chips

MDPI

Communication

FPGA Prototyping of Web Service Using REST and SOAP Packages

Chee Er Chang [1], Azhar Kassim Mustapha [1] and Faisal Mohd-Yasin [2,*]

[1] Faculty of Engineering, Multimedia University, Cyberjaya 63000, Selangor, Malaysia
[2] School of Engineering and Built Environment, Griffith University, Nathan, QLD 4111, Australia
* Correspondence: f.mohd-yasin@griffith.edu.au

Abstract: This Communication reports on FPGA prototyping of an embedded web service that sends XML messages under two different packages, namely Simple Object Access Protocol (SOAP) and Representational State Transfer (REST). The request and response messages are communicated through a 100 Mbps local area network between a Spartan-3E FPGA board and washing machine simulator. The performances of REST-based and SOAP-based web services implemented on reconfigurable hardware are then compared. In general, the former performs better than the latter in terms of FPGA resource utilization (~12% less), message length (~57% shorter), and processing time (~4.5 µs faster). This work confirms the superiority of REST over SOAP for data transmission using reconfigurable computing, which paves the way for adoption of these low-cost systems for web services of consumer electronics such as home appliances.

Keywords: web service; FPGA; REST; SOAP; IoT

Citation: Chang, C.E.; Mustapha, A.K.; Mohd-Yasin, F. FPGA Prototyping of Web Service Using REST and SOAP Packages. *Chips* **2022**, *1*, 210–217. https://doi.org/10.3390/chips1030014

Academic Editor: Gaetano Palumbo

Received: 8 September 2022
Accepted: 30 November 2022
Published: 5 December 2022

1. Introduction

An embedded system is a microprocessor that performs specific tasks. It ranges from a single processor with only a few electronic components to a complicated system with multiple chips on electronic boards. An embedded system is the key element to create the peer-to-peer network of smart devices across the Internet, commonly known as the Internet-of-Things (IoT) [1]. Various web applications are being developed to perform different operations such as information searching, data transferring, and device controlling [2]. A web service is one example of a web application. It is the software system that enables interactions between machines over a network [3]. The main data format used for web services is called Extensible Markup Language (XML). The packaging mechanism specifies how to arrange XML messages during information exchanges between machines. The prevailing standard is the Simple Object Access Protocol (SOAP), which is commonly implemented on computer servers. As web services are expanding to consumer electronics appliances [4], embedded systems are the logical selection for these products to save cost [5]. The primary limitation of embedded systems is the resource constraints. Hence, a new package called Representational State Transfer (REST) that was originally developed by Fielding [6] is used to reduce the computation overhead.

SOAP and REST have their strengths and weaknesses. The comparison of both has been discussed at length by several papers, and hence will not be repeated here. In short, the SOAP-based web services are generally designed to provide more complex services, while the less formal REST-based web services are useful for low-resource systems. Several groups compared the performances of both packages on conventional web services for different applications such as electronic payment [7], mashup technology [8], interaction styles [9], multimedia conferencing [10], enterprise application integration [11], etc. The real test, however, is when both packages are implemented and compared on resource-constrained systems. While there are plentiful publications that report on the embedded

web services utilizing either SOAP or REST, there are only a small number of studies that attempted to implement both and compared their performances. Three groups made this comparison for mobile web devices [12–14], while another group for controlling an indoor actuator [15]. However, we have not seen any paper that provides the comparative performance of web services on reconfigurable systems such as FPGA boards. Such a comparison between SOAP and REST on these systems is important because of the following reasons. First, we would like to check the prevailing assumption that the latter is better than the former by performing apple-to-apple comparison. Such advantages have been reported on other platforms such as computer servers and mobile computing, but not on reconfigurable systems. Second, the comparison would be helpful for network engineers to know the versatility and superiority of REST-based web services, to the point that it should have its own standard. In order to answer this pertinent question, we report on the FPGA prototyping of an embedded web service that sends XML messages via SOAP and REST packages. The simple request and response messages are communicated through a 100 Mbps local area network between a Spartan-3E FPGA board and washing machine simulator. The latter is chosen because, due to IoT, future home appliances will be equipped with embedded web services for ubiquitous communications [16]. Section 2 provides the details of implementation, while Section 3 tabulates the comparative data. Finally, Section 4 offers concluding remarks on the pros and cons of REST- and SOAP-based web services.

2. Design and Implementation

This section details the implementation of the web services on the FPGA board. The overall setup will be explained in Section 2.1. Then, Section 2.2 will explain the differences between REST and SOAP server modules. Finally, the FPGA implementation of the web services is covered in Section 2.3.

