

ESCALONAMENTO DE APLICAÇÕES UTILIZANDO ANÁLISE DE PADRÕES DE USO NO INTEGRADE

MAC499 – Trabalho de Formatura Supervisionado
Aluno: Thiago Henrique Coraini
Orientador: Marcelo Finger

Departamento de Ciência da Computação
Instituto de Matemática e Estatística
Universidade de São Paulo

O que é o *InteGrade*

O *InteGrade* [1] é um projeto entre universidades brasileiras que busca desenvolver um *middleware* para grades computacionais oportunistas. Uma grade computacional oportunista é formada por diversos computadores comuns (não dedicados) ligados em rede que, nos períodos de ociosidade, compartilham seus recursos e juntos disponibilizam um grande poder de processamento para a execução de aplicações.

Escalonamento em grades oportunistas

O escalonamento em grades oportunistas deve ter como preocupação principal garantir que o usuário local de um computador participante da grade não sinta uma degradação na performance de seu PC. Dessa forma, nunca uma aplicação deve ser escalonada para uma máquina que não esteja ociosa, ou que não irá manter-se ociosa por muito tempo.

Para tratar esse problema foi proposto, dentro do projeto *InteGrade*, uma forma de realizar-se previsões sobre o uso futuro de um determinado computador, baseando-se na análise de padrões de uso daquela máquina [2]. Caso seja possível obterem-se resultados confiáveis com essa previsão, o escalonamento pode evitar que aplicações sejam submetidas para computadores que não permanecerão ociosos por muito tempo.

Arquitetura do sistema

Dentro do *InteGrade*, foi desenvolvido um módulo responsável por realizar as previsões de uso futuro com base na análise de padrões de uso [3]. Esse módulo, chamado *LUPA* – *Local Usage Pattern Analyzer*, está constantemente monitorando a utilização de CPU e memória do computador. Essas informações são colocadas em estruturas que representam um período de 48 horas consecutivas e, uma vez ao dia, executa-se um algoritmo que irá agrupar essas estruturas, de acordo com sua semelhança, em aglomerados (*clusters*). Assim, espera-se que cada *cluster* represente um padrão de uso daquele computador.

Para fazer uma previsão, o *LUPA* compara as 24 horas de utilização recente com os *clusters* previamente calculados. Ao encontrar o *cluster* mais próximo dessa utilização, assume que a utilização nas próximas horas seguirá o padrão dos elementos desse *cluster*. Como esses elementos compreendem um período de 48 horas, dependendo do período de previsão requisitado pode ser necessário fazer-se um encadeamento de previsões, ou seja, várias previsões sucessivas que terão seus resultados acumulados para a resposta final.

