

UNIVERSIDADE DE SÃO PAULO
Instituto de Ciências Matemáticas e de Computação

ISSN - 0103-2585

**AVALIAÇÃO DE DESEMPENHO DE
WEB SERVICES ATTACHMENTS**

**Júlio César Estrella
Rubens Kenji Takaki Toyohara
Regina Helena Carlucci Santana
Marcos José Santana
Sarita Mazzini Bruschi**

Nº 336

RELATÓRIO TÉCNICO



São Carlos
Fev/2009

Avaliação de Desempenho de Web Services Attachments

Júlio César Estrella, Rubens Kenji Takaki Toyohara, Regina Helena Carlucci
Santana, Marcos José Santana, Sarita Mazzini Bruschi

Departamento de Sistemas de Computação
Grupo de Sistemas Distribuídos e Programação Concorrente
Universidade de São Paulo - Instituto de Ciências Matemáticas e de Computação
{jcezar, kenji, rcs, mjs, sarita}@icmc.usp.br
<http://www.lasdpc.icmc.usp.br>

Resumo Os *Web Services* são uma tecnologia baseada na arquitetura orientada a serviços (*SOA*) que permitem a comunicação entre aplicações por meio da Internet. Utilizando *Web services*, é possível enviar informações de qualquer tipo, estejam estas em qualquer formato de codificação. Neste trabalho serão apresentados e avaliados em termos de desempenho três técnicas para anexos de dados em *Web Services: Pure Binary, MTOM* e *SwA*. A influência da rede e do tamanho de arquivos serão considerados nesta avaliação. Por fim, os resultados obtidos são analisados.

1 Introdução

A integração de sistemas heterogêneos, além da interoperabilidade entre tais sistemas, são considerados uns dos principais desafios presentes na computação. Para suprir tais necessidades, surgiu o conceito de orientação por serviços, consistindo basicamente de uma arquitetura composta por serviços distribuídos na rede, os quais podem ser combinados ou reusados para criar aplicações de negócios. A adoção desta arquitetura tem como principais objetivos o baixo acoplamento, reuso e interoperabilidade entre as aplicações envolvidas. Os *Web Services* são uma implementação deste conceito, disponibilizados em forma de serviços que podem ser facilmente utilizados por outras aplicações através de uma rede como a Internet. Assim, com a utilização de *Web Services*, aplicações podem se comunicar entre si por meio de especificações de padrões baseados no protocolo da Internet e de um formato de dados universal conhecido como *XML*. Dentre os principais padrões que compõem o conceito de *Web Services* estão *WSDL*, *SOAP*, e *UDDI*.

Os conceitos de arquitetura orientada a serviços (*SOA*) e de *Web Services* são tópicos bastante difundidos em TI e recentemente na academia, envolvendo diversas vertentes de pesquisa, entre eles: composição de serviços, otimização na busca de *Web Services* em registro de informações [5], [2], [12], [13], caracterização de carga de trabalho, *Web Services* com qualidade de serviços (*QoS*) [14], [18], entre outros. Assim como outros protocolos de comunicação, é possível enviar qualquer tipo de informação usando *Web Services*, esteja esta em qualquer formato de codificação. Para este fim, existem algumas técnicas que permitem o encapsulamento das informações em mensagens *SOAP*. Neste trabalho, serão abordados três técnicas de envio de mensagens com dados anexados usando *Web Services: Pure Binary, MTOM e SwA*.

No decorrer deste estudo, estas técnicas serão avaliadas e analisadas em termos de desempenho, verificando-se também a influência de diferentes redes e a influência de arquivos de tamanhos distintos. Este relatório técnico está organizado da seguinte maneira: na seção 2 serão introduzidos os principais conceitos de *SOA* e *Web Services* e na seção 3 as técnicas de *attachment* (anexo de dados) serão descritas com mais detalhes. Na seção 4 será definido o ambiente de testes onde os experimentos foram executados. O planejamento de experimentos, definindo fatores fixos e fatores que serão variados durante os experimentos, será abordado na seção 5. Na seção 6 os resultados obtidos serão analisados de acordo com a influência da rede e as técnicas serão comparadas entre si. Na seção 7 as principais dificuldades durante o desenvolvimento do trabalho serão discutidas. Por fim, nas duas últimas seções, serão apresentadas as conclusões e definidos alguns trabalhos futuros.

