

Análise de desempenho de arquitetura SOAP e REST para comunicação entre sistemas

Ricardo Voigt¹, Osmar de Oliveira Braz Junior¹

¹Centro de Educação Superior do Alto Vale do Itajaí (CEAVI) – Universidade do Estado de Santa Catarina (UDESC) – Ibirama, SC – Brasil

voigt@gmail.com, osmar.braz@udesc.br

Resumo. A medição de análise de desempenho é realizada com auxílio da ferramenta Apache jMeter e são verificadas métricas, como menor desvio padrão por requisição, tempo médio, mínimo e máximo de resposta por requisição, volume de transferência de dados por segundo para processar as requisições e tamanho do pacote por solicitação. A análise é realizada em servidor local e em servidor remoto na nuvem, para uma verificação mais realista. Como resultado é observado que o Web service na arquitetura REST possui um melhor desempenho em relação ao SOAP, nos testes de desempenho com 1, 10 e 100 usuários simultâneos realizando a mesma operação.

Palavras-chave: SOAP, REST, Web service, Desempenho.

Abstract. Performance analysis measurement is performed with the help of Apache jMeter tool and are checked metrics such as lower standard deviation per request, average time, minimum and maximum response request, data transfer volume per second to process requests and size package by request. The analysis is performed on the local server and remote server in the cloud, for a more realistic verification. As a result it is observed that the Web service in REST architecture has a better performance compared to SOAP, in performance tests with 1, 10 and 100 concurrent users performing the same operation.

Keywords: SOAP. REST. Web service. Performance.

1. Introdução

A indústria de desenvolvimento de software tem se tornado uma das mais importantes indústrias do nosso tempo. Empregando milhares de trabalhadores em todos os países, essa indústria cria muitos produtos essenciais para manter o funcionamento de muitas indústrias, empresas e também nosso estilo de vida (LEFFINGWELL, 2004).

De acordo com Albinader (2006), inúmeras ferramentas e tecnologias surgiram nos últimos anos, destinadas ao desenvolvimento de aplicações para a Internet. Entretanto, nenhuma é completa e todas possuem suas limitações e particularidades, como por exemplo, a troca de informação de forma simples e automática.

A necessidade da integração entre aplicações, a utilização unificada de processos encontrados em diferentes sistemas e escritos em diferentes linguagens são alguns exemplos de carências encontradas em empresas nos dias de hoje. A fim de sanar estas questões, a tecnologia dos Web services foi criada, permitindo assim, disponibilizar

formas de integrar sistemas distintos, modularizar serviços, capacitar a integração e consumo de informações (MENÉNDEZ, 2002).

O SOAP (Simple Object Access Protocol) é um protocolo de comunicação baseado em XML que possibilita a comunicação de mensagens entre aplicações por meio de HTTP (Hypertext Transfer Protocol), e que é normalmente utilizado em Web services (W3SCHOOLS, 2014).

Nos últimos anos, uma nova categoria de serviços web, denominados serviços web RESTful vem ganhando espaço e sendo aplicada em empresas de desenvolvimento de software que necessitam realizar as comunicações entre sistemas heterogêneos. Os serviços web RESTful são serviços que se apoiam no estilo arquitetural REST (Representational State Transfer) e que usam os recursos disponíveis no protocolo HTTP (FIELDING, 2000).

Contrariamente à crença comum, a avaliação de desempenho é uma arte. Como uma obra de arte, avaliação bem sucedida não pode ser produzida mecanicamente. Cada avaliação requer um conhecimento íntimo do sistema que está sendo modelado e uma cuidadosa seleção da metodologia, carga de trabalho e ferramentas utilizadas. (JAIN, 1991).

Segundo Menascé e Almeida (2003), a disponibilidade, qualidade e desempenho dos serviços disponibilizados na web, precisam ser constantemente avaliados para que a necessidade do usuário final seja garantida. Diante disso, surge a necessidade de avaliar o desempenho de arquitetura SOAP e REST de Web service a fim de identificar a arquitetura que tenha um melhor desempenho.

