

Métricas para avaliação de desempenho computacional em sistemas produtivos

Josilene Aires Moreira (UFPb- CFT) josileneairesmoreira@hotmail.com
Ricardo Moreira da Silva (UFPb- CFT- PPGE) ricardomoreira0203@hotmail.com

Resumo:

A performance de sistemas produtivos é um dado muito importante para efeito de programação e controle, quer para avaliação, quer para planejamento da produção. Empresas operando com sistemas informatizados não são mais tão incomuns como em tempos de outrora. Fábricas de automotivos são um bom exemplo de empresas onde a robótica já tem papel decisivo no controle de qualidade de produção, de modo que a quantidade de dados a serem processados em suas linhas de fabricação cresce de maneira assustadora e os sistemas de informações tornam-se por demais complexos. Nesse ambiente, um sistema computacional trabalhando de maneira paralela (adquirindo forma de um supercomputador) se torna necessária a fim de obter resultados mais rápidos. Essa expansão da capacidade de processamento cria quatro problemas: (1) necessita-se de métricas para avaliação dos sistemas produtivos que produzam informações para rápidas tomadas de decisão, (2) necessita-se avaliar o resultados por diversas métricas, (3) o próprio sistema computacional deve ser avaliado e (4) as métricas devem espelhar os itens avaliados sem serem tendenciosas. Esse artigo é um estudo de caso, onde se apresentam algumas métricas de avaliação de performance e mostra-se um experimento onde foi simulado um pesado sistema de dados, equivalendo a um sistema de produção sendo monitorado com supercomputadores.

Palavras Chaves: Métricas, Performance, Controle de Sistemas Produtivos

1. Introdução

A performance de sistemas produtivos avaliados por supercomputadores depende basicamente das características da *workload* (conjunto de programas que representa os dados recebidos pelo sistema) que está sendo processada e do método de escalonamento utilizado para gerenciar a fila de execução dos programas (Feitelson, 2001). Ao pretender medir o desempenho de um sistema com supercomputadores, é necessário escolher uma métrica de performance que possa representá-lo sem introduzir distorções, sem influenciar os resultados e sem favorecer um determinado escalonador do computador.

No entanto, para essas avaliações de desempenho vêm utilizando um conjunto de métricas que têm demonstrado resultados conflitantes (Feitelson, 1998). Através de uma métrica obtém-se que um determinado escalonador tem melhor desempenho, enquanto que ao empregar-se outra métrica é possível concluir que o mesmo escalonador obteve um desempenho pior.

Ora, desde que o objetivo de uma avaliação de performance é chegar a conclusões sobre o comportamento de um determinado sistema, ou mesmo comparar dois ou mais sistemas quando sujeitos à restrições semelhantes, a métrica deveria ser um fator que contribuisse para se chegar a conclusões, e não que influenciasse nos resultados. Não é do interesse do

avaliador preocupar-se sobre o quanto a métrica interfere no processo de avaliação, mas sim utilizar uma medida que expresse independentemente os resultados.

Assim, realizou-se a avaliação comparativa de dois métodos de escalonamento tradicionalmente utilizados em sistemas produtivos com controles por supercomputadores paralelos, usando algumas métricas conhecidas para expressar os resultados e propondo a idéia de uma nova maneira de comparar desempenho denominada Performance Relativa.

A simulação prática utiliza duas variações de um mesmo método de escalonamento: o *Conservative Backfilling* e o *Easy Backfilling*. Estes dois métodos são avaliados com o objetivo de analisar os resultados obtidos com as diversas métricas e de mostrar que a Performance Relativa não sofre interferência da *workload*, assim como possibilita uma boa visualização comparativa dos mesmos. Para tanto são utilizados *traces* (dados reais) dos centros de supercomputadores KTH, SDSC e CTC, como também uma *workload* sintética ou seja uma simulação.

2. O que são escalonadores

Em um ambiente de sistemas produtivos, usando processamento paralelo, um programa deve explicitar o grau de paralelismo(n) e o tempo estimado(tr) necessários à sua execução(Downey, 1999). A relação entre o número de processadores e o tempo estimado é conhecida como forma ou área do programa. Os dados coletados do sistema produtivo através de sensores, representam para a máquina “programas”. Acontece que um programa que chega, pode não encontrar recursos disponíveis para ser executado e então é colocado em uma fila, a qual é controlada pelo escalonador do supercomputador. O escalonador tem a função de receber as requisições dos programas e decidir quando estes iniciarão a sua execução e que processadores usarão. Dependendo da política de escalonamento escolhida, os sistemas produtivos podem apresentar variações de desempenho, não reais (Cirne, 2001). Vamos apresentar os dois métodos de escalonamento avaliados no contexto deste trabalho:

