

Avaliação de Desempenho de SOAP sobre HTTP, SMTP e BEEP

Cláudio Schepke¹, Samuel Souza¹, Vinicius Viana²

¹Faculdade Três de Maio – SETREM. Tecnologia em Redes de Computadores – Av:
Santa Rosa, 2405 – Três de Maio – RS – Brasil

²Instituto de Informática – Universidade Federal do Rio Grande do Sul (UFRGS)
Caixa Postal 15.064 – 91.501-970 – Porto Alegre – RS – Brasil

cschepke@inf.ufrgs.br, vinicius@inf.ufrgs.br, muvuca_rs@hotmail.com

Resumo. Webservices são serviços de rede voltados à aplicação. Uma das tecnologias existentes em webservices é o protocolo SOAP, responsável pela troca de informações entre dois computadores, o qual pode ser facilmente implementado em diversas linguagens de programação, além de poder usar diferentes protocolos de transmissão. Neste sentido, este trabalho tem por objetivo comparar o desempenho de SOAP através do encapsulamento do seu conteúdo utilizando HTTP, SMTP e BEEP. Os resultados obtidos demonstram que tanto HTTP, quanto BEEP apresentam custos mais baixos de transmissão, cada qual se sobressaindo ao outro, dependendo das características referentes ao tipo de comunicação estabelecida.

1. Introdução

Webservices oferecem um novo paradigma para a construção de aplicações web distribuídas [Cerami 2002]. Qualquer serviço disponibilizado sob a Internet que utiliza um sistema de comunicação baseado em XML pode ser considerado um webservice [Elizabeth Castro 2001], independente do sistema operacional e linguagem de programação utilizado [Cerami 2002]. A idéia do modelo consiste em centralizar as operações em torno da aplicação no servidor, diferentemente do que ocorre em sistemas tradicionais, onde a implementação é centralizada no cliente. Exemplos de aplicações Web para os quais esta proposta é uma boa solução são a verificação de cartões de crédito, tradutores de linguagem e atualização de pacotes e repositórios de dados.

A arquitetura de um webservice é definida segundo a pilha de protocolos abaixo:

- Serviço de transporte - é a camada de transporte das mensagens entre as aplicações, podendo, para tanto, usar diferentes protocolos como HTTP, BEEP e SMTP;
- Mensagens XML - é a camada de codificação das mensagens para um formato XML, processo feito pelo SOAP;
- Descrição de serviços - é a parte da pilha responsável pela descrição de uma interface pública para um webservice específico. Manipulada via WDSL (*Web Services Description Language*);

- Descoberta de serviços - é a camada onde é feita a centralização dos serviços em um registro comum, disponibilizando recursos para a publicação e localização de funcionalidades. Manipulada via UDDI (*Universal Discovery Description and Integration*).

A Figura 1 ilustra a pilha de protocolos utilizados em aplicações webservices.

