ISSN 2047-3338

# Decreasing Response Time in Distributed Computing for SOA Based Applications

Farooq Javed[1], M. Shafique[2] and Salman Ahmad[3]
[1,3]Department of Computer Science & IT, University of Sargodha, Sargodha, Punjab Pakistan
[2]Department of CS&E, University of Engineering & Technology, Lahore, Pakistan
[1]farooqjavid@yahoo.com, [2]se.muhammad.shafique@gmail.com, [3]jajja999@yahoo.com

***Abstract*– This article debates vital performance glitches in the design of SOA based application while it is deployed in distribute environment to take the benefits of distributed computing. There is major factor involve in the distributed computing environment is the network latency among the server machines containing different modules of application to reach one goal. They all communicate with each other on the bases of message passing. Latency among the server machines may reduce and over all response time is increased by applying different protocols like TCP/IP, HTTP, and Named Pipe for message passing and RPC. These are effective in reducing the overall response time. Besides these all in Data Driven applications there are various types of data mapping mechanism like entity framework and Nhibernate are introduced. But still there is one more possibility left to make architecture more efficient and fast. We may increase response time by moving business logic layer from application server to database server. Load testing experiments prove that this technique bring significant improvement on large data transactions which needs heavy processing at business logic layer.**

***Index Terms*– Service Oriented Architecture, SOA, Response Time, Distributed Computing and SOA Based Application**

## I. INTRODUCTION

THE performance assessment of distributed computing environment is a complicated and bit problematic because of the complex collaboration among a potentially large number of system modules. Before discussing some relation between SOA and Distributed computing; it is needed to understand both these terms. Distributed computing is used to solve many computational problems by means of distributed systems. In distributed computing, we divide a problem into different sub-problems and each of which is assigned to one or more computer systems [3]. These computers exchange information with each other by message passing [4]. Service Oriented Architecture (SOA) based application while it is deployed in distribute environment to take the benefits of distributed computing.

There is major factor involve in the distributed computing environment is the network latency among the server machines containing different modules of application to reach one goal. They all communicate with each other on the bases of message passing. Latency among the server machines may reduce and over all response time is increased by applying different protocols like TCP/IP, HTTP, and Named Pipe for message passing and RPC. These are effective in reducing the overall response time. Besides these all in Data Driven applications there are various types of data mapping mechanism like entity framework and Nhibernate are introduced. But still there is one more possibility left to make architecture more efficient and fast. The question is how can we increase response time by making architectural change in the system? The answer of this question is, we my increase response time by moving business logic layer from application server to database server. Load testing experiments prove that this technique bring significant improvement on large data transactions which needs heavy processing at business logic layer.

But there are a few considerations to be kept in mind before developing such application and that is the selection of database management system. Database management system should be versatile and should contain upgradable capability because we are going to make business logic layer from application level to database level. It will provide us better load management and will reduce a server layer.

### *A) Major Performance Indicators*

The total amount of time which a system takes to respond to some service request is called its response time. Service request can be of any type like a complex database query, loading a full web page, disk IO or a memory fetch. If transmission time is ignored for just a moment, then response time will be the sum of the wait time and service time. The amount of time taken to do the requested work is service time.

In a packet-switched network, the network latency can be measured in two ways; either round-trip delay time (the one way latency from source to destination plus the one way latency from the destination back to the source) or one way (the time a packet takes from source sending to the destination receiving it). Round trip latency can be measured from a single point; therefore, it is more frequently cited. The amount of time spent by the destination system for the processing of a packet is excluded from the round trip latency.

Another service provided by many software platforms that can be used to measure round trip latency is called ping. Ping simply sends a response back whenever it receives any packet; it never carry out any processing, thus it is a first rough way of measuring latency. Instead of using TCP (a real communication protocol), ping utilizes ICMP which is used only for diagnostic or control purposes, therefore, accurate measurements principally cannot be performed by ping. Furthermore, different traffic shaping policies are applied to different protocols by routers and ISP's [5], [6].

## II. RELATED WORK

### *A) Distributed computing*

Computer science field in which we study distributed systems is called distributed computing. In a distributed system, computer system components distributed (physically located on different locations) over the network coordinate and communicate about their actions to perform through message passing [1] in order to achieve a common goal. Most important characteristic of a distributed system is that they lack global clock, concurrency and independent fault tolerance of components [1]. Massively multiplayer online games, SOA-based systems and peer-to-peer applications are examples of distributed systems. Distributed programming is the process of writing computer software packages which are used in the distributed systems; these packages are called distributed programs [2]. Other than message passing, many other mechanisms exist for communication including message queues and RPC like connectors. Location transparency is another goal and challenge of distributed systems. Fig. 1 is showing distinction between distributed and parallel computing.

