

Uma Abordagem de Co-Escalonamento Adaptativo para Ambientes de Processamento Oportunista Multiprocessados

R.P. Mendonça e M.A.R. Dantas

Universidade Federal de Santa Catarina (UFSC)
Departamento de Informática e Estatística (INE)
Laboratório de Pesquisa em Sistemas Distribuídos (LaPeSD)
CEP 88040-900 – Florianópolis – SC
{rodrigop,mario}@inf.ufsc.br

Resumo. A popularização de configurações com múltiplos núcleos trouxe mais um desafio para a comunidade científica, visto que ainda é limitado o número de aplicações desenvolvidas buscando aproveitar o potencial dessa arquitetura. Os ambientes de processamento oportunista distribuídos também precisam se adequar a essas novas configurações, ou não se espera que seja possível utilizar toda a capacidade ociosa dos recursos. Sendo assim, este trabalho propõe uma abordagem de co-escalonamento adaptativo para um ambiente de processamento oportunista de CPUs multi-core. O mecanismo adotado utiliza múltiplas threads visando aproveitar toda a capacidade de processamento disponível nos recursos multiprocessados.

1. Introdução

Tem sido verificada uma tendência na utilização de configurações multi-core como opção para um maior desempenho de inúmeras aplicações. Todavia, essas configurações por si só não representam a resposta para muitas aplicações. Em outras palavras, o conhecimento de como melhor utilizar esses ambientes através de parâmetros de utilização e submissão são aspectos necessários para um possível melhor desempenho da aplicação. Neste sentido, este trabalho apresenta uma proposta de co-escalonamento adaptativo para um ambiente de processamento oportunista buscando garantir que toda a capacidade ociosa de seus recursos multiprocessados seja utilizada de maneira eficiente. Visto que, as aplicações executadas nesses ambientes, na sua maioria, não seguem paradigmas de programação que aproveitem tais configurações.

Várias pesquisas vêm sendo realizadas com respeito à utilização eficiente de configurações com múltiplas unidades de processamento. Uma pesquisa antiga, porém muito relevante sobre o assunto é [Cvetanovic 1987], que investiga os efeitos de particionamento, alocação e granularidade das tarefas em configurações de múltiplos processadores. Em [Pourreza e Graham 2007] foram analisados alguns aspectos da utilização de configurações multi-core e de múltiplos processadores em *clusters* processando aplicações MPI. O trabalho [El-Moursy et al. 2006] explora algumas políticas de co-escalonamento em recursos multiprocessados. Sendo assim, é observado um esforço da comunidade para garantir a utilização eficiente de máquinas com múltiplas unidades de processamento, o que para esse trabalho é dado o foco nos ambientes de processamento oportunista de recursos multiprocessados.

2. Proposta

Em [Hosken e Dantas 2004] foi proposto um sistema de processamento oportunista para a internet semelhante ao Boinc [BOINC 2007]. Esse sistema foi denominado de ATHA. O ATHA foi desenvolvido na linguagem Java, visando a portabilidade diante dos recursos computacionais heterogêneos existentes em um ambiente de processamento oportunista para a internet. Ademais, esse sistema utiliza o paradigma mestre-escravo para a execução de suas tarefas. Para isso existe um módulo mestre, onde é submetida a aplicação que será processada de maneira paralela e distribuída. Ainda no módulo mestre é informado, como argumento, o intervalo de dados que será processado pela aplicação escolhida. Uma vez submetida uma aplicação e seus argumentos, o módulo mestre é responsável por dividir o intervalo de dados recebidos em pedaços menores e iguais que serão submetidos às máquinas escravas disponíveis, sendo que na versão original do ATHA, cada máquina escrava recebe e executa apenas um pedaço do problema por vez. Quando a máquina escrava finaliza o seu processamento, ela devolve o resultado parcial para a máquina mestre, que por sua vez se encarrega de submeter um novo intervalo do problema para ser processado remotamente, isso enquanto o resultado final da aplicação não for alcançado.

Visando utilizar toda a capacidade disponível de um ambiente de processamento oportunista de recursos multiprocessados, este trabalho propõe uma estratégia de co-escalamento adaptativo quanto ao número de tarefas iniciadas nas máquinas escravas. Para isso, será utilizado o ambiente de processamento oportunista ATHA, modificando os seus módulos mestre e escravo para quando se identificar a disponibilidade de recursos com múltiplas unidades de processamento, o módulo mestre passará a submeter novos intervalos de dados para serem processados por esse recurso. Para esses intervalos, o módulo escravo irá iniciar novas instâncias da aplicação escolhida, utilizando múltiplas *threads*. Com isso, espera-se que toda a capacidade de processamento disponível no ambiente oportunista de recursos multiprocessados seja utilizada de maneira eficiente, ainda que a aplicação executada não tenha sido desenvolvida com a preocupação de aproveitar essas configurações.

Referências

- Hosken, A. and Dantas, M. A. R. (2004), “The Atha Environment: Experience With A User Friendly Environment For Opportunistic Computing”, 18th International Symposium on High Performance Computing Systems and Applications - HPCS.
- Pourreza, H., Graham P. (2007), “On the Programming Impact of Multi-Core, Multi-Processor Nodes in MPI Clusters”, 21th International Symposium on High Performance Computing Systems and Applications - HPCS.
- Cvetanovic, Zarka. (1987) “The Effects of Problem Partitioning, Allocation and Granularity on Performance of Multiple-Processor Systems”, IEEE Trans. On Computers, vol C-36, no 4, pp. 421-432.
- El-Moursy, A.; Garg, R.; Albonesi, D. H. & Dwarkadas, S. (2006) “Compatible phase co-scheduling on a CMP of multi-threaded processors”. IEEE International Parallel & Distributed Processing Symposium – IPDPS.
- BOINC (2007). <http://boinc.berkeley.edu/>, Dezembro.