ISSN: 1226-7244 (Print)
ISSN: 2288-243X (Online)
논문번호 14-01-18
146

Journal of IKEEE.Vol.18,No.1,146~151,March 2014
http://dx.doi.org/10.7471/ikeee.2014.18.1.146

SOAP/RESTFUL 기반 웹 서비스 미들웨어 성능분석

Performance Analysis of Web Service Middleware based on SOAP/RESTFUL

송 병 권*

Byung-Kwen Song*

Abstract

The Smart Grid provide a lot of additional service between ICT(Information Communication Technology) and fusion service. The smart grid using various protocol need to use independent middleware system for exchanging information data. The application system of Smart Grid is exchanging information on the almost All-IP system. The current All-IP middleware system consist of SOAP and RESTful. This paper analysis SOAP and RESTful. This result make use of building Smart Grid System.

요 약

스마트 그리드는 기존 전력망에 ICT(Information Communication Technology)기술과 타산업간의 지속적인 융합으로 다양한 부가서비스를 제공한다. 다양한 프로토콜을 사용하고 있는 스마트 그리드 응용시스템이 상호간에 독립적으로 정보를 교환하기 위해서는 기존 프로토콜에 독립적인 미들웨어 사용이 필요하다. 스마트 그리드 응용시스템들은 대부분 All-IP 기반에서 정보 교환이 이루어진다. 현재 All-IP 기반의 미들웨어는 대표적으로 SOAP(Simple Object Access Protocol)과 RESTful(REpresentational State Transfer ful)이 있다. 본 논문에서는 SOAP과 RESTful의 성능을 비교 분석하여 향후 스마트 그리드 응용 시스템 구축 시 사용할 수 있는 SOAP와 RESTful의 성능을 비교분석하였다.

Key words : Web Service, SOAP, RESTful, Performance Analysis, Middleware

Ⅰ. 서론

스마트 그리드는 통신, 네트워크, 자동제어 등의 ICT 기술을 접목시킨 차세대 전력망이다. 이것은 전력 인프라의 상호 호환성, 신뢰성, 효율성, 안전성을 향상시켜 고품질의 에너지와 다양한 부가서비스를 제공한다. 또한, 스마트 그리드는 소비자와 공급자간에 양방향으로 실시간 전력정보를 교환함으로써 에너지 효율을 극대화 시킬 수 있다. [1]

AMI(Advanced Metering Infrastructure), 디지털 변전소, 스마트 배전시스템 및 기존 배전 자동화 시스템 등 다중 전력IT 시스템으로 발전 하고 있는 스마트 그리드는 다양한 통신데이터와 프로토콜이 사용되고 있다. 통합된 하나의 시스템으로 보이기 위해서는 투명성을 제공하는 웹 서비스를 스마트 그리드에 접목시켜 효율을 향상시킬 수 있다.

웹 서비스(Web Services)는 W3C(World Wide Web Consortium)에서 정의한 서비스로서 네트워크를

* Dept. of Electronics Engineering, Seokyeong University
bksong@skuniv.ac.kr, 02-940-7739

※ Acknowledgment
"This research was supported by the research grant of Seokyeong University(2013) and the Energy Technology Development Project through Korea Institute of Energy Technology Evaluation and Planning (KETEP) funded Ministry of Trade, Industry & Energy(MOTIE) (No. : 20131020402080)."
Manuscript received Mar. 3, 2014; revised Mar. 18, 2014 ; accepted Mar, 18. 2014

통해 서로 상이한 시스템 간 상호 정보 교환을 위한 소프트웨어 시스템 서비스 지향형 분산 시스템의 일부라고 말할 수 있다. 이러한 웹 서비스는 HTTP(HyperText Transfer Protocol)와 같은 표준 인터넷 프로토콜로 정보를 교환 할 수 있어 프록시(proxy)나 방화벽(firewall)에 제한 없이 데이터를 교환 할 수 있다는 장점이 있다.

