Política de escalonamento baseada na conexão para Servidores Web

Cristina Duarte Murta[1] Tarcísio Paulo Corlassoli[2]
[1] Departamento de Informática Universidade Federal do Paraná
Curitiba - Paraná - Brasil
{cristina@inf.ufpr.br}
[2] Departamento de Informática Centro Federal de Educação Tecnológica do Paraná
Via do conhecimento, Pato Branco, Paraná, Brasil
{tpc@pb.cefetpr.br}

Resumo—
O presente trabalho propõe uma nova política de escalonamento para o processamento de requisições HTTP em servidores Web. Esta nova política chama-se FCF (*Fastest Connection First*) pois atribui prioridades às requisições HTTP baseando-se no tamanho do arquivo solicitado e na velocidade da conexão com o usuário. As requisições para arquivos menores feitas através de conexões mais rápidas recebem maior prioridade. A nova política foi comparada através de simulação com as políticas de uso corrente em servidores Web. Os resultados apresentam evidências de que as diferenças de conectividade observadas na Web afetam o desempenho do servidor, e que essa informação pode ser utilizada para melhorar significativamente o desempenho do sistema.

Palavras-chave—**Política de Escalonamento, Servidor Web, Internet.**

Abstract—
This paper presents a novel scheduling policy for HTTP requests processing in Web servers. This policy, called FCF (Fastest Connection First), gives priority to HTTP requests based on the size of the requested file and on the speed of the user's connection. The requests for smaller files through faster connections receive the highest priorities. This policy was compared with the standard policies. The results show that the different levels of connectivity in the Internet affect the Web server performance. The server system will get better performance metrics taking into account this information.

Keywords— **Scheduling policy, Web server, Internet**

I. INTRODUÇÃO

Servidores Web tratam inúmeras requisições simultaneamente, oriundas de um imenso e heterogêneo universo de usuários. A diversidade de usuários diz respeito não apenas a cultura, idioma e localização geográfica, mas também a plataformas de hardware, software e, principalmente, a condições de conectividade. Pesquisa realizada pelo *Graphics, Visualization, & Usability Center do Georgia Institute of Technology* [TEN 98] mostrou que, em 1998, as condições de conectividade variavam até 600 vezes, por exemplo, de um modem de 14,4 Kbps a uma conexão de 10 Mbps. Além da largura de banda, fatores dinâmicos como a quantidade de tráfego no *link*, o número de *hops* no caminho de uma conexão TCP, o atraso, a taxa de transmissão dos *links* intermediários e características das diversas implementações do protocolo TCP contribuem para a variabilidade das condições de conectividade observadas na Internet.

Outra característica muito variável na Web é o tamanho dos arquivos transferidos nas requisições HTTP, normalmente modelados por distribuições de cauda pesada [BAR 98, MUR 99, BAN 01]. Para requisições HTTP estáticas, que são a maioria [BAR 98], o tempo de processamento pode ser considerado proporcional à quantidade de bytes servidos. Considerando, portanto, a grande variabilidade dos tempos de processamento, a política mais efetiva para o tempo médio de resposta é a *SRPT (Shortest Remaining Processing Time)* [HAR 98, CON 67]. Esta política foi avaliada para servidores Web, tanto através de análise, quanto de simulação e implementação. Em todos os casos a política mostrou bom desempenho.

A utilização de uma política que prioriza as requisições para pequenos arquivos reduz, drasticamente, o tempo médio de resposta do sistema. Em servidores Web nos quais a política de escalonamento não considera o tamanho da resposta, as requisições pequenas esperam pelo atendimento de requisições muito grandes, aumentando excessivamente o tempo médio de resposta. Esse fato é agravado pela distribuição do tamanho dos arquivos requisitados. Como já observado, diversos estudos comprovaram que o tamanho dos arquivos solicitados em requisições Web segue distribuição de cauda pesada. Isto significa que uma pequena fração dos arquivos (os maiores) corresponde à maior parte da carga imposta ao servidor. Em cargas de serviço, que seguem este tipo de distribuição, o uso de uma política de escalonamento que prioriza arquivos pequenos, reduz o tempo médio de resposta e não prejudica excessivamente o processamento dos arquivos grandes. Desta forma, requisições para arquivos grandes possuem à

disposição a maior parte dos recursos do sistema, o que evita a *starvation* destas requisições.

O objetivo deste trabalho é propor uma política de escalonamento que reconheça e trate as duas características apontadas: a diversidade das condições de conectividade dos usuários na Internet e a variabilidade dos tamanhos dos arquivos transferidos. A proposta é de uma política de escalonamento que dê maior prioridade às requisições dos usuários que possuem conexões mais rápidas (maior banda, menor latência, menos congestionamento) bem como, para as requisições de arquivos pequenos.

Intuitivamente, uma política de escalonamento, que prioriza requisições originárias de conexões mais rápidas, pode reduzir o tempo de serviço destas requisições, com conseqüente diminuição do tempo de duração das respectivas conexões. O encerramento de uma conexão indica que os recursos do servidor estão disponíveis para o atendimento de novas requisições, o que gera um aumento da taxa de serviço. Além disso, dar prioridade para usuários com melhor conectividade, auxilia o servidor a manter um tempo médio de resposta condizente com as expectativas dos usuários. Usuários que possuem conexões mais velozes esperam respostas rápidas. Quando as requisições são atendidas por servidores sobrecarregados, o tempo de serviço pode ser perceptível por um usuário que possui conexão rápida. Para usuários com conexões mais lentas o tempo devido à espera no servidor é muito menos perceptível.

