

# Um Framework de Escalonamento Flexível para Submissão de Processos em uma Grade Computacional

Valéria Quadros dos Reis\*, Marcos José Santana , Regina Helena Carlucci Santana

Laboratório de Sistemas Distribuídos e Computação Concorrente  
Instituto de Ciências Matemáticas e de Computação - ICMC-USP  
Av. Trabalhador São-carlense, 400 Centro  
CEP: 13566-970 São Carlos, SP

{valeria,mjs,rqs}@icmc.usp.br

## 1. Introdução

Devido aos elevados custos para a aquisição de computadores, capazes de executar em tempo hábil aplicações com grande necessidade de processamento, tem crescido o interesse pela Computação em Grade a qual consiste na execução de processos em infra-estruturas formadas pela união de recursos, muitas vezes espalhados geograficamente, e pertencentes a diferentes organizações e domínios.

Além dos recursos físicos, grades computacionais necessitam de *softwares* capazes de abstrair detalhes próprios da natureza desses sistemas, tais como a autenticação em diversos domínios, obtenção de informações e alocação dos recursos disponíveis. Alguns projetos se propõem a integrar tais funcionalidades em sistemas para computação em grade, dentre os quais destaca-se o projeto Globus [Globus, 2004].

A característica de atribuição de tarefas a recursos de um Sistema Computacional Distribuído é uma atividade difícil e que torna-se ainda mais complexa em uma plataforma de grade devido à grande diversidade de recursos existentes. Sendo assim, está sendo desenvolvido um *framework* o qual possa realizar a submissão de tarefas a recursos de uma grade computacional de maneira a satisfazer algum interesse, expresso pelo usuário através de políticas de escalonamento. O *framework* não permanece limitado às políticas que possui, sendo, por esse motivo, denominado flexível.

## 2. Globus

O Globus é um conjunto de ferramentas de código aberto destinadas à criação e utilização de infra-estruturas de grade de maneira segura. Ele é composto por diversos serviços independentes os quais possuem interfaces disponíveis para serem utilizadas no desenvolvimento de novas aplicações [Jacob et al., 2003].

O gerenciador de recursos do Globus, não possui um escalonador e, dessa maneira, é responsabilidade do usuário escolher em qual máquina ele deseja executar a sua aplicação, o que, freqüentemente, resulta em um mau uso dos recursos.

Outra limitação do Globus é que ele não possui mecanismos para a comunicação entre processos, prejudicando a execução de aplicações que necessitam trocar mensagens entre suas tarefas.

---

\*Projeto Financiado pela CAPES.

### 3. AMIGO

O AMIGO (*DynAMical FlexIble SchedulinG EnvirOnment*) é uma ferramenta responsável pela gerência do escalonamento de processos em plataformas computacionais distribuídas. Isso porque ele possibilita a inclusão de políticas de escalonamento as quais podem ser focadas em diferentes objetivos, de acordo com o determinado pelo usuário [Souza, 2000].

A termo dinâmico refere-se à característica do AMIGO de alterar a política de escalonamento em uso em tempo de execução. Essa possibilidade de alteração na forma de atribuição de tarefas aos processadores atribui flexibilidade ao ambiente e é realizada de forma transparente ao usuário. Até o momento, existem duas versões do AMIGO para ambientes de passagem de mensagem, sendo uma para PVM (*Parallel Virtual Machine*) e a outra para MPI (*Message Passing Interface*).

### 4. Metodologia

Através da interface para programação disponibilizada pelo Globus, foi possível a implementação de uma aplicação para submissão de tarefas em uma grade computacional [Laszewski et al., 2001]. Para isso, foram utilizados os serviços de segurança para a autenticação e autorização, de gerenciamento de dados, e de gerenciamento de recursos. O serviço de descoberta de recursos foi implementado utilizando-se informações obtidas a partir do NWS (*Network Weather Service*) [NWS, 2004]. Essas informações são disponibilizadas através de *Web Services*, para que a interoperabilidade do sistema não seja comprometida.

Na primeira fase, o cliente faz uma consulta a um servidor responsável por armazenar todos os recursos disponíveis no sistema. A resposta, consistindo em uma lista com o endereço de todos os recursos, é utilizada pelo cliente para consultar os próprios recursos e obter informações sobre o estado de cada um deles, como, por exemplo, carga da CPU e porcentagem de memória não-paginada.

As informações colhidas são muito importante pois servem como parâmetros para as políticas de escalonamento em uso que, por sua vez, através da aplicação de suas regras, estabelecem quais os recursos mais aptos para processar as tarefas do cliente. Por fim, a transferência de dados necessários para execução das tarefas é feita e, em seguida, o processamento das mesmas. Caso a aplicação a ser submetida não realize comunicação entre suas tarefas, então para cada tarefa são criadas *threads* responsáveis por monitorar a execução de seu processo. Caso contrário, ela é submetida para um conjunto de máquinas onde exista o PVM/MPI executando o AMIGO *daemon*.

### Referências

- Globus (2004). Página do projeto Globus. Disponível em <http://www.globus.org>.
- Jacob, B., Ferreira, L., Bleberstein, N., Gilzean, C., Girard, J., Strachowski, R., and Yu, S. (2003). Enabling applications for grid computing with globus. Relatório disponível em <http://www.ibm.com/redbooks>.
- Laszewski, G. V., Foster, I., Gawor, J., and Lane, P. (2001). A Java Commodity Grid Kit. *Concurrency and Computation: Practice and Experience*, 13(8-9).
- NWS (2004). Página do projeto NWS. Disponível em <http://nws.cs.ucsb.edu/>.
- Souza, P. S. L. (2000). Amigo: uma contribuição para convergência na Área de escalonamento de processos. Tese de Doutorado, ICMC-USP, São Carlos, SP.