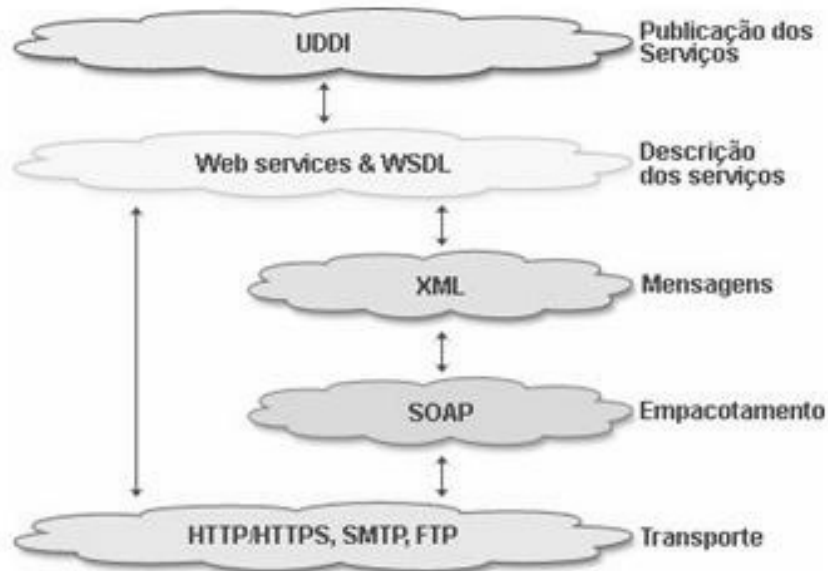


Figura 1. Pilha de Protocolos Webservices

A flexibilidade da organização de webservices permite a utilização de diferentes protocolos de comunicação. Neste sentido, este trabalho propõe-se a avaliar algumas implementações de SOAP, um protocolo para a transmissão de mensagens XML, capaz de usar diferentes protocolos de transporte como substrato. Para tanto, são comparadas implementações de SOAP com HTML, SMTP e BEEP, analisando os custos de comunicação para cada um dos casos.

Este trabalho está dividido em 7 seções. A próxima seção apresenta alguns trabalhos relacionados. Já a sessão 3 define o protocolo SOAP. Na sequência são apresentados os protocolos de transporte adotados no estudo na seção 4. A seção 5 descreve as implementações realizadas, bem como a forma de avaliação adotada. Na seção 6 são demonstrados os resultados obtidos e uma avaliação dos mesmos. Por fim, na seção 7, são relatadas as conclusões obtidas.

2. Trabalhos relacionados

Em [Litoiu 2002] são questionados os modelos tradicionais de implementação. Nesse artigo são demonstrados diferentes resultados de testes de performance para protocolos de transferência. São abordados principalmente resultados sobre latência e escalabilidade em uma implementação APACHE-SOAP. No entanto, não foram feitos comparativos entre os protocolos testados.

Em [Werner et al 2005] são demonstrados inicialmente os protocolos de transporte compatíveis e existentes. Posteriormente é implementado um webservices

sobre o protocolo UDP utilizando IP multicast. Novamente, os protocolos de transporte são tratados, mas sem uma abrangência maior.

3. SOAP

SOAP *Simple Object Access Protocol* é um protocolo leve para a troca de informações estruturadas em um ambiente distribuído e descentralizado [Martin Gudgin et al 2003]. SOAP utiliza a tecnologia XML para definir uma estrutura extensível de troca de mensagens, fornecendo uma construção que pode usar uma variedade de protocolos subjacentes. A estrutura foi desenvolvida para ser independente de qualquer modelo de programação e de implementações cuja semântica seja específica.

Entre os objetivos de SOAP estão a simplicidade e a extensibilidade. Para tanto, são omitidas características normalmente encontradas em sistemas distribuídos, tais como questões relacionadas à confiabilidade, segurança, correlação, roteamento e testes nos diferentes padrões de troca de mensagens (*Message Exchange Patterns* - MEPs), entre outras. Embora a estrutura busque definir muitas características, são definidas apenas duas MEPs. Outras características devem ser definidas como extensões por outras especificações.

A estrutura de comunicação de SOAP consiste em:

- Modelo de processamento - SOAP fornece um modelo de processamento distribuído no qual uma mensagem enviada pode passar por zero ou mais intermediários. Esse modelo especifica como um receptor SOAP processa uma mensagem de forma isolada em relação a qualquer outra mensagem, não fazendo nenhuma correlação entre elas. No caso do envio de múltiplas mensagens em sequência que dependem da resposta da mensagem precedente, é de responsabilidade de cada um definir um processamento combinado. O modelo pode suportar diferentes MEPs, incluindo mensagens de sentido único, interações do tipo requisição/resposta e conversação ponto-a-ponto.

- Modelo de extensibilidade - SOAP provê a extensão do ambiente de comunicação. Desta forma, as características omitidas para tornar o sistema mais simples podem ser estendidas, possibilitando, por exemplo, a criação de novos modelos de comunicação.

