

# Resumo

A popularidade de temas como computação nas nuvens e computação verde é consequência da crescente necessidade de maximizar a utilização de recursos computacionais. Há uma constante busca pelo aumento da capacidade de processamento sem a necessidade de incremento da quantidade de recursos e principalmente do consumo de energia.

Em grades computacionais a alocação das aplicações pelos recursos disponíveis é feito na maioria das vezes através de sistemas gerenciadores. A capacidade de um ambiente de grade maximizar o uso de seus recursos sem comprometer custos está diretamente ligado a eficiência de seu sistema gerenciador. Os sistemas gerenciadores de grade atuais funcionam como *brokers* que alocam aplicações em subconjuntos de recursos disponíveis. No processo de alocação, caso uma aplicação já esteja executando sobre um determinado recurso o gerenciador não irá alocar outra ao recurso até que a primeira tenha concluído sua execução. Assim, como não há compartilhamento simultâneo não haverá perda de eficiência causada pela troca de contexto das aplicações.

Entretanto, na maioria das vezes uma aplicação não utiliza ao mesmo tempo todos os serviços (*CPU*, *GPU*, entrada/saída, etc.) oferecidos por um recurso. Dessa forma, a alocação dedicada de um recurso a uma única aplicação subutiliza o mesmo. Além disso, como a alocação é feita por bloco, ou seja, um determinado conjunto de recursos é alocado a uma única aplicação até seu término, recursos poderão ficar ociosos uma vez que a aplicação não necessariamente utilizará todos os recursos durante toda sua execução.

Com o crescente número e variedade de aplicações e recursos tentando tirar proveito da computação em grades, o uso de um *broker* central torna-se mais complexo e não escalável. Uma alternativa para o uso de um *broker* central orientado ao sistema é um modelo de gerenciamento baseado nos conceitos de controle distribuído e aplicações autônomas. Nesta abordagem, aplicações se automonitoram, se ajustam dinamicamente aos recursos disponíveis e executam ações a fim de alcançar seus objetivos. O principal desafio desta idéia é definir qual deve ser o comportamento dessas entidades autônomas para que a sociedade (ou conjunto) de aplicações seja sustentável. O objetivo deste trabalho é avaliar a viabilidade desta idéia e propor uma solução de implementação para a mesma.

**Palavras-chave:** sistema gerenciador de grades, computação autônoma, sistema gerenciador de aplicações EasyGrid

# Abstract

The popularity of cloud computing and green IT has been motivated by the growing demand for improved and cost effective resource utilization. System designers today must aim to increase processing capacity or output without simply aggregating ever larger numbers of resources at the expense of being faced with the corresponding explosion in power consumption.

In grid computing, the allocation of applications to available resources is mostly performed by Resource Management Systems (RMSs). Thus the capacity of the grid to maximize resource usage without compromising costs is directly linked to the efficiency of its RMS. Current grid RMSs typically operate as *brokers* to allocate jobs to the best subset of resources available. In the currently adopted allocation model, each job is scheduled exclusively to a given (set of) resource(s). Thus the RMS won't allocate another job to those resources until the previous job has ended, with the hypothesis being to avoid any loss in efficiency due to context switching between concurrently executing jobs.

However, most applications do not use all of the services (CPU, GPU, I/O, etc.) offered by a resource at the same time. Thus, the allocation of just one application to a resource may leave that resource severely underutilized. Moreover, resources are allocated to jobs in blocks of a size equivalent to application's maximum processor requirement. In other words, a fixed subset of resources is allocated exclusively to a unique application until it completes its execution, independent of whether or not individual resources are in fact being utilized for the entire duration.

With the growing variety of applications aiming to take advantage of grids with large numbers of heterogeneous resources, designing an efficient single central broker becomes extremely complex and unlikely to scale up. An alternative proposal is a management model based on autonomic computing concepts. In this approach, applications are self-managed, and thus are capable of discovering the best available resources dynamically and autonomously. Distributed Application Management Systems understand the specific requirements of their applications and can be designed to permit sharing of underutilized resources. The main challenge to turn this idea into a viable concept is to define the necessary behaviors required by a society of autonomous applications to maintain a sustainable and efficient utilization of the grid environment. The objective of this work is to evaluate the viability of this proposal and define some appropriate behaviors for autonomic MPI applications.

**Keywords:** grid middleware, autonomic computing, EasyGrid AMS