Volume 6, Issue 4, April 2016 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

Service\Object Distribution Evaluation

Mustapha M. Baua'a
Financial Dept., Thiqar Oil Division Fields,
South Oil Company, Nassiriya, Iraq

Ahmed M. Hailan
Thi-Qar University College of Computer Science
and Mathematics, Thi-Qar, Iraq

Abstract— The autonomous components positioned at different locations on computer networks are known as Distributed Systems. The components are communicating and coordinating their actions by passing messages. The independent entity in Distributed Systems can be Operating System, Object, Web Service, or business process. However, Web Services have been used to implement the Case Study, where Service represents as the atomic component at Distributed System. Also RMI has been used to implement the same Case Study. The model applied for two applications of the case study was Client\Server, where focusing on two methods for downloading and uploading files are used. Validated results and implemented applications are shown and clearly prove the importance use of SOA principals in improving Distributed System Performance.

Keywords— Distributed Systems; SOA; RMI.

I. INTRODUCTION

Distributed systems are set of independent components located at computers network communicating and coordinating their works only by passing messages using certain middleware. The component in distributed systems can be service, platform, operating system, object, etc. where; there are many middleware that have been used to connect these scattered components on the network such as Service-Oriented Architecture SOA, Common Object Request Broker Architecture CORBA, and Remote Method Invocation RMI. RMI has the ability of connecting distributed objects but it deal with single language, while CORBA can works with different programming languages. However, SOA has the ability of building new distributed systems. The common types of distributed systems are Distributed Computing, Information, or Pervasive systems. The techniques distributed components use to interact between each other are tightly coupled and loosely coupled. The term loosely here has the meaning of decreasing the dependency of components on each other.

Service-Oriented Architecture SOA is an architectural style that has been utilized to define loose coupling distributed components for reusability, stateless services, extensibility, and interoperable benefits [1, 2]. It's neither system architecture nor a complete system. SOA has the ability of designing a logical software system in order to provide clients with services or other distributed services over Internet by publishing and finding them using interfaces. The important aims at SOA are building autonomous protocols, and identify loosely coupled needs for distributed systems [3]. The most common implementation for Service-Oriented Architecture SOA is Web Services. Web Services is an electronic service that can be published on Web in order to call it remotely or locally. The way Web Services impact on Distributed Systems is increased, where the important characteristic is interoperability provided by them over Internet especially in e-business. Usually, Web Services uses interface techniques for requests acceptation. The interface contains the operations Web Services provides, in which operation types, parameter types, but not whole operation details. The operation\method details are implemented by Service itself, this prevents users from accessing data Services. However, the basic processing source the Web services depend on is Service. The set of objects or methods or even mini services that have being used to serve specific purposes are called Services. Many Service characteristics are supporting such as self-contained, reusability, autonomous from other services and support standard business functionality.

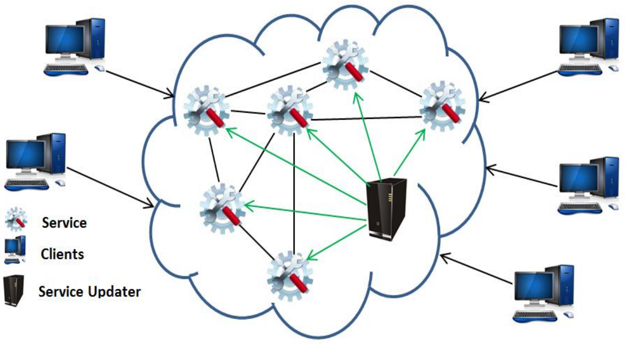


Fig. 1 Distributed Service-Based System.

The figure of number (1) clarifies the Distributed Service-Based System where service considers the basic milestone of the system. I put Service updater in order to make services up to date. This figure abstractly figures out the clue for Oil Field Divisions. And how to convert the departments into set of services cooperate and interact between each other.