- Estrutura de ligação do protocolo - Através da estrutura de ligação de SOAP é possível utilizar diferentes protocolos subjacentes. *Binding* é conjunto de regras formais que define a passagem de uma mensagem SOAP para um outro protocolo. A estrutura de ligação contém regras que definem a forma como *binding* é feito, além das relações existentes entre *bindings* e módulos SOAP que implementam esses *bindings*.

- Construção de mensagem - Uma mensagem SOAP é especificada como um componente de dados de um XML. A construção da mensagem define como deverá ser enviada uma mensagem SOAP através de um documento XML.

As implementações existentes de SOAP fazem uso geralmente dos protocolos HTTP e SMTP. Em termos de avaliação de protocolos, algumas propostas relacionadas a este trabalho são descritas em [Ingo Paschke 2003] e [Sandra Cüsters 2004]. No primeiro texto é feita uma descrição mais detalhada do funcionamento do protocolo BEEP, além de uma comparação mais teórica entre BEEP e HTTP. Já no segundo

trabalho é feita uma avaliação mais prática do protocolo a partir do desenvolvimento de algumas implementações.

4. Protocolos de Aplicação

SOAP possibilita a adoção de diferentes protocolos de aplicação, tais como: HTTP, SMTP, FTP e BEEP, que possibilitam a troca de mensagens entre as aplicações. A seguir são apresentados os protocolos testados.

4.1. HTTP

HTTP (Hypertext Transfer Protocol) é um protocolo em nível de aplicação usado para a distribuição de informações de hipermídia [R. Fielding et al 1999]. Ele é um protocolo amplamente utilizado na Internet, principalmente sobre o protocolo TCP, para a transferência de arquivos devido a sua objetividade e rapidez, o que justifica sua utilização em ambientes de informação distribuídos e cooperativos.

O protocolo HTTP é baseado no paradigma requisição e resposta. Um cliente estabelece uma conexão com um servidor e envia uma solicitação ASCII a este, o qual analisa o pedido e responde, sendo esta resposta semelhante ao MIME [Tanenbaum 2003]. A conexão deve ser estabelecida antes de cada pedido de cliente e encerrada após a obtenção da resposta. Do ponto de vista de webservices este protocolo não se apresenta o mais indicado, uma vez que o modelo foi projetado para a simples transferência de arquivos, sendo atualmente remodelado para RPCs (Remote Procedure Call).

4.2. BEEP

BEEP é um mecanismo que permite a troca simultânea e independente de mensagens entre dois nós [M. Rose 2001]. BEEP suporta de forma dinâmica protocolos de aplicação para modelos de comunicação tais como ponto-a-ponto, cliente-servidor ou servidor-servidor. Ele suporta também múltiplos canais sobre uma ou mais sessões autenticadas, além de diferentes MIMEs, incluindo XML [Freed, N. and N. Borenstein 1996].

BEEP adiciona uma camada de abstração sobre os protocolos de transporte como TCP para facilitar o desenvolvimento e uso de protocolos de aplicação (chamados de *profiles*). Esta camada adicional provê todos os mecanismos básicos necessários para a troca de dados de forma simultânea e independente sem o contexto de uma identidade única do usuário, como dados codificados, negociação da capacidade da rede, segurança integrada, entre outros. Além disso, a camada suporta mensagens textuais e binárias.

4.3. SMTP

O objetivo de SMTP é a transferência de e-mails de forma segura e eficiente sob TCP [J. Klensin 2001]. O protocolo funciona de forma independente de um sub-sistema particular de transmissão e requer somente um canal que disponibilize um fluxo de dados ordenado. Uma importante característica de SMTP é a sua capacidade de retransmitir mensagens através de diferentes ambientes utilizados pelo serviço de transporte, o qual provê um ambiente de comunicação interprocessos. Assim, é possível ocultar uma rede ou um subconjunto da mesma.