O objetivo deste artigo é realizar um estudo de desempenho entre estes dois protocolos, com o auxílio da ferramenta Apache jMeter. Foram implementados e hospedados em servidor local e remoto, Web service em SOAP e REST na linguagem C# com dados públicos, e realizado testes com um, dez e cem usuários simultâneos a fim de identificar o desempenho de cada arquitetura através de métricas de menor desvio padrão por requisição, tempo médio, mínimo e máximo de resposta por requisição, volume de transferência de dados por segundo para processar as requisições e tamanho do pacote por solicitação. Com os resultados obtidos neste artigo é possível se basear para escolhas em futuras implementações de integrações entre sistemas.

A análise de desempenho contém uma abordagem quantitativa, onde os resultados são quantificados, se centrando na objetividade. Os resultados são compreendidos com base na análise de dados brutos obtidos com o auxílio de ferramentas (FONSECA, 2002). Observando as métricas de desempenho utilizadas a partir de uma ferramenta que executa testes funcionais e mede o desempenho de aplicações, objetiva-se gerar conhecimentos para aplicação prática, que é a solução de um problema específico, dirigidos à escolha da melhor arquitetura de Web service, para solução de problemas de desempenho e integração entre sistemas. Segundo Gil (2007), a pesquisa exploratória tem como objetivo proporcionar maior familiaridade com o problema, e torná-lo mais explícito e geralmente envolve a análise de exemplos que estimulem a compreensão do problema. Assim sendo, o enfoque da análise de desempenho é exploratória para a descoberta da arquitetura que obtém o melhor desempenho no cenário proposto.

O artigo está organizado da seguinte forma, além desta introdução, são apresentados os trabalhos correlatos e a ferramenta de teste utilizada, a seguir são apresentadas as arquiteturas SOAP e REST e realizado uma comparação entre elas na quarta seção, em seguida são apresentadas a metodologia da pesquisa e os resultados obtidos neste artigo, e por fim, são feitas as considerações finais sobre o trabalho.

2. Trabalhos Correlatos

Para a criação deste trabalho foi realizado um estudo de três artigos existentes no contexto de arquiteturas SOAP e REST para webservices.

A abordagem apresentada em Shi (2006), o principal foco são Web services semânticos. É explorado o que SOAP e RESTful têm em comum e combina as melhores características provenientes de Web services SOAP e REST. Apresenta um estudo que relaciona os Web services e a Web semântica, ambas sobre HTTP e sem a utilização de frameworks, tais como SOAP ou WSDL. De acordo com o autor, este método viabiliza aos usuários utilizar e compartilhar serviços por meio da computação distribuída.

Ferreira e Mota (2014) fazem uma análise comparativa entre as duas arquiteturas, e destaca situações onde utilizar uma arquitetura SOAP, como por exemplo, em aplicações que requerem um contrato rígido no formato de intercambio de dados. Os autores também destacam que REST pode ser útil quando não se tem rigidez na troca de informações, podendo ser acessado por qualquer navegador e que possa receber XML ou json. Entretanto no trabalho dos autores não é realizada uma análise de desempenho utilizando uma ferramenta para realizar as requisições nos serviços.

No trabalho de Bigolin (2012) é realizado uma análise e implementação de Web services SOAP e REST. O autor faz uma comparação entre as arquiteturas, implementa os Web services utilizando a linguagem PHP. Bigolin (2012) apenas realiza a medição do tempo de execução em segundos e não especifica a quantidade de usuários utilizados na análise.

As semelhanças entre o presente estudo e os artigos analisados referem-se ao fato de terem realizado uma análise comparativa entre as duas arquiteturas SOAP e REST e destacam fatores de usar uma arquitetura em alguns cenários ou não, o que difere deste trabalho que analisa as duas arquiteturas para evidenciar por meio de métricas qual possui o melhor desempenho, por meio de conexões simultâneas nas duas arquiteturas.

3. Ferramenta de teste Apache jMeter

A ferramenta de teste utilizada nos testes de desempenho dos Web services em SOAP e REST foi o Apache jMeter versão 3.0 r1743807. O Apache jMeter é uma ferramenta construída em Java para ambiente desktop, com a finalidade de realizar testes de desempenho e estresse em aplicações cliente/servidor, como por exemplo, web sites e sistemas web (JMETER, 2016).

