International Journal of Energy, Information and Communications
Vol.6, Issue 3 (2015), pp.19-28
http://dx.doi.org/10.14257/ijeic.2015.6.3.03

Web Services Oriented Architecture for DPI based Network Forensics Grid

Jyotsna Sharma and Maninder Singh

Computer Science & Engineering Department
Thapar University,Patiala, INDIA
{jyotsana.sharma,msingh}@thapar.edu

Abstract

The traditional security solutions to network security threats have several limitations. Deep Packet Inspection(DPI) based Forensic Analysis of network traffic allows indepth and unhurried analysis of anomalies which otherwise might go unnoticed. This highly resource intensive task can be performed with efficiency by using Grid Infrastructure. Service Oriented Architecture based on web services has been proposed in this paper, which can ensure long term scalability and extensibility of the application.

Keywords: *DPI, IDS, Forensic Analysis of Network Traffic, Network Forensics, Grid Computing, Web Services,WSRF*

1. Introduction

Traditional approaches to network security, like Anomaly Detection and Protocol Decoding, prove inadequate to detect and/or prevent the present day, highly sophisticated network attacks. An IDS based on Deep Packet Inspection (DPI) technology, where all the data within a packet payload is inspected and compared to a database of predefined attack signatures, is a better approach. Since real-time monitoring of the payload can be very expensive in terms of both the human and the hardware resources, reconstructive traffic analysis, or network forensics[1], where network traffic can be archived[2] and its subsets can be analyzed as necessary, to identify the threats and/or intrusions, is a more viable option. But, the computational requirements for such analysis are quite significant and employing special hardware to address the demand even for a medium size network is quite expensive. Efforts involving the harnessing of the capabilities of a grid for the forensic analysis have yielded results that can address these issues.

Grids help optimize the infrastructure to balance workloads and provide extra capacity for high-demand applications [3]. Grid Computing systems based on web services adds efficiency and simplifies the addition of standards-based core functionality to those systems [4]. The convergence of web services and grid computing simplifies the design, development and deployment of grid services in virtual organizations with diverse compute and resource characteristics. Recently, the researchers at Korea have introduced mobile cloud and grid web services to provide intelligent services in a smart city, which is a good indication of ubiquitous applications of the technology [5]. Environmental Researchers in the U.S. have introduced a grid gateway web service to enable science web portals to transparently access the access High Performance Computer (HPC) resources [6].

This paper discusses the approach for deploying a web services oriented architecture for the DPI based forensic analysis of Network Traffic using Grid infrastructure. The network traffic can be captured by the consumer and submitted to the grid on which DPI based forensic analysis is performed. The grid application for the forensic analysis can be deployed as a WSRF web service to yield a scalable loosely coupled solution which is interoperable, robust, and dependable too[7]. The paper also discusses the motivation for

ISSN: 2093-9655 IJEIC

selection of the WSRF implementation chosen after extensive research on the different available implementations.

2. DPI based Forensic Analysis of Network Traffic using Grid Infrastructure

2.1. DPI based Forensic Analysis of Network Traffic

Deep Packet Inspection (DPI) addresses the limitations of firewalls and traditional IDS techniques, which don't look beyond layer 3 of the OSI protocol stack. DPI operates at Layer 3-7 of the OSI model and looks at the payload of the packets and compare bytes within the payload to a known list of malicious bytes. The essence of such a signature based scanning is a multi-pattern matching algorithm [8], where the payload is inspected to detect malicious patterns such as worm code signatures [9]. Searching through the payload for multiple string patterns within the data-stream is a computationally expensive task because the payload contents are unconstrained as opposed to the economical analysis of the packet headers owing to the location of header fields being restricted by protocol standards. Fast string matching is the key element to DPI based IDSs and is a vital component for earlier attack detection.

The computational and resource intensive real-time deep packet inspection of all network traffic, would not scale to networks larger than a single workgroup. The practical solution is to perform reconstructive traffic analysis by capturing and archiving all traffic and analyzing subsets as necessary. Also known as network forensics, the method can identify the source and nature of the security attacks.

