

# Novas Políticas de Invocação Paralela para Seleção Dinâmica de Servidores Replicados\*

Wesley E. Q. de Sousa, Nabor C. Mendonça

<sup>1</sup>Mestrado em Informática Aplicada, Universidade de Fortaleza  
Av. Washington Soares, 1321 – 60811-905 Fortaleza – CE

wesleyevaristo@gmail.com, nabor@unifor.br

**Resumo.** *Servidores geograficamente replicados são comumente utilizados na internet para oferecer maior desempenho, confiabilidade e escalabilidade às aplicações clientes. Nesse contexto, surge o problema de como selecionar o servidor que melhor atenderá a requisição de uma dada aplicação cliente. Este trabalho propõe três novas políticas de seleção de servidores replicados, a saber: First Connection, First Read e Best Partial. As três políticas seguem o princípio da invocação paralela ao conjunto de servidores replicados, com a diferença de que as conexões estabelecidas com cada servidor passam a ser dinamicamente monitoradas, com as conexões mais lentas sendo progressivamente encerradas até que reste uma única conexão ativa. Resultados empíricos, obtidos através da invocação de servidores replicados localizados em diversos países, mostram que as novas políticas apresentam um desempenho bastante favorável frente ao da política paralela clássica e relativamente próximo ao da escolha do melhor servidor previamente identificado para cada cenário investigado.*

**Abstract.** *Geographically replicated servers are commonly used in the internet to offer greater performance, reliability and scalability to client applications. In this context, there is the problem of how to select the replicated server that will best serve the request of a giving client application. This paper proposes three novel replicated server selection policies, namely First Connection, First Read and Best Partial. The three policies are based on the parallel invocation principle, in which all replicated servers are invoked in parallel, with the difference that all connections established with each server are now dynamically monitored, with slower connections being gradually killed until there is only one connection alive. Empirical results, obtained from the invocation of real servers replicated over several countries, show that the proposed policies perform much better than the classical parallel invocation policy, and relatively close to the selection of the best server previously identified for each scenario evaluated.*

## 1. Introdução

A crescente variedade dos serviços oferecidos na internet e a sua frequente utilização a partir de praticamente todos os lugares do planeta têm elevado bastante o nível

---

\*Este trabalho é parcialmente financiado pelo Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), processo no. 309140/2008-0, e pela Fundação Cearense de Apoio ao Desenvolvimento Científico e Tecnológico (FUNCAP), processo no. 9129/08.

de exigência dos usuários, que buscam cada vez mais serviços que respondam de forma rápida e confiável suas solicitações. Uma solução comumente adotada para propiciar maior desempenho, confiabilidade e escalabilidade aos serviços é a replicação [Salas et al. 2006]. O uso da replicação melhora o desempenho das aplicações na medida em que serviços críticos ou de alta demanda podem ser oferecidos por múltiplos servidores, permitindo uma distribuição equilibrada das solicitações entre os vários servidores e reduzindo assim a carga de trabalho individual de cada servidor. A replicação também propicia uma maior disponibilidade aos serviços, uma vez que requisições enviadas a um servidor que esteja indisponível ou sobrecarregado poderão ser redirecionadas para outro que ofereça um serviço idêntico ou equivalente.

Um grande desafio proposto pela replicação de serviços, particularmente quando são utilizados servidores geograficamente dispersos – uma prática cada vez mais comum, decorrente da crescente globalização dos serviços –, é a seleção do servidor mais adequado para cada cliente em um dado momento. Esta escolha não é uma tarefa trivial e tem sido objeto de intensos estudos nos últimos anos [Sayal et al. 1998, Dykes et al. 2000, Hanna et al. 2001, Dharmalingam and Collier 2003, Conti et al. 2005, Junior et al. 2007, Mendonça et al. 2008]. Alguns desses estudos avaliaram empiricamente, através de experimentos conduzidos no ambiente real da Web, o desempenho de várias políticas de seleção de servidores propostas na literatura, com foco naquelas implementadas no lado das aplicações clientes [Junior et al. 2007, Mendonça et al. 2008]. Os resultados desses estudos mostram que dois tipos de políticas destacam-se em termos de desempenho: as políticas estatísticas, que analisam dados históricos sobre invocações anteriores para selecionar o servidor mais apropriado no momento de uma nova invocação, e as políticas paralelas, que realizam solicitações simultâneas ao grupo de servidores, descartando todas as respostas recebidas exceto a primeira. O bom desempenho dessas políticas pode ser atribuído ao fato delas se adaptarem mais facilmente a cenários dinâmicos, em que os desempenhos da rede e dos servidores variam com frequência, como é caso de muitos serviços da Internet. Entretanto, ambas as abordagens sofrem de graves limitações, e que têm impedido a sua maior utilização na prática, como a baixa escalabilidade das políticas paralelas e a alta dependência das políticas estatísticas em ter dados históricos sempre atualizados.

Este trabalho propõe três novas políticas de seleção de servidores replicados. As três políticas propostas, denominadas *First Connection*, *First Read* e *Best Partial*, seguem o princípio da invocação paralela ao conjunto de servidores replicados, com a diferença de que, ao contrário da política de invocação paralela clássica [Mendonça et al. 2008], as conexões estabelecidas com cada servidor passam a ser dinamicamente monitoradas, com as conexões mais lentas sendo progressivamente encerradas até que reste uma única conexão ativa. Cada uma das três políticas adota uma estratégia diferente de monitoramento do desempenho das conexões, apresentando um comportamento menos ou mais agressivo com relação à velocidade de encerramento das conexões consideradas ineficientes. Mais especificamente, a política *First Connection* invoca todo o conjunto de servidores replicados e mantém ativa apenas a primeira conexão efetivamente estabelecida com a aplicação cliente. A política *First Read* tem um comportamento similar, mas mantém ativa apenas a conexão de onde a aplicação cliente leu o primeiro pacote de dados referente à resposta do serviço. Por fim, a política *Best Partial* mantém todas as conexões ativas inicialmente, encerrando-as gradualmente, com base no seu desempenho parcial, até que reste apenas

uma. As três políticas, juntamente com a política paralela clássica, foram avaliadas empiricamente, através da invocação de servidores da Web localizados em diversos países. Os resultados obtidos mostram que, em clientes com largura de banda restrita, as novas políticas apresentam um desempenho bastante favorável frente ao da política paralela clássica e relativamente próximo ao da escolha do melhor servidor previamente identificado para cada cenário investigado.