Por ser uma ferramenta inteiramente escrita em Java, o jMeter é compatível com qualquer ambiente capaz de suportar a máquina virtual Java versão 1.4 ou superior. A cada solicitação realizada com a ferramenta, são totalizadas as informações de resposta e apresentado no relatório de sumário, o numero de amostras, a média, o mínimo, o

máximo, o desvio padrão, o percentual de erro, a vazão aproximada (requisição / segundo) e os kilobytes por segundo de transferência (JMETER, 2016).

4. SOAP x REST

Web services desenvolvidos no protocolo SOAP, tem sido muito utilizado pela indústria de software, com integração de sistemas, em processos distribuídos, e principalmente em projetos que requerem um nível maior de segurança e confiabilidade. A Tabela 1 mostra as funcionalidades que a arquitetura apresenta, SOAP, possui a descrição do serviço em WSDL enquanto que REST não possui um padrão de descrição do serviço. REST utiliza exclusivamente do protocolo HTTP, e SOAP possui vários tipos de protocolos suportados. A segurança por sua vez, pode ser aplicada em SOAP por meio de HTTPS ou de WS-Security, já em REST a única forma de realizada uma requisição com segurança é através do protocolo HTTPS.

Tabela 1 – Comparativo SOAP x REST

| | SOAP | REST |
|-----------------------------|--------------------------|--|
| Descrição do Serviço | WSDL | Nenhum padronizado |
| Comunicação | Síncrona / Assíncrona | Síncrona |
| Web service | É o próprio serviço | Tudo é um recurso |
| Descoberta | WSDL | WADL |
| Estado | Com estado | Sem estado |
| Publicação | ESB ou UDDI | APP |
| Pesquisa | Baseado em serviços | Baseado em recursos |
| Formato de Mensagem | SOAP/XML | Não padronizada |
| Linguagens | C++, C#, Java, .Net, etc | HTML, Json, Javascript, Java, .Net, Ruby |
| Protocolos | Vários | HTTP |
| Segurança | HTTPS / WS-Security | HTTPS |

Fonte: (Adaptado de BIGOLIN, 2012)

Segundo Padilha (2013), uma mensagem SOAP tem um formato específico e bem definido, denominado envelope SOAP. Os dados enviados para um Web service SOAP precisam estar representados no envelope, que além dos dados, contém outras informações relevantes para se obter acesso a um recurso, como seu nome, seus parâmetros, seu endereço específico dentre outras informações.

Já uma mensagem REST não possui um formato de dados específico pré-definido, onde qualquer tipo de hipertexto pode ser usado, como por exemplo, XML, JSON, texto plano e inúmeros outros tipos. Em um mesmo Web service pode existir recursos que utilizam mensagens de tipos distintos, aumentando a flexibilidade, conforme cada tipo de aplicação. Além disso, ao utilizar o formato de JSON, por exemplo, ao invés de XML, uma mesma mensagem pode ficar muito mais compacta, diminuindo o fluxo de dados na rede, tornando mais simples de interpretar a mensagem por quem solicita determinado recurso.

5. Metodologia

Para o desenvolvimento dos Web services foi utilizada a linguagem de programação C# e framework .Net framework 4.0 em conjunto com o Visual Studio 2013, que permitiu a construção dos Web services em SOAP e REST e processos de compilação, build e publicação dos serviços. Foi utilizada uma base de dados (MySQL), com dados públicos de domínios gov.br, extraídos a partir do Portal Brasileiro de Dados Abertos (dados.gov.br) composta por 1493 domínios e 609 entidades relacionados entre si, para

a avaliação da busca e retorno da mensagem em cada Web service. Os dois Web services foram hospedados em um servidor local IIS 10 instalados no sistema operacional Windows 10.

Para realizarmos uma comparação tanto local quanto em ambiente externo, os dois Web services foram hospedados em um servidor cloud, na startup de servidores de hospedagem umbler (<https://www.umbler.com/br>), em um cloud server.