While, Object-Oriented architecture is the most important programming paradigm use today in constructing distributed systems. These distributed objects use a middleware in order to interact among each other. One of those middleware used is Remote Method Invocation (RMI), where Object is considered the basic atomic for that system [4].

Object is a set of methods and data that represents the reason behind hiding for the behavior of object from user. There are many programming languages that support creating objects such as C++, and Java. Objects have references that can be called by them and also object can implement many Interfaces. Objects can be distributed over many computers and construct finally distributed objects. Distributed Objects is just like client\server architecture where server manages the objects and client can invoke the object's methods of Remote Method Invocation RMI [5].

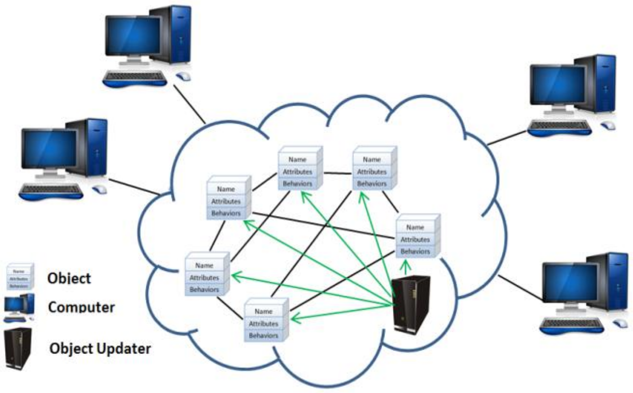


Fig. 2 Distributed Object-Based System.

The figure of number 2 figures out the same distributed systems I suggest but with Object-Oriented principals. Where one can see the basic milestone for this system is Object.

However, the idea is about building a Distributed Systems depending on the SOA principals. The aim is to convert every department or class as a service. Then the whole organization become a set of services interchange their documents through passing messages. I found using SOA as a Distributed Systems paradigm very appropriates for arranging, utilize, and reuse distributed systems capabilities for service deploying dynamically. The focus for this paper is on interchanging files between clients and service.

This paper organized as follows: introduction in section 1. State of the arts is represented in section 2. In section 3 the analyzing of basic systems are introduced. Case study, implementations and results are presents in section 4 and 5 respectively. In section 6 the conclusion is represented.

II. STATE OF THE ARTS

In [6] they propose a new architecture framework that is benefit of SOA principles in order to deploy Unified Communication over the Internet. Where, they are providing distributed communication networks for a variety of regions. The OSGi framework has been used because it depends on SOA principles. Also, one of the OSGi benefits is the service provide dynamic composition change on the equipment of different networks without needs to restarts. The focus on their work was on giving an end-to-end infrastructure and the methodology of deploying and delivering to the service.

RE-portal was written as unique Java servlet where it yielded to tightly couple system among the user management, the equipment, and the client interface. In [7] they propose a Distributed Service-Oriented Architecture for the new RE-portal. They re-engineer RE-portal by providing service wrappers around the embedded tools of the RE-portal, and then they integrate services by JSP-based web service client.

Authors of [10] answers the skeptics who consider Web Services as a model of distributed systems, where they see it as poor implementation of Remote Procedure call RPC. As a result they do their study to compare the design and implementation of a small file server application applied by RMI and Web Services. The comparison is verifying the good performance of Web Services even with high latency in Internet or business transactions.

III. DISTRIBUTED SERVICE\OBJECT-BASED SYSTEMS ANALYZE

Companies now are able to reduce the cost of doing e-business, deploying solutions faster, and an opening up new opportunities by using Web Services. The common model supported by Web Services now is program-to-program communications. The model built on existing emerging Internet standards protocols. Web Services are represented by Web Services Description Language WSDL, and transmitted by Simple Object Access Protocol SOAP through Hypertext Transfer Protocol HTTP. A consumer discovers services and gets their binding information on Universal Description Discovery and Integration UDDI-based registries. According to WSDL documentation provider and consumer are formed their messages by XML. This uses of XML for representing Web Services messages make it more convenient to work with heterogeneous environment. Also the characteristics of Web Services such as applications integrated rapidly, their messages centered more on service semantics and less on network protocol semantics, etc. is to make it more ideal for e-business connection [8, 9]. In addition, if properly selected and implemented Web Services can expose many benefits in terms of e-business development. The benefits are New IT solutions delivered faster to lower cost, wrap legacy software systems for integration with newly IT systems, business processes, customers, and partners are integrated, and widen customers markets by entering new markets.

