

Aplicando uma técnica baseada em perfis para balanceamento adaptativo de carga no ISAM

Gustavo Frainer, Rodrigo Real, Adenauer Yamin,
Luciano da Silva, Iara Augustin, Cláudio Geyer

Universidade Federal do Rio Grande do Sul
Instituto de Informática
{frainer, rreal, adenauer, lucc, august, geyer}@inf.ufrgs.br

Introdução

A consolidação da área de *grid computing* [FOS 99] como uma alternativa real de processamento utilizando recursos geograficamente distribuídos, apresenta uma série de novos desafios para a obtenção de alto desempenho na execução de aplicações paralelas e distribuídas. Por outro lado, é possível encontrar iniciativas neste sentido [REA 2002, YAM 2003, Set 2003], buscando viabilizar a utilização de *grid computing* para o processamento de problemas computacionalmente intensivos.

Dois aspectos dificultam o escalonamento de tarefas no contexto de *grid computing*: (i) a heterogeneidade de recursos (ii) o regime de uso compartilhado usualmente empregado. Estes dois aspectos ficam potencializados em função da regularidade como as tarefas da aplicação paralela podem ser organizadas. A situação em que cada partição do problema a ser processado tem custos computacionais diferentes entre si, e não conhecidos de antemão naturalmente dificulta a obtenção de bons escalonamentos.

Este trabalho tem por objetivo avaliar o comportamento da execução de aplicações irregulares no escopo do projeto ISAM [ISA 2002]. Este projeto contempla um ambiente de desenvolvimento associado a um *middleware* direcionado para o gerenciamento de recursos em redes heterogêneas, suportando mobilidade física e lógica, adaptação dinâmica e a execução de aplicações distribuídas baseadas em componentes.

Será avaliado o emprego de dois mecanismos adaptativos que contemplam: (i) geração automática de um perfil que estima o custo computacional das partições do problema, (ii) dimensionamento e alocação de tarefas considerando a disponibilidade computacional momentânea do recurso de processamento. O segundo mecanismo já foi aplicado a alguns problemas regulares [YAM 2003, FEH 2003, BRU 2003] enquanto que o primeiro foi imaginado especificamente para lidar com aplicações irregulares.

Aplicações irregulares no ISAM

Em [YAM 2003] um modelo para criação de aplicações mestre-escravos utilizando a plataforma ISAM é apresentado. Parte deste modelo é um mecanismo de “*time goal*” que procura ajustar o tamanho da tarefa ao poder computacional livre atual de cada escravo. Este mecanismo funciona bem com aplicações regulares, nas quais o custo computacional da realização de uma tarefa é diretamente proporcional ao seu “tamanho”. En-

tretanto muitos problemas caem na categoria de aplicações irregulares, nas quais outros fatores além do tamanho da partição influenciam no custo computacional. O balanceamento de carga através do “*time goal*” fica comprometido neste tipo de aplicação já que não é possível, ou muito custoso, prever o custo computacional de uma tarefa.

Felizmente esta irregularidade geralmente não apresenta um padrão aleatório. Em vários casos ela obedece um princípio de localidade. Assim, determinadas regiões ou trechos de um problema tem custos computacionais similares. Nestes casos é possível propor um mecanismo de perfil, o qual irá dar uma ideia de qual o custo computacional de cada uma das partes em relação as outras e irá permitir a realização de previsões sobre o tempo de processamento necessário para cada uma delas.

Aplicando os Mecanismos Adaptativos ao Algoritmo de RayTracing

Para permitir um melhor entendimento de como a ideia de perfil pode ser implementada em um problema real, uma versão paralela do algoritmo de RayTracing foi criada utilizando-o. O algoritmo de RayTracing faz a renderização de imagens em 3 dimensões. Ele pode ser facilmente paralelizável já que cada ponto da imagem final pode ser processado de forma independente e é irregular pois regiões da imagem com muitos objetos são mais complexas para serem processadas do que regiões com poucos objetos. Como mencionado anteriormente um esquema mestre-escravo é utilizado.

No início da execução paralela do RayTracing o mestre particiona a imagem a ser gerada em n regiões de igual dimensão. O *perfil* é obtido aplicando-se o algoritmo de RayTracing a um pequeno número de pontos de cada região. A previsão do tempo médio que será gasto em cada ponto daquela região é então dada pela seguinte fórmula:

$$P(r) = \sum_{k=1}^n \frac{T(k)}{n} \quad (1)$$

onde r é a região, n o número total de pontos calculados naquela região e $T(k)$ o tempo gasto para processar o ponto k .

A previsão obtida não será um indicador exato do custo de processamento necessário para a região, mas sim uma estimativa. Isto é uma característica do próprio mecanismo de perfil.

A partir da previsão do custo computacional de cada região é possível implementar uma estratégia de “*time goal*” para controlar a granulosidade da tarefa, conforme explicado em [YAM 2003]. Nesta estratégia o escravo determina qual deve ser o custo computacional da sua próxima tarefa baseado em quanto tempo ele demorou para realizar a tarefa anterior. O mestre irá tentar entregar para este escravo uma tarefa de custo computacional mais próximo possível do que foi requisitado. Devido a irregularidade das regiões da imagem, determinar uma área que tenha um custo esperado exatamente igual ao desejado seria muito difícil, portanto o mestre permite uma certa margem de erro nesta escolha.