2.1. Overall Setup

In order to emulate data communications between home appliances using a TCP/IP protocol suite, XML messages are sent between a server and client. A Spartan-3E board from Xilinx (San Jose, CA, USA) is configured as the web server because it has an on-board Media Independent Interface (MII) and Serial Peripheral Interface (SPI) [17]. A washing machine simulator from Bytronic International Ltd. (Rugeley, UK) [18] is chosen as the client (home appliance device). A Java program called Direct Socket Control is used to open a TCP connection as a web service client to communicate with the web server. This interface is used to control the home appliance. The FPGA board is operating at 50 MHz clock rate, and the local network speed is 100 Mbps.

2.2. REST and SOAP Servers

The designs of REST and SOAP server modules are shown in Figure 1. In Figure 1a, when a REST request is received at the HTTP level, the HTTP module will extract the "method" and "parameters" or URI from the HTTP header. The extracted information will be sent to the application program. The program will decide on how to respond to the client. If service descriptions are requested by the client, the server will load the XML formatted service description document. For other requests, the server wraps up the raw data to send and issues the service response to the client in XML format. The HTTP module will then attach HTTP response headers and send out the response or service descriptions to the client. Similar steps are executed for the SOAP-based server as shown in Figure 1b. The major difference is the existence of the "SOAP envelope". In order to extract the raw data from the HTTP module, a module called "Simple SAX parsing" is needed in the SOAP server to unwrap the raw data for processing in the application program. The procedure for sending the response message back to the client is the same as for the REST server.

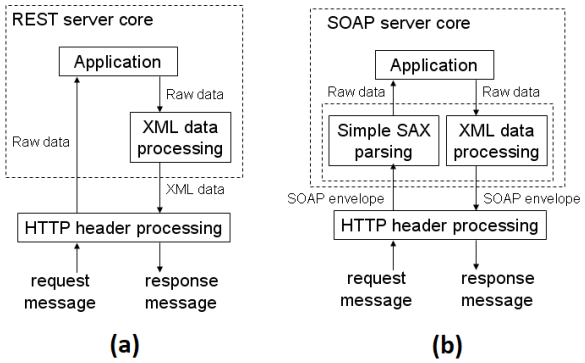

Figure 1. (**a**): REST server; (**b**): SOAP server.

2.3. FPGA Implementation of Embedded RESTful Web Services

The FPGA prototype is written and synthesized using Very High Speed Integrated Circuit Hardware Design Language (VHDL). It covers communications from the data link layer to application layer in the TCP/IP protocol suite. This embedded web server has three external interfaces. The MII port connects the server to the physical layer (100 Mbps fast Ethernet LAN). The device interface port controls the home appliance (washing machine simulator). Lastly, an SPI port is used to store the non-volatile information such as the application information, service descriptions, and server configuration information.

The web server consists of two modules, the TCP/IP processor and the application. The implementation for both had been detailed in [19], and hence will not be repeated here. Nevertheless, the summary is provided herein. The TCP/IP processor resembles a reduced network processor and covers all basic protocols including error checking. This web server uses a static IC address and is able to send and receive a TCP segment. The size of the segment is limited to only a single packet or 576 bytes of data. Upon receiving the TCP segment, the application module checks for the data validity as a HTTP request. It then extracts the REST web service information from the HTTP headers and verifies the service request. The response will be sent back to the client in XML format through the XML Writer module. Figure 2 shows the examples of request and response messages that are produced by XML Writer for both packages.

```
(SOAP request)

POST / HTTP/1.1
Host: 10.102.56.205

<?xml version="1.0"?>
<soap:Envelope
  xmlns:soap=http://www.w3.org/2001/12/soap-envelope>
  <soap:Body>
    <retrieve>
      hardware
    </retrieve>
  </soap:Body>
</soap:Envelope>


(SOAP response)

HTTP/1.1 200 OK
Content-Type: text/xml

<?xml version="1.0"?>
<soap:Envelope
  xmlns:soap=http://www.w3.org/2001/12/soap-envelope>
  <soap:Body>
    <RET>
      Washing_Machine
    </RET>
  </soap:Body>
</soap:Envelope>
```

Figure 2. *Cont.*

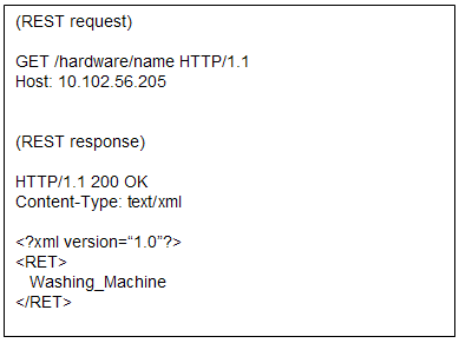

Figure 2. Examples of SOAP and RESTful messages.