The Deep Packet Capturing(DPC) of complete data packets, crossing a network can be done by using tools such as TcpDump, Windump, Ettercap, dSniff or Wireshark (earlier known as Ethereal). Deep Packet Inspection is then performed for forensic analysis to uncover the root cause of network problems and identify the security threats. Once an intrusion has been detected, historical data may allow a system administrator to determine, conclusively, exactly how many systems were affected [10]. The packets indicating intrusion can be accurately rendered to their original format 11].

2.2. Grid Computing Infrastructure

Grid computing is a cost-effective and scalable means of solving large scale computing problems that spans not only locations but also organizations, machine architectures and software boundaries to provide unlimited power, collaboration and information access to everyone connected to a grid [12].

A grid is a collection of distributed computing resources available over a network that appear as one large virtual computing system,to an end user or application. Workloads can be provisioned and deployed by locating available pools of computing resources. The resources of many computers in a grid can be applied to a single problem at a time [13]. The most common resource is computing cycles provided by the processors of the machines on the grid, which specifically is referred to as a Computational Grid. Another type of grid, the Data grid/Information Grid, utilizes memory and/or secondary storage in the grid to increase capacity, performance, sharing, and reliability of data while providing its access across multiple organizations [14].

Grids help optimize the infrastructure to balance workloads and provide extra capacity for high-demand applications [3]. By providing a quick solution to usage changes, grid computing provides businesses with extra capability when needed, without consuming large amounts of energy [15]. Using servers to capacity cuts down on the energy needed. Grid computing also enables research-oriented organizations to solve problems that were infeasible to solve due to computing and data-integration constraints. It enables sharing of expensive scientific equipment too.

2.3. DPI based Forensic Analysis using Grid Infrastructure

Currently, networks are quite fast, generating traffic at speeds greater than 1 Gbit/sec. Capturing packets reliably from a high speed network requires an high speed interface that doesn't lose packets as the network approaches saturation. The amount of storage dedicated to traffic capture determines how far into the past we have data for forensic analysis. The computational and storage requirements for the inspection and analysis of such traffic are very high. A specialised or dedicated computational infrastructure can be very expensive in terms of cost and efficiency.

A better cost effective solution to this problem is using the capabilities of a grid to detect enormous intrusion packets and improve the drawbacks of traditional IDSs. Grid Infrastructure is chosen for performing the DPI based Forensic Analysis because Grids can provide high computational abilities along-with the features of distributed resource usage and communication. The scalability feature of grids is also a compelling reason to choose the grid infrastructure for the problem as the network traffic may peak to very high levels at times. Another motivating reason is that Grid Computing supports reduction in energy consumption by utilizing servers to capacity [16].

3. Service Oriented Architecture for DPI based Network Forensics Grid

3.1. Web Services

Web Services is a software system that support interoperable machine-to-machine interaction over a network. Web services interface is described in a machine processable format. Web services provide flexible, extensible, and widely adopted XML-based mechanisms for describing, discovering, and invoking network services; in addition, its document-oriented protocols are well suited to the loosely coupled interactions that many argue are preferable for robust distributed systems [17]. A Web Service is callable by another program across the Web in a platform/language/object model neutral fashion (using standardized Web Services protocols). In short, a Web service is to an application what a Web page is to a person. The Web Services protocols include Simple Object Access Protocol (SOAP); Universal Description, Discovery, and Integration (UDDI); and Web Services Description Language (WSDL). Web services have emerged as the architecture of choice for grid standards such as the Web Services Resource Framework (WSRF) [18].

Web services publish details of their functions and interfaces, but they keep their implementation details private; thus a client and a service that support common communication protocols can interact regardless of the platforms on which they run, or the programming languages in which they are written. This makes Web services particularly applicable to a distributed heterogeneous environment. The key specifications used by Web services are:

• XML (eXtensible Markup Language)—a markup language for formatting and exchanging structured data.

• SOAP (originally Simple Object Access Protocol, but technically no longer an acronym)—an XML-based protocol for specifying envelope information, contents and processing information for a message.