2 SOA e Web Services

A arquitetura orientada a serviços (*SOA - Service-oriented architecture*) pode ser definida basicamente como um conjunto de serviços que podem se comunicar

entre si através de passagem de mensagens baseadas no paradigma requisição-resposta [11]. Funcionalidades bem definidas e/ou lógicas de negócio de uma aplicação são modularizadas e apresentadas na forma de serviços disponibilizados pela arquitetura. A maioria das implementações de *SOA* utilizam-se de *Web Services*. Entretanto, *SOA* pode ser implementado utilizando qualquer outra tecnologia padronizada baseada na *Web*. Uma arquitetura básica consiste de um serviço consumidor (*service consumer*) e um provedor de serviços (*service provider*), onde o serviço consumidor faz uma requisição para o provedor de serviços, que, por sua vez, retorna um resultado para o serviço consumidor. Um provedor de serviços também pode ser um serviço consumidor. Para permitir comunicação entre os serviços envolvidos, a requisição e a resposta devem ser descritas de forma que ambos os serviços entendam. Para isso, padrões são utilizados para definir interfaces que descrevem as operações envolvidas entre os serviços. A figura 1 ilustra uma interação básica entre um serviço provedor e um serviço consumidor.

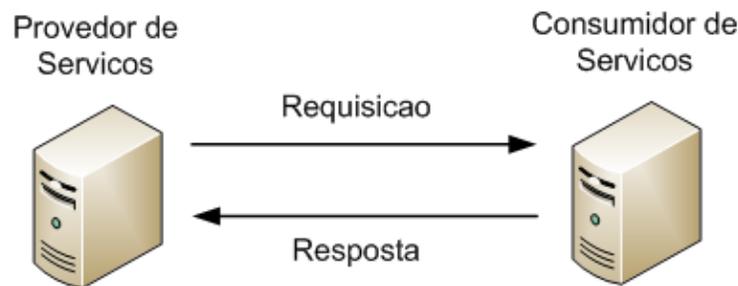


Figura 1. Interação Provedor-Consumidor [1]

Web Services são amplamente tidos como a maneira em que *SOA* deve ser realizada na prática [16]. Um *Web Service* é definido pela *W3C* (*World Wide Web Consortium*) como uma aplicação identificada por uma *URI* (*Uniform Resource Identifier*), cujas interfaces e ligações são definidas, descritas e descobertas utilizando-se uma linguagem padrão como XML (*Extensible Markup Language*). Pelas definições apresentadas, pode-se observar que diferentes explicações sobre *Web Services* estão disponíveis, desde as mais simples até as mais completas como a sugerida pela *W3C*. Os principais padrões de Internet utilizados para esta tecnologia são: *XML*, *WSDL*, *SOAP*, *UDDI*, e *HTTP*. Alguns foram modelados especificamente para serem utilizados entre *Web Services*. Outros são padrões que já existiam antes, mas também foram utilizados para definir o conceito de *Web Services*. Os *Web Services* ainda não possuem um desempenho muito eficiente em relação a outros tipos de protocolos de comunicação como *IOPP* ou *RMI*, porém, permitem interoperabilidade, ou seja, independência entre plataformas.

A interação de *Web Services* ocorre a partir de um protocolo de comunicações baseado em *XML* chamado *SOAP* (*Simple Object Access Protocol*). O padrão *WSDL* (*Web Service Description Language*) é utilizado para descrever a interface de um *Web Service*. A descrição inclui detalhes como definição de tipos de dados, operações suportadas pelo *Web Service*, formato das mensagens de entrada e saída, entre outros.

3 Web Services attachments

Existem duas maneiras de se anexar informações em documentos *SOAP/XML*:

- Armazenar estas informações na própria mensagem *SOAP*. Esta técnica é conhecida como *Pure Binary*. É necessário converter estes dados para modo texto usando o tipo `xs:base64Binary` ou `xs:hexBinary`, tipos de dados pré-definidos do XML para referenciar codificação base64 em modo texto e modo binário. A principal desvantagem desta abordagem é que este encapsulamento de dados no envelope *SOAP* causa *overhead* no custo de processamento, principalmente ao decodificar a mensagem para o modo binário.
- Usar uma *URI* do tipo `xs:anyURI` no documento *XML* para referenciar os dados binários. As informações são anexadas fora da mensagem *SOAP*, sendo identificadas por uma *URI*. Tais dados são empacotados usando mecanismos como *DIME* ou *MIME* (os mesmos utilizados para anexos por e-mail). Neste trabalho são apresentadas duas técnicas que empacotam dados externamente à mensagem *SOAP*: *SwA* e *MTOM*.

3.1 MTOM

MTOM (*Message Transmission Optimization Mechanism*) é um método que envia dados binários de forma eficiente entre *Web Services*. Sua base é o **XOP** (*XML-binary Optimized Packing*), um protocolo usado para transmitir dados binários na mensagem *SOAP*. *XOP* é uma alternativa de serialização ¹, para se comportar de forma semelhante a um pacote *MIME* de múltiplas partes, sendo o documento *XML* o elemento raiz. Este documento difere de mensagens *SOAP* usando *Pure Binary* pelo fato dos dados serem substituídos por referências de partes *MIME*, que não são codificados em *base64*.