현재 사용되는 있는 웹 서비스 미들웨어는 크게 XML (eXtensible Markup Language)기반의 SOAP과 URI(Uniform Resource Identifier) 기반의 RESTful로 구분할 수 있다. [2][3]

본 논문의 구성은 다음과 같다. 본론의 1절과 2절에서는 SOAP과 RESTful의 전반적인 내용을 기술한 뒤, 다음 3절에서는 SOAP과 RESTful의 성능 분석 방법에 대해서 기술한다. 마지막 4절에서는 앞에서 언급한 내용을 토대로 도출된 실험결과와 추후과제에 대해 기술 하였다.

II. 본론

1. SOAP

SOAP은 SOA(Service Oriented Architecture)개념을 사용하여 웹 서비스를 기술하는 WSDL(Web Services Description Language)과 서비스 위치 정보를 저장하거나 제공하는 UDDI(Universal Description Discovery and Integration)로 구성되어 있다.

가. 정의

SOAP은 데이터 표현에 있어서 XML로 데이터를 기술하기 때문에 XML 파싱이 가능한 환경에서는 전송되는 메시지를 쉽게 이해할 수 있으며, 인터넷상 데이터 교환에 유리하다. 또한 윈도우나 리눅스와 같은 특정 플랫폼으로부터 독립적이며, 프로토콜을 HTTP을 사용하기 때문에 방화벽에 자유로워, 데이터 교환에 매우 유리하다.

나. SOAP 서비스 구조

SOAP 서비스 구조는 [그림 1]과 같다. SOAP에서 서비스 제공자(Service Provider)는 UDDI Registry를 통해 서비스를 등록(Publish)하고, 서비스 요청자(Service Requester)는 UDDI Registry에서 탐색하는(Find) 구조를 가지고 있다. 이러한 SOAP 서비스에서 사용하는 메시지는 XML로 기술하기 때문에 쉽게 확장이 가능하다.

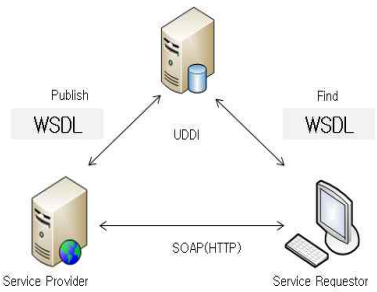

Fig. 1. The organization of SOAP Service
그림 1. SOAP 서비스 구조

다. SOAP 메시지

SOAP은 메시지는 다양한 전송 프로토콜에 사용할 수 있으며, HTTP 프로토콜을 가장 많이 사용한다. HTTP를 통해서 전송된 SOAP은 다음과 같다.

```
POST /SOAPserver/services/SOAPhello?wsdl HTTP/1.1
Content-Type: text/xml; charset=UTF-8
SOAPAction: "urn:getHello"
User-Agent: Axis2
Host: 210.110.34.175:8080
Content-Length: 10233

<?xml version='1.0' encoding='UTF-8'><SOAP-ENV.....
```

Fig. 2. the example of SOAP over HTTP binding
그림 2. HTTP 바인딩 된 SOAP의 예

[그림 2]에서 보면 알 수 있듯이 HTTP프로토콜에 바인딩 되어 있으며, HTTP 프로토콜 내부를 통해서 SOAP의 메시지 종류와 SOAP 클라이언트가 서버측에 요청한 SOAP의 기능, SOAP의 데이터 크기, 인코딩 방식에 대해서 기술된다.

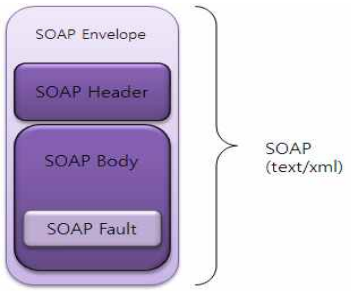

Fig. 3. The organization of SOAP Message
그림 3. SOAP 메시지 구조

SOAP은 이기종의 플랫폼에서 동작하는 어플리케이션간의 연동을 목적으로 상호 이해 가능한 메시지를 송수신함으로 원격지에 있는 서비스의 객체나 데이터 API를 자유롭게 사용할 수 있다.