• WSDL (Web Services Description Language)—an XML-based language used to describe the attributes, interfaces and other properties of a Web service. A WSDL document can be read by a potential client to learn about the service.

Although a Web service can support any communication protocol, and may offer its clients a choice, the most common is SOAP over either HTTP or HTTPS. This contributes to the appeal of Web services, as HTTP and HTTPS are ubiquitous and typically do not raise issues of firewall traversal in an organization that allows bi-directional HTTP traffic.

3.2. OGSA, OGSI, WSRF and GT4

The Open Grid Services Architecture (OGSA) represents an evolution towards a Grid system architecture based on Web services concepts and technologies. The goal of the OGSA is to standardize all the services of the grid application by specifying a set of standard interfaces for them.The Globus Toolkit(GT) is an open source software toolkit used for building grid-based applications, being developed by the Globus Alliance and many others all over the world. The first prototype Grid service implementation was demonstrated on January 29, 2002, at a Globus Toolkit tutorial held at Argonne National Laboratory. Since then, the Globus Toolkit 3.0 and 3.2 offered an OGSA implementation based on the Open Grid Services Infrastructure (OGSI), a precursor to WSRF [19]. The Grid Community invested a lot of effort into the specification and implementation of OGSI but soon realized that it was a heavyweight specification with too much definition in one specification; and was too much object-oriented and also stateful. Globus Toolkit 3(GT3) is an implementation of the OGSI, but since it failed to converge with existing web services standards, a new standard to supersede it, was announced in 2004 and fully implemented in the Globus Toolkit version 4(GT4). GT4 is the widely used, mature web services based toolkit for building grid applications [20]. It provides a set of OGSA capabilities based on the Web Services Resource Framework(WSRF) which specifies the creation, addressing, inspection, and lifetime management of stateful resources. Grids and web services started far apart in technology and applications, but WSRF completed the convergence (Figure 1).

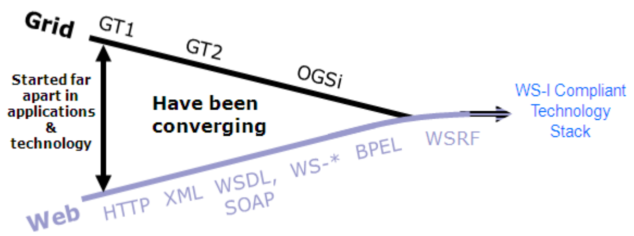

Figure 1. Convergence of Grid Computing and Web Services
Source: www.grid.org.tr/etkinlikler/egitim/sunumlar/WhatIsGridComputing.ppt

Web services are fundamentally stateless, but the resources on a grid are not. WSRF, a joint effort between the Grid and Web Services communities, defines a way to associate a state with a web service, called resource properties. WSRF is the infrastructure on which the OGSA architecture is built on. The relationship between OGSA,GT4 and WSRF has been clearly indicated by Globus (Figure 2). The composition of a stateful resource and the web service through which it can accessed is called a *WS-Resource.* The GT4 contains Java and C implementations of the WSRF.

