Performance Evaluation of Web Services In Linux On Multicore

P. Bala Subramanyam Raju *bsr3011@gmail.com*
Research Scholar,
S.V University,
Tirupathi Chittoor (Dt) AP, India

P. Govindarajulu *pgovindarajulu@yahoo.com*
Professor, Dept. of Computer Science,
S.V University,
Tirupathi Chittoor (Dt) AP, India

Abstract

Contemporary Business requires the ability to seamlessly exchange information between internal business units, customers, and partner, is vital for success. Most organizations employ a variety of different applications to store and exchange data in dissimilar way and therefore cannot "communicate" to one another productively [1]. Service Oriented Architecture (SOA) components provide services to other components via communication protocols typically over a network [2].The technologies like DCOM, RMI, COBRA, Web Services etc. are developed using SOA, which contributed best to fulfill requirements to some extent, but components result from these technologies are mostly either language specific or platform specific,[3]. The services or components developed for one platform may not be able to communicate and reusable in other platform, as they are mostly language specific or platform specific.

"World Wide Web Consortium (W3C) International community to develop web standards" issued WS-* specifications for programming language vendors for Web services, which confirms a standard means of interoperating between different software applications running on a variety of platforms or frameworks [4][5]. This paper tests web services performance gain along with interoperability, reusability by using "NAS Parallel Benchmarks (NPB)" set of program [6] developed by NASA Advanced Supercomputing Division to evaluate the performance of supercomputers.

Keywords: SOA, WSDL, COBRA, RMI, Stub, Skeleton, SOAP Protocol, HTTP Protocol.

1. INTRODUCTION

Service-Oriented Architecture (SOA) is an approach to create and expose the business functions to the outer world via communication protocols [7]. One of the key aspects of SOA is that, interactions occur with loosely coupled services that operate independently.SOA Architecture and its elements are shown in figures1.1, figure 1.2. It provides service reuse and build applications so quickly, this benefits business to reduce time to market (TTM) and save money [8].

The SOA [9] can operate independently of specific technologies and can be implemented using a wide range of technologies including: RPC, REST, DCOM, CORBA OPC-UA, Java RMI, Web Services, and WCF Services, JAX-WS 2.2[10]. WCF Services [11] [12] and JAX-WS2.2 are web services that confirms to WS-*specifications.

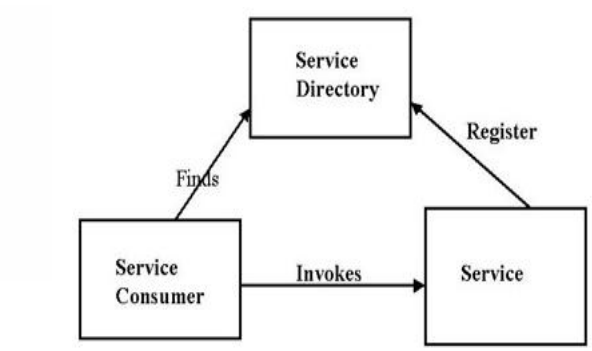

FIGURE 1.1: Shows the Architecture of SOA.

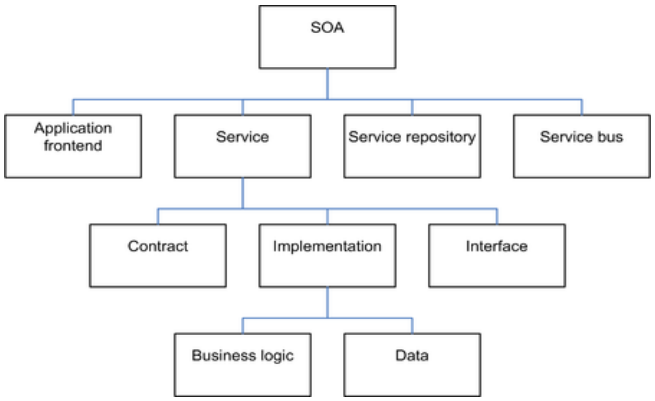

FIGURE 1.2: Shows the Elements of SOA.