3. Results

The REST and SOAP web service implementations are designed with identical specifications except for the packaging mechanism. The comparisons of different parameters are presented herein in terms of resource usage, message length, and processing time.

3.1. Resource Usage

Table 1 shows the overall resource usages of both packages on the application layer of the FPGA board, while Table 2 lists the details of each module. The data indicate that the REST-based web services utilize fewer resources than the SOAP counterpart. The differences are caused by the additional stage used to parse the SOAP envelope. This additional module is the SAX parser to analyze the SOAP service requests, and then generate a response to the SOAP envelope [20]. In total, an average reduction of 12% in resource usage for Flip Flop (FF), Look-Up Table (LUT), and Slice is achieved by the SOAP server.

Table 1. Resource usage of application layer.

Packaging Mechanism	FF	4-LUT	Slice
REST	901	2987	1606
SOAP	951	3531	1901
Reduction using REST (in % with reference to SOAP)	5.3	15.4	15.5

Table 2. Resource usage of each module.

Module	REST		SOAP	
	FF	MUX	FF	MUX
1. Application	330	25	343	25
2. SAX	(n/a)	(n/a)	21	-
3. XML Writer	79	-	79	-
4. HTTP	277	36	267	20

3.2. Message Length

Table 3 shows the message length of several operations for SOAP- and REST-based web services. Four types of messages are sent, namely retrieve device name, retrieve available function, perform a function, and erroneous request. The REST messages range from 46 to 202 characters, while SOAP messages are quite uniform, from 178 to 294 characters. Both are well below a single TCP segment's payload size of 576 bytes. The differences in the REST and SOAP web service message sizes were primarily induced by overhead of the SOAP envelope format. The bar chart in Figure 3 shows the comparison between different types of messages. The message size reductions between REST and SOAP are about 60~75%

for service request messages, and 30~50% for service response messages. Furthermore, it is noticeable that the differences between the REST and SOAP message length remain rather static, that is, about 90–140 characters or bytes even when message size grows.

Table 3. Message length (characters) of different services.

Operation	Request (Req.)/ Response (Resp.)	Message Length (Character)		Difference in Character	Reduction Using REST (in % with Reference to SOAP)
		REST	SOAP		
Retrieve device name	Req.	46	189	143	75.7
	Resp.	97	189	92	48.7
Retrieve available functions	Req.	56	189	133	70.4
	Resp.	202	294	92	31.3
Perform a function (spin)	Req.	68	183	115	62.8
	Resp.	86	178	92	51.7
Erroneous request (missing 1 char.)	Req.	67	182	115	63.2
	Resp.	90	182	92	50.5

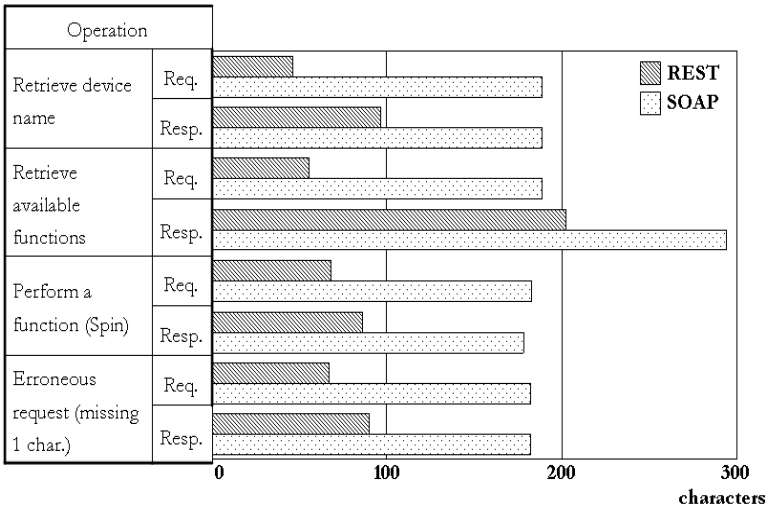

Figure 3. Bar chart of message length comparisons.

3.3. Processing Time

Table 4 and Figure 4 show the difference in processing time between SOAP and REST web services. This parameter records the total time it takes to handle a service request in the HTTP module. In particular, an internal timer is started when a request reaches the HTTP module, and is stopped once a response is sent out by this module. It is clear from the data that SOAP requests take 4 to 5 μs longer to process than REST requests. Furthermore, the processing time reduction can be as large as 50% when the server handles an erroneous request, i.e., when the service request was rejected without invoking the top level application. The longer delay is caused by two factors, namely the SOAP parser module and longer message length. On average, REST is faster than SOAP by 4.5 μs.