Conservative Backfilling - É uma otimização do método FCFS (First Come First Served), onde os programas que chegam para executar são colocados em uma fila e são alocados na ordem de chegada. Como os programas apresentam formas diferentes, à medida que são alocados para execução vão deixando “buracos”. Os algoritmos de escalonamento tentam reduzir ao mínimo, esta subutilização de recursos através da alocação de outros programas da fila capazes de preenchê-los. No algoritmo do *Conservative Backfilling*, quando o escalonador encontra o primeiro programa que não dispõe de recursos para ser executado imediatamente, percorre a fila procurando um programa que possa ser executado com os recursos que estão disponíveis no momento. Este programa pode ser adiantado na fila, desde que não atrase a execução de nenhum outro ocupante da fila de espera. Deste modo, o algoritmo garante previsibilidade, pois no momento da submissão já é possível prever quando um determinado programa irá executar.

Easy Backfilling - O algoritmo de Easy Backfilling também percorre a fila no momento em que um programa não pode ser executado, a fim de encaixar um outro que aproveite os recursos disponíveis. A diferença é que este método é mais agressivo, pois permite adiantar qualquer programa da fila de espera desde que isto não cause atraso ao primeiro programa da fila. Este algoritmo ganha em agressividade, propondo uma maior utilização do sistema, mas perde em previsibilidade, pois é impossível saber quando exatamente um programa será executado ou quanto atraso ele irá sofrer no momento da sua submissão.

3. Métricas de Performance

Entendido como funcionam os escalonadores mais utilizados nos sistemas produtivos interligados por um sistema computacional, são necessárias métricas para medir efetivamente o desempenho. As métricas são os métodos de sumarização utilizados para representar o desempenho de um sistema. Existem algumas métricas comumente usadas para representar sistemas, onde, a seguir veremos a descrição de algumas delas e a apresentação da métrica denominada Performance Relativa, proposta por Moreira(2003).

3.1. Turn-around time médio

O *turn-around time* (tt) ou tempo de resposta é definido como o intervalo entre o instante em que o programa é submetido ao sistema até o momento em que toda a sua saída é produzida e o programa termina. É uma métrica interessante por representar a performance do ponto de vista do usuário, que submete o seu programa e espera algum tempo até que ele execute (Cirne,2001). Por outro lado, a média possui a característica de ser influenciada por valores extremos(Costa Neto,1977). Por exemplo, uma *workload* que possua a maioria dos programas com *turn-around time* curto e alguns poucos programas com *turn-around time* muito longo, certamente terá sua média dos *turn-around time* afetada por estes valores muito altos, o que elevará a média geral.

3.2. Slowdown médio

O *slowdown*, também conhecido como fator de expansão, é o *turn-around time*(tt) normalizado pelo tempo de execução(te), $s = tt / te$. O *slowdown* é utilizado a fim de reduzir valores extremos associados a programas muito longos. A idéia principal por trás do *slowdown* é que um programa espere na fila um tempo proporcional ao seu tempo de execução. Por outro lado, o *slowdown* supervaloriza a importância dos programas muito curtos.

3.3. Bounded slowdown médio

O *bounded slowdown* procura atenuar o *slowdown* através da definição de um limite para o tempo de execução $bs = tt / \max(te, \tau)$. A diferença é que, para programas muito pequenos, o *slowdown* será limitado pelo valor limite (τ) escolhido, e não pelo tempo de execução. O comportamento desta métrica vai depender do valor de τ . A utilização tanto do *slowdown* como do *bounded slowdown* levam a novos problemas, pois podem encorajar o sistema a fazer com que os programas tenham um tempo de execução maior, a fim de baixar os valores destas métricas.

3.4. Média geométrica do tempo de resposta

A média geométrica é definida por $medgeom(x_1, \dots, x_n) = \sqrt[n]{x_1 * \dots * x_n}$, e tem a característica de ser menos influenciada por valores extremos (Cirne,2001). A idéia de usar média geométrica ao invés da média aritmética tem o objetivo de reduzir o efeito dos programas com *turn-around time* muito longo ou muito curto sobre a média geral dos *turn-around time* da *workload*. Esta métrica é usada para agregar o tempo de execução dos programas que compõem o *Spec Benchmark* (2002).