O restante do artigo está organizado da seguinte maneira. A Seção 2 descreve as principais estratégias de seleção de servidores replicados propostas na literatura, discutindo suas características e principais limitações. A Seção 3 apresenta as novas políticas propostas. A Seção 4 descreve a metodologia utilizada na avaliação das novas políticas e analisa os resultados obtidos. Por fim, a Seção 5 conclui o trabalho e oferece algumas sugestões para trabalhos futuros.

## 2. Trabalhos Relacionados

Políticas de seleção de servidores geograficamente replicados podem ser implementadas de três formas distintas, dependendo do local onde a seleção do servidor efetivamente é realizada [Dharmalingam and Collier 2003]: (1) no lado do servidor, (2) no lado do cliente ou (3) na infraestrutura de rede que interliga ambos os lados.

Segundo [Dharmalingam and Collier 2003], políticas de seleção implementadas no lado do servidor ou na infraestrutura da rede geralmente servem para equilibrar a carga entre os servidores e não levam em consideração nem localização nem as características de rede dos clientes. Por outro lado, quando a seleção do servidor é realizada no lado do cliente, é possível contemplar as condições da rede em cada cliente, bem como as variações na carga dos servidores. Em [Dykes et al. 2000], os autores classificam as políticas de seleção implementadas do lado do cliente em políticas estáticas, estatísticas e dinâmicas. As políticas estáticas utilizam critérios de seleção que não mudam em tempo de execução, como a localização física de cada servidor. As políticas estatísticas realizam suas escolhas em função da análise de invocações anteriores aos mesmos servidores, mantidas em uma base histórica. As políticas dinâmicas, por sua vez, escolhem um servidor em tempo de invocação do serviço, como através da invocação paralela de todo o conjunto de servidores, e são sensíveis a variações de carga nos servidores ou na rede. Há ainda políticas híbridas, que combinam características de políticas de mais de uma classe, como políticas de invocação paralela que apenas invocam os servidores com os melhores resultados na base histórica [Junior et al. 2007].

Diversos estudos têm sido realizados especificamente para avaliar o desempenho de políticas de seleção de servidores implementadas no lado dos clientes (por exemplo, [Hanna et al. 2001, Dykes et al. 2000, Junior et al. 2007, Mendonça et al. 2008]). Algumas conclusões desses estudos são que políticas estáticas só são recomendadas para ambiente conhecidos, onde o desempenho de cada servidor é previsível [Hanna et al. 2001, Dykes et al. 2000]. Em ambientes menos previsíveis, como é o caso da Internet, estudos mais recentes apontam que os melhores resultados são obtidos com as políticas estatísticas e as políticas dinâmicas baseadas na invocação em paralelo do conjunto de servidores [Junior et al. 2007, Mendonça et al. 2008]. No estudo descrito em [Mendonça et al. 2008], em particular, a política de invocação paralela apresentou o melhor desempenho entre todas as políticas avaliadas no cliente onde a largura de banda

era suficiente para absorver a concorrência no recebimento das respostas. Por outro lado, quando não se leva em consideração a banda disponível em cada cliente, as políticas estatísticas se apresentaram como a melhor estratégia de seleção. Já em [Junior et al. 2007], as políticas baseadas no princípio da invocação paralelas e suas variações apresentaram os melhores desempenhos em todos os cenários avaliados.

Apesar dos ganhos de desempenho observados com políticas estatísticas e de invocação paralela, ainda há uma série de restrições quanto à sua utilização prática. Por exemplo, a precisão de se utilizar dados históricos para prever o desempenho futuro de servidores pode ser fortemente prejudicada se esses dados não forem atualizados com frequência, ou se o desempenho da rede ou dos servidores estiver sujeito a flutuações repentinas. No caso das políticas de invocação paralela, fatores como o tamanho da resposta do serviço, o número de conexões simultâneas ativas e a banda local disponível em cada cliente podem afetar negativamente o tempo de resposta do servidor que primeiro responder a requisição do cliente, resultando em um desempenho inferior àquele que seria obtido se o mesmo servidor fosse invocado isoladamente. Além disso, essa técnica oferece baixa escalabilidade, uma vez que a sua utilização por um número grande de clientes implicaria em um aumento significativo da demanda por todo o grupo de servidores replicados. Mesmo quando apenas um subconjunto dos servidores replicados é invocado em paralelo, conforme a política híbrida proposta em [Junior et al. 2007], dados históricos desatualizados ainda poderiam prejudicar a escolha do melhor subconjunto de servidores no momento da invocação.

Um outra solução bastante utilizada na invocação de serviços replicados é o acesso paralelo a diferentes fragmentos da resposta do serviço, onde cada fragmento é provido por um ou mais servidores [Rodriguez and Biersack 2002]. Variações dessa solução são comumente encontradas nas chamadas Redes de Distribuição de Conteúdo (*Content Distribution Networks* – CDNs) [Kangasharju et al. 2002], onde cópias dos recursos (ou de seus fragmentos) são mantidas em servidores geograficamente dispersos, visando diminuir a distância física entre servidores e clientes. O problema dessa abordagem é que ela não suporta a seleção de servidores com conteúdos dinâmicos ou semanticamente equivalentes, limitando-se a serviços de conteúdos estáticos.