A configuração da máquina utilizada para o teste de desempenho local, e do servidor cloud nos testes externos é descrita na Tabela 2. No servidor local, além de ser o executor dos testes, também hospedou os Web services para os testes locais. O servidor cloud é um servidor compartilhado, em uma máquina virtual, hospedado na nuvem, onde não é possível assumir que possui somente o web service executando no momento da requisição. O processador dos dois servidores, local e cloud possuem uma configuração semelhante. A versão do servidor web (IIS) é diferente devido à versão do Windows, onde o Windows 10 tem o IIS 10 e o Windows Server 2012 possui a versão 8.5, ambas sendo as últimas versões do serviço, e que não possuem diferenças de desempenho para a análise realizada.

Tabela 2 – Configuração da máquina local e remota utilizadas para o teste de desempenho

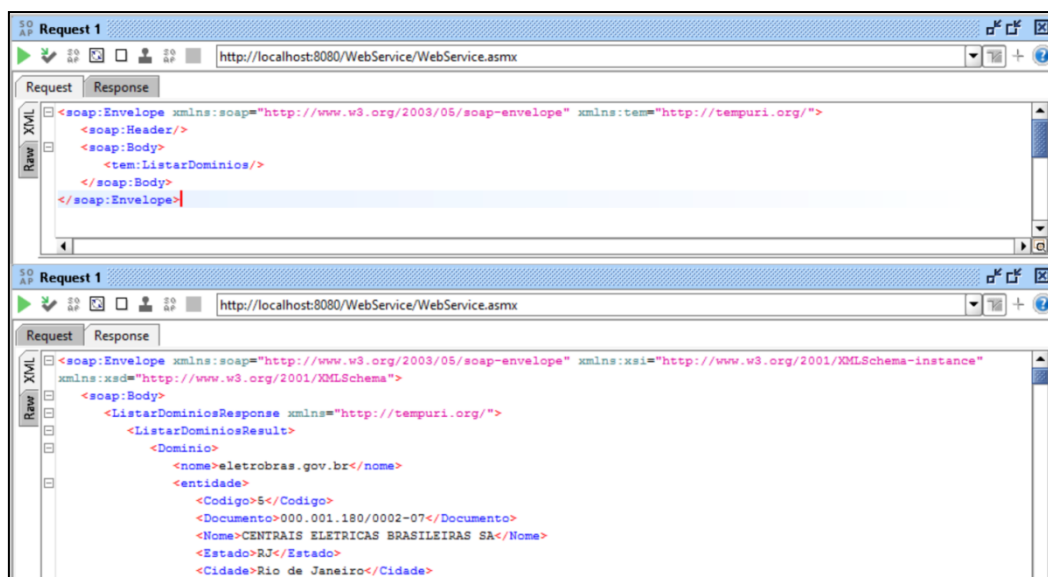
| | Servidor Local | Servidor Cloud |
|----------------------------|--|------------------------|
| Modelo | HP Pavilion dm4x3116tx | Cloud (umbler.com) |
| Processador | Intel(R) Core(TM) i7-3612QM CPU @2.10GHz | 2.4GHz |
| Memória RAM | 16GB – DDR3 – 1600 MHz | 1GB |
| Sistema Operacional | Windows 10 64bits | Windows Server 2012 R2 |
| Servidor Web | IIS 10 | IIS 8.5 |

Os testes de desempenho foram realizados com a ferramenta Apache jMeter versão 3.0, com testes de 1, 10 e 100 usuários requisitando informações de domínio e entidade através do método ListarDominios() de cada Web service. O teste foi realizado de forma isolada em cada Web service SOAP e REST, local e remoto, a fim de não possuir interferências de desempenho por executá-los simultaneamente, e também por não solicitar a banda da internet. Os dados foram consumidos da tabela domínio que possui uma chave estrangeira com a tabela entidade. Cada requisição realizada no método ListarDominios() retorna uma lista de domínios com 1493 registros.

Os Web services SOAP e REST foram construídos para responder as requisições recebidas retornando uma listagem de dados da tabela domínio e entidade, a fim de gerar um fator de medição de desempenho da execução dos testes.