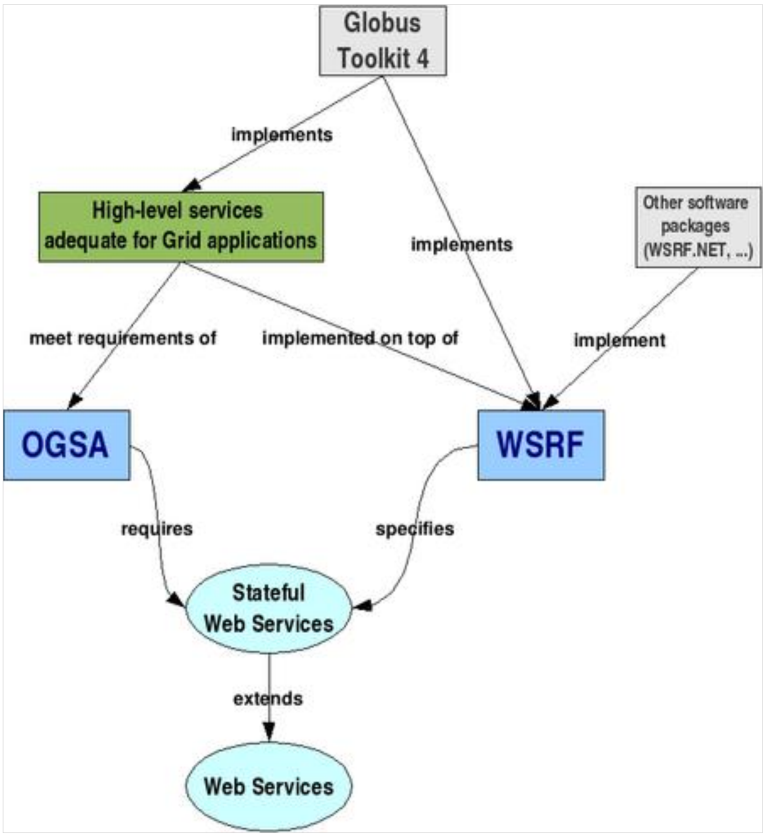

Figure 2. OGSA, WSRF and GT4
Source: www.globus.org

4. Web Services Oriented Architecture of the DPI based Network Forensics Grid

SOA (Service Oriented Architecture) for the grid provides methods for exposing services and allowing computers to talk to each other in a highly heterogeneous environment. The benefits of a SOA architecture on the grid is that it provides loose coupling which results in the system being flexible, scalable, services are replaceable and adds fault tolerance [21].

A Deep packet Capturing (DPC) machine archives the network traffic and forwards it to the web-services based applications on the Grid, which we shall refer to as the DPI-FA Grid. In the web service oriented architecture model of the DPI-FA Grid, every application runs as a service and is associated with a *WS-resource.* The web service is defined in the Web Services Description Language (WSDL) document. The *Resource Properties Document schema*, which explicitly describes a view of the resource with which the client interacts, is referenced by the WSDL description of the service . By exploiting the *Resource Properties Document schema*, WSRF enables the mechanical definition of simple, generic messages which interact with the *WS-Resource*. A *WS-Resource* is accessed through a Web service at a particular location which is described by a WS-Addressing [WS-Addressing] EndpointReference (EPR), which contains the URI of the Web service endpoint as follows:


```
…

String url = "http://local/dpiWebService"
EndpointReferenceType epr = new EndpointReferenceType();
epr.setAddress(new Address(url));
epr.setProperties(props);
```

This EPR is known to the client/requester as the service-name. The contents of the EPR are generated by the service provider and are not intended to be understood by the client. The client takes the EPR from the response message and associates it with the client's programmatic representation of the resource (for example, a program object, reference or variable) so that all calls which use that representation are directed to the resource. The 'GetResourcePropertyDocument' operation can then be used to get the retrieve the values of all resource properties associated with the WS-Resource. The SOAP envelope for the service returns the response to the request with the the 'GetResourcePropertyDocumentResponse' message [22].

The DPI based Forensic Analysis Web Service (DFA-WS) will be a network accessible entity that will process the SOAP messages. SOAP messages will be received by the hosting environment (Globus Container), which is domain independent. It would identify and invoke the appropriate code to handle the message. The WS implementation handles the SOAP message through domain specific code.

The service interface (a WSDL document) is published to the service registry. A service requester(a potential client) queries the service registry to search for a service that meets its needs. The registry returns a list of suitable services, and the client selects one and passes a request message to it. The client then binds to the service provider in order to execute the service (using SOAP). The web services are deployed into the GT4 web container. The GT4 Java WS Core Code implements WSRF and WS-Notification as well as supporting code for security and management[24]. The Service is coded using Java and the Globus *java ws-core* API, which depends on the Java Commodity Grid (CoG) Kit which is a GT component and some 3rd party software including Apache Axis, Apache Xerces, Apache Tomcat JNDI, Apache Addressing, and Apache XML Security, to name a few.