3.5. Performance relativa

A performance relativa toma como base a relação entre dois valores de *turn-around time* que se deseja comparar. A metodologia utilizada é a seguinte:

1. Define-se quais supercomputadores ou sistemas deseja-se comparar, dois a dois (chamemos de **A** e **B**);
2. Toma-se uma determinada *workload* de tamanho **n** a ser submetida aos dois escalonadores;
3. Registra-se o *turn-around time* de cada programa obtido em cada escalonador (**tt_a** e **tt_b**);
4. Calcula-se a razão entre os **n** pares dos *turn-around times*, (**tt_a / tt_b**);
5. Calcula-se a distribuição acumulada de frequência das razões obtidas, a fim de obter a representação gráfica.
6. Para cada razão **i** interpreta-se o resultado da seguinte maneira :
 - a) Se **tt_a / tt_b = 1**, o programa teve o mesmo *turn-around time* nos dois escalonadores;
 - b) Se **tt_a / tt_b < 1** o programa obteve melhor *turn-around time* no escalonador **A**;
 - c) Se **tt_a / tt_b > 1** o programa obteve melhor *turn-around time* no escalonador **B**.

FONTE: MOREIRA(2003)

Quadro1 –Metodologia da Performance

Relativa

Vejamos o resultado no gráfico 1 abaixo, através da distribuição acumulada de frequência.

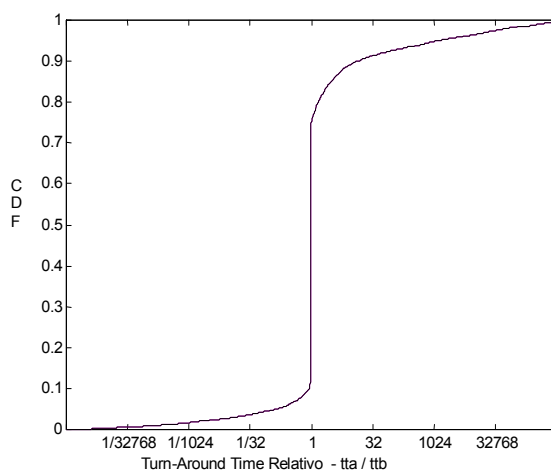


Gráfico 1 - Performance relativa

Através do gráfico vemos que aproximadamente 62 % dos *turn-around times* tiveram o mesmo valor nos dois escalonadores; o escalonador A foi melhor em, aproximadamente, 12% dos *turn-around times* e o escalonador B foi melhor em aproximadamente 16% dos *turn-around times*.

Apesar de ter o conceito matemático muito simples, o uso desta métrica é interessante porque, ao contrário das métricas baseadas na média aritmética, esta não sofre influência dos valores extremos e consegue retratar dado a dado, sinal a sinal capturado do sistema avaliado, onde se tem performance melhor. No caso específico do experimento demonstrado, as características da *workload*, tais como ser dominada por programas muito longos ou ainda possuir valores extremos que afetem a média aritmética dos *turn-around times* não interferem no resultado desta métrica, desde que a comparação é feita sinal a sinal.

Este mecanismo de avaliação permite que se conheça exatamente em quanto um algoritmo usado no sistema computacional que controla o sistema produtivo foi melhor do que o outro, além de permitir e identificar para quais programas, ou seja, onde, em qual etapa do sistema de produção, o parâmetro escolhido foi melhor. A partir destas informações é possível isolar o conjunto de sinais, que representam, por exemplo máquinas do sistema produtivo, onde cada algoritmo foi melhor, a fim de identificar e estudar suas características. Além disso, esta métrica permite visualizar graficamente, de forma clara, o resultado comparativo entre os

escalonadores. Como desvantagem, podemos citar que para comparar mais de dois escalonadores é necessário fazê-lo em pares, obtendo sempre a relação de um turn-around time para com o outro.

4. Experimento

Foi utilizada uma abordagem baseada na simulação dos métodos de escalonamento *Conservative Backfilling* e *Easy Backfilling*, a fim de medir a performance destes algoritmos com as diversas métricas. Foram usados três *logs* de *workloads* provenientes de diferentes centros de supercomputadores disponíveis na *web* (Feitelson(2001) e Hazlewood(2001)) e ainda uma quarta *workload* sintética obtida através de um modelo de carga.

O simulador utilizado recebe como entrada uma *workload* contendo a descrição dos programas com informações sobre a quantidades de *nodes(n)*, o *request time(tr)*, o tempo de submissão, o tempo de início e o tempo de término de cada um; a outra entrada é o método de escalonamento e a terceira entrada é a descrição do supercomputador, incluindo o n° de nodes. O processamento destes programas é simulado através dos dois métodos de escalonamento citados, obtendo-se os *turn-around times* como resultado. Estes *turn-around times* são então comparados através das métricas : Turn-around time médio (tt médio); Slowdown médio; Bounded slowdown médio com $\tau = 10$; Bounded slowdown médio com $\tau = 100$; Média geométrica e Performance relativa.

