

## **Avaliação de Desempenho de Serviços Web SOAP e Restful: estudos de caso para embarcar em dispositivos na Internet das Coisas**

**Daniela Barreiro Claro<sup>1,2</sup>, George Pacheco<sup>2</sup>, Mario Romeu da Silva Junior<sup>2</sup>, Alberto Vianna<sup>2</sup>, Igor Lopes de Faria<sup>2</sup>, George C. G. Barbosa<sup>1,2</sup>, Diógenes Pires<sup>2</sup>, Flávio Dusse<sup>2</sup>, Leandro Andrade<sup>2</sup>, Milton Ferreira<sup>2</sup>, Robespierre Pita<sup>2</sup>**

<sup>1</sup>FORMAS – Grupo de Pesquisa em Formalismos e Aplicações Semânticas  
Departamento de Ciência da Computação - Universidade Federal da Bahia

<sup>2</sup>PGComp – Programa de Pós-Graduação em Ciência da Computação  
Departamento de Ciência da Computação - Universidade Federal da Bahia

[Dclaro@ufba.br](mailto:Dclaro@ufba.br), [georgepacheco1@gmail.com](mailto:georgepacheco1@gmail.com), [mario.romeu@gmail.com](mailto:mario.romeu@gmail.com),  
[albertoviannadias@gmail.com](mailto:albertoviannadias@gmail.com), [igorlopesfaria@gmail.com](mailto:igorlopesfaria@gmail.com),  
[gcgbarbosa@gmail.com](mailto:gcgbarbosa@gmail.com), [diogenespires@bol.com.br](mailto:diogenespires@bol.com.br), [dussebr@gmail.com](mailto:dussebr@gmail.com),  
[pierre.pita@gmail.com](mailto:pierre.pita@gmail.com), [miltonpferreira@gmail.com](mailto:miltonpferreira@gmail.com),  
[leandrojsa@dcc.ufba.br](mailto:leandrojsa@dcc.ufba.br),

**Resumo.** *O uso de serviços web embarcados em dispositivos eletrônicos tem crescido concomitantemente com o uso da Internet das Coisas. Diversos dispositivos têm utilizado o formato de serviços web para prover a interoperabilidade e difundir determinadas funcionalidades do próprio dispositivo. Porém, pouco se analisa em relação ao desempenho dos serviços embarcados. Este artigo propôs uma metodologia específica para avaliar o desempenho dos serviços web baseados em SOAP e em REST na Internet das Coisas. Os resultados obtidos demonstraram que ambos os serviços são similares em relação ao desempenho, entretanto, algumas diferenças ainda são observadas e são discutidas neste trabalho.*

### **1. Introdução**

Atualmente muito se tem discutido a respeito da Internet das Coisas (*Internet of Things* – IoT), cujo propósito está na conexão de “coisas” do mundo físico ao mundo virtual da Web de forma que estes passem a ser reconhecidos e tratados como qualquer outro recurso, usando para isso protocolos e padrões já existentes, como por exemplo HTTP (*Hypertext Transfer Protocol*) e URIs (*Uniform Resource Identifier*) [GUINARD 2010].

A possibilidade de inserção de diferentes dispositivos físicos (“coisas”) na Web representa um cenário propício para utilização dos dados e serviços compartilhados por eles na concepção de novas aplicações. Um fator relevante, que de certa forma pode tornar-se impeditivo nesse contexto, está atrelado ao formato de como eles se conectam à Internet, já que, em sua maioria, esta conectividade é subsidiada por softwares dedicados e interfaces proprietárias [GUINARD 2010]. Em se tratando dos mesmos fornecedores, a comunicação entre dispositivos muitas vezes ocorre de maneira fácil e rápida. Com o avanço e a proliferação da Internet das Coisas, novos dispositivos estão sendo incorporados na rede, independentes dos fabricantes, o que vem dificultando a comunicação e conseqüentemente a interoperabilidade entre estes dispositivos. Neste

sentido, interoperar dois dispositivos distintos, de fabricantes distintos com o intuito que haja uma mínima comunicação entre eles, tem se tornado uma tarefa árdua para a maioria dos desenvolvedores.