Apache Axis is one of the best free tools available for implementing and deploying web services, and also for implementing the web service clients. The code is designed to be used with Apache Axis as a SOAP engine plus other relevant WS components such as the WS-Addressing and WS-Security [23, 25, 26]. The end user can upload the .pcap file to the portal server based on the WSRF through a web browser. The portal used is based on the Java Portlet model. Exhibit 1 describes the deployment of the GT4 Java Web Service in a step-by-step manner.

Exhibit 1. Deploying the GT4 Java Web Service

Step 1: Define the service's interface in the Grid Web Service Description language(GWSDL) file, which describes the service's abstract interface.

Step 2:Implement Service by coding the service in java (extends GridSrviceImpl) and specify resource properties

Step 3: Configure the Deployment by defining Deployment Parameters in Web Services Deployment Descriptor (WSDD) file and Java Naming and Directory Interface (JNDI)) deployment file that describes various aspects of the service's configuration.

Step 4: Compile and generate the Grid Archive(GAR) file, which is a single file containing all the files and information the grid services container would need to deploy the service. The GAR files are deployed using globus-deploy-gar and undeployed using globus-undeploy-gar.

This step converts the GWSDL into WSDL and creates the stub classes from it. It compiles the stub classes and the service implementation. All the files are organized into a very specific directory structure. Compilation generates application specific interface routines that handle the de-marshalling/marshalling of the Web service's arguments from/to SOAP messages. (Ant,build.xml)

Step 5: Deploy the service into a grid service using Ant. The Ant task calls the generateLauncher target which is specified in $GLOBUS_LOCATION/share/globus_wsrf_common/build-launcher.xml.

5. Related Work

The implementation presented in this paper Globus Toolkit Java version of WS-Core. Apart from this, there are at least four more different popular implementations of the SOAP, according to the programming model and the programming language. The discussion in this section supports the selection of the Globus Java WS-Core.

An additional implementation of the standards which is part of the Globus Toolkit is implemented in C and lets one develop Grid services in C. pyGridWare [27] is also part of the Globus Toolkit, which allows the user to rapidly develop Grid services in Python. Similarly, WSRF.NET [28] is used to develop Grid services in any .NET language. With this implementation developing Grid services is not much different than programming Web services, the developer only needs to annotate what parts of the service should be made persistent. Lastly there is also a Perl implementation of the standards called WSRF::Lite [29]. These five implementations have been compared in terms of functionality and performance by Humphrey et.al. [30]. This comparison concludes that the Globus Toolkit provides the most complete implementation of WSRF and has superior response times to invocation calls when compared to WSRF.NET, WSRF::Lite, and pyGridWare. Heinis *et. al.*, extended the single-client setup study by using a multiple client setup to show how the system performs with a larger number of concurrent requests [31]. The Globus Toolkit has a robust implementation of WSRF, therefore it eliminates the problem of dealing with an incomplete server-side implementation of WSRF and concentrates work on the client-side issues of communicating with WSRF-based web services from various platforms. In addition, the Globus Toolkit offers a high-performance version written in the C language and a platform-agnostic version written in Java[32].

Humphrey *et. al.*, have clearly indicated that the C WS Core of the Globus Toolkit 4, gives the best performance when evaluated for the key primitive functions and a notification function. GT4 Java WS Core and WSRF.Net were reasonably better than the WSRF::Lite and the pyGridWare. pyGridWare,the Python WS Core from the Lawrence Berkeley National Laboratory was faster in certain scenarios only. The Perl-based WSRF::Lite from the University of Manchester implements TLS caching as it works on the Microsoft IIS, hence is faster with https [30]. GT4 C WS Core though performs fastest, GT4 Java WS Core seems to be a better choice as it has more support and use at both the academic institutions as well as the industry and is faster and easier to develop and implement. If a user wants to create a C client application he must write C implementation of the delegation by means of C WS Core API, because for C, there Delegation API is not available, only C WS Core API and C WS-GRAM API are there, whereas to write Java client application, submitting and managing job, Java WS Core API, Delegation API and WS-GRAM API are all available[33].

