tipo de instâncias.

Tabela 5.1: Características detalhadas das instâncias de máquinas virtuais.

Tipo da instância	t2.2xlarge	t2.xlarge
Sistema operacional	Linux Ubuntu Server 16.04 LTS (HVM)	Linux Ubuntu Server 16.04 LTS (HVM)
Família	Propósito geral	Propósito geral
vCPUs	8	4
Processador físico	Intel Broadwell E5-2686v4	Intel Broadwell E5-2686v4
Velocidade de clock	2.3 GHz	2.3 GHz
Arquitetura	64-bit	64-bit
Memória RAM	32 GB	16 GB
Armazenamento	120 GB	120 GB
Tipo do armazenamento	SSD de propósito geral	SSD de propósito geral
Performance de rede	Moderada	Moderada

## 5.2 Base de dados

A base de dados utilizada nos experimentos desta dissertação foi proveniente dos microdados públicos do Censo Demográfico 2010 - Amostra (CD2010A), disponível no servidor de FTP do IBGE [35], e foi completamente replicada em todas as máquinas virtuais presentes no *cluster* virtual. Optamos pelo uso dos dados públicos do censo, pois além de não necessitar de autorizações especiais, facilita a replicação dos experimentos

por outros pesquisadores. A base é menor em termos de número de registros quando comparada com a base de dados do Universo, porém ela possui mais variáveis à disposição do usuário, reunindo informações mais completas acerca dos domicílios e das pessoas. Essas informações mais detalhadas proporcionam a realização de consultas mais complexas, com elevado número de cruzamentos de dados, cálculos, agregações e, conseqüentemente, custo de processamento.

No *site* do IBGE, é possível fazer o *download* dos arquivos em formato *zip* para cada unidade da federação, sendo que São Paulo está dividido em dois arquivos por conta da grande quantidade de registros. Dentro de cada arquivo *zip* é possível encontrar 4 arquivos de texto posicionais, um para cada tema pesquisado no censo: Domicílios, Pessoas, Mortalidade e Emigração. Para fins de experimento utilizamos apenas os microdados de Domicílios e de Pessoas, visto que são as tabelas com maior quantidade de registros, com respectivamente 6.192.332 e 20.635.472 registros cada. Por questões de limitação de espaço físico devido ao alto custo financeiro por GB armazenado na nuvem, somente foram carregadas as dimensões necessárias ao conjunto de consultas definido no experimento. Seguindo essa lógica, só foram criados índices sobre as chaves primárias, atributo de particionamento virtual e colunas utilizadas nas consultas propostas.

Ainda no *site* do IBGE, e juntamente com os arquivos de microdados, encontra-se a Planilha de *Layout* dos Microdados da Amostra [34], com os códigos e nomes das variáveis disponíveis em cada tema; posição inicial e final de cada uma delas no arquivo; e tamanho da parte inteira e decimal, quando aplicável.

A seguir descrevemos os passos executados para a importação dos dados para o banco PostgreSQL:

1. A primeira etapa foi a extração de todos os arquivos *zip* em um mesmo diretório;
2. Em seguida, foi realizada a importação de todo o conteúdo dos arquivos para o banco de dados, sendo uma tabela para o tema de Domicílios (CD2010ADOMI) e uma tabela para o tema de Pessoas (CD2010APESS). Ao término dessa etapa, cada tabela continha apenas um atributo com *n* tuplas, sendo uma tupla para cada linha do arquivo de entrada;
3. A partir das tabelas criadas no passo anterior, foi aplicado um *script* para extrair o valor de cada variável de acordo com sua posição e tamanho. Esse *script* foi elaborado com base na Planilha de *Layout* dos Microdados da Amostra.

4. O quarto passo foi a transformação das tabelas de fato para o formato específico que o BME trabalha. Todas as colunas que representam variáveis categorizadas receberam o prefixo *COD* nos seus nomes e algumas variáveis derivadas foram criadas. Esses detalhes são importantes, pois as consultas utilizadas no experimento foram extraídas de aplicações em produção no IBGE e que utilizam essa base.
5. Em seguida, executamos um *script* para criação de algumas variáveis categorizadas específicas para a execução das consultas do sistema BME-TABULADOR. Originalmente, no sistema em produção no IBGE, essas variáveis não são necessárias, pois elas são calculadas em tempo de execução através de sub-consultas. Porém, como o apoio a sub-consultas ainda não está totalmente disponível no ParGRES, esta etapa teve que ser acrescentada;
6. A sexta etapa consistiu na carga das tabelas de dimensões. Essas são as tabelas presentes nas junções com as tabelas de fatos e proporcionam os descritores das categorias de cada variável;
7. Já com todas as tabelas criadas e carregadas, realizamos a criação de índices, chaves primárias e estrangeiras;
8. Por fim, foi realizada a atualização das estatísticas dos índices e desfragmentação da base de dados.

Observando a base dos microdados pública da Amostra do Censo Demográfico 2010, é possível perceber que o menor nível geográfico de identificação dos microdados é a área de ponderação. Não estão disponíveis variáveis geográficas dos níveis territoriais de distrito, subdistrito ou setor censitário. Restringir o nível geográfico de identificação dos microdados é também uma forma de tratamento para garantir o sigilo de informações individuais, conforme normatização abordada previamente [2].

O atributo chave da tabela (variável de controle V0300), codificado pelo IBGE com valores sequenciais, foi utilizado na configuração do ParGRES como o atributo de fragmentação virtual (AFV) pois apresenta distribuição de valores sequencial e uniforme, sem distorção nos dados [10]. Ele compõe as chaves primárias de ambas as tabelas de fatos.

A chave primária da tabela de Domicílios foi definida como CODMUNICIPIO + V0300, enquanto a chave primária de Pessoas como CODMUNICIPIO + V0300 + V0504. O atributo V0504, representa a ordem lógica das pessoas no domicílio, apresentando valores sequenciais para cada domicílio. Na base de dados utilizada neste experimento o



atributo V0300 apresenta valores únicos e poderia ter sido definido como a chave primária da tabela de Domicílios, pois foi recodificado para valores únicos e sequenciais pelo IBGE no momento da liberação dos microdados. Já na base em produção do BME, o atributo V0300 não é único e compõe a chave primária de Domicílios junto com o atributo CODSETOR (CODSETOR + V0300). A situação é análoga para a tabela de Pessoas, onde o atributo V0300 compõe a chave primária composta, juntamente com o CODSETOR e V0504 (CODSETOR + V0300 + V0504). Neste experimento optamos por definir as chaves primárias das tabelas de fato utilizando o atributo da dimensão geográfica de municípios (CODMUNICIPIO) junto com o atributo AFV V0300 para mantermos maior correspondência com a realidade encontrada na base do BME e facilitar a tradução das consultas daquela base a utilizada no experimento. Cabe ressaltar que o atributo CODSETOR não é divulgado pelo IBGE no arquivo de microdados.

A junção entre as tabelas de fatos de Domicílios e Pessoas se dá pela chave primária de Domicílios, pois de acordo com a organização da pesquisa censitária, todas as pessoas entrevistadas sempre devem estar associadas a um, e somente um, domicílio cadastrado.

## 5.3 Consultas selecionadas para o experimento

Nesta seção descrevemos as consultas utilizadas no experimento. Foram selecionados dois conjuntos de consultas, um proveniente da aplicação BME e outro da aplicação BME-TABULADOR. Apesar de ambos os conjuntos serem executados sobre a mesma base, as consultas possuem características distintas quanto à complexidade, conforme será apresentado a seguir.

### 5.3.1 Consultas da aplicação BME

Para a realização dos experimentos com o framework ParGRES, selecionamos 14 consultas típicas da aplicação BME sobre a base do Censo Demográfico 2010 - Dados da Amostra. As consultas estendem as apresentadas por Paes [10] sobre a base de dados do Censo Demográfico 2000 - Dados da Amostra.

A seguir apresentamos a descrição de cada consulta, com seus cálculos, junções entre as tabelas de fatos e dimensões e a existência ou não cláusulas restritivas. O SQL das consultas está disponível no Apêndice B.

- **Q01: Número de domicílios segundo a situação em relação à sua locali-**

**zação (urbano ou rural) no Brasil.**

Essa consulta acessa a tabela de fatos CD2010ADOMI para calcular a frequência expandida (levando em conta o peso que cada registro tem na amostra) e a frequência simples (apenas a contagem de registros) de domicílios, segundo a situação em relação à sua localização (urbano ou rural). É feita uma junção com as dimensões G000, T030 e CD275000, para transformar os códigos das categorias nos seus respectivos descritores. Essa consulta não possui nenhum predicado, logo todas as tuplas da tabela de fatos são acessadas durante sua execução.

- **Q02: Número de pessoas por classes de idade em anos no Brasil.**

Essa consulta acessa a tabela de fatos CD2010APESS para calcular a frequência, expandida e a frequência simples de pessoas, agrupadas pelas classes de idade da população. É feita uma junção com as dimensões G000, T030 e CD549400, para transformar os códigos das categorias nos seus respectivos descritores. Essa consulta não possui nenhum predicado, logo todas as tuplas da tabela de fatos são acessadas durante sua execução.

- **Q03: Média da idade calculada em anos das pessoas no Brasil.**

Essa consulta acessa a tabela de fatos CD2010APESS para calcular a média da idade em anos, frequência expandida e a frequência simples das pessoas. É feita uma junção com as dimensões G000 e T030, para transformar os códigos das categorias nos seus respectivos descritores. Essa consulta não possui nenhum predicado, logo todas as tuplas da tabela de fatos são acessadas durante sua execução.

- **Q04: Somatório do total de filhos tidos pelas pessoas, por classes de número de filhos no Brasil.**

Essa consulta acessa a tabela de fatos CD2010APESS para calcular a soma do número de filhos tidos, frequência expandida e a frequência simples das pessoas, agrupadas por classes de número de filhos. É feita uma junção com as dimensões G000, T030 e CD290200, para transformar os códigos das categorias nos seus respectivos descritores. Essa consulta não possui nenhum predicado, logo todas as tuplas da tabela de fatos são acessadas durante sua execução.

- **Q05: Número de pessoas por determinadas religiões ou cultos no Brasil. (códigos das religiões: 0, 110, 240, 750, 810).**

Essa consulta acessa a tabela de fatos CD2010APESS para calcular a frequência expandida e a frequência simples das pessoas, agrupadas por tipo de religião ou culto.

É feita uma junção com as dimensões G000, T030, CD281210, para transformar os códigos das categorias nos seus respectivos descritores. Essa consulta possui um predicado seletivo, recuperando 76,62% das tuplas da tabela de fatos durante sua execução.

- **Q06: Média da idade em anos das pessoas com idade entre 20 e 40 anos no Brasil.**

Essa consulta acessa a tabela de fatos CD2010APESS para calcular a média da idade, a frequência expandida e a frequência simples das pessoas com idade entre 20 e 40 anos. É feita uma junção com as dimensões G000 e T030 para transformar os códigos das categorias nos seus respectivos descritores. Essa consulta possui um predicado seletivo, recuperando 33,92% das tuplas da tabela de fatos durante sua execução.

- **Q07: Média da idade em anos das pessoas, por sexo e curso mais elevado frequentado, no município do Rio de Janeiro.**

Essa consulta acessa a tabela de fatos CD2010APESS para calcular a média da idade, a frequência expandida e a frequência simples das pessoas, agrupadas por sexo e curso mais elevado frequentado, no município do Rio de Janeiro. É feita uma junção com as dimensões G235, T030, CD279500 e CD284000, para transformar os códigos das categorias nos seus respectivos descritores. Essa consulta possui um predicado bem seletivo, recuperando 1,51% das tuplas da tabela de fatos durante sua execução.

- **Q08: Média da idade em anos das pessoas, por sexo e curso mais elevado frequentado, nos municípios do Rio de Janeiro e Natal.**

Essa consulta acessa a tabela de fatos CD2010APESS para calcular a média da idade, a frequência expandida e a frequência simples das pessoas, agrupadas por sexo e curso mais elevado frequentado, nos municípios do Rio de Janeiro e Natal. É feita uma junção com as dimensões G235, T030, CD279500 e CD284000, para transformar os códigos das categorias nos seus respectivos descritores. Essa consulta possui dois predicados bem seletivos, recuperando 1,70% das tuplas da tabela de fatos durante sua execução.

- **Q09: Número de pessoas por sexo, em zonas rurais e urbanas, no Brasil.**

Essa consulta realiza a junção das tabelas de fatos CD2010APESS e CD2010ADOMI para calcular a frequência, expandida e a frequência simples das pessoas agrupadas

por sexo e situação do domicílio (zona urbana ou rural). É feita uma junção entre as tabelas de fatos e as dimensões G000, T030, CD275000 e CD279500, para transformar os códigos das categorias nos seus respectivos descritores. Essa consulta não possui nenhum predicado, logo todas as tuplas das tabelas de fatos são acessadas durante sua execução.

- **Q10: Média de idade em anos das pessoas, por sexo e curso mais elevado frequentado, nas zonas rurais e urbanas do município do Rio de Janeiro.**

Essa consulta realiza a junção das tabelas de fatos CD2010APESS e CD2010ADOMI para calcular a média da idade, a frequência expandida e a frequência simples das pessoas, agrupadas por sexo, curso mais elevado frequentado e situação do domicílio (zona urbana ou rural), no município do Rio de Janeiro. É feita uma junção com as dimensões G235, T030, CD279500, CD284000 e CD275000, para transformar os códigos das categorias nos seus respectivos descritores. Essa consulta possui um predicado bem seletivo, recuperando 1,51% das tuplas das tabelas de fatos durante sua execução.

- **Q11: Média de idade das pessoas, por sexo e curso mais elevado frequentado, nas zonas rurais e urbanas dos municípios do Rio de Janeiro e Natal.**

Essa consulta realiza a junção das tabelas de fatos CD2010APESS e CD2010ADOMI para calcular a média da idade, a frequência expandida e a frequência simples das pessoas, agrupadas por sexo, curso mais elevado frequentado e situação do domicílio (zona urbana ou rural), nos municípios do Rio de Janeiro e Natal. É feita uma junção com as dimensões G235, T030, CD279500, CD284000 e CD275000, para transformar os códigos das categorias nos seus respectivos descritores. Essa consulta possui dois predicados bem seletivos, recuperando 1,70% das tuplas das tabelas de fatos durante sua execução.

- **Q12: Bolsa-Família - Característica de pobreza dos domicílios particulares permanentes com crianças em fase escolar em cada unidade da federação.**

Essa consulta realiza a junção das tabelas de fatos CD2010APESS e CD2010ADOMI para calcular a média de idade, média do rendimento domiciliar per capita, frequência expandida e a frequência simples de crianças em fase escolar (entre 7 e 14 anos), em domicílios particulares permanentes com rendimento domiciliar mensal

per-capita inferior a R\$140,00, nas unidades da federação. É feita uma junção com as dimensões G032, T030 e CD275200, para transformar os códigos das categorias nos seus respectivos descritores. Essa consulta possui dois predicados bem seletivos, recuperando 4,62% das tuplas das tabelas de fatos durante sua execução.

- **Q13: Número de pessoas alfabetizadas e não alfabetizadas, nas zonas urbanas e rurais em cada unidade da federação.**

Essa consulta realiza a junção das tabelas de fatos CD2010APESS e CD2010ADOMI para calcular a frequência expandida e a frequência simples de pessoas alfabetizadas e não alfabetizadas, agrupado pela situação do domicílio (zona urbana ou rural), nas unidades da federação. É feita uma junção com as dimensões G032, T030, CD275000 e CD283400, para transformar os códigos das categorias nos seus respectivos descritores. Essa consulta possui um predicado seletivo, recuperando 92,57% das tuplas das tabelas de fatos durante sua execução.

- **Q14: Deficientes - Número de pessoas, por tipo de domicílio, que tenham deficiência mental ou de ouvir, em zonas urbanas e rurais, por tipo do domicílio em cada unidade da federação.**

Essa consulta realiza a junção das tabelas de fatos CD2010APESS e CD2010ADOMI para calcular a frequência expandida e a frequência simples de pessoas com deficiência mental ou de ouvir, agrupado pela situação do domicílio (zona urbana ou rural) e tipo do domicílio, nas unidades da federação. É feita uma junção a junção entre as tabelas de fatos com as dimensões G032, T030 e CD275000, CD281800, CD281500 e CD275300, para transformar os códigos das categorias nos seus respectivos descritores. Essa consulta possui dois predicados bem seletivos, recuperando 6,38% das tuplas das tabelas de fatos durante sua execução. Entre os predicados é utilizado o operador **OR** (disjunção).

É importante ressaltar que as consultas propostas por Paes [10] foram baseadas na base de dados do Censo Demográfico 2000 - Dados da Amostra, logo algumas das nossas consultas tiveram que sofrer adaptações em decorrência de ausência e/ou mudanças metodológicas de estruturação de algumas variáveis. Além disso, os microdados do tema de Famílias do Censo Demográfico 2010 não foram disponibilizados publicamente, conforme havia ocorrido em 2000. Elencamos a seguir as diferenças conceituais entre as consultas executadas por Paes [10] e as apresentadas neste trabalho:

- **Consultas Q07, Q08, Q10, Q11:** A variável quantitativa “anos de estudo” não

foi disponibilizada no Censo Demográfico 2010 da mesma maneira que em 2000, tendo sido substituída pela variável categorizada “curso mais elevado frequentado”. Para que essas consultas mantivessem a semelhança com as de Paes [10] em termos de complexidade de cálculos, optamos por realizar o cálculo da média da variável quantitativa “idade em anos” no lugar da média da variável quantitativa “anos de estudo” e incluir a variável categorizada “curso mais elevado frequentado”. Dessa forma a quantidade de cálculos e semântica da consulta seriam mantidas, visto que o resultado ainda traria informações sobre a escolaridade das pessoas.

- **Consulta Q12:** Nessa consulta, Paes [10] utiliza uma junção com a tabela de fatos do tema Famílias para realizar o filtro pela variável quantitativa “rendimento per-capita da família”. Como não dispúnhamos dos microdados do tema Famílias divulgados publicamente para o Censo Demográfico 2010, optamos por utilizar a variável quantitativa “rendimento domiciliar per-capita”, realizando a junção com a tabela de fatos de Domicílios. Pode-se dizer que é uma aproximação razoável, já que segundo a Tabela 229 do SIDRA [27], cerca de 85% dos domicílios brasileiros tem apenas uma família.
- **Consulta Q14:** Nesse caso, Paes [10] optou por realizar a consulta agrupando o resultado pela variável categorizada “tipo da família”. Devido às mesmas restrições apontadas para consulta Q12, optamos por realizar a consulta utilizando a variável categorizada “tipo do domicílio”, para conseguirmos um efeito similar em termos de cruzamento de dados.

A Tabela 5.2 resume as principais características das consultas selecionadas da aplicação BME. A primeira coluna descreve a identificação da consulta, a segunda coluna o número total de agregações realizadas pela consulta, a terceira coluna define se a consulta apresenta algum predicado, a quarta coluna mostra o número de tabelas de dimensões envolvidas na consulta, a quinta coluna descreve o nome das tabelas de fatos acessadas pela consulta, a sexta coluna mostra a quantidade total de tuplas recuperadas e, por fim, a sétima coluna descreve a porcentagem de tuplas recuperadas em comparação com o total de registros contidos nas tabelas de fatos.

### 5.3.2 As consultas da aplicação BME-TABULADOR

Escolhemos 3 consultas utilizadas no processo de tabulação dos dados do Censo Demográfico 2010 - Dados da Amostra para o sistema SIDRA.

Tabela 5.2: Principais características das consultas do BME.

Consulta	Agregações	Predicados	Tabelas de Dimensões	Tabelas de Fatos	Tuplas recuperadas	% Tuplas recuperadas
Q01	2	N	3	CD2010ADOMI	6.192.332	100%
Q02	2	N	3	CD2010APESS	20.635.472	100%
Q03	4	N	2	CD2010APESS	20.635.472	100%
Q04	3	N	3	CD2010APESS	20.635.472	100%
Q05	2	S	3	CD2010APESS	15.812.791	76,62%
Q06	4	S	2	CD2010APESS	7.000.396	33,92%
Q07	4	S	4	CD2010APESS	310.660	1,51%
Q08	4	S	4	CD2010APESS	349.888	1,70%
Q09	2	N	4	CD2010ADOMI, CD2010APESS	20.635.472	100%
Q10	4	S	5	CD2010ADOMI, CD2010APESS	310.660	1,51%
Q11	4	S	5	CD2010ADOMI, CD2010APESS	349.888	1,70%
Q12	6	S	3	CD2010ADOMI, CD2010APESS	954.380	4,62%
Q13	2	S	4	CD2010ADOMI, CD2010APESS	19.101.734	92,57%
Q14	2	S	6	CD2010ADOMI, CD2010APESS	1.316.982	6,38%

Um ponto que devemos destacar é que o apoio à cláusula *GROUPING SETS*, utilizadas por todas as consultas do BME-TABULADOR, não está implementada no PARGRES. Foi necessária a reescrita das consultas de forma que cada *cluster* da cláusula *GROUPING SETS* fosse individualizado em uma consulta separada (através de uma cláusula *GROUP BY*). Essa alteração aumentou substancialmente a quantidade real de consultas que foram executadas nesta parte do experimento. Para obtenção do resultado equivalente à consulta original, bastaria que fossem concatenados todos os resultados obtidos por cada consulta reescrita.

O sistema SIDRA, destino dos dados gerados, trabalha tanto com um único arquivo contendo os dados necessários ao carregamento das tabelas agregadas, quanto com múltiplos arquivos complementares. A concatenação em um único arquivo nada mais é que

um facilitador para o trabalho diário no processo de tabulação. Se for desejado, é possível fazer a concatenação dos arquivos, inclusive, através de comandos específicos do sistema operacional, fato que costuma ocorrer mesmo no ambiente de produção quando determinadas consultas têm que ser executadas de forma independente. Como neste trabalho nosso interesse é avaliar o tempo total de execução das consultas, dadas as diversas configurações de ambientes distribuídos, a concatenação dos arquivos de resultados não nos traria nenhuma informação relevante e, portanto, não foi realizada.

A seguir, apresentamos a descrição de cada consulta, com seus cálculos, junções entre as tabelas de fatos e a existência ou não cláusulas restritivas. Vale ressaltar que, diferentemente do que ocorre com as consultas do BME, apenas os códigos das dimensões são recuperados nas consultas, não sendo realizadas junções das tabelas de fatos com as dimensões envolvidas. O objetivo do BME-TABULADOR é unicamente alimentar o SIDRA, portanto os descritores das dimensões não são necessários neste processo.

- **Consulta T3459 - Pessoas de 10 anos ou mais de idade, ocupadas na semana de referência, por religião, segundo o sexo, a posição na ocupação e a categoria do emprego no trabalho principal e as classes de rendimento nominal mensal de todos os trabalhos - Características Gerais da População [28].**

Essa consulta acessa a tabela de fatos CD2010APESS para calcular a frequência expandida (levando em conta o peso que cada registro tem na amostra) de pessoas, agrupado pela religião, sexo, posição na ocupação, categoria do emprego no trabalho principal e classes de rendimento nominal mensal de todos os trabalhos, no Brasil. Essa consulta possui um predicado seletivo, recuperando 39,92% das tuplas da tabela de fatos durante sua execução. O resultado pode ser observado na Figura 5.1.

- **Consulta T3525 - Domicílios particulares permanentes com rendimento domiciliar, moradores em domicílios particulares permanentes com rendimento domiciliar e valor do rendimento nominal médio e mediano mensal dos domicílios particulares permanentes com rendimento domiciliar, por situação do domicílio, tipo de material das paredes externas, número de cômodos, número de dormitórios e existência de água canalizada e forma de abastecimento de água [29].**

Essa consulta acessa a tabela de fatos CD2010ADOMI para calcular a frequência expandida de domicílios, o número de moradores e a média do rendimento domiciliar dos



Tabela 3459 - Pessoas de 10 anos ou mais de idade, ocupadas na semana de referência, por religião, segundo o sexo, a posição na ocupação e a categoria do emprego no trabalho principal e as classes de rendimento nominal mensal de todos os trabalhos - Características Gerais da População										
Variável - Pessoas de 10 anos ou mais de idade, ocupadas na semana de referência (Pessoas)										
Brasil										
Posição na ocupação e categoria do emprego no trabalho principal - Total										
Ano - 2010										
Classes de rendimento nominal mensal de todos os trabalhos	Religião x Sexo									
	Católica Apostólica Romana		Evangélicas		Espirita		Outras religiosidades		Não sabe	
	Homens	Mulheres	Homens	Mulheres	Homens	Mulheres	Homens	Mulheres	Homens	Mulheres
Total	33.089.604	23.328.142	9.381.308	8.579.588	950.330	1.173.452	1.243.523	1.107.271	54.757	29.613
Até 1/2 salário mínimo	2.454.808	2.494.116	453.566	890.767	12.784	33.562	51.928	95.099	4.208	2.692
Mais de 1/2 a 1 salário mínimo	7.388.518	6.503.353	1.930.483	2.636.617	83.604	166.309	210.723	289.387	16.021	10.738
Mais de 1 a 2 salários mínimos	10.590.232	7.125.573	3.530.249	2.989.379	221.825	334.431	416.506	368.008	19.518	9.525
Mais de 2 a 3 salários mínimos	3.921.866	1.872.339	1.350.358	667.724	140.153	161.123	179.918	101.479	5.556	2.409
Mais de 3 a 5 salários mínimos	2.965.417	1.587.947	949.151	482.039	160.156	183.787	148.129	87.564	3.349	1.486
Mais de 5 a 10 salários mínimos	2.241.999	1.190.602	578.865	290.475	184.073	180.564	116.942	72.606	3.058	1.013
Mais de 10 a 15 salários mínimos	448.040	215.597	92.839	44.719	49.903	39.771	24.453	13.325	371	256
Mais de 15 a 20 salários mínimos	395.034	146.474	69.316	26.679	42.615	25.889	22.706	10.577	273	82
Mais de 20 a 30 salários mínimos	227.060	75.205	35.410	11.130	25.727	12.911	13.531	4.728	234	59
Mais de 30 salários mínimos	164.184	43.100	23.373	7.072	15.301	6.497	10.176	2.871	96	5
Sem rendimento	2.292.446	2.073.836	367.699	532.988	14.189	28.608	48.512	61.628	2.074	1.347
Fonte: IBGE - Censo Demográfico										

Figura 5.1: Tabela SIDRA 3459.

domicílios particulares permanentes com rendimento, agrupado pela situação do domicílio, tipo de material das paredes externas, número de cômodos, número de dormitórios e existência de água canalizada e forma de abastecimento de água, no Brasil. Essa consulta possui dois predicados, recuperando 94,58% das tuplas da tabela de fatos durante sua execução. O resultado pode ser observado na Figura 5.2.

Tabela 3525 - Domicílios particulares permanentes com rendimento domiciliar, moradores em domicílios particulares permanentes com rendimento domiciliar e valor do rendimento nominal médio e mediano mensal dos domicílios particulares permanentes com rendimento domiciliar, por situação do domicílio, tipo de material das paredes externas, número de cômodos, número de dormitórios e existência de água canalizada e forma de abastecimento de água												
Brasil												
Número de dormitórios - Total												
Ano - 2010												
Número de cômodos - Total												
Tipo de material das paredes externas	Variável x Existência de água canalizada e forma de abastecimento x Situação do domicílio											
	Domicílios particulares permanentes com rendimento domiciliar (Unidades)				Moradores em domicílios particulares permanentes com rendimento domiciliar (Pessoas)				Valor do rendimento nominal médio mensal dos domicílios particulares permanentes com rendimento domiciliar (Reais)			
	Tinham - em pelo menos um cômodo		Não tinham água canalizada		Tinham - em pelo menos um cômodo		Não tinham água canalizada		Tinham - em pelo menos um cômodo		Não tinham água canalizada	
	Urbana	Rural	Urbana	Rural	Urbana	Rural	Urbana	Rural	Urbana	Rural	Urbana	Rural
Total	44.995.796	4.909.904	949.508	2.021.425	146.700.385	17.151.910	3.517.508	8.254.718	2.945,69	1.450,32	1.144,96	704,95
Alvenaria com revestimento	38.215.911	3.323.117	474.108	930.940	122.731.664	11.536.625	1.678.831	3.595.944	3.188,17	1.527,60	1.418,51	750,73
Alvenaria sem revestimento	4.216.174	591.602	229.966	355.845	15.135.134	2.160.251	859.798	1.426.271	1.522,23	1.077,41	903,83	629,89
Madeira aparelhada	2.121.170	777.059	134.676	272.869	7.195.430	2.618.349	550.310	1.258.636	1.736,37	1.564,34	996,60	831,69
Taipa revestida	71.210	78.292	21.175	141.238	270.402	302.225	83.261	586.900	964,70	814,85	606,62	623,09
Taipa não revestida	51.898	47.165	36.229	183.450	208.476	196.139	143.137	775.661	892,20	668,71	522,02	529,61
Madeira aproveitada	225.709	50.938	44.661	62.796	837.210	182.265	169.214	280.969	1.174,06	1.133,58	769,55	704,93
Palha	1.877	2.837	1.353	27.729	6.856	12.514	5.459	132.774	1.815,37	774,69	433,84	582,01
Outro material	91.847	38.866	7.341	45.646	315.212	143.407	27.498	192.539	2.402,39	933,50	795,18	630,15
Sem parede	-	28	-	909	-	135	-	5.025	-	718,53	-	774,40
Fonte: IBGE - Censo Demográfico												

Figura 5.2: Tabela SIDRA 3525.