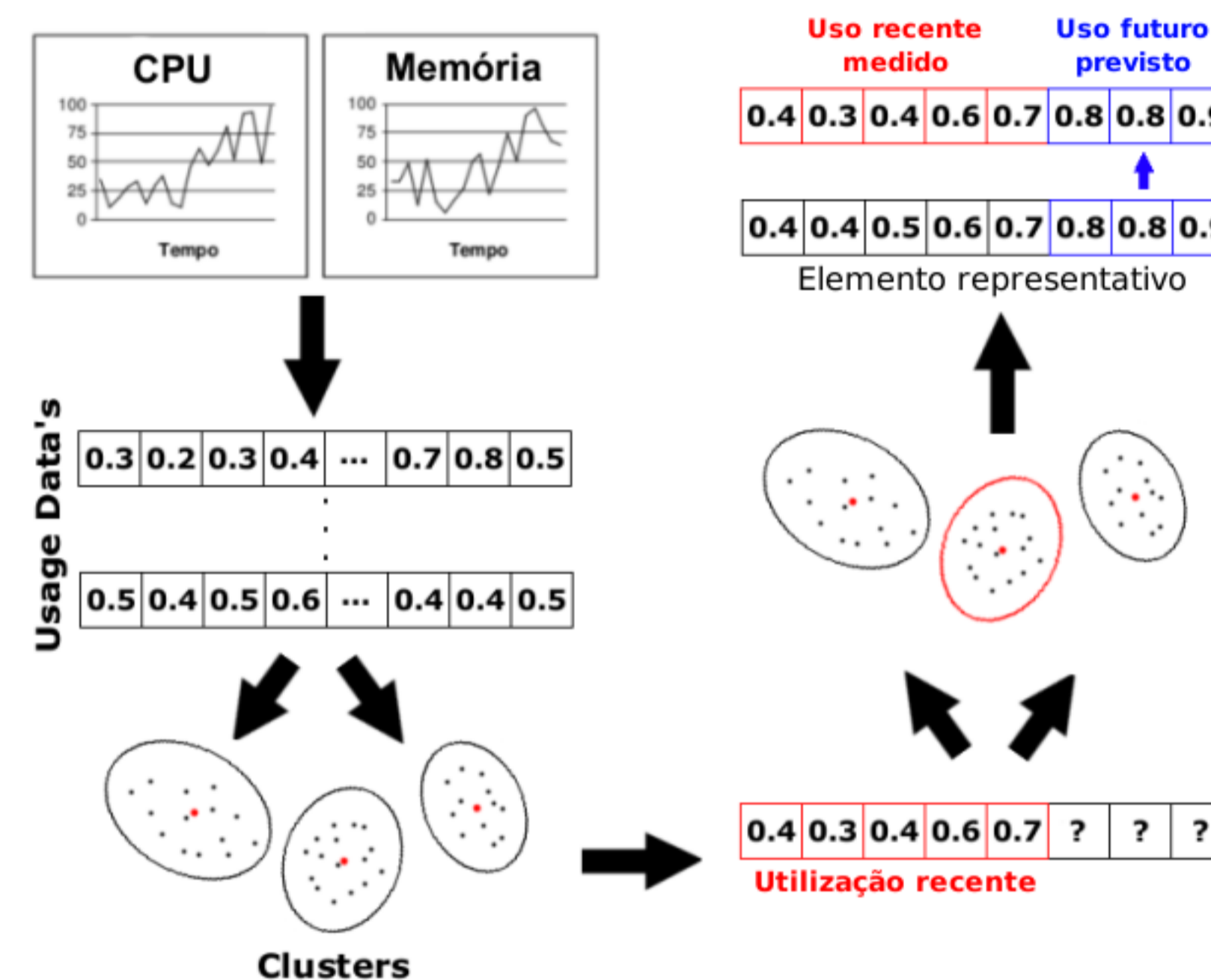


Figura 1: Funcionamento do LUPA. **1-** A utilização do computador é registrada em intervalos de 5 minutos. **2-** Esses dados são agrupados em *UsageData's*, cada um representando 48 horas. **3-** Os *UsageData's* são agrupados por semelhança em *clusters*. **4-** No momento de uma previsão, verifica-se como foi a utilização nas últimas 24 horas. **5-** Encontra-se o *cluster* que possui como elemento representativo aquele mais próximo da utilização recente. **6-** Realiza-se a previsão assumindo que a utilização seguirá o padrão correspondente àquele elemento representativo.

Arquitetura do Escalonador

O escalonador desenvolvido neste trabalho, implementado em Java, trabalha junto ao nó servidor da grade. O que ele faz é, ao receber uma lista inicial de nós candidatos a executar uma aplicação, filtrar essa lista de modo a:

1. Remover quaisquer nós que, segundo previsões do *LUPA*, não irão atender os requisitos de disponibilidade pelo tempo solicitado
2. Reordená-la seguindo alguma heurística específica, dependendo do algoritmo utilizado

Alguns algoritmos foram implementados neste escalonador. São eles:

- *CanRunGridApplication* – Apenas efetua um *polling* dentre os nós da lista original, removendo desta aqueles que não irão manter-se ociosos por tempo suficiente.
- *HowLongCanRunGridApplication* – Escalona as aplicações de maneira a dar preferência pelos nós onde a previsão é de que a disponibilidade de recursos se manterá por mais tempo.

- *GreedyAverageResourceUsage* – Solicita ao *LUPA* de cada nó a média de utilização pelas próximas horas, e então realiza o escalonamento dando preferência para os computadores em que essa média é menor (maior disponibilidade).
- *BestFitAverageResourceUsage* – Usa as mesmas médias de disponibilidade de recursos do algoritmo anterior, porém escalona primeiramente os nós com previsão de menor disponibilidade (porém, sempre suficiente para os requisitos da aplicação).

Resultados

Como resultado desse trabalho, o *InteGrade* passa a contar com um escalonador mais poderoso e inteligente do que o que possuía anteriormente. O escalonador, como mostrado, possui vários algoritmos que usam as previsões do *LUPA* de diferentes maneiras, e a idéia é que, no futuro, esses algoritmos possam ser trocados automaticamente pelo escalonador, caso ele detecte que os escalonamentos não estão sendo bem sucedidos.

Além disso, experimentos estão sendo feitos para tentar comparar a qualidade dos algoritmos citados. Acreditamos, a princípio, que os algoritmos *Greedy* e *BestFit* tenham grande chance de apresentar um melhor desempenho, já que definem uma ordenação para o escalonamento baseados na previsão de ociosidade. Enquanto o algoritmo *Greedy* possivelmente permita que as aplicações encerrem mais rapidamente, já que escalona primeiramente para as máquinas mais ociosas, é possível que, em cenários com grande quantidade de submissões de aplicações à grade, o algoritmo *BestFit* apresente um resultado melhor, já que ele deixará as máquinas mais ociosas livres para aplicações mais pesadas que porventura venham a ser submetidas.

Referências

- [1] A. Goldchleger, *InteGrade: Um Sistema de Middleware para Computação em Grade Oportunista*. PhD thesis, Universidade de São Paulo, 2004.
- [2] G. Bezerra, *Análise de Conglomerados Aplicada ao Reconhecimento de Padrões de Uso de Recursos Computacionais*. PhD thesis, Dissertação de mestrado. Universidade de São Paulo, 2006.
- [3] D. M. R. Conde, *Análise de Padrões de Uso em Grades Computacionais*. PhD thesis, Dissertação de mestrado. Universidade de São Paulo, 2008.