Computação de Alto desempenho em Nuvens

Ementa:

1. Why Move HPC Applications to the Cloud?
2. What is Cloud Computing?
3. What do HPC applications look like?
4. Deploying Infrastructure and Applications
5. Executing Traditional HPC Application Code in the Cloud
6. Designing Cloud-friendly HPC Applications
7. Exploiting Hardware Accelerators in Clouds
8. Optimizing Infrastructure for MPI Applications
9. Harnessing Low-Cost Virtual Machines on the Spot
10. Ensuring Application Continuity with Fault Tolerance Techniques
11. Avoiding Resource Wastage

Programa:

I: Foundations

1. What is Cloud Computing?

This part introduces the fundamentals of Cloud Computing. Besides describing the origin and definition, this part also presents service and delivery models of this computing model. We also introduce virtualization and containerization, which are core technologies of Cloud Computing.

2. What do HPC applications look like?

High-performance computing (HPC), which aims at offering a fast computing solution from the machines and the programming standpoints, has also adopted the cloud computing paradigm and associated systems as a serious alternative. However, the strong requirement of efficiency w.r.t HPC implementations raises several questions and concerns when it comes to the deployment on the cloud. To understand the main points and guide the thoughts for scientific and technical investigations, it is important to understand the key aspects of HPC. This is the aim of the current part , which provides a view of the HPC landscape and techniques, so as to help for a better understanding of this specific context.

II: Running HPC Applications in Cloud

3. Deploying Infrastructure and Applications

Executing HPC workloads on the cloud usually requires deploying and configuring different infrastructure components so that they can work together to execute the desired workload. This can be done in several ways, using different tools and paradigms. This part provides an overview of the key cloud infrastructure elements for HPC workloads and how they can be instantiated and managed. It also discusses the advantages and drawbacks of using different forms for performing this process and which aspects must be observed by the user before choosing one of them.

4. Executing Traditional HPC Application Code in the Cloud

This part discusses strategies and techniques to pave the way for executing traditional HPC applications in the Cloud. We follow one direction that may appear unusual at first glance. Indeed, we sketch the problems, issues, and solutions when the effort is put into the HPC scheduler. We mean that the HPC applications are not rewritten, but the HPC scheduler has been cloudified. Thus, it is now available as any other Cloud service, on-demand. Then, this part introduces the issues for a Cloud orchestrator controller that enables the autoscaling of containerized HPC Clusters in the Cloud. The proposed solution aims to trigger the creation or suppression of containerized HPC compute nodes according to metrics collected at the containerized HPC scheduler's job queue level. In summarizing, we underline approaches specialized to address the unification of HPC and Cloud. Following our options, perspectives are numerous, and we summarize some of them as future works.

5. Designing Cloud-friendly HPC Applications

This part presents how we can design cloud-friendly HPC applications. The key objective is first to detail the main features of cloud computing, then detail how the existing HPC application models can address each feature. In the end, we will have the best insights for developing HPC applications to run in the cloud and the hardware aspects that influence the applications' performance. Thus, we plan to provide a short best-practices guide that could be followed when planning to use (or migrate to) cloud testbeds to run parallel applications.

6. Exploiting Hardware Accelerators in Clouds

In this part, we plan to provide an overview of cloud accelerators available on different cloud providers. We plan to introduce, show how to program, instantiate and discuss a deployment workflow. The goal is to provide an overview, but each step details shall be discussed in the following sections. Accelerator Optimized Instances on the Cloud Cloud providers offer several accelerators such as GPUs, TPUs, and FPGAs. This section aims to introduce the instance types and discuss the main characteristics of each computer architecture. (GPUs - Graphic Processing Units, TPUs - Tensor Processing Units, FPGAs - Field-Programmable Gate Arrays..) Programming for Cloud Accelerators This section plans to show some guidelines on deploying applications for accelerators available on the cloud. We focus on data storage and accelerator instantiation. Also, we show how to train and save the model in the cloud provider bucket, deploy new versions of the neural networks, and have the endpoint for inference globally available.