SOAP에서 사용하는 메시지 구조는 [그림 3]과 같다. SOAP 메시지의 Envelope에는 SOAP Header, SOAP Body 그리고 SOAP Fault의 XML 형태로 구성된다. 이러한 SOAP 메시지 형태는 다음 [그림 3]과 같다. [4]

SOAP Envelope는 SOAP 메시지를 포함하고 있으며, SOAP의 내부 구성 형태를 정의한다. SOAP Header는 SOAP을 사용하는 어플리케이션의 인코딩이나 암호화에 관련된 추가적인 정보를 포함한다. SOAP Body는 SOAP 메시지에서 실제로 전송되는 데이터를 포함한다. SOAP Fault는 에러가 발생하면 Body 안에 표현 되며, 에러 메시지를 나타낸다. Fault에는 필수 요소인 Code, 와 Reason 요소들이 포함 되어 있다.

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope
    xmlns:SOAP-ENV=
    "http://schemas.xmlsoap.org/soap/envelope">
    <SOAP-ENV:Body>
        <setProtocol
        xmlns="http://protocol.device.com">
        iec61850
        </setProtocol>
        <getDeviceNumber
            xmlns="http://example.device.com/ws">
                <productId>1987554</productId>
            </getDeviceNumber>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Fig. 4. The Example of SOAP Message
그림 4. SOAP 메시지 형태의 예

2. RESTful

가. 정의

RESTful은 ROA(Resource Oriented Architecture)를 근거로 하여 서비스 요청자가 필요한 그래픽, 오디오, 영상, 텍스트, 하이퍼링크와 같은 리소스를 요청하면 중간 매개체 없이 서비스 제공자는 해당 리소스를 직접 응답해 주는 웹 서비스 미들웨어이다.

특히 RESTful 미들웨어는 기존 SOAP에서 사용하는 XML을 사용할 수도 있고, 모바일 환경이나 무선 센서 네트워크와 같은 저 사양의 시스템에서는 XML 대신 JSON(JavaScript Object Notation)이나 일반 텍스트 문자도 사용이 가능하다.

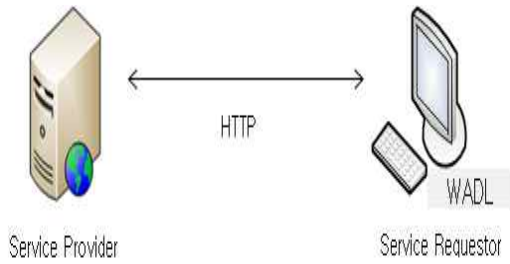

Fig. 5. The organization of RESTful Service
그림 5. RESTful 서비스 구조

나. RESTful 서비스 구조

RESTful은 서비스를 설명하는 WADL(Web Application Description Language)를 이용하여 원하는 리소스 정보를 구할 수 있다.[5] 또한 RESTful은 서버에서 클라이언트에 대한 정보를 저장, 보관, 관리하지 않는 무상태(Statelessness)을 특성을 가지고 있다. RESTful은 HTTP 프로토콜을 사용하며 데이터를 송수신한다. [6]

다. RESTful 메시지

```
GET /RestExample/apis/example/hello/msg=hello
```

Fig. 6. The example of RESTful over HTTP
그림. 6. HTTP 프로토콜을 이용하는 RESTful

[그림 6]과 같이 RESTful은 HTTP을 사용하며, RESTful은 HTTP 메소드를 사용하며 위의 그림은 GET 서비스를 사용한 예이다. RESTful은 URI는 주소이자 리소스의 이름을 뜻한다. URI를 사용하기 때문에 확장성이 좋다. 이러한 RESTful의 특징을 표현한 것이 [표 1] 이다.

이러한 RESTful은 부수적인 레이어나 세션 관리를 추가하지 않고 HTTP로 데이터를 전달하는 방식이다. 또한 구현을 단순화시키고, 확장성과 성능을 높일 수 있는 웹 서비스 미들웨어이다. 그리고 각각의 리소스들은 고유의 URI을 가지고 있으며, HTTP의 기본 메소드인 GET/PUT/POST/DELETE만으로 접근 할 수 있다.