### **3. Novas Políticas de Invocação Paralela**

As novas políticas foram concebidas de modo a contemplar duas importantes propriedades: (i) limitar o processo de seleção a um número reduzido de servidores, como ocorre com as políticas estatísticas; e (ii) não depender da existência de dados históricos atualizados, como ocorre com as políticas de invocação paralela. A primeira propriedade garante uma maior escalabilidade no acesso ao serviço replicado, além de evitar o problema da sobrecarga local em clientes com banda restrita. Já a segunda propriedade dá maior liberdade para a utilização da política em cenários onde dados históricos inexistem ou o desempenho dos servidores é difícil de prever com base apenas em resultados passados.

Atualmente, essas duas propriedades estão presentes nas políticas estatísticas e de invocação paralela, respectivamente, de forma mutuamente exclusiva. Assim, visando atender esses dois requisitos simultaneamente, este trabalho propõe novas variações da política de invocação paralela, alicerçadas na ideia de monitorar e, quando necessário, encerrar dinamicamente as conexões concorrentes estabelecidas com cada servidor. Deste

modo, torna-se possível selecionar os melhores servidores em tempo de invocação, com base no desempenho parcial de cada conexão, limitando o acesso ao serviço a um número reduzido de servidores. Além disso, o fato da seleção considerar informações sobre o desempenho dos servidores obtidas a partir de suas respectivas conexões elimina a necessidade de coletar dados históricos e mantê-los atualizados.

### 3.1. Princípios Básicos

Inicialmente, o comportamento das novas políticas é semelhante ao da política de invocação paralela clássica, com todos os servidores replicados sendo invocados concorrentemente pela aplicação cliente. No entanto, ao invés de aguardar o recebimento da resposta completa de um dos servidores, as novas políticas reduzem dinamicamente o número de conexões ativas, utilizando critérios de seleção baseados nos tempos parciais que compõem o tempo total de resposta percebido no cliente, até que haja um único servidor respondendo à invocação do serviço. Se a invocação for bem sucedida, seu resultado é repassado à aplicação cliente. Caso contrário, invoca-se novamente, de forma concorrente, o conjunto de servidores replicados exceto aquele que falhou. Este processo se repete até que um dos servidores responda com sucesso ou então até que todos falhem, quando propaga-se uma exceção para a aplicação.

Do ponto de vista do cliente, o tempo de resposta ( $T_R$ ) de um serviço é dado pela seguinte fórmula [Dykes et al. 2000]:

$$T_R = T_1 + T_2 + T_3 + T_4 \quad (1)$$

onde  $T_1$  é o tempo de resolução do nome de domínio do serviço,  $T_2$  é o tempo de estabelecimento da conexão com o servidor,  $T_3$  é o tempo transcorrido entre o envio da requisição HTTP e o recebimento do primeiro pacote de resposta, e  $T_4$  é o tempo gasto para receber os demais pacotes da resposta. Esses tempos serão utilizados para ilustrar as estratégias de seleção adotadas pelas novas políticas, descritas a seguir.

### 3.2. *First Connection*

Esta política seleciona o servidor que primeiro estabelecer uma conexão com a aplicação cliente, imediatamente descartando os outros pedidos de conexão enviados aos demais servidores. Considerando a fórmula mostrada na Equação 1, o servidor escolhido pela política *First Connection* será aquele com a menor soma do tempos de resolução de nome e do tempo de conexão ( $T_1 + T_2$ ) no momento da invocação.

### 3.3. *First Read*

Esta política seleciona o servidor cuja conexão é a primeira a entregar um pacote de dados à aplicação cliente. Todos os pedidos de conexões pendentes ou conexões estabelecidas ainda sem entregar nenhum dado de resposta são imediatamente descartadas. Com relação à fórmula da Equação 1, o servidor escolhido pela política *First Read* será aquele com a menor soma do tempo de resolução de nome, do tempo de conexão e do tempo de recebimento do primeiro pacote ( $T_1 + T_2 + T_3$ ) no momento da invocação. Este tempo inclui o tempo de envio da requisição HTTP ao servidor, o tempo de processamento da requisição pelo servidor, e o tempo de transmissão do primeiro pacote de resposta ao cliente.

O tempo de recebimento do primeiro pacote de resposta de uma requisição HTTP tende a se aproximar do tempo total de resposta do serviço quanto menor for o tamanho da resposta. Uma política semelhante à *First Read*, baseada na latência das requisições HTTP, foi proposta em [Sayal et al. 1998]. Em [Sayal et al. 1998], foi encontrada uma boa correlação (0.73) entre a latência da requisição HTTP e o tempo total de resposta do serviço. No entanto, essa correlação pode não ser tão boa para serviços com respostas proporcionalmente muito maiores do que o tamanho de um único pacote de dados, com veremos mais adiante, na Seção 4.

### 3.4. *Best Partial*

Esta política monitora dinamicamente o progresso das conexões estabelecidas com cada um dos servidores replicados, e seleciona aquele que apresentar o melhor desempenho em tempo de invocação. Tão logo um único servidor seja escolhido, todas as demais conexões pendentes são imediatamente interrompidas. Além disso, durante a monitoração das conexões, enquanto não houver uma escolha definitiva, aqueles servidores com os piores desempenhos serão gradualmente descartados, até que as invocações concorrentes estejam limitadas a um grupo reduzido de servidores, que corresponderão aos servidores com os melhores desempenhos parciais no momento da solicitação do serviço pela aplicação cliente. Com relação à Equação 1, o servidor escolhido pela política *Best Partial* será aquele com o menor tempo de resposta parcial, ou seja, aquele com a menor soma dos tempos  $T_1$ ,  $T_2$  e  $T_3$ , mais o tempo de recebimento dos dados parciais da resposta ( $T_4$ ) no momento do encerramento da penúltima conexão ativa.