- Consulta T3534 - Domicílios particulares permanentes com rendimento domiciliar, moradores em domicílios particulares permanentes com rendimento domiciliar e valor do rendimento nominal médio e mediano mensal dos domicílios particulares permanentes com rendimento domiciliar, por situação do domicílio, número de cômodos e existência de alguns bens duráveis [30].

Essa consulta acessa a tabela de fatos CD2010ADOMI para calcular a frequência expandida de domicílios, o número de moradores e a média do rendimento domiciliar dos domicílios particulares permanentes com rendimento, agrupado pela situação do domicílio, número de cômodos e existência de alguns bens duráveis, no Brasil. Essa consulta possui dois predicados, recuperando 94,58% das tuplas da tabela de fatos durante sua execução. O resultado pode ser observado na Figura 5.3.

Tabela 3534 - Domicílios particulares permanentes com rendimento domiciliar, moradores em domicílios particulares permanentes com rendimento domiciliar e valor do rendimento nominal médio e mediano mensal dos domicílios particulares permanentes com rendimento domiciliar, por situação do domicílio, número de cômodos e existência de alguns bens duráveis												
Brasil												
Ano - 2010												
Variável - Domicílios particulares permanentes com rendimento domiciliar (Unidades)												
Alguns bens duráveis existentes no domicílio	Situação do domicílio	Número de cômodos										
		Total	1 cômodo	2 cômodos	3 cômodos	4 cômodos	5 cômodos	6 cômodos	7 cômodos	8 cômodos	9 cômodos	10 cômodos ou mais
Total	Urbana	47.254.531	204.714	1.314.412	3.796.384	7.030.680	12.400.612	8.715.919	5.497.669	3.496.724	1.858.348	2.939.068
	Rural	7.531.139	87.021	258.703	599.510	1.250.344	1.864.894	1.613.811	905.059	502.158	227.270	222.371
Rádio	Urbana	39.002.661	105.274	850.274	2.742.205	5.479.952	10.204.649	7.378.997	4.755.182	3.092.311	1.672.494	2.721.323
	Rural	5.826.560	40.719	150.094	395.741	887.668	1.447.220	1.315.940	762.051	431.649	197.631	197.847
Televisão	Urbana	45.904.024	156.630	1.165.558	3.559.932	6.753.572	12.088.266	8.542.908	5.413.876	3.457.575	1.842.719	2.922.989
	Rural	6.392.916	33.148	136.544	410.210	982.031	1.635.057	1.463.818	838.368	468.409	213.581	211.748
Máquina de lavar roupa	Urbana	24.932.407	14.927	283.874	1.186.287	2.802.521	6.038.483	4.732.975	3.434.465	2.481.038	1.436.817	2.521.019
	Rural	1.259.356	2.116	12.084	47.054	126.349	279.837	292.890	200.353	132.256	70.367	96.049
Geladeira	Urbana	45.564.392	120.185	1.067.611	3.451.814	6.643.984	12.030.769	8.553.725	5.439.255	3.473.681	1.850.623	2.932.745
	Rural	6.039.224	22.997	107.122	342.956	877.115	1.548.261	1.422.443	827.865	464.981	212.955	212.529
Telefone celular	Urbana	41.256.374	126.167	994.038	3.073.114	5.834.407	10.763.250	7.689.497	4.977.766	3.229.252	1.744.978	2.823.905
	Rural	4.679.393	26.047	98.731	270.618	635.197	1.182.797	1.091.403	652.987	369.269	171.897	180.446
Telefone fixo	Urbana	22.130.585	9.628	196.104	843.510	2.243.294	5.113.762	4.268.462	3.212.065	2.371.350	1.391.326	2.481.082
	Rural	520.358	414	3.150	11.957	35.908	94.781	114.492	93.617	69.060	38.992	57.987
Microcomputador	Urbana	20.678.791	9.106	192.228	771.553	1.917.448	4.684.083	3.969.399	3.077.819	2.279.108	1.349.086	2.428.959
	Rural	651.720	459	3.547	14.660	43.298	125.659	149.828	116.604	82.359	47.140	68.165
Motocicleta para uso particular	Urbana	8.514.068	13.409	131.273	473.324	1.028.291	2.189.305	1.764.818	1.187.210	757.824	391.970	576.644
	Rural	2.284.473	8.748	33.894	106.159	300.492	555.902	558.479	344.913	197.996	91.497	86.393
Automóvel para uso particular	Urbana	20.168.181	7.203	158.954	676.060	1.761.060	4.451.366	3.922.869	3.062.204	2.291.422	1.358.979	2.478.065
	Rural	1.871.982	2.976	14.051	49.328	152.727	400.973	463.273	328.874	213.934	109.913	135.933
Fonte: IBGE - Censo Demográfico												

Figura 5.3: Tabela SIDRA 3534.

A Tabela 5.3 resume as principais características das consultas selecionadas da aplicação BME-TABULADOR. A primeira coluna descreve a identificação da consulta, a segunda coluna o número total de agregações realizadas pela consulta, a terceira coluna define se a consulta apresenta algum predicado, a quarta coluna mostra o número de cláusulas *GROUPING SETS* envolvidas na consulta, a quinta coluna descreve o nome das

tabelas de fatos acessadas pela consulta, a sexta coluna mostra a quantidade total de tuplas recuperadas e, por fim, a sétima coluna descreve a porcentagem de tuplas recuperadas em comparação com o total de registros contidos nas tabelas de fatos.

Tabela 5.3: Principais características das consultas do BME-TABULADOR.

Consulta	Agregações	Predicados	<i>Grouping Sets</i>	Tabelas de Fatos	Tuplas recuperadas	% Tuplas recuperadas
<b>T3459</b>	1	S	4	CD2010APESS	8.238.003	39,92%
<b>T3525</b>	4	S	48	CD2010ADOMI	5.856.627	94,58%
<b>T3534</b>	4	S	44	CD2010ADOMI	5.856.627	94,58%

## 5.4 Experimentos realizados e resultados

Foram realizados quatro experimentos diferentes, cada um deles executado em diversos *clusters* virtuais, com a base de dados totalmente replicada em cada máquina virtual. O objetivo é avaliar o desempenho do processamento de consultas OLAP sobre os dados reais do Censo Demográfico 2010 - Dados da Amostra, utilizando o *middleware* C-ParGRES e o paralelismo intra-consulta. Os experimentos foram:

1. Experimento sem concorrência: Consultas BME e BME-TABULADOR são submetidas sequencialmente;
2. Experimento sem concorrência: Consultas BME são submetidas sequencialmente em grupos;
3. Experimento com concorrência: Consultas BME são submetidas em grupos;
4. Experimento com concorrência em um ambiente com carga mista: Consultas BME e BME-TABULADOR são submetidas em grupos.

Os experimentos com processamento de consultas sem concorrência foram executados com 1, 2, 4, 8 e 16 instâncias de máquinas virtuais. Os processamentos com concorrência foram executados com 2, 4, 8 e 16 instâncias de máquinas virtuais. A execução com apenas 1 instância de máquina virtual, neste caso, apresentou elevado tempo de resposta para executar uma pequena porção das consultas pretendidas, demonstrando a inviabilidade deste cenário em termos de tempo de resposta das consultas. Decidimos não levar adiante

essa parte do experimento, pois trabalhar com apenas uma instância de máquina virtual e com alta demanda de consultas concorrentes não é o objetivo para o qual o C-ParGRES foi concebido. Em ambientes com essa configuração, alternativas devem ser avaliadas ao uso do C-ParGRES.

É importante ressaltar que o módulo *Dimensioner* do C-ParGRES *Starter* não foi utilizado na etapa experimental. Para que a comparabilidade dos tempos de execução e *speedup* das consultas fosse possível, a quantidade de máquinas virtuais utilizadas em cada etapa dos experimentos deveria ser previamente definida. Além disso, um dos requisitos de funcionamento do módulo *Dimensioner* é justamente um conjunto pré-existente de execuções de consultas para servir de entrada para o algoritmo.

Cada consulta selecionada para os experimentos possui características que aumentam ou diminuem sua complexidade e custo de execução, que pode variar em função das junções com as tabelas de fatos, filtros aplicados e cálculos efetuados. A seguir vamos detalhar cada um dos experimentos realizados e os tempos de execução obtidos para as consultas.

#### 5.4.1 Experimento 1: Consultas BME e BME-TABULADOR, sem Concorrência

Neste primeiro experimento, as consultas selecionadas dos sistemas BME e BME-TABULADOR foram executadas de forma sequencial e sem concorrência com 1, 2, 4, 8 e 16 instâncias de máquinas virtuais. Realizamos 5 execuções de cada consulta para obtermos a média do tempo de execução (em segundos). Todas as consultas foram processadas com o paralelismo intra-consulta.

As Tabelas 5.4 e 5.5 apresentam os tempos de execução médios e o desvio padrão do tempo médio de execução para cada consulta e quantidades de máquinas virtuais que compõem o *cluster* virtual. As Tabelas 5.4 e 5.5 fornecem evidências dos bons resultados obtidos, que também podem ser observados nas Figuras 5.4, 5.5, 5.6, 5.7, 5.8 e 5.9.

As Figuras 5.4, 5.5 e 5.6 apresentam o tempo de execução normalizado para cada uma das consultas mostradas nas Tabelas 5.4 e 5.5. A normalização do tempo foi obtida dividindo-se o tempo de execução de cada consulta pelo maior tempo de execução registrado de sua consulta associada, isto é, o tempo de execução sequencial da própria consulta.

Com exceção das consultas Q1, Q7, Q8, Q10 e Q11, que já são rápidas por natureza,

todas as outras consultas apresentam aceleração linear ou sub-linear para todas as configurações de *clusters* de máquinas virtuais, como mostrado nas Figuras 5.7, 5.8 e 5.9. As curvas de aceleração nas Figuras 5.7, 5.8 e 5.9 são mostradas usando escala logarítmica para eixo  $y$  e o eixo  $x$  representa o número de máquinas virtuais presentes no *cluster* virtual.

Tabela 5.4: Experimento 1: Tempo médio de execução (T) em segundos e Desvio Padrão do tempo (DP) das consultas BME, **sem concorrência**.

Consulta	VMs=1		VMs=2		VMs=4		VMs=8		VMs=16	
	T	DP	T	DP	T	DP	T	DP	T	DP
<b>Q01</b>	18,60	1,82	9,80	1,79	5,00	0,00	3,60	0,55	2,75	0,50
<b>Q02</b>	59,40	4,72	31,00	7,87	14,60	0,55	8,60	0,89	5,25	0,50
<b>Q03</b>	55,40	0,55	30,60	6,99	14,60	0,55	8,60	0,89	5,20	1,10
<b>Q04</b>	78,20	1,10	40,60	6,43	19,80	0,45	11,00	0,71	6,20	0,45
<b>Q05</b>	55,00	2,00	29,60	3,21	15,00	0,71	8,00	0,71	5,00	0,71
<b>Q06</b>	44,60	0,89	22,80	1,30	11,40	0,55	6,80	0,45	4,20	0,84
<b>Q07</b>	11,60	2,30	6,00	0,71	3,80	0,84	2,20	0,45	2,00	0,00
<b>Q08</b>	13,20	2,86	7,00	1,41	3,80	1,10	2,60	0,55	2,40	0,55
<b>Q09</b>	125,00	5,96	61,20	3,56	30,40	0,55	16,60	0,55	9,20	0,45
<b>Q10</b>	14,20	2,17	7,40	0,55	4,20	0,45	2,80	0,45	2,20	0,45
<b>Q11</b>	17,40	0,55	10,80	2,95	5,00	0,00	3,40	0,55	2,80	0,45
<b>Q12</b>	146,60	23,29	81,80	20,90	37,20	0,84	19,00	0,71	11,80	2,49
<b>Q13</b>	172,20	5,76	81,20	4,97	41,40	0,89	22,40	0,55	12,40	0,55
<b>Q14</b>	79,00	6,44	37,00	0,71	20,20	0,84	11,80	0,84	8,80	1,10

#### 5.4.2 Experimento 2: Grupos de Consultas BME, sem Concorrência

O segundo experimento consistiu na execução de 4 grupos de consultas, cada um contendo as 14 consultas selecionadas do BME, com as seguintes ordens:

- G1 = {Q12, Q04, Q02, Q05, Q09, Q13, Q10, Q11, Q07, Q08, Q03, Q06, Q14, Q01}

Tabela 5.5: Experimento 1: Tempo médio de execução (T) em segundos e Desvio Padrão do tempo (DP) das consultas BME-TABULADOR, **sem concorrência**.

Consulta	VMs=1		VMs=2		VMs=4		VMs=8		VMs=16	
	T	DP	T	DP	T	DP	T	DP	T	DP
<b>T3459</b>	271,60	1,67	133,00	1,00	70,60	0,55	39,60	1,14	25,20	0,84
<b>T3525</b>	1322,80	13,41	658,60	16,94	350,60	0,55	198,00	15,02	141,60	4,22
<b>T3534</b>	1145,00	5,29	547,40	0,89	293,80	0,84	150,40	3,65	103,20	4,71

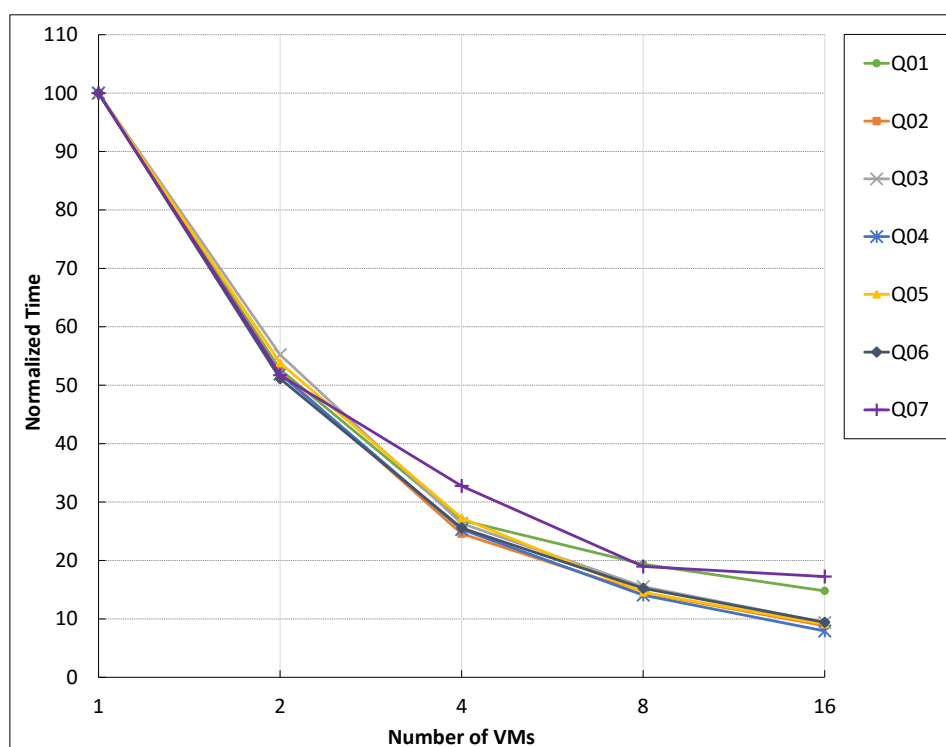


Figura 5.4: Experimento 1: Tempos de execução normalizados das consultas BME Q01-Q07, **sem concorrência**.

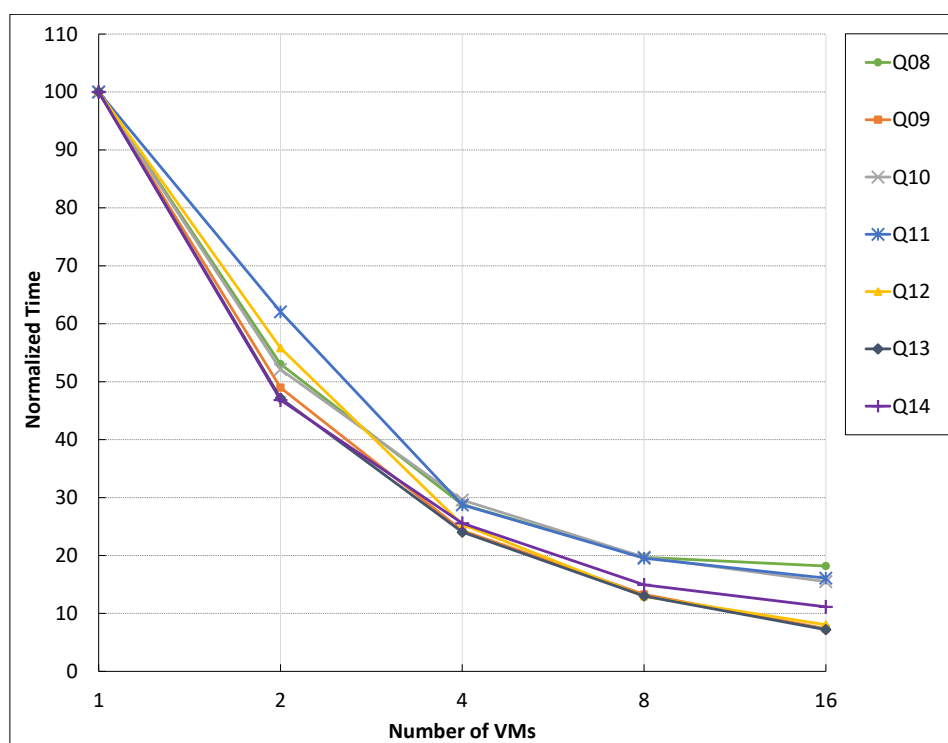


Figura 5.5: Experimento 1: Tempos de execução normalizados das consultas BME Q08-Q14, **sem concorrência**.

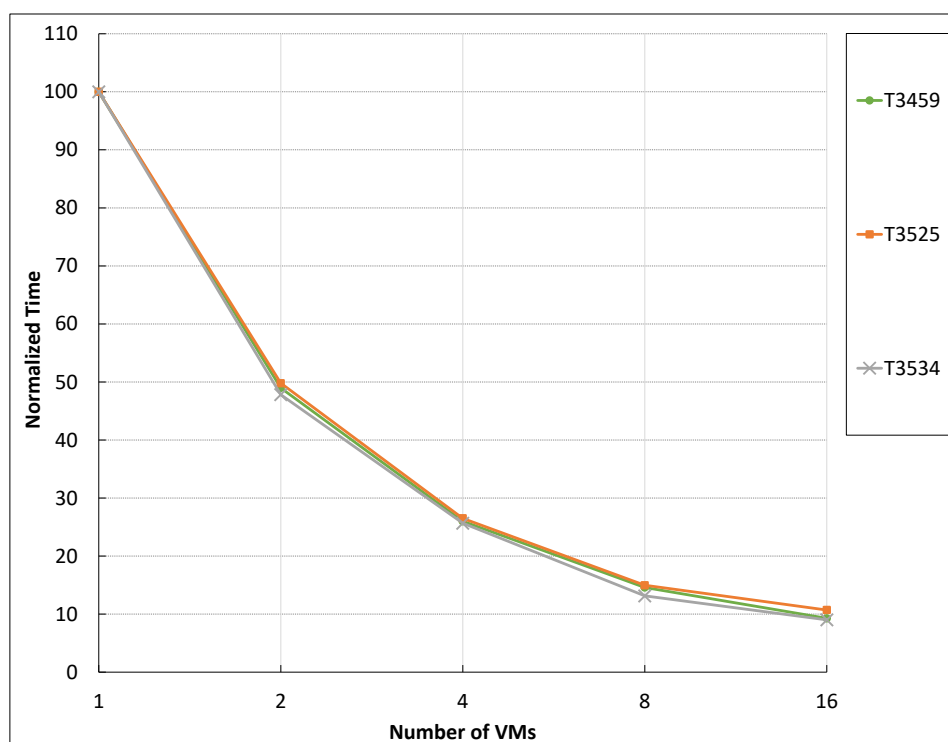


Figura 5.6: Experimento 1: Tempos de execução normalizados das consultas BME-TABULADOR, **sem concorrência**.



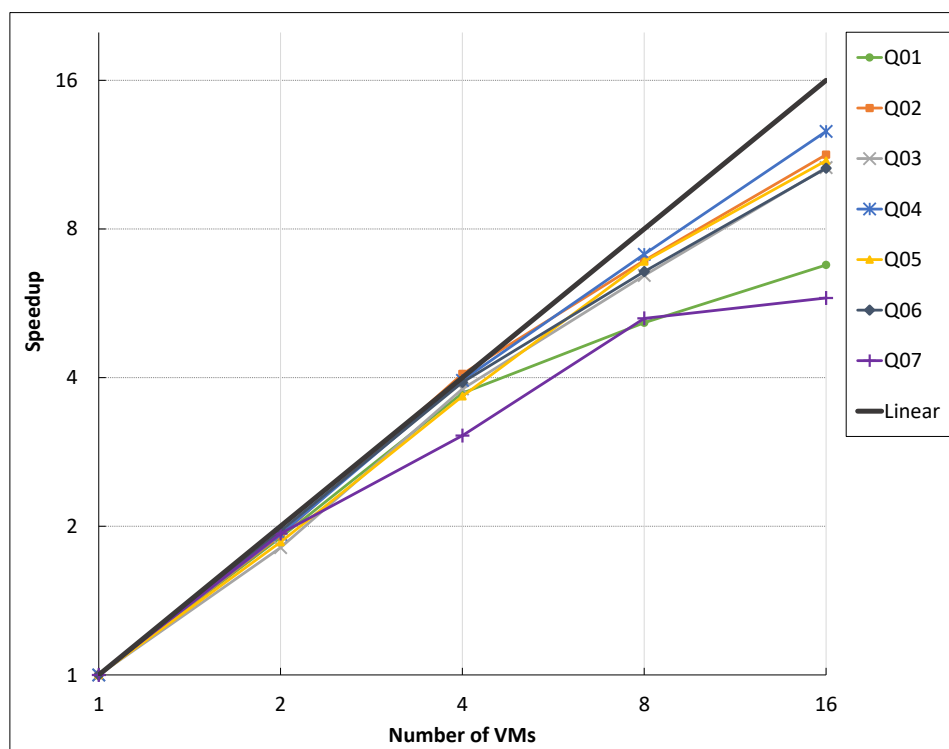


Figura 5.7: Experimento 1: *Speedup* das consultas BME Q01-Q07, **sem concorrência**.

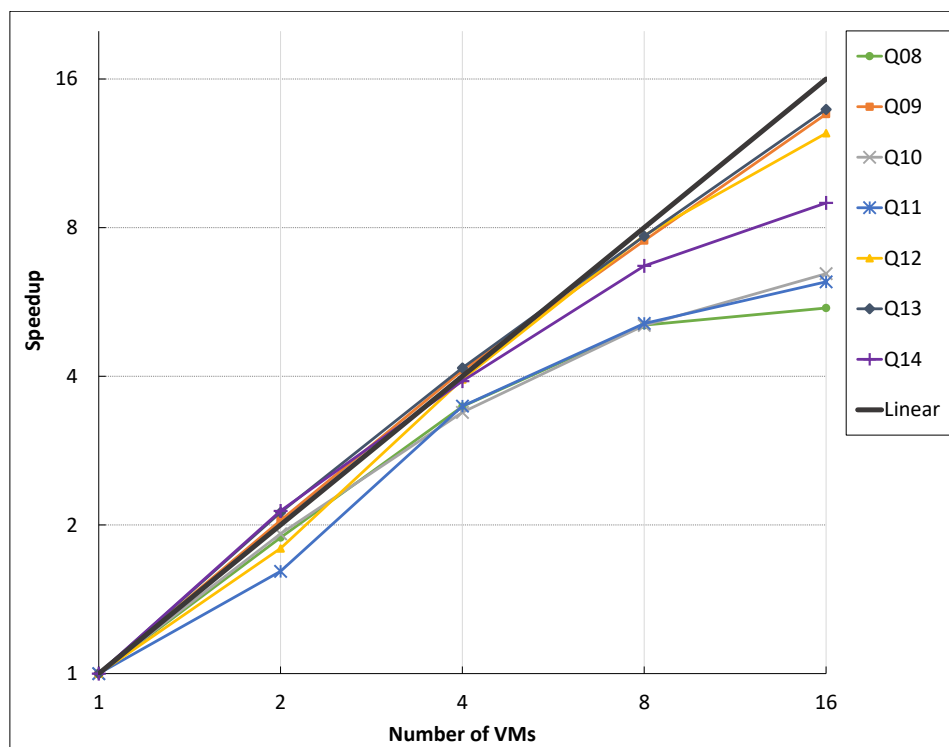


Figura 5.8: Experimento 1: *Speedup* das consultas BME Q08-Q14, **sem concorrência**.

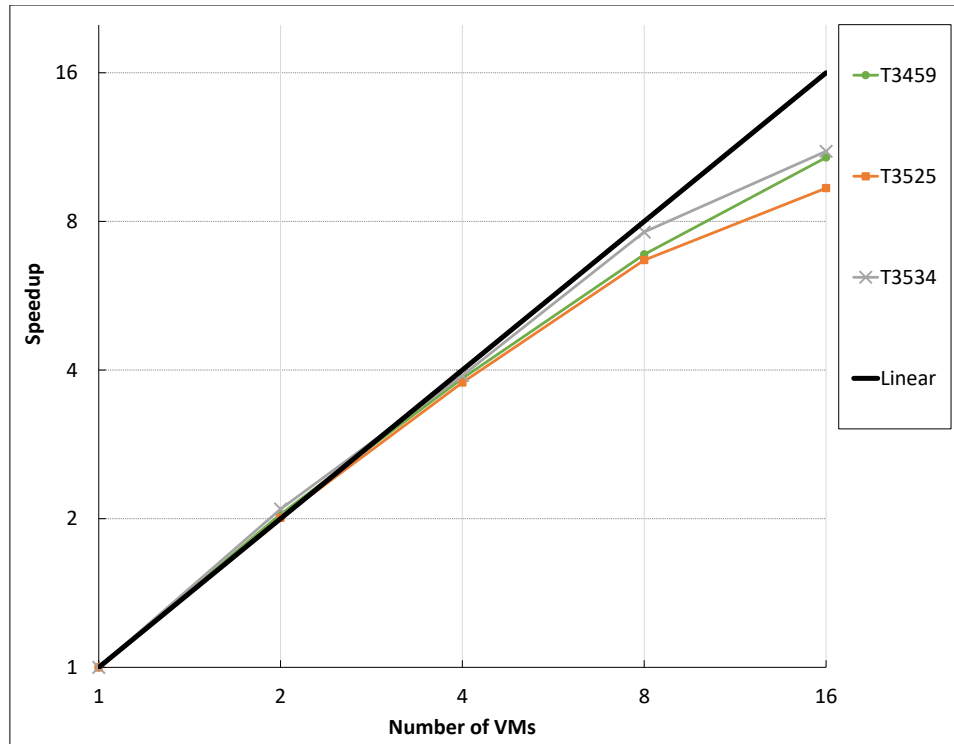


Figura 5.9: Experimento 1: *Speedup* das consultas BME-TABULADOR, **sem concorrência**.

- $G2 = \{Q01, Q10, Q11, Q12, Q07, Q08, Q13, Q03, Q06, Q14, Q02, Q05, Q09, Q04\}$
- $G3 = \{Q14, Q13, Q07, Q08, Q12, Q01, Q03, Q06, Q04, Q10, Q11, Q02, Q05, Q09\}$
- $G4 = \{Q04, Q12, Q03, Q06, Q14, Q02, Q05, Q09, Q01, Q13, Q10, Q11, Q07, Q08\}$

Os grupos e as ordens das consultas foram propostos por PAES [10] para simular o comportamento que os usuários têm na aplicação BME. Os grupos de consultas do BME foram executados de forma sequencial e sem concorrência com 1, 2, 4, 8 e 16 instâncias de máquinas virtuais. Realizamos 5 execuções de cada grupo de consultas para obtermos a média do tempo de execução (em segundos) para cada consulta, independente da ordem de execução e grupo. Todas as consultas foram processadas com o paralelismo intra-consulta.