Com o auxílio da ferramenta SoapUI, foi realizada uma requisição ao Web service em SOAP, que é demonstrada na Figura 1, como pode ser observado o envelope SOAP, na aba request, que representam os dados que são enviados ao Web service para realizar a solicitação. O envelope é composto pelo método ListarDominios, sem nenhum parâmetro. Na aba response, é demonstrada a resposta, em formato XML, obtida ao método requisitado. A resposta contém todos os registros da base de dados. A listagem é composta pelo objeto Domínio, que contém o atributo nome. Está relacionado o objeto entidade ao domínio. O objeto Entidade contém os atributos: Código, Documento, Nome, Estado, Cidade e Cep. O objeto domínio representa um domínio do governo, e o objeto entidade, por sua vez, representa os dados da entidade responsável pelo domínio.

Figura 1 – Solicitação ao método ListarDominios() e retorno dos dados no formato XML ao Web service implementado em SOAP

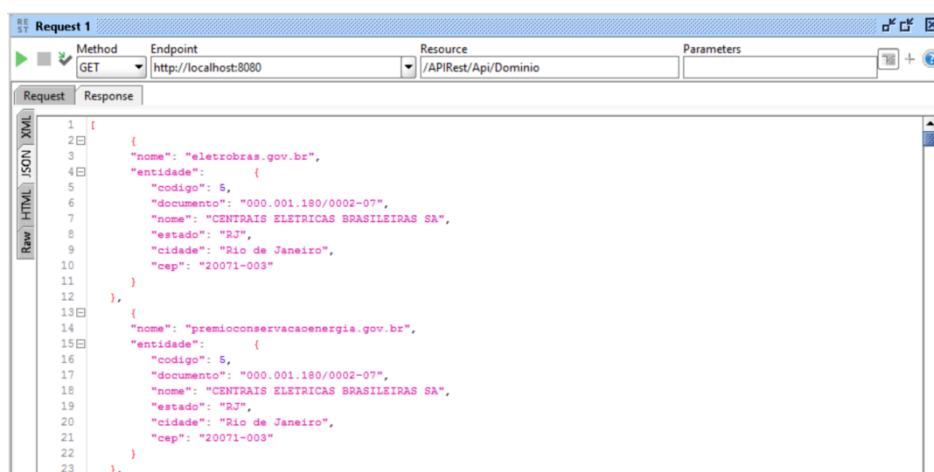


Na Figura 2, é obtido o retorno de uma chamada HTTP realizada ao Web service REST com o auxílio da ferramenta SOAPUI. O retorno do Web service está no formato JSON. O retorno é composto pelo objeto domínio, com o atributo nome e pelo objeto entidade, com os atributos: código, documento, nome, estado, cidade e cep.

O tamanho do pacote no Web service da arquitetura SOAP é de 648KB, enquanto que o Web service na arquitetura REST é de 444KB.

A partir dos Web services respondendo as requisições e retornando os registros em cada formato, de XML e JSON, foi possível realizar testes de desempenho com a ferramenta Apache jMeter.

Figura 2 – Solicitação ao método ListarDominios() e retorno dos dados no formato JSON ao Web service implementado em REST



6. Metodologia

A análise dos resultados é demonstrada de forma comparativa entre as requisições SOAP e REST e também é feita a avaliação remota da desempenho em relação à

requisição local. De acordo com a Tabela 3 e a Tabela 4, é possível analisar os valores obtidos através dos testes realizados com a ferramenta jMeter, nas requisições para o Web service REST e para o Web service SOAP.

Os testes foram realizados em duas etapas. A primeira etapa contou com a execução dos testes no servidor local. A Tabela 3 apresenta os dados relevantes dos testes. Na segunda etapa foram realizados os testes no servidor cloud e os dados foram apresentados na Tabela 4. A quantidade de amostras foi realizada de forma igual em ambos os Web services, com testes realizados com 1, 10 e 100 usuários simultâneos. Para a análise de cada métrica, adotamos os resultados obtidos com os testes realizados com 100 usuários simultâneos.