The rest of the paper is organized as follows; Section 2 provides brief introduction about WCF Services and JAX-WS2.2, related research work outlined in section 3. Proposed problem statement, Algorithm of NPB application are discussed in section 4. Section 5 specifies hardware and software used. Implementation details are discussed in section 6, Execution Results are shown in section 7. Conclusion and future work is presented in section 8.

2. WCF and JAX-WS

Windows Communication Foundation (WCF) is a framework for building service oriented applications [13], where we can send data asynchronously from one service endpoint to other. WCF Services exposes business process as collection of endpoints. An endpoint consist of three properties [14]

1. **Address**- indicates where the service can be found
2. **Binding** -specifies how a client can communicate
3. **Contracts**- identify the operations available

A service endpoint can be part of continuously available service hosted by IIS or service hosted in an application. An end point can be a client service requests data from a service endpoint [15]. WCF supports Industry standard formats such as WSDL, XML Schema and WS-Policy Service Metadata used to configure clients. Major Features of WCF are listed

1. Service orientation,
2. Interoperability,
3. Security,
4. Multiple Transports and Encodings,
5. AJAX and REST support,
6. Customization.

WCF Services can be run like normal executables, can be hosted by an external agent, such as IIS or Windows Activation Service (WAS). WAS enables WCF applications to be activated automatically as Web services or manually as executables. The Architecture of WCF Services and communication process is shown in the below figures 2.1and 2.2

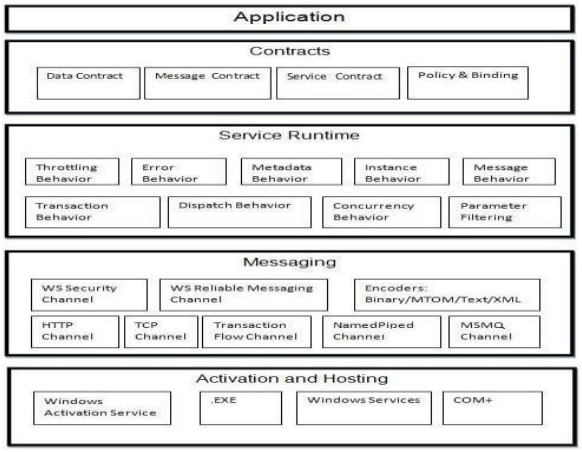

FIGURE 2.1: Shows the Architecture of WCF.

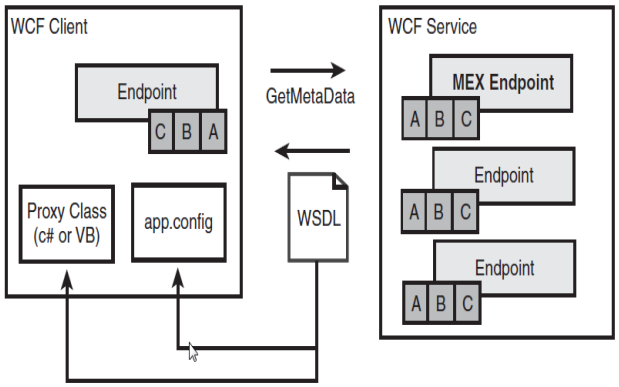

FIGURE 2.2: Shows the Communication Process between Service and Client Using Endpoint.

Java API for XML Web services (JAX-WS) is a technology for building web services and clients that communicate using XML.JAX-WS allows developers to write message-oriented as well as Remote Procedure Call9RPC) web services [16]. JAX-WS hides the complexity of SOAP messages structure, encoding, conventions for representing service invocations and responses. On the server side, the developer specifies the web service operations by defining methods in an interface written in the java programming language [17] [18]. The developer codes one or more classes that implement those methods. Client programs are also easy to code.

JAX-WS uses technologies defined by the W3C technologies HTTP, SOAP, WSDL which specifies an XML format for describing a service as a set of endpoints operating on messages; this is a big advantage of platform independence.