5. Implementação

Os testes foram realizados utilizando aplicações desenvolvidas para esta finalidade: Um cliente e um servidor BEEP desenvolvidos em Java utilizando o pacote Bleepite, uma implementação do protocolo feita em Java desenvolvido pela IBM; Um cliente e um servidor HTTP desenvolvidos em Java utilizando o pacote Apache-SOAP; Um cliente SMTP desenvolvido em Java utilizando o pacote Apache-SOAP. Também foram utilizados um servidor Tomcat e a bridge SMTP2HTTP que acompanha o pacote Apache-SOAP para transformar as requisições HTTP em SMTP e vice versa. Desta forma o servidor SOAP HTTP foi o mesmo utilizado para responder as requisições do cliente SOAP-SMTP.

Os pacotes SOAP utilizados nas aplicações HTTP e SMTP foram gerados pela própria aplicação através de chamadas aos métodos do pacote Apache-SOAP. O pacote BEEP não fornece suporte a SOAP, por isso foi utilizado o fluxo de bytes XML capturado das aplicações HTTP e SMTP. Esse fluxo foi repetido pela aplicação SOAP.

A aplicação consistia em enviar uma string de tamanho pré-definido para o servidor. Uma vez recebido, o servidor devolvia a string em um pacote SOAP de resposta. O tamanho das strings utilizadas na medição foram de 1, 100 e 1200 bytes. Para cada uma destas strings foi originado um pacote SOAP a ser enviado com tamanhos de 458, 559 e 1620, respectivamente, enquanto que os pacotes SOAP de resposta possuíam 474, 575 e 1675 bytes.

O ambiente de testes consistia de duas máquinas conectadas por um switch. As medições do uso da rede foram feitas através do Ethereal, uma ferramenta que possibilita a captura de pacotes de uma determinada rede. Os testes também avaliaram a hipótese de utilizar uma única conexão para o envio e o recebimento de um único pacote SOAP e o envio e recebimento de dez pacotes SOAP simultâneos.

6. Resultados e Análise dos Resultados

Os resultados obtidos são apresentados nas tabelas abaixo. Na Tabela 1 são mostrados o número de pacotes utilizados para cada um dos casos analisados. Já na Tabela 2 é apresentado o número total de bytes enviados em cada uma das abordagens. Os dados obtidos a partir da coleta, demonstram que o número de pacotes necessários para a transmissão de mensagens com diferentes tamanhos é mais eficiente através de HTTP no caso de apenas uma mensagem ser transmitida.

Tabela 1. Número de Pacotes BEEP, HTTP e SMTP usando SOAP

Payload (bytes)	Loop	BEEP	HTTP	SMTP
1	1	23	13	57
100	1	26	13	73
1200	1	29	20	62
1	10	47	130	X
100	10	65	130	X
1200	10	96	160	X

No entanto, caso vários pacotes sejam enviados ao mesmo destinatário, BEEP possibilita manter aberta uma única conexão, pela qual são feitas todas as transmissões. Assim, para todas as 10 múltiplas mensagens o desempenho de BEEP foi consideravelmente melhor.

Tabela 2. Número de bytes enviados usando BEEP, HTTP e SMTP com SOAP

Payload (bytes)	Loop	BEEP	HTTP	SMTP
1	1	3297	2124	5451
100	1	3721	2262	6679
1200	1	6172	5151	8126
1	10	13151	20496	X
100	10	16819	22476	X
1200	10	41408	46176	X

Em relação aos resultados obtidos com SMTP, nota-se que o número de pacotes e da quantidade de bytes utilizados para a transmissão é consideravelmente maior. Isso ocorre por causa das sucessivas verificações periódicas de acesso ao servidor pop3 feitos pelo cliente SOAP, enquanto ele esperava uma resposta de uma requisição enviada por SMTP. Normalmente o servidor demorava a responder e várias verificações vazias eram efetuadas, consumindo banda, mas não trazendo dados na prática. Os resultados para o envio de múltiplos pacotes através de SMTP não foram implementados. O número total de bytes transmitido por SMTP para apenas um byte de dados é 2 vezes maior do que o de HTTP.