Kubert and Thai have also investigated the performance of the Globus Toolkit WSRF and WSRF::Lite implementations. The latency, serialization and deserialization benchmark results indicated WSRF::Lite performed better than GT4 [34].

REST(Representational State Transfer) is emerging as an easier and more flexible alternative to SOAP and WSDL based Web Services[35]. RESTful web services make use of JavaScript Object Notation(JSON) [36] instead of XML for resource representations and exploit HTTP methods explicitly for the resource manipulation instead of defining their own [37]. The adoption of REST by mainstream Web 2.0 service providers, including Google,Yahoo and Facebook and large organizations like Wal-Mart, indicates its growing acceptance, but REST can not replace SOAP everywhere, it comes with its own set of issues. A very good comparison is made between REST and SOAP & WS* by Cesare *et. al.*, [38], which indicates that the choice should be made after careful consideration of key aspects according to the requirements of an application.

6. Conclusion

The research on network security indicates that Deep Packet Inspection (DPI) based IDS has a better capability in securing networks from the new and emerging threats. DPI based forensic analysis of network traffic can carry out in-depth and unhurried analysis of the attacks and help to design a stronger security system, but is quite resource intensive. It calls for a lot of computational power as well as storage space(both short term and long term). Industry giants are successfully embracing web services and utilizing the Grid abilities, one recent example is the growing popularity of the Amazon Web Services being offered by Amazon [39, 40].

This paper proposes the use of the high computational abilities of Grid infrastructure for performing the task in a more cost effective, fast and reliable way. The development of the grid application using WSRF maximises the potential of the Grid capabilities, enable it to be loosely coupled and scale to arbitrary size. Amongst the various WSRF implementations and the emerging implementations like RESTful WS implementations, GT4 Java WS Core is a good choice as it is established, well supported, and proven good performance.

Acknowledgements

We would like to express our gratitude to the reviewers whose advice contributed to major improvements in the paper. We are thankful to our colleague at Thapar University, Ratinder Kaur who provided the necessary motivation. Sincere acknowledgment of gratitude is for our young children in our respective families for being the source of joy and relaxation needed after hours of sitting with the laptop and our respective spouses for being patient and supportive.

References

[1] M. Reith, C. Carr, and G. Gunsch, "An examination of digital forensic models," International Journal of Digital Evidence.,Vol. 1, Issue 3 **(2002)**.
[2] V. Corey, C. Peterman, S. Shearin, M.S. Greenberg, and J.V. Bokkelen, Network forensics analysis, IEEE **(2002)**, Volume 6, Issue 6, pp. 60 –66.
[3] N. Chawla, Grid computing, Developer IQ, 6(1), Techmedia Publications **(2007)**
[4] M. Gillespie, Web Services Extend High-Performance Computing Grid Capabilities, Retrieved **(2012),** from https://software.intel.com/en-us/articles/web-services-extend-high-performance-computing-grid-capabilities
[5] J.P. Park,C.H. Yun, H.S. Jung,Y.W. Lee, Mobile Cloud and Grid Web Service in a Smart City. Proceedings of The Fifth International Conference on Cloud Computing, GRIDs, and Virtualization, pp. 20-25, **(2014)** May 25-29;Venice,Italy.
[6] A. A. Yıldırım, D. Tarboton, P. Dash, & D. Watson, Design and Implementation of a Web Service-Oriented Gateway to Facilitate Environmental Modeling using HPC Resources. Proceedings of the 7th