Table 4. Processing time.

Operation	Processing Time (μs)		Difference (μs)	Reduction (% with Reference to SOAP)
	REST	SOAP		
Retrieve device name	26	32	4	12.5
Retrieve available functions	34	39	5	12.8
Perform a function (spin)	33	38	5	13.2
Erroneous request (missing 1 char.)	4	8	4	50.0

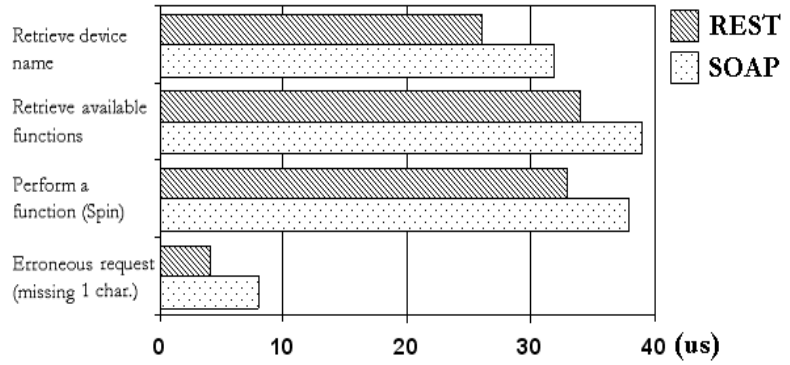

Figure 4. Bar chart of performance comparisons.

4. Conclusions

This study implements embedded web services on Xilinx's Spartan-3E Starter Kit FPGA board to communicate with a home appliance over a 100 Mbps home network. The performances of REST-based and SOAP-based web services are then compared. To the best of our knowledge, this is the first study that compares the performance of both servers that are implemented on an FPGA board. In general, the REST-based web server performs better than the SOAP-based web server in terms of FPGA resource utilization (~12% less), message length (~57% shorter), and processing time (4.5 μs faster). Table 5 summarizes the qualitative comparisons. This study supports the findings of previously published works that compare the performances of both packages on other embedded devices such as mobile web services [12–14] and actuators [15].

Table 5. Overall comparisons between REST and SOAP.

	REST	SOAP
Processing complexity	Lower, due to low service abstraction and format verbosity.	Higher.
Memory requirement	Lower, due to short message length.	Higher.
Performance	Faster, due to low format and message overhead.	Slower.
Service abstraction	Lower.	Higher, thus able to provide advanced services.
Standards	None, thus more freedom but hard for long-term management.	Yes, thus more systematic for long-term management.
Resources and supports	Less.	A lot, thus easier to develop.

We are linking the superiority of REST over SOAP and their relationships with the use of the FPGA board as follows. The data indicate that low abstraction services and simple message format for REST reduce the FPGA processing power requirement. The short message length reduces the FPGA memory requirement, and low processing complexity

and short message length contribute to faster FPGA performance. There is no surprise in the results, which is good news for appliance manufacturers and network engineers that consider reconfigurable computing as a low-cost solution for data transmission of multiple devices in the IoT. It should be noted, however, that this is a simple comparative study, where we only send two simple XML messages. Follow-up studies are necessary with complex messages.

Author Contributions: Conceptualization, F.M.-Y.; methodology, C.E.C. and F.M.-Y.; validation, C.E.C.; formal analysis, C.E.C. and F.M.-Y.; data curation, C.E.C. and F.M.-Y.; writing—original draft preparation, C.E.C. and F.M.-Y.; writing—review and editing, F.M.-Y.; supervision, F.M.-Y. and A.K.M.; project administration, F.M.-Y. and A.K.M.; funding acquisition, F.M.-Y. and A.K.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Panasonic Corporation (formerly Matsushita Electric Industrial Co. Japan), grant number EP20070522001.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data that support the findings of this study are available from the authors upon reasonable request.