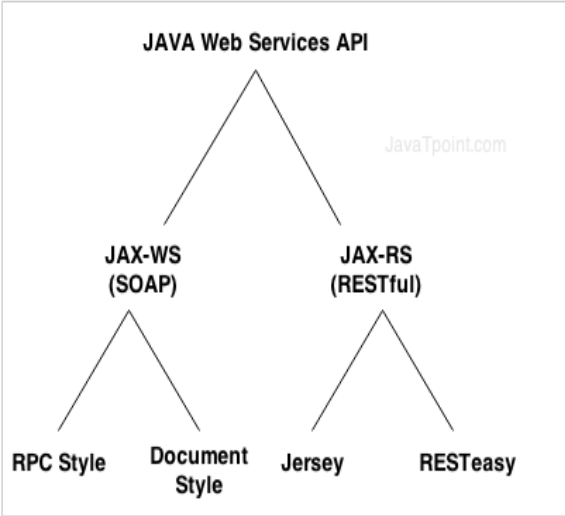

FIGURE 2.3: Shows the Types of Web Services Supported by java.service Architecture.

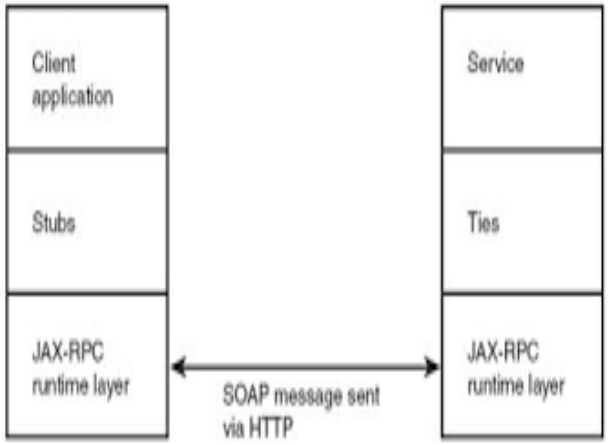

FIGURE 2.4: Shows the Communication Pattern of JAX-WS.

3. RELATED WORK

The Authors Markus Stopper and Bernd Gastermann have introduced the WCF's various Message exchange patterns, and hosting process. They have implemented WCF services for industrial application in production environment with a central manufacturing information system (MIS).This application steadily delivers important information about the current manufacturing process in real-time. They have designed the software to give indication of whether the production order's deadline can be reached or not [19].

Wei.Zhang and Jing.Li tested the extensibility of WCF i.e. the ability to change the product more easily over time and flexibility to customize for specific requirements of applications. In this research they used web-MIS application. Web-MIS applicaton takes advantage of the extensibility to integrate DI (Dependency Injection) and AOP (Aspect-Oriented Programming). Using WCF we can achieve higher softwarequality [20].

The Authors Tu Nguyen Thi Thanh, Thang Huynh Quyetresearch attempts to introduce the development of OPC UA SDK based WCF Technology, which is used for monitoring and controlling systems. They have successfully developed OPC UA SDK for an environmental monitoring application (EMA).This SDK in turn makes system architects and developers easy to design and implement applications in terms of environmental monitoring. In addition, this also reduces the development time and cost for such applications [21].

MdTanvir Ahmad, Prof. M. AfsarAlam, Shah Imran Alamhave introduced various SOA implementations like contract First WCF with XML serialization, Data Contract serialization and restful service. It has been witnessed from performance analysis that, by designing/implementing SOA using Contract-First Approach, with Data Contract Sterilization saves time and money. With this approach WS* security can be achieved properly.

It has also been observed that SOA designing/implementation with WCF RESTful Service is good in integration with latest GUI and concludes that, SOA capability can be extended to large scale by making it accessible to an individual and organization level. Significance performance gain with WCF RESTful Service. The analysis of SOAP WCF Service and WCF RESTful Service clearly shown that the RESTful WCF returns the response much faster than the WCF WSDL Service, almost 10 times faster. Also the response object size for RESTful WCF Service is smaller than WCF WSDL Service [22].