A incorporação de serviços web dentro de dispositivos tem por principal objetivo facilitar a interoperabilidade entre os dispositivos e conseqüentemente ampliar a troca de mensagens entre dispositivos heterogêneos. Embarcar serviços web em placas como Arduino, Raspberry e até mesmo VANT (Veículos Aéreos Não Tripulados) têm sido um dos grandes desafios. Devido à baixa capacidade de recursos disponíveis nestas placas e conseqüentemente nos Drones, surge uma necessidade de avaliar os serviços web para este propósito específico. Os dois principais tipos de serviços web que vem sendo utilizados são os serviços web que utilizam o protocolo SOAP (Simple Object Access Protocol) e os serviços web baseados na arquitetura REST (*Representational State Transfer*). Ambos os serviços têm sido utilizados com os dispositivos eletrônicos na Internet das Coisas.

Diversos frameworks já foram propostos com o intuito de desenvolver serviços web, dentre os quais destacam-se: Axis2 [Axis2 2015], o CXF [CXF 2015] e o Zend Framework 2 [Zend2 2015]. Com o intuito de embarcar serviços web em dispositivos, alguns testes foram realizados a fim de avaliar os seus respectivos desempenhos. Neste contexto, o presente trabalho propõe uma análise de desempenho de serviços web (SOAP e REST) utilizando os frameworks Axis2, tanto para notebook quanto em uma placa Raspberry Pi, CXF, Zend framework 2 e ainda um Barramento de Serviços para a Internet das Coisas.

O presente artigo está assim organizado. A seção 2 apresenta a metodologia utilizada para o desenvolvimento deste trabalho, a seção 3 apresenta uma descrição dos frameworks e dos experimentos realizados, a seção 4 apresenta os respectivos resultados e a seção 5 apresenta algumas conclusões e direcionamentos futuros.

## 2. Metodologia

A metodologia adotada neste trabalho foi desenvolvida baseada em estudos de casos realizados por alunos da Pós-Graduação da Universidade Federal da Bahia com o intuito de avaliar o desempenho dos serviços web para a sua possível utilização na Internet das Coisas.

Os alunos foram divididos em duplas e cada dupla ficou responsável por uma análise de desempenho. Houve uma maior exploração no framework Axis2, por se tratar de um framework bem difundido e muito utilizado no desenvolvimento tanto de serviços web baseados em REST quanto SOAP, independente dos serviços serem embarcados.

As métricas utilizadas e as ferramentas utilizadas foram definidas previamente com o intuito de que todas as duplas executassem os mesmos conjuntos de testes e assim os resultados pudessem ser comparados. Os resultados obtidos foram comparados entre SOAP e REST com o intuito de garantir as configurações de cada máquina na qual os testes foram executados. Os experimentos e resultados são apresentados nas próximas seções deste trabalho.

## 3. Experimentos

Os experimentos controlados foram realizados com o intuito de obter resultados factíveis de comparação.

Quanto ao serviço utilizado, foi desenvolvido uma operação simples, referente a uma calculadora, cujo objetivo era somar dois valores inteiros e retornar um inteiro como resultado.

```
public class CalculadoraSoma {  
    public int soma(int i1, int i2) {  
        return i1 + i2; }  
}
```

A utilização de um serviço simples foi de minimizar os possíveis ruídos, mantendo o foco na análise dos ambientes utilizados.

Em seguida, a quantidade de requisições foi definida com o intuito de ter uma média mais precisa, considerando as cargas de inicialização e alguns problemas eventuais no tráfego de dados. Todos os experimentos foram executados com 1000 requisições e a média foi obtida. Todas as primeiras medições foram descartadas, com o intuito de retirar a primeira carga dos serviços web.

A medição do desempenho foi realizada por uma ferramenta denominada Apache JMeter 2.13 [JMeter 2015]. O JMeter é uma ferramenta para executar testes de carga e stress em aplicações Web e possui funções de análise de resultado de testes de desempenho, incluindo tamanho das requisições/respostas em bytes e geração de gráficos comparativos dos tempos de resposta. JMeter foi configurada para disparar automaticamente 1000 requisições HTTP para cada tipo de protocolo, destinadas a cada um dos serviços web (SOAP e REST).