Concurrent systems can be classified roughly as "parallel" or "distributed" by using the standards mention below:

- In parallel computing, all processors may have a right to access a shared memory for the sake of exchanging information between processors as shown in (c) of Fig. 1 [7].
- Each processor has its reserved memory (distributed memory) in distributed computing. Processors exchange their information by passing messages to each other as shown in (a) and (b) of Fig. 1 [8].

### *B) Service Oriented Architecture (SOA)*

Service Oriented Architecture (SOA) is a design pattern consisting of separate and distinct parts (Components, modules) of software to provide its application functionality as service to other soft-wares or application by using different protocols like HTTP, HTTPS, SOAP and TCP. Its services are independent of product or technology [9]. A service is a unit of functionality e.g., fetching online bank account information [10]. Services can be map with other applications to achieve a complete functionality of relatively large software application [11]. It is easy for computer to cooperate with each other over the network by using SOA. In SOA services are made in such a way so they can interact with each other without any enrolment human. They are also platform independent and in general follow common standards to format their data and any computer may run certain number of services. Fig. 2 shows the SOA based application in distributed computing environment. In this figure it is clear that business logic layer is deployed on separate server and data base layer is deployed on separate Machine. Both servers are connected to each other via network.

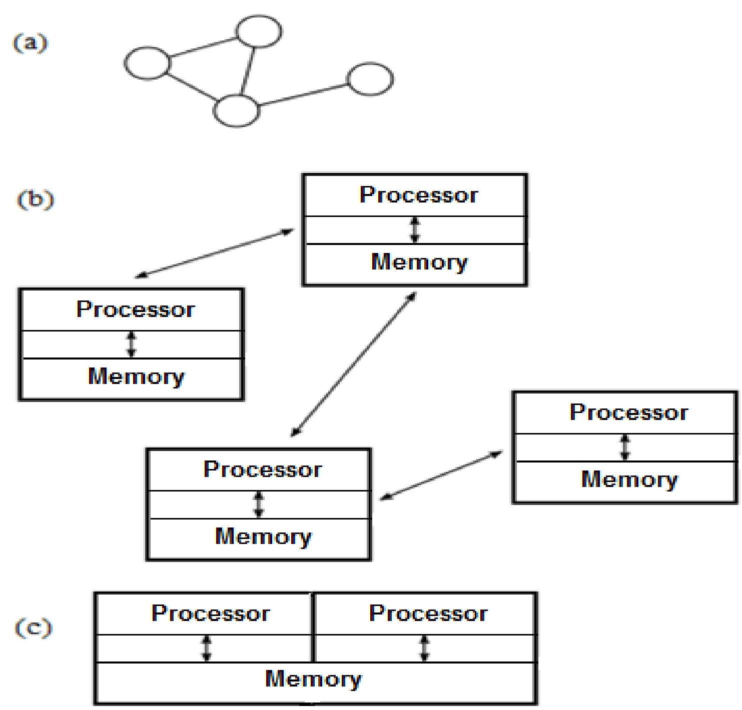


Fig. 1. Distributed computing vs. parallel computing

*C) EEE 802 Reference Model Network Operations*

The configuration of the local area network is under consideration. It consists of user systems (file server and workstations) attached to a token-ring network through ring adapters. Each adapter has a number of transmit/ receive buffers. It also contains a processor whose major tasks are to control the data transfer between the ring and the transmit/receive buffers, to manage these buffers and to control the interface to the user system. File transfer is performed over a logical connection between the file server and the workstation. The file server can manage multiple connections simultaneously. The protocol under consideration is a subset of the IEEE 802.2 Type 2 Logical Link Control protocol".

It provides procedures for connection establishment, connection termination, flow control, and error recovery. A file is transmitted as a series of Information (I-) frames. For the file transfer environment, information flow on a given connection is unidirectional; i.e., on one connection, I-frames are either sent from the file server to a workstation, or vice versa.