RMI is similar to Remote Procedure Call (RPC) with an extension to include distributed objects. The implementer can get full benefit of using Object Oriented Programming in building Distributed systems, involving use of objects, classes, and inheritance. Other benefit is passing parameters by reference, where all objects have single reference and RMI provide ability to pass this reference as a parameter. The important thing RMI adds is the transition from objects to distributed objects. Each process on Distributed Systems contains a set of objects where some of which can be invoked locally and others invoked remotely. RMI supports transparent calls for remote operation among objects carried out in variable Java Virtual Machines JVM.

Distributed objects are suffering from some problems such as dependencies, implementation difficulties, etc. Also Distributed objects have no support for object configuration deployment. Implicit dependencies on the object oriented model expose a problem of safe composition of configuration, and also for third-party developers when they are needed to apply specific entities in a distributed configuration. In order to solve the dependency problem the implementer should specify not only the interfaces provide by objects but also the dependencies on objects of distributed configuration. For distributed systems configuration issues and because the object oriented do not support deployment configuration objects should be distributed manually. With large number of nodes distributed systems the deployment process will be tiresome and error-prone.

IV. CASE STUDY

The case study is about designing, coding, and implementing a system for Oil Fields Division of Thi-Qar in South Oil Company. It involves eight departments and each department has its classes. These departments need to interchange their official documents quickly and in secure manner. However, the focus is on documents interchanging client and service. The Web Services have been used to implement our focusing and for comparison issues RMI also have been used to implement the same idea.

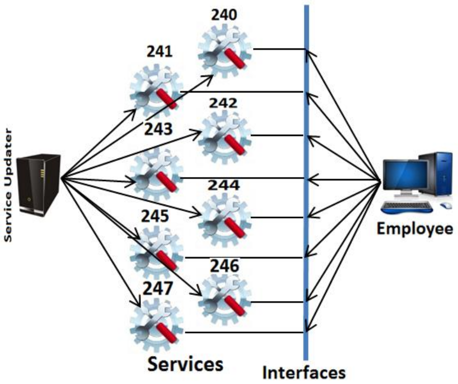


Fig. 3 Thi-qar Oil Fields Division Distributed Service-Based System abstract figure.

The figure 3 represents an abstract view of the constructed system. Three major parts this system consists of, one is Service Side, second is Interface, and third is Client side. In service side there are many Web Services that provides clients with their needs. For each department a main web service has been designed and each one contains mini Web Services. Interface represents significant mean that connects clients with service's department he needs. The client side has one implementation with many frames and the choice which departments he needs to connect with is free.

In this paper the focus on the raw idea where how a system can be constructed using Service depending on SOA. For honestly issue one department is implemented (i.e. financial department) and for comparison issues with the same thing for RMI application.