Os experimentos foram divididos em Axis 2, CXF, Zend Framework 2 e Barramento de Serviços. Cada experimento foi conduzido por uma dupla de alunos e descrito conforme abaixo:

### 3.1. Axis2

Neste primeiro experimento foi utilizado o Apache Axis 2, versão 1.6.2, para a criação de ambos os serviços web de forma transparente. Com o intuito de minimizar os ruídos nos resultados, o Axis2 foi utilizado em modo *standalone*, funcionando como um servidor de aplicação Web. A comparação entre serviços web (SOAP e REST), ambos os modelos executam o mesmo serviço denominado *CalculadoraSoma*, conforme descrito na seção anterior. Além desta avaliação, o framework Axis2 foi avaliado em mais dois aspectos: quanto aos serviços *stateless* vs *stateful* e quanto ao Axis 2 embarcado.

#### 3.1.1 Axis 2: Raspberry Pi

A placa controladora utilizada foi uma Raspberry Pi de 900 MHz quad-core ARM Cortex-A7, 1 GB RAM, MicroSDHC slot, contendo um SD Card de 16 GB, Classe 4.

#### 3.1.2 Axis 2: Stateful x Stateless

O protocolo SOAP, por padrão, não suporta a conservação de estado durante a troca de mensagens [W3C 2007]. Essa limitação define os serviços web como *stateless*. Contudo, o Axis2 permite a criação de serviços web *stateful* através da manipulação de sessões dentro do servidor de aplicação. Existem alguns escopos de sessões implementados pelo Axis2, como: *request session scope*, *SOAP Session Scope*, *Transport session scope* e *Application scope*. Eles definem as propriedades e o ciclo de vida das sessões no

servidor. Para este trabalho, o escopo de aplicação (*Application scope*) foi utilizado visto que: (i) não é necessário adicionar dados na requisição (como o `JSessionId`); (ii) o escopo é carregado somente uma vez e permanece carregado no servidor até o final da sua existência, evitando sobrecarga de processamento. Foram criados dois serviços diferentes para esta medição: *stateless* (CalculadoraSoma) e um segundo serviço, no qual o estado do resultado da soma é mantido. Os testes de medição foram realizados comparando os resultados de serviços *stateless* e *stateful* implementados com o Axis2.

### 3.2. CXF Framework

Apache CXF é um framework open source que dá suporte a criação e consumo de serviços web. Ele permite que os serviços sejam desenvolvidos por meio de APIs como JAX-WS e JAX-RS e oferece suporte a uma variedade de protocolos de mensagem como SOAP, XML/HTTP, RESTful HTTP ou CORBA. Este framework atende a uma ampla variedade de protocolos de transportes, tais como HTTP, JMS ou JBI [CXF 2015]. A versão utilizada no experimento é a versão 3.04 do CXF. Utilizou-se também o Apache Tomcat 7.0.59 e o Eclipse Kepler como IDE.

### 3.3. Zend Framework 2

O ZendFramework 2 (ZF2) é um framework de propósito geral desenvolvido em PHP que fornece uma variada biblioteca para suprir demandas de vários tipos de aplicações Web. O ZF2 não foi desenhado especificamente para trabalhar com serviços web, porém dispõe de classes para trabalhar com REST e SOAP. Neste trabalho foram utilizadas as classes `Zend\Soap\Server` e `Zend\Soap\AutoDiscovery` para trabalhar com SOAP e a classe `Zend\Mvc\Controller\AbstractRestfulController` para REST. A classe `Server` do pacote `Soap` processa as requisições submetidas ao servidor, enquanto a classe `AutoDiscovery` gera automaticamente o documento WSDL de descrição do serviço. Já a classe `AbstractRestfulController` fornece uma abstração para a implementação de um serviço RESTful.

### 3.4. Barramento de Serviços

A falta de protocolos de comunicação claros, padronizados e interoperáveis, bem como o uso de linguagens distintas por dispositivos na Internet das Coisas, exige um conhecimento específico para cada componente [GUINARD e TRIFA 2009]. Neste sentido, se faz necessário a definição de uma infraestrutura que possibilite gerenciar a publicação, descoberta, composição, utilização e compartilhamento de dispositivos na IoT. Neste cenário, um barramento de serviços (Enterprise Service Bus – ESB) pode ser utilizado como infraestrutura [SILVA e PRAZERES 2013], [GRAMACHO et al, 2013].