4. PROBLEM STATEMENT AND ALGORITHM OF NPB APPLICATION

Most of research work on SOA is concentrated on extensibility and performance gain. Less motivation towards interoperability. The management of resources to increase capacity (serving number of clients) and performance is a major constraint because the service demands might vary in peak timings. Therefore more efficient and cost-effective solution is needed in implementing SOA services.

The goal of this work is to provide the more performance with the same resources (hardware) along with testing Cross platform interoperability, language interoperability, and reusability. We have used Nasa Parallel Benchmark (NPB) application for this purpose. Below is the abstract and untouched algorithm of NPB application.

Algorithm for Main method

Step: 1 Start
Step: 2 Read the operation, class from arguments
Step: 3 if operation equal to IS
    Call method ProcessIS with class argument
Step: 4 Else if operation equal to CG
    Call method ProcessCG with class argument
Step: 5 Else if operation equal to MG
    Call method ProcessMG with class argument

Step: 6 Else if operation equal to FT
Call method ProcessFT with class argument
Step: 7 Stop

Algorithm for ProcessIS

Step: 1 Start
Step: 2 Read the class parameter
Step: 3 perform Integer Sort operation based on the class argument
Step: 4 output the result
Step: 5 Stop

Algorithm for ProcessCG

Step: 1 Start
Step: 2 Read the class parameter
Step: 3 perform Conjugate Gradient operation based on the class argument
Step: 4 output the result
Step: 5 Stop

Algorithm for ProcessMG

Step: 1 Start
Step: 2 Read the class parameter
Step: 3 perform Multi-Grid operation based on the class argument
Step: 4 output the result
Step: 5 Stop

Algorithm for ProcessFT

Step: 1 Start
Step: 2 Read the class parameter
Step: 3 perform discrete 3D fast Fourier Transform operation based on the class argument
Step: 4 output the result
Step: 5 Stop

5. EXPERIMENTAL TESTBED

Server Configuration		Client Configuration	
Processor	: Intel® Core™ i7-4770k[23]	**Processor**	: Intel® Core™ i3-
No of Cores	: 4	**No of Cores**	: 2
No of Threads	: 8	**No of Threads**	: 4
Base Frequency	: 3.5Ghz	**Base Frequency**	: 2.4Ghz
Turbo Frequency	: 3.9Ghz	**Turbo Frequency**	: No
Intel® Smart Cache	: 8MB	**Intel® Smart Cache**	: 3 MB
RAM	: 8GB/1600MHz	**RAM**	: 4GB/1333MHz
LINUX KERNEL: 4.5[25]			
Operating System: Linux Mint 17.3-Mate 64 Bit [26]		**Operating System:** Windows 10 64 bit	

Web Server : Glass Fish Server 4.1.1	**Web Server** : IIS
Java JDK: 1.7	**C#.NET** : 6.0
Java EE : 7.0	**ASP.NET** : 4.6
Net beans IDE: 8.0.2	**Visual Studio IDE:** 2015

6. IMPLEMENTATION DETAILS

In order to prove advanced web services provides not only programming language interoperability but also platform, we created server service which executes the NPB operations in Linux OS using Java JAX-WS. Client services are created in Linux Java JAX-WS as desktop application, another client service in windows system using .NET WCF Services as web application. Both the clients are created to call Server service by passing operation, class as arguments. The clients, server are hosted in the LAN network to have communication between them.

Initially executed NPB standalone application using NetBeans IDE and results have been tabulated in table 6.The Linux hosted JAX-WS as server service which in turn executes the NPB when called by client service. We have made few modifications to NPB source so that it will be suitable in the JAX-WS Service.

The client services .NET WCF Service, and Java service requested the server service. The execution results are sent as response from server and the results are tabulated in table 6. Figure 6 showing the experiment architecture. And figures 6.1, 6.2, 6.3, 6.4 shows the few execution screens.