V. IMPLEMENTATIONS AND RESULTS

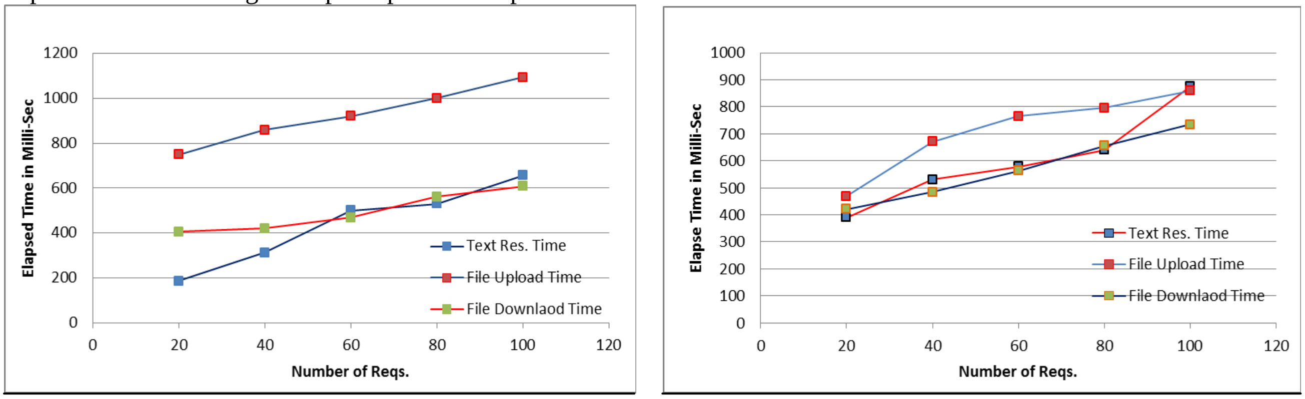
The implementations are categorized into two types: one is implemented using Web Services and second is using RMI. Also each one of these implementations executed twice one on virtual environment (localhost) and other on real environment (multi-computers). In the two implementations the concept of Java-Threads has been used in order to implement number of requests. And for Multi-computers two computers have been used one as a service provider and second as client.

I. Implementations: Virtual Environment (localhost)

The figures below clarify the execution time of Distributed Service-Based System using SOA principals that has been constructed. The part of the application paper use here is encoded and implemented on Sony Vaio Laptop with Intel® Core™ i5-3230M CPU @ 2.60GHz, and 32-bit Operating System. The Web Service is used to implement the application it consists of eight departments and they are connected with each other by Web Services. Each department has a Web service and in each Web Service there are many mini-services according to a department needs. There is one client implementation and this client can connect to any department he wants. The same thing is happen for the application implemented with RMI, where time performance clarified.

I.I. Performance Analysis

The performance is analyzed by checking the request time for client request from send till it gets back Web Service's answer. The Java-Multi-Threads has been used to implement multiple client requests. The two applications are implemented and executed on localhost. The only difference is the figure 4 for Distributed Systems where web service is applied and the figure 5 is for Distributed Systems where Object-oriented is applied. The flowcharts below show the impact benefits of using SOA principals in comparison to RMI.



a b

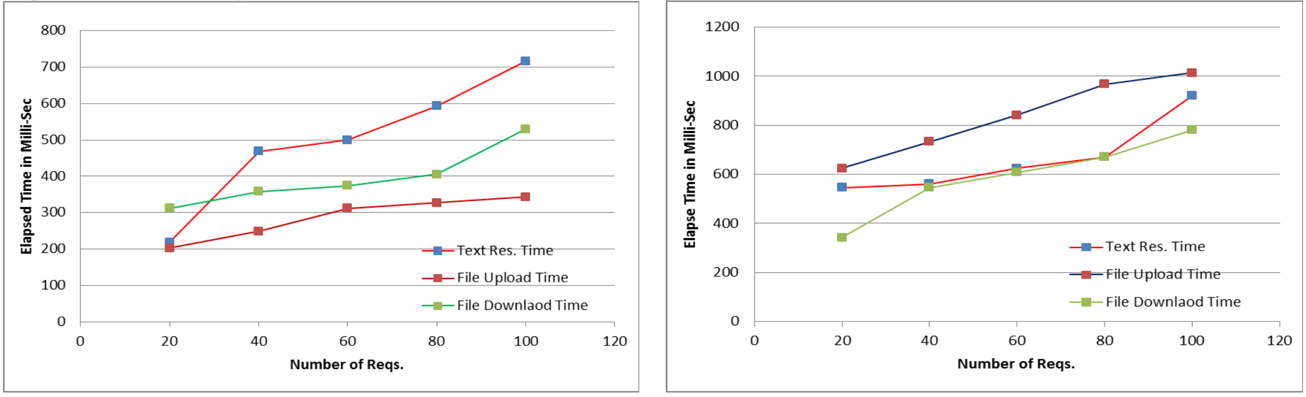
Fig. 4 (a) Distributed Service-Based System Using SOA on Localhost; (b) Distributed Object-Based System Using RMI on Localhost.