Com o intuito de avaliar o desempenho dos serviços nestes tipos de barramentos, utilizou-se uma implementação de ESB mantida pela fundação Apache, conhecida como ServiceMix. Foram desenvolvidos e implantados no ESB dois serviços: um baseado em REST e outro SOAP, ambos com a mesma funcionalidade: CalculadoraSoma.

## 4. Resultados

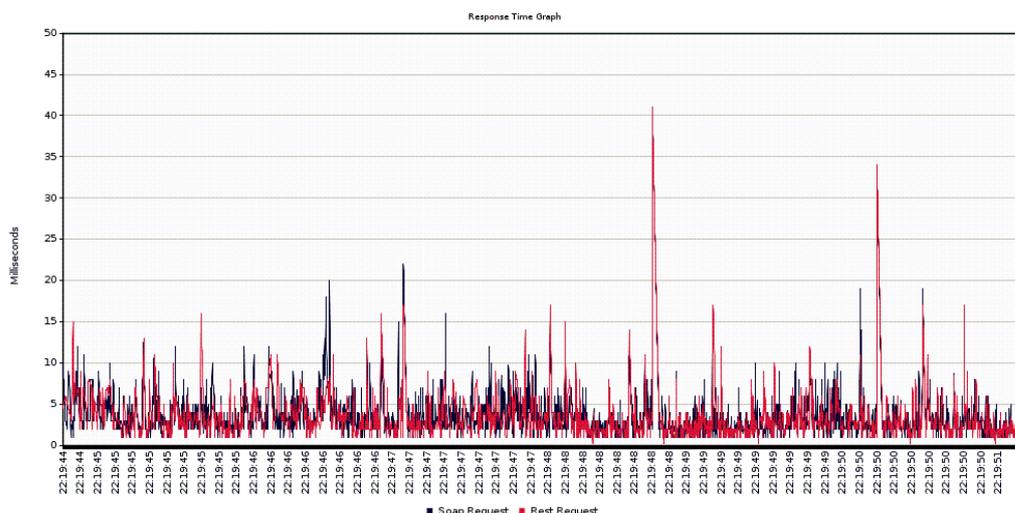
Os resultados foram referentes ao tempo de execução e ao tamanho dos bytes trafegados. Os resultados obtidos foram realizados através do Apache JMeter, conforme descrito na seção dos Experimentos.

## 4.1. Axis 2

O framework Axis2 foi testado em dois tipos de hardware diferente: notebook com as requisições locais para o servidor hospedado na mesma máquina (localhost). No segundo teste, o mesmo serviço foi embarcado em um RaspberryPi 2 Model B e o Jmeter foi executado a partir de um notebook. Os resultados das duas execuções são apresentados nas seguintes subseções.

### 4.1.1 Axis 2: Notebook

O hardware utilizado foi um notebook Intel(R) Core(TM) i3-3217U CPU @ 1.80GHz, 8Gb RAM DDR3, Solid State Drive 120GB Intel Sata3.



**Figura 1. Comparação de tempo de resposta Axis 2 em um Notebook (SOAP x REST).**

A Figura 1 apresenta as 1000 requisições para ambos os serviços (REST e SOAP). Efetivamente, não é possível verificar diferenças significativas no desempenho entre os serviços web, por terem tempos de respostas muito próximos. A flutuação destes valores de tempo total de execução entre o menor e maior valor se encaixa na maior parte dos casos na mesma faixa em ambos os protocolos. Em algumas execuções, o REST teve um desempenho melhor e em outras o SOAP, sem uma resposta justificável. Quanto ao tamanho das requisições/respostas (Tabela 1), efetivamente o SOAP possui um tamanho maior que no REST devido à estrutura do envelope SOAP no XML trafegado, o que causa um maior *overhead* envolvido no empacotamento e desempacotamento de mensagens.