O processo utilizado pelo protocolo *XOP* consiste nos seguintes passos:

1. *XOP* extrai o dado binário em um documento *XML*;
2. *XOP* serializa documentos *XML* em pacotes localizados dentro de formatos de empacotamento extensivo, tal como *MIMEType*.
3. O novo documento *XML* marca a localização destes pacotes de dados binários em elementos especiais que contém *URI*, identificando cada localização correspondente.

¹ Processo de transformação de objetos Java, sejam em formato binário, ou em formato de texto como o XML, para posterior armazenamento em disco, buffer de memória ou transmissão na rede de comunicação

MTOM especifica uma implementação concreta deste mecanismo para o envio de pacotes *SOAP* no protocolo *HTTP*, e usa o mecanismo *XOP* para otimizar a transmissão de mensagens *SOAP* com anexos. No documento *XML* é descrito como serializar o envelope *SOAP* usando o formato *XOP* e o empacotamento *MIME Multipart/Related* [15]. *XOP* e *MTOM* são padrões *W3C* desde 2005, mas não são propostos como parte da especificação original de *SOAP*.

Para identificar *MTOM* na transmissão da mensagem *SOAP*, são necessários quatro atributos [17]:

1. A mídia `application/xop+xml` é obrigatória.
2. O tipo de mídia da mensagem *HTTP* é `multipart/related`.
3. O tipo de mídia para o elemento raiz do pacote *MIME multipart/related* é `application/xop+xml`
4. O parâmetro `start-info` indica o *content-type* de `application/soap+xml`.

Assim, as referências de objetos binários encontram-se agrupadas em um *infonet XML* dentro de um documento *SOAP*. A figura 2 ilustra uma abstração do mecanismo *MTOM* usando empacotamento de dados em *XOP*.

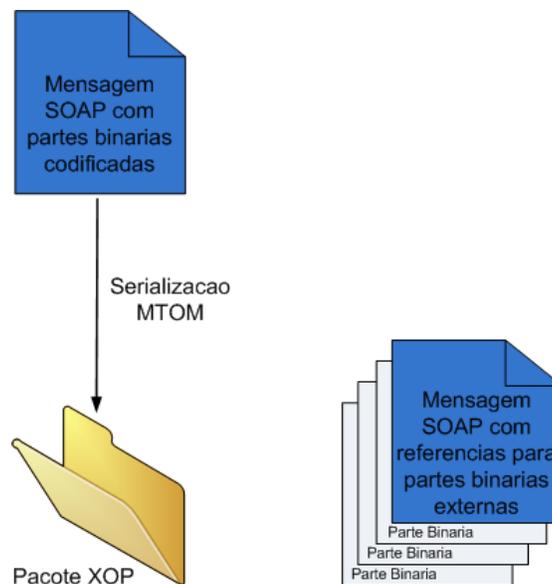


Figura 2. MTOM [9]

3.2 SwA

A técnica SwA define como uma mensagem *SOAP* pode ser composta em uma mensagem *MIME* do tipo `multipart/related`. Cada parte da mensagem é separada por um único *string* delimitador definido no começo da mensagem [3]. A

diferença entre *MTOM* e *SwA* é que o *SwA* não mostra a referência do objeto como um *infoset XML*, ao contrário que o *MTOM* faz. Sendo assim, a representação do conteúdo binário anexado não está em formato *XML*. A figura 3 ilustra um envelope *SOAP* anexando uma imagem no formato *png*.

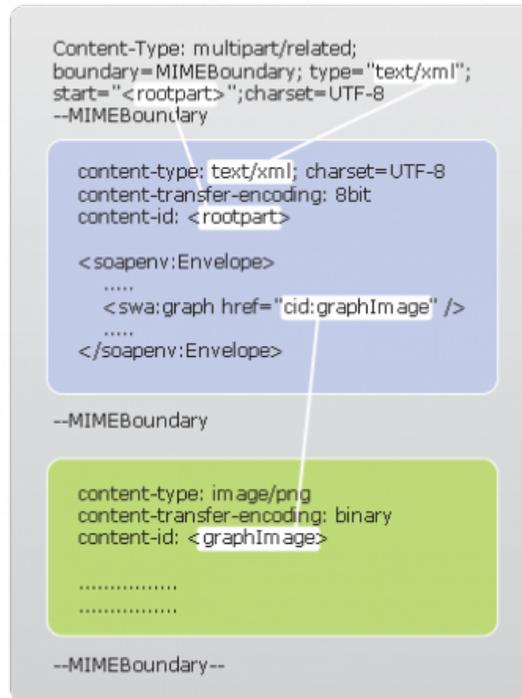


Figura 3. SWA [8]

Este arquivo se encontra fora do envelope *SOAP*, delimitado pelo mecanismo *MIMEType*, e possui um *ID* que é referenciado pelo campo *href* da mensagem *SOAP*.