Como a tarefa de avaliar o desempenho parcial de uma conexão é um processo subjetivo, a política *Best Partial* adota um critério de avaliação baseado em limiares de progresso previamente configurados. Assim, dependendo dos valores atribuídos a esses limiares, a política *Best Partial* apresentará um comportamento mais agressivo ou mais conservador em relação à velocidade com que as invocações pendentes são interrompidas e ao número de invocações simultâneas permitido. Teoricamente, quanto mais agressiva for a configuração da política, ou seja, quanto mais sensíveis forem os seus limiares de progresso, mais o seu comportamento aproximar-se-á do comportamento da política *First Read*. Por outro lado, quanto mais conservadora for a sua configuração, ou seja, quanto mais tolerantes forem os seus limiares de progresso em relação ao desempenho parcial das conexões, mais o seu comportamento será próximo ao comportamento da política de invocação paralela clássica.

A monitoração do desempenho das invocações concorrentes é feita através do cálculo da taxa de transferência acumulada de cada conexão. Os valores das taxas são coletados e analisados em tempo real, com o objetivo de identificar o nível de dispersão entre eles. Para tanto, utiliza-se um algoritmo baseado no diagrama de caixa (*boxplot*), uma técnica para análise exploratória de dados, que, entre outras características, permite a identificação de valores atípicos (conhecidos como *outliers*), tanto superiores quanto inferiores [Tukey 1970].

A utilização da política *Best Partial* requer a configuração de dois parâmetros: o percentual de leitura mínimo,  $k$ , e o fator de dispersão dos limiares,  $f$ . A combinação de ambos irá definir o grau de agressividade ou conservadorismo do processo de escolha do servidor com o melhor desempenho. O primeiro parâmetro possibilita que o usuário

controle quando a política iniciará o processo de seleção, ou seja, a política só poderá iniciar o processo de seleção após alguma conexão ter efetuado a leitura de pelo menos  $k\%$  dos dados da resposta. O segundo parâmetro indica o grau de dispersão entre os limiares de progresso das conexões, utilizados pela técnica do diagrama de caixa para a identificação de valores extremos superiores e inferiores, ou seja, conexões com taxas de transferência acumulada muito acima ou muito abaixo da mediana das taxas das demais conexões.

Uma vez atingido o percentual mínimo de leitura (parâmetro  $k$ ) por alguma das conexões, a política *Best Partial* passa a aplicar a técnica do diagrama de caixa da seguinte forma: tão logo uma conexão apresente um desempenho muito acima das demais (*outlier superior*), esta será mantida e todas as outras imediatamente encerradas. No caso de haver mais de um *outlier superior*, seleciona-se o de maior valor, ou seja, a conexão com a maior taxa de transferência acumulada será mantida e todas as demais encerradas. A técnica continuará sendo reaplicada enquanto não houver um *outlier superior*. Neste ínterim, quaisquer conexões que apresentem um desempenho muito abaixo das demais (*outliers inferiores*) serão imediatamente encerradas. Não havendo dispersão significativa entre os desempenhos das conexões monitoradas (ou seja, sem *outliers superiores* nem inferiores detectados), as conexões mais lentas são gradualmente eliminadas, até que reste apenas uma única conexão ativa.

### 3.5. Implementação

As novas políticas propostas, bem como a política de invocação paralela tradicional, foram implementadas em Java, na forma de um arcabouço de seleção de servidores denominado WSSF (*Web Service Selection Framework*). O arcabouço WSSF foi projetado para ser utilizado como um *proxy* HTTP programável, podendo ser facilmente estendido com a implementação de novas políticas de seleção. Essa estratégia de desenvolvimento tem como principal vantagem o baixo acoplamento entre as aplicações clientes e a implementação das políticas de seleção de servidores, além de ser independente da tecnologia ou *middleware* de invocação de serviços utilizada pelas aplicações clientes. Maiores detalhes sobre a implementação do arcabouço WSSF podem ser obtidos em [de Sousa 2010]

## 4. Avaliação Experimental

As quatro políticas implementadas no arcabouço WSSF foram avaliadas empiricamente, por meio de três experimentos que consistiam no envio de requisições periódicas a três diferentes grupos de servidores replicados na internet, localizados em regiões geograficamente distantes. As seções a seguir descrevem a metodologia de avaliação utilizada e os resultados obtidos em cada um dos três experimentos realizados.

### 4.1. Metodologia

A métrica de medida de desempenho adotada nos experimentos foi o tempo de resposta percebido pelas aplicações clientes, que corresponde ao tempo transcorrido entre o instante imediatamente anterior ao envio da requisição HTTP até o recebimento total dos dados da resposta enviados pelo servidor. A principal motivação para a utilização dessa métrica é a sua sensibilidade aos diversos fatores que podem afetar o tempo de resposta do serviço observado pela aplicação cliente, como diferenças de configuração e carga

**Tabela 1. Servidores do projeto Mozilla utilizados nos experimentos.**

ID	URL	Localização
S1	http://gd.tuwien.ac.at	Áustria
S2	http://mozilla.phphosts.org	Eslováquia
S3	http://mozilla.saix.net	África do Sul
S4	http://pv-mirror02.mozilla.org	Estados Unidos
S5	http://mirror.atratoip.net	Holanda
S6	http://mozilla.mtk.nao.ac.jp	Japão
S7	http://ftp.up.pt	Portugal
S8	http://mirror.kr.freebsd.org	Coréia do Sul
S9	http://ftp.twaren.net	Taiwan

de trabalho entre os servidores, condições gerais da rede entre servidores e clientes, e diferenças nas configurações locais dos clientes.