II. Implementations: Real Environment (Multi-Computers)

The flowcharts below clarify the execution time of the two applications in both methods e.g. Service Oriented and Object Oriented. The two figures results are gained by executing the implementation on two computers one is Laptop and other desktop.

II.I. Performance Analysis

The benchmark results presented in flowcharts are (figures: 4, 5, 6, &7) clarify the lake performance of RMI as implementation of Object-Oriented Architecture OOA in comparison of SOA.



a b

Fig. 5 (a) Distributed Service-Based System Using SOA in real environment; (b) Distributed Object-Based System Using RMI in real environment.

VI. CONCLUSION

Service-oriented architecture SOA is an architectural style and not a specific language. Building loose distributed systems fortunately had been supported by SOA principals. However, the benefit was obtained from these principals in order to build Distributed Service-Based System. The System showed a good time performance in comparison to Distributed Object-Based System.

In future work the plane is to expand the comparison of involving CROBA, and others middleware. Also the will work be on a new paper about parallelizing Service.

REFERENCES

[1] Martin Keen, Rafael Coutinho, Salvatore Sollami, Sundaragopal Venkatraman, Steve Baber, Henry Cui, and Craig Fleming, "Developing Web services Applications", ©Copyright IBM Corp. 2012. All rights reserved.

[2] Olivier Pilotte, Roger Boudreault, Wei Lim Farzad Amirjavid, and Hamid Mcheick, "Service-Oriented Architecture and Load Balancing", Architecture des applications d'entreprise, Department d'informatique et de mathematique, May 13m 2010.

[3] Mike P. Papazoglou, Willem-Jan van den Heuvel, "Service-Oriented Architectures: approaches, techniques, and research issues", The VLDB Journal (2007) 16:389-415, DOI 10.1007/s00778-007-0044-3.

[4] Soule, Paul, "Autonomics Development: A Domain-Specific Aspect Language Approach", 7-39 Book Series: Autonomic Systems, ©2010 Springer Basel AG.

[5] George Coulouris, Jean Dollimore, Time Kindberg, and Gordon Blair,"Distributed Systems: Concepts and Design", Lancaster University, ISBN 10: 0-13-214301-1, ISBN 13: 978-0-13-214301-1, ©2012.

[6] Shrikant Kanaparti, "A Distributed-SOA model for Unified Communication Servies", Samsung Electorins, 2008 IEEE International Conference on Web Services, E-mail: shrix@ieee.org.

[7] William M. Mongan, Maxim Shevertalov, and Spiros Mancoridis, "Re-Engineering a Reverse Engineering Portal to a Distributed SOA", Drexel University, Department of Computer Science, 3141 Chestnut Street, Philadelphia, PA 19104, E-mail: {wmm24, ms333, spiros}@drexel.edu.

[8] Heather Kreger, "Web Services Conceptual Architecture (WSCA 1.0)", IBM Software Group, May 2001.

[9] [1]Phil Bianco, [2]Rick Ktoermanski, [1]Paulo Merson, "Evaluating a Service-Oriented Architecture", [1]Software Engineering Institute, [2]Summa Technologies, September 2007, Technical Report CMU/SEI-2007-TR-015, ESC-TR-2007-015.

[10] William R. Cook, Janel Barfield, "Web Services versus Distributed Objects: A Case Study of Performance and Interface Design", Department of Computer Science, University of Texas at Austin, Austin, Texas 78712, Email: {wcook, janelbarfield}@cs.utexas.edu, publish year 2007.