Tabela 3 – Avaliação de desempenho em diferentes amostras locais

| <i>Web Service</i> | Amostras | Média | Min. | Máx. | Desvio Padrão | % Erro | Vazão | KB/s | Média de Bytes |
|--------------------|----------|-------|------|------|---------------|--------|-------|----------|----------------|
| REST | 1 | 387 | 387 | 387 | 0 | 0 | 2,6 | 1022,14 | 405060 |
| REST | 10 | 20 | 20 | 22 | 0,6 | 0 | 10,9 | 4294,97 | 405060 |
| REST | 100 | 46 | 21 | 472 | 82,63 | 0 | 20,1 | 7939,91 | 405060 |
| SOAP | 1 | 436 | 436 | 436 | 0 | 0 | 2,3 | 827,23 | 369330 |
| SOAP | 10 | 131 | 17 | 449 | 152,60 | 0 | 10,9 | 3928,91 | 369330 |
| SOAP | 100 | 222 | 32 | 653 | 215,37 | 0 | 85,9 | 30985,72 | 369330 |

O Web service REST, teve o menor tempo médio de resposta no servidor local de 46 milissegundos, e no servidor remoto de 18356 milissegundos, enquanto que o Web service SOAP obteve um tempo maior, em 222 milissegundos no servidor local e 80245 milissegundos no servidor remoto. Isso representa uma variação percentual superior em 382,61% do Web service REST em relação ao Web service SOAP no servidor local, e 337,16% no servidor remoto respectivamente. O tempo mínimo também foi menor em relação aos testes realizados no Web service SOAP, com 21 contra 32 no servidor local e 1370 e 20505 respectivamente. Isso representa um variação percentual superior em 52,38% no tempo mínimo nos testes locais e nos testes com servidor remoto gerou 1396,72% no tempo mínimo.

O Web service REST se mostrou mais eficiente em relação ao tempo máximo de resposta, analisado pela coluna Max da Tabela 4, onde levou 472 milissegundos, enquanto o Web service SOAP durou 653 milissegundos para responder a requisição. O Web service REST gerou uma variação percentual superior no tempo máximo em 38,35%, nos testes locais e nos testes com servidor remoto gerou 119,38% superior no tempo máximo.

O desvio padrão no Web service REST teve um menor desvio de 82,63 e o Web service SOAP 215,37 uma variação muito maior em relação ao testes locais, comparando no caso de 100 usuários simultâneos. O Web service REST obteve uma variação percentual superior em 160,64% em relação ao SOAP no teste local e 154,51% no teste remoto. Em todos os testes realizados o percentual de erro retornou 0%, o que significa que todas as requisições foram atendidas tanto pelo Web service REST quanto pelo Web service SOAP.

A métrica de vazão, que é a taxa com que um sistema responde as requisições, obteve um valor de 20,1 para o Web service REST e 85,9 no Web service SOAP, no servidor local, e 2,2 no servidor remoto em ambos os Web services.

Na métrica de volume de transferência de dados por segundo, coluna KB/s da Tabela 3, os testes realizados no Web service REST se mostrou ser superior ao SOAP, retornando um maior volume de dados por segundo. Nesta métrica, o teste no Web service REST obteve uma variação superior em 290,25% em relação ao SOAP no servidor local. Já no servidor remoto, o Web service REST foi inferior ao SOAP, onde o SOAP foi 56,37% melhor em relação ao REST. Não foi possível identificar o motivo desta diferença no servidor remoto nesta métrica, o que pode ser ocasionado por algum processamento ocorrendo no servidor no momento da avaliação, visto que, por ser um servidor cloud em um serviço compartilhado, não é possível ter 100% de uso exclusivo para o Web service.