Todos experimentos foram executados no laboratório do Mestrado em Informática Aplicada da Universidade de Fortaleza, ao longo do primeiro semestre de 2010. As aplicações clientes foram executadas em uma estação de trabalho com as seguintes características: processador Intel Pentium 4,3 GHz; 1,5 GB de memória RAM; disco rígido de 50 GB; sistema operacional Linux Debian, versão 2.22.3; e conexão à Internet de 1Gbps. Entretanto, para simular uma conexão de rede mais próxima à realidade da maioria dos usuários brasileiros, limitou-se a largura de banda na estação cliente para 1Mbps. O mecanismo de cache do protocolo HTTP foi desabilitado pelas aplicações clientes, através da diretiva Cache-Control. Essa decisão visou eliminar a influência de fatores externos aos experimentos que pudessem interferir no desempenho das políticas avaliadas.

Como o serviço replicado a ser invocado pelas aplicações clientes, foram utilizados três conjuntos de quatro servidores geograficamente dispersos, escolhidos a partir do total de 136 servidores da Web que atualmente compõem o serviço de hospedagem de recursos do projeto Mozilla [Mozilla 2010]. A escolha pelo serviço de hospedagem do Mozilla justifica-se pela sua variedade de recursos e pelo fato de seus servidores estarem fisicamente espalhados por praticamente todas as regiões do planeta.

A utilização de um número pequeno de servidores em cada experimento visou facilitar a criação de diferentes cenários de replicação, onde a escolha do melhor servidor não dependesse de um único critério, como tempo de conexão ou latência. Para isso, foi realizado um experimento inicial, onde todos os 136 servidores do projeto Mozilla foram continuamente invocados, através da solicitação de um recurso de 1MB de tamanho, durante um período de 48 horas. Nesse período, foram coletados os dados referentes à mediana do tempo de conexão, tempo de primeira leitura e tempo de resposta total de cada servidor. Esses dados foram então utilizados para ranquear os servidores de acordo com cada tempo medido, o que permitiu a identificação dos servidores que apresentavam as maiores variações de desempenho em relação a cada um desses tempos. Ao final, foram selecionados nove servidores com características bastante distintas, os quais serviram de base para a definição dos três conjuntos de servidores utilizados nos experimentos. A Tabela 1 lista o identificador, a URL e a localização geográfica dos nove servidores inicialmente selecionados. Ressalta-se que os nove servidores estão fisicamente localizados em quatro continentes diferentes – Europa (4), África (1), América do Norte (1) e Ásia (3) – todos geograficamente distantes do local de origem das requisições – no caso, Fortaleza.

**Tabela 2. Configurações da política *Best Partial* avaliadas nos experimentos.**

Ciclo	BP1	BP2	BP3
1	$k = 10\%, f = 1$	$k = 10\%, f = 1,5$	$k = 20\%, f = 1,5$
2	$k = 10\%, f = 1$	$k = 10\%, f = 1,5$	$k = 20\%, f = 1,5$
3	$k = 2\%, f = 1,5$	$k = 5\%, f = 1,5$	$k = 10\%, f = 1,5$

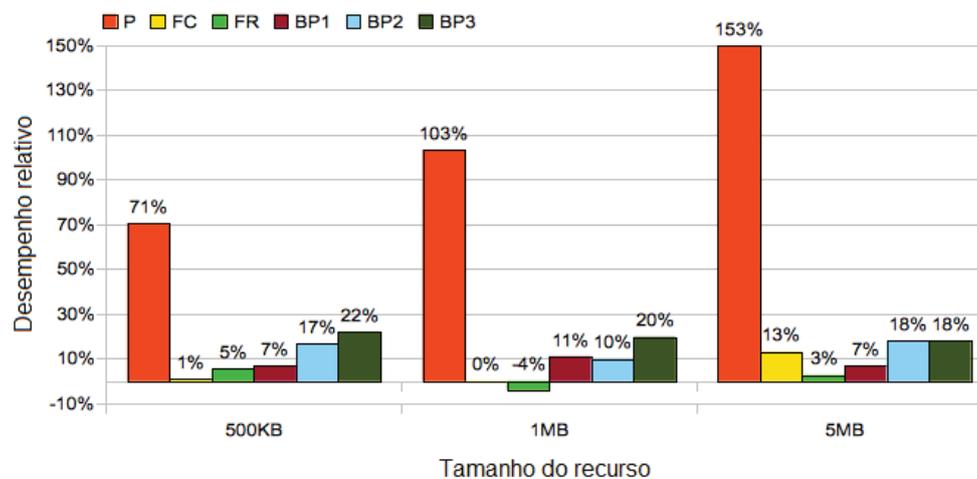
As requisições ao serviço replicado de cada experimento foram organizadas em ciclos e sessões, seguindo a mesma estratégia adotada em avaliações similares descritas na literatura [Junior et al. 2007, Mendonça et al. 2008]. Uma sessão era composta por três ciclos, onde cada ciclo correspondia à solicitação de um recurso de tamanho distinto, a saber, 500KB, 1MB e 5MB, respectivamente. As requisições dentro de cada ciclo foram realizadas em duas etapas: primeiramente, efetuavam-se requisições a cada servidor que compunha o conjunto de réplicas do serviço isoladamente e, logo em seguida, efetuavam-se requisições ao conjunto de réplicas como um todo, via o arcabouço WSSF, utilizando cada uma das quatro políticas de seleção avaliadas. No caso da política *Best Partial*, foram consideradas três variações, denominadas BP1, BP2 e BP3, cada uma apresentando um comportamento diferente em relação à velocidade de encerramento das conexões. A política BP1 atuava de forma mais agressiva, a política BP2 de forma moderada, e a política BP3 de forma mais conservadora. A Tabela 2 mostra os valores dos parâmetros  $k$  e  $f$  utilizados para configurar essas três variações da política *Best Partial* em cada ciclo dos experimentos.

Além do desempenho das quatro políticas de seleção de servidores efetivamente implementadas, foi considerado também o desempenho de uma política hipotética, denominada *Best Server*, que sempre escolheria o servidor com melhor tempo de resposta em um determinado momento. Os tempos de resposta atribuídos a essa política hipotética corresponderam ao menor tempo de resposta obtido entre as requisições isoladas ao grupo de réplicas realizadas no início de cada ciclo. Dessa forma, foi possível capturar o melhor desempenho esperado em cada ciclo de invocação, permitindo que os resultados da política *Best Server* fossem utilizados como base para avaliar o desempenho das outras políticas em relação ao da política “ideal”.