A vantagem do *SwA* é que não é necessário converter dados binários para caracteres usando algoritmos de codificação binária. Entretanto, um dos principais problemas no modelo *SwA* é não ser compatível com alguns conceitos relacionados aos *Web Services*, pelo fato de não utilizar o formato *XML* para tratar do anexo de dados. Por exemplo, o *SwA* não é compatível com o formato *WS-Security*, dificultando o envio de dados binários anexados de forma segura, principalmente quando se trata de troca de dados entre múltiplas organizações.

4 Ambiente de testes

O principal objetivo deste trabalho é avaliar técnicas de transferência de arquivos em *Web Services* sobre a influência de redes diferentes. Assim, dentre os elementos que compõem o ambiente de testes, incluem duas máquinas: uma representando um cliente e outra um servidor (provedor de *Web Services*). Uma terceira máquina se comporta como um *gateway*, responsável pelo controle de banda da transmissão de arquivos entre o cliente e o servidor. Em nosso ambiente, o cliente e o servidor se encontram em redes distintas, e para permitir a comunicação entre eles, o *gateway* utiliza duas interfaces de rede (duas placas de rede *gigabit*) e um *firewall*. A figura ilustra a interação entre o cliente, *gateway*, e servidor.

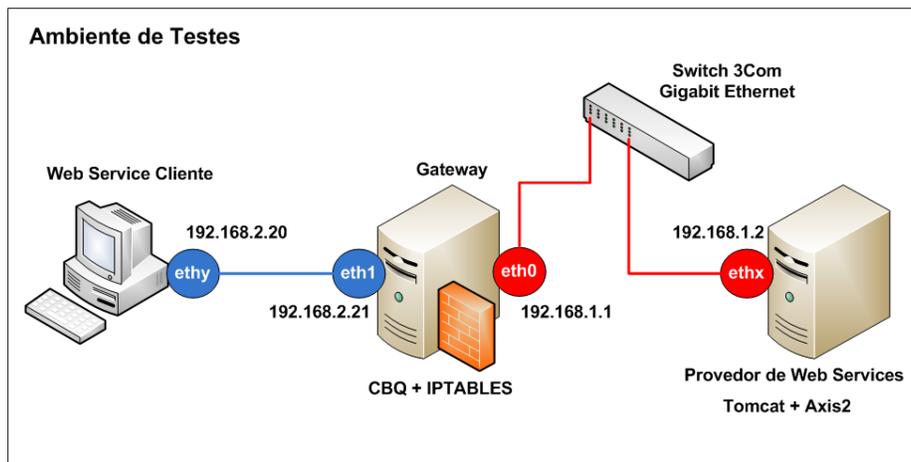


Figura 4. Ambiente de testes

As máquinas utilizadas são homogêneas, ou seja, possuem a mesma configuração de *hardware*, como mostra a tabela 1.

Tabela 1. Informações de hardware para a configuração do ambiente de testes

Sistema Operacional	Ubuntu Linux 8.04
CPU	Intel Core 2 Quad – 64 Bits – 2.4 GB
Memória	2GB
HD	250GB
Switch	3Com Gigabit Ethernet
Placas de Rede	Realtek Gigabit Ethernet

Além disso, algumas ferramentas e tecnologias são utilizadas para o desenvolvimento do ambiente de experimentos, dentre eles:

- **WSATPerf**: Uma ferramenta que foi desenvolvida para simular o comportamento de um cliente e um provedor de serviços sob a ótica de attachment de dados. Esta ferramenta coleta todos os dados necessários para a avaliação dos experimentos.
- **Apache Tomcat**: Responsável por disponibilizar os serviços no provedor de serviços.
- **Apache Axis2**: Motor de processamento de mensagens *SOAP*.
- **Iptables**: Usado neste contexto para permitir que duas redes conversem entre si de forma transparente via *NAT (Network Address Translation)*.
- **CBQ/iproute**: Responsável pelo controle de banda no *gateway*. *CBQ* [6] é um algoritmo de gerenciamento de tráfego desenvolvido pelo grupo de pesquisa de redes de computadores do *LBNL (Lawrence Berkeley National Laboratory)* [7].
- **GenFile**: Um gerador para criar arquivos XML utilizados nos experimentos. Este gerador é capaz de gerar arquivos de tamanhos específicos escolhidos pelo usuário [3].

O processo de execução dos experimentos foi realizado utilizando a ferramenta *WSATPerf* implementada especificamente para avaliar dados anexados em mensagens *SOAP* sob diversas perspectivas. Tal ferramenta é parametrizável, de forma a permitir a configuração de vários fatores como mostra a tabela 2.