A Tabela 5.6 apresenta os tempos de execução médios e o desvio padrão do tempo médio de execução para cada consulta do BME e quantidades de máquinas virtuais que compõem o *cluster* virtual, independentemente do grupo no qual a consulta foi executada. A Tabela 5.7 apresenta os tempos de execução médios e o desvio padrão do tempo médio de execução para cada grupo de consultas do BME e quantidades de máquinas virtuais que compõem o *cluster* virtual. As Tabelas 5.6 e 5.7 fornecem evidências dos bons resultados

obtidos, que também podem ser observados nas Figuras 5.10, 5.11, 5.12, 5.13, 5.14 e 5.15.

Tabela 5.6: Experimento 2: Tempo médio de execução (T) em segundos e Desvio Padrão do tempo (DP) das consultas BME, grupos de consultas **sem concorrência**.

Consulta	VMs=1		VMs=2		VMs=4		VMs=8		VMs=16	
	T	DP	T	DP	T	DP	T	DP	T	DP
<b>Q01</b>	17,80	0,37	9,60	0,28	5,05	0,10	2,95	0,30	1,55	0,34
<b>Q02</b>	57,60	2,96	29,55	0,19	14,95	0,19	8,15	1,14	4,40	0,37
<b>Q03</b>	56,35	0,60	29,00	0,28	14,50	0,12	8,05	0,77	4,50	0,26
<b>Q04</b>	79,90	2,21	40,50	0,26	20,40	0,43	11,15	1,77	5,60	0,16
<b>Q05</b>	56,05	1,18	29,75	0,25	15,25	0,66	8,25	0,84	4,55	0,19
<b>Q06</b>	44,50	0,48	23,25	0,25	11,60	0,28	6,90	0,70	3,50	0,12
<b>Q07</b>	11,75	1,80	6,70	0,48	3,60	0,28	2,30	0,12	1,75	0,25
<b>Q08</b>	12,00	0,98	5,80	0,16	3,50	0,20	2,45	0,19	1,90	0,12
<b>Q09</b>	129,90	4,75	62,65	0,44	31,20	0,00	16,15	0,57	8,25	0,25
<b>Q10</b>	14,50	1,05	7,95	0,34	4,40	0,16	2,45	0,25	1,70	0,12
<b>Q11</b>	18,15	0,34	10,05	0,19	5,15	0,19	3,45	0,25	2,40	0,23
<b>Q12</b>	147,30	9,79	79,00	2,99	40,35	5,38	18,78	0,49	9,85	0,19
<b>Q13</b>	173,55	3,65	84,10	0,48	42,30	0,20	22,05	1,33	11,90	0,12
<b>Q14</b>	78,30	1,94	40,30	0,48	21,00	0,43	11,55	0,98	7,80	0,43

As Figuras 5.10 e 5.11 apresentam o tempo de execução normalizado para cada uma das consultas mostradas na Tabela 5.6. A normalização do tempo foi obtida dividindo-se o tempo de execução de cada consulta pelo maior tempo de execução registrado de sua consulta associada independente do grupo no qual foi executada, isto é, o tempo de execução sequencial da própria consulta.

A Figura 5.12 apresenta o tempo de execução normalizado para os grupos de consultas mostrados na Tabela 5.7. A normalização do tempo foi obtida dividindo-se o tempo de execução de cada grupo pelo maior tempo de execução registrado de seu grupo associado, isto é, o tempo de execução sequencial do próprio grupo.

Com exceção das consultas Q7, Q8 e Q11, que já são rápidas por natureza, todas as

Tabela 5.7: Experimento 2: Tempo médio de execução (T) em segundos e Desvio Padrão do tempo (DP) dos grupos de consultas BME, **sem concorrência**.

Grupo	VMs=1		VMs=2		VMs=4		VMs=8		VMs=16	
	T	DP	T	DP	T	DP	T	DP	T	DP
<b>G1</b>	893,00	20,42	463,40	10,01	244,60	27,66	137,30	13,83	70,00	1,22
<b>G2</b>	923,40	67,21	459,60	3,05	230,20	0,84	122,40	2,19	69,20	1,30
<b>G3</b>	899,40	15,85	456,40	3,78	229,80	0,45	119,40	1,82	69,40	1,34
<b>G4</b>	874,80	26,20	453,40	3,44	228,40	1,52	119,40	2,07	70,00	2,35
<b>Total</b>	3590,60	-	1832,80	-	933,00	-	498,50	-	278,60	-

outras consultas apresentam aceleração linear ou sub-linear para todas as configurações de *clusters* de máquinas virtuais, como mostrado nas Figuras 5.13 e 5.14. A análise dos resultados por grupos, sem levar em consideração o desempenho individual de cada consulta, aponta uma aceleração linear para todas as configurações de *clusters* de máquinas virtuais, como mostrado na Figura 5.15. As curvas de aceleração nas Figuras 5.13, 5.14 e 5.15 são mostradas usando escala logarítmica para eixo *y* e o eixo *x* representa o número de máquinas virtuais presentes no *cluster* virtual.

### 5.4.3 Experimento 3: Grupos de Consultas BME, com Concorrência

A execução concorrente de consultas é uma característica presente na aplicação BME. Decidimos então avaliar o comportamento do C-ParGRES diante de um cenário deste tipo. Dessa forma, o terceiro experimento consistiu na execução dos 4 grupos de consultas definidos na seção 5.4.2, porém dessa vez sendo executados de forma concorrente através de 4 sessões de terminal distintas. Cada sessão de terminal foi responsável por executar completamente o grupo que lhe foi atribuído, sendo uma consulta de cada vez, ou seja, em qualquer momento havia no máximo 4 consultas em execução concorrente no *cluster* virtual. Ao definir a sequência de consultas de cada grupo, Paes *et al.* [10] ressalta o cuidado para que as consultas não coincidissem nas mesmas posições entre os grupos, ou seja, duas consultas iguais não foram executadas simultaneamente, reduzindo a possibilidade de uma consulta se beneficiar do processamento em memória de outra ou, inversamente, atrasar a execução de ambas devido à concorrência de acesso aos mesmos setores do disco.

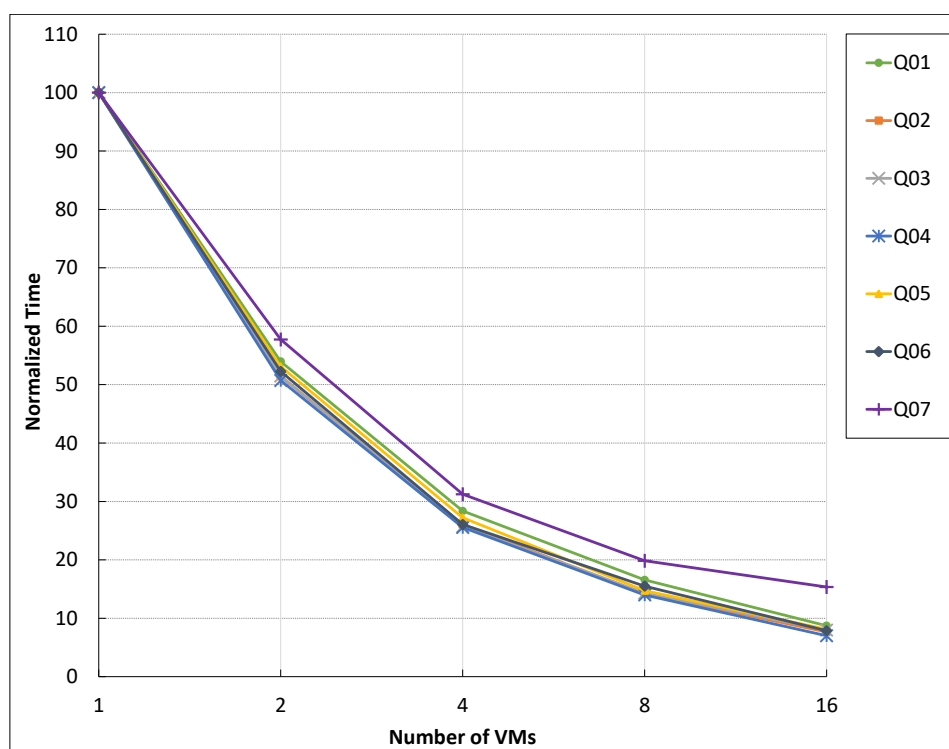


Figura 5.10: Experimento 2: Tempos de execução normalizados das consultas BME Q01-Q07, grupos de consultas **sem concorrência**.

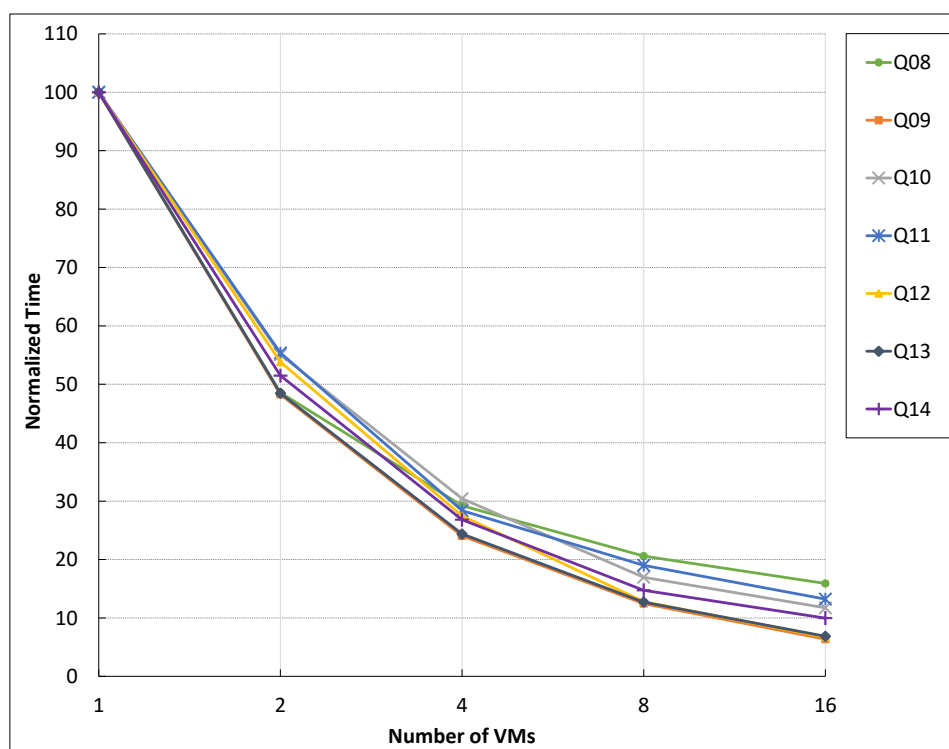


Figura 5.11: Experimento 2: Tempos de execução normalizados das consultas BME Q08-Q14, grupos de consultas **sem concorrência**.

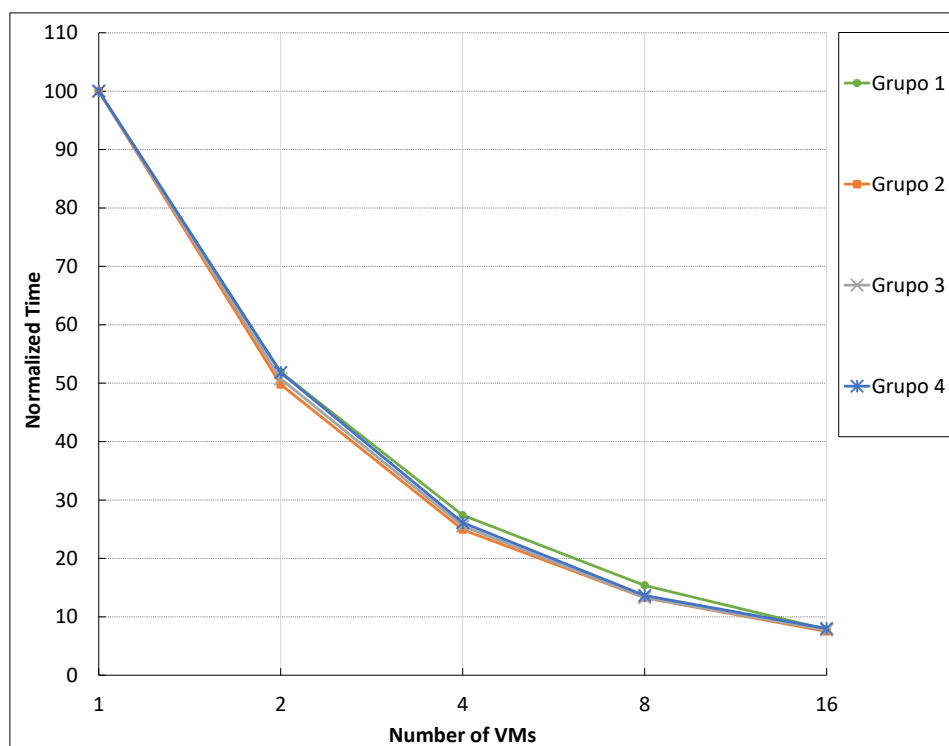


Figura 5.12: Experimento 2: Tempos de execução normalizados dos grupos de consultas BME G1-G4, **sem concorrência**.

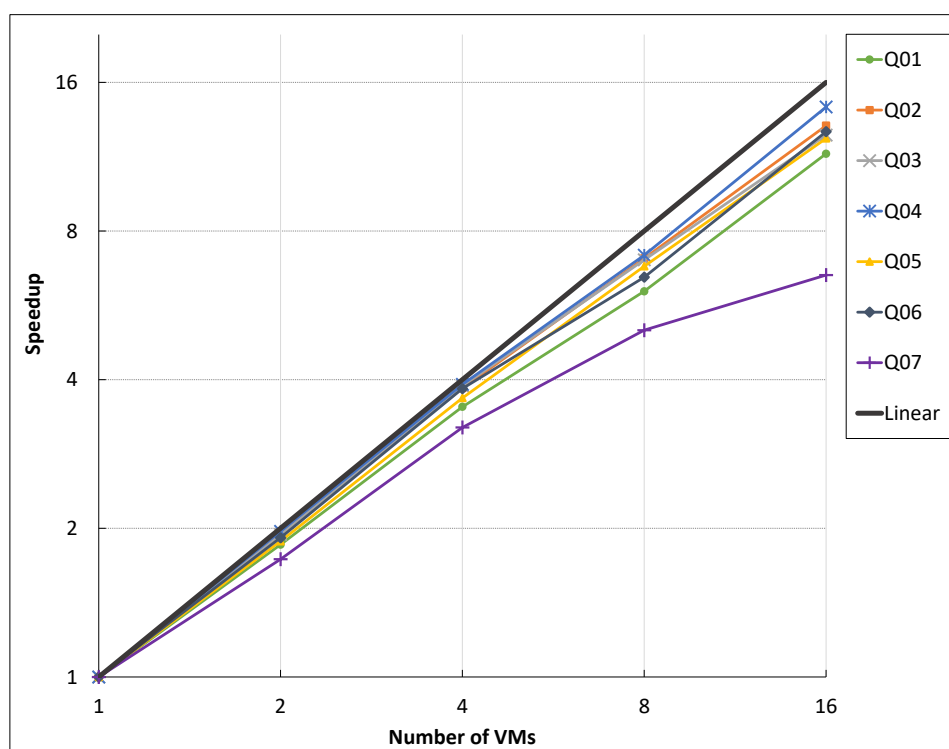


Figura 5.13: Experimento 2: *Speedup* das consultas BME Q01-Q07, grupos de consultas **sem concorrência**.

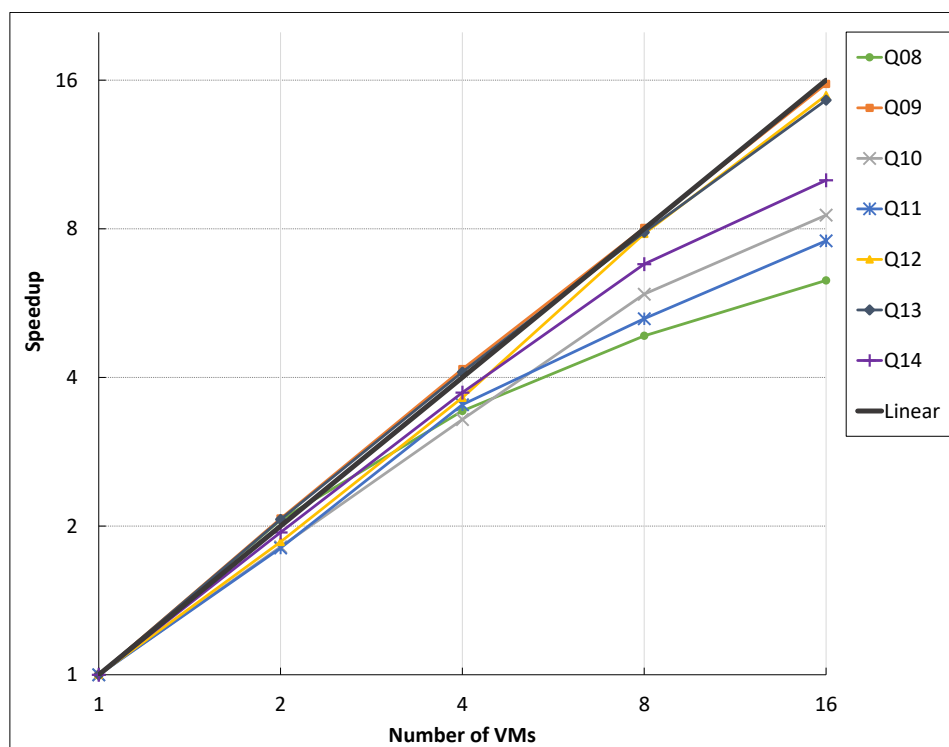


Figura 5.14: Experimento 2: *Speedup* das consultas BME Q08-Q14, grupos de consultas sem concorrência.

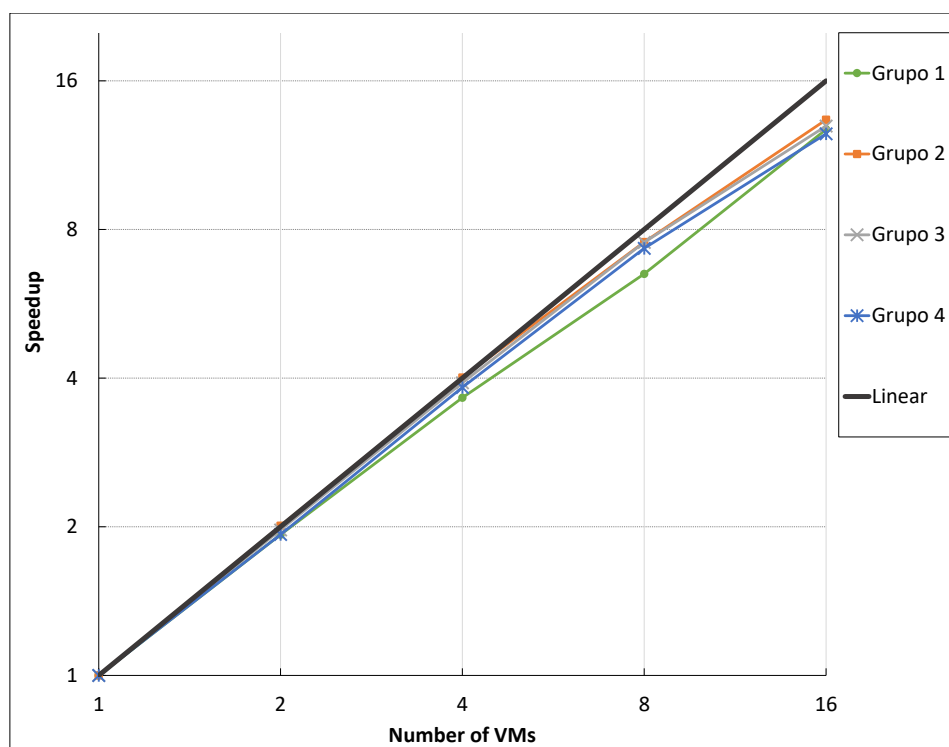


Figura 5.15: Experimento 2: *Speedup* dos grupos de consultas BME G1-G4, **sem concorrência**.

Neste experimento, os grupos selecionados foram executados simultaneamente usando 2, 4, 8 e 16 máquinas virtuais. Realizamos 5 execuções de cada grupo de consulta para obter o tempo médio de execução (em segundos). Todas as consultas foram processadas com paralelismo intra-consulta.

A Tabela 5.8 apresenta os tempos de execução médios e o desvio padrão do tempo médio de execução para cada consulta do BME e quantidades de máquinas virtuais que compõem o *cluster* virtual, independentemente do grupo no qual a consulta foi executada. A Tabela 5.9 apresenta os tempos de execução médios e o desvio padrão do tempo médio de execução para cada grupo de consultas do BME e quantidades de máquinas virtuais que compõem o *cluster* virtual. Os mesmos dados podem ser observados nos gráficos das Figuras 5.16, 5.17, 5.18, 5.19, 5.20 e 5.21.

As Figuras 5.16 e 5.17 apresentam tempo de execução normalizado para cada consulta apresentada na Tabela 5.8. A normalização do tempo foi obtida dividindo-se o tempo de execução de cada consulta pelo maior tempo de execução registrado de sua consulta associada independente do grupo no qual foi executada, isto é, o tempo de execução sequencial da própria consulta.

A Figura 5.18 apresenta o tempo de execução normalizado para os grupos de consultas mostrados na Tabela 5.9. A normalização do tempo foi obtida dividindo-se o tempo de execução de cada grupo pelo maior tempo de execução registrado de seu grupo associado, isto é, o tempo de execução sequencial do próprio grupo.

Diferentemente do primeiro experimento, nem todas as consultas apresentaram acelerações lineares ou sub-lineares. Q7, Q8, Q10 e Q11 apresentaram uma degradação de velocidade ao usar 16 máquinas virtuais. De fato, todas as consultas apresentaram um aumento de velocidade linear ou sub-linear até 8 máquinas virtuais. No caso das consultas Q7, Q8, Q10 e Q11, elas já são rápidas, por isso é difícil para o C-ParGRES melhorar o tempo de execução. Além disso, o banco de dados particionado virtualmente pode se encaixar na memória principal das 8 máquinas virtuais, o que explica a aceleração para todas as consultas nas Figuras 5.19 e 5.20 para execuções com até 8 máquinas virtuais.

A análise dos resultados por grupos, sem levar em consideração o desempenho individual de cada consulta, aponta uma aceleração linear para todas as configurações de *clusters* de máquinas virtuais, como mostrado na Figura 5.21. As curvas de aceleração nas Figuras 5.19, 5.20 e 5.21 são mostradas usando escala logarítmica para eixo *y* e o eixo *x* representa o número de máquinas virtuais presentes no *cluster* virtual.



Tabela 5.8: Experimento 3: Tempo médio de execução (T) em segundos e Desvio Padrão do tempo (DP) das consultas BME, grupos de consultas **com concorrência**.

Consulta	VMs=2		VMs=4		VMs=8		VMs=16	
	T	DP	T	DP	T	DP	T	DP
<b>Q01</b>	10,20	0,23	5,59	0,14	3,40	0,40	3,20	0,43
<b>Q02</b>	33,20	0,59	17,05	0,85	8,60	0,43	7,75	0,38
<b>Q03</b>	33,35	0,30	17,05	0,19	8,60	0,28	7,95	0,55
<b>Q04</b>	44,70	0,95	23,11	0,84	11,60	0,63	11,15	0,89
<b>Q05</b>	33,75	0,53	17,25	0,10	8,55	0,30	8,20	0,59
<b>Q06</b>	26,75	0,34	13,90	0,26	7,60	0,63	6,55	0,77
<b>Q07</b>	7,35	0,25	3,85	0,25	2,50	0,20	4,20	0,49
<b>Q08</b>	7,30	0,48	4,50	0,66	2,75	0,34	4,35	0,60
<b>Q09</b>	73,40	1,18	36,20	0,57	18,50	0,76	15,30	0,96
<b>Q10</b>	8,35	0,30	4,80	0,16	3,15	0,70	4,55	0,53
<b>Q11</b>	10,50	0,42	6,55	0,41	3,70	0,53	5,55	0,94
<b>Q12</b>	87,65	3,59	43,48	1,93	21,74	1,16	18,35	0,30
<b>Q13</b>	95,45	4,58	48,10	0,62	24,75	0,96	19,45	0,57
<b>Q14</b>	49,20	1,05	24,95	0,57	13,00	0,49	11,00	0,59

Tabela 5.9: Experimento 3: Tempo médio de execução (T) em segundos e Desvio Padrão do tempo (DP) dos grupos de consultas BME, **com concorrência**.

Grupo	VMs=2		VMs=4		VMs=8		VMs=16	
	T	DP	T	DP	T	DP	T	DP
<b>G1</b>	521,80	2,49	267,30	11,33	136,95	6,41	129,80	6,30
<b>G2</b>	524,20	3,35	264,55	4,53	143,60	11,63	125,40	6,35
<b>G3</b>	519,00	5,29	264,80	6,30	136,60	4,62	128,00	5,24
<b>G4</b>	519,60	5,98	268,85	13,32	136,60	7,27	127,00	6,60
<b>Total</b>	2084,60	-	1065,50	-	553,75	-	510,20	-

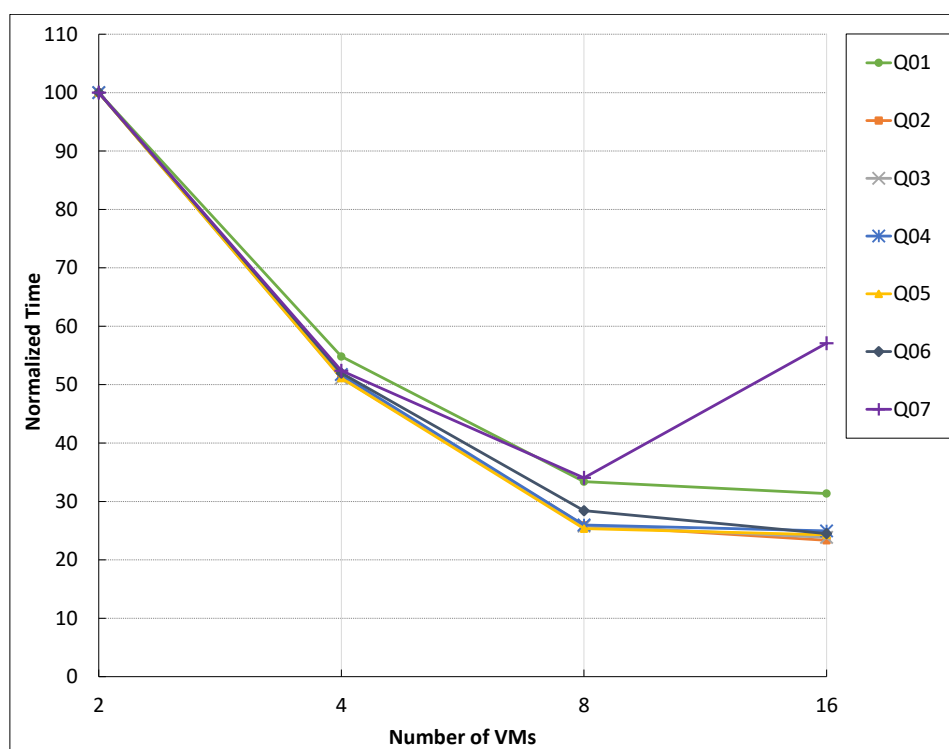


Figura 5.16: Experimento 3: Tempos de execução normalizados das consultas BME Q01-Q07, grupos de consultas **com concorrência**.

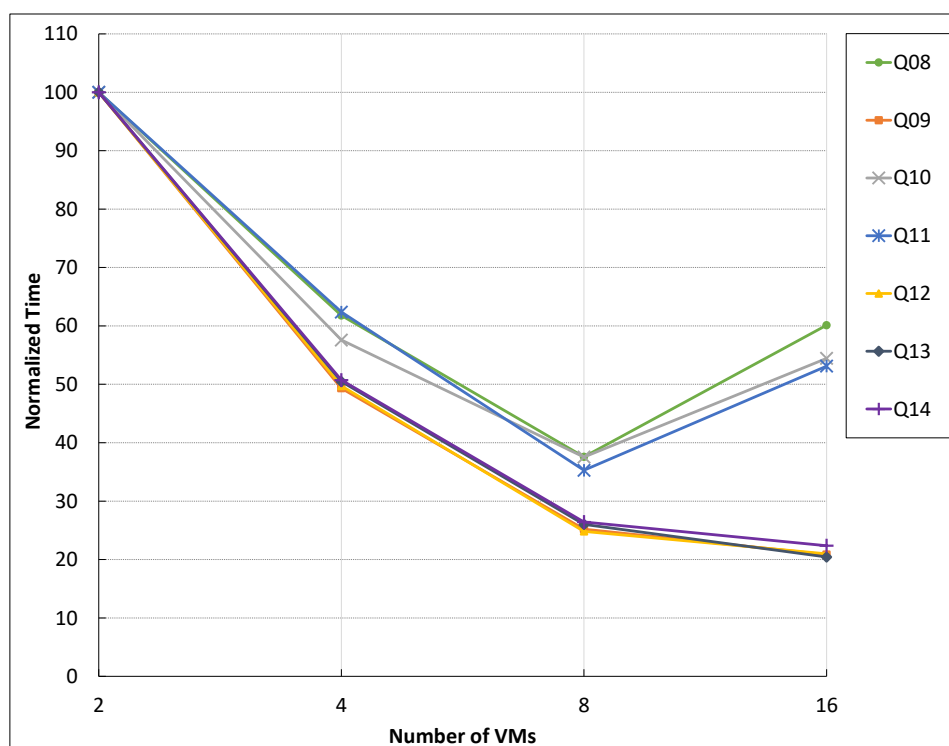


Figura 5.17: Experimento 3: Tempos de execução normalizados das consultas BME Q08-Q14, grupos de consultas **com concorrência**.