Table 1. The Http method.
표 1. HTTP 메소드

HTTP 요청 메소드	
GET	리소스 조회
PUT	리소스 수정
POST	리소스 생성
DELETE	리소스 제거

HTTP 메소드의 GET은 클라이언트가 서버에게 찾고자 하는 리소스의 정보를 요청할 때 사용이 되며, PUT은 서버에 존재하는 리소스에 대해서 수정을 요청할 때 사용된다. POST는 기존에 존재하지 않는 리소스를 서버측에 생성 요청 하는 것이며, DELETE는 리소스를 제거 할 때 사용된다.

RESTful은 서비스의 요청이 클라이언트 측으로부터 발생되면 서버측은 그에 해당되는 응답코드를 보내주게 된다. HTTP의 응답코드에는 데이터 송수신이 성공했다는 것을 뜻하는 2XX번호 응답코드와 재전송을 나타내는 3XX번호 대역들이 있다. 4XX는 클라이언트의 데이터 요청이나 잘못된 접근이 발생했을 때 발생하는 번호이며, 5XX는 서버측에서 문제가 발생했을 때 클라이언트 측으로 전달해주는 응답 코드이다.

3. SOAP / RESTful 성능 비교

가. 실험 환경

서버와 클라이언트는 운영체제에 종속적이지 않는 Java로 구현하였다. 플랫폼은 윈도우 환경에서 동일하게 성능 비교를 하였다. [표 2]과 [표 3]는 서버와 클라이언트의 실험 환경을 나타낸다.

Table 2. The Spec of Service simulation
표 2. 서버 실험 환경

운영체제	Window7 Professional 64bits
CPU	Intel Core i5-2500, 3.5GHz(4core)
어플리케이션	Java 1.7.0_21
서버	Apache
메모리	삼성전자 DDR3 8G

Table 3. The Spec of Client simulation
표 3. 클라이언트 실험 환경

운영체제	Window7 Professional 64bits
CPU	Intel Core i3-2100,3.1GHz(4core)
어플리케이션	Java 1.7.0_21
메모리	삼성전자 DDR3 8G

4. 성능 비교 및 평가

가. 개요

SOAP과 RESTful의 성능 비교는 [표 4]에 나타난 성능 변수 설정에 따라 수행하였다.

Table 4. Perormance Parameters of Experimental Environment
표 4. 실험 환경 성능 변수

성능변수	Senario 1	Senario 2	Senario 3
송·수신 횟수(회)	1,000		
서비스 지연 시간(ms)	1,000		
데이터의 크기(byte)	1,000	2,000	3,000

데이터의 송·수신 횟수를 1,000번으로 하여 서버와 클라이언트 간에 많은 정보 교환이 발생하는 경우를 가정하였고, 서비스 지연 시간을 1,000ms로 주어 송·수신된 데이터를 처리 하는 시간으로 가정하였다. 그

리고 데이터의 크기에 변화를 주어 전송되는 데이터의 양에 따른 SOAP과 RESTful의 성능을 비교 분석하였다.

나. 성능 측정 방법

성능 비교의 지표는 서버와 클라이언트 간의 데이터를 송·수신하는 응답 시간(Round Trip Time : RT)을 기준으로 수행하였다. 클라이언트는 서버 측에 데이터를 보내기 전에 자신의 시스템 시간(Send Time : S_t)을 측정한 후에 데이터를 보낸다. 그 후 서버는 클라이언트로부터 받은 데이터를 처리 한 뒤 클라이언트에게 처리한 데이터를 전송한다. 클라이언트는 서버측로부터 처리된 데이터를 받을 때의 자신의 시스템 시간(Receive Time : R_t)을 측정한다. 이렇게 측정한 S_t와 R_t의 시간차를 통해 SOAP과 RESTful의 성능을 분석하였다.

Round Trip Time(RT)

= Receive Time(R_t) - Send Time(S_t)

측정된 RT의 값이 작을수록 서버와 클라이언트의 메시지 처리 및 데이터 송·수신 속도가 빨라 성능이 더 좋다는 것을 의미한다.