Acknowledgments: The authors would like to express our gratitude to Multimedia University and Panasonic Corporation for providing the postgraduate scholarships to Chang Chee Er and for funding this research project.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Li, S.; Xu, L.D.; Zhao, S. The internet of things: A survey. *Inf. Syst. Front.* **2015**, *17*, 243–259. [CrossRef]
2. Ooi, C.P.; Tan, W.H.; Cheong, S.N.; Lee, Y.L.; Baskaran, V.M.; Low, Y.L. FPGA-based embedded architecture for IoT home automation application. *Indones. J. Electr. Eng. Comput. Sci.* **2019**, *14*, 646–652. [CrossRef]
3. Gottschalk, K.; Graham, S.; Kreger, H.; Snell, J. Introduction to Web services architecture. *IBM Syst. J.* **2002**, *41*, 170–177. [CrossRef]
4. Brzoza-Woch, R.; Nawrocki, P. FPGA-Based Web Services—Infinite Potential or a Road to Nowhere? *IEEE Internet Comput.* **2016**, *20*, 44–51. [CrossRef]
5. Shelby, Z. Embedded web services. *IEEE Wirel. Commun.* **2010**, *17*, 52–57. [CrossRef]
6. Fielding, R.T. Architectural Styles and the Design of Network-Based Software Architectures. Ph.D. Thesis, University of California, Irvine, CA, USA, 2000.
7. Markonnen, J. Performance and Usage Comparison between REST and SOAP Web Services. Master's Thesis, Aalto University, Espoo, Finland, 2019.
8. Su, H.; Cheng, B.; Wu, T.; Li, X. Mashup service release based on SOAP and REST. In Proceedings of the 2011 International Conference on Computer Science and Network Technology, Harbin, China, 24–26 December 2011; IEEE: Manhattan, NY, USA, 2012; Volume 2, pp. 1091–1095.
9. Potti, P.K.; Ahuja, S.; Umapathy, K.; Prodanoff, Z. Comparing performance of web service interaction styles: Soap vs. Rest. In Proceedings of the Conference on Information Systems Applied Research, New Orleans, LA, USA, 1–18 November 2012; Volume 2167, p. 1508.
10. Belqasmi, F.; Singh, J.; Melhem, S.Y.B.; Glitho, R.H. SOAP-based vs. RESTful web services: A case study for multimedia conferencing. *IEEE Internet Comput.* **2012**, *16*, 54–63. [CrossRef]
11. Kumari, S.; Rath, S.K. Performance comparison of soap and rest based web services for enterprise application integration. In Proceedings of the 2015 International Conference on Advances in Computing, Communications and Informatics (ICACCI), Kochi, India, 10–13 August 2015; IEEE: Manhattan, NY, USA, 2015; pp. 1656–1660.
12. AlShahwan, F.; Moessner, K. Providing soap web services and restful web services from mobile hosts. In Proceedings of the 2010 Fifth International Conference on Internet and Web Applications and Services, Barcelona, Spain, 9–15 May 2010; IEEE: Manhattan, NY, USA, 2010; pp. 174–179.
13. Mohamed, K.; Wijesekera, D. Performance analysis of web services on mobile devices. *Procedia Comput. Sci.* **2012**, *10*, 744–751. [CrossRef]
14. Ali, M.; Zolkipli, M.F.; Zain, J.M.; Anwar, S. Mobile cloud computing with SOAP and REST web services. *J. Phys. Conf. Ser.* **2018**, *1018*, 012005. [CrossRef]

15. Malik, S.; Kim, D.H. A comparison of RESTful vs. SOAP web services in actuator networks. In Proceedings of the 2017 Ninth International Conference on Ubiquitous and Future Networks (ICUFN), Milan, Italy, 4–7 July 2017; IEEE: Manhattan, NY, USA, 2017; pp. 753–755.
16. Riihijarvi, J.; Mahonen, P.; Saaranen, M.J.; Roivainen, J.; Soininen, J.P. Providing network connectivity for small appliances: A functionally minimized embedded web server. *IEEE Commun. Mag.* **2001**, *39*, 74–79. [CrossRef]
17. Xilinx. *Spartan-3E FPGA Family Data Sheet*; Xilinx: San Jose, CA, USA, 2018.
18. Bytronic. Washing Machine Simulator. Bromsgrove, UK. Available online: http://www.bytronic.net/product/washing-machine-simulator (accessed on 3 September 2022).
19. Chang, C.E.; Mohd-Yasin, F.; Mustapha, A.K. An implementation of embedded restful web services. In Proceedings of the 2009 Innovative Technologies in Intelligent Systems and Industrial Applications, Kuala Lumpur, Malaysia, 25–26 July 2009; IEEE: Manhattan, NY, USA, 2009; pp. 45–50.
20. Chang, C.E.; Mohd-Yasin, F.; Mustapha, A.K. RBStreX: Hardware XML parser for embedded system. In Proceedings of the 2009 International Conference for Internet Technology and Secured Transactions (ICITST), London, UK, 9–12 November 2009; IEEE: Manhattan, NY, USA, 2010; pp. 1–6.