

FACULDADE NATALENSE PARA O DESENVOLVIMENTO DO RIO GRANDE DO NORTE

Atualização de *Web Services* em Tempo Real

Vinícius Brandão Mendes

Orientador: Prof. Ricardo Wendel

Relatório técnico apresentado ao Curso Especialização em Desenvolvimento de Sistemas Corporativos da FARN como trabalho de conclusão de curso.

Natal, RN, abril de 2011

Atualização de *Web Services* em Tempo Real

Vinícius Brandão Mendes

Relatório técnico aprovado em 25 de abril de 2011 pela banca examinadora composta pelos seguintes membros:

Prof. Ricardo Wendel (orientador) DCA/UFRN

Prof. Dr. Pablo Javier Alsina DCA/UFRN

M. Sc. André Macedo Santana DCA/UFRN

Agradecimentos

Ao meu orientador, professor Ricardo Wendel, sou grato pela orientação.

Aos meus pais, Joalmi Mendes de Oliveira e Maria Gleide Brandão Mendes, pelo apoio total durante a execução do trabalho e durante toda a minha vida, com educação, instrução e companheirismo.

A minha noiva Rosana Curvelo de Souza pelo constante incentivo.

Ao colega Fábio Miranda Costa pelo apoio durante o curso.

Aos meus irmãos, familiares e amigos pela paciência e apoio dispensados durante a execução do trabalho.

Resumo

Neste trabalho será relatado o desenvolvimento de um cliente de *Web Service* baseado em SOAP que deve realizar várias requisições para sincronizar uma grande massa de dados em um tempo relativamente curto. Foram abordadas três técnicas: requisições sob demanda do usuário; atualizador com requisições síncronas e sequenciais em um único processo; e atualizador com requisições síncronas em vários processos.

Palavras-chave: *Web services*, SOAP, computação distribuída, programação concorrente.

Abstract

On the following work, will be reported the development of a Web Service client based on SOAP that should make many requests to synchronize a huge data in a short time. Three techniques were considered: request on user demand; update with synchronous and sequential requests in a single process; and updater with synchronous requests with multiprocessing.

Keywords: Web Services, SOAP, distributed computing, multiprocessing.

Sumário

| | |
|--|------------|
| Sumário | i |
| Lista de Figuras | iii |
| 1 Introdução | 1 |
| 1.1 Descrição geral | 1 |
| 1.2 Objetivo | 1 |
| 1.3 Motivação | 1 |
| 1.4 Metodologia | 2 |
| 1.4.1 Módulo para realizar comunicação utilizando SOAP | 2 |
| 1.4.2 Estratégia de atualização de dados sob demanda | 2 |
| 1.4.3 Atualizador de dados independente de este ser requisitado ou não | 2 |
| 2 Web Services | 3 |
| 2.1 Aplicação | 3 |
| 2.2 Descoberta | 3 |
| 2.3 Descrição | 3 |
| 2.4 Empacotamento | 4 |
| 2.4.1 SOAP | 4 |
| 3 Multiprocessamento | 5 |
| 4 Implementação | 7 |
| 4.1 Requisições sob demanda | 7 |
| 4.2 Atualizar todos os dados de tempos em tempos | 7 |
| 4.2.1 Requisições sequenciais | 7 |
| 4.2.2 Requisições paralelas com múltiplos processos | 7 |
| 5 Resultados | 9 |
| 5.1 Requisições sob demanda | 9 |
| 5.2 Atualizar todos os dados de tempos em tempos | 9 |
| 5.2.1 Requisições sequenciais | 9 |
| 5.2.2 Requisições paralelas com múltiplos processos | 9 |
| 6 Conclusão | 11 |
| Referências bibliográficas | 12 |

Lista de Figuras

Capítulo 1

Introdução

Este trabalho trata de um cliente de *Web Service* SOAP que deve obter o mais rapidamente possível uma grande quantidade de dados do servidor e persistí-los em seu banco de dados. Existem inúmeras formas de resolver esse problema. Neste trabalho serão abordadas algumas técnicas possíveis.

1.1 Descrição geral

O trabalho é baseado em um *Web Service* SOAP, que é um protocolo para troca de informações estruturadas em uma plataforma descentralizada e distribuída.

Um portal de internet precisa oferecer a seus usuários em tempo real dados que são obtidos através de um servidor que os fornece através de um *web service*. Para não ter atraso na informação oferecida, ele precisa requisitar ao fornecedor de dados o mais frequentemente possível ao mesmo tempo que tem que manter um bom tempo de resposta para seus usuários.

1.2 Objetivo

O objetivo do trabalho é resolver os problemas relacionados à otimização do tempo de resposta e à frequência de atualização do dado. Por se tratar de um problema de otimização de tempo de resposta é necessário pensar na latência e no *overhead* gerado por uma conexão HTTP e analisar a melhor arquitetura de sistema possível.

1.3 Motivação

Com o avanço da informática e de conceitos como a computação em nuvem, cada vez mais os servidores estão em constante comunicação ao redor do mundo. Seja para autenticar usuários, seja para persistir dados ou seja para solicitar dados. Um portal de notícias, por exemplo, exibe, além das notícias, dados meteorológicos e econômicos. Esses dados normalmente não são produzidos pelo próprio portal e então são necessárias parcerias com fornecedores de conteúdo que disponham de tal informação. Estes fornecedores, em geral, disponibilizam os dados através de um *Web Service*.