International Congress on Environmental Modelling and Software (iEMSs) **(2014)**; San Diego, CA, USA
[7] S. Parastatidis, P. Watson, and J. Webber, "Grid Computing Using Web Services," Technical Report (CS-TR-926), University of Newcastle upon Tyne, School of Computing Science **(2005)**
[8] Z. Chen and C. Lin, "Antiworm NPU Based Parallel Bloom Filters For TCP/IP Content Processing in Giga-Ethernet Lan," In IEEE Proceedings of Local Computer Network, The First Workshop on Network Security(WoNS) , **(2005)** November 15; Sydney, Australia
[9] J. Newsome, B. Karp, and D. Song, "Polygraph: Automatically generating signatures for polymorphic worms," Proceedings of the IEEE Symposium on Security and Privacy, **(2005)** May; Oakland
[10] L. Musthaler. Rewind and replay what happens on your network. Retrieved **(2011)** from www.networkworld.com/newsletters/techexec/ 2007/ 0716techexec1.html
[11] P. Venezia, Netdetector captures intrusions. Infoworld , July **(2003)** , Issue 27
[12] I. Foster, and C. Kesselman, "The Grid: Blueprint for a New Computing Infrastructure," Morgan Kaufmann, San Francisco **(1998)**
[13] J. Joseph, and C. Fellenstein, Grid Computing. Prentice Hall/IBM Press **(2004)**
[14] B. Jacob, M. Brown, K. Fukui, and N. Trivedi, Introduction to Grid Computing, IBM Redbooks (**2005)**
[15] IPSoft website. Resources, Articles, Green Computing. Retrieved from www.ipsoft.com/us/component/content/article/146 **(2008)**
[16] Chesapeake energy deploys oracle grid computing, Retrieved **(2007)** from http://www.oracle.com/us/corporate/press/015273_EN
[17] S.C. Kendall, J. Waldo, A. Wollrath, and G. Wyant, A Note on Distributed Computing, Sun Microsystems,TR-94-29 **(1994)**
[18] Globus Alliance, IBM and HP, The WS-Resource Framework. http://www.globus.org/wsrf/ **(2004)**
[19] The Globus toolkit, [online], Jan. **(2014)** Available from: http://www.globus.org/tookit/
[20] M. Li and M. Baker The Grid: Core Technologies , John Wiley & Sons **(2005)**
[21] L. Srinivasan, and J. Treadwell, An Overview of Service-oriented Architecture, Web Services and Grid Computing by HP Software Global Business Unit ,November 3, **(2005)**
[22] T. Banks, OASIS WSRF Primer. Committee Draft 01. Retrieved from docs.oasis-open.org/wsrf/wsrf-primer-1.2-primer-cd-01.pdf **(2005)**
[23] Apache AXIS: ws.apache.org/axis/
[24] Web Services Brokered Notification 1.3 http://docs.oasis-open.org/wsn/wsn-ws_brokered_notification-1.3-spec-os.htm, OASIS Standard, 1 October **(2006)**
[25] Web Services Addressing 1.0 - Core, W3C Recommendation. http://www.w3.org/TR/ws-addr-core , **(2006)**
[26] OASIS Web Services Security (WSS) TC, https://www.oasis-open.org/committees/wss/
[27] pyGridWare: Python Web Services Resource Framework, http://dsd.lbl.gov/gtg/projects/pyGridWare/
[28] M. Humphrey, G. Wasson, M. Morgan, and N. Beekwilder, An Early Evaluation of WSRF and WS-Notification via WSRF.NET, in Grid Computing Workshop (associated with Supercomputing 2004), **(2004)** Pittsburgh, PA, USA
[29] WSRF::Lite – Perl Grid Services, http://www.sve.man.ac.uk/Research/AtoZ/ILCT
[30] M. Humphrey, G. Wasson, K. Jackson, J. Boverhof, M. Rodriguez, J. Bester, J. Gawor, S. Lang, I. Foster, S. Meder, S. Pickles, and M. McKewon,State and Events for Web Services: A Comparison of Five WS-Resource Framework and WS-Notification Implementations, in Proceedings of the IEEE International Symposium on High Performance Distributed Computing (HPDC-14) **(2005)** Research Triangle Park, NC, USA
[31] T. Heinis, C. Pautasso, O. Deak, and G. Alonso, "Publishing Persistent Grid Computations as WS Resources," in Proceedings of the 1st IEEE International Conference on e-Science and Grid Computing, December (**2005)** Melbourne, Australia
[32] Huang, Shihong, et al. "Remote computing resource management from mobile devices by utilising WSRF."International Journal of Computer Aided Engineering and Technology, 2.2-3 , December 2005. **(2009)**: pp.199-217
[33] Report No.2 on Globus Toolkit 4 evaluation by joint JINR (Dubna), KIAM and SINP MSU team , Phase 2: Evaluation 1 (1/06 – 20/07), Retrieved **(2015)** from theory.sinp.msu.ru/dokuwiki/lib/exe/fetch.php?cache.=cache&media=egee:gt4:gt4eval_report2.pdf
[34] R. Kübert; H. Thai, A performance comparison of four WSRF implementations, International Journal of Electronic Business (IJEB), Vol. 9, No. 5/6 **(2011)**
[35] R. Fielding, "Architectural styles and the design of network-based software architectures," Ph.D. dissertation, University of California, Irvine **(2000)**
[36] D. Crockford, "The application/json media type for javascript object notation (JSON)," RFC 4627, Jul. **(2006)**
[37] B. Schuller, J. Rybicki, K.Benedyczak, High-Performance Computing on the Web: Extending UNICORE with RESTful Interfaces,Proceedings of AFIN 2014, The Sixth International Conference on Advances in Future Internet, November **(2014)** ISBN: 978-1-61208-377-3