Embora exista uma variação muito grande das condições de conectividade na Internet, os trabalhos da literatura atual sobre políticas de escalonamento em servidores Web, não levam em consideração o comportamento da Internet. No entanto, as variações da Internet exercem grande influência no desempenho e no comportamento do mesmo [ALM 97, DIL 97].

Este trabalho está organizado como se segue. A seção seguinte apresenta os trabalhos relacionados. A terceira seção descreve a política proposta. A quarta seção descreve a metodologia utilizada para avaliação de desempenho da política e a quinta seção apresenta os resultados. As conclusões estão na última seção.

II. Trabalhos Relacionados

O tamanho do processo gerado por uma requisição HTTP estática pode ser inferido a partir do tamanho do arquivo requisitado. Assim, antes da requisição ser atendida, é possível prever o tamanho do processo. Os tamanhos dos arquivos requisitados a um servidor Web são caracterizados por distribuição de cauda pesada [BAR 98].

Considerar o tamanho de um processo na política de escalonamento de um servidor, tem efeitos positivos sobre o tempo médio de resposta. SRPT (*Shortest Remaining Processing Time)* é conhecida como a disciplina de serviço que minimiza o tempo médio de resposta para qualquer distribuição do tempo entre chegadas e da taxa de serviço [CON 67]. O principal argumento contra o uso dessa política é a possibilidade de *starvation* dos processos grandes. Esta é uma preocupação legítima para várias distribuições dos tamanhos das tarefas mas não para o caso das distribuições de cauda pesada. O artigo [BAN 01] mostra analiticamente que, quando a carga de serviço segue este tipo de distribuição, a disciplina SRPT apresenta menor tempo de resposta do que a disciplina PS (*processor sharing*) para 99% dos jobs, em qualquer situação de carga.

Comparações entre SRPT e PS no contexto específico dos servidores Web, feitas através de análise, simulação e experimentação, são apresentadas nos trabalhos [BAN 01, HAR 01]. Em todos os casos os resultados são bastante favoráveis à disciplina SRPT. Contudo, deve-se ressaltar que o experimento foi realizado em uma rede local, e não em condições reais de conectividade da Internet. Avaliar um servidor Web em uma rede local pode ser muito diferente de avaliá-lo nas condições de conectividade da Internet, pois numa rede local a latência e muito menor. Além disso, a banda disponível em uma rede local é normalmente muito maior e mais homogênea do que a banda observada na Internet. Para responder a perguntas do tipo "uma requisição de um arquivo pequeno feita por um usuário com conexão lenta deve ter maior ou menor prioridade do que a mesma requisição feita através de uma conexão rápida?" deve-se considerar as variações de conectividade na Internet.

Uma política de escalonamento para servidores Web baseada no tempo necessário para transferir a requisição foi proposta em [CRO 99]. O tempo de transferência foi definido em função do tamanho do arquivo requisitado. Os testes também foram realizados em uma rede local e os resultados mostraram redução no tempo médio de resposta para um quarto do tempo obtido pelo servidor Apache.

A latência da rede tem grande impacto sobre o tempo de resposta de um servidor Web, de acordo com [DIL 97]. Este artigo apresenta testes com cargas de trabalho em ambientes que simulam uma Intranet, praticamente sem latência de rede, e a Internet com suas variações de latência. O problema das grandes latências é que o servidor precisa manter um processo ocupado com a requisição até que o último pacote de dados seja enviado ao usuário.

Esses resultados reforçam a idéia da política proposta, no sentido de dar prioridade às requisições de conexões que tenham menor latência na rede. Ao permitir que estas requisições sejam concluídas mais rapidamente, reduz-se o impacto que a latência da Internet provoca no desempenho do servidor.

No artigo [CHE 98] também é proposta uma política de escalonamento baseada no tamanho da requisição. Para se evitar *starvation* das requisições grandes o autor propôs que o cálculo da prioridade fosse baseado em três parâmetros, a saber: tamanho da requisição, tempo de chegada da

requisição no sistema e uma constante para controlar o balanceamento entre minimização do tempo de resposta e justiça. Bons resultados foram obtidos utilizando-se esta forma de cálculo. Porém, de acordo com os trabalhos [CRO 99, BAN 01, HAR 01] e considerando as características da carga Web, a prioridade poderia ter sido definida apenas com base no tamanho da requisição. Muitas das premissas utilizadas no trabalho [CHE 98] foram também utilizadas nos trabalhos [HAR 98, CRO 99, BAN 01, HAR 01] com exceção da necessidade de evitar *starvation* através do tempo de chegada da requisição.

Não foram localizados trabalhos na bibliografia atual que levem em consideração o desempenho da Internet para definir prioridades no processamento das requisições.

III. A POLÍTICA FASTEST CONNECTION FIRST (FCF)

Um servidor Web geralmente atende a várias requisições concorrentemente. Alguns servidores implementam essa multiprogramação criando um novo processo ou *thread* para cada nova conexão. Dependendo da implementação, o custo para criar um novo processo pode ser alto, gerando um *overhead* inaceitável. Outros servidores, na tentativa de minimizar o *overhead*, implementam um mecanismo conhecido como *pool* de processos, onde um determinado número de processos são criados durante a inicialização do servidor [MUR 96]. Seja qual for o caso, o servidor não faz uso de prioridades para o tratamento das requisições [ALM 98, CHE 98].

