Utilização dos parâmetros CPU e memória para determinar a carga computacional de um *farm* com o MONARC2

Flaviano Jacobs Luzzatto, Prof. Marcelo Trindade Rebonatto

Universidade de Passo Fundo
Campus I – Km 171 – BR 285, Bairro São José – CEP 99001-970 – Passo Fundo/RS
flavizz@universia.com.br, rebonatto@upf.br

Introdução

As grades computacionais representam uma forma de processamento integrado através da união de centros de computação (*farms*) [BAR 03]. Elas apresentam muitas adversidades em seu ambiente operacional. Estas adversidades representam um desafio como o que remete a aspectos relativos às diferentes maneiras com que os centros de computação se inter-relacionam. Nesse contexto, a utilização de simuladores objetiva aprimorar o projeto de grades. As simulações podem ser realizadas por programas como o MONARC2, desenvolvido no projeto *Models of Networked Analysis at Regional Centers* (MONARC) [MON 05]. Tal ferramenta dispõe, além de escalonador local, de escalonador distribuído de tarefas (*jobs*) para a realização das simulações. No entanto, o escalonador distribuído de tarefas do MONARC2 pode não ser fiel com relação aos resultados obtidos. A ferramenta faz uso apenas da carga de memória das unidades de processamento para realizar o escalonamento das tarefas [DRA 06].

Após a análise do mecanismo de escalonamento distribuído de tarefas do MONARC2, novas estratégias de determinação da carga computacional de um *farm* foram apontadas. Estas estratégias influenciam diretamente na escolha do destino de uma tarefa no ato de sua migração e, consequentemente, numa aproximação maior da realidade efetiva da grade. A estratégia que considera a carga dos processadores dos *farms* foi explorada em diferentes aspectos. A mesma foi testada por simulações compostas de tarefas contendo variações na questão da exigência de memória e CPU. Em adição, a estratégia que considera a carga dos processadores também foi executada em ambientes de grades com diferentes configurações de *hardware*. Os resultados do uso da estratégia considerando carga ocupacional de CPU são analisados e comparados com o método de escalonamento padrão (carga de memória).

O escalonador distribuído de tarefas do MONARC2

O escalonador distribuído do MONARC baseia-se em funções para especificar que recursos computacionais serão utilizados em determinada operação de submissão de *jobs* e destina-los ao *farm* apropriado, ou seja, com menor carga computacional no momento da migração. O cálculo da carga computacional dos *farms* é realizado levando em consideração a utilização de memória das suas unidades processadoras. Além disso, é considerada também a quantidade de memória sendo requisitada pelos *jobs* que já se encontram na fila local de execução [MON 05]. O problema encontrado é que tal cálculo não reflete fielmente a realidade do estado de recursos dos *farms*.

Modificações no escalonador distribuído de tarefas

As estratégias de escalonamento original (memória) e modificada (CPU) foram executadas sendo extraído o valor de tempo de simulação em todas as repetições. Das repetições (30 para cada estratégia em cada um dos 64 ambientes computacionais originados), obteve-se a média aritmética em cada caso, valor usado na análise de variância, método indicado para este caso [PAS 06]. As análises foram comparadas com o teste de Kruskall Wallis para certificar os resultados [PAS 06].

A Tabela 1 ilustra uma sumarização dos resultados, indicando o número de diferenças significativas encontradas analisando a estratégia de CPU em relação à estratégia original de escalonamento.

Tabela 1 – Memória X CPU

Média tempo simulação (ms)	Diferença significativa		Total
	Com	Sem	
Acima	5	7	12
Abaixo	40	12	52
Total	45	19	64

Na linha definida como "abaixo", dos 52 casos (81,25%) com estratégia utilizando carga de processadores com média de tempo (em milisegundos) abaixo da estratégia original, 40 casos (62,5%) produziram diferença significativa e 12 não. Na linha "acima", ou seja, onde a nova estratégia obteve tempo médio de simulação acima do tempo médio da estratégia original, apenas 12 casos (18,75%) ocorreram. Destes, 5 casos com diferença significativa e 7 casos sem.

Considerações finais

Ao considerar outros fatores no método de escalonamento espera-se, também, obter resultados melhores dependendo da característica dos *jobs* que transitam pelos ambientes das grades. Para que o simulador possa desempenhar um papel próximo da realidade do ambiente sendo simulado o ideal é que o seu escalonador distribuído possua um mecanismo de escalonamento que trate maiores quantidades de tipos possíveis de tarefas em prol de um melhor desempenho em tais situações.

Referências

[BAR 02] BARRETO, M. E. Estudo sobre computação baseada em clusters e grid. Exame de qualificação PPGC-UFRGS. Porto Alegre – RS, 2002.

[DRA 06] DRAGOS, A. MONARC2: Simulation of Distributed Systems. Techniques of Performance Improvement. Test Cases. MonAlisa Projects. MONARC2 Diploma Thesis 2004. Califórnia Institute of Tecnology – Caltech. Pasadena. California – US.

[MON 05] MONARC. Models of Networked Analysis at Regional Centres. Disponível em: <http://monarc.web.cern.ch/MONARC/> Acesso em: Agosto de 2005.

[PAS 06] PASQUALOTTI, A. Análise Estatística, 2006. (entrevista realizada na Divisão de Avaliação Institucional da UPF, no dia 16 de junho de 2006).