**Tabela 1. Valores dos testes do Axis 2 (Soap x Rest) em um Notebook**

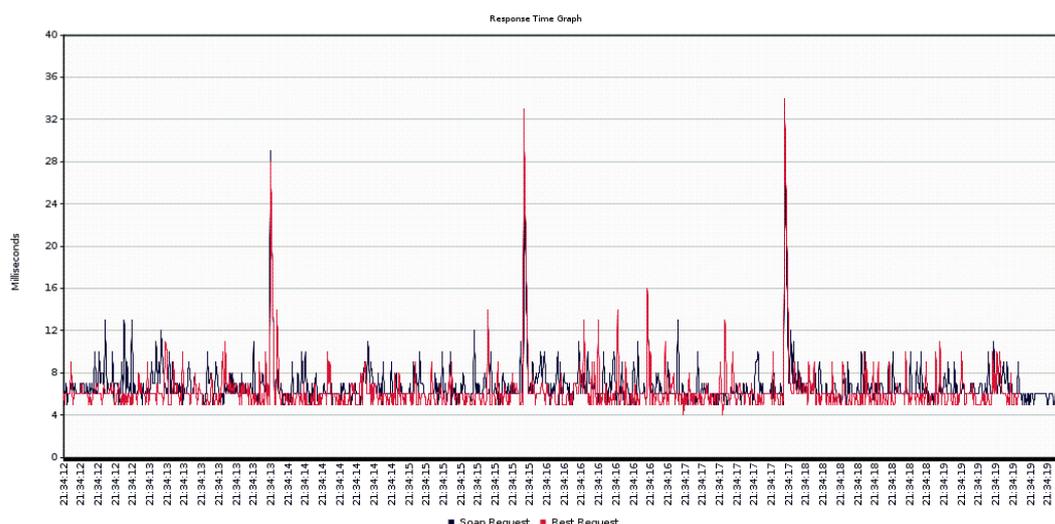
Modo	Média tempo de resposta (ms)	Tamanho do pacote (Bytes)
SOAP	2.04	437
REST	3.17	281

Todavia, este “custo adicional” do SOAP não é visto como uma característica que inviabiliza a tecnologia ou a coloca em um patamar inferior ao REST, tendo em vista que

este tamanho adicional do envelope somente possui impacto expressivo no tamanho do XML da requisição/resposta, em casos nos quais a mesma é tão pequena que poucos bytes adicionais representam uma porcentagem significativa no tamanho total do XML. Em casos de requisições reais com objetos complexos estes bytes adicionais do envelope SOAP são poucos expressivos.

#### 4.1.2 Axis 2: Raspberry pi 2 Model B, V 1.1

A Figura 2 apresenta as execuções de ambos os serviços web REST e SOAP na placa Raspberry Pi.



**Figura 2. Comparação de tempo de resposta Axis 2 em um Raspeberry pi (SOAP x REST).**

A Figura 2 apresenta as 1000 requisições para ambos os serviços web REST e SOAP. Efetivamente, não é possível verificar diferenças significativas no desempenho entre os serviços web (SOAP e REST), por terem tempos de respostas muito próximos.

**Tabela 2. Valores dos testes do Axis 2 (Soap x Rest) em um Raspeberry pi**

Modo	Média tempo de resposta (ms)	Tamanho do pacote (Bytes)
SOAP	12,72	367
REST	13,43	212

Quanto ao tamanho (Tabela 2), embora a mesma situação permaneça, ou seja, o SOAP tenha um envelope maior e conseqüentemente a quantidade dos dados trafegados também são maiores, este custo adicional é mínimo mesmo em se tratando de uma Raspberry Pi.