Tabela 2. Fatores customizáveis da ferramenta de avaliação

Número de Threads	1 à 12
Número de Interações por Thread	1 à 12
Tipo de operação	download, upload
Técnica	pure binary, swa, mtom
Tipo de Arquivo	texto, binário (imagem, audio, video)
Tamanho do Arquivo	(10, 100)KB, (1, 5, 10)MB
Taxa de Transferência da Rede	100kbps, (1, 10, 100)mbps
Existência de Cache	sim, não

5 Planejamento de experimentos

A execução dos experimentos é feita variando três fatores, de forma a permitir a verificação da influência em relação às técnicas de attachments, rede e tamanho de arquivos. A parametrização destes fatores é definida na tabela 3.

Tabela 3. Parametrização dos fatores utilizados nos experimentos

Técnicas	<i>pure binary, swa, mtom</i>
Tamanho do Arquivo	<i>10KB, 100KB, 1MB, 10MB</i>
Largura de Banda da Rede	<i>100Kbps, 1Mbps, 10Mbps, 100Mbps</i>

Outro ponto fundamental a ser considerado na fase de planejamento de experimentos é a definição dos fatores fixos que devem ser considerados na parametrização da aplicação utilizada. Tais fatores são evidenciados na tabela 4.

Tabela 4. Valores fixos associados a alguns parâmetros utilizados nos experimentos

Operação	<i>upload</i>
Número de Threads	<i>12</i>
Número de Iterações por Thread	<i>12</i>
Número de Arquivos Enviados	<i>01</i>
Timeout de Conexão	<i>4 minutos</i>
Java Heap Space	<i>1024MB</i>
Tipo de Arquivo	<i>texto</i>

Sendo assim, para cada experimento, o cliente dispara 12 *threads*, no qual cada *thread* envia um arquivo por vez (no total de 12 arquivos) para o provedor de serviços. Para evitar excesso de perda de requisições, foram definidos um *timeout* de 4 minutos e um espaço de pilha *Java* de 1GB, pois em algumas técnicas a influência da memória é bastante evidente. Estes parâmetros foram definidos tanto no lado do cliente quanto no lado do provedor. Os arquivos de tamanhos específicos foram gerados de forma aleatória em formato texto. A análise de resultados foi feita baseada no tempo médio de resposta de cada *thread*. Somente são considerados os tempos de resposta de requisições bem sucedidas, ou seja, resultados obtidos por meio de requisições não-completadas não são consideradas no cálculo do valor médio do tempo de resposta. Entretanto, estas requisições mal sucedidas são consideradas como variáveis de resposta para gráficos que demonstram o comportamento das *threads*.

De maneira mais detalhada, em cada *thread*, é calculado o tempo médio de resposta de cada iteração com requisição aceita. Deste modo, são obtidos os tempos de resposta médio de cada *thread*. Assim, a variável de resposta é obtida calculando-se a média dos tempos de resposta médio obtido por cada *thread*, resultando também o desvio padrão, a variância, tempos de resposta máximo e mínimo, e o intervalo de confiança, com 95% de grau de confiança. Para a apresentação dos resultados, 10 foram realizadas para cada experimento. Isso permite maior confiabilidade para a análise dos resultados.

6 Análise de resultados

Nesta seção, são analisados a rede e as técnicas de attachments com base nos resultados obtidos durante os experimentos. A influência da rede foi verificada através de gráficos de bloxpot, onde cada coluna caracteriza um nível de largura de banda da rede. Também foi fixado um nível de tamanho de arquivo e uma técnica específica para cada gráfico. A influência das técnicas de attachment foi verificada de forma semelhante, mas usando técnicas representando cada coluna, em vez da largura de banda da rede. O tempo de resposta médio é ilustrado por cada uma das caixas do gráfico bloxpot, onde a linha divisória da caixa representa o tempo médio em si e o limite superior e inferior desta caixa representam o intervalo de confiança. A Figura 5 mostra uma legenda que resume a forma como os resultados apresentados nos gráficos das figuras 6 e 7 devem ser interpretados.

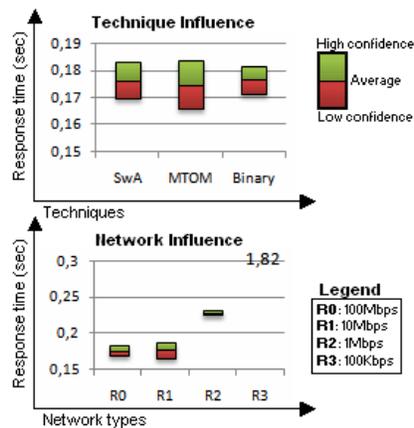


Figura 5. Legenda usada para a interpretação dos gráficos

6.1 Influência da rede

Na Figura 6 os diferentes tipos de redes considerados, em conjunto com diferentes tamanhos de arquivos foram analisados. Os resultados mostram que a influência para redes (100Kbps) de menor largura de banda é mais evidente. Além disso, para uma rede de maior largura de banda, não há diferença perceptível no tempos médios de resposta entre as técnicas. Assim, devido à alta velocidade de transmissão na rede, este fator não é um fator limitante, sendo a variável de resposta mais influenciada em relação a outros fatores como o processamento das mensagens SOAP.

