

Escalonamento de aplicações utilizando análise de padrões de uso no InteGrade

Thiago Henrique Coraini
Orientador: Marcelo Finger

MAC499 – Trabalho de Formatura Supervisionado

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

17 de novembro de 2008

O que é o InteGrade

- Projeto entre algumas universidades brasileiras
- É um *middleware* para grades computacionais oportunistas
- Usar computadores comuns para compor a grade
- Aproveitá-los apenas quando não estiverem sendo usados

O problema do escalonamento

- Utilizar recursos somente nos momentos de ociosidade
- Garantir prioridade ao dono da máquina
- O ideal seria poder prever como estaria a utilização do computador num futuro próximo

Previendo o uso futuro

Análise de Padrões de Uso

- Proposta para tentar prever o uso futuro de um computador
- Detectar padrões de utilização recorrentes no computador

Previendo o uso futuro

Análise de Padrões de Uso

- Proposta para tentar prever o uso futuro de um computador
- Detectar padrões de utilização recorrentes no computador

LUPA – Local Usage Pattern Analyzer

Módulo do InteGrade responsável por monitorar o uso, realizar a análise dos padrões e efetuar previsões sobre a utilização futura

Objetivos deste trabalho

Este trabalho consistiu em:

- 1 Realizar alguns aprimoramentos na implementação do *LUPA*
- 2 Ampliar a interface de consulta ao *LUPA*
- 3 Implementar um escalonador de aplicações para a grade, que usasse os dados do *LUPA*

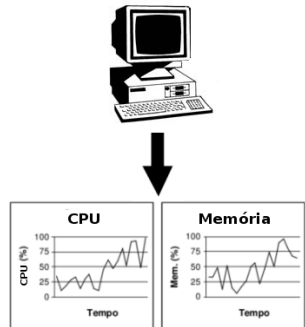
O funcionamento do *LUPA*

O LUPA (*Local Usage Pattern Analyzer*) funciona basicamente da seguinte maneira:

O funcionamento do *LUPA*

O *LUPA* (*Local Usage Pattern Analyzer*) funciona basicamente da seguinte maneira:

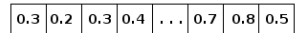
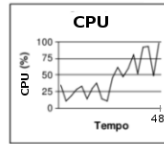
- 1 Monitora o uso de CPU e memória em intervalos pré-definidos



O funcionamento do *LUPA*

O LUPA (*Local Usage Pattern Analyzer*) funciona basicamente da seguinte maneira:

- 1 Monitora o uso de CPU e memória em intervalos pré-definidos
- 2 Agrupa essas medições em períodos de 48 horas (**UsageData**)

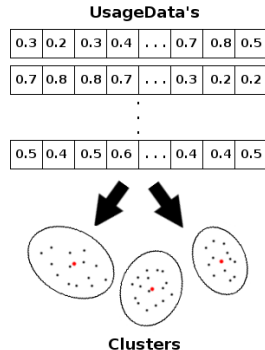


UsageData

O funcionamento do LUPA

O LUPA (*Local Usage Pattern Analyzer*) funciona basicamente da seguinte maneira:

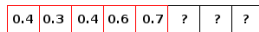
- 1 Monitora o uso de CPU e memória em intervalos pré-definidos
- 2 Agrupa essas medições em períodos de 48 horas (**UsageData**)
- 3 Reúne **UsageData's** semelhantes em aglomerados (*clusters*)



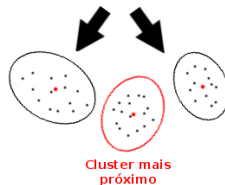
O funcionamento do LUPA

O LUPA (*Local Usage Pattern Analyzer*) funciona basicamente da seguinte maneira:

- 1 Monitora o uso de CPU e memória em intervalos pré-definidos
- 2 Agrupa essas medições em períodos de 48 horas (**UsageData**)
- 3 Reúne **UsageData**'s semelhantes em aglomerados (*clusters*)
- 4 Acha o *cluster* mais próximo da utilização recente