#### 4.1.3 Axis 2: Stateless X Stateful

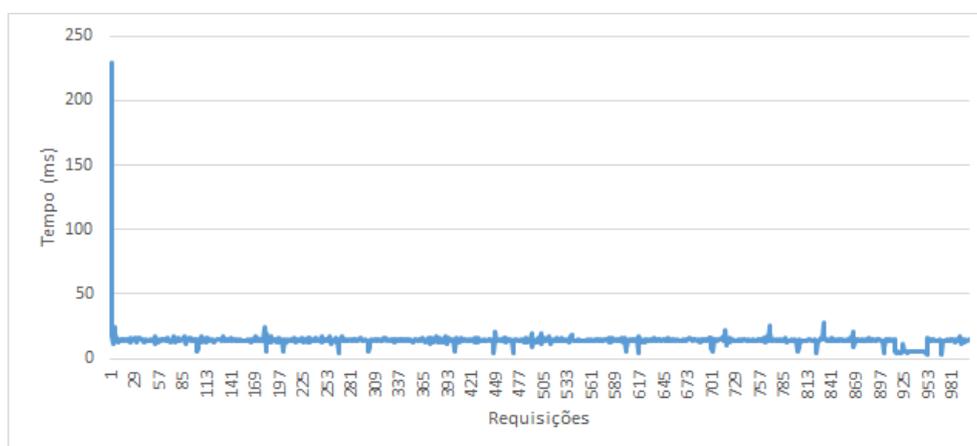
Os testes de comparação desses dois modos de serviços web (*stateless* e *stateful*) foram realizados utilizando a ferramenta JMeter. Na Tabela 3 é possível verificar a média dos valores de tempo de resposta obtidos durante as 1000 requisições realizadas e o tamanho

dos pacotes. Vale salientar que os testes foram realizados em um notebook HP com Intel core i5, 6 GB de memória RAM, no sistema operacional Windows 7.

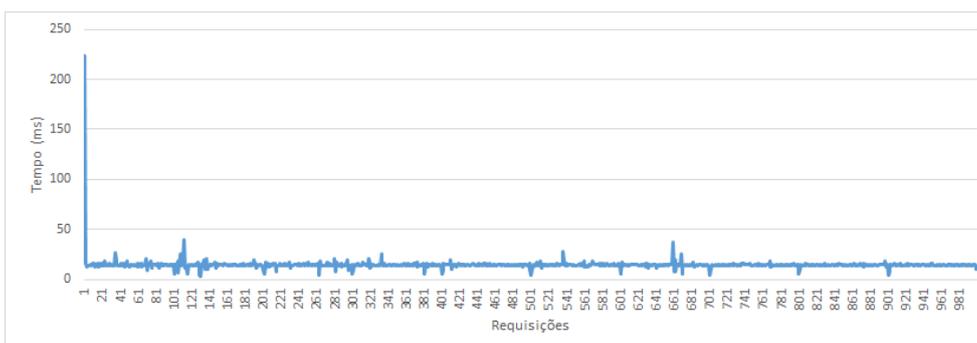
**Tabela 3. Valores obtidos nos testes dos serviços web (Stateful x Stateless)**

Modo	Média tempo de resposta (ms)	Tamanho do pacote (Bytes)
Stateless	14,23	259
Stateful	14,81	267

Observa-se que o serviço *stateless* obteve um desempenho superior ao serviço *stateful* (Tabela 3). Nas Figuras 3 e 4, pode-se visualizar uma representação das medições. Nelas é possível verificar como a primeira requisição é mais onerosa.



**Figura 3. Tempo das respostas do serviço stateless**



**Figura 4. Tempo das respostas do serviço stateful**

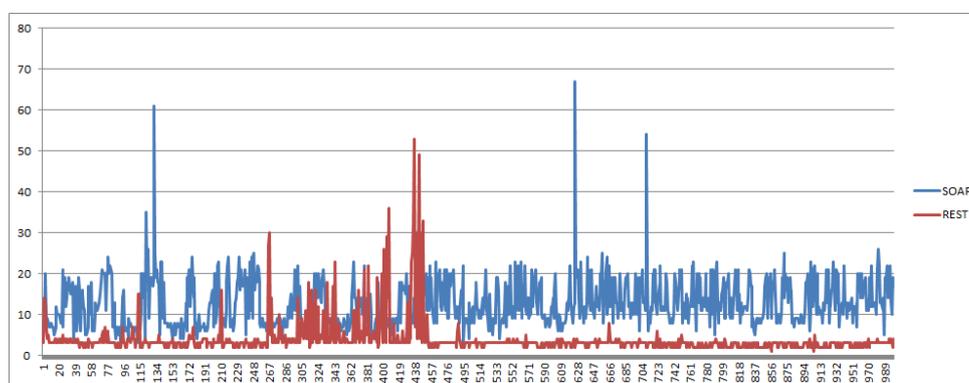
## 4.2. CXF Framework

Na Tabela 4 observa-se a média dos tempos de respostas às 1000 requisições realizadas para cada um dos serviços implantados, bem como o tamanho do pacote enviado. Novamente o método HTTP utilizado na requisição ao serviço REST foi o GET, enquanto requisições SOAP usam o método POST.