Há também algumas diferenças entre as redes de 1Mbps e 10Mbps. Esta diferença torna-se um pouco mais definidos quando o tamanho do arquivo cresce. Isto mostra que ainda há uma certa influência sobre a rede no tempo de resposta

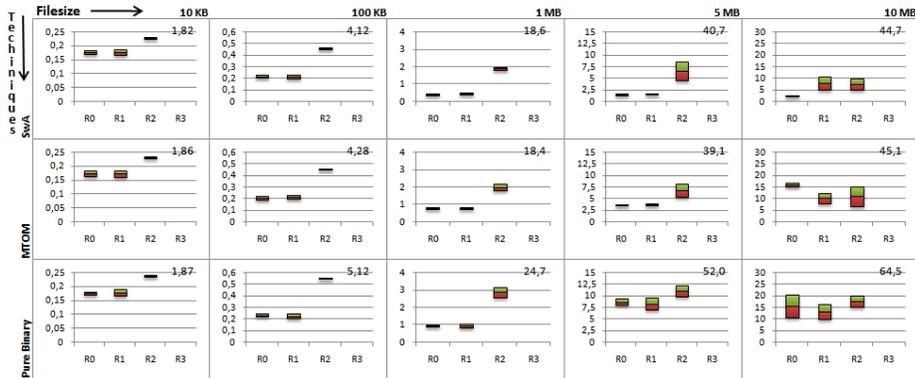


Figura 6. Influência da rede no tempo de resposta

das mensagens, mas menor do que em 100Kbps e 1Mbps redes. Além disso, é possível notar uma nítida influência sobre o tamanho do arquivo, devido à variação entre os diferentes tamanhos de arquivos. Outro ponto que é importante a considerar é o comportamento anormal de arquivos de 10MB em nossos experimentos. Isso acontece devido à influência de outros fatores como o uso excessivo de memória. Este fato é melhor explicado na próxima seção.

6.2 Influência das técnicas

As principais características que podem afetar o desempenho das técnicas avaliadas são o tempo de processamento das mensagens *SOAP* para identificar o conteúdo anexado, e a memória utilizada para armazenamento de requisições em espera que aguardam a liberação do processamento. Tais fatores podem ser percebidos pela maneira em que uma determinada técnica pode anexar seu arquivo na mensagem *SOAP*. Por exemplo, enquanto o *MTOM* e *SwA* utilizam referências usando *URI* para anexar as informações, o *Pure Binary* anexa as informações no corpo `<SOAPBody>` da mensagem *SOAP*. O processo de conversão *base64* utilizado em *Pure Binary* para incluir o conteúdo dentro da mensagem *SOAP*, e converter de volta para o formato original, envolve um custo maior de processamento em relação às outras duas técnicas.

Além disso, pode-se notar diferenças menos perceptíveis, mas evidentes em relação às técnicas *MTOM* e *SwA*, apesar de ambas anexarem as informações fora da mensagem *SOAP*. *MTOM* é uma técnica criada para resolver problemas de incompatibilidade em relação ao *SwA*, incluindo as referências de arquivos anexados em um *infoset XML*. Porém, este processo também pode representar um certo custo de desempenho, tanto de processamento, quanto de memória utilizada. Para arquivos grandes, o *MTOM* divide o arquivo em porções menores, referenciando cada porção no *infoset XML*. Entretanto, tais porções do arquivo podem não chegar na mesma ordem em que foram referenciadas, o que obriga a utilização de um *buffer*. O processo de serializar partes do arquivo implica

em custo de processamento e também a alta utilização de memória devido à necessidade de *buffers* [4]. Este comportamento é ilustrado nas figuras 7, em que cada coluna representa uma largura de banda de rede. Isto é mais evidente em arquivos maiores, onde a técnica *Pure Binary* obteve o pior desempenho. Os experimentos com *SwA* apresentaram os menores tempos médio de resposta. Em arquivos de tamanho menor, não houve variações de tempo médio de resposta, pois exigem menos tempo de processamento em relação aos arquivos maiores.