Um último fator a ser considerado é de que os protocolos realizam a quebra dos dados para envio em vários pacotes, a partir de tamanhos diferentes. Isto pode ser visto para os casos em que HTTP envia mensagens com 1 e 100 bytes com o mesmo número de pacotes, enquanto que o mesmo não ocorre com as demais implementações.

A Tabela 3, apresenta uma comparação entre a carga útil do arquivo XML em relação ao tamanho total do arquivo, tanto de envio como o de recebimento. Os resultados mostram que o tamanho das informações do arquivo XML se mantém mais ou menos constante, independente do tamanho do vetor de bytes enviado. A medida que aumenta o tamanho da mensagem útil, diminui a relação entre o tamanho total do arquivo e dos campos específicos de XML.

Tabela 3. Tamanho dos pacotes XML de envio e recebimento para cada um dos casos

Dados	XML Envio	XML Recebimento
1	458	474
100	559	575
1200	1620	1675

7. Conclusão

O uso de diferentes tecnologias e protocolos para a implementação de webservices possibilita uma grande flexibilidade na estrutura da arquitetura. Entretanto, a escolha dessas tecnologias e dos protocolos a serem utilizados em uma determinada implementação devem ser baseadas na análise das opções disponíveis, tendo como foco os requisitos do sistema que a implementação deseja atender. Para tanto, é preciso entender quais as vantagens oferecidas por cada um dos protocolos.

Este trabalho apresentou uma comparação entre os mecanismos de transmissão BEEP, HTTP e SMTP usados pelo protocolo SOAP, que é capaz de utilizar diferentes substratos para o envio de arquivos XML. Os resultados práticos mostraram que o protocolo BEEP é mais eficiente nos casos em que a transmissão de múltiplas mensagens de um mesmo cliente para um servidor é o principal requisito do sistema, pois este protocolo apresenta um grande *overhead* inicial para o estabelecimento da conexão. Após essa etapa, os pacotes possuem um tamanho bem menor que os pacotes SOAP HTTP e SMTP. Nos casos em que o requisito do sistema é o envio de mensagens únicas entre cliente e servidor o HTTP apresentou um desempenho significativamente melhor que os outros dois protocolos.

O uso de diferentes tecnologias e protocolos para a implementação de webservices possibilita uma grande flexibilidade na estrutura da arquitetura. No entanto, é imprescindível a escolha de recursos que sejam mais eficientes para uma determinada configuração.

References

- A. Tanenbaum (2004). “Redes de Computadores”. Campus. Rio de Janeiro.
- Cerami, E. (2002). “Web Services Essentials: Distributed Applications with XML-RPC, SOAP, UDDI & WSDL”. O’Reilly & Associates, 1005 Gravenstein Highway North, Sebastopol, CA 95472.
- C.Sampaio. SOA e Webservices em Java (2006), Editora Brasport.
- Elizabeth Castro (2001). “XML para a World Wide Web”. Campus.
- Freed, N. and N. Borenstein (1996). “Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies”. <http://www.ietf.org/rfc/rfc2045.txt>.
- Ingo Paschke (2003). “Implementierung und Evaluation eines SOAP/BEEP-Mappings”.
- J. Klensin (2001). “RFC 2821 - Simple Mail Transfer Protocol.”
- Litoiu, M. (2002). “Migrating to Web services - latency and scalability”. Fourth International Workshop on .Page(s): 13 – 20.
- M. Rose (2001). “RFC 3080 - The Blocks Extensible Exchange Protocol Core.”
- Martin Gudgin et al (2003). “SOAP Version 1.2 Part 1: Messaging Framework.” <http://www.w3.org/TR/soap12-part1/>.
- R. Fielding et al (1999). “RFC 2616 - Hypertext Transfer Protocol – HTTP/1.1.”
- Sandra Cüsters (2004). “Blocks Extensible Exchange Protocol.”

Werner, C.; Buschmann, C.; Jacker, T.; Fischer, S. (2005). "Enhanced transport bindings for efficient SOAP messaging" 2005 IEEE International Conference on Volume , Issue, 11-15 July 2005 Page(s): 193 - 200 vol.1.