**Tabela 4. Valores obtidos nos testes no CXF**

Tipo	Média tempo de resposta (ms)	Tamanho do pacote (Bytes)
REST	3,88	163
SOAP	12,89	189

A partir dos valores constantes na Tabela 4, percebe-se que, nesse experimento, o tamanho do pacote enviado por uma requisição SOAP foi 14% maior em relação ao serviço REST e o tempo médio de resposta de um serviço SOAP mostrou-se aproximadamente 332% maior. Na Figura 5, pode-se visualizar uma representação das medições na qual em poucos casos o tempo de resposta do REST foi maior que a média do SOAP.

**Figura 5. Comparação de tempo de resposta no CXF (SOAP x REST)**

### 4.3. Zend Framework 2

Os resultados referentes ao Zend são apresentados na Tabela 5.

**Tabela 5. Métricas obtidas a partir de requisições REST e SOAP no Zend**

Modo	Média tempo de resposta (ms)	Tamanho do pacote (Bytes)
SOAP	31,89	527
REST	17,31	155

É notório observar que a quantidade de dados analisados em um pacote XML adicional encapsulado no protocolo HTTP, resultou em um aumento de 240% na quantidade de dados trafegados. O REST mantém métricas de desempenho e *overhead* 84.39% superiores ao SOAP. O que não indica superioridade do REST em qualquer cenário, pois, a aplicabilidade do SOAP por conta de sua portabilidade é ilimitada.

### 4.4. Barramento de Serviços

Na Tabela 6 observa-se a média dos tempos de respostas em relação às 1000 requisições realizadas para cada um dos serviços implantados, bem como o tamanho do pacote enviado. Nesse ponto vale salientar que o método HTTP utilizado na requisição ao serviço REST foi o GET, enquanto requisições SOAP usam o método POST.

**Tabela 6. Valores obtidos nos testes no Apache ServiceMix**

Tipo	Média tempo de resposta (ms)	Tamanho do pacote (Bytes)
REST	1,49	228
SOAP	2,15	217

A partir dos valores constantes na Tabela 6, percebe-se que, nesse experimento, o tamanho do pacote enviado por uma requisição SOAP foi 5% menor em relação ao serviço REST, por outro lado o tempo médio de resposta de um serviço SOAP mostrou-se aproximadamente 45% maior.

## 5. Trabalhos Relacionados

No trabalho de [ESTRELLA et al, 2008] são discutidas diretrizes para se efetuar uma avaliação de desempenho de serviços web baseados na arquitetura SOA. No trabalho, alguns parâmetros de Qualidade de Serviço (QoS) são definidos, tendo como objetivo medir o desempenho do serviço. Os autores também discutem como deve ser o método de avaliação e pontos importantes que devem ser abordados durante o processo de medição de um serviço web. Diferentemente do presente trabalho, os autores não abordam os serviços web relativos à Internet das Coisas.

Em [NUNES et al, 2014] é discutido o desempenho de serviços web REST em dispositivos com recursos limitados, no caso do experimento, fazendo uso do Raspberry Pi. Diferentemente do presente trabalho, nenhuma comparação com serviços web SOAP em dispositivos e em barramento foi efetivada.

Autores em [Hamad et al, 2010] propõem um trabalho semelhante ao apresentado neste artigo, comparando o desempenho do tempo de resposta e o tamanho da mensagem em byte de serviço web RESTful e SOAP. Diferentemente deste trabalho, os serviços implementados são avaliados para dispositivos móveis e não fazem uso da Internet das Coisas, ou seja, não consideram um dispositivo com recursos limitados e nem um barramento de serviços. Além do mais, nenhuma destas abordagens se propõe a avaliar os serviços web *stateless* e *stateful*.