A seguir são apresentados os resultados dos três experimentos. Para melhor destacar as diferenças de desempenho entre as políticas avaliadas, foram traçados gráficos ilustrando seu desempenho relativo, tomando como referência os resultados da política *Best Server*. O desempenho relativo de cada política foi calculado pela fórmula  $((T_P \times 100) \div T_{BS}) - 100$ , onde  $T_P$  é o tempo de resposta mediano de uma das políticas implementadas e  $T_{BS}$  é o tempo de resposta mediano da política *Best Server*. Por questões de espaço, apenas os gráficos relativos ao desempenho das políticas no período comercial (de 8h às 18h, considerando o horário local no cliente) serão apresentados neste artigo. Os resultados referentes ao período não comercial, bem como uma análise qualitativa dos resultados de cada experimento, podem ser encontrados na versão estendida do trabalho [de Sousa 2010].

#### 4.2. Experimento 1

O objetivo deste primeiro experimento foi avaliar o comportamento das novas políticas propostas em um cenário no qual o servidor com o melhor desempenho é o mesmo que possui também os menores tempos de conexão e de latência de leitura. Para isso, foram



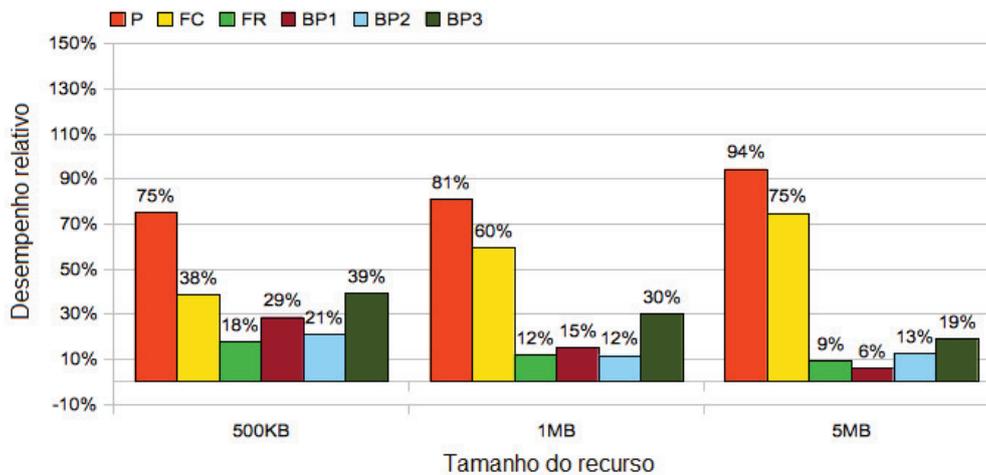
**Figura 1. Desempenho relativo das políticas de seleção de servidores em relação à política *Best Server* no Experimento 1.**

selecionados os servidores S1, S2, S3 e S4 (ver Tabela 1) para compor o conjunto de réplicas do serviço. Esse conjunto foi escolhido porque sabia-se, com base nos resultados do experimento inicial com todos os 136 servidores, que o servidor S4 possuía as menores medianas dos tempos de resposta, de conexão e de latência de leitura em comparação aos outros três servidores.

Ao todo, foram realizadas 13.170 requisições ao conjunto de réplicas, ocorrendo erro em apenas 3 delas (cerca de 0,04% do total de requisições). Isso mostra um elevado grau de disponibilidade do serviço, o que também pôde ser observado para os outros conjuntos de réplicas utilizados nos outros dois experimentos. O gráfico da Figura 1 ilustra os desempenhos relativos das seis políticas de seleção de servidores (incluindo as três variações da política *Best Partial*) avaliadas neste experimento.

De acordo com o gráfico, observa-se que a política de invocação paralela (P) obteve o pior desempenho para os três tamanhos de respostas. A política *First Connection* (FC) apresentou o melhor desempenho para as respostas de 500KB, vindo logo em seguida a política *First Read* (FR) e depois as três configurações da política *Best Partial* (BP), respectivamente, BP1, BP2 e BP3. Por outro lado, para as respostas de 1MB e 5MB, a política FR se destacou, alcançando o melhor desempenho. Em seguida, veio a política FC e as três configurações da política BP. Comparando apenas o desempenho das configurações dessa última, observa-se que a primeira apresentou o melhor desempenho em todos tamanhos de resposta. Entretanto, as configurações BP2 e BP3 não foram muito piores, visto que apresentaram desempenhos relativamente próximos ao desempenho da configuração BP1.

A política P apresentou uma perda relativa bem superior às demais políticas. Além disso, observou-se que à medida que o tamanho da resposta aumenta, cresce também a perda relativa desta política. Curiosamente, algumas políticas apresentaram perda relativa zero ou ligeiramente negativa, como a FC e a FR. Isto significa, neste cenário específico, que essas políticas obtiveram um desempenho tão bom quanto o desempenho de se invocar o melhor servidor isoladamente. Das três configurações da política BP, a BP1 foi a que apresentou a menor perda relativa para todos o tamanhos de resposta.