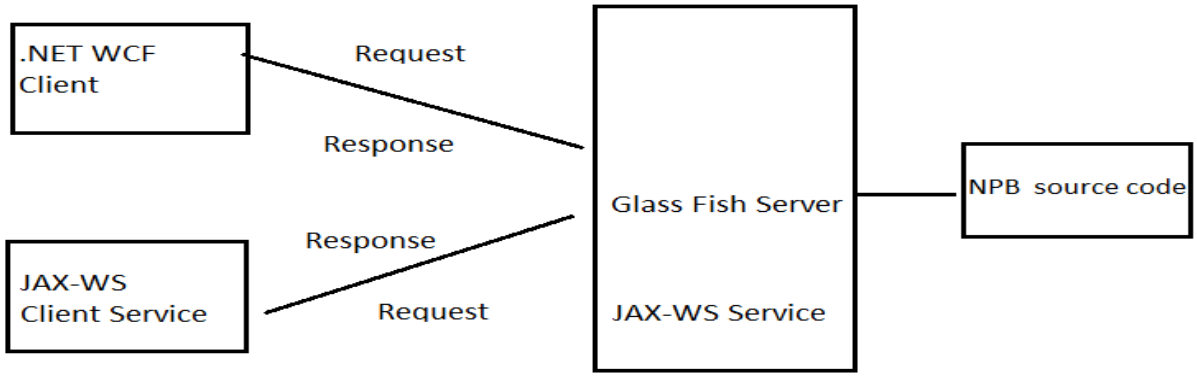

FIGURE 6: Shows the Experimental Architecture .

Improved Abstract algorithm of NPB application to fit in client server environment.

Algorithm Web Server Service Class

Step: 1 Start
Step: 2 wait for client request
Step: 3 if request is received
Call processrequest method with operation, class values
Step: 4 receive the result from process request method
Step: 5 return the result to the client as response
Step: 6 return to step 2
Step: 7 stop

Algorithm Web Client Service Class

Step: 1 Start
Step: 2 Read the operation and class values
Step: 3 call Web server service in the network Along with operation, class values
Step: 4 receive the result as response
Step: 5 output the result
Step: 6 stop

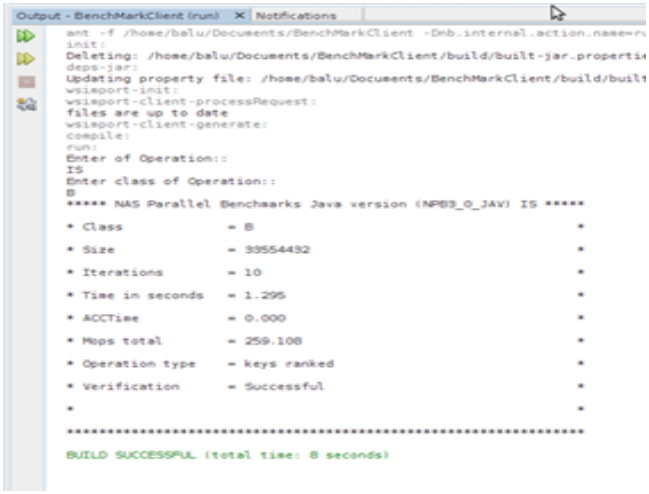

FIGURE 6.1: Shows Exec of IS Operation with Class B.

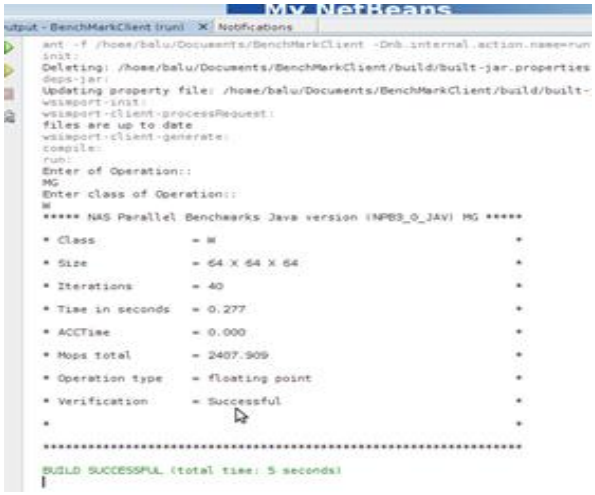

FIGURE 6.2: Shows Execof MG Operation with Class W.