다. 성능 측정 결과 및 분석

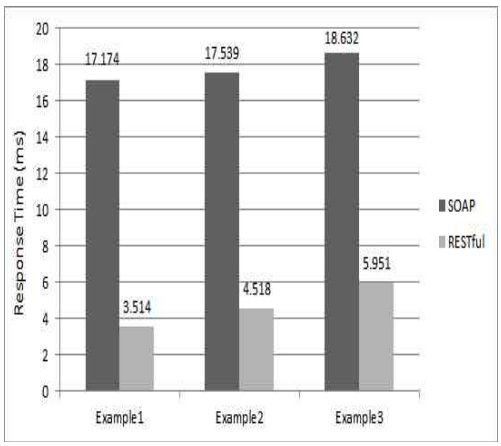

Fig. 7. The Response time of sending data size
그림 7. 데이터의 크기에 따른 응답 시간

SOAP과 RESTful의 성능을 비교한 결과 송·수신되는 데이터의 크기가 커질수록 SOAP과 RESTful 모두 긴 응답 시간이 측정되었다. 하지만 두 웹 서비스의 응답 시간은 확연하게 차이가 발생하였다.

응답 시간을 비교해 보면 데이터의 크기가 1,000byte일 경우 SOAP는 17.174ms인데 비해 RESTful은 3.514ms로 약4.887배, 데이터의 크기가 2,000byte일 경우 SOAP는 17.539ms인데 비해 RESTful은 4.518ms로 약 3.882배, 데이터의 크기가 3,000byte일 경우 SOAP는 18.632ms인데 비해 RESTful은 5.951ms로 약 3.131배가 차이났다. 평균적으로 RESTful의 응답 시간이 SOAP의 응답 시간보다 약 4배정도 빠른 결과가 도출이 되었다.

III 결론

본 논문에서는 윈도우 플랫폼에서 Java 기반의 SOAP과 RESTful의 성능을 비교분석하였다. 그 결과 RESTful이 SOAP보다 비교적 더 빠른 응답 시간이 측정되었다.

이러한 결과는 SOAP이 서비스 요청자와 제공자 간의 정보 교환에서 XML형식으로 된 데이터를 인코딩/디코딩 하는 작업이 오버헤드가 발생하기 때문에 응답시간이 RESTful보다 늦게 나왔다. 하지만 RESTful은 리소스의 URI를 사용하여 HTTP의 기본 메소드만으로 리소스 자체를 그대로 실어 나르기 때문에 RESTful이 빠른 응답 속도를 보였다.

이러한 차이점으로 인해 본 논문에서의 결과는 RESTful이 SOAP보다 더 우세한 성능을 보였다. 하지만 RESTful은 표준이 정해져있지 않기 때문에 개발, 관리 등이 SOAP보다 상대적으로 어렵다. 따라서 RESTful이 더 빠르다고 해서 무조건 RESTful을 사용하는 것이 아니라, 웹 서비스를 구축하는 시스템의 배경과 환경을 파악하여서 어떠한 웹 서비스 방식이 그 시스템에 잘 맞을 것인지를 분석한 후에 적절한 웹 서비스 방식을 사용해야 할 것이다.

추후 과제로는 스마트 그리드의 전력IT에서 사용하는 IEC61850, DNP3.0(Distributed Network Protocol), DLMS(Device Language Message Specification)프로토콜을 SOAP과 RESTful방식의 웹 서비스 방식으로 변경하여 성능 평가가 필요하다.

References

[1] Jae-Chul Kim, Sung-Min Cho, Hee-Sang Shin, “Advanced Power Distribution System Configuration for Smart Grid”, IEEE transactions on smart grid, v.4 no.1 p353-358, 2013
[2] Y.M.Pakr, A.K.Moon, H.K.Yoo, Y.C.Jung, S.K.Kim, “SOAP-based Web Services vs. RESTful Web Services”, Electronics and Telecommunications Trends, ETRI, Vol.25, No.2, pp.112-120, 2010
[3] Sam Brannen, “Spring Web Services : SOAP vs. RESTful”, Swiftmind
[4] W3C , www.w3c.org
[5] Leonnard Richardson, Sam Rudy, “RESTful Web Services”, O’Reilly Media Inc., 2007
[6] Sang-il Kim, Hwa-sung Kim, "API Selection and Automatic Open API Composition Method Based on REST Protocol", The Journal of the Korea Information and Communications Society, v38C, no.7 pp.587-594, 2013

BIOGRAPHY

Song Byungkwen (Life Member)
Reference of IKEEE Journal Vol. 17 No. 2