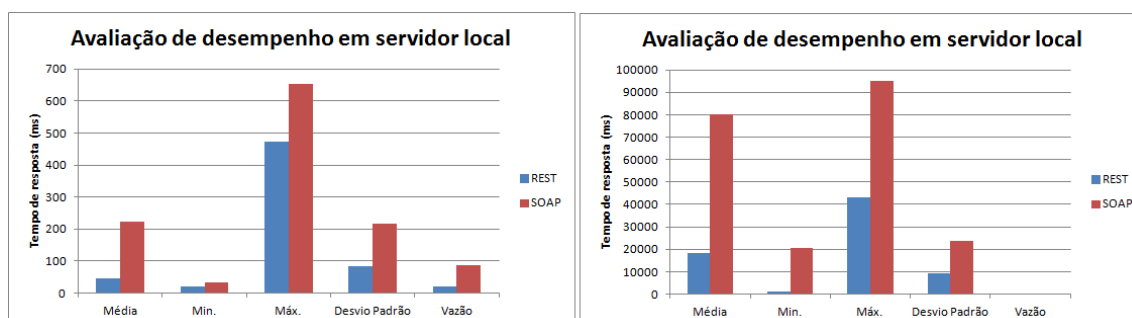
Tabela 4 – Avaliação de desempenho em diferentes amostras remoto

| Web Service Remoto | Amostras | Média | Min. | Máx. | Desvio Padrão | % Erro | Vazão | KB/s | Média de Bytes |
|--------------------|----------|-------|-------|-------|---------------|--------|-------|--------|----------------|
| REST | 1 | 474 | 474 | 474 | 0 | 0 | 2,1 | 834,53 | 405060 |
| REST | 10 | 1183 | 275 | 3634 | 963,62 | 0 | 2,4 | 956,86 | 405060 |
| REST | 100 | 18356 | 1370 | 43290 | 9287,22 | 0 | 2,2 | 866,70 | 405060 |
| SOAP | 1 | 1216 | 1216 | 1216 | 0 | 0 | 2,1 | 296,61 | 369330 |
| SOAP | 10 | 6308 | 1752 | 8365 | 1891,53 | 0 | 2,4 | 389,20 | 369330 |
| SOAP | 100 | 80245 | 20505 | 94970 | 23636,94 | 0 | 2,2 | 378,18 | 369330 |

Em relação ao tamanho do pacote de cada requisição única, podemos obter esta informação com o auxílio da ferramenta SoapUI. O Web service SOAP retornou um pacote com tamanho de 648KB, enquanto o REST retornou um pacote com tamanho de 444KB. O tamanho dos pacotes também são maiores em SOAP. Analisando com vários clientes requisitando a mesma informação, a quantidade de tráfego na rede é muito maior com SOAP, tornando o REST uma opção mais viável. O tamanho do pacote REST é de 444KB e em SOAP é de 648KB para os dados inseridos no banco de dados. Isso representa uma variação de 45,95% entre os dois modelos implementados.

Na Figura 3, é possível identificar de forma visual, que tanto nos testes locais, como remoto o desempenho ficou muito semelhante com suas devidas proporções. Por ser um ambiente externo não controlado, pode haver algumas diferenças, como por exemplo, na diferença da média entre SOAP e REST local e no servidor remoto. Isso se dá a conexão, tempo de resposta que influenciam neste indicador. O desvio padrão e a vazão foram os que mantiveram a mesma proporção. Comprovando que em ambos os cenários, o Web service REST teve um desempenho melhor em relação ao Web service SOAP.

Figura 3 – Comparativo de desempenho entre web services SOAP e REST



7. Considerações Finais

Como resultado dos testes, pode-se afirmar que definitivamente a implementação de Web service em REST provou ser mais rápida e eficiente nas métricas de menor desvio padrão por requisição, menor tempo médio, mínimo e máximo por resposta de requisição, volume de transferência de dados por segundo para processar requisições e menor tamanho de pacote por solicitação, gerando assim uma melhor performance. A implementação do Web service REST tornou-se melhor para este modelo de dados e produziu resultados melhores em relação ao Web service SOAP.

Os resultados no servidor cloud, tiveram por objetivo apresentar não somente um ambiente controlado local, mas sim um ambiente externo, como é utilizado em diversas empresas de tecnologias. Assim sendo, obteve-se um resultado equivalente ao executado no servidor local, para a comprovação de que o teste de desempenho proposto, dos Web services suportariam, 1, 10 e 100 conexões simultâneas sem gerar uma queda de serviços, derrubar o servidor local ou o servidor remoto por excesso de conexões. Retornando a resposta das requisições solicitadas através da URL de cada Web service.