Dado que no mundo jornalístico a agilidade na entrega de informações é um fator primordial para o sucesso, surge a necessidade de otimizar o tempo de consumo de tais fornecedores a fim de entregar aos usuários a informação mais recente possível.

1.4 Metodologia

O projeto foi dividido em três partes:

- módulo para realizar comunicação utilizando SOAP;
- estratégia de atualização de dados sob demanda;
- atualizador de dados independente de este ser requisitado ou não.

1.4.1 Módulo para realizar comunicação utilizando SOAP

Foi necessário desenvolver um módulo capaz de fazer a comunicação utilizando o protocolo SOAP, servi-los aos clientes e persisti-los em seu banco de dados.

1.4.2 Estratégia de atualização de dados sob demanda

Uma das possíveis opções de atualização de dados é fazê-lo sob demanda. A medida em que um usuário solicita o dado, o portal o requisita ao fornecedor e gera a resposta para o usuário. Esta foi a primeira estratégia analisada.

1.4.3 Atualizador de dados independente de este ser requisitado ou não

Foi analisada uma estratégia de atualizar os dados independentemente de eles terem sido requisitados pelos usuários ou não. Esta estratégia se mostra interessante por reduzir o tempo de resposta para o usuário.

Capítulo 2

Web Services

Web service é uma interface acessível através da rede para uma funcionalidade de uma aplicação construída usando tecnologias definidas como padrão na Internet [Doug Tidwell & Kulchenko 2001]. É apenas mais uma camada de troca de mensagens entre uma aplicação e outra. A principal vantagem no seu uso é prover comunicação entre diferentes aplicações independente da plataforma ou linguagem de programação utilizada por elas, o que garante interoperabilidade entre sistemas desde que ambos utilizem o mesmo protocolo de comunicação entre si.

Através do uso de *web services* é possível fazer com que uma aplicação faça chamadas de métodos remotos de outra aplicação. É simples como gerar uma requisição HTTP que encapsule qual método será chamado e quais os parâmetros e esperar uma resposta com o resultado da chamada do método. Para isso se faz necessário quebrar a camada de aplicação da pilha de camadas do modelo de redes TCP/IP em quatro camadas: aplicação, descoberta, descrição e empacotamento.

2.1 Aplicação

A camada de aplicação é o código que precisará se comunicar com a outra aplicação através do *web service*, que pode ser escrito em qualquer linguagem.

2.2 Descoberta

Esta camada é responsável por disponibilizar metadados sobre *web services* de modo a facilitar a busca pela aplicação necessária para cada projeto.

2.3 Descrição

A descrição define como o *web service* deve ser utilizado. Quais os métodos que ele expõe, quais os parâmetros necessários em cada um deles e quais as possíveis respostas. O *Web Service Description Language* (WSDL) é um padrão muito utilizado para descrever um *web service* através do uso de XML.

2.4 Empacotamento

Trata do empacotamento das mensagens de forma que ambas as partes sejam capazes de entendê-las. Existem vários formatos de empacotamento, em sua maioria baseados em XML (*Extensible Markup Language*), o qual possibilita a representação do significado do dado e por existirem *parsers* XML para praticamente todas as linguagens e plataformas do mercado. Neste relatório iremos abordar o protocolo de empacotamento SOAP que foi utilizado na implementação do projeto.

2.4.1 SOAP

SOAP (*Simple Object Access Protocol*) é um protocolo cujo objetivo é trocar informações estruturadas em um ambiente distribuído e descentralizado [Gudgin et al. 2007]. É um padrão reconhecido e desenvolvido pelo World Wide Web Consortium (W3C) para desempenhar o papel da camada de empacotamento de um *web service*. Sua especificação define um envelope baseado em XML para conter a informação a ser transmitida e uma série de regras para traduzir dados específicos de uma de plataforma e aplicação para representações XML.

Capítulo 3

Multiprocessamento

Falar sobre programação concorrente com múltiplos processos.

Capítulo 4

Implementação

4.1 Requisições sob demanda

4.2 Atualizar todos os dados de tempos em tempos

4.2.1 Requisições sequenciais

4.2.2 Requisições paralelas com múltiplos processos

Capítulo 5

Resultados

5.1 Requisições sob demanda

5.2 Atualizar todos os dados de tempos em tempos

5.2.1 Requisições sequenciais

5.2.2 Requisições paralelas com múltiplos processos

Capítulo 6

Conclusão

Referências Bibliográficas

Doug Tidwell, James Snell & Pavel Kulchenko (2001), *Programming Web Services with SOAP*, 1 edi, O'Reilly.

Gudgin, Martin, Marc Hadley, Noah Mendelsohn, Yves Lafon, Jean-Jacques Moreau, Anish Karmarkar & Henrik Frystyk Nielsen (2007), SOAP version 1.2 part 1: Messaging framework (second edition), W3C recommendation, W3C. <http://www.w3.org/TR/2007/REC-soap12-part1-20070427/>.