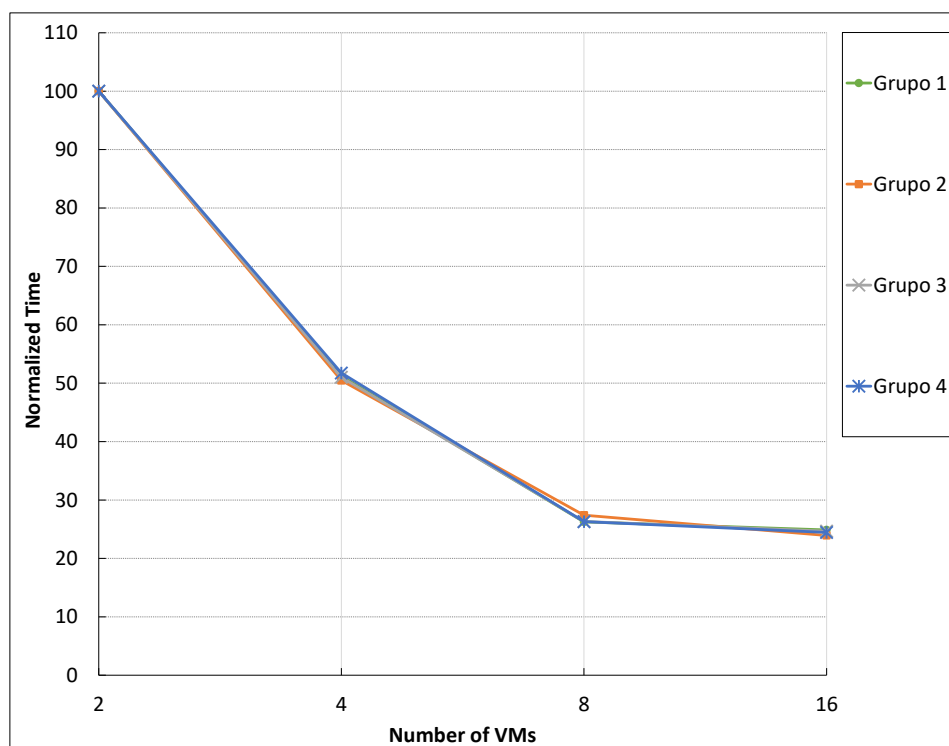


Figura 5.18: Experimento 3: Tempos de execução normalizados dos grupos de consultas BME G1-G4, **com concorrência**.

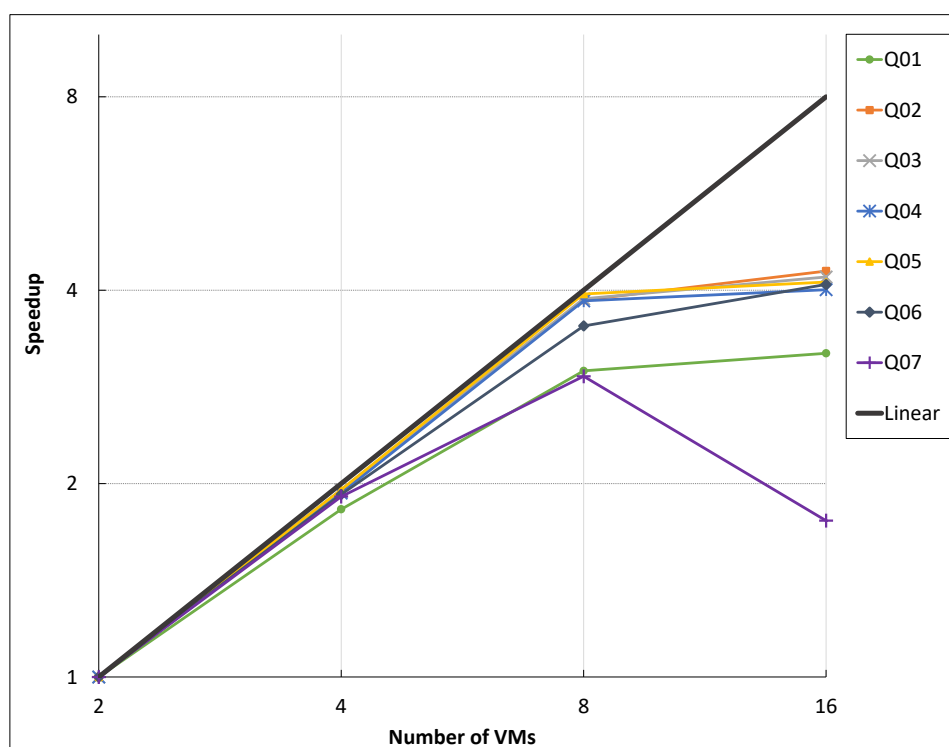


Figura 5.19: Experimento 3: *Speedup* das consultas BME Q01-Q07, grupos de consultas **com concorrência**.

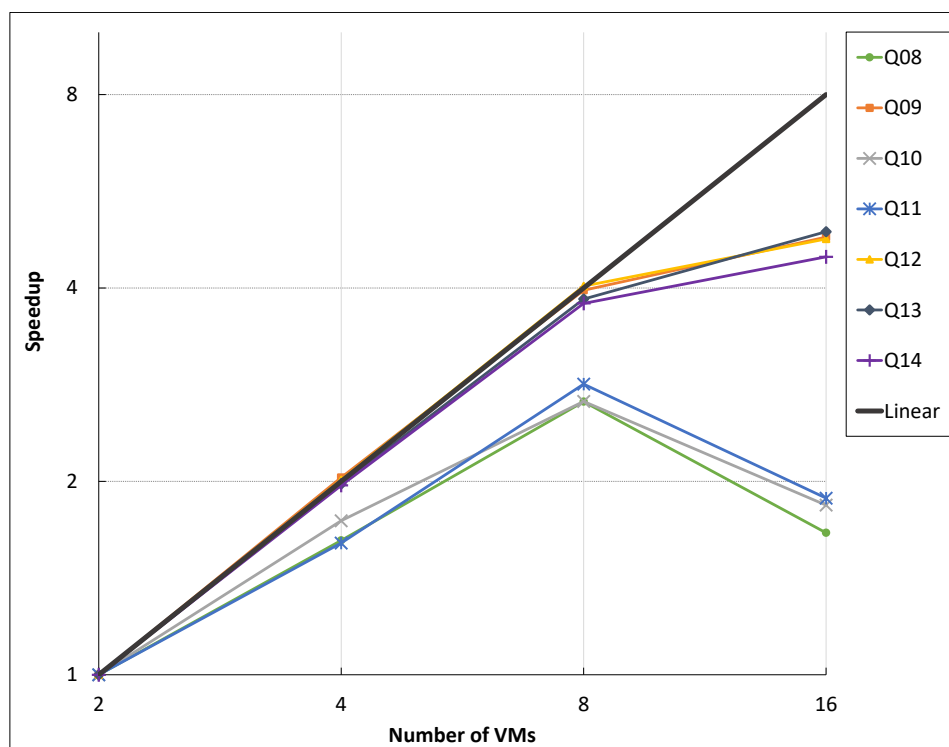


Figura 5.20: Experimento 3: *Speedup* das consultas BME Q08-Q14, grupos de consultas com concorrência.

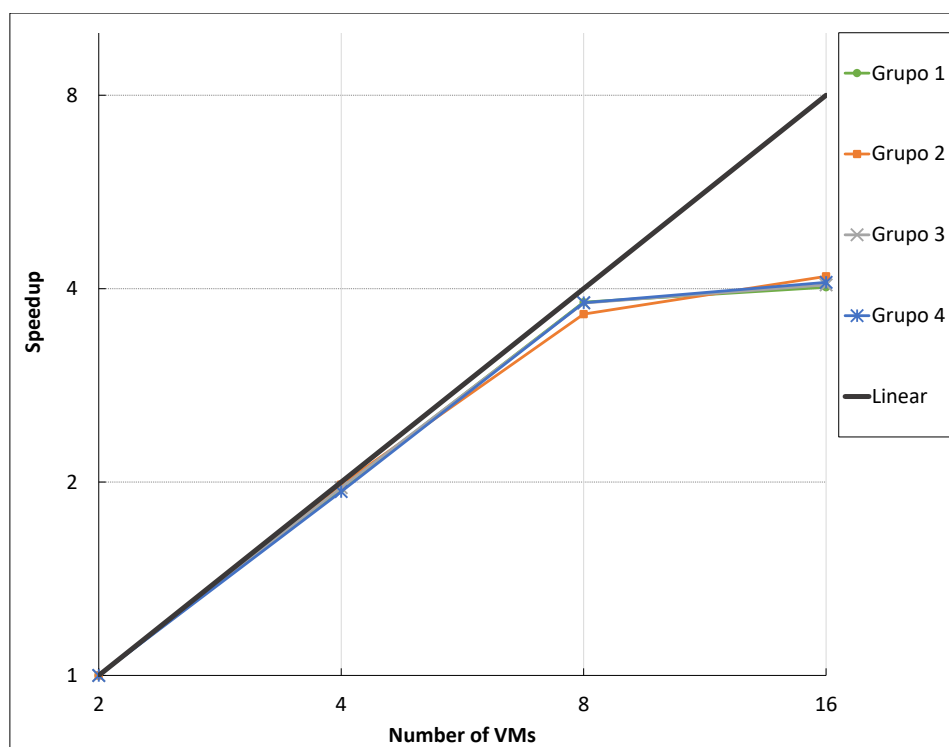


Figura 5.21: Experimento 3: *Speedup* dos grupos de consultas BME G1-G4, com concorrência.

#### 5.4.4 Experimento 4: Consultas BME e BME-TABULADOR, com Concorrência em um Ambiente com Carga Mista

No quarto e último experimento, avaliamos o comportamento do **C-ParGRES** em um ambiente de concorrência similar ao apresentado no experimento 3. Contudo, além dos 4 grupos de 14 consultas do BME executando em sessões de terminal independentes, acrescentamos mais uma sessão com as 3 consultas da aplicação BME-TABULADOR. Conforme já mencionado, as consultas do BME-TABULADOR são consideravelmente mais custosas que as do BME, o que possibilita verificar como o **C-ParGRES** se comporta em ambientes com cargas de trabalho diferenciadas.

Neste experimento, os grupos selecionados foram executados simultaneamente usando 2, 4, 8 e 16 máquinas virtuais. Realizamos 5 execuções de cada grupo de consulta para obter o tempo médio de execução (em segundos). Todas as consultas foram processadas com paralelismo intra-consulta.

A Tabela 5.10 apresenta os tempos de execução médios e o desvio padrão do tempo médio de execução para cada consulta do BME e quantidades de máquinas virtuais que compõem o *cluster* virtual, independentemente do grupo no qual a consulta foi executada. A Tabela 5.11 apresenta os tempos de execução médios e o desvio padrão do tempo médio de execução para cada consulta do BME-TABULADOR. A Tabela 5.12 apresenta os tempos de execução médios e o desvio padrão do tempo médio de execução para cada grupo de consultas do BME e BME-TABULADOR e quantidades de máquinas virtuais que compõem o *cluster* virtual. Os mesmos dados podem ser observados nos gráficos das Figuras 5.22, 5.23, 5.24, 5.25, 5.26, 5.27, 5.28 e 5.29.

As Figuras 5.22, 5.23 e 5.24 apresentam os resultados normalizados do tempo de execução para cada consulta apresentada nas Tabelas 5.10 e 5.11. A normalização foi obtida dividindo-se cada tempo de execução da consulta pelo maior tempo de execução de sua consulta associada, isto é, a execução sequencial da consulta.

A Figura 5.25 apresenta o tempo de execução normalizado para os grupos de consultas mostrados na Tabela 5.12. A normalização do tempo foi obtida dividindo-se o tempo de execução de cada grupo pelo maior tempo de execução registrado de seu grupo associado, isto é, o tempo de execução sequencial do próprio grupo.

Similarmente ao terceiro experimento, Q7, Q8, Q10 e Q11 apresentaram uma degradação de aceleração ao usar 8 e 16 máquinas virtuais. De fato, todas as consultas apresentaram um *speedup* linear ou super linear até 4 máquinas virtuais, provavelmente

devido ao uso do cache, como pode ser observado nas Figuras 5.26, 5.27 e 5.28. No caso das consultas Q7, Q8, Q10 e Q11, elas já são rápidas, por isso é difícil para o C-ParGRES melhorar o tempo de execução adicionando mais máquinas virtuais. Vale a pena notar que as consultas BME-TABULADOR tiveram melhor desempenho em termos de aceleração do que as do BME, pois são mais intensivas em computação e apropriadas ao paralelismo intra-consulta do C-ParGRES.

A análise dos resultados por grupos, sem levar em consideração o desempenho individual de cada consulta, aponta uma aceleração linear ou super para todas as configurações de *clusters* de máquinas virtuais, como mostrado na Figura 5.29. As curvas de aceleração nas Figuras 5.26, 5.27, 5.28 e 5.29 são mostradas usando escala logarítmica para eixo  $y$  e o eixo  $x$  representa o número de máquinas virtuais presentes no *cluster* virtual.

Tabela 5.10: Experimento 4: Tempo médio de execução (T) em segundos e Desvio Padrão do tempo (DP) das consultas BME, **com concorrência** em um ambiente com carga mista.

Consulta	VMs=2		VMs=4		VMs=8		VMs=16	
	T	DP	T	DP	T	DP	T	DP
<b>Q01</b>	16,95	0,91	6,20	0,16	4,60	0,49	3,61	0,35
<b>Q02</b>	58,45	2,68	19,60	0,16	16,75	1,91	9,70	0,48
<b>Q03</b>	58,95	4,91	19,45	0,44	17,30	1,89	9,95	0,75
<b>Q04</b>	73,95	3,83	25,80	1,18	20,30	2,45	13,60	1,19
<b>Q05</b>	56,15	2,85	18,95	0,82	17,25	2,56	9,75	0,75
<b>Q06</b>	44,85	1,11	16,25	0,64	12,45	1,85	7,80	0,91
<b>Q07</b>	12,20	0,91	5,85	1,05	5,55	0,50	4,80	0,43
<b>Q08</b>	12,70	0,74	5,80	0,77	6,00	0,59	6,30	0,35
<b>Q09</b>	121,85	3,95	42,45	0,85	31,80	2,81	18,25	0,89
<b>Q10</b>	14,95	0,64	6,40	0,78	5,50	0,74	6,05	0,64
<b>Q11</b>	19,55	1,72	8,45	1,36	7,55	1,28	6,85	0,30
<b>Q12</b>	136,90	4,45	44,45	0,55	29,35	3,24	22,80	4,36
<b>Q13</b>	160,90	8,28	56,05	1,00	39,10	3,94	25,05	1,09
<b>Q14</b>	76,25	4,14	26,95	0,75	19,65	3,89	13,30	0,26

Tabela 5.11: Experimento 4: Tempo médio de execução (T) em segundos e Desvio Padrão do tempo (DP) das consultas BME-TABULADOR, **com concorrência** em um ambiente com carga mista.

Consulta	VMs=2		VMs=4		VMs=8		VMs=16	
	T	DP	T	DP	T	DP	T	DP
<b>T3459</b>	209,60	36,78	81,40	13,11	59,40	12,36	49,00	34,91
<b>T3525</b>	935,00	258,48	393,60	40,99	285,00	104,09	137,00	1,41
<b>T3534</b>	758,80	196,72	320,20	27,14	186,80	33,83	152,60	56,63

Tabela 5.12: Experimento 4: Tempo médio de execução (T) em segundos e Desvio Padrão do tempo (DP) dos grupos de consultas BME e BME-TABULADOR, **com concorrência** em um ambiente com carga mista.

Grupo	VMs=2		VMs=4		VMs=8		VMs=16	
	T	DP	T	DP	T	DP	T	DP
<b>G1</b>	861,80	74,15	303,40	3,71	234,40	53,82	163,60	25,40
<b>G2</b>	875,60	46,23	303,00	3,54	232,20	61,19	153,25	3,11
<b>G3</b>	863,20	71,71	302,20	11,01	233,40	63,69	155,40	11,41
<b>G4</b>	857,80	48,42	302,00	5,52	232,60	51,64	159,00	15,30
<b>Tabulador</b>	1903,40	456,99	795,20	77,79	531,20	136,98	338,60	57,37
<b>Total</b>	5361,80		2005,80		1463,80		969,85	

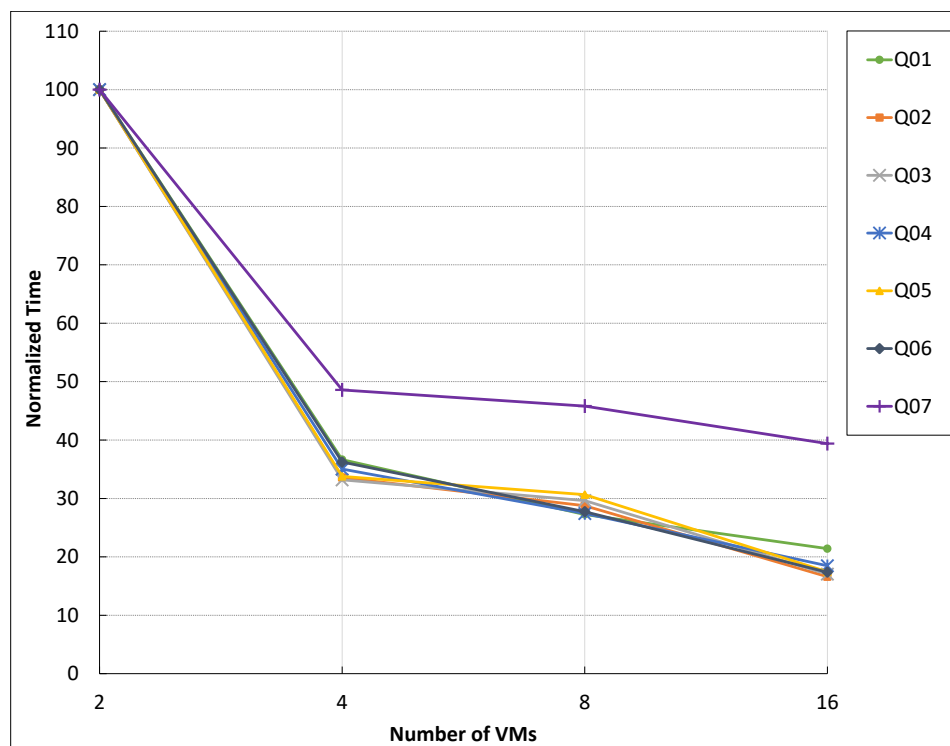


Figura 5.22: Experimento 4: Tempos de execução normalizados das consultas BME Q01-Q07, **com concorrência** em um ambiente com carga mista.

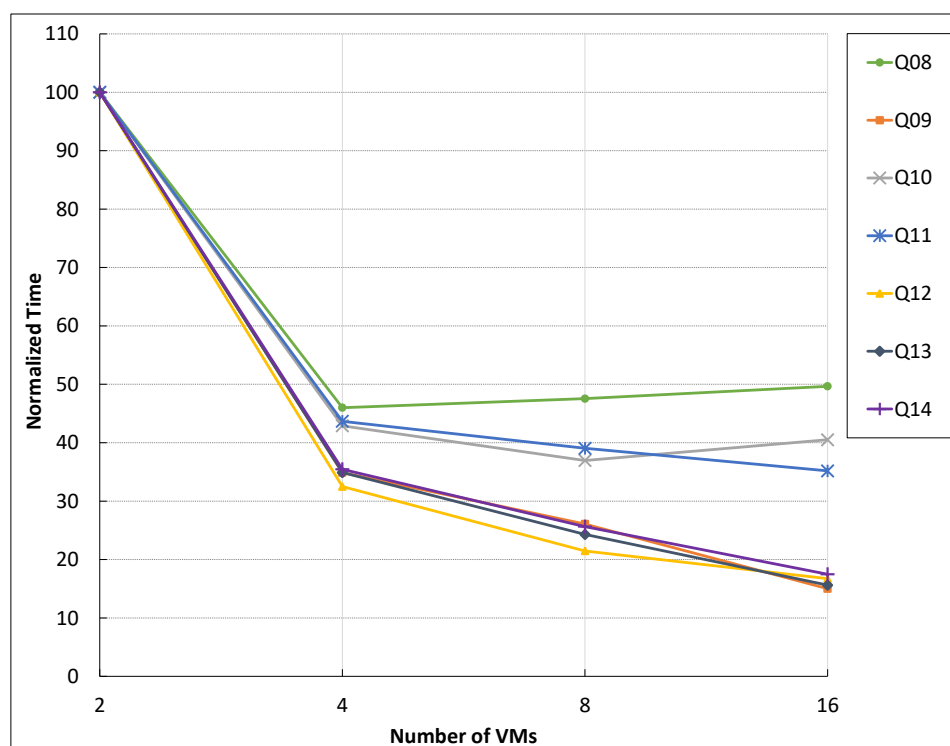


Figura 5.23: Experimento 4: Tempos de execução normalizados das consultas BME Q08-Q14, **com concorrência** em um ambiente com carga mista.



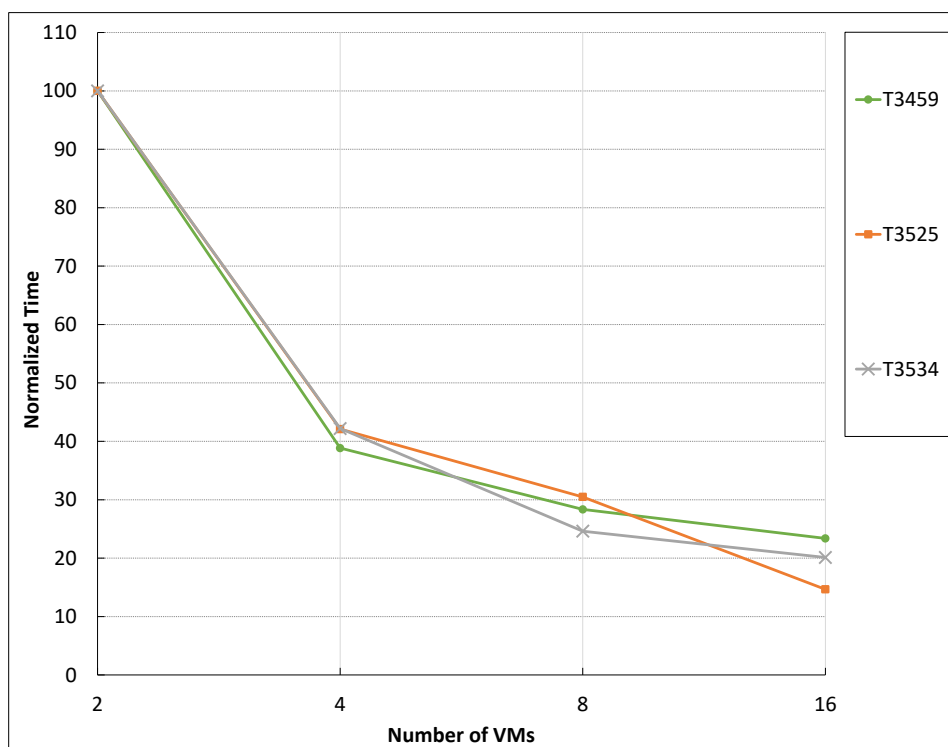


Figura 5.24: Experimento 4: Tempos de execução normalizados das consultas BME-TABULADOR, **com concorrência** em um ambiente com carga mista.

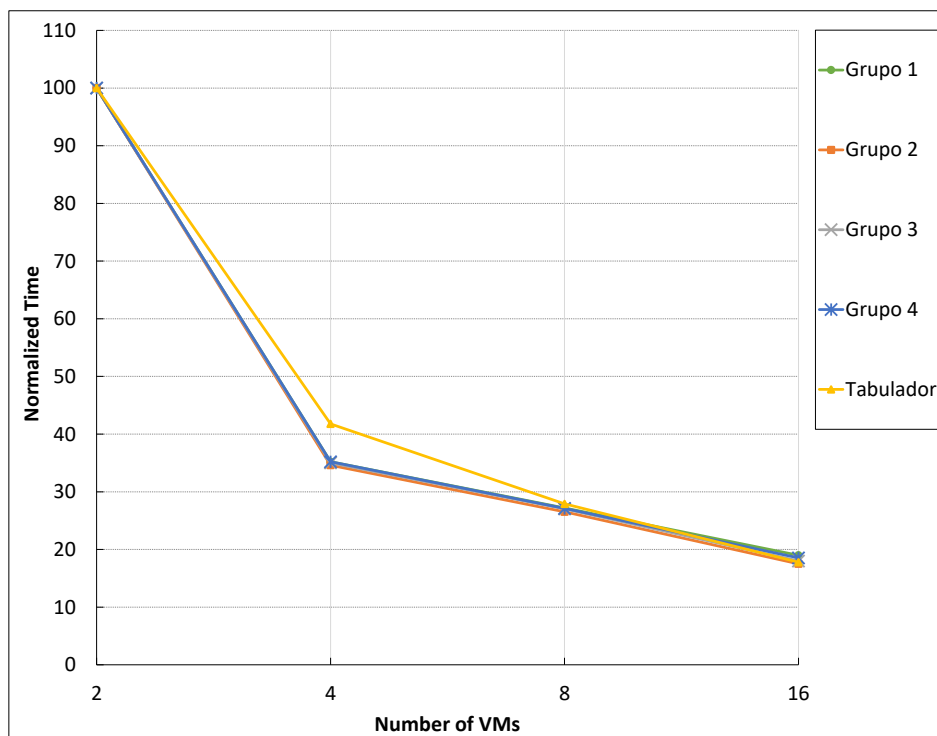


Figura 5.25: Experimento 4: Tempos de execução normalizados dos grupos de consultas BME G1-G4 e BME-TABULADOR, **com concorrência** em um ambiente com carga mista.

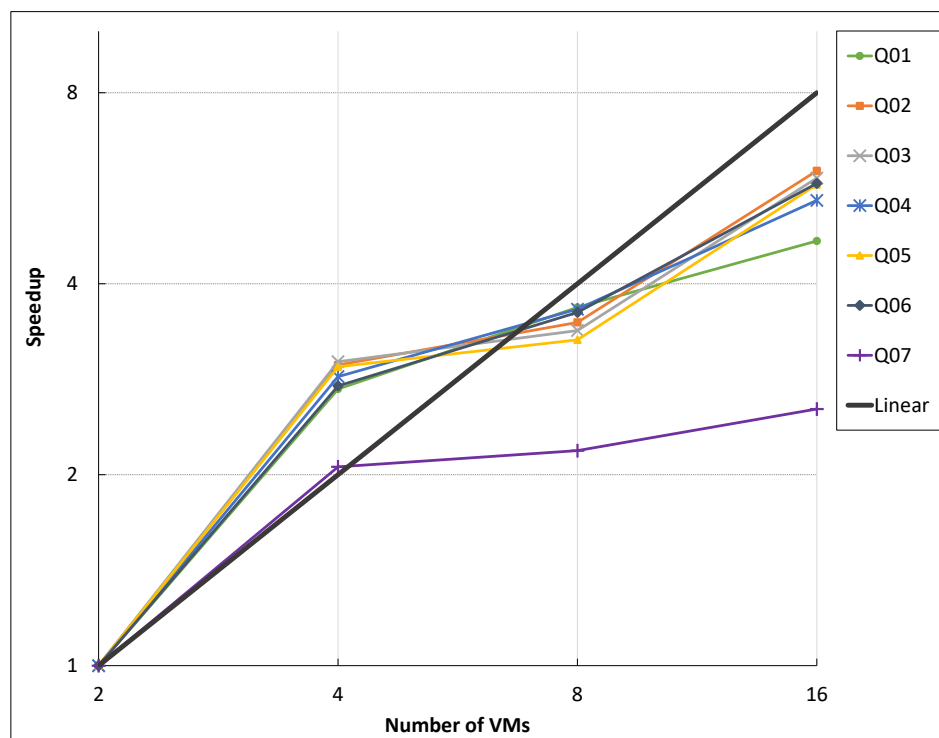


Figura 5.26: Experimento 4: *Speedup* das consultas BME Q01-Q07, com concorrência em um ambiente com carga mista.