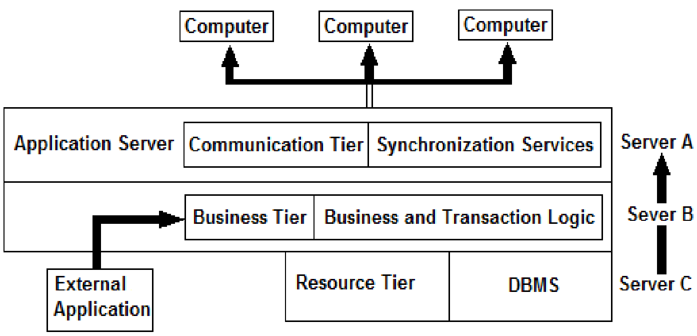


Fig. 2. SOA in Distributed Computing Environment

## III. PROBLEM AREA

According to Fig. 2 there is problem of data switching cost between Server-B and Server-C over the network. Suppose we run a transaction on millions of record where we need to run a formula and have to save it back to server-c then switching cost may have to bear which will increase net response time of the application. Fig. 3 is providing visual imagination of the scenario.

## IV. PROPOSED METHODOLOGY

There are two possible solutions of this problem:

- We may place business logic layer and DBMS on same server machine, hence we can say that both layers database layer, business logic layer which is the part of application layer will become together to avoid network latency.

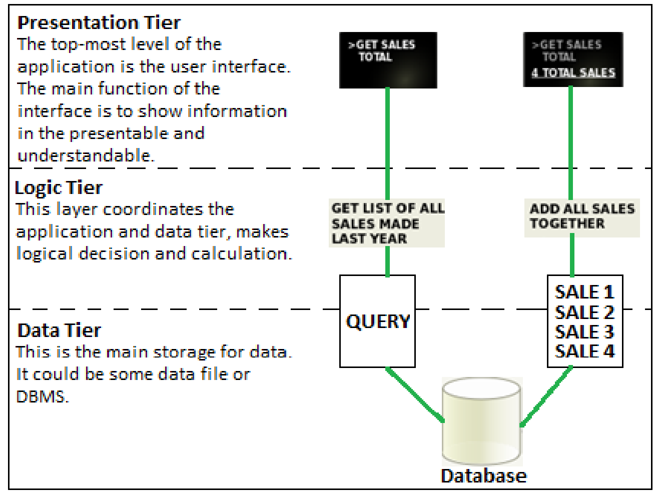


Fig. 3. Data switching over the Servers employing different layers

- We may place business logic layer and DBMS on same server machine, business layer just use as logic manger and business logic should be written in DBMS supported language in the form of stored procedures.

*A) Case Study*

In this case study, we take 1000 records whose fields are needed to multiply with some formula and result is back to store in data base. We may analyze the situation in mathematical form by driving some ration for current methodology and in our proposed methodology.

*Database and Logic layer on Different Server machines*

Fig. 4 is showing the scenario in which database server and application server for business logic layer are interconnected via network (LAN, MAN, WAN, WLAN etc) and Data B is fetched by application. Following are the Steps for applying logic on a single record:

- Fetch data from Database server
- Data travel to the server containing business logic layer
- It takes time to travel data hence
  - ***tl***→*Network latency between database server and Application server*

$$\sum_{i=1}^{n} tli = tl1 + tl2 + tl3 + \cdots + tln$$

Where n>0 (n=1, 2,3,4,5...)

Each fetching and submission and submission of data back to database server takes a complete circle or round therefore this latency become double for each fetching. There for above equation is modified as for n fetching and storing back to database server.

$$2\sum_{i=1}^{n} tli = 2(tl1 + tl2 + tl3 + \cdots + tln)$$

- *Tl→ Total Network latency between database server and Application server for n time fetching or storing of data.*
  *Eq(1).*