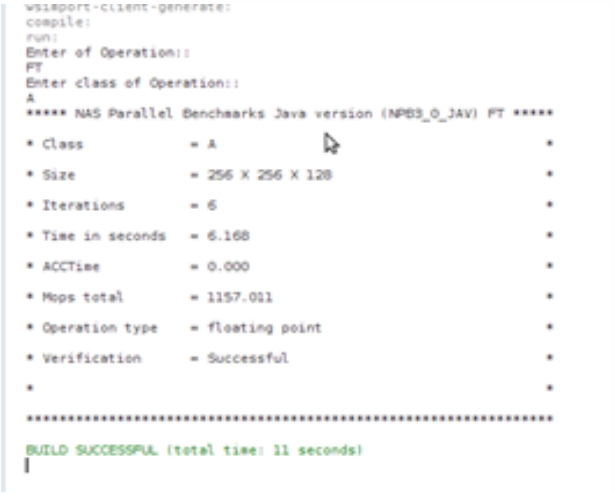

FIGURE 6.3: Shows ExecofFT Operation with Class A.

FIGURE 6.4: Shows Execof CG Operation with Class.

7. NORMAL EXECUTION AND WEB SERVICE EXECUTION

	Normal Execution					Web Services Execution			
IS	**Class S**	**Class W**	**Class A**	**Class B**		**Class S**	**Class W**	**Class A**	**Class B**
Size	65536	1048576	8388608	33554432		65536	1048576	8388608	33554432
Time in Sec	0.007	0.022	0.260	1.109		0.006	0.016	0.254	1.089
MOPS total	93.637	476.630	322.639	306.433		109.243	655.366	330.261	308.122
Iterations	10	10	10	10		10	10	10	10
CG	**Class S**	**Class W**	**Class A**	**Class B**		**Class S**	**Class W**	**Class A**	**Class B**
Size	1400	7000	14000	75000		1400	7000	14000	75000
Time in Sec	0.042	0.295	1.071	85.502		0.042	0.277	1.034	84.209
MOPS total	1587.0	1430.714	1397.255	639.853		1627.0	1518.520	1447.253	650.853
Iterations	15	15	15	75		15	15	15	75
MG	**Class S**	**Class W**	**Class A**	**Class B**		**Class S**	**Class W**	**Class A**	**Class B**
Size	32*32*32	64*64*64	256*256*256	256*256*256		32*32*32	64*64*64	256*256*256	256*256*256
Time in Sec	0.027	0.272	1.595	7.603		0.027	0.232	1.432	6.903
MOPS total	337.723	2452.172	2493.276	2620.186		350.723	2874.960	2546.673	2685.190
Iterations	4	40	4	20		4	40	4	20
FT	**Class S**	**Class W**	**Class A**	**Class B**		**Class S**	**Class W**	**Class A**	**Class B**
Size	64*64*64	128*128*128	256*256*128	512*256*256		64*64*64	128*128*128	256*256*128	512*256*256
Time in Sec	0.166	0.452	6.004	84.988		0.160	0.315	5.791	83.988
MOPS total	1067.271	824.497	1188.615	1083.136		1107.293	1183	1232.334	1153.136
Iterations	4	6	6	20		4	6	6	20

8. CONCLUSION AND FUTURE WORK

The above results shows that Web services provides better performance for legacy source code, which can be easily embedded with few modifications. This integration of legacy source code saves lot of development effort, time and cost for the organization by reusing the existing development effort. It is also proved that Web services provides not only language interoperability, but also platform interoperability by utilizing the JAX-WS service hosted in Linux in .NET WCF service.

We have conducted this experiment NPB application for only S, W, A, B Classes due to lack of server hardware. It can be further extended to the classes C, D, E, and F which requires huge hardware resources.

9. REFERENCES

[1] Internet: http://www.altova.com/whitepapers/webservices.pdf [Jul, 24, 2016].