Nos gráficos apresentados na Figura 7 é possível verificar uma grande variação no intervalo de confiança para a rede de 100Kbps. Isto ocorre devido a uma grande rajada de requisições em uma rede de baixa capacidade de transmissão, o que torna menos visível a diferença entre as técnicas. Neste caso, a velocidade da rede é mais influente que a tempo de processamento de mensagens

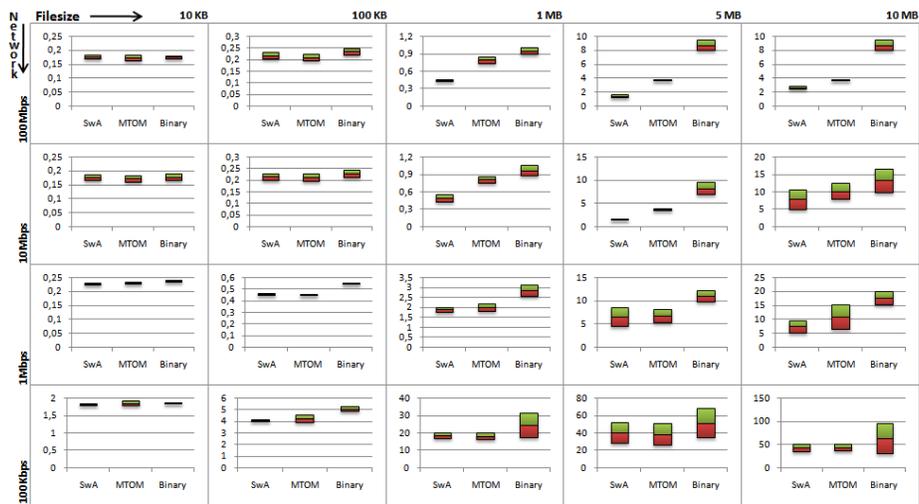


Figura 7. Influência das técnicas no tempo de resposta

A tabela 5 indica a percentagem média de requisições que falharam para cada experimento realizado. Estes valores são calculados utilizando o número de perda de pacotes entre as 144 requisições. Repare que as falhas são mais concentradas na rede de alta velocidade com arquivos maiores. Isto acontece devido ao aumento do fluxo de dados a partir da alta taxa de transferência de arquivos. Consequentemente, o provedor de serviços necessita de mais memória para alocar as requisições em espera para a liberação de recursos para o processamento. Este problema é mais evidente com técnicas de attachments que requerem mais tempo de processamento de mensagens, principalmente a técnica *pure binary*. Algumas falhas também podem ocorrer quando a rede apresenta baixa velocidade de transmissão. A grande quantidade de requisições somada ao baixa fluxo de dados pode causar perdas de pacotes durante a transmissão dos arquivos.

Tabela 5. Erros/Acertos

Falhas	100Mbps			10Mbps		
	BIN	MTOM	SwA	BIN	MTOM	SwA
10KB	0%	0%	0%	0%	0%	0%
100KB	0%	0%	0%	0%	0%	0%
1MB	0%	0%	0%	0%	0%	0%
5MB	2,84%	0%	0%	4,16%	0%	0%
10MB	81,11%	51,80%	0%	61,59%	2,43%	1,66%
	1Mbps			100Kbps		
	BIN	MTOM	SwA	BIN	MTOM	SwA
10KB	0%	0%	0%	0%	0%	0%
100KB	0%	0%	0%	0%	0%	0%
1MB	0%	0%	0%	1,66%	0%	0,06%
5MB	1,80%	0,06%	0,13%	10,27%	2,36%	3,19%
10MB	24,37%	2,15%	2,15%	26,73%	4,65%	5,06%

7 Dificuldades

Dentre as principais dificuldades encontradas no decorrer do desenvolvimento do projeto destaca-se:

- Desenvolvimento da aplicação - devido a dificuldade de encontrar materiais adequados para a construção de aplicações *Web Services* com *Axis2*
- Configuração do ambiente de testes (principalmente a configuração das regras utilizadas pelo *CBQ* e *IPTables*)
- Entender a diferença entre as técnicas, devido a falta de materiais disponíveis sobre a comparação entre as técnicas
- Identificar falhas nas aplicações, conseqüentemente gerando resultados inconsistentes, e corrigi-los a tempo

8 Conclusões

Neste trabalho foram apresentadas e avaliadas três técnicas para o envio de mensagens SOAP com dados binários anexados: Pure Binary Pura, MTOM e SWA. Essas técnicas foram avaliadas considerando a influência da rede e de diferentes tamanhos de arquivos. Um ambiente de testes foi configurado para permitir a execução dos experimentos. Também foi apresentada a ferramenta WSATPerf para dar suporte ao estudo. Nos experimentos, foram encontradas mais evidências no que diz respeito a influência da rede, quando a largura de banda é menor, principalmente nos experimentos em que foram utilizados arquivos de tamanho grande. Em situações com maior largura de banda de rede, há pouca diferença

entre o tempo de resposta dos experimentos. Dentre as técnicas, os experimentos usando a técnica Pure Binary apresentaram os piores resultados. Além disso, a técnica SWA foi a mais eficiente, uma vez que possui o menor o tempo de resposta. Há também uma maior concentração de falhas em redes mais rápidas usando arquivos maiores. Sob tais condições, presume-se que tais fatores requerem mais memória para alocar requisições em espera para serem processados, resultando em perdas de requisições, se as requisições não podem ser processadas pelo provedor de serviço, ou seja, se os arquivos anexados às mensagens SOAP, não foram escritos no disco do provedor de serviços.