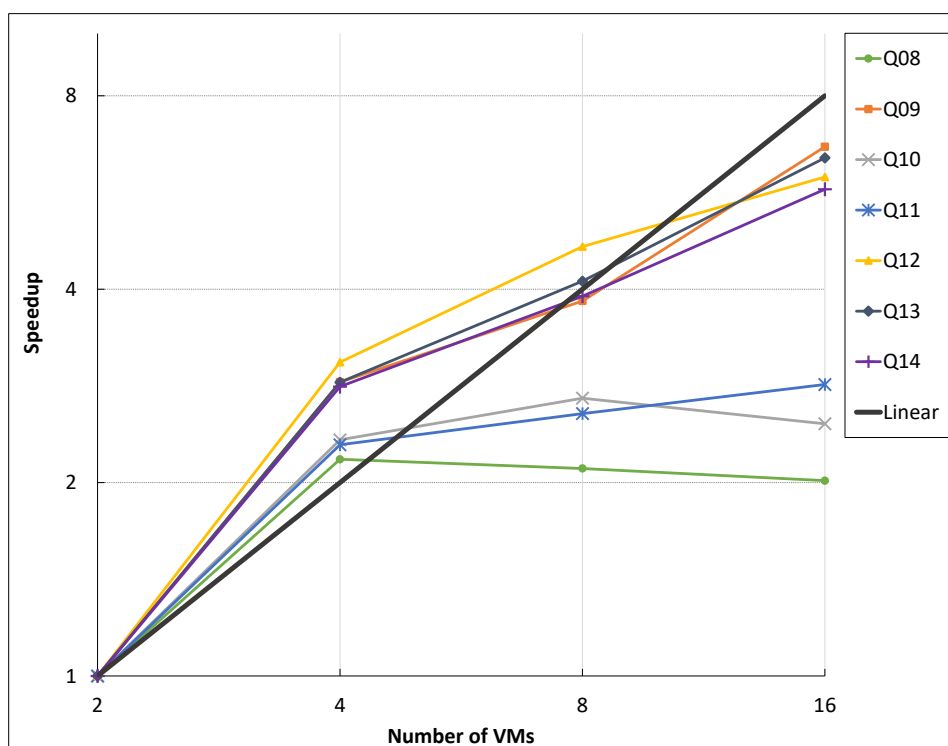


Figura 5.27: Experimento 4: *Speedup* das consultas BME Q08-Q14, com concorrência em um ambiente com carga mista.

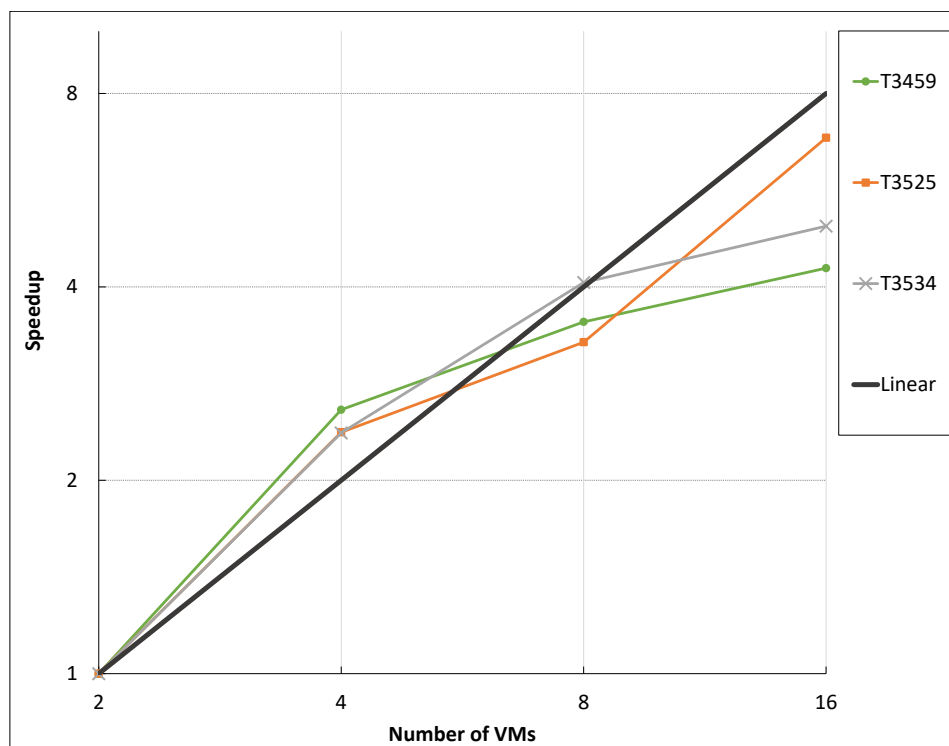


Figura 5.28: Experimento 4: *Speedup* das consultas BME-TABULADOR, **com concorrência** em um ambiente com carga mista.

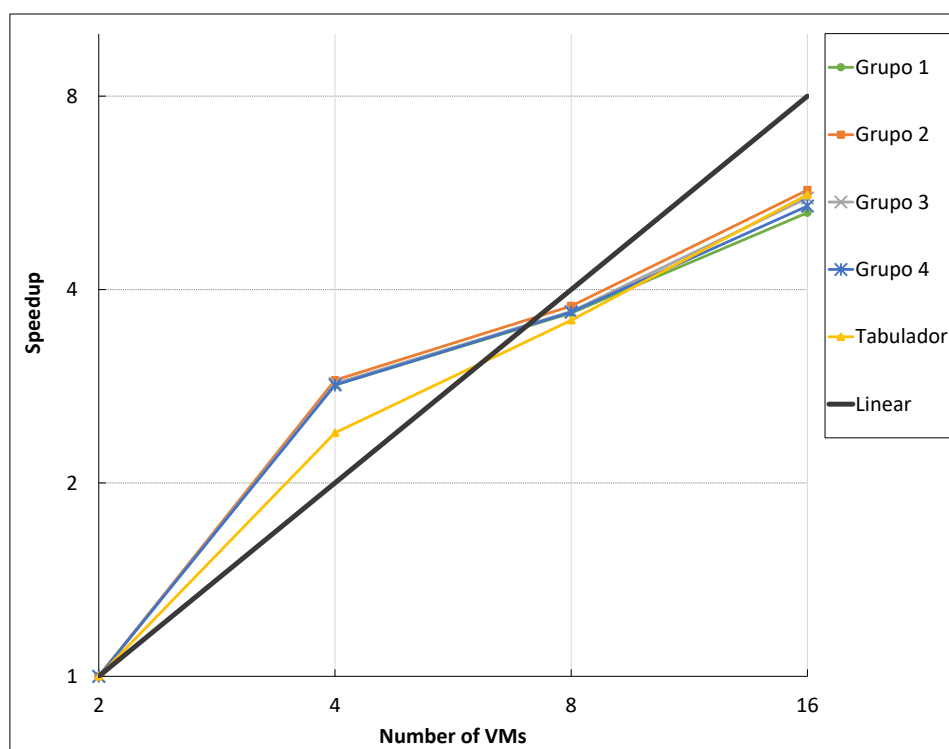


Figura 5.29: Experimento 4: *Speedup* dos grupos de consultas BME G1-G4 e BME-TABULADOR, **com concorrência** em um ambiente com carga mista.

# Capítulo 6

## Trabalhos Relacionados

Este capítulo apresenta trabalhos relacionados com esta dissertação, ou seja, abordagens semelhantes para melhorar o desempenho do processamento de consultas OLAP através de técnicas de processamento paralelo em nuvens [56, 66].

### 6.1 HBaseSQL

Popescu *et al.*[57] propõem uma arquitetura de *software* adaptável que é capaz de alternar entre MapReduce e SGBD paralelo para processar com eficiência as consultas OLAP. A alternância entre as duas arquiteturas é realizada de maneira transparente com base em um modelo de custo intuitivo, que calcula o tempo de execução esperado. Consultas rápidas são direcionadas ao SGBD paralelo, enquanto as mais demoradas são atendidas pelo MapReduce. Apesar dos resultados experimentais demonstrarem que a arquitetura adaptativa atinge ganhos no tempo de execução das consultas em vários cenários, a solução proposta não implementa um paralelismo intra-consulta. A seleção dos dados é aplicada nos diversos nós do *cluster*, porém os dados têm de ser trazidos para um nó *master* e, somente nele, são executadas as junções e agregações da consulta. Além disso, a implementação foi realizada por meio de um motor de execução específico do MySQL aliado ao gerenciador de armazenamento HBase, o que torna a solução difícil de ser portada para outros ambientes.

### 6.2 CloudTPS

Wei *et al.*[65] propõem uma camada de *middleware* que fica entre um aplicativo da Web e seu armazenamento de dados em nuvem, para fornecer uma plataforma escalável

e consistente de execução de consultas com junções. O sistema reforça a consistência de dados e é escalável linearmente de acordo com a carga de trabalho, que pode ser composta por consultas com junções e transações de leitura/gravação. A escalabilidade do CloudTPS se dá com base na premissa de que as consultas emitidas por aplicativos da Web (incluindo transações e consultas com junções) irão acessar, na maioria das vezes, um pequeno número de registro de dados em comparação com o tamanho total da base. Dessa forma, o CloudTPS tem seu foco em consultas e transações OLTP e não provê suporte à execução de consultas OLAP em paralelo.

## 6.3 MPI Cloud Platform

Zhu e Guo[68] propõem uma plataforma de nuvem baseada em MPI para executar operações SQL em bancos de dados distribuídos. A plataforma apoia um armazenamento maciço de dados e implementa um *cluster* distribuído de SGBDs MySQL. Cada nó do *cluster* é independente dos demais, possuindo seus próprios recursos de disco, memória e CPU. A arquitetura é representada através de uma hierarquia de nós de três níveis, sendo o primeiro nível composto pelo nó *master*; o segundo nível pelos nós principais; e o terceiro pelos nós escravos. Os nós principais contêm partições do banco de dados, de forma que a união de todas as partições represente o banco de dados completo. Os nós escravos possuem uma cópia do banco de dados particionado de seus respectivos nós principais, ou seja, ocorre a replicação de dados do segundo para o terceiro nível da hierarquia.

A *MPI Cloud Platform* possui um controlador lógico e semântico que realiza a tradução do SQL e determina quais nós irão executar a consulta. Os resultados são acessados pelo nó *master* através de RMA (*Remote Memory Access*) por meio de comandos MPI. São apoiados os comandos SQL *SELECT*, *INSERT*, *UPDATE* e *DELETE*. Apesar da semelhança entre a implementação da *MPI Cloud Platform* e C-ParGRES, existem diferenças no diz respeito a arquitetura e ao formato de comunicação entre os nós do *cluster*. Além disso, Zhu e Guo mencionam que existem limitações acerca da quantidade de registros retornados pelas consultas e que não há apoio a tipos de dados mais complexos. Também é possível observar algumas limitações acerca do uso de funções de agregação em conjunto com a cláusula *GROUP BY*, limitando a aplicação desta plataforma ao ambiente de consultas OLAP.

## 6.4 OLAP\*

Cuzzocrea e Moussa[11] propõem uma estrutura baseada em nuvem implementada sobre o Mondrian, um servidor OLAP de código aberto escrito em JAVA[22]. Combinando técnicas de particionamento de *data warehouse* com paradigmas de computação na nuvem, OLAP\* apoia consultas OLAP em ambientes de *Big Data*, sendo capaz de fornecer paralelismo inter- ou intra-consulta. OLAP\* realiza o roteamento de consultas SQL, determinando quais nós do *cluster* devem ser responsáveis pela execução da consulta, e o pós-processamento dos cubos de dados gerados por cada nó para realizar a agregação em uma única estrutura.

Para validar a implementação Cuzzocrea e Moussa utilizam uma variante do conhecido *benchmark* TPC-H, o TPC-H\*d. O TPC-H\*d foi desenvolvido especificamente para testar o desempenho de OLAP\*, enfatizando dados multidimensionais. Apesar dos autores ressaltarem o bom desempenho alcançado nos testes de OLAP\* com a base do TPC-H\*d, não foram apresentados dados concretos de análises de desempenhos, nem as consultas utilizadas. Os autores também não deixam claro quais os comandos SQL são suportados e quais são as restrições do *middleware* proposto.

Embora essas abordagens representem um avanço, somente o C-ParGRES, PowerDB e OLAP\* fornecem o paralelismo intra-consultas. Em um trabalho anterior[17], os autores mostram que o particionamento virtual dinâmico tem um desempenho melhor do que outras abordagens para consultas do *benchmark* TPC-H.

# Capítulo 7

## Conclusão

Este capítulo apresenta as conclusões desta dissertação, ressaltando as principais contribuições e limitações do trabalho. Também apresenta as perspectivas de trabalhos futuros que podem ser desenvolvidos como desdobramento dos resultados obtidos.

### 7.1 Contribuições

A abordagem proposta por esta dissertação é um novo *middleware* para processamento paralelo de consultas OLAP em nuvens de computador. O **C-ParGRES** estende o ParGRES, um *middleware* desenvolvido para *cluster* de banco de dados, com o objetivo fornecer processamento de alto desempenho através do paralelismo inter- e intra-consultas de forma transparente e sem comprometer a autonomia de aplicação. Além disso, o **C-ParGRES** é capaz de estimar a quantidade necessária de máquinas virtuais para implantar na nuvem para processamento de consultas. O **C-ParGRES** implementa a replicação completa do banco de dados em vários *sites* e *clusters* virtuais da nuvem, aumentando assim a disponibilidade de dados e a qualidade do serviço.

O **C-ParGRES** foi desenvolvido como um serviço (DBaaS) na nuvem da Amazon AWS. Avaliamos o desempenho do **C-ParGRES** utilizando a base de dados do Censo Demográfico 2010 - Dados da Amostra, pesquisa produzida pelo Instituto Brasileiro de Geografia e Estatística – IBGE. As consultas são provenientes de duas aplicações OLAP reais, ambas do IBGE: Banco Multidimensional de Estatísticas – BME e BME-TABULADOR. O ambiente computacional do experimento apresentou uma carga de trabalho mista, visto que as consultas da aplicação BME têm complexidade muito inferior às demandadas pelo BME-TABULADOR. A análise dos resultados mostrou que em situações com alto grau de concorrência, o **C-ParGRES** é uma opção viável para proporcionar a aceleração do

processamento das consultas, em especial para as consultas mais complexas e demoradas. Em todas as configurações testadas, os resultados mostram uma aceleração linear ou quase linear na maioria das execuções das consultas, à medida que mais máquinas virtuais são adicionadas ao *cluster* virtual. As consultas que não apresentaram aceleração linear já são muito rápidas, o que não compromete o tempo de processamento global dos grupos de consultas. Além disso, nenhum dos casos ficou pior que o seu respectivo tempo sequencial. Os resultados reforçam que o **C-ParGRES** se apresenta como uma solução atraente para o suporte a aplicativos OLAP em nuvens de computadores, em especial para atender à execução de consultas OLAP complexas.

## 7.2 Limitações

Podemos citar como limitações do nosso trabalho, o fato do **C-ParGRES** ter sido desenvolvido especificamente para a nuvem da Amazon AWS. A utilização em outro provedor de nuvem é possível, visto que todos apresentam lógica semelhante de funcionamento, porém implicaria a reprogramação de diversas classes que fazem o papel de intermediadoras com a API do serviço de nuvem. Uma solução mais genérica poderia envolver o uso de interfaces e classes abstratas para que as regras de negócio da aplicação se tornem independentes do serviço de nuvem escolhido, bastando a implementação de métodos específicos para cada provedor.

Outro ponto que deve ser destacado como uma limitação foi a utilização do módulo *Dimensioner* seguindo a abordagem proposta por Coutinho *et al.* [12]. Fizemos uma adaptação simples deste módulo, já que originalmente a implementação utiliza como parâmetros de entrada, para os requisitos de demanda de processamento, as informações extraídas do repositório de proveniência do SciCumulus. Nossos parâmetros de entrada foram os tempos de execução das consultas registrados nos arquivos *log*. Nosso algoritmo, no entanto, não leva em conta características específicas de consultas SQL, como por exemplo a quantidade de tuplas das tabelas envolvidas, número de junções realizadas, número de agregações definidas, o quão restritivos são os predicados da consulta, etc..

A replicação total da base também pode ser apontada como uma limitação do nosso trabalho, pois torna mais oneroso o processo de gerenciamento de atualizações e eleva os custos financeiros associados ao espaço em disco utilizado pelas máquinas virtuais. No entanto, como nos nossos experimentos a base era relativamente pequena e as atualizações não são frequentes (por se tratar de um banco OLAP), a solução de replicação total da



base pôde ser aplicada.

Por fim, temos uma limitação no analisador sintático proveniente do ParGRES, no que diz respeito ao não suporte de certos comandos da gramática SQL-99. No nosso caso particular, a cláusula *GROUPING SETS* é utilizada pelas consultas originais do sistema BME-TABULADOR e não é suportada pelo ParGRES. Foi necessária a reescrita das consultas deste sistema, bem como a criação de colunas auxiliares nas tabelas de fatos para contornar essa limitação. O ParGRES já possui suporte a cláusula *GROUP BY*, logo uma implementação que derivasse da lógica envolvida para o tratamento desta cláusula não seria tão complexa de ser feita e agregaria ainda mais funcionalidades ao *middleware*.

## 7.3 Trabalhos Futuros

O trabalho apresentado nesta dissertação pode ser estendido em várias direções promissoras. Algumas das perspectivas de trabalhos futuros são apresentadas a seguir:

- i) A primeira possibilidade de ampliação deste trabalho seria fornecer suporte para replicação parcial da base de dados [49], em oposição à replicação completa que foi implementada. O funcionamento do ParGRES com bases parcialmente replicadas é de fundamental importância para que torne-se viável trabalhar com bancos de dados muito grandes.
- ii) Outra direção promissora é o suporte à elasticidade do *cluster* virtual em caso de variações de desempenho das máquinas virtuais que o compõem. Essa implementação seria razoavelmente simples pois o ParGRES suporta adição de nós dinamicamente, após o serviço ter sido iniciado. Bastaria desenvolver uma estratégia de monitoramento do *cluster* virtual em termos de tempo de resposta das consultas e com informações de uso de CPU, disco, tráfego de rede, dentre outros indicadores.
- iii) A customização do Módulo *Dimensioner* para trabalhar com a análise de características das consultas (junções, agregações, predicados, etc.) e da base de dados (relacionamento entre tabelas, quantidades de linhas das tabelas de fatos, definições de índices, histórico de execuções, etc.) para proporcionar uma previsão da quantidade de máquinas virtuais mais precisa.
- iv) Podemos ainda citar como uma melhoria para a arquitetura do C-ParGRES, a possibilidade de trabalhar com mais de um *cluster* virtual em localidades de nuvem distintas

(*Multi-site*). Essa implementação pode ser realizada no Módulo de *Deployment* para que mais de um *cluster* virtual seja criado (cada um com suas configurações específicas) e a comunicação entre eles seja permitida.

- v) Por fim, testes com servidores *Bare Metal*, em oposição a servidores virtualizados, podem revelar resultados promissores quanto a aceleração de consultas. Em servidores *Bare Metal* o sistema operacional é instalado diretamente no hardware, proporcionando menor latência no acesso aos dados, pois elimina etapas de processamento via camadas de software. Para aplicações sensíveis a desempenho (como é o caso da maioria das aplicações OLAP), o *Bare Metal* pode trazer ganhos substanciais de desempenho em relação aos servidores virtualizados.

# Referências

- [1] AKAL, F., BÖHM, K., SCHEK, H.-J. Olap query evaluation in a database cluster: A performance study on intra-query parallelism. In *Proceedings of the 6th East European Conference on Advances in Databases and Information Systems* (London, UK, UK, 2002), ADBIS '02, Springer-Verlag, p. 218–231.
- [2] ALBIERI, S., BIANCHINI, Z. M. *Sigilo das informações individuais no Censo Demográfico 2010*. IBGE - Instituto Brasileiro de Geografia e Estatística, 9 2012. Disponível em [ftp://ftp.ibge.gov.br/Censos/Censo\\_Demografico\\_2010/metodologia/anexos/anexo\\_1\\_sigilo\\_cd2010.pdf](ftp://ftp.ibge.gov.br/Censos/Censo_Demografico_2010/metodologia/anexos/anexo_1_sigilo_cd2010.pdf).
- [3] ALCÂNTARA, J., OLIVEIRA, T., BESSANI, A. Ginja: One-dollar cloud-based disaster recovery for databases. In *Proceedings of the 18th ACM/IFIP/USENIX Middleware Conference* (New York, NY, USA, 2017), Middleware '17, ACM, p. 248–260.
- [4] AMAZON WEB SERVICES (AWS). Amazon Elastic Compute Cloud (Amazon EC2), 2018. Disponível em <https://aws.amazon.com/ec2>.
- [5] AMAZON WEB SERVICES (AWS). AWS SDK for Java Documentation, 2018. Disponível em <https://docs.aws.amazon.com/sdk-for-java/index.html#lang/en-us>.
- [6] AMAZON WEB SERVICES (AWS). Security Groups for Your VPC, 2018. Disponível em [https://docs.aws.amazon.com/vpc/latest/userguide/VPC\\_SecurityGroups.html](https://docs.aws.amazon.com/vpc/latest/userguide/VPC_SecurityGroups.html).
- [7] AMAZON WEB SERVICES (AWS). Serviços de computação em nuvem, 2018. Disponível em <https://aws.amazon.com>.
- [8] APACHE. Apache Maven, 2018. Disponível em <https://maven.apache.org/>.
- [9] BRASIL. Constituição da República Federativa do Brasil de 1988, 2018. Disponível em [http://www.planalto.gov.br/ccivil\\_03/constituicao/constituicaocompilado.htm](http://www.planalto.gov.br/ccivil_03/constituicao/constituicaocompilado.htm).
- [10] CAMPOS, M. P. Processamento de alto desempenho em consultas analíticas sobre base de dados geo-estatísticos. Dissertação de Mestrado, Universidade Federal do Rio de Janeiro, Rio de Janeiro, RJ, Brasil, Setembro 2008.
- [11] CUZZOCREA, A., MOUSSA, R. A cloud-based framework for supporting effective and efficient olap in big data environments. In *2014 14th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid)(CCGRID)* (May 2014), vol. 00, p. 680–684.

- [12] DE C. COUTINHO, R., DRUMMOND, L. M., FROTA, Y., DE OLIVEIRA, D. Optimizing virtual machine allocation for parallel scientific workflows in federated clouds. *Future Generation Computer Systems* 46 (2015), 51 – 68.
- [13] DE OLIVEIRA, D., BAIÃO, F. A., MATTOSO, M. *Towards a Taxonomy for Cloud Computing from an e-Science Perspective*. Springer London, London, 2010, p. 47–62.
- [14] DELNAT, W., TRUYEN, E., RAFIQUE, A., VAN LANDUYT, D., JOOSEN, W. K8-scalar: A workbench to compare autoscalers for container-orchestrated database clusters. In *Proceedings of the 13th International Conference on Software Engineering for Adaptive and Self-Managing Systems* (New York, NY, USA, 2018), SEAMS '18, ACM, p. 33–39.
- [15] EL-KHAMRA, Y., KIM, H., JHA, S., PARASHAR, M. Exploring the performance fluctuations of hpc workloads on clouds. In *2010 IEEE Second International Conference on Cloud Computing Technology and Science* (Nov 2010), p. 383–387.
- [16] FOSTER, I., KESSELMAN, C., Eds. *The Grid: Blueprint for a New Computing Infrastructure*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2004.
- [17] FURTADO, C., LIMA, A. A., PACITTI, E., VALDURIEZ, P., MATTOSO, M. Adaptive hybrid partitioning for olap query processing in a database cluster. *International journal of high performance computing and networking* 5, 4 (2008), 251–262.
- [18] FURTADO, C., LIMA, A. A. B., PACITTI, E., VALDURIEZ, P., MATTOSO, M. Adaptive hybrid partitioning for OLAP query processing in a database cluster. *IJHPCN* 5, 4 (2008), 251–262.
- [19] GRABS, T., BÖHM, K., SCHEK, H. Powerdb-ir - information retrieval on top of a database cluster. In *Proceedings of the 2001 ACM CIKM International Conference on Information and Knowledge Management, Atlanta, Georgia, USA, November 5-10, 2001* (2001), p. 411–418.
- [20] GRABS, T., BÖHM, K., SCHEK, H. Powerdb-ir - scalable information retrieval and storage with a cluster of databases. *Knowl. Inf. Syst.* 6, 4 (2004), 465–505.
- [21] HE, Q., ZHOU, S., KOBLER, B., DUFFY, D., MCGLYNN, T. Case study for running hpc applications in public clouds. In *Proceedings of the 19th ACM International Symposium on High Performance Distributed Computing* (New York, NY, USA, 2010), HPDC '10, ACM, p. 395–401.
- [22] HITACHI VANTARA COMMUNITY. Mondrian, 2018. Available at <https://github.com/pentaho/mondrian/>.
- [23] IBGE - INSTITUTO BRASILEIRO DE GEOGRAFIA E ESTATÍSTICA. Princípios Fundamentais das Estatísticas Oficiais, 1994. Disponível em [https://www.ibge.gov.br/institucional/documentos-ibge.html?option=com\\_content&view=article&id=16148](https://www.ibge.gov.br/institucional/documentos-ibge.html?option=com_content&view=article&id=16148).
- [24] IBGE - INSTITUTO BRASILEIRO DE GEOGRAFIA E ESTATÍSTICA. *Sinopse do Censo Demográfico 2010*. Instituto Brasileiro de Geografia e Estatística, 2011. Disponível em <https://biblioteca.ibge.gov.br/visualizacao/livros/liv49230.pdf>.

- [25] IBGE - INSTITUTO BRASILEIRO DE GEOGRAFIA E ESTATÍSTICA. *Resultados gerais da amostra do Censo Demográfico 2010*. Instituto Brasileiro de Geografia e Estatística, 2012. Disponível em [https://biblioteca.ibge.gov.br/visualizacao/periodicos/99/cd\\_2010\\_resultados\\_gerais\\_amostra.pdf](https://biblioteca.ibge.gov.br/visualizacao/periodicos/99/cd_2010_resultados_gerais_amostra.pdf).
- [26] IBGE - INSTITUTO BRASILEIRO DE GEOGRAFIA E ESTATÍSTICA. Tabela 200 - População residente, por sexo, situação e grupos de idade - Amostra - Características Gerais da População, 2012. Disponível em <https://sidra.ibge.gov.br/tabela/200>.
- [27] IBGE - INSTITUTO BRASILEIRO DE GEOGRAFIA E ESTATÍSTICA. Tabela 229 - Famílias residentes em domicílios particulares por tipo de família, situação do domicílio e número de componentes das famílias, 2012. Disponível em <https://sidra.ibge.gov.br/Tabela/229>.
- [28] IBGE - INSTITUTO BRASILEIRO DE GEOGRAFIA E ESTATÍSTICA. Tabela 3459 - Pessoas de 10 anos ou mais de idade, ocupadas na semana de referência, por religião, segundo o sexo, a posição na ocupação e a categoria do emprego no trabalho principal e as classes de rendimento nominal mensal de todos os trabalhos - Características Gerais da População, 2012. Disponível em <https://sidra.ibge.gov.br/Tabela/3459>.
- [29] IBGE - INSTITUTO BRASILEIRO DE GEOGRAFIA E ESTATÍSTICA. Tabela 3525 - Domicílios particulares permanentes com rendimento domiciliar, moradores em domicílios particulares permanentes com rendimento domiciliar e valor do rendimento nominal médio e mediano mensal dos domicílios particulares permanentes com rendimento domiciliar, por situação do domicílio, tipo de material das paredes externas, número de cômodos, número de dormitórios e existência de água canalizada e forma de abastecimento de água, 2012. Disponível em <https://sidra.ibge.gov.br/Tabela/3525>.
- [30] IBGE - INSTITUTO BRASILEIRO DE GEOGRAFIA E ESTATÍSTICA. Tabela 3534 - Domicílios particulares permanentes com rendimento domiciliar, moradores em domicílios particulares permanentes com rendimento domiciliar e valor do rendimento nominal médio e mediano mensal dos domicílios particulares permanentes com rendimento domiciliar, por situação do domicílio, número de cômodos e existência de alguns bens duráveis, 2012. Disponível em <https://sidra.ibge.gov.br/Tabela/3534>.
- [31] IBGE - INSTITUTO BRASILEIRO DE GEOGRAFIA E ESTATÍSTICA. *Metodologia do Censo Demográfico 2010*. Instituto Brasileiro de Geografia e Estatística, 2013. Disponível em <https://biblioteca.ibge.gov.br/visualizacao/livros/liv81634.pdf>.
- [32] IBGE - INSTITUTO BRASILEIRO DE GEOGRAFIA E ESTATÍSTICA, 2018. Disponível em <https://www.ibge.gov.br/>.
- [33] IBGE - INSTITUTO BRASILEIRO DE GEOGRAFIA E ESTATÍSTICA. BME - Banco Multidimensional de Estatísticas, 2018. Disponível em <https://www.bme.ibge.gov.br>.