[38] Pautasso, Cesare, Olaf Zimmermann, and Frank Leymann. "Restful web services vs. big'web services: making the right architectural decision. Proceedings of the 17th international conference on World Wide Web. ACM, **(2008)**
[39] Amazon Web Services, Retrieved **(2014)** from http://aws.amazon.com
[40] K. Raz. Amazon Web Services Deploys NVIDIA GRID GPUs **(2013),** Retrieved from NVIDIA News, http://nvidianews.nvidia.com/news/amazon-web-services-deploys-nvidia-grid-gpus-2775398

Authors

Dr. Maninder Singh is an Associate Professor at the Computer Science and Engineering Department,Thapar University, Patiala and is also heading the Centre of Information and Technology Management (CITM), responsible for University wide Network, ERP, IT Strategy: planning, deployment and management. He received his Bachelor's Degree from Pune University, Master's Degree, with honours in Software Engineering from Thapar Institute of Engineering & Technology, and holds his Doctoral Degree with specialization in Network Security from Thapar University.His research interest includes Network Security, Grid Computing, Secure coding and is a strong torchbearer for Open Source Community. He has many research publications in reputed journals and conferences. He is on the Roll-of-honour @ EC-Council USA, being certified as Ethical Hacker (C|EH), Security Analyst (ECSA) and Licensed Penetration Tester (LPT).

Dr. Singh architected Thapar University's network presence, which is successfully implemented in a heterogeneous environment of wired as well wireless connectivity. He has successfully completed many consultancy projects (network auditing and penetration testing) for renowned national bank(s) and corporate. In 2003 his vision for developing an Open Source Based network security toolkit was published by a leading national newspaper. Linux For You magazine from India declared him a 'Tux Hero' in 2004. He is a Senior Member of IEEE, Senior Member of ACM and Life Member of Computer Society of India. He has been volunteering his services for Network Security community as a reviewer and project judge for IEEE design contests. Recently Dr. Singh was aired on "Centre Stage" @ Headlines Today, national channel.

Jyotsna Sharma is a research scholar at the CSED,Thapar University. She has focused her research on DPI based Forensic Analysis of Network Traffic using Grid Infrastructure. She is an M.Phil. in Computer Science and also a Graduate Member of The Institution of Engineers(India). She is a Certified Ethical Hacker(C|EH) from the EC-Council. She has several research articles to her credit and has also contributed a chapter to the 'Handbook of Research on Grid Technologies and Utility Computing, an IGI Global Publication, and is currently authoring a book on 'Web Engineering'.She received the Suman Sharma National Award from the Institution of Engineers(India) for academic distinction in the computer engineering discipline. She won the 2009 Google Global Community Scholarship for GHC2009. She has several years experience as an Assistant Professor and a Software Developer.