O servidor Web Apache, por exemplo, cria um *pool* de processos e admite as requisições de acordo com a ordem de chegada (FIFO), não importando seu tipo ou tamanho. As requisições admitidas no *pool* são atendidas, concorrentemente, pela política PS (*processor sharing*). Assim, a escolha do processo que pode utilizar um recurso do sistema (processador, disco, interface de rede) é feita pela ordem de chegada da requisição.

A política FCF muda a ordem na qual as requisições ou processos recebem recursos do sistema. O próximo processo a receber recursos será aquele que estiver atendendo a requisição de maior prioridade. Esta prioridade será definida em função de dois parâmetros: menor quantidade de bytes a processar e maior taxa estimada de transmissão.

O primeiro parâmetro é utilizado para dar prioridade ao menor processo. O segundo parâmetro visa dar prioridade à requisição que pode ser transmitida no menor tempo possível. A taxa de transferência depende das condições de conectividade do usuário e das variações dinâmicas da Internet. Na seção IV descrevemos um modelo de simulação das condições de conectividade da Internet que leva em consideração os dois parâmetros.

A política FCF opera da seguinte forma. Primeiramente busca-se na fila de requisições aquela que tiver o menor número de bytes a processar. A seguir, outra pesquisa é realizada na mesma fila, buscando-se a requisição em processamento que tem a maior taxa de transmissão estimada. Esta nova pesquisa, porém, é restrita às requisições que possuírem um número de bytes a processar menor ou igual ao produto de uma constante (beta) pelo menor número de bytes encontrado na primeira pesquisa. Beta define a abrangência da faixa de pesquisa. Ao variar a constante beta damos maior ou menor peso para a taxa de transmissão e para o tamanho do processo. A taxa de transmissão é calculada dividindo-se o tempo estimado para transferir o arquivo pelo tamanho do arquivo. Quanto maior for esta taxa, maior será a prioridade da requisição. O processo para estimar o tempo de transferência é apresentado na sessão IV.

O objetivo da política é considerar as condições de conectividade do usuário e todos os aspectos inerentes à transmissão de dados pela Internet. Conexões mais velozes tendem a ter RTT menores e janelas de receptor e de congestionamento maiores [ACH 96]. Conseqüentemente, as requisições feitas através destas conexões, se atendidas com maior prioridade no servidor, podem ser concluídas mais rapidamente, liberando os recursos do servidor para o atendimento de novas requisições.

Requisições Web são servidas através de uma conexão TCP. Neste protocolo cada pacote ou conjunto de pacotes de dados, enviado pelo remetente, deve ser confirmado pelo destinatário. Devido ao controle de congestionamento realizado pelo TCP (*slow start*), apenas um número limitado de pacotes pode ser enviado a cada vez, devendo os demais aguardar confirmação da chegada dos pacotes já enviados. A política FCF visa garantir que os *buffer*s das conexões velozes sempre possuam dados para enviar quando as confirmações chegarem. Ou seja, os pacotes já estarão disponíveis para o envio quando puderem ser enviados. A política FCF procura tratar a requisição no servidor de acordo com o nível de conectividade medido dinamicamente para a respectiva conexão. Se a conexão é rápida, a requisição também será tratada rapidamente no servidor.

A política FCF foi avaliada através de uma simulação na qual a taxa de transferência dos arquivos foi calculada simulando-se também a Internet. Em uma situação real, a estimativa da taxa de transmissão (*bandwidth* da conexão), deve ser feita utilizando-se parâmetros que possam informar a condição de conectividade do usuário, por exemplo, o tamanho das janelas de congestionamento e do receptor e a estimativa do RTT médio. Em servidores Linux, os dados sobre a janela de congestionamento, a janela do receptor e o RTT médio podem ser facilmente obtidos do TCB (*Transmission Control Block*), campos *tcb_cwnd, tcb_swindow* e *tcb_srt,* respectivamente [BAL 98]. O tamanho do arquivo a ser transferido pode ser obtido do sistema de arquivos tão logo a conexão tenha sido estabelecida e a solicitação HTTP tenha sido recebida. De

posse destas informações é possível configurar a prioridade da conexão utilizando-se da chamada de sistema *setsockopt()*. A descrição exata de como configurar a prioridade da conexão é apresentada em [HAR 01]. A descrição dos campos *tcb_cwnd, tcb_swindow* e *tcb_srt* do TCB e como acessá-los é dada em [BAL 98]. Utilizando-se destes recursos do sistema operacional a política estará alterando apenas a ordem na qual as requisições são processadas na fila de rede. Alterações também precisam ser realizadas para modificar a ordem de processamento das requisições na fila de disco.

IV. O Experimento de Simulação

O objetivo deste trabalho é propor e avaliar a política FCF simulando as condições de transmissão de arquivos da Internet. O ambiente de simulação consiste de modelos para o servidor Web, para a Internet e a geração de carga de trabalho.