$$2Tl = 2\sum_{i=1}^{n} tli$$

- *2Tl→Time for complete round trip for n fetching and storing of data.*

*Note:* tl and Tl is always greater than 0

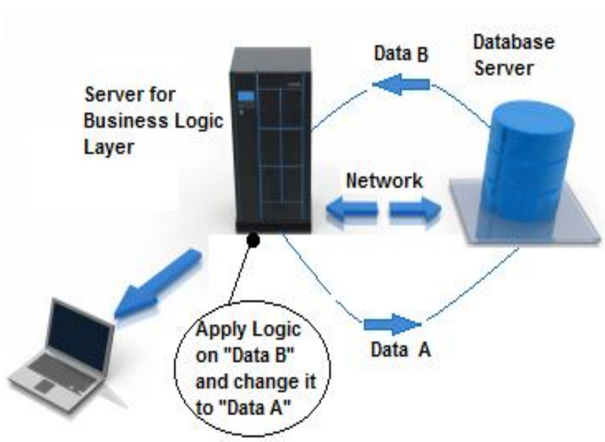


Fig. 4. Database and Logic layer on Different Server machines

*Database and Logic layer on Same Server machines*

Fig. 5 is showing the scenario in which database and business logic layer is one same server machine not connected via network (LAN, MAN, WAN, WLAN etc).

In this scenario following are the Steps for applying logic on a single record:

- Request business logic operation on data from Database server
- Operation is performed on same machine
- A little time is involve to request from Application server to database sever
- *th→Network latency between database server and Application server*

Each fetching and submission of data back to database server do not require complete circle or round therefore this latency remain half for each operation. There for above equation is modified as for n fetching and storing back to database server.

$$\sum_{i=1}^{n} thi = th1 + th2 + th3 + \cdots + thn$$

*Where n>0 (n=1, 2,3,4,5...)*

- *Th→ Total Network latency between database server and Application server for n time fetching or storing of data.*
  *Eq (2).*

$$Th = \sum_{i=1}^{n} thi$$

*Note:* th and Th is always greater than 0

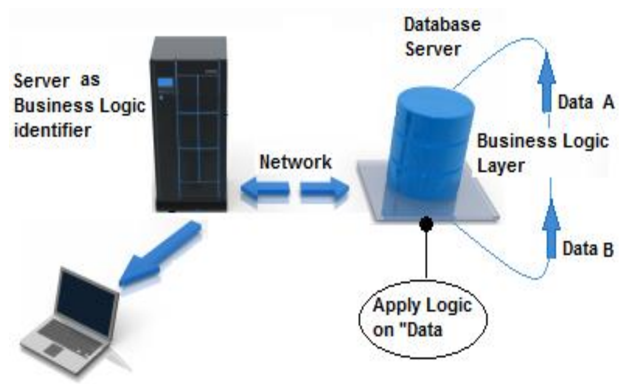


Fig. 5. Database and Logic layer on Same Server machines

## V. CONCLUSION

From all above discussion of both scenarios, it is clear from Eq (1) and Eq (2) that

*Tl=2Th*

*or*

*1/2Tl=Th*

Hence proposed methodology is better than previous one because it reduced the round trip or circle time to it's half.

$$Tl > Th > 0$$

## REFERENCES


[1]. Coulouris, George; Jean Dollimore; Tim Kindberg; Gordon Blair (2011). Distributed Systems: Concepts and Design (5th Edition). Boston: Addison-Wesley. ISBN 0-132-14301-1

[2]. Ghosh, Sukumar (2007), Distributed Systems – An Algorithmic Approach, Chapman & Hall/CRC, ISBN 978-1-58488-564-1

[3]. Andrews, Gregory R. (2000), Foundations of Multithreaded, Parallel, and Distributed Programming, Addison–Wesley, ISBN 0-201-35752-6

[4]. Dolev, Shlomi (2000), Self-Stabilization, MIT Press, ISBN 0-262-04178-2

[5]. http://www.knowplace.org/pages/howtos/traffic_shaping_with_linux/network_protocols_discussion_traffic_shaping_strategies.php (visited 12-15-2014)

[6]. https://aitaseller.wordpress.com/2012/09/19/basic-qos-part-1-traffic-policing-and-shaping-on-cisco-ios-router/(visited 12-15-2014)

[7]. Keidar, Idit (2008), "Distributed computing column 32 – The year in review", ACM SIGACT News39 (4): 53–54, doi:10.1145/1466390.1466402Standard ECMA-89: Local Area Networks Token Ring (September 1983)

[8]. Papadimitriou, Christos H. (1994), Computational Complexity, Addison–Wesley, ISBN 0 201-53082-1

[9]. http://msdn.microsoft.com/en-us/library/bb833022.aspx (visited 12-15-2014)

[10]. http://www.opengroup.org/soa/source-book/soa/soa.htm#soa_definition (visited 12-15-2014)

[11]. Velte, Anthony T. (2010). Cloud Computing: A Practical Approach. McGraw Hill. ISBN 978-0-07-162694-1