O principal fator se deve que Web services REST são baseados em JSON, um formato muito leve, basicamente texto puro, e os Web services SOAP são baseados em XML, um formato mais pesado, que exige um maior esforço da parte da aplicação e, consequentemente, exige mais recursos como memória e processamento no servidor.

Desta forma, aplicações distribuídas com muitas requisições e fluxo intenso de troca de mensagens com muitas informações via Web services podem ter um ganho significativo de desempenho com o uso de Web services REST.

A possibilidade de retornar dados em diferentes representações torna os Web services mais dinâmicos e oferece diferentes opções ao cliente requisitante. Esta característica dos Web services em REST não são encontradas nativamente nos Web services em SOAP, os quais trabalham com representação XML por padrão. Nos serviços desenvolvidos para esta análise, o retorno dos Web services no estilo REST podem ser requisitados nas representações JSON, sendo que em Web services SOAP, representações diferentes da XML não são possíveis sem serem parte do conteúdo do pacote SOAP. Entretanto, aplicações específicas podem adotar o uso de Web services SOAP devido a outras vantagens, como por exemplo, as extensões WS-* (WS-Addressing, WS-Reliability, WS-Addressing, WS-Policy).

Em trabalhos futuros, podem ser considerados outros critérios de testes, como a influência da banda de internet e tempo de resposta entre servidores. Detalhando os tempos intermediários para que seja possível identificar as suas influências. Outro item que pode ser testado, é a adoção de testes com tipos de processadores distintos, voltado mais para o ambiente de infraestrutura e redes.

Referencias

- Albinader Neto, et al. Web Services em Java Rio de Janeiro, Brasport. 2006
- Bigolin, M. Análise e implementação de web services. Trabalho de Conclusão de Curso, Universidade Federal do Rio Grande do Sul, Instituto de Informática. 2012.

- Ferreira, C; Mota R. Comparando aplicação web service rest e soap. Trabalho de Conclusão de Curso, Universidade Paranaense Unipar, Paranavaí, 2014.
- Fielding, R. T. Architectural styles and the design of network-based software architectures. Tese de Doutorado, University OfCalifornia, Irvine, 2000.
- Fonseca, J. J. S. Metodologia da pesquisa científica. Fortaleza: UEC, 2002. Apostila.
- Gil, A. C. Como elaborar projetos de pesquisa. 4. ed. São Paulo: Atlas, 2007.
- Jain, Raj. The art of computer systems performance analysis: techniques for experimental design, measurement, simulation and modeling. New York: John Wiley & Sons, 1991. 685p. il. (Wiley Professional Computing).
- Jmeter, A. The apache software foundation. Disponível em: <http://jmeter.apache.org/>. Acesso em: 05 jun. 2016.
- Leffingwell, D; Muirhead, D. Tactical Management of Agile Development: Achieving Competitive Advantage. Boulder, Colorado, 2004.
- Menascé, Daniel A.; Almeida, Virgílio A. F. Planejamento de Capacidade para Serviços na Web: Métricas, modelos e métodos. Rio de Janeiro: Campus, 2003.
- Menéndez, A. I. M. Uma ferramenta de apoio ao desenvolvimento de Web Services. Dissertação de Mestrado, Universidade Federal de Campina Grande, curso de Pós-Graduação em Informática, 2002.
- Mumbaikar, S; Padiya P. Web Services Based On SOAP and REST Principles. International Journal of Scientific and Research Publications, Volume 2, Issue 5, Mai 2013, ISSN 2250-3153.
- Padilha, B. Projeto e Implementação de Serviços Analíticos para o DataUSP-PósGrad. Monografia – Instituto de Matemática e Estatística, Universidade de São Paulo USP, São Paulo, 2013.
- Shi, X. Sharing service semantics using SOAP-based and REST Web services. IT Professional (8): 18-24, mar./abr. 2006.
- Streibel, M. Implementando Web Services. Trabalho de Especialização, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2005.
- W3schools. XML SOAP, 2016. Disponível em: http://www.w3schools.com/xml/xml_soap.asp. Acesso em: 30 mai. 2016.