[2] Internet: https://en.wikipedia.org/wiki/Service-oriented_architecture[Jul, 24, 2016].

[3] Internet: http://www.careerride.com/Net-Web-Service-Need.aspx[Jul, 24, 2016].

[4] Internet: http://www.dotnet-tricks.com/Tutorial/webservice/ccHW280913-Understanding-WS-star-standards-and-specifications.html [Jul,9,2016].

[5] Internet: https://www.w3.org/2002/ws/Activity [Jul, 24, 2016].

[6] Internet: http://www.nas.nasa.gov/publications/npb.html[Jul, 24, 2016].

[7] Internet: https://msdn.microsoft.com/en-in/library/bb833022.aspx[Jul, 24, 2016].

[8] Internet: http://searchsoa.techtarget.com/definition/service-oriented-architecture[Jul, 24, 2016].

[9] Internet: https://en.wikipedia.org/wiki/Service-oriented_architecture[Jul, 24, 2016].

[10] Internet: http://download.oracle.com/otn-pub/jcp/jaxws-2.2-mrel3-evalu-oth-JSpec/jaxws-2_2-mrel3-spec.pdf?AuthParam=1459229313_10c3b5dc52161b9d6472646637abce2b [Jul, 24, 2016].

[11] Internet: https://msdn.microsoft.com/en-us/library/ms733128 (v=vs.110).aspx [Jul, 24, 2016].

[12] Internet: https://msdn.microsoft.com/en-us/library/aa480210.aspx[Jul, 24, 2016].

[13] Internet: https://msdn.microsoft.com/en-us/library/ms731082(v=vs.110).aspx [Jul, 24, 2016].

[14] Internet: https://msdn.microsoft.com/en-us/library/ms733107(v=vs.110).aspx [Jul, 24, 2016].

[15] Internet: https://msdn.microsoft.com/en-us/library/ms731079(v=vs.110).aspx [Jul, 24, 2016].

[16] Internet: http://docs.oracle.com/javaee/6/tutorial/doc/bnayl.html [Jul, 24, 2016].

[17] Internet: http://www.mkyong.com/tutorials/jax-ws-tutorials/ [Jul, 24, 2016].

[18] Internet: http://www.ibm.com/developerworks/webservices/tutorials/ws-jax/ws-jax.html [Jul, 24, 2016].

[19] Markus Stopper and Bernd Gastermann “Service-oriented Communication Concept based on WCF.NET for Industrial Applications” International Multiconference of Engineers and computer Scientists 2010 Vol III IMECS 2010, March 17-19, 2010 Hong Kong.

[20] W. Zhang and J. Li, "Research and Application of WCF Extensibility", Web Information Systems and Mining (WISM), 2010 International Conference on, Sanya, 2010, pp. 363-367.doi: 10.1109/WISM.2010.112
URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5662927&isnumber=5662251.

[21] Tu Nguyen Thi Thanh, Thang Huynh Quyet,” Development of an OPC UA SDK Based WCF Technology and Its Deployment for Environmental Monitoring Applications”, ICCASA 2012, Ho Chi Minh City, Vietnam, November 26-27, 2012, pp 347-356
http://link.springer.com/chapter/10.1007%2F978-3-642-36642-0_34.

[22] Kola Siva Tharun, MuppallaPrudhvi, Satti Swami Reddy “ADVANTAGES OF WCF OVER WEB SERVICES” International Journal of Computer Science and Mobile Computing, IJCSMC, Vol. 2, Issue. 4, April 2013, pg.340 – 345 ISSN 2320–088X.

[23] http://ark.intel.com/products/75123/Intel-Core-i7-4770K-Processor-8M-Cache-up-to-3_90-GHz [Jul, 24, 2016].

[24] http://ark.intel.com/products/84699/Intel-Core-i3-5020U-Processor-3M-Cache-2_20-GHz [Jul, 24, 2016].

[25] https://www.kernel.org/ [Jul, 24, 2016].

[26] https://www.linuxmint.com/edition.php?id=206 [Jul, 24, 2016].