A. Modelo de Simulação para o Servidor Web

Um servidor Web é um programa que aceita conexões com o objetivo de servir requisições de acesso às páginas Web armazenadas no seu sistema de arquivos. O atendimento de requisições utiliza majoritariamente três recursos do sistema: CPU, disco e interface de rede. O processamento das requisições no servidor pode ser descrito da seguinte forma: estabelecimento da conexão TCP, recebimento da requisição HTTP, processamento do cabeçalho da requisição, leitura do arquivo do disco ou do cache, empacotamento e envio dos dados para o usuário, fechamento da conexão e registro no arquivo de *log*. Dentro desta seqüência de ações são calculados os custos de CPU, disco e interface de rede. O servidor foi simulado, portanto, utilizando-se três filas: fila de CPU, fila de disco e fila de interface de rede. A mesma abordagem é utilizada nos trabalhos [CHE 98, CRO 99, BAN 01, HAR 01].

A fila de CPU representa o tempo gasto para fazer o *parser* da requisição, empacotamento e desempacotamento dos segmentos TCP e IP, controle de transferência dos dados do disco para a memória e da memória para a interface de rede, entre outros aspectos do tratamento de uma requisição HTTP. Considerando-se que nos estágios de recebimento e *parser* da requisição ainda não estão disponíveis as informações para definir a prioridade, a fila de CPU terá suas requisições processadas na ordem de chegada, para admissão no *pool* de processos e, de acordo com a disciplina de escalonamento *processor sharing*, para as requisições no *pool*. Este modelo é considerado uma abstração da política padrão utilizada em servidores Web [BES 97].

O processamento da fila de disco consiste em transferir dados do arquivo requisitado, passando-se estes dados do sistema de arquivos para o *buffer* do *socket*. Os dados são lidos do disco em blocos de 16 Kbytes (*block mode*). Depois de ler um bloco de dados, o descritor da conexão é transferido para a fila de rede. A fila da interface de rede terá o custo de transferir os dados do *buffer* do *socket* para o *buffer* da interface de rede. Neste ponto, os pacotes podem ser enviados pela rede. Quando a requisição tiver seus dados totalmente transferidos para o *buffer* de rede ela é retirada da fila de disco. Contudo, caso existam bytes remanescentes a serem transmitidos, a requisição deve voltar para a fila de disco aguardando que este envie novos dados para o *buffer* de *socket*.

A requisição somente estará concluída quando tiver cumprido a soma dos custos das filas de CPU, disco e rede, considerando-se que uma requisição não pode ser processada simultaneamente nas três filas [CRO 99, BAL 98]. O processamento é concorrente nas três filas.

Foram feitos experimentos com dois conjuntos de políticas para comparação. O primeiro conjunto, denominado FIFO nos gráficos deste trabalho, consiste na implementação da política FIFO para as filas de disco e de rede. O segundo conjunto, denominado FCF, implementa a política FCF nas filas de disco e de rede. A fila de CPU tem o mesmo tratamento, descrito acima, para os dois conjuntos.

B. Simulação da Carga

A carga de trabalho de um servidor Web corresponde ao conjunto de requisições HTTP recebidas durante um período de tempo. Para obter um ambiente que seja o mais realístico possível, os acessos ao servidor foram simulados utilizando-se os conceitos do SURGE [BAR 98, BAR 97]. O SURGE é um *benchmark* utilizado para medir o desempenho de servidores Web através de um fluxo contínuo de requisições que simula o acesso de centenas ou milhares de usuários. Estes acessos reproduzem as principais características de uma carga real de trabalho imposta a um servidor Web. Dentre estas características estão a distribuição de cauda pesada dos tamanhos de arquivo, a popularidade dos arquivos, a localidade temporal, o número de arquivos por página, o *think time* e o número de usuários concorrentes.

C. Modelo para Simulação da Internet

O terceiro componente do ambiente de simulação é a Internet. Nela encontramos aspectos dos protocolos TCP e HTTP, variações na velocidade de conexão dos usuários, variações na velocidade dos *links* intermediários que conectam o usuário, quantidade de tráfego disputando o *link*, número de *hops* entre usuário e servidor e, por fim, carga e desempenho do provedor de acesso [PAD 95, RUB 96, LUI 98, BAR 99]. Os parâmetros da rede utilizados nesta simulação são o RTT, o *bandwidth* e a quantidade de tráfego disputando os *links*, que simulará condições de congestionamento. Para definir estes parâmetros utilizamos valores apresentados em alguns trabalhos. A partir do

tamanho do arquivo, de um RTT e de um *bandwidth*, a taxa de transmissão do arquivo é calculado.

O RTT é definido a partir do RTT mínimo. O RTT é determinado pela velocidade da luz, pelo retardo imposto pelos pontos de conexão (roteadores, protocolos) e pelo atraso devido ao tráfego da rede. No trabalho [ACH 96] foi apresentado um estudo do RTT para 90 *links*. Os menores RTTs mínimos, na faixa de 10 a 20ms, foram observados entre *hosts* fisicamente mais próximos, a maioria dentro dos Estados Unidos e alguns no Canadá. Os *links* para *hosts* estrangeiros tiveram os maiores RTTs mínimos, entre 90 e 600 ms. Foram avaliados *hosts* de quatorze países.

Com base nas informações do artigo [ACH 96] foram estabelecidas quatro classes de RTT mínimo, a saber: classe **X**, 20 ms, classe **Y**, 60 ms, classe **Z**, 110 ms e classe **W**, 250 ms. Foi considerado que 25% dos RTTs são da classe **X**, 35% da classe **Y**, 15% da classe **Z** e 25% da classe **W**. As classes **X** e **Y** têm o objetivo de simular acessos de usuários próximos e medianamente próximos. As classes **Z** e **W** visam simular acessos de usuários mais distantes.