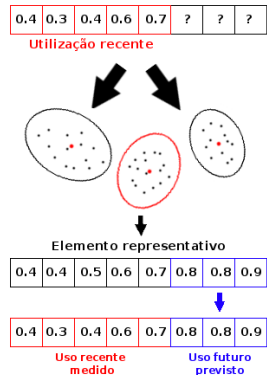
Utilização recente



O funcionamento do LUPA

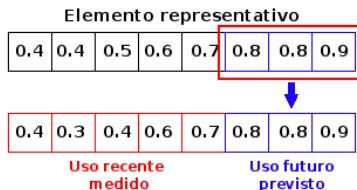
O LUPA (*Local Usage Pattern Analyzer*) funciona basicamente da seguinte maneira:

- 1 Monitora o uso de CPU e memória em intervalos pré-definidos
- 2 Agrupa essas medições em períodos de 48 horas (**UsageData**)
- 3 Reúne **UsageData**'s semelhantes em aglomerados (*clusters*)
- 4 Acha o *cluster* mais próximo da utilização recente
- 5 A partir de um elemento representativo desse *cluster*, infere a utilização futura



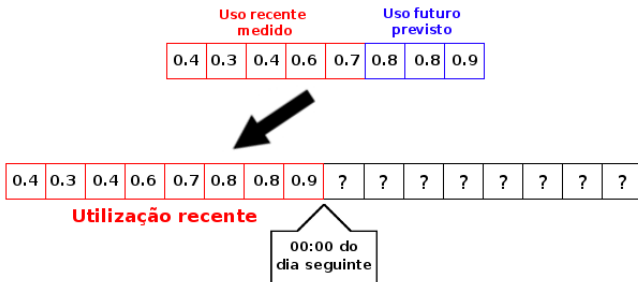
O funcionamento do *LUPA*

Durante a previsão, ao encontrar o elemento representativo apropriado, existe uma limitação no tempo de previsão



O funcionamento do *LUPA*

- Para poder fazer previsões sem limitação de tempo, a solução foi realizar várias previsões sucessivas, encadeadas, e combinar o resultado no final
- A última previsão realizada é usada como medição de utilização recente



O escalonamento

- Para fazer consultas ao *LUPA*, devemos saber quais os requisitos da aplicação
- No futuro, a idéia é descobrir isso automaticamente
- Hoje, isso é passado manualmente na submissão da aplicação

Base path:	/
Binaries:	linux_i686
Constraints:	cpuUsage < 70 and hours == 5
Preferences:	

Application type: Regular BS

O escalonamento

- Primeiramente, faz uma pré-seleção dos nós de acordo com critérios como plataforma e sistema operacional
- Faz uma análise dos requisitos solicitados no momento da submissão
- Através de consultas ao *LUPA*, decide quais os melhores nós para rodarem a aplicação

Algoritmos do escalonador

CanRunGridApplication

- Impede a execução em computadores que não manterão ociosidade
- Não realiza nenhuma ordenação na lista de nós candidatos a executar a aplicação

Algoritmos do escalonador

HowLongCanRunGridApplication

- Verifica, junto ao *LUPA*, por quanto tempo cada nó manterá uma certa disponibilidade
- Dá preferência no escalonamento para aqueles nós que manterão tal disponibilidade por mais tempo

Algoritmos do escalonador

GreedyAverageResourceUsage

- Solicita a média prevista de uso nas próximas horas para todos os nós disponíveis
- Escolhe primeiramente aqueles computadores em que a previsão é de maior disponibilidade de recursos

Algoritmos do escalonador




BestFitAverageResourceUsage

- Solicita a média prevista de uso nas próximas horas para todos os nós disponíveis
- Escolhe primeiramente aqueles computadores em que a previsão de disponibilidade é a menor possível que ainda atenda aos requisitos da aplicação

Resultados

- O escalonador desenvolvido é muito mais inteligente e adaptável do que o que existia anteriormente
- Experimentos estão sendo feitos para tentar comparar a qualidade dos algoritmos apresentados
- O InteGrade está sendo executado nos laboratórios da Rede Linux (IME), em aproximadamente 20 máquinas
- Porém, algumas áreas complementares do projeto ainda precisam ser desenvolvidas para que o escalonamento funcione de maneira realmente eficiente e correta

Referências

-  A. Goldchleger, *InteGrade: Um Sistema de Middleware para Computação em Grade Oportunista*. PhD thesis, Universidade de São Paulo, 2004.
-  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.
-  D. M. R. Conde, *Análise de Padrões de Uso em Grades Computacionais*. PhD thesis, Dissertação de mestrado. Universidade de São Paulo, 2008.

Obrigado!

Perguntas?