As características das *workloads* estão resumidas no quadro abaixo :

Nome	Máquina	processadores	Programas	período
CTC	Cornell Theory Center	430	61.021	Jul/96 a Mai/97
KTH	Swedish Royal Institute of Technology	100	28.474	Set/96 a Ago/97
SDSC	San Diego Super Computer Center	128	12.631	Jan/99 a Mai/99
SYNT	Modelo Sintético	800	46.510	-

FONTE: MOREIRA(2003)

Quadro 2 – Descrição das Workloads

Resultados da simulação – O quadro abaixo mostra os resultados de performance obtidos para cada método de escalonamento avaliado segundo as diversas métricas.

dados	Tt médio		média geom tt		Slowdown médio		Bounded slowdown		Slowdown médio	
	EAS Y	CON S	EAS Y	CON S	EAS Y	CON S	EAS Y	CON S	EAS Y	CON S
CTC	11755	13409	1846	2323	5.32	13.97	5.32	13.97	4.36	8.60
KTH	14893	16202	1628	2133	158.1	209.3	79.44	92.26	20.55	20.82
SDSC	20158	20008	1474	1738	59.50	61.46	59.50	61.46	43.60	38.28
SYNT	10321	10695	1483	1561	67.99	42.52	25.36	16.47	7.65	5.64

FONTE: Experimento

Quadro 3 – Resultados do experimento

Os valores do quadro 3 estão expressos em segundos, com as colunas **easy** mostrando o valor obtido para o *Easy Backfilling* e as colunas **cons** trazendo o valor obtido para o *Conservative Backfilling*. Analisando-se os resultados comparativamente temos: Através da métrica média geométrica, Easy obteve melhor performance em todas as *workloads*. Através da métrica *turn-around time* médio, o método *Conservative* foi melhor para a *workload* SDSC enquanto *Easy* foi melhor para as outras cargas de dados. Através da métrica *slowdown* médio, o *Conservative* foi melhor para a *workload* sintética, enquanto *Easy* foi melhor para as outras cargas. Através do *bounded slowdown* médio com $\tau=10$, *Conservative* foi melhor para a

workload sintética, enquanto *Easy* foi melhor para as outras cargas de dados. Através do *bounded slowdown médio* com $\tau=100$, *Conservative* foi melhor para a *workload* sintética e para SDSC, enquanto *Easy* foi melhor para CTC e KTH.

Dessa forma, os resultados de performance apresentaram resultados conflitantes dependendo da métrica utilizada, o que já era esperado (Feitelson, 2002). A fim de esclarecer melhor estes conflitos, foi utilizada a métrica de performance relativa para avaliar o desempenho dos escalonadores, obtendo os resultados mostrados a seguir.

Performance Relativa			
Dados	EASY	CONS	IGUAL
CTC	25.66 %	12.01 %	62.31 %
KTH	35.22 %	18.58 %	46.18 %
SDSC	33.16 %	23.61 %	43.21 %
SYNT	35.22 %	18.58 %	50.05 %

FONTE: Experimento **Quadro 4 – Resultados da performance relativa**

Os valores do quadro 4 foram calculados através da razão do **tt conservative / tt easy** de cada programa. Os valores das colunas **easy** e **cons** representam o percentual onde cada método foi melhor. A coluna IGUAL representa o percentual em que os programa obtiveram o mesmo *turn-around time* nos dois métodos de escalonamento.

Muito embora o experimento demonstrou a performance de todas as cargas usando-se a performance relativa, apenas o gráfico da carga sintética foi aqui apresentado. Esse foi obtido calculando-se a distribuição acumulada de frequência dos valores calculados. A porção da