Uma simulação mais realística do RTT não envolve apenas o RTT mínimo mas também as variações decorrentes do tráfego na rede. As condições do tráfego geram variações no RTT que podem ser modeladas por distribuição de cauda pesada, como a de Pareto [ACH 96]. Na simulação da Internet o RTT utilizado é modelado por uma distribuição de Pareto limitada, que tem como parâmetro o RTT mínimo de cada classe.

O RTT gerado pela distribuição é limitado a oito vezes o RTT mínimo. Os RTTs que ultrapassam este limite são considerados *time out* e, conseqüentemente, perda de pacotes. A probabilidade disso ocorrer é de 5%. Esta é a probabilidade de perda de pacotes apresentada pelo artigo [CAR 98]. Outro parâmetro exigido pela distribuição de Pareto, além do valor mínimo, é o grau de variabilidade, denominado alfa, que é fixado em 1,4. Esse modelo para definição dos RTTs gera valores compatíveis com as medidas apresentadas no artigo [CRO 95], no qual, para 5262 *hosts* servidores, observou-se uma média de 241ms e um desvio padrão de 435ms para o RTT. A velocidade de transmissão é modelada pelas freqüências exibidas na Tabela I.

TABELA I
VELOCIDADE DA CONEXÃO DO USUÁRIO

Bandwidth	Freqüência (%)	Classe de RTT
14 Kbps	1,80	110 e 250
28 Kbps	16,40	110 e 250
33 Kbps	18,40	110 e 250
56 Kbps	33,60	110 e 250
128 Kbps	6,00	20 e 60
1 Mbps	13,30	20 e 60
4 Mbps	3,50	20 e 60
10 Mbps	6,70	20 e 60

Estas velocidades de transmissão foram obtidas em uma pesquisa realizada pelo *Graphics, Visualization, & Usability Center* do *Georgia Institute of Technology* com 2710 usuários de diversos países em outubro de 1998 [TEN 98]. É importante observar que a característica que motiva a proposta da política FCF é a variabilidade das condições de conectividade dos usuários. Acreditamos que dados mais recentes podem alterar a freqüência apresentada na Tabela I mas não acreditamos que houve diminuição da variabilidade das condições de conectividade. Pelo contrário, é provável que tenha havido um aumento da variabilidade, uma vez que, enquanto já há redes de alta velocidade em operação, nada indica que houve progressos para os usuários com conexões lentas. Na simulação cada requisição é realizada através de uma conexão representada por uma linha da Tabela I, com as características definidas. As conexões com velocidades iguais ou acima de 128 Kbps são enquadradas somente nas classes X e Y de RTT, enquanto as demais, com velocidades abaixo de 128 Kbps, são enquadradas nas classes Z e W.

V. RESULTADOS DA SIMULAÇÃO

A. Descrição do Experimento

A carga de serviço foi gerada tendo como base um servidor Web com 10.000 arquivos diferentes. Em cada execução da simulação foram atendidas 1.006.149 requisições. O tamanho médio dos arquivos foi de 14.072 bytes. O número de usuários simultâneos foi variado de 300 a 1000, representando a carga do sistema. Cada uma das políticas (FIFO e FCF) foi avaliada com a mesma carga de trabalho, isto é, os mesmos arquivos foram requisitados e na mesma seqüência. A constante beta teve valor igual a cinco, para gerar os resultados apresentados.

B. Comparação entre as políticas

Uma forma de avaliar a justiça de uma política de escalonamento é através da métrica chamada *slowdown*. Neste trabalho o *slowdown* foi calculado dividindo-se o tempo de permanência da requisição no servidor (filas mais serviço) pelo custo de transmissão na Internet. Um *slowdown* baixo significa que o servidor está tratando as requisições de forma compatível com o seu custo de processamento e com o tratamento recebido por elas na Internet. O gráfico da Figura 1 apresenta o *slowdown* médio e o maior *slowdown* observado para ambas as políticas. O gráfico é apresentado com duas escalas sendo a da esquerda para o *slowdown* médio e a da direita para o maior *slowdown*.

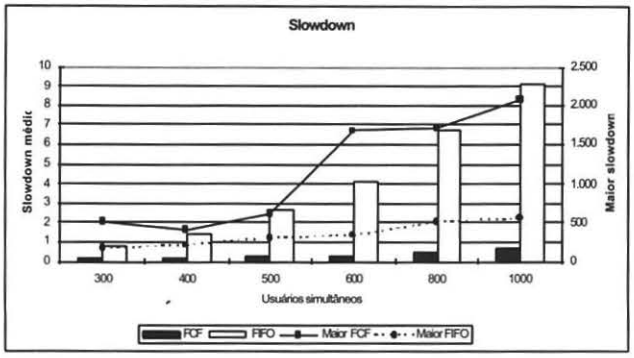

Figura 1 : Slowdown médio

Podemos observar que o *slowdown* médio para a política FCF é inferior à unidade para qualquer valor de carga. A obtenção de um *slowdown* médio bastante baixo para FCF é decorrência, principalmente, do maior peso que a política de escalonamento atribui ao tamanho da requisição. Dessa forma, evita-se que pequenas requisições permaneçam durante muito tempo no servidor. No caso da FIFO, o atendimento de uma requisição muito grande obriga as requisições, comparativamente muito pequenas (que são a maioria), a aguardar na fila um tempo dezenas ou centenas de vezes maior que seu tamanho, o que contribui para aumentar o *slowdown*. Por outro lado, a política FCF apresentou valores máximos de *slowdown* bem maiores do que os da política FIFO. Isso demonstra que alguns processos estão sendo prejudicados em benefício de outros. Esse fato é perfeitamente previsto uma vez que, em sistemas com recursos limitados, não se pode dar benefícios a alguns processos sem que outros sejam penalizados. Deve ser evitado, contudo, que alguns processos sejam demasiadamente prejudicados a ponto de não serem concluídos ou levarem um tempo inaceitável para conclusão. Esta situação de *starvation* é analisada na próxima subseção.