## 6. Conclusão

O presente trabalho teve como principal objetivo avaliar os serviços web baseados em REST e SOAP, tanto em um framework difundido e amplamente conhecido como o Axis 2, quanto em um dispositivo com recursos limitados. Além disso, outros frameworks mais atuais foram avaliados como o CXF e o ZF2, além de avaliar os serviços *stateful* e *stateless*. Os resultados encontrados mostraram que não há diferenças significativas em relação à execução de serviços web REST e SOAP. Porém, em relação à quantidade dos dados trafegados, por se tratar de um envelope e XML, há alguma diferença mais significativa. Como trabalhos futuros, pretende-se incorporar estes serviços web em dispositivos distintos com o intuito de avaliar o desempenho de cada um deles em relação a ambos os serviços web.

## Referências

- Boulic, R. and Renault, O. (1991) “3D Hierarchies for Animation”, In: *New Trends in Animation and Visualization*, Edited by Nadia Magnenat-Thalmann and Daniel Thalmann, John Wiley & Sons ltd., England.
- Dyer, S., Martin, J. and Zulauf, J. (1995) “Motion Capture White Paper”, [http://reality.sgi.com/employees/jam\\_sb/mocap/MoCapWP\\_v2.0.html](http://reality.sgi.com/employees/jam_sb/mocap/MoCapWP_v2.0.html), December.
- Guinard, D. Towards Opportunistic Applications in a Web of Things. In: *IEEE International Conference on Pervasive Computing and Communications Workshops*, p. 863-864, 2010.
- Guinard, D. and Trifa, V. Towards the Web of Things: Web Mashups for Embedded Devices. In: *Proceedings of Workshop on Mashups, Enterprise Mashups and Lightweight Composition on the Web, International World Wide Web Conferences*, Madrid, Spain, 2009.
- Gramacho, S. L. D. L. ; Prazeres, C. V. S. ; Figueiredo, G. B. . An ad-hoc web of things service bus. In: *the 19th Brazilian symposium, 2013, Salvador. Proceedings of the 19th Brazilian symposium on Multimedia and the web - WebMedia '13*. New York: ACM Press, 2013. p. 233-236.
- Silva, C. H. F. ; Prazeres, C. V. S. . Barramento de serviços para publicação de dispositivos na Web das Coisas. In: *X Workshop de Trabalhos de Iniciação Científica em Sistemas Multimídia e Web, 2013, Salvador. Anais do WTIC do WebMedia 2013, 2013*.
- W3C (2007) “SOAP Version 1.2 Part 1: Messaging Framework (Second Edition)”, <http://www.w3.org/TR/soap12/>, Acesso em: Abril, 2015.
- Estrella, J.C ; Santana R.H.C ; Santana M.J. ; Tavares T.C ; Kuehne B.T. ; Silva J.C.F ; Paccanaro L.C. Diretrizes para Avaliação de Desempenho de Web Services. *Wperformance - VII Workshop em Desempenho de Sistemas Computacionais e de Comunicação, 2008, Belém*.
- Nunes, L.H. ; Nakamura L.H.V. ; Libardi, R.M. ; Estrella J.C. ; Santana M.J. ; Santana R.H.C. Análise de Desempenho em Dispositivos Limitados e Emulados. Estudo de caso: Raspberry Pi e Web Services RESTful. *Wperformance - XIII Workshop em Desempenho de Sistemas Computacionais e de Comunicação, 2014, Brasília*.
- HAMAD, Hatem; SAAD, Motaz; ABED, Ramzi. Performance Evaluation of RESTful Web Services for Mobile Devices. *Int. Arab J. e-Technol.*, v. 1, n. 3, p. 72-78, 2010.
- Axis2. Apache Axis 2. Disponível em: <<http://ws.apache.org/axis2/>>. Acessado em 12 de abril de 2015.
- CXF. Apache CXF. Disponível em: <<http://cxf.apache.org/>>. Acessado em 12 de abril de 2015.
- JMeter. Apache Jmeter. Disponível em: <<http://jmeter.apache.org/>>. Acessado em 12 de abril de 2015.
- Zend2. Zend Framework 2. Disponível em : <<http://framework.zend.com/>>. Acessado em 12 de abril de 2015.