III: Cost and Performance Optimizations

7. Optimizing Infrastructure for MPI Applications

MPI is a de facto standard and an effective mechanism for developing portable parallel applications. An MPI executing program is called a job, and it consists of a collection of tasks that exchange data. With the advent of cloud computing,

There is increased interest in running MPI parallel applications on the cloud. The elastic characteristics of the cloud, with the ability to allocate vast computational resources on demand, are a good match for large parallel applications that are often coded with MPI. Nevertheless, MPI computing imposes specific demands on cloud infrastructure. That infrastructure needs to be optimized for the specific mode of operation represented by MPI applications. In particular, large numbers of compute nodes with appropriate compute and memory resources must be interconnected with fast networks. Multi-level cloud schedulers must coordinate to assign and allocate those nodes to submitted MPI jobs. All this infrastructure must work efficiently to deliver a productive experience to demanding MPI users.

8. Harnessing Low-Cost Virtual Machines on the Spot

Public cloud providers offer computing resources through a plethora of Virtual Machine (VM) instances of different capacities. Each instance is composed of a pre-determined set of virtualized hardware components of different types and/or quantities (number of cores, memory, storage and bandwidth capacities, etc.), in an attempt to satisfy the demands of a diverse range of user applications. Typically, cloud providers offer these instances under several contract models that differ in terms of availability guarantees and prices (On-demand, Spot, Reserved). This part provides an overview on how users might utilize and benefit from the variety of instances and different contract models on offer from public cloud providers to reduce their financial outlays. A methodology to dynamically schedule applications with deadline constraints in both hibernation-prone Spot VMs and On-Demand Instances in order to lower costs in relation to a pure On-demand solution is described.

Independent of the chosen contract model, identifying the appropriate instance type for applications is also important when attempting to trim expenses. Since it may not be obvious, a short discussion motivates why this decision is not solely related to defining the required resource capacities the chosen instances should have. Finally, given that some cloud providers have recently introduced the concept of Burstable Instances that can boost their performance for a limited period of time, the part closes with a summary of approaches that exploit the discounted rates afforded by this new instance class.

9. Ensuring Application Continuity with Fault Tolerance Techniques

Outages are not uncommon in clouds and, in this case, the cloud provider and/or HPC applications need to implement fault tolerance mechanisms in order to ensure reliability and the correct execution of the applications. In this part , we present an overview of the related literature about fault tolerance (FT) techniques most used by clouds and HPC applications that run on them, basically checkpoint-rollback and replication, as well as fault detection approaches and existing reliable storage in clouds.

10. Avoiding Resource Wastage

HPC cloud aims at exploiting cloud infrastructures to run HPC workloads. Compared with traditional HPC clusters, cloud offers several advantages in terms of rapid access to elastic and diversified computing resources, economies of use, and release the users from deploying and maintaining physical infrastructures. Nevertheless, users are responsible for managing the resources rented from clouds to run their workloads, a task that becomes even more complex if we consider the heterogeneity of resources and the diversity of pricing models implemented by cloud providers. Inefficient resource management not only increases end-of-month costs, it often degrades the application performance. This part discusses the resource wastage problem in the context of HPC cloud and provides existing state-of-the-art solutions to tackle such situations. To this end, this part describes the classes of HPC applications that benefit from running in the cloud, followed by a formulation of the resource management problem. Then, different metrics to detect resource inefficiencies are introduced and a comparative analysis of several scheduling-based resource optimisation strategies are provided. Although several advances happened in past years in the HPC cloud space, this study identifies the limitations of current solutions that need to be addressed in future research.

Bibliografia:

1- Nick Antonopoulos • Lee Gillam editors, Cloud Computing Principles, Systems and Applications, Springer

2- Edson Borin, Lucia Drummond, Alba Melo, Philippe Navaux, Jean-Luc Guadiot editors. High Performance Computing in Clouds: Moving Applications to a Scalable and Cost-Effective Environment, Springer, em producao.