**Figura 2. Desempenho relativo das políticas de seleção de servidores em relação à política *Best Server* no Experimento 2.**

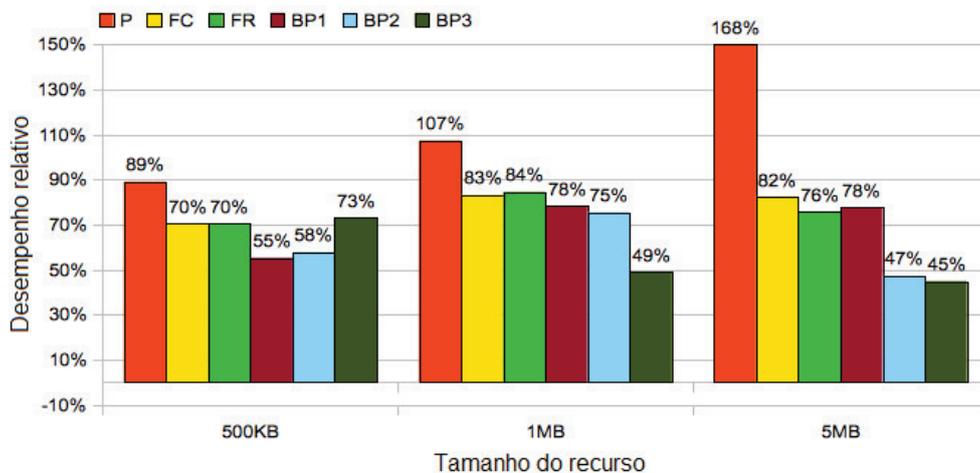
### 4.3. Experimento 2

O objetivo deste segundo experimento foi avaliar o comportamento das novas políticas propostas em um cenário heterogêneo em relação ao tempo de resposta percebido pelas aplicações clientes, onde existe claramente um servidor com o melhor desempenho, porém, homogêneo no tocante aos tempos de conexão e de latência de leitura. Para isso, substituiu-se o servidor S4 pelo servidor S5 no conjunto de réplicas selecionado para o primeiro experimento. Esta substituição foi feita porque conhecia-se, de antemão, que o novo servidor, assim como seu antecessor, possuía um menor tempo de resposta mediano em relação aos demais. No entanto, seus tempos de conexão e de latência de leitura não apresentavam destaque em relação aos outros servidores, formando um cenário homogêneo nesse sentido.

Ao todo, foram realizadas 6.930 requisições ao conjunto de réplicas, ocorrendo erro em apenas 2 delas (aproximadamente 0,02% do total de requisições). O gráfico da Figura 2 ilustra os resultados do segundo experimento. Neste gráfico, observa-se que as políticas P e FC apresentaram os piores desempenhos em todos os ciclos, enquanto, de um modo geral, o melhor desempenho foi atribuído às políticas FR e BP. Nas respostas mais curtas, as configurações BP1 e BP2 juntamente com a política FR obtiveram desempenhos bem próximos, vindo logo em seguida a configuração BP3. Além disso, verificou-se o aumento das perdas relativas das políticas P e FC à medida que cresceu o tamanho da resposta. Com relação às demais políticas, observou-se que ocorreu o efeito inverso, ou seja, a perda relativa diminuiu à medida que o tamanho da resposta aumentou. Vale ressaltar também que, independentemente do tamanho da resposta, a política FR e as configurações BP1 e BP2 apresentaram os melhores desempenhos, visto que suas perdas relativas foram as mais baixas.

### 4.4. Experimento 3

O objetivo deste terceiro experimento foi avaliar o comportamento das novas políticas quando o servidor com o melhor desempenho não possui os melhores tempos de conexão e de latência de leitura. Para isso, foram selecionados quatro novos servidores para compor o conjunto de réplicas, a saber: S6, S7, S8 e S9 (ver Tabela 1). Novamente, a escolha



**Figura 3. Desempenho relativo das políticas de seleção de servidores em relação à política *Best Server* no Experimento 3.**

desse grupo específico de servidores levou em conta os resultados do experimento inicial com todos os 136 servidores. Por exemplo, sabia-se que o servidor S9 possuía o menor tempo de resposta mediano em relação aos demais, no entanto, seus tempos de conexão e de latência de leitura são maiores que os tempos dos outros servidores escolhidos.

Foram realizadas um total de 4.530 requisições ao conjunto de réplicas neste experimento, ocorrendo erro em apenas 123 delas (cerca de 2,71% do total de requisições). O gráfico da Figura 3 ilustra os resultados do terceiro experimento. Neste gráfico, observa-se que, de um modo geral, as variações da política BP apresentaram os melhores desempenhos, embora a perda oferecida por essas políticas em relação ao desempenho da política *Best Server* tenha aumentado de forma significativa.<sup>1</sup> A configuração BP3 se sobressaiu nas respostas de 1MB e 5MB, enquanto, para as respostas mais curtas, as configurações BP1 e BP2 foram as melhores. Novamente, a política P apresentou o pior desempenho em todos os ciclos, com as políticas FR e FC apresentando o segundo pior desempenho para as respostas de 1MB e 5MB, respectivamente. Além disso, observou-se que a configuração BP3 apresentou perdas relativas inversamente proporcionais ao tamanho das respostas.

#### 4.5. Discussão

Com base nos resultados dos três experimentos, observou-se que a política de invocação paralela ofereceu de forma destacada o pior desempenho entre as políticas avaliadas, independentemente do cenário e tamanho da resposta. Vale salientar que foram escolhidos cenários com certas características que pudessem evidenciar as limitações dessa política, como, por exemplo, a restrição da largura de banda no cliente. O objetivo era submeter as novas estratégias de invocação paralela aos mesmos cenários, para avaliar se as limitações da política de invocação paralela clássica seriam de fato superadas.

Quanto à política *First Connection*, esta ficou entre as melhores políticas no primeiro experimento, apresentando desempenho equivalente ao da política *Best Server* nas

<sup>1</sup>Uma possível explicação para esse aumento é a grande flutuação observada entre os desempenhos individuais dos quatro servidores utilizados no terceiro experimento.

respostas com 500KB e 1MB de tamanho. Entretanto, essa política apresentou desempenho inferior nos demais experimentos, não permanecendo entre as três melhores políticas avaliadas. Atribui-se esses resultados ao fato do primeiro experimento ter sido o único a utilizar um cenário no qual o servidor com o melhor desempenho também era aquele que se conectava mais rápido à aplicação.