- [34] IBGE - INSTITUTO BRASILEIRO DE GEOGRAFIA E ESTATÍSTICA. Censo Demográfico 2010: Documentação dos microdados da amostra, 2018. Disponível em [ftp://ftp.ibge.gov.br/Censos/Censo\\_Demografico\\_2010/Resultados\\_Gerais\\_da\\_Amostra/Microdados/Documentacao.zip](ftp://ftp.ibge.gov.br/Censos/Censo_Demografico_2010/Resultados_Gerais_da_Amostra/Microdados/Documentacao.zip).
- [35] IBGE - INSTITUTO BRASILEIRO DE GEOGRAFIA E ESTATÍSTICA. Censo Demográfico 2010: Microdados da amostra, 2018. Disponível em [https://ww2.ibge.gov.br/home/estatistica/populacao/censo2010/resultados\\_gerais\\_amostra/resultados\\_gerais\\_amostra\\_tab\\_uf\\_microdados.shtm](https://ww2.ibge.gov.br/home/estatistica/populacao/censo2010/resultados_gerais_amostra/resultados_gerais_amostra_tab_uf_microdados.shtm).
- [36] IBGE - INSTITUTO BRASILEIRO DE GEOGRAFIA E ESTATÍSTICA. O IBGE - Institucional, 2018. Disponível em <https://www.ibge.gov.br/institucional/o-ibge.html>.
- [37] IBGE - INSTITUTO BRASILEIRO DE GEOGRAFIA E ESTATÍSTICA. SIDRA - Sistema IBGE de Recuperação Automática, 2018. Disponível em <https://sidra.ibge.gov.br>.
- [38] IBGE - INSTITUTO BRASILEIRO DE GEOGRAFIA E ESTATÍSTICA. Tópicos da ajuda - SIDRA, 2018. Disponível em <https://sidra.ibge.gov.br/ajuda>.
- [39] JACKSON, K. R., RAMAKRISHNAN, L., MURIKI, K., CANON, S., CHOLIA, S., SHALF, J., WASSERMAN, H. J., WRIGHT, N. J. Performance analysis of high performance computing applications on the amazon web services cloud. In *2010 IEEE Second International Conference on Cloud Computing Technology and Science* (Nov 2010), p. 159–168.
- [40] KIEFER, T., LEHNER, W. Private table database virtualization for dbaas. In *2011 Fourth IEEE International Conference on Utility and Cloud Computing* (Dec 2011), p. 328–329.
- [41] KIMBALL, R., ROSS, M. *The data warehouse toolkit: the complete guide to dimensional modeling, 2nd Edition*. Wiley, 2002.
- [42] KOTOWSKI, N., LIMA, A. A., PACITTI, E., VALDURIEZ, P., MATTOSO, M. Parallel query processing for olap in grids. *Concurrency and Computation: Practice and Experience* 20, 17 (2008), 2039–2048.
- [43] LIMA, A. A., FURTADO, C., VALDURIEZ, P., MATTOSO, M. Parallel olap query processing in database clusters with data replication. *Distributed and Parallel Databases* 25, 1-2 (2009), 97–123.
- [44] LIMA, A. A., MATTOSO, M., VALDURIEZ, P. Adaptive virtual partitioning for olap query processing in a database cluster. *Journal of Information and Data Management* 1, 1 (2010), 75.
- [45] LIMA, A. A., MATTOSO, M., VALDURIEZ, P. Adaptive virtual partitioning: Further developments. *Journal of Information and Data Management* 1, 1 (2010), 89.
- [46] LIU, J., PACITTI, E., VALDURIEZ, P., DE OLIVEIRA, D., MATTOSO, M. Multi-objective scheduling of scientific workflows in multisite clouds. *Future Generation Comp. Syst.* 63 (2016), 76–95.

- [47] LIU, J., PACITTI, E., VALDURIEZ, P., MATTOSO, M. Scientific workflow scheduling with provenance data in a multisite cloud. *T. Large-Scale Data- and Knowledge-Centered Systems 33* (2017), 80–112.
- [48] MARTINS, S. L., PARDALOS, P. M., RESENDE, M. G. C., RIBEIRO, C. C. Greedy randomized adaptive search procedures for the steiner problem in graphs. In *Randomization Methods in Algorithm Design, Proceedings of a DIMACS Workshop, Princeton, New Jersey, USA, December 12-14, 1997* (1997), p. 133–146.
- [49] MATHIASON, G. *Virtual Full Replication for Scalable Distributed Real-Time Databases*. Tese de Doutorado, Linköping University, Sweden, 2009.
- [50] MATTOSO, M., ZIMBRÃO, G., LIMA, A. A., BAIÃO, F., BRAGANHOLO, V., AVELEDA, A., MIRANDA, B., ALMENTERO, B., COSTA, M. Pargres: uma camada de processamento paralelo de consultas sobre o postgresql. In *WSL2005-6 Workshop Software Livre* (2005), p. 259–264.
- [51] MATTOSO, M., ZIMBRÃO, G., LIMA, A. A., BAIÃO, F., BRAGANHOLO, V. P., AVELEDA, A., MIRANDA, B., ALMENTERO, B. K., COSTA, M. N. Pargres: a middleware for executing olap queries in parallel. *COPPE/UFRJ Technical Report ES-690*, <http://pargres.nacad.ufrj.br/Documentos/ES-690.pdf> (2005).
- [52] MISHIMA, T., FUJIWARA, Y. Madeus: Database live migration middleware under heavy workloads for cloud environment. In *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data* (New York, NY, USA, 2015), SIGMOD '15, ACM, p. 315–329.
- [53] ORACLE. OpenJDK, 2018. Disponível em <https://openjdk.java.net/install/>.
- [54] ÖZSU, M. T., VALDURIEZ, P. Distributed and parallel database systems. In *Computing Handbook, Third Edition: Information Systems and Information Technology*. 2014, p. 13: 1–24.
- [55] PAES, M., LIMA, A. A., VALDURIEZ, P., MATTOSO, M. High-performance query processing of a real-world olap database with pargres. In *International Conference on High Performance Computing for Computational Science* (2008), Springer, p. 188–200.
- [56] POPESCU, A., BALMIN, A., ERCEGOVAC, V., AILAMAKI, A. Predict: Towards predicting the runtime of large scale iterative analytics. *PVLDB 6*, 14 (2013), 1678–1689.
- [57] POPESCU, A. D., DASH, D., KANTERE, V., AILAMAKI, A. Adaptive query execution for data management in the cloud. In *Proceedings of the Second International Workshop on Cloud Data Management* (New York, NY, USA, 2010), CloudDB '10, ACM, p. 17–24.
- [58] PRESIDÊNCIA DA REPÚBLICA. LEI No 5.534, DE 14 DE NOVEMBRO DE 1968, 1968. Disponível em [http://www.planalto.gov.br/ccivil\\_03/LEIS/L5534.htm](http://www.planalto.gov.br/ccivil_03/LEIS/L5534.htm).
- [59] PRESIDÊNCIA DA REPÚBLICA. LEI Nº 8.184, DE 10 DE MAIO DE 1991, 1991. Disponível em [http://www.planalto.gov.br/ccivil\\_03/leis/l8184.htm](http://www.planalto.gov.br/ccivil_03/leis/l8184.htm).

- [60] SHATDAL, A., NAUGHTON, J. F. Adaptive parallel aggregation algorithms. In *Acm Sigmod Record* (1995), vol. 24, ACM, p. 104–114.
- [61] THE POSTGRESQL GLOBAL DEVELOPMENT GROUP. PostgreSQL: The World's Most Advanced Open Source Relational Database, 2018. Disponível em <https://www.postgresql.org/>.
- [62] UNSTATS - UNITED NATIONS STATISTICAL COMMISSION. Fundamental Principles of Official Statistics (A/RES/68/261 from 29 January 2014), 2014. Disponível em <https://unstats.un.org/unsd/dnss/gp/fundprinciples.aspx>.
- [63] VALDURIEZ, P. Parallel database systems: Open problems and new issues. *Distributed and Parallel Databases* 1, 2 (1993), 137–165.
- [64] VAQUERO, L., RODERO-MERINO, L., CACERES, J., LINDNER, M. A break in the clouds: towards a cloud definition. *SIGCOMM Comput. Commun. Rev.* 39, 1 (2009), 50–55.
- [65] WEI, Z., PIERRE, G., CHI, C.-H. Scalable join queries in cloud data stores. In *Proceedings of the 2012 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (Ccgrid 2012)* (Washington, DC, USA, 2012), CCGRID '12, IEEE Computer Society, p. 547–555.
- [66] WEISS, S., KULIKOWSKI, C. *Computer Systems That Learn: Classification and Prediction Methods from Statistics, Neural Nets, Machine Learning, and Expert Systems*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1991.
- [67] YASEEN, Q., PANDA, B. Tackling insider threat in cloud relational databases. In *2012 IEEE Fifth International Conference on Utility and Cloud Computing* (Nov 2012), p. 215–218.
- [68] ZHU, L., GUO, Y. A database middleware architecture for mpi-based cloud platform. In *2014 13th International Symposium on Distributed Computing and Applications to Business, Engineering and Science* (Nov 2014), p. 107–110.



## APÊNDICE A - TABELAS DE FATOS

Neste Apêndice são apresentados os atributos das tabelas de fatos CD2010ADOMI e CD2010APESS da base de dados CD2010A utilizada nos experimentos. Os atributos iniciados pelo prefixo “COD” são categorizados e possuem uma tabela de dimensão associada; os demais atributos são numéricos e, portanto, não possuem tabela de dimensão. Todos os atributos listados abaixo são do tipo numérico e têm suas descrições apresentadas sob o padrão de denominação adotado no BME.

Tabela A.1: Tabela CD2010ADOMI e seus atributos

ATRIBUTO	DIMENSÃO	DESCRIÇÃO
CODANOPESQ	T030	Ano da pesquisa do CD2010
CODPAIS	G000	País
CODREGEOGR	G031	Região Geográfica
CODUFCENSO	G032	Unidade da Federação
CODMESO	G233	Mesorregião (2010)
CODMICRO	G234	Microrregião (2010)
CODMUNICIPIO	G235	Município (2010)
CODDISTRITO	G236	Distrito (2010)
CODSUBDIST	G237	Subdistrito (2010)
CODREGMETRO	G239	Região metropolitana (2010)
CODAREAPOND	G242	Área de ponderação (2010)
CODAREADIVULGA	G246	Área de divulgação (2010) - Aglomerados Subnormais
CODSETOR	G238	Setor Censitário (2010)
V0300		CONTROLE
PESODOMI		Peso do domicílio

## (continuação da Tabela A.1 )

ATRIBUTO	DIMENSÃO	DESCRIÇÃO
CODV0201	CD275400	Domicílio, condição de ocupação
CODV0202	CD275500	Domicílio, material predominante paredes externas
V0203		Cômodos, número
CODV0203	CD276200	Cômodos, número, classe
V0204		Cômodos como dormitório, número
CODV0204	CD276400	Cômodos como dormitório, número, classe
CODV0205	CD276600	Banheiro de uso exclusivo, número, classe
CODV0206	CD276800	Sanitário ou buraco para dejeções, existência
CODV0207	CD276900	Esgotamento sanitário, tipo
CODV0208	CD277000	Abastecimento de água, forma
CODV0209	CD277200	Abastecimento de água, canalização
CODV0210	CD277300	Lixo, destino
CODV0211	CD275700	Energia elétrica, existência
CODV0212	CD275800	Energia elétrica, companhia distribuidora, medidor, existência
CODV0213	CD277400	Rádio, existência
CODV0214	CD278200	Televisão, existência
CODV0215	CD277600	Máquina de lavar roupa, existência
CODV0216	CD277500	Geladeira, existência
CODV0217	CD277700	Telefone celular, existência
CODV0218	CD277800	Telefone fixo, existência
CODV0219	CD278100	Microcomputador, existência
CODV0220	CD278000	Microcomputador com acesso à Internet, existência
CODV0221	CD277900	Motocicleta, para uso particular, existência
CODV0222	CD278300	Automóvel, uso particular, existência
CODV0301	CD279200	Morador, residência em outro país
V0401		Morador, número

## (continuação da Tabela A.1 )

ATRIBUTO	DIMENSÃO	DESCRIÇÃO
CODV0401	CD278500	Morador, número, classe
CODV0402	CD278600	Morador, responsável pelo domicílio particular
CODV0701	CD278700	Morador, falecimento, existência
CODV1005	CD274900	Setor, situação
CODV1006	CD275000	Domicílio, situação
V2011		Domicílio, aluguel mensal, reais
CODV2011	CD276000	Domicílio, aluguel mensal, reais, classe
V2012		Domicílio, aluguel mensal, salários mínimos
CODV2012	CD276100	Domicílio, aluguel mensal, salários mínimos, classe
CODV4001	CD275200	Domicílio, espécie
CODV4002	CD275300	Domicílio, tipo
V6203		Densidade morador/cômodo, número
CODV6203	CD276300	Densidade morador/cômodo, número, classe
V6204		Densidade morador/dormitório, número
CODV6204	CD276500	Densidade morador/dormitório, número, classe
V6529		Rendimento domiciliar
CODV6529	CD335600	Rendimento domiciliar, classe
V6530		Rendimento domiciliar, salários mínimos
CODV6530	CD335700	Rendimento domiciliar, salários mínimos, classe
V6531		Rendimento domiciliar per capita
CODV6531	CD335800	Rendimento domiciliar per capita, classe
V6532		Rendimento domiciliar per capita, salários mínimos
CODV6532	CD335900	Rendimento domiciliar per capita, salários mínimos, classe
CODM0201	CD336001	Imputação V0201, marca

## (continuação da Tabela A.1 )

ATRIBUTO	DIMENSÃO	DESCRIÇÃO
CODM0202	CD336003	Imputação V0202, marca
CODM0203	CD336004	Imputação V0203, marca
CODM0204	CD336005	Imputação V0204, marca
CODM0205	CD336006	Imputação V0205, marca
CODM0206	CD336007	Imputação V0206, marca
CODM0207	CD336008	Imputação V0207, marca
CODM0208	CD336009	Imputação V0208, marca
CODM0209	CD336010	Imputação V0209, marca
CODM0210	CD336011	Imputação V0210, marca
CODM0211	CD336012	Imputação V0211, marca
CODM0212	CD336013	Imputação V0212, marca
CODM0213	CD336014	Imputação V0213, marca
CODM0214	CD336015	Imputação V0214, marca
CODM0215	CD336016	Imputação V0215, marca
CODM0216	CD336017	Imputação V0216, marca
CODM0217	CD336018	Imputação V0217, marca
CODM0218	CD336019	Imputação V0218, marca
CODM0219	CD336020	Imputação V0219, marca
CODM0220	CD336021	Imputação V0220, marca
CODM0221	CD336022	Imputação V0221, marca
CODM0222	CD336023	Imputação V0222, marca
CODM0301	CD336024	Imputação V0301, marca
CODM0401	CD336025	Imputação V0401, marca
CODM0402	CD336026	Imputação V0402, marca
CODM0701	CD336027	Imputação V0701, marca
CODM2011	CD336002	Imputação V2011, marca

Tabela A.2: Tabela CD2010APESS e seus atributos

ATRIBUTO	DIMENSÃO	DESCRIÇÃO
CODANOPESQ	T030	Ano da pesquisa do CD2010
CODPAIS	G000	País
CODREGEAGR	G031	Região Geográfica
CODUFCENSO	G032	Unidade da Federação
CODMESO	G233	Mesorregião (2010)
CODMICRO	G234	Microrregião (2010)
CODMUNICIPIO	G235	Município (2010)
CODDISTRITO	G236	Distrito (2010)
CODSUBDIST	G237	Subdistrito (2010)
CODREGMETRO	G239	Região metropolitana (2010)
CODAREAPOND	G242	Área de ponderação (2010)
CODAREADIVULGA	G246	Área de divulgação (2010) - Aglomerados Subnormais
CODSETOR	G238	Setor Censitário (2010)
V0300		CONTROLE
V0504		Ordem lógica das pessoas no domicílio
PESOPESS		Pessoa, peso
CODV0502	CD279400	Condição no domicílio
CODV0601	CD279500	Sexo
CODV0604	CD280300	Mãe viva, existência
V0605		Mãe da pessoa, ordem
CODV0606	CD280400	Raça ou cor
CODV0613	CD281300	Registro de nascimento, existência
CODV0614	CD281400	Dificuldade de enxergar, existência
CODV0615	CD281500	Dificuldade de ouvir, existência
CODV0616	CD281600	Dificuldade de caminhar ou subir degraus, existência
CODV0617	CD281800	Deficiência mental ou intelectual, existência

(continuação da Tabela A.2 )

ATRIBUTO	DIMENSÃO	DESCRIÇÃO
CODV0618	CD282000	Natural do município
CODV0619	CD282100	Natural da UF
CODNACIONALIDADE	CD701342	Nacionalidade
CODV0620	CD282200	Nacionalidade
V0621		Natural de outro país, ano de chegada
CODV0621	CD282300	Natural de outro país, ano de chegada, classe
CODV0622	CD316100	Naturalidade, UF ou país de nascimento
V0623		Residência na UF, anos
CODV0623	CD282600	Residência na UF, anos, classe
V0624		Residência no município, anos
CODV0624	CD282700	Residência no município, anos, classe
CODV0625	CD316200	Residência anterior, UF, município ou país estrangeiro
CODV0626	CD316300	Residência em 31/07/2005, UF, município ou país estrangeiro
CODV0627	CD283400	Alfabetização
CODV0628	CD283500	Estudante, escola ou creche
CODV0629	CD283600	Estudante, curso
CODV0630	CD283800	Estudante, ensino fundamental, série ou ano
CODV0631	CD316400	Estudante, ensino médio, série
CODV0632	CD283900	Estudante, outro curso superior concluído
CODV0633	CD284000	Não estudante, curso mais elevado frequentado
CODV0634	CD284500	Não estudante, curso mais elevado concluído
CODV0635	CD284600	Não estudante, curso mais elevado concluído, espécie
CODV0636	CD284700	Estudante, escola ou creche, localidade
CODV0637	CD284800	Estado Conjugal
V0638		Cônjuge, ordem

(continuação da Tabela A.2 )

ATRIBUTO	DIMENSÃO	DESCRIÇÃO
CODV0639	CD285000	União, natureza
CODV0640	CD285200	Estado civil, tipo
CODV0641	CD285300	Trabalho remunerado, existência
CODV0642	CD285400	Trabalho remunerado, afastamento temporário
CODV0643	CD285500	Trabalho não remunerado, auxílio outro morador, existência
CODV0644	CD285600	Trabalho, próprio consumo, existência
CODV0645	CD286000	Trabalhos, número
CODV0648	CD286300	Trabalho principal, posição na ocupação
CODV0649	CD286400	Empregador, número de empregados
CODV0650	CD286500	Empregado, contribuição previdenciária
CODV0651	CD287200	Trabalho principal, rendimento bruto, tipo
CODV0652	CD287300	Demais trabalhos, rendimento bruto, tipo
V0653		Trabalho principal, horas semanais, número
CODV0653	CD286600	Trabalho principal, horas semanais, classe
CODV0654	CD285800	Providências para conseguir trabalho, existência
CODV0655	CD285900	Disponibilidade de trabalho, existência
CODV0656	CD288600	Aposentadoria ou pensão, existência
CODV0657	CD289000	Bolsa-família ou PETI, existência
CODV0658	CD289200	Programas sociais ou transferências, existência
CODV0659	CD289400	Outras fontes, existência
CODV0660	CD286700	Trabalho principal, localidade
CODV0661	CD286800	Trabalho principal, retorno para casa
CODV0662	CD287000	Trabalho principal, tempo de deslocamento
CODV0663	CD289600	Filhos nascidos vivos, existência
CODV0664	CD289700	Filhos vivos em 31/07/2010, existência

## (continuação da Tabela A.2 )

ATRIBUTO	DIMENSÃO	DESCRIÇÃO
CODV0665	CD290900	Último filho nascido vivo, sexo
CODV0667	CD291700	Último filho nascido vivo, vivo em 31/07/2010
CODV0668	CD317800	Último filho nascido vivo, falecimento, data, existência
CODV0669	CD289800	Filhos nascidos mortos, existência
CODV0670	CD289900	Informante, tipo
V0671		Informante, ordem
CODV5020	CD336148	Família, identificação
CODV5110	CD638635	Empregado, contribuição previdenciária oficial no trabalho principal
CODV5120	CD638636	Empregado, contribuição previdenciária oficial em qualquer trabalho
V5130		Pessoa, ordem lógica na família
V6033		Idade calculada, anos (I)
CODV6033	CD317900	Idade calculada, anos, classe (I)
V6036		Idade calculada, anos
CODV6036	CD549400	Idade calculada, anos, classe
V6037		Idade calculada, meses
CODV6037	CD317910	Idade calculada, meses, classe
CODV6040	CD280000	Idade, tipo
CODV6121	CD281210	Religião ou culto
CODV6121GR	CD336147	Religião, grupo
CODV6121SGR	CD336161	Religião, subgrupo
CODV6222	CD282400	Outra naturalidade, UF de nascimento
CODV6224	CD282500	Outra naturalidade, país de nascimento
CODV6252	CD282800	Residência anterior, UF
CODV6254	CD282900	Residência anterior, município
CODV6256	CD283000	Residência anterior, país estrangeiro



## (continuação da Tabela A.2 )

ATRIBUTO	DIMENSÃO	DESCRIÇÃO
CODV6262	CD283100	Residência em 31/07/2005, UF
CODV6264	CD283200	Residência em 31/07/2005, município
CODV6266	CD283300	Residência em 31/07/2005, país
CODV6352	CD316500	Não estudante, curso mais elevado concluído, superior de graduação, espécie
CODV6352GR	CD336163	Não estudante, curso mais elevado concluído, superior de graduação, grupo
CODV6352SGR	CD336162	Não estudante, curso mais elevado concluído, superior de graduação, subgrupo
CODV6354	CD316600	Não estudante, curso mais elevado concluído, mestrado, espécie
CODV6354GR	CD336165	Não estudante, curso mais elevado concluído, mestrado, grupo
CODV6354SGR	CD336164	Não estudante, curso mais elevado concluído, mestrado, subgrupo
CODV6356	CD316700	Não estudante, curso mais elevado concluído, doutorado, espécie
CODV6356GR	CD336167	Não estudante, curso mais elevado concluído, doutorado, grupo
CODV6356SGR	CD336166	Não estudante, curso mais elevado concluído, doutorado, subgrupo
CODV6362	CD316800	Estudante, escola ou creche, UF
CODV6364	CD316900	Estudante, escola ou creche, município
CODV6366	CD317000	Estudante, escola ou creche, país estrangeiro
CODV6400	CD317920	Nível de instrução
CODV6461	CD286100	Trabalho principal, ocupação (2010)
CODV6461GR	CD336152	Trabalho principal, ocupação, grupo
CODV6461GR2	CD286101	Trabalho principal, ocupação, grupo 2
CODV6461GR3	CD286102	Trabalho principal, ocupação, grupo 3
CODV6462	CD336159	Trabalho principal, ocupação (2000)

## (continuação da Tabela A.2 )

ATRIBUTO	DIMENSÃO	DESCRIÇÃO
CODV6471	CD286200	Trabalho principal, setor de atividade (2010)
CODV6471GR	CD336158	Trabalho principal, setor de atividade, grupo
CODV6472	CD336160	Trabalho principal, setor de atividade (2000)
V6511		Trabalho principal, rendimento bruto, reais
CODV6511	CD287400	Trabalho principal, rendimento bruto, reais, classe
V6513		Trabalho principal, rendimentos
CODV6513	CD317400	Trabalho principal, rendimentos, classe
V6514		Trabalho principal, rendimentos, salários mínimos
CODV6514	CD287500	Trabalho principal, rendimentos, salários mínimos, classe
V6521		Demais trabalhos, rendimento bruto, reais
CODV6521	CD287600	Demais trabalhos, rendimento bruto, reais, classe
V6524		Demais trabalhos, rendimentos, salários mínimos
CODV6524	CD287700	Demais trabalhos, rendimentos, salários mínimos, classe
V6525		Todos os trabalhos, rendimentos, reais
CODV6525	CD287800	Todos os trabalhos, rendimentos, reais, classe
V6526		Todos os trabalhos, rendimentos, salários mínimos
CODV6526	CD287900	Todos os trabalhos, rendimentos, salários mínimos, classe
V6527		Total de rendimentos brutos, reais
CODV6527	CD288000	Total de rendimentos brutos, reais, classe
V6528		Total de rendimentos brutos, salários mínimos

## (continuação da Tabela A.2 )

ATRIBUTO	DIMENSÃO	DESCRIÇÃO
CODV6528	CD288100	Total de rendimentos brutos, salários mínimos, classe
V6529		Rendimento domiciliar, reais
CODV6529	CD288200	Rendimento domiciliar, reais, classe
V6530		Rendimento domiciliar, salários mínimos
CODV6530	CD288300	Rendimento domiciliar, salários mínimos, classe
V6531		Rendimento domiciliar per capita
CODV6531	CD288400	Rendimento domiciliar per capita, classe
V6532		Rendimento domiciliar per capita, salários mínimos
CODV6532	CD288500	Rendimento domiciliar per capita, salários mínimos, classe
V6591		Outras fontes, rendimentos, julho de 2010, reais
CODV6591	CD317600	Outras fontes, rendimentos, julho de 2010, reais, classe
CODV6602	CD317100	Trabalho principal, UF
CODV6604	CD317200	Trabalho principal, município
CODV6606	CD317300	Trabalho principal, país estrangeiro
V6631		Filhos homens nascidos vivos, número
CODV6631	CD290600	Filhos homens nascidos vivos, número, classe
V6632		Filhas nascidas vivas, número
CODV6632	CD290300	Filhas nascidas vivas, número, classe
V6633		Filhos nascidos vivos, número
CODV6633	CD317930	Filhos nascidos vivos, número, classe
V6641		Filhos homens vivos em 31/07/2010, número
CODV6641	CD290700	Filhos homens vivos em 31/07/2010, número, classe
V6642		Filhas vivas em 31/07/2010, número

## (continuação da Tabela A.2 )

ATRIBUTO	DIMENSÃO	DESCRIÇÃO
CODV6642	CD290400	Filhas vivas em 31/07/2010, número, classe
V6643		Filhos vivos em 31/07/2010, número
CODV6643	CD317940	Filhos vivos, em 31/07/2010, número, classe
V6660		Último filho nascido vivo, idade calculada em anos
CODV6660	CD317950	Último filho nascido vivo, idade calculada em anos, classe
V6664		Último filho nascido vivo, 12 meses anteriores a 31/07/2010, número
CODV6664	CD290000	Último filho nascido vivo, 12 meses anteriores a 31/07/2010, existência
V6681		Último filho nascido vivo, falecimento, mês
CODV6681	CD291800	Último filho nascido vivo, falecimento, mês
V6682		Último filho nascido vivo, falecimento, ano
CODV6682	CD291900	Último filho nascido vivo, falecimento, ano
V6691		Filhos homens nascidos mortos em 31/07/10, número
CODV6691	CD290800	Filhos homens nascidos mortos em 31/07/10, número, classe
V6692		Filhas nascidas mortas em 31/07/10, número
CODV6692	CD290500	Filhas nascidas mortas em 31/07/10, número, classe
V6693		Filhos nascidos mortos em 31/07/10, total, número
CODV6693	CD290100	Filhos nascidos mortos em 31/07/10, total, número, classe
V6800		Filhos tidos, total, número
CODV6800	CD290200	Filhos tidos, total, número, classe
CODV6900	CD336149	Condição de atividade
CODV6910	CD336150	Condição de ocupação

## (continuação da Tabela A.2 )

ATRIBUTO	DIMENSÃO	DESCRIÇÃO
CODV6920	CD336151	Situação de ocupação
CODV6930	CD336153	Trabalho principal, posição na ocupação e categoria do emprego
CODV6940	CD336154	Trabalho principal, subgrupo e categoria do emprego
CODM0601	CD336029	Imputação V0601, marca
CODM0604	CD336030	Imputação V0604, marca
CODM0605	CD336031	Imputação V0605, marca
CODM0606	CD336032	Imputação V0606, marca
CODM0613	CD336036	Imputação V0613, marca
CODM0614	CD336037	Imputação V0614, marca
CODM0615	CD336038	Imputação V0615, marca
CODM0616	CD336039	Imputação V0616, marca
CODM0617	CD336040	Imputação V0617, marca
CODM0618	CD336041	Imputação V0618, marca
CODM0619	CD336042	Imputação V0619, marca
CODM0620	CD336043	Imputação V0620, marca
CODM0621	CD336044	Imputação V0621, marca
CODM0622	CD336045	Imputação V0622, marca
CODM0623	CD336046	Imputação V0623, marca
CODM0624	CD336047	Imputação V0624, marca
CODM0625	CD336048	Imputação V0625, marca
CODM0626	CD336049	Imputação V0626, marca
CODM0627	CD336050	Imputação V0627, marca
CODM0628	CD336051	Imputação V0628, marca
CODM0629	CD336052	Imputação V0629, marca
CODM0630	CD336053	Imputação V0630, marca
CODM0631	CD336054	Imputação V0631, marca
CODM0632	CD336055	Imputação V0632, marca