A cada requisição de tarefa recebida pelo mestre, a lista de partições ainda não processadas é percorrida, buscando a de custo computacional mais próximo daquele que foi indicado pelo escravo como ideal. A cada item da lista avaliado, três situações podem ser encontradas:

- o custo computacional da partição está próximo o suficiente do requisitado. Neste caso aquela partição é entregue ao escravo para processamento;
- o custo computacional da partição está muito abaixo do requisitado. Neste caso uma nova tarefa é construída agrupando-se iterativamente a partição encontrada com outras partições de pesos menores, até que o custo total chegue ao requisitado ou que não haja mais tarefas com custos menores;
- o custo computacional da partição está muito acima do requisitado. Neste caso a área encontrada é dividida em duas. Os pesos esperados das novas áreas são calculados e todo processo é realizado novamente sobre a lista composta por estas duas tarefas. Posteriormente novas tarefas que tiverem sido geradas por este método, mas não distribuídas para o escravo, são incorporadas a lista de partição não calculadas.

Conforme o processamento das tarefas vai terminando, os escravos requisitam novas tarefas e o mecanismo descrito acima continua sendo aplicada até que toda a imagem tenha sido calculada.

Resultados Preliminares

Para avaliar a eficiência da aplicação de RayTracing usando os mecanismos adaptativos, foram executada várias vezes a renderização de duas imagens com características em 10 máquinas heterogêneas conectadas por uma rede local. Para comparar a técnica adaptativa com outras não adaptativas, a mesma imagem foi gerada utilizando-se uma distribuição de tarefas estática, na qual cada máquina recebe uma tarefa de igual tamanho no início da execução, e de *self-scheduling*(SS) [SHA 2001], na qual o processamento total é dividido em um grande número de pedaços no início da execução e estes pedaços vão sendo distribuídos as máquinas seguindo a política *receiver-initiated*. Para cada uma das técnicas as imagens foram renderizadas 5 vezes e a média do tempo levado para realizar a operação calculado. O resultados podem ser vistos na tabela 1.

Figura	Perfil	Estático	SS(100 pedaços)	SS(10000 pedaços)
SnowMan	87.1s	163.1s	96.9s	120.4s
Transparency	1196.5s	2129.54	1500.1s	1220s

Tabela 1: Resultados Obtidos

Este exemplo mostra que a utilização de perfis adaptativos resultou em melhores resultados.

É importante ressaltar também que os melhores resultados obtidos com *self-scheduling* utilizaram uma repartição em um número de pedaços adequado para a imagem específica e para o ambiente que estava sendo usado para o processamento. Esta escolha foi feita depois de vários testes, e não poderia ser feita de uma forma automática sem um grande custo de computação. Uma das grandes vantagens que a técnica de perfil procura oferecer é um comportamento uniformemente bom com diferentes condições e diferentes instâncias do problema.

Considerações Finais

Este trabalho avaliou o emprego do ISAM na execução paralela de aplicações nas quais é encontrada uma irregularidade no custo computacional das tarefas a serem executadas.

O uso de estratégias adaptativas apresentou bons resultados no problema proposto. A combinação da criação de um *perfil* da execução com uma alocação que considera a disponibilidade atual do recurso permitiu que fosse feito um balanceamento de carga adequado a situação corrente do sistema. Com isso foi possível aumentar o grau de paralelismo gerando baixos custos de comunicação.

Espera-se que o mecanismo de perfil testado neste artigo com uma aplicação de RayTracing venha se mostrar eficiente e facilmente implementável também com várias outras aplicações de natureza irregular, se concretizando como uma solução geral para se lidar com este tipo de problema. Em um trabalho anterior [FRA 2003] experimentos iniciais com a técnica de perfil haviam sido realizados com uma aplicação de geração de fractais com bons resultados, o que fomenta esta expectativa.

Referências

- [BRU 2003] BRUSAMARELLO, L. et al. Timing verification based on floating vector simulation on a heterogeneous distributed system. In: XVIII SIM - SOUTH SYMPOSIUM ON MICROELECTRONICS, 2003. **Anais...**, 2003.
- [FEH 2003] FEHLBERG, F. W. **ISAM e seu uso em simulações numericamente intensivas**. Trabalho para conclusão do curso de Ciência da Computação. Porto alegre: Universidade Federal do Rio Grande do Sul (UFRGS), 2003.
- [FOS 99] FOSTER, I.; KESSELMAN, C. (Eds.). **The Grid: blueprint for a new computing infrastructure**. San Francisco, CA: Morgan Kaufmann, 1999.
- [FRA 2003] FRAINER, G. C. et al. Perfis Adaptativos para balanceamento de carga no ISAM. In: IV WORKSHOP EM SISTEMAS COMPUTACIONAIS DE ALTO DESEMPENHO (WSCAD 2003), 2003, São Paulo - Brazil. **Anais...**, 2003. To appear.
- [ISA 2002] ISAM. **Projeto ISAM**. Disponível em <http://www.inf.ufrgs.br/isam>. Acesso em Junho 2003, WWW.
- [REA 2002] REAL, R. A. et al. Unicluster: uma proposta de ICP para PAD. **SCIENTIA - Revista do Programa Interdisciplinar de Pós-graduação em Computação Aplicada**, São Leopoldo, RS, Brasil, v.13, n.1, p.165–176, 2002.
- [Set 2003] Seti@Home Team. **Search for extraterrestrial intelligence**. <http://setiathome.ssl.berkeley.edu/> (acesso em junho de 2003).
- [SHA 2001] SHAO, G. **Adaptive scheduling of master/worker applications on distributed computational resources**.
- [YAM 2003] YAMIN, A. C. et al. Towards merging context-aware, mobile and grid computing. **The International Journal of High Performance Computing Applications**, v.17, n.2, p.191–203, 2003.