A política *First Read*, por sua vez, ficou entre as melhores no primeiro e segundo experimentos. Especialmente no primeiro experimento, esta política obteve um desempenho muito próximo ao da política *Best Server*, independentemente do tamanho do recurso. Por outro lado, essa política apresentou um desempenho bem inferior no terceiro experimento, onde o servidor com o melhor desempenho não oferecia um bom tempo de latência de leitura. Isso explica o fato da política *First Read* nem sempre ter selecionado o melhor servidor para este cenário.

Já a política *Best Partial* esteve sempre entre as três políticas que mais se aproximaram da política *Best Server*, independentemente do cenário ou tamanho da resposta. Configurada para atuar de forma mais conservadora, terceira variação dessa política (BP3) se sobressaiu no terceiro experimento, no qual as políticas *First Connection* e *First Read* obtiveram baixo desempenho. Nos demais experimentos, as configurações mais agressivas (BP1 e BP2) apresentaram melhores desempenhos em relação à BP3. Observou-se também que, de modo geral, à medida que as respostas vão crescendo, os resultados da política *Best Partial* tendem a se aproximar dos resultados da política *Best Server*. Conclui-se, portanto, que essa política é a mais indicada para cenários heterogêneos onde o tamanho dos recursos é um fator crítico no acesso às réplicas do serviço.

Finalmente, é importante destacar que, assim como a política paralela clássica, as novas políticas proposta tendem a gerar um aumento da demanda por todo o conjunto de servidores, devido à necessidade de monitorar o desempenho individual de cada conexão. No entanto, com as novas políticas essa demanda ocorre apenas no início do processo de invocação, sendo rapidamente reduzida à medida que as conexões mais lentas são encerradas.

## 5. Conclusão

Este trabalho propôs três novas políticas dinâmicas para seleção de servidores replicados, baseadas no princípio da invocação paralela ao conjunto de réplicas do serviço. As três políticas têm, como principal contribuição, a redução no nível de concorrência no acesso aos recursos replicados, e o fato de não dependerem da existência de informações atualizadas sobre o desempenho de cada réplica. Resultados obtidos experimentalmente, utilizando um serviço replicado real da Web, mostram que as novas políticas oferecem desempenhos comparáveis aos da escolha do melhor servidor previamente identificado para cada um dos três conjuntos de réplicas avaliados. Além disso, as novas políticas apresentam rendimento bem superior à política de invocação paralela clássica, com destaque para a política *Best Partial*, que se manteve entre as três melhores políticas em todos os cenários investigados.

Como tópicos para trabalhos futuros, sugere-se a realização de novos experimentos envolvendo um maior número de servidores e aplicações clientes com diferentes características de conexão e localização física. Outro tópico interessante seria investigar como a política *Best Partial* poderia ser estendida de modo a ajustar dinamicamente seus

parâmetros de configuração – por exemplo, em função do tamanho dos recursos requisitados ou de oscilações no desempenho dos servidores ou da rede. Por fim, mais políticas de seleção poderiam ser implementadas no arcabouço WSSF, que passaria a fornecer, nativamente, as principais estratégias de seleção de servidores descritas na literatura.

### Agradecimentos

Os autores agradecem a Leandro Sales Pinto, pela sua valiosa contribuição na fase inicial desta pesquisa.

### Referências

- Conti, M., Gregori, E., and Lapenna, W. (2005). Client-side content delivery policies in replicated web services: parallel access versus single server approach. *Perform. Eval.*, 59(2+3):137–157.
- de Sousa, W. E. Q. (2010). Implementação e avaliação de novas políticas paralelas de seleção de servidores replicados. Master's thesis, Universidade de Fortaleza.
- Dharmalingam, K. and Collier, M. (2003). Transparent and scalable clientside server selection using netlets. In *IEEE International Conference on Open Programmable Network Architectures, IEEE-OPENARCH*.
- Dykes, S. G., Dykes, R. G., Jeffery, C. L., and Robbins, K. A. (2000). An empirical evaluation of client-side server selection algorithms. In *IEEE INFOCOM 2000, The Conference on Computer Communications*, pages 1361–1370, Tel Aviv, Israel.
- Hanna, K. M., Natarajan, N., and Levine, B. N. (2001). Evaluation of a novel two-step server selection metric. In *Proceedings of IEEE International Conference on Network Protocols*, Califórnia, Estados Unidos.
- Junior, J. G. R., Carmo, G. T. S., Valente, M. T. O., and Mendonca, N. C. (2007). Smart proxies for accessing replicated web services. *IEEE Distributed Systems Online*, 8.
- Kangasharju, J., Roberts, J., and Ross, K. W. (2002). Object replication strategies in content distribution networks. *Computer Communications*, 25(4):376–383.
- Mendonça, N. C., Silva, J. A. F., and Anido, R. O. (2008). Client-side selection of replicated web services: An empirical assessment. *Journal of Systems and Software*, 81(8):1346 – 1363.
- Mozilla. Mozilla project [online]. (2010). Available from: <http://www.mozilla.org/>.
- Rodriguez, P. and Biersack, E. W. (2002). Dynamic parallel access to replicated content in the internet. *IEEE/ACM Transactions on Networking*, 10:455–465.
- Salas, J., Perez-Sorrosal, F., Pati No-Martínez, M., and JimÉnez-Peris, R. (2006). WS-Replication: a framework for highly available web services. In *WWW '06: Proceedings of the 15th international conference on World Wide Web*, pages 357–366, New York, NY, USA. ACM.
- Sayal, M., Breitbart, Y., Scheuermann, P., and Vingralek, R. (1998). Selection algorithms for replicated web servers. *SIGMETRICS Perform. Eval. Rev.*, 26(3):44–50.
- Tukey, J. W. (1970). *Exploratory Data Analysis*. Addison-Wesley.