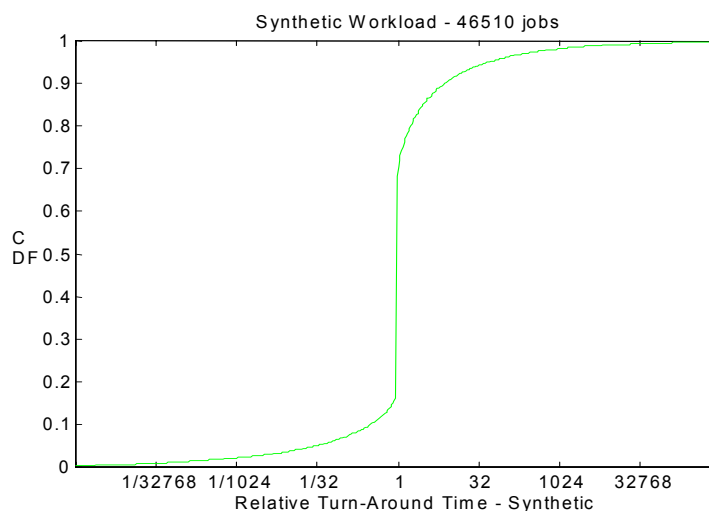


Gráfico 2 - Performance relativa - SYNT

curva à esquerda de 1 representa onde o método *Conservative Backfilling* foi melhor; a porção da curva à direita de 1 indica onde o método *Easy Backfilling* foi melhor. Percebeu-se que para todas as *workloads* o método *Easy Backfilling* foi melhor.

5- Considerações finais

Os sistemas de informações das empresas que utilizam a robótica avançada crescem, em número de parâmetros a serem controlados de maneira exponencial. Sistemas de computação

simples já começam a dar sinais que precisam migrar para sistemas baseados em supercomputadores paralelos distribuídos. Isso requer medições de performances tanto nos sistemas como na própria máquina. Aqui foi apresentada alguma dessas métricas, sendo que, diante dos experimentos de Moreira(2003) pode-se afirmar, que as métricas baseadas em estatísticas sumarizadoras, em ambiente complexo, podem revelar resultados conflitantes e atrapalhar o processo decisório. Deste modo o acompanhamento passo a passo das interações proporcionado pela Performance Relativa faz com que essa métrica seja a mais adequada para avaliar dois sistemas complexos a serem medidos, inclusive a máquina que controlará o sistema produtivo.

6. Referências Bibliográficas

ALLEN D, *The elusive goal of workload characterization*. Perf. Eval. Rev. 26(4), pp. 14-29, March 1999

CIRNE Walfredo, *Using Moldability to Improve the Performance of Supercomputer Jobs*. Tese de doutorado. University of California, San Diego. 2001.

COSTA NETO, Pedro L. O. *Estatística*. Editora Edgard Blücher Ltda, 1977

D. FEITELSON. *Analyzing the Root Causes of Performance Evaluation Results*. Technical Report 2002, School of Computer Science and Engineering, The Hebrew University of Jerusalem, Mar 2002. <http://www.cs.huji.ac.il/~feit/papers/Root02TR.ps.gz>

D. FEITELSON. *The Effect of Metrics and Workloads on Performance Evaluation*. Technical Report 2001, The Hebrew University of Jerusalem, Oct 2001. <http://www.cs.huji.ac.il/~feit/papers/design.ps.gz>

D. FEITELSON. *Metrics for Parallel Job Scheduling and Their Convergence*. In Job Scheduling Strategies for Parallel Processing. and L. Rudolph (Eds.), pp. 188-206, Springer-Verlag, 2001. Lecture Notes in Computer Science Vol. 2221. ©Copyright2001 by Springer-Verlag. <http://link.springer.de/link/service/series/0558/bibs/2221/22210188.htm>

D. FEITELSON and L. Rudolph. *Metrics and Benchmarking for Parallel Job Scheduling*. In Job Scheduling Strategies for Parallel Processing, Dror Feitelson and Larry Rudolph (Eds.), pp. 1-24, Springer-Verlag, Lecture Notes in Computer Science vol. 1459, 1998. <http://www.cs.huji.ac.il/~feit/parsched/p-98-1.ps.gz>

DOWNEY. *A Parallel Workload Model and its Implications for Processor Allocation*. In 6th Intl. Symp. High Perf. Distributed Computing, Aug 1999. <http://www.sdsc.edu/~downey/allocation/>

DOWNEY. *Predicting queue times on space-sharing parallel computers*. 11th International Parallel Processing Symposium (IPPS'97), Geneva, Switzerland, Apr 1997. <http://www.sdsc.edu/~downey/predicting/>

HAZLEWOOD. *NPACI JobLog Repository*. <http://joblog.npaci.edu/> acesso em 12/05/2003

MOREIRA, Josilene Aires. *O Impacto da Métrica e do Escalonador sobre a Performance dos Supercomputadores Paralelos*. Dissertação de mestrado. UFCG / DSC. Campina Grande PB. 2003.

Platform Computing Corp. *Load Sharing Facility Web Page*. <http://www.platform.com/products/wm/LSF/index.asp>

Standard Performance Evaluation Corporation. *The SPEC web page*. <http://www.spec.org/>