9 Trabalhos futuros

Como um trabalho futuro a pretensão é continuar o desenvolvimento da ferramenta WSATPerf, construir uma interface que monitore em tempo real o cliente e provedor. Além disso, a ferramenta poderá analisar outros fatores que influencie significativamente a transmissão de arquivos através de Web Services, como por exemplo:

- Número de threads por cliente
- Uso de clientes distintos em diferentes máquinas ou dispositivos (telefone celular, palm, etc.)
- Influência da cache, tanto do lado do fornecedor de serviços como do lado do cliente,
- Os tipos de arquivos
- Influência da compressão/descompressão de mensagens SOAP.
- Analisar métricas de desempenho do lado do provedor de serviços, envolvendo parâmetros como:
 - Threads – Sucesso/Falha
 - Throughput
- Tempo de processamento da requisição para o Web Service presente no provedor
- Tempo de empacotamento da mensagem SOAP

Além disso, deve ser incluída a técnica Flex-SWA [10] para ser avaliada juntamente com as técnicas SWA, MTOM e Pure Binary.

Referências

1. J. Bih. Service oriented architecture (soa) a new paradigm to implement dynamic e-business solutions., 2006. Ubiquity, New York, p.1-1.
2. M. B. Blake, A. L. Sliva, M. zur Muehlen, and J. V. Nickerson. Binding now or binding later: The performance of uddi registries. In *HICSS '07: Proceedings of the 40th Annual Hawaii International Conference on System Sciences*, page 171c, Washington, DC, USA, 2007. IEEE Computer Society.
3. J. C. Estrella. Real time compression of soap messages in a soa environment. *ZIGDOC '08: Proceedings of the 26th Annual ACM International Conference on Design of Communication*, pages 163–168, 2008.

4. J. Evdemon. More thoughts on mtom., 2005. Disponivel em: <http://blogs.msdn.com/jevdemon/archive/2005/05/07/415321.aspx>. Ultimo acesso em: 22 nov 2008.
5. P. Farkas and H. Charaf. Web services planning concepts. *Journal of WSCG*, 11(1), February 2003.
6. S. Floyd. Link-sharing and resource management models for packet networks. *IEEE/ACM Transactions on Networking*, 3:365–386, 1995.
7. S. Floyd, C. Leres, V. Paxson, V. Jacobson, K. Fall, and S. McCanne. Network research group of lawrence berkeley national laborator, 2009. <http://www-nrg.ee.lbl.gov/>.
8. T. Gunarathn. Using soap with attachments in apache axis2, 2007. <http://wso2.org/library/1148>.
9. H. Hass. Foundations and future directions of web services, 2005. <http://www.w3.org/2005/Talks/1114-hh-ecows/>.
10. S. Heinzl, M. Mathes, T. Friese, M. Smith, and B. Freisleben. Flex-swa: Flexible exchange of binary data based on soap messages with attachments. In *ICWS '06: Proceedings of the IEEE International Conference on Web Services*, pages 3–10, Washington, DC, USA, 2006. IEEE Computer Society.
11. R. R. Kodali. What is service-oriented architecture?, 2005. Disponivel em: <http://www.javaworld.com/javaworld/jw-06-2005/jw-0613-soa.html>. Ultimo acesso em: 22 nov 2008.
12. J. Liu, N. Gu, Y. Zong, Z. Ding, and Q. Zhang. Service registration and discovery in a domain-oriented uddi registry. In *CIT '05: Proceedings of the The Fifth International Conference on Computer and Information Technology*, pages 276–283, Washington, DC, USA, 2005. IEEE Computer Society.
13. J. Liu, J. Liu, and L. Chao. Design and implementation of an extended uddi registration center for web service graph. *Web Services, 2007. ICWS 2007. IEEE International Conference on*, pages 1174–1175, July 2007.
14. D. Menasce and M. Woodside. Application qos level. *IEEE Internet Computing*, 10(3):13–15, May-June 2006.
15. M. Nottingham. Xop and mtom, 2004. Disponivel em: <http://www.mnot.net/blog/2004/02/14/xop>. Ultimo acesso em: 12 dez 2008.
16. M. P. Papazoglou. Service-oriented computing: Concepts, characteristics and directions. In *Proceedings of the Fourth International Conference on Web Information Systems Engineering (WISE-03)*. IEEE CS Press, 2003.
17. Y. Yang. Faster data transport means faster web services with mtom/xop, 2007. Disponivel em: <http://www.devx.com/xml/Article/34797>. Ultimo acesso em: 12 dez 2008.
18. G. Yeom and D. Min. Design and implementation of web services qos broker. In *Proceedings of the International Conference on Next Generation Web Services Practices (NWeSP'05)*. IEEE CS Press, 2005.