Uma segunda métrica importante é o número de requisições pendentes no servidor. Requisições pendentes são aquelas cujas conexões já poderiam ter sido fechadas pelo servidor, mas que ainda não foram concluídas devido à demora na transferência pela Internet. Na simulação, uma requisição está pendente se ela não depende mais de processamento em nenhum dos recursos, mas ainda não foi totalmente transferida pela Internet. Para requisições de conexões lentas, mesmo que o servidor disponha de recursos para concluir seu processamento, isso não ocorre pois o envio de pacotes e o recebimento de ACKs se prolonga por um tempo maior, obrigando o servidor a manter a conexão aberta. O gráfico da Figura 2 mostra que o número de conexões pendentes em função da carga é crescente para a política FIFO e aproximadamente estável para a política FCF.

A redução do número médio de requisições pendentes na política FCF de 246 com 400 usuários para 207 com um número maior de usuários, deve-se ao fato do aumento do número de requisições de conexões mais velozes. Como o servidor atende prioritariamente as requisições dos usuários de conexões mais velozes, estes tendem a ter respostas mais rápidas e, conseqüentemente, fazer mais requisições em menor tempo. O aumento do número de requisições de conexões mais velozes diminui o tempo de transferência na Internet e, consequentemente, reduz ainda mais o número de requisições pendentes.

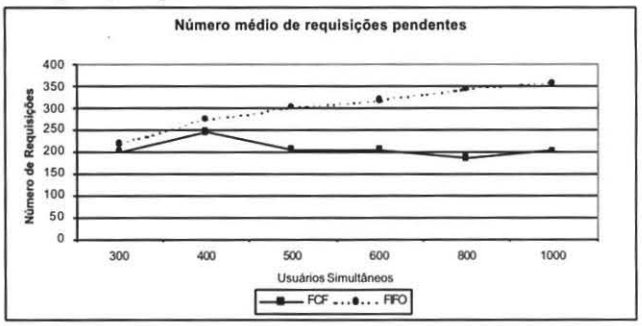

Figura 2 : Número médio de requisições pendentes

O gráfico da Figura 3 mostra o tempo médio de resposta para cargas de trabalho variando entre 300 e 1000 usuários simultâneos. Para todas as cargas, o tempo médio de resposta da política FIFO é no mínimo 60% maior do que o da política FCF. Isso demostra que a política proposta reduz significativamente o tempo médio de resposta no servidor.

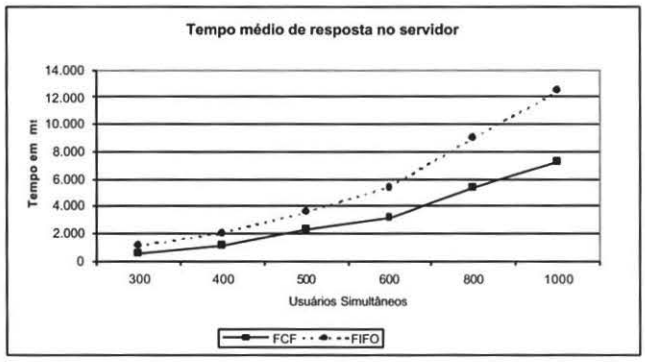

Figura 3 : Tempo médio de resposta no servidor

C. Avaliação do Efeito da Política FCF sobre as Requisições de Arquivos Grandes

Para avaliar o efeito da política sobre as requisições de arquivos grandes, ou seja, de uma possível situação de *starvation*, a carga foi dividida em oito classes de arquivos. O percentual de bytes e requisições em cada classe é exibido pelo gráfico da Figura 4. Como podemos observar, a classe dos arquivos acima de 128Kbytes representa apenas 1,41% das requisições, mas corresponde a 39,50% dos bytes a serem processados e transmitidos. Portanto, essa classe, apesar de ter um número reduzido de

requisições, representa grande parte da carga imposta ao servidor e à Internet - considerando que o custo de processamento é proporcional ao tamanho do arquivo. O fato de um pequeno número de requisições corresponder a grande parte da carga de serviço impede que os processos grandes sejam demasiadamente prejudicados. Observamos no gráfico da Figura 4, que cerca de 85% dos arquivos requisitados são menores que 16Kbytes e representam apenas 21,50% da carga. Dessa forma, mesmo dando-se prioridade máxima a 85% da carga, o servidor ainda vai dispor de 78,5% dos recursos para tratar os 15% restantes das requisições. Considerando-se esta característica da carga Web, não há necessidade da utilização de prioridades dinâmicas, como a utilizada no artigo [CHE 98], para evitar que ocorra *starvation* dos processos grandes. Essa linha de raciocínio é apresentada nos trabalhos [HAR 98, CRO 99, BAN 01, HAR 01].

Por outro lado, uma vez que a grande maioria das requisições são para pequenos arquivos, dar prioridade a estas requisições tem uma influência muito grande no tempo médio de resposta.