(continuação da Tabela A.2 )

ATRIBUTO	DIMENSÃO	DESCRIÇÃO
CODM0633	CD336056	Imputação V0633, marca
CODM0634	CD336057	Imputação V0634, marca
CODM0635	CD336058	Imputação V0635, marca
CODM0636	CD336059	Imputação V0636, marca
CODM0637	CD336060	Imputação V0637, marca
CODM0638	CD336061	Imputação V0638, marca
CODM0639	CD336062	Imputação V0639, marca
CODM0640	CD336063	Imputação V0640, marca
CODM0641	CD336064	Imputação V0641, marca
CODM0642	CD336065	Imputação V0642, marca
CODM0643	CD336066	Imputação V0643, marca
CODM0644	CD336067	Imputação V0644, marca
CODM0645	CD336068	Imputação V0645, marca
CODM0648	CD336069	Imputação V0648, marca
CODM0649	CD336070	Imputação V0649, marca
CODM0650	CD336071	Imputação V0650, marca
CODM0651	CD336072	Imputação V0651, marca
CODM0652	CD336073	Imputação V0652, marca
CODM0653	CD336074	Imputação V0653, marca
CODM0654	CD336075	Imputação V0654, marca
CODM0655	CD336076	Imputação V0655, marca
CODM0656	CD336077	Imputação V0656, marca
CODM0657	CD336078	Imputação V0657, marca
CODM0658	CD336079	Imputação V0658, marca
CODM0659	CD336080	Imputação V0659, marca
CODM0660	CD336081	Imputação V0660, marca
CODM0661	CD336082	Imputação V0661, marca
CODM0662	CD336083	Imputação V0662, marca
CODM0663	CD336084	Imputação V0663, marca

(continuação da Tabela A.2 )

ATRIBUTO	DIMENSÃO	DESCRIÇÃO
CODM0664	CD336085	Imputação V0664, marca
CODM0665	CD336086	Imputação V0665, marca
CODM0667	CD336087	Imputação V0667, marca
CODM0668	CD336088	Imputação V0668, marca
CODM0669	CD336089	Imputação V0669, marca
CODM0670	CD336090	Imputação V0670, marca
CODM0671	CD336091	Imputação V0671, marca
CODM6033	CD336092	Imputação V6033, marca
CODM6121	CD336096	Imputação V6121, marca
CODM6222	CD336097	Imputação V6222, marca
CODM6224	CD336098	Imputação V6224, marca
CODM6252	CD336099	Imputação V6252, marca
CODM6254	CD336100	Imputação V6254, marca
CODM6256	CD336101	Imputação V6256, marca
CODM6262	CD336102	Imputação V6262, marca
CODM6264	CD336103	Imputação V6264, marca
CODM6266	CD336104	Imputação V6266, marca
CODM6352	CD336105	Imputação V6352, marca
CODM6354	CD336106	Imputação V6354, marca
CODM6356	CD336107	Imputação V6356, marca
CODM6362	CD336108	Imputação V6362, marca
CODM6364	CD336109	Imputação V6364, marca
CODM6366	CD336110	Imputação V6366, marca
CODM6461	CD336111	Imputação V6461, marca
CODM6462	CD638637	Imputação V6462, marca
CODM6471	CD336112	Imputação V6471, marca
CODM6472	CD638638	Imputação V6472, marca
CODM6511	CD336113	Imputação V6511, marca
CODM6521	CD336114	Imputação V6521, marca

(continuação da Tabela A.2 )

ATRIBUTO	DIMENSÃO	DESCRIÇÃO
CODM6591	CD336115	Imputação V6591, marca
CODM6602	CD336116	Imputação V6602, marca
CODM6604	CD336117	Imputação V6604, marca
CODM6606	CD336118	Imputação V6606, marca
CODM6631	CD336119	Imputação V6631, marca
CODM6632	CD336120	Imputação V6632, marca
CODM6633	CD336121	Imputação V6633, marca
CODM6641	CD336122	Imputação V6641, marca
CODM6642	CD336123	Imputação V6642, marca
CODM6643	CD336124	Imputação V6643, marca
CODM6660	CD336125	Imputação V6660, marca
CODM6681	CD336126	Imputação V6681, marca
CODM6682	CD336127	Imputação V6682, marca
CODM6691	CD336128	Imputação V6691, marca
CODM6692	CD336129	Imputação V6692, marca
CODM6693	CD336130	Imputação V6693, marca
CODM6800	CD336131	Imputação V6800, marca



## APÊNDICE B - TABELAS DE FATOS

Neste Apêndice são apresentados os comandos SQL utilizados nos experimentos. Os comandos da aplicação BME são apresentados já com as modificações necessárias para execução em passo único pelo ParGRES. Já os comandos do BME-TABULADOR são apresentados com a estrutura original gerada pela aplicação, visto que a adaptação para execução pelo ParGRES gerou uma quantidade extremamente alta de subcomandos, inviabilizando sua exposição neste Apêndice. Contudo, a exposição dos comandos originais torna possível o entendimento da complexidade das consultas em questão.

### B.1 Consultas da aplicação BME

**Q01:**

```
SELECT 76 AS geografia_cod,
       9000 AS geografia_dim_cod,
       'Pais' AS geografia_dim_desc,
       g000.denominacao AS geografia_desc,
       cd2010adomi.codanopesq AS tempo_cod,
       8030 AS tempo_dim_cod,
       t030.denominacao AS tempo_desc,
       tab1.denominacao AS col1_den,
       tab1.ordem AS col1_ord,
       tab1.codigo AS col1_cod,
       SUM(cd2010adomi.pesodomi) AS frequencia,
       COUNT(*) AS contagem
FROM g000,
     t030,
     cd275000 tab1,
```

```
cd2010adomi
WHERE cd2010adomi.codanopesq = 2010
AND cd2010adomi.codv1006 = tab1.codigo
AND cd2010adomi.codanopesq = t030.codigo
GROUP BY g000.denominacao,
         cd2010adomi.codanopesq,
         t030.denominacao,
         tab1.denominacao,
         tab1.ordem,
         tab1.codigo
```

**Q02:**

```
SELECT 76 AS geografia_cod,
       9000 AS geografia_dim_cod,
       'Pais' AS geografia_dim_desc,
       g000.denominacao AS geografia_desc,
       cd2010apess.codanopesq AS tempo_cod,
       8030 AS tempo_dim_cod,
       t030.denominacao AS tempo_desc,
       tab1.denominacao AS col1_den,
       tab1.ordem AS col1_ord,
       tab1.codigo AS col1_cod,
       SUM(cd2010apess.pesopess) AS frequencia,
       COUNT(*) AS contagem
FROM g000,
     t030,
     cd549400 tab1,
     cd2010apess
WHERE cd2010apess.codanopesq = 2010
AND cd2010apess.codv6036 = tab1.codigo
AND cd2010apess.codanopesq = t030.codigo
GROUP BY g000.denominacao,
         cd2010apess.codanopesq,
```

```
t030.denominacao,  
tab1.denominacao,  
tab1.ordem,  
tab1.codigo
```

**Q03:**

```
SELECT 76 AS geografia_cod,  
       9000 AS geografia_dim_cod,  
       'Pais' AS geografia_dim_desc,  
       g000.denominacao AS geografia_desc,  
       cd2010apess.codanopesq AS tempo_cod,  
       8030 AS tempo_dim_cod,  
       t030.denominacao AS tempo_desc,  
       SUM(cd2010apess.v6036 * cd2010apess.pesopess) /  
         SUM(cd2010apess.pesopess) AS col1,  
       SUM(cd2010apess.pesopess) AS frequencia,  
       COUNT(*) AS contagem  
FROM g000,  
     t030,  
     cd2010apess  
WHERE cd2010apess.codanopesq = 2010  
      AND cd2010apess.codanopesq = t030.codigo  
GROUP BY g000.denominacao,  
         cd2010apess.codanopesq,  
         t030.denominacao
```

**Q04:**

```
SELECT 76 AS geografia_cod,  
       9000 AS geografia_dim_cod,  
       'Pais' AS geografia_dim_desc,  
       g000.denominacao AS geografia_desc,
```

```
        cd2010apess.codanopesq AS tempo_cod,
        8030 AS tempo_dim_cod,
        t030.denominacao AS tempo_desc,
        tab1.denominacao AS col1,
        tab1.ordem AS col1_ord,
        tab1.codigo AS col1_cod,
        SUM(cd2010apess.v6800 * cd2010apess.pesopess) AS col2,
        SUM(cd2010apess.pesopess) AS frequencia,
        COUNT(*) AS contagem
FROM g000,
     t030,
     cd290200 tab1,
     cd2010apess
WHERE cd2010apess.codanopesq = 2010
      AND cd2010apess.codv6800 = tab1.codigo
      AND cd2010apess.codanopesq = t030.codigo
GROUP BY g000.denominacao,
         cd2010apess.codanopesq,
         t030.denominacao,
         tab1.denominacao,
         tab1.ordem,
         tab1.codigo
```

**Q05:**

```
SELECT 76 AS geografia_cod,
        9000 AS geografia_dim_cod,
        'Pais' AS geografia_dim_desc,
        g000.denominacao AS geografia_desc,
        cd2010apess.codanopesq AS tempo_cod,
        8030 AS tempo_dim_cod,
        t030.denominacao AS tempo_desc,
        tab1.codigo AS col1,
        tab1.ordem AS col1_ord,
```

```
        tab1.codigo AS col1_cod,  
        SUM(cd2010apess.pesopess) AS frequencia,  
        COUNT(*) AS contagem  
FROM g000,  
     t030,  
     cd281210 tab1,  
     cd2010apess  
WHERE cd2010apess.codanopesq = 2010  
      AND cd2010apess.codv6121 = tab1.codigo  
      AND cd2010apess.codanopesq = t030.codigo  
      AND (cd2010apess.codv6121 = 750  
           OR cd2010apess.codv6121 = 810  
           OR cd2010apess.codv6121 = 0  
           OR cd2010apess.codv6121 = 110  
           OR cd2010apess.codv6121 = 240)  
GROUP BY g000.denominacao,  
         cd2010apess.codanopesq,  
         t030.denominacao,  
         tab1.denominacao,  
         tab1.ordem,  
         tab1.codigo
```

**Q06:**

```
SELECT 76 AS geografia_cod,  
       9000 AS geografia_dim_cod,  
       'Pais' AS geografia_dim_desc,  
       g000.denominacao AS geografia_desc,  
       cd2010apess.codanopesq AS tempo_cod,  
       8030 AS tempo_dim_cod,  
       t030.denominacao AS tempo_desc,  
       SUM(cd2010apess.v6036 * cd2010apess.pesopess) /  
         SUM(cd2010apess.pesopess) AS col1,  
       SUM(cd2010apess.pesopess) AS frequencia,
```

```
        COUNT(*) AS contagem
FROM g000,
     t030,
     cd2010apess
WHERE cd2010apess.codanopesq = 2010
      AND cd2010apess.codanopesq = t030.codigo
      AND (cd2010apess.v6036 BETWEEN 20 AND 40)
GROUP BY g000.denominacao,
         cd2010apess.codanopesq,
         t030.denominacao
```

**Q07:**

```
SELECT cd2010apess.codmunicipio AS geografia_cod,
       9235 AS geografia_dim_cod,
       'Municipio (2010)' AS geografia_dim_desc,
       g235.denominacao AS geografia_desc,
       cd2010apess.codanopesq AS tempo_cod,
       8030 AS tempo_dim_cod,
       t030.denominacao AS tempo_desc,
       tab1.denominacao AS col1,
       tab1.ordem AS col1_ord,
       tab1.codigo AS col1_cod,
       tab2.denominacao AS col2,
       tab2.ordem AS col2_ord,
       tab2.codigo AS col2_cod,
       SUM(cd2010apess.v6036 * cd2010apess.pesopess) /
       SUM(cd2010apess.pesopess) AS col3,
       SUM(cd2010apess.pesopess) AS frequencia,
       COUNT(*) AS contagem
FROM g235,
     t030,
     cd279500 tab1,
     cd284000 tab2,
```

```
cd2010apess
WHERE (cd2010apess.codmunicipio = 3304557)
AND cd2010apess.codanopesq = 2010
AND cd2010apess.codv0601 = tab1.codigo
AND cd2010apess.codv0633 = tab2.codigo
AND cd2010apess.codanopesq = t030.codigo
AND cd2010apess.codmunicipio = g235.codigo
GROUP BY cd2010apess.codmunicipio,
         g235.denominacao,
         cd2010apess.codanopesq,
         t030.denominacao,
         tab1.denominacao,
         tab1.ordem,
         tab1.codigo,
         tab2.denominacao,
         tab2.ordem,
         tab2.codigo
```

**Q08:**

```
SELECT cd2010apess.codmunicipio AS geografia_cod,
       9235 AS geografia_dim_cod,
       'Município (2010)' AS geografia_dim_desc,
       g235.denominacao AS geografia_desc,
       cd2010apess.codanopesq AS tempo_cod,
       8030 AS tempo_dim_cod,
       t030.denominacao AS tempo_desc,
       tab1.denominacao AS col1,
       tab1.ordem AS col1_ord,
       tab1.codigo AS col1_cod,
       tab2.denominacao AS col2,
       tab2.ordem AS col2_ord,
       tab2.codigo AS col2_cod,
       SUM(cd2010apess.v6036 * cd2010apess.pesopess) /
```

```
SUM(cd2010apess.pesopess) AS col3,
SUM(cd2010apess.pesopess) AS frequencia,
COUNT(*) AS contagem
FROM g235,
t030,
cd279500 tab1,
cd284000 tab2,
cd2010apess
WHERE (cd2010apess.codmunicipio = 2408102
      OR cd2010apess.codmunicipio = 3304557)
AND cd2010apess.codanopesq = 2010
AND cd2010apess.codv0601 = tab1.codigo
AND cd2010apess.codv0633 = tab2.codigo
AND cd2010apess.codanopesq = t030.codigo
AND cd2010apess.codmunicipio = g235.codigo
GROUP BY cd2010apess.codmunicipio,
g235.denominacao,
cd2010apess.codanopesq,
t030.denominacao,
tab1.denominacao,
tab1.ordem,
tab1.codigo,
tab2.denominacao,
tab2.ordem,
tab2.codigo
```

**Q09:**

```
SELECT 76 AS geografia_cod,
9000 AS geografia_dim_cod,
'Pais' AS geografia_dim_desc,
g000.denominacao AS geografia_desc,
cd2010adomi.codanopesq AS tempo_cod,
8030 AS tempo_dim_cod,
```



```
        t030.denominacao AS tempo_desc,
        tab1.denominacao AS col1,
        tab1.ordem AS col1_ord,
        tab1.codigo AS col1_cod,
        tab2.denominacao AS col2,
        tab2.ordem AS col2_ord,
        tab2.codigo AS col2_cod,
        SUM(cd2010apess.pesopess) AS frequencia,
        COUNT(*) AS contagem
FROM g000,
     t030,
     cd279500 tab1,
     cd275000 tab2,
     cd2010apess,
     cd2010adomi
WHERE (cd2010apess.v0300 = cd2010adomi.v0300
      AND cd2010apess.codmunicipio = cd2010adomi.codmunicipio)
      AND cd2010adomi.codanopesq = 2010
      AND cd2010apess.codv0601 = tab1.codigo
      AND cd2010adomi.codv1006 = tab2.codigo
      AND cd2010adomi.codanopesq = t030.codigo
GROUP BY g000.denominacao,
         cd2010adomi.codanopesq,
         t030.denominacao,
         tab1.denominacao,
         tab1.ordem,
         tab1.codigo,
         tab2.denominacao,
         tab2.ordem,
         tab2.codigo
```

**Q10:**

```
SELECT cd2010adomi.codmunicipio AS geografia_cod,
```

```
9235 AS geografia_dim_cod,
'Municipio (2010)' AS geografia_dim_desc,
g235.denominacao AS geografia_desc,
cd2010adomi.codanopesq AS tempo_cod,
8030 AS tempo_dim_cod,
t030.denominacao AS tempo_desc,
tab1.denominacao AS col1,
tab1.ordem AS col1_ord,
tab1.codigo AS col1_cod,
tab2.denominacao AS col2,
tab2.ordem AS col2_ord,
tab2.codigo AS col2_cod,
tab3.denominacao AS col3,
tab3.ordem AS col3_ord,
tab3.codigo AS col3_cod,
SUM(cd2010apess.v6036 * cd2010apess.pesopess) /
SUM(cd2010apess.pesopess) AS col4,
SUM(cd2010apess.pesopess) AS frequencia,
COUNT(*) AS contagem
FROM g235,
t030,
cd279500 tab1,
cd275000 tab2,
cd284000 tab3,
cd2010apess,
cd2010adomi
WHERE (cd2010apess.v0300 = cd2010adomi.v0300
AND cd2010apess.codmunicipio = cd2010adomi.codmunicipio)
AND (cd2010apess.codmunicipio = 3304557)
AND (cd2010adomi.codmunicipio = 3304557)
AND cd2010adomi.codanopesq = 2010
AND cd2010apess.codv0601 = tab1.codigo
AND cd2010adomi.codv1006 = tab2.codigo
AND cd2010apess.codv0633 = tab3.codigo
AND cd2010adomi.codanopesq = t030.codigo
```

```
AND cd2010adomi.codmunicipio = g235.codigo
GROUP BY cd2010adomi.codmunicipio,
         g235.denominacao,
         cd2010adomi.codanopesq,
         t030.denominacao,
         tab1.denominacao,
         tab1.ordem,
         tab1.codigo,
         tab2.denominacao,
         tab2.ordem,
         tab2.codigo,
         tab3.denominacao,
         tab3.ordem,
         tab3.codigo
```

**Q11:**

```
SELECT cd2010adomi.codmunicipio AS geografia_cod,
       9235 AS geografia_dim_cod,
       'Município (2010)' AS geografia_dim_desc,
       g235.denominacao AS geografia_desc,
       cd2010adomi.codanopesq AS tempo_cod,
       8030 AS tempo_dim_cod,
       t030.denominacao AS tempo_desc,
       tab1.denominacao AS col1,
       tab1.ordem AS col1_ord,
       tab1.codigo AS col1_cod,
       tab2.denominacao AS col2,
       tab2.ordem AS col2_ord,
       tab2.codigo AS col2_cod,
       tab3.denominacao AS col3,
       tab3.ordem AS col3_ord,
       tab3.codigo AS col3_cod,
       SUM(cd2010apess.v6036 * cd2010apess.pesopess) /
```

```
SUM(cd2010apess.pesopess) AS col4,
SUM(cd2010apess.pesopess) AS frequencia,
COUNT(*) AS contagem
FROM g235,
t030,
cd279500 tab1,
cd275000 tab2,
cd284000 tab3,
cd2010apess,
cd2010adomi
WHERE (cd2010apess.v0300 = cd2010adomi.v0300
      AND cd2010apess.codmunicipio = cd2010adomi.codmunicipio)
AND (cd2010apess.codmunicipio = 2408102
     OR cd2010apess.codmunicipio = 3304557)
AND (cd2010adomi.codmunicipio = 2408102
     OR cd2010adomi.codmunicipio = 3304557)
AND cd2010adomi.codanopesq = 2010
AND cd2010apess.codv0601 = tab1.codigo
AND cd2010adomi.codv1006 = tab2.codigo
AND cd2010apess.codv0633 = tab3.codigo
AND cd2010adomi.codanopesq = t030.codigo
AND cd2010adomi.codmunicipio = g235.codigo
GROUP BY cd2010adomi.codmunicipio,
g235.denominacao,
cd2010adomi.codanopesq,
t030.denominacao,
tab1.denominacao,
tab1.ordem,
tab1.codigo,
tab2.denominacao,
tab2.ordem,
tab2.codigo,
tab3.denominacao,
tab3.ordem,
tab3.codigo
```

**Q12:**

```
SELECT cd2010adomi.codufcenso AS geografia_cod,
       9032 AS geografia_dim_cod,
       'Unidade da Federacao' AS geografia_dim_desc,
       g032.denominacao AS geografia_desc,
       cd2010adomi.codanopesq AS tempo_cod,
       8030 AS tempo_dim_cod,
       t030.denominacao AS tempo_desc,
       tab1.denominacao AS col1,
       tab1.ordem AS col1_ord,
       tab1.codigo AS col1_cod,
       SUM(cd2010apess.v6036 * cd2010apess.pesopess) /
       SUM(cd2010apess.pesopess) AS col2,
       SUM(cd2010adomi.v6531 * cd2010adomi.pesodomi) /
       SUM(cd2010adomi.pesodomi) AS col3,
       SUM(cd2010apess.pesopess) AS frequencia,
       COUNT(*) AS contagem
FROM g032,
     t030,
     cd275200 tab1,
     cd2010apess,
     cd2010adomi
WHERE (cd2010apess.v0300 = cd2010adomi.v0300
      AND cd2010apess.codmunicipio = cd2010adomi.codmunicipio)
AND g032.ind_exibicao = 'S'
AND cd2010adomi.codanopesq = 2010
AND (cd2010adomi.codv4001 = 1
     OR cd2010adomi.codv4001 = 2
     OR cd2010adomi.codv4001 = 3
     OR cd2010adomi.codv4001 = 4)
AND (cd2010adomi.v6531 <= 140)
AND cd2010adomi.codv4001 = tab1.codigo
```

```
AND cd2010adomi.codanopesq = t030.codigo
AND cd2010adomi.codufcenso = g032.codigo
GROUP BY cd2010adomi.codufcenso,
         g032.denominacao,
         cd2010adomi.codanopesq,
         t030.denominacao,
         tab1.denominacao,
         tab1.ordem,
         tab1.codigo
```

**Q13:**

```
SELECT cd2010adomi.codufcenso AS geografia_cod,
       9032 AS geografia_dim_cod,
       'Unidade da Federacao' AS geografia_dim_desc,
       g032.denominacao AS geografia_desc,
       cd2010adomi.codanopesq AS tempo_cod,
       8030 AS tempo_dim_cod,
       t030.denominacao AS tempo_desc,
       tab1.denominacao AS col1,
       tab1.ordem AS col1_ord,
       tab1.codigo AS col1_cod,
       tab2.denominacao AS col2,
       tab2.ordem AS col2_ord,
       tab2.codigo AS col2_cod,
       SUM(cd2010apess.pesopess) AS frequencia,
       COUNT(*) AS contagem
FROM g032,
     t030,
     cd283400 tab1,
     cd275000 tab2,
     cd2010apess,
     cd2010adomi
WHERE (cd2010apess.v0300 = cd2010adomi.v0300
```

```
        AND cd2010apess.codmunicipio = cd2010adomi.codmunicipio)
AND g032.ind_exibicao = 'S'
AND (cd2010apess.codv0627 = 1
     OR cd2010apess.codv0627 = 2)
AND cd2010adomi.codanopesq = 2010
AND cd2010apess.codv0627 = tab1.codigo
AND cd2010adomi.codv1006 = tab2.codigo
AND cd2010adomi.codanopesq = t030.codigo
AND cd2010adomi.codufcenso = g032.codigo
GROUP BY cd2010adomi.codufcenso,
         g032.denominacao,
         cd2010adomi.codanopesq,
         t030.denominacao,
         tab1.denominacao,
         tab1.ordem,
         tab1.codigo,
         tab2.denominacao,
         tab2.ordem,
         tab2.codigo
```

**Q14:**

```
SELECT cd2010adomi.codufcenso AS geografia_cod,
       9032 AS geografia_dim_cod,
       'Unidade da Federacao' AS geografia_dim_desc,
       g032.denominacao AS geografia_desc,
       cd2010adomi.codanopesq AS tempo_cod,
       8030 AS tempo_dim_cod,
       t030.denominacao AS tempo_desc,
       tab1.denominacao AS col1,
       tab1.ordem AS col1_ord,
       tab1.codigo AS col1_cod,
       tab2.denominacao AS col2,
       tab2.ordem AS col2_ord,
```

```
        tab2.codigo AS col2_cod,
        tab3.denominacao AS col3,
        tab3.ordem AS col3_ord,
        tab3.codigo AS col3_cod,
        tab4.denominacao AS col4,
        tab4.ordem AS col4_ord,
        tab4.codigo AS col4_cod,
        SUM(cd2010apess.pesopess) AS frequencia,
        COUNT(*) AS contagem
FROM g032,
     t030,
     cd281800 tab1,
     cd281500 tab2,
     cd275000 tab3,
     cd275300 tab4,
     cd2010apess,
     cd2010adomi
WHERE (cd2010apess.v0300 = cd2010adomi.v0300
      AND cd2010apess.codmunicipio = cd2010adomi.codmunicipio)
AND g032.ind_exibicao = 'S'
AND cd2010adomi.codanopesq = 2010
AND ((cd2010apess.codv0617 = 1)
     OR (cd2010apess.codv0615 = 1
         OR cd2010apess.codv0615 = 2
         OR cd2010apess.codv0615 = 3))
AND (cd2010adomi.codv4002 = 11
     OR cd2010adomi.codv4002 = 12
     OR cd2010adomi.codv4002 = 13
     OR cd2010adomi.codv4002 = 14
     OR cd2010adomi.codv4002 = 15
     OR cd2010adomi.codv4002 = 51
     OR cd2010adomi.codv4002 = 52
     OR cd2010adomi.codv4002 = 53
     OR cd2010adomi.codv4002 = 61
     OR cd2010adomi.codv4002 = 62
```



```
OR cd2010adomi.codv4002 = 63
OR cd2010adomi.codv4002 = 64
OR cd2010adomi.codv4002 = 65
OR cd2010adomi.codv4002 = 71
OR cd2010adomi.codv4002 = 72
OR cd2010adomi.codv4002 = 73
OR cd2010adomi.codv4002 = 74)
AND cd2010apess.codv0617 = tab1.codigo
AND cd2010apess.codv0615 = tab2.codigo
AND cd2010adomi.codv1006 = tab3.codigo
AND cd2010adomi.codv4002 = tab4.codigo
AND cd2010adomi.codanopesq = t030.codigo
AND cd2010adomi.codufcenso = g032.codigo
GROUP BY cd2010adomi.codufcenso,
        g032.denominacao,
        cd2010adomi.codanopesq,
        t030.denominacao,
        tab1.denominacao,
        tab1.ordem,
        tab1.codigo,
        tab2.denominacao,
        tab2.ordem,
        tab2.codigo,
        tab3.denominacao,
        tab3.ordem,
        tab3.codigo,
        tab4.denominacao,
        tab4.ordem,
        tab4.codigo
```