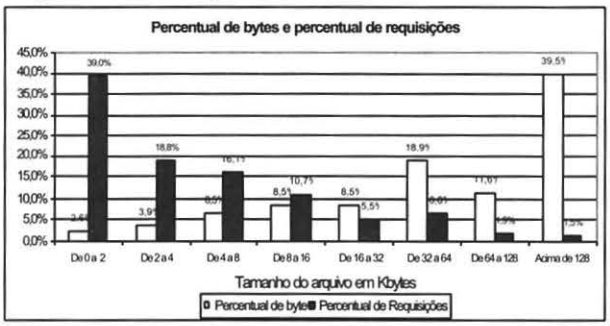

Figura 4 : Percentual de bytes *versus* percentual de requisições

No gráfico da Figura 5 pode-se observar o tempo de resposta por tamanho de arquivo para 1000 usuários simultâneos. O eixo y é apresentado em escala logarítmica. Apenas as requisições para arquivos com mais de 128Kbytes sofrem pequena penalização na política FCF. Há ganhos muito expressivos para todas as demais classes.

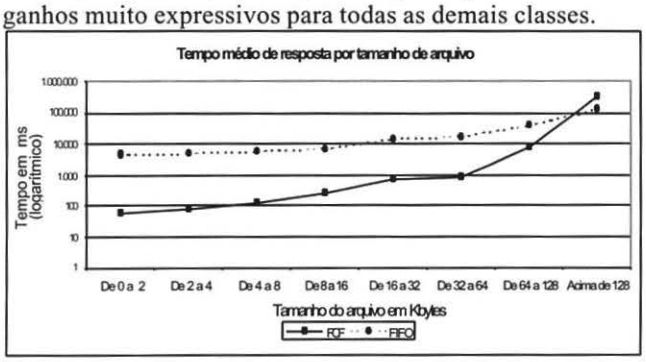

Figura 5 : Tempo médio de resposta por tamanho de arquivo

D. Avaliação do Efeito da Política FCF sobre requisições feitas através de conexões lentas

O gráfico da Figura 6 mostra o tempo médio de resposta em função da velocidade de conexão para 1000 usuários simultâneos. Nele observamos que os tempos médios de resposta acompanham as condições de conectividade, isto é, usuários de conexões mais rápidas têm melhores tempos de resposta. Usuários de conexões mais lentas observam um tempo médio de resposta maior do que o observado para a política FIFO. Apesar deste aumento, não ocorre *starvation* nas conexões lentas. Isso pode ser explicado pelo fato de que a velocidade de conexão é apenas um dos critérios para definir a prioridade. O tamanho da requisição tem maior influência na definição da prioridade. A redução no tempo médio de resposta gerado pela prioridade dada às requisições menores contribui para que as conexões mais lentas não sofram *starvation*. Para comprovar esta afirmação devemos recorrer ao gráfico da Figura 3, onde podemos observar que o tempo médio de resposta para 1000 usuários na opção FIFO é de 12.446 ms, e para FCF é de 7.202 ms.

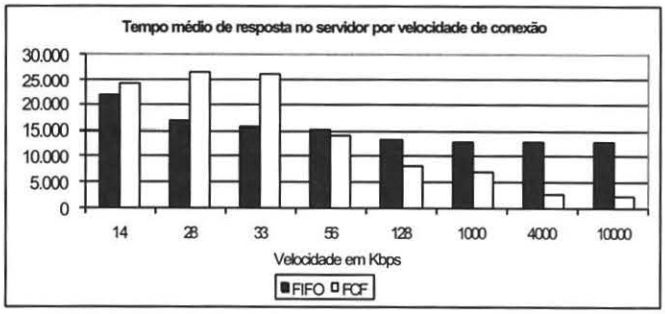

Figura 6 : Tempo médio de resposta por velocidade de conexão

VI. Conclusão e Trabalhos Futuros

A necessidade de interação com a Internet e as características da carga de serviço tornam o servidor Web um ambiente computacional ímpar. A variabilidade das condições de conectividade, aliada às características dos protocolos TCP e HTTP afetam consideravelmente o desempenho do servidor. Considerar as características da carga de serviço e das condições de conectividade traz ganhos de desempenho para o servidor, conforme foi mostrado neste trabalho através da política FCF (*Fastest Connection First*). Nessas condições a política proposta demonstrou ser uma opção interessante para garantir um melhor desempenho de servidores Web.

Com a política FCF observamos uma excelente redução do tempo médio de resposta e do número de requisições concorrentes, sem penalização excessiva das requisições grandes e das requisições feitas através de conexões lentas. Observamos também, que o tempo de resposta torna-se

mais justo, comparativamente ao tamanho da requisição e à velocidade de transmissão na Internet. Os usuários passam a esperar tempos compatíveis com suas condições de conectividade. Por fim, ficou comprovado que o benefício dado aos usuários com conexões velozes e às requisições menores não traz demora excessiva aos demais usuários e requisições.

Uma continuação importante deste trabalho é a implementação da política em um ambiente operacional real, tal como a composição Linux e Apache. Várias outras métricas de desempenho foram avaliadas mas, devido a restrições de espaço, não são apresentadas nos resultados.

REFERÊNCIAS BIBLIOGRÁFICAS

[ALM 98] ALMEIDA, Jussara; DABU, Mihaela; MANIKUTTY, Anand; CAO, Pei. *Providing Differentiated Levels of Service in Web Content Hosting*. In proceedings of the Workshop on Internet Server Performance, Madison, Wisconsin, June 1998.