## B.2 Consultas da aplicação BME-TABULADOR

**T3459:**

```
SELECT
  A.INDICADORA_GEOG1,
  A.INDICADORA_A00,
  A.INDICADORA_A01,
  A.INDICADORA_B00,
  A.INDICADORA_C00,
  A.INDICADORA_C01,
  A.INDICADORA_D00,
  SUM(A.PESOPESS) AS V916
FROM (SELECT
  CASE
    WHEN CD2010APESS.CODV6121 IN (110, 111, 112) THEN 1
    WHEN CD2010APESS.CODV6121 IN (
      210, 219, 220, 221, 222, 223, 224, 225,
      229, 230, 231, 232, 239, 240, 241, 242,
      243, 244, 245, 249, 250, 251, 259, 260,
      261, 262, 269, 270, 279, 280, 289, 300,
      310, 311, 312, 319, 320, 329, 330, 339,
      340, 349, 350, 359, 360, 369, 370, 379,
      380, 389, 390, 399, 400, 401, 409, 420,
      429, 430, 439, 440, 449, 450, 451, 452,
      453, 454, 455, 460, 469, 470, 479, 480,
      489, 490, 492, 499
    ) THEN 2
    WHEN CD2010APESS.CODV6121 IN (610, 619) THEN 6
    WHEN CD2010APESS.CODV6121 IN (620, 629, 630, 639, 640, 641, 649)
      THEN 7
    WHEN CD2010APESS.CODV6121 IN (
      120, 130, 140, 149, 199, 510, 519, 520,
      529, 530, 590, 599, 710, 711, 719, 740,
      741, 749, 750, 751, 752, 753, 754, 755,
      759, 760, 761, 762, 763, 764, 765, 766,
```

```
790, 791, 792, 793, 799, 810, 811, 819,
820, 821, 829, 830, 831, 832, 833, 834,
839, 850, 890, 891, 892, 893, 894, 895,
896
) THEN 8
WHEN CD2010APESS.CODV6121 IN (0, 1, 2) THEN 9
WHEN CD2010APESS.CODV6121 = 990 THEN 10
WHEN CD2010APESS.CODV6121 = 999 THEN 11
ELSE -1
END AS INDICADORA_A00,
CASE
WHEN CD2010APESS.CODV6121 IN (
210, 219, 220, 221, 222, 223, 224, 225,
229, 230, 231, 232, 239, 240, 241, 242,
243, 244, 245, 249, 250, 251, 259, 260,
261, 262, 269, 270, 279, 280, 289
) THEN 3
WHEN CD2010APESS.CODV6121 IN (
300, 310, 311, 312, 319, 320, 329, 330,
339, 340, 349, 350, 359, 360, 369, 370,
379, 380, 389, 390, 399, 400, 401, 409,
420, 429, 430, 439, 440, 449, 450, 451,
452, 453, 454, 455, 460, 469, 470, 479,
480, 489
) THEN 4
WHEN CD2010APESS.CODV6121 IN (490, 492, 499) THEN 5
ELSE -1
END AS INDICADORA_A01,
CASE
WHEN CD2010APESS.CODV0601 = 1 THEN 1
WHEN CD2010APESS.CODV0601 = 2 THEN 2
ELSE -1
END AS INDICADORA_B00,
CASE
WHEN CD2010APESS.CODV0648 IN (1, 2, 3, 4) THEN 1
```

```
WHEN CD2010APESS.CODV0648 = 7 THEN 7
WHEN CD2010APESS.CODV0644 = 1 THEN 8
WHEN CD2010APESS.CODV0648 = 6 THEN 6
WHEN CD2010APESS.CODV0648 = 5 THEN 5
ELSE -1
END AS INDICADORA_C00,
CASE
WHEN CD2010APESS.CODV0648 = 1 THEN 2
WHEN CD2010APESS.CODV0648 IN (2, 3) THEN 3
WHEN CD2010APESS.CODV0648 = 4 THEN 4
ELSE -1
END AS INDICADORA_C01,
CASE
WHEN CD2010APESS.V6525 > 0 AND
    CD2010APESS.V6525 <= 255 THEN 1
WHEN CD2010APESS.V6525 > 255 AND
    CD2010APESS.V6525 <= 510 THEN 2
WHEN CD2010APESS.V6525 > 510 AND
    CD2010APESS.V6525 <= 1020 THEN 3
WHEN CD2010APESS.V6525 > 1020 AND
    CD2010APESS.V6525 <= 1530 THEN 11
WHEN CD2010APESS.V6525 > 1530 AND
    CD2010APESS.V6525 <= 2550 THEN 4
WHEN CD2010APESS.V6525 > 2550 AND
    CD2010APESS.V6525 <= 5100 THEN 5
WHEN CD2010APESS.V6525 > 5100 AND
    CD2010APESS.V6525 <= 7650 THEN 6
WHEN CD2010APESS.V6525 > 7650 AND
    CD2010APESS.V6525 <= 10200 THEN 7
WHEN CD2010APESS.V6525 > 10200 AND
    CD2010APESS.V6525 <= 15300 THEN 8
WHEN CD2010APESS.V6525 > 15300 THEN 9
WHEN CD2010APESS.V6525 = 0 THEN 10
ELSE -1
END AS INDICADORA_D00,
```

```
CD2010APESS.V6036,
CD2010APESS.CODV6910,
CD2010APESS.PESOPESS,
CD2010APESS.CODPAIS AS INDICADORA_GEOG1,
CD2010APESS.CODANOPESQ AS INDICADORA_TEMPORAL
FROM IBGE.CD2010APESS
WHERE CD2010APESS.CODANOPESQ = 2010
AND (
CD2010APESS.V6036 >= 10
AND CD2010APESS.CODV6910 = 1
)) A
GROUP BY A.INDICADORA_GEOG1,
        GROUPING SETS (
        (
        A.INDICADORA_A00, A.INDICADORA_B00,
        A.INDICADORA_C00, A.INDICADORA_D00
        ),
        (
        A.INDICADORA_A00, A.INDICADORA_B00,
        A.INDICADORA_C01, A.INDICADORA_D00
        ),
        (
        A.INDICADORA_A01, A.INDICADORA_B00,
        A.INDICADORA_C00, A.INDICADORA_D00
        ),
        (
        A.INDICADORA_A01, A.INDICADORA_B00,
        A.INDICADORA_C01, A.INDICADORA_D00
        )
        )
HAVING (
(
NVL(A.INDICADORA_A00, -1) <> -1
OR NVL(A.INDICADORA_A01, -1) <> -1
)
```

```
AND (
NVL(A.INDICADORA_B00, -1) <> -1
)
AND (
NVL(A.INDICADORA_C00, -1) <> -1
OR NVL(A.INDICADORA_C01, -1) <> -1
)
AND (
NVL(A.INDICADORA_D00, -1) <> -1
)
)
```

**T3525:**

```
SELECT
  A.INDICADORA_GEOG1,
  A.INDICADORA_A00,
  A.INDICADORA_A01,
  A.INDICADORA_B00,
  A.INDICADORA_B01,
  A.INDICADORA_C00,
  A.INDICADORA_C01,
  A.INDICADORA_D00,
  A.INDICADORA_D01,
  A.INDICADORA_E00,
  A.INDICADORA_E01,
  A.INDICADORA_E02,
  SUM(PESODOMI) AS V884,
  SUM(V6529 * A.PESODOMI) / SUM(PESODOMI) AS V878,
  SUM(V0401 * A.PESODOMI) AS V1883
FROM (SELECT
  CASE
    WHEN CD2010ADOMI.CODV1006 IN (-1, 1, 2) THEN 9999
    ELSE -1
```

```
END AS INDICADORA_A00,
CASE
  WHEN CD2010ADOMI.CODV1006 = 1 THEN 1
  WHEN CD2010ADOMI.CODV1006 = 2 THEN 2
  ELSE -1
END AS INDICADORA_A01,
CASE
  WHEN CD2010ADOMI.CODV0202 IN (-1, 1, 2, 3, 4, 5, 6, 7, 8, 9)
    THEN 9999
  ELSE -1
END AS INDICADORA_B00,
CASE
  WHEN CD2010ADOMI.CODV0202 = 1 THEN 1
  WHEN CD2010ADOMI.CODV0202 = 2 THEN 2
  WHEN CD2010ADOMI.CODV0202 = 3 THEN 3
  WHEN CD2010ADOMI.CODV0202 = 4 THEN 4
  WHEN CD2010ADOMI.CODV0202 = 5 THEN 5
  WHEN CD2010ADOMI.CODV0202 = 6 THEN 6
  WHEN CD2010ADOMI.CODV0202 = 7 THEN 7
  WHEN CD2010ADOMI.CODV0202 = 8 THEN 8
  WHEN CD2010ADOMI.CODV0202 = 9 THEN 9
  ELSE -1
END AS INDICADORA_B01,
CASE
  WHEN CD2010ADOMI.CODV0203 IN (-1, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11)
    THEN 9999
  ELSE -1
END AS INDICADORA_C00,
CASE
  WHEN CD2010ADOMI.CODV0203 = 1 THEN 1
  WHEN CD2010ADOMI.CODV0203 = 2 THEN 2
  WHEN CD2010ADOMI.CODV0203 = 3 THEN 3
  WHEN CD2010ADOMI.CODV0203 = 4 THEN 4
  WHEN CD2010ADOMI.CODV0203 = 5 THEN 5
  WHEN CD2010ADOMI.CODV0203 = 6 THEN 6
```

```
WHEN CD2010ADOMI.CODV0203 = 7 THEN 7
WHEN CD2010ADOMI.CODV0203 = 8 THEN 8
WHEN CD2010ADOMI.CODV0203 = 9 THEN 9
WHEN CD2010ADOMI.CODV0203 IN (10, 11) THEN 10
ELSE -1
END AS INDICADORA_C01,
CASE
  WHEN CD2010ADOMI.CODV0204 IN (-1, 1, 2, 3, 4, 5, 6, 7, 8, 9)
    THEN 9999
  ELSE -1
END AS INDICADORA_D00,
CASE
  WHEN CD2010ADOMI.CODV0204 = 1 THEN 1
  WHEN CD2010ADOMI.CODV0204 = 2 THEN 2
  WHEN CD2010ADOMI.CODV0204 = 3 THEN 3
  WHEN CD2010ADOMI.CODV0204 = 4 THEN 4
  WHEN CD2010ADOMI.CODV0204 = 5 THEN 5
  WHEN CD2010ADOMI.CODV0204 IN (6, 7, 8, 9) THEN 6
  ELSE -1
END AS INDICADORA_D01,
CASE
  WHEN CD2010ADOMI.CODV0209 IN (-1, 1, 2, 3) THEN 9999
  ELSE -1
END AS INDICADORA_E00,
CASE
  WHEN CD2010ADOMI.CODV0209 = 1 THEN 1
  WHEN CD2010ADOMI.CODV0209 = 2 THEN 6
  WHEN CD2010ADOMI.CODV0209 = 3 THEN 11
  ELSE -1
END AS INDICADORA_E01,
CASE
  WHEN CD2010ADOMI.CODV0209 = 1 AND
    CD2010ADOMI.CODV0208 = 1 THEN 2
  WHEN CD2010ADOMI.CODV0208 = 2 AND
    CD2010ADOMI.CODV0209 = 1 THEN 3
```



```

    WHEN CD2010ADOMI.CODV0209 = 1 AND
        CD2010ADOMI.CODV0208 = 3 THEN 4
    WHEN CD2010ADOMI.CODV0209 = 1 AND
        CD2010ADOMI.CODV0208 IN (4, 5, 6, 7, 8, 9, 10) THEN 5
    WHEN CD2010ADOMI.CODV0209 = 2 AND
        CD2010ADOMI.CODV0208 = 1 THEN 7
    WHEN CD2010ADOMI.CODV0209 = 2 AND
        CD2010ADOMI.CODV0208 = 2 THEN 8
    WHEN CD2010ADOMI.CODV0209 = 2 AND
        CD2010ADOMI.CODV0208 = 3 THEN 9
    WHEN CD2010ADOMI.CODV0209 = 2 AND
        CD2010ADOMI.CODV0208 IN (4, 5, 6, 7, 8, 9, 10) THEN 10
    ELSE -1
END AS INDICADORA_E02,
CD2010ADOMI.PESODOMI,
CD2010ADOMI.V6529,
CD2010ADOMI.CODV4001,
CD2010ADOMI.V0401,
CD2010ADOMI.CODPAIS AS INDICADORA_GEOG1,
CD2010ADOMI.CODANOPESQ AS INDICADORA_TEMPORAL
FROM IBGE.CD2010ADOMI
WHERE CD2010ADOMI.CODANOPESQ = 2010
AND CD2010ADOMI.CODV4001 = 1
AND CD2010ADOMI.V6529 > 0
AND CD2010ADOMI.V6529 <= 9999999) A
GROUP BY A.INDICADORA_GEOG1,
    GROUPING SETS (
        (A.INDICADORA_A00, A.INDICADORA_B00, A.INDICADORA_C00,
            A.INDICADORA_D00, A.INDICADORA_E00),
        (A.INDICADORA_A00, A.INDICADORA_B00, A.INDICADORA_C00,
            A.INDICADORA_D00, A.INDICADORA_E01),
        (A.INDICADORA_A00, A.INDICADORA_B00, A.INDICADORA_C00,
            A.INDICADORA_D00, A.INDICADORA_E02),
        (A.INDICADORA_A00, A.INDICADORA_B00, A.INDICADORA_C00,
            A.INDICADORA_D01, A.INDICADORA_E00),

```

(A.INDICADORA\_A00, A.INDICADORA\_B00, A.INDICADORA\_C00,  
A.INDICADORA\_D01, A.INDICADORA\_E01),  
(A.INDICADORA\_A00, A.INDICADORA\_B00, A.INDICADORA\_C00,  
A.INDICADORA\_D01, A.INDICADORA\_E02),  
(A.INDICADORA\_A00, A.INDICADORA\_B00, A.INDICADORA\_C01,  
A.INDICADORA\_D00, A.INDICADORA\_E00),  
(A.INDICADORA\_A00, A.INDICADORA\_B00, A.INDICADORA\_C01,  
A.INDICADORA\_D00, A.INDICADORA\_E01),  
(A.INDICADORA\_A00, A.INDICADORA\_B00, A.INDICADORA\_C01,  
A.INDICADORA\_D00, A.INDICADORA\_E02),  
(A.INDICADORA\_A00, A.INDICADORA\_B00, A.INDICADORA\_C01,  
A.INDICADORA\_D01, A.INDICADORA\_E00),  
(A.INDICADORA\_A00, A.INDICADORA\_B00, A.INDICADORA\_C01,  
A.INDICADORA\_D01, A.INDICADORA\_E01),  
(A.INDICADORA\_A00, A.INDICADORA\_B00, A.INDICADORA\_C01,  
A.INDICADORA\_D01, A.INDICADORA\_E02),  
(A.INDICADORA\_A00, A.INDICADORA\_B01, A.INDICADORA\_C00,  
A.INDICADORA\_D00, A.INDICADORA\_E00),  
(A.INDICADORA\_A00, A.INDICADORA\_B01, A.INDICADORA\_C00,  
A.INDICADORA\_D00, A.INDICADORA\_E01),  
(A.INDICADORA\_A00, A.INDICADORA\_B01, A.INDICADORA\_C00,  
A.INDICADORA\_D00, A.INDICADORA\_E02),  
(A.INDICADORA\_A00, A.INDICADORA\_B01, A.INDICADORA\_C00,  
A.INDICADORA\_D01, A.INDICADORA\_E00),  
(A.INDICADORA\_A00, A.INDICADORA\_B01, A.INDICADORA\_C00,  
A.INDICADORA\_D01, A.INDICADORA\_E01),  
(A.INDICADORA\_A00, A.INDICADORA\_B01, A.INDICADORA\_C00,  
A.INDICADORA\_D01, A.INDICADORA\_E02),  
(A.INDICADORA\_A00, A.INDICADORA\_B01, A.INDICADORA\_C01,  
A.INDICADORA\_D00, A.INDICADORA\_E00),  
(A.INDICADORA\_A00, A.INDICADORA\_B01, A.INDICADORA\_C01,  
A.INDICADORA\_D00, A.INDICADORA\_E01),  
(A.INDICADORA\_A00, A.INDICADORA\_B01, A.INDICADORA\_C01,  
A.INDICADORA\_D00, A.INDICADORA\_E02),  
(A.INDICADORA\_A00, A.INDICADORA\_B01, A.INDICADORA\_C01,

A.INDICADORA\_D01, A.INDICADORA\_E00),  
(A.INDICADORA\_A00, A.INDICADORA\_B01, A.INDICADORA\_C01,  
A.INDICADORA\_D01, A.INDICADORA\_E01),  
(A.INDICADORA\_A00, A.INDICADORA\_B01, A.INDICADORA\_C01,  
A.INDICADORA\_D01, A.INDICADORA\_E02),  
(A.INDICADORA\_A01, A.INDICADORA\_B00, A.INDICADORA\_C00,  
A.INDICADORA\_D00, A.INDICADORA\_E00),  
(A.INDICADORA\_A01, A.INDICADORA\_B00, A.INDICADORA\_C00,  
A.INDICADORA\_D00, A.INDICADORA\_E01),  
(A.INDICADORA\_A01, A.INDICADORA\_B00, A.INDICADORA\_C00,  
A.INDICADORA\_D00, A.INDICADORA\_E02),  
(A.INDICADORA\_A01, A.INDICADORA\_B00, A.INDICADORA\_C00,  
A.INDICADORA\_D01, A.INDICADORA\_E00),  
(A.INDICADORA\_A01, A.INDICADORA\_B00, A.INDICADORA\_C00,  
A.INDICADORA\_D01, A.INDICADORA\_E01),  
(A.INDICADORA\_A01, A.INDICADORA\_B00, A.INDICADORA\_C00,  
A.INDICADORA\_D01, A.INDICADORA\_E02),  
(A.INDICADORA\_A01, A.INDICADORA\_B00, A.INDICADORA\_C01,  
A.INDICADORA\_D00, A.INDICADORA\_E00),  
(A.INDICADORA\_A01, A.INDICADORA\_B00, A.INDICADORA\_C01,  
A.INDICADORA\_D00, A.INDICADORA\_E01),  
(A.INDICADORA\_A01, A.INDICADORA\_B00, A.INDICADORA\_C01,  
A.INDICADORA\_D01, A.INDICADORA\_E02),  
(A.INDICADORA\_A01, A.INDICADORA\_B00, A.INDICADORA\_C01,  
A.INDICADORA\_D01, A.INDICADORA\_E00),  
(A.INDICADORA\_A01, A.INDICADORA\_B00, A.INDICADORA\_C01,  
A.INDICADORA\_D01, A.INDICADORA\_E01),  
(A.INDICADORA\_A01, A.INDICADORA\_B00, A.INDICADORA\_C01,  
A.INDICADORA\_D01, A.INDICADORA\_E02),  
(A.INDICADORA\_A01, A.INDICADORA\_B01, A.INDICADORA\_C00,  
A.INDICADORA\_D00, A.INDICADORA\_E00),  
(A.INDICADORA\_A01, A.INDICADORA\_B01, A.INDICADORA\_C00,  
A.INDICADORA\_D00, A.INDICADORA\_E01),  
(A.INDICADORA\_A01, A.INDICADORA\_B01, A.INDICADORA\_C00,  
A.INDICADORA\_D00, A.INDICADORA\_E02),

```

        (A.INDICADORA_A01, A.INDICADORA_B01, A.INDICADORA_C00,
         A.INDICADORA_D01, A.INDICADORA_E00),
        (A.INDICADORA_A01, A.INDICADORA_B01, A.INDICADORA_C00,
         A.INDICADORA_D01, A.INDICADORA_E01),
        (A.INDICADORA_A01, A.INDICADORA_B01, A.INDICADORA_C00,
         A.INDICADORA_D01, A.INDICADORA_E02),
        (A.INDICADORA_A01, A.INDICADORA_B01, A.INDICADORA_C01,
         A.INDICADORA_D00, A.INDICADORA_E00),
        (A.INDICADORA_A01, A.INDICADORA_B01, A.INDICADORA_C01,
         A.INDICADORA_D00, A.INDICADORA_E01),
        (A.INDICADORA_A01, A.INDICADORA_B01, A.INDICADORA_C01,
         A.INDICADORA_D00, A.INDICADORA_E02),
        (A.INDICADORA_A01, A.INDICADORA_B01, A.INDICADORA_C01,
         A.INDICADORA_D01, A.INDICADORA_E00),
        (A.INDICADORA_A01, A.INDICADORA_B01, A.INDICADORA_C01,
         A.INDICADORA_D01, A.INDICADORA_E01),
        (A.INDICADORA_A01, A.INDICADORA_B01, A.INDICADORA_C01,
         A.INDICADORA_D01, A.INDICADORA_E02))
HAVING (
(NVL(A.INDICADORA_A00, -1) <> -1
OR NVL(A.INDICADORA_A01, -1) <> -1)
AND (NVL(A.INDICADORA_B00, -1) <> -1
OR NVL(A.INDICADORA_B01, -1) <> -1)
AND (NVL(A.INDICADORA_C00, -1) <> -1
OR NVL(A.INDICADORA_C01, -1) <> -1)
AND (NVL(A.INDICADORA_D00, -1) <> -1
OR NVL(A.INDICADORA_D01, -1) <> -1)
AND (NVL(A.INDICADORA_E00, -1) <> -1
OR NVL(A.INDICADORA_E01, -1) <> -1
OR NVL(A.INDICADORA_E02, -1) <> -1))

```

**T3534:**

SELECT

```
A.INDICADORA_GEOG1,
A.INDICADORA_A00,
A.INDICADORA_A01,
A.INDICADORA_B00,
A.INDICADORA_B01,
A.INDICADORA_C00,
A.INDICADORA_C01,
A.INDICADORA_C02,
A.INDICADORA_C03,
A.INDICADORA_C04,
A.INDICADORA_C05,
A.INDICADORA_C06,
A.INDICADORA_C07,
A.INDICADORA_C08,
A.INDICADORA_C09,
A.INDICADORA_C10,
SUM(PESODOMI) AS V884,
SUM(A.V6529 * A.PESODOMI) / SUM(A.PESODOMI) AS V878,
SUM(A.V0401 * A.PESODOMI) AS V1883
FROM (SELECT
CASE
WHEN CD2010ADOMI.CODV1006 IN (-1, 1, 2) THEN 9999
ELSE -1
END AS INDICADORA_A00,
CASE
WHEN CD2010ADOMI.CODV1006 = 1 THEN 1
WHEN CD2010ADOMI.CODV1006 = 2 THEN 2
ELSE -1
END AS INDICADORA_A01,
CASE
WHEN CD2010ADOMI.CODV0203 IN (-1, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11)
THEN 9999
ELSE -1
END AS INDICADORA_B00,
CASE
```

```
WHEN CD2010ADOMI.CODV0203 = 1 THEN 1
WHEN CD2010ADOMI.CODV0203 = 2 THEN 2
WHEN CD2010ADOMI.CODV0203 = 3 THEN 3
WHEN CD2010ADOMI.CODV0203 = 4 THEN 4
WHEN CD2010ADOMI.CODV0203 = 5 THEN 5
WHEN CD2010ADOMI.CODV0203 = 6 THEN 6
WHEN CD2010ADOMI.CODV0203 = 7 THEN 7
WHEN CD2010ADOMI.CODV0203 = 8 THEN 8
WHEN CD2010ADOMI.CODV0203 = 9 THEN 9
WHEN CD2010ADOMI.CODV0203 IN (10, 11) THEN 10
ELSE -1
END AS INDICADORA_B01,
CASE
  WHEN CD2010ADOMI.CODV4001 = 1 THEN 9999
  ELSE -1
END AS INDICADORA_C00,
CASE
  WHEN CD2010ADOMI.CODV0213 = 1 THEN 1
  ELSE -1
END AS INDICADORA_C01,
CASE
  WHEN CD2010ADOMI.CODV0214 = 1 THEN 2
  ELSE -1
END AS INDICADORA_C02,
CASE
  WHEN CD2010ADOMI.CODV0215 = 1 THEN 3
  ELSE -1
END AS INDICADORA_C03,
CASE
  WHEN CD2010ADOMI.CODV0216 = 1 THEN 4
  ELSE -1
END AS INDICADORA_C04,
CASE
  WHEN CD2010ADOMI.CODV0217 = 1 THEN 5
  ELSE -1
```

```
END AS INDICADORA_C05,
CASE
  WHEN CD2010ADOMI.CODV0218 = 1 THEN 6
  ELSE -1
END AS INDICADORA_C06,
CASE
  WHEN CD2010ADOMI.CODV0219 = 1 THEN 7
  ELSE -1
END AS INDICADORA_C07,
CASE
  WHEN CD2010ADOMI.CODV0221 = 1 THEN 9
  ELSE -1
END AS INDICADORA_C08,
CASE
  WHEN CD2010ADOMI.CODV0222 = 1 THEN 10
  ELSE -1
END AS INDICADORA_C09,
CASE
  WHEN CD2010ADOMI.CODV0220 = 1 THEN 8
  ELSE -1
END AS INDICADORA_C10,
CD2010ADOMI.PESODOMI,
CD2010ADOMI.V6529,
CD2010ADOMI.CODV4001,
CD2010ADOMI.V0401,
CD2010ADOMI.CODPAIS AS INDICADORA_GEOG1,
CD2010ADOMI.CODANOPESQ AS INDICADORA_TEMPORAL
FROM IBGE.CD2010ADOMI
WHERE CD2010ADOMI.CODANOPESQ = 2010
AND CD2010ADOMI.CODV4001 = 1
AND CD2010ADOMI.V6529 > 0
AND CD2010ADOMI.V6529 <= 9999999) A
GROUP BY A.INDICADORA_GEOG1,
        GROUPING SETS (
          (A.INDICADORA_A00, A.INDICADORA_B00, A.INDICADORA_C00),
```

(A.INDICADORA\_A00, A.INDICADORA\_B00, A.INDICADORA\_C01),  
(A.INDICADORA\_A00, A.INDICADORA\_B00, A.INDICADORA\_C02),  
(A.INDICADORA\_A00, A.INDICADORA\_B00, A.INDICADORA\_C03),  
(A.INDICADORA\_A00, A.INDICADORA\_B00, A.INDICADORA\_C04),  
(A.INDICADORA\_A00, A.INDICADORA\_B00, A.INDICADORA\_C05),  
(A.INDICADORA\_A00, A.INDICADORA\_B00, A.INDICADORA\_C06),  
(A.INDICADORA\_A00, A.INDICADORA\_B00, A.INDICADORA\_C07),  
(A.INDICADORA\_A00, A.INDICADORA\_B00, A.INDICADORA\_C08),  
(A.INDICADORA\_A00, A.INDICADORA\_B00, A.INDICADORA\_C09),  
(A.INDICADORA\_A00, A.INDICADORA\_B00, A.INDICADORA\_C10),  
(A.INDICADORA\_A00, A.INDICADORA\_B01, A.INDICADORA\_C00),  
(A.INDICADORA\_A00, A.INDICADORA\_B01, A.INDICADORA\_C01),  
(A.INDICADORA\_A00, A.INDICADORA\_B01, A.INDICADORA\_C02),  
(A.INDICADORA\_A00, A.INDICADORA\_B01, A.INDICADORA\_C03),  
(A.INDICADORA\_A00, A.INDICADORA\_B01, A.INDICADORA\_C04),  
(A.INDICADORA\_A00, A.INDICADORA\_B01, A.INDICADORA\_C05),  
(A.INDICADORA\_A00, A.INDICADORA\_B01, A.INDICADORA\_C06),  
(A.INDICADORA\_A00, A.INDICADORA\_B01, A.INDICADORA\_C07),  
(A.INDICADORA\_A00, A.INDICADORA\_B01, A.INDICADORA\_C08),  
(A.INDICADORA\_A00, A.INDICADORA\_B01, A.INDICADORA\_C09),  
(A.INDICADORA\_A00, A.INDICADORA\_B01, A.INDICADORA\_C10),  
(A.INDICADORA\_A01, A.INDICADORA\_B00, A.INDICADORA\_C00),  
(A.INDICADORA\_A01, A.INDICADORA\_B00, A.INDICADORA\_C01),  
(A.INDICADORA\_A01, A.INDICADORA\_B00, A.INDICADORA\_C02),  
(A.INDICADORA\_A01, A.INDICADORA\_B00, A.INDICADORA\_C03),  
(A.INDICADORA\_A01, A.INDICADORA\_B00, A.INDICADORA\_C04),  
(A.INDICADORA\_A01, A.INDICADORA\_B00, A.INDICADORA\_C05),  
(A.INDICADORA\_A01, A.INDICADORA\_B00, A.INDICADORA\_C06),  
(A.INDICADORA\_A01, A.INDICADORA\_B00, A.INDICADORA\_C07),  
(A.INDICADORA\_A01, A.INDICADORA\_B00, A.INDICADORA\_C08),  
(A.INDICADORA\_A01, A.INDICADORA\_B00, A.INDICADORA\_C09),  
(A.INDICADORA\_A01, A.INDICADORA\_B00, A.INDICADORA\_C10),  
(A.INDICADORA\_A01, A.INDICADORA\_B01, A.INDICADORA\_C00),  
(A.INDICADORA\_A01, A.INDICADORA\_B01, A.INDICADORA\_C01),  
(A.INDICADORA\_A01, A.INDICADORA\_B01, A.INDICADORA\_C02),



```
(A.INDICADORA_A01, A.INDICADORA_B01, A.INDICADORA_C03),
(A.INDICADORA_A01, A.INDICADORA_B01, A.INDICADORA_C04),
(A.INDICADORA_A01, A.INDICADORA_B01, A.INDICADORA_C05),
(A.INDICADORA_A01, A.INDICADORA_B01, A.INDICADORA_C06),
(A.INDICADORA_A01, A.INDICADORA_B01, A.INDICADORA_C07),
(A.INDICADORA_A01, A.INDICADORA_B01, A.INDICADORA_C08),
(A.INDICADORA_A01, A.INDICADORA_B01, A.INDICADORA_C09),
(A.INDICADORA_A01, A.INDICADORA_B01, A.INDICADORA_C10))
HAVING (
(NVL(A.INDICADORA_A00, -1) <> -1
OR NVL(A.INDICADORA_A01, -1) <> -1)
AND (NVL(A.INDICADORA_B00, -1) <> -1
OR NVL(A.INDICADORA_B01, -1) <> -1)
AND (NVL(A.INDICADORA_C00, -1) <> -1
OR NVL(A.INDICADORA_C01, -1) <> -1
OR NVL(A.INDICADORA_C02, -1) <> -1
OR NVL(A.INDICADORA_C03, -1) <> -1
OR NVL(A.INDICADORA_C04, -1) <> -1
OR NVL(A.INDICADORA_C05, -1) <> -1
OR NVL(A.INDICADORA_C06, -1) <> -1
OR NVL(A.INDICADORA_C07, -1) <> -1
OR NVL(A.INDICADORA_C08, -1) <> -1
OR NVL(A.INDICADORA_C09, -1) <> -1
OR NVL(A.INDICADORA_C10, -1) <> -1))
```