[CRO 95] CROVELLA, Mark E.; CARTER, Robert. *Dynamic Server Selection in the Internet*. In proceedings of the Third IEEE Workshop on the Architecture and Implementation of High Performance, August 1995.

[BAR 98] BARFORD, Paul; CROVELLA, Mark. *Generating Representative Web Workloads for Network and Sever Performance Evaluation*. In proceeding of the 1998 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer System, Madison, July 1998.

[ALM 97] ALMEIDA, Virgilio A. F.; ALMEIDA, Jussara; YATES, David. *WebMonitor: a Tools for Measuring World-Wide Web Server Performance*. In proceedings of the Seventh IFIP Conference on High Performance Networking (HPN), White Plains, NY, April 1997.

[DIL 97] DILLEY, John; FRIEDRICH, Rich; JIN, Tai; ROLIA, Jerome. *Measurement Tools and Modeling Techniques for Evaluating Web Server Performance*. In proceedings 9th Int. Conf. on Modelling Techniques and Tools. Springer-Verlag, 1997.

[BES 97] BESTAVROS, Azer; KATAGAI, Noami; LONDOÑO, Jorge. *Admission Control and Scheduling for High-Performance WWW Servers*. Tech. Report. BUCS-TR-97-015, Boston University, Computer Science Department, August 1997.

[PAD 95] PADMANABHAN, Venkata N.; MOGUL, Jeffrey C. *Improving HTTP Latency*. Computer Networks and ISDN Systems, v.28, pp. 25-35, December 1995.

[MUR 96] MURTA, Cristina D.; MARQUES, Jussara A.; ALMEIDA, Virgílio A.F. *Performance Analysis of a WWW Server*. 22nd International Conference on Technology Management and Performance Evaluation of Enterprise-Wide Information Systems, San Diego, California, December 8-13, 1996.

[CHE 98] CHERKASOVA, Ludmila. *Scheduling Strategy to Improve Response Time for Web Applications*. In High-performance computing and networking: international conference and exhibition, 1998.

[TEN 98] Tenth WWW User Survey (Conducted October 1998), Graphics, Visualization & Usability (GVU) Center at Georgia Tech, http://www.gvu.gatech.edu/user_surveys

[RUB 96] RUBARTH-LAY, James. *Keeping the 400lb. Gorilla at Bay: Optimizing Web Performance*. For LIS 385 T.6, Electronic Distribution of Organizational Information, Spring 1996.

[LUI 98] LUI, Binzhang; ABDULLA, Ghaleb; JOHNSON, Tommy; FOX, Edward. *Web Response Time and Proxy Caching*. In proceedings of WebNet98, Orlando, FL, November 1998.

[MUR 99] MURTA, Cristina Duarte. *Modelo de Particionamento de Espaço para Caches da World Wide Web*. Tese de Doutorado. Departamento de Ciência da Computação, Universidade Federal de Minas Gerais, agosto de 1999.

[ACH 96] ACHARYA, Anurag; SALTZ, Joel. *A Study oj Internet Round-Trip Delay*. Technical report CS-TR-3736, Department of Computer Science, University of Maryland, USA, December 1996.

[CAR 98] CARDWELL, Neal; SAVAGE, Stefan; ANDERSON, Tom. *Modeling the Performance of Short TCP Connections*. Technical report. Department of Computer Science and Engineering, Univ. of Washington, 1998.

[HAR 98] HARCHOL-BALTER, Mor; CROVELLA, Mark; PARK, SunSim. *The case for SRPT scheduling in web servers*. Technical report MIT-LCS-TR-767, MIT Lab for Computer Science, October 1998.

[CRO 99] CROVELLA, Mark; FRANGIOSO, Bob; HARCHOL-BALTER, Mor. *Connection Scheduling in Web Servers*. USENIX Symposium on Internet Technologies and Systems, p. 243-254, Boulder, Colorado, October 1999.

[BAN 01] BANSAL, Nikhil; HARCHOL-BALTER, Mor. *Analysis of SRPT Scheduling: Investigating Unfairness*. In proceedings of ACM Sigmetrics 2001, Conference on Measurement and Modeling of Computer Systems.

[HAR 01] HARCHOL-BALTER, Mor; BANSAL, Nikhil; SCHROEDER, Bianca; AGRAWAL, Mukesh. *Size-based Scheduling to Improve Web Performance*. In proceedings of ACM Sigmetrics 2001 Conference on Measurement and Modeling of Computer Systems.

[BAR 99] BARFORD, Paul; CROVELLA, Mark. *A Performance Evaluation of HyperText Transfer Protocols*. In proceedings of ACM SIGMETRICS '99, p. 188-197, May 1999.

[CON 67] CONWAY, Richard; MAXWELL, William; MILLER, Louis. Theory of Scheduling. Addison-Wesley Publishing Company, 1967.

[BAL 98] BALAKRISHNAN, Hari; PADMANABHAN, Venkata N.; SESHAN, Srinivasan; STEMM, Mark; KATZ, Randy. *TCP behavior of a busy Internet server: Analysis and improvements*. In proceedings of IEEE INFOCOM, p. 252-262, March 1998.

[BAR 97] BARFORD, Paul; CROVELLA, Mark. *An architecture for a WWW workload generator*, Wide Web Consortium Workshop on Workload Characterization, October 1997.