Volume 2 No. 2, February 2012 ISSN 2223-4985

International Journal of Information and Communication Technology Research

http://www.esjournals.org

Performance of SOA for Information Retrieval System Using Web Services

[1]Thippeswamy.K, [2]Manjaiah.D.H

[1]Department of CSE, JNT University, Anantapur
Andrapradesh , INDIA

[2]Department of Computer Science, Mangalore University, Mangalagangothri,
Mangalore, Karnataka, INDIA

ABSTRACT

Information retrieval systems using web services are increasingly being used on larger databases and by more users. Current system allow users to connect to a multiple database either locally or perhaps on another machine. The resource demands limit the performance of IR systems especially as the size of text collections and the number of applications increase. Service oriented architecture (SOA) computing offers a solution to these problems. Systems based on distributed architectures can use resources more efficiently by spreading work across a network of workstations and by enabling parallel computation. An IR system is an ideal application to distribute across a network of workstations. The amount of information available and the number of people accessing data over networks is rapidly increasing. To meet future demands, a SOA based IR system must provide concurrent, efficient access to multiple databases collections located on remote locations. However, due to the mix of I/O and CPU intensive operations, IR systems present unique problems for distributed system designers. The disparity between I/O and processor speeds exacerbates these problems. Another concern is network traffic, since the amount of data transferred over the network by a SOA based IR system fluctuates considerably.

Keywords: SOA, IR, ESB, Web Services

1. INTRODUCTION

The existing prototype of information retrieval system is based on query, an existing, unified IR system. We have implemented a flexible simulation model to serve as a platform for analyzing performance issues given a wide variety of system parameters and configurations. We validate the accuracy of our simulation model using the prototype. We present a series of experiments that are designed to measure system utilization and identify bottlenecks. We vary numerous system parameters, such as the number of applications, databases and user collections, number of terms per query, response time, and system load to generalize our results for other distributed IR systems models.

1.1 Information Retrieval System using Web Services

We have implemented a prototype of information query retrieval system that is based on SOA; an inference network. The system adopts a variation of the client-server paradigm consisting of a set of applications connected to a set of retrieval databases through a central administration process, the information exchange and query retrieval system as illustrated in Figure 1.

The Enterprise service Bus (ESB) is the user interface to the databases. Clients initiate the work performed in the system by sending commands to the database servers. The commands include the retrieval operations such as query evaluation and document retrieval. The client also determines the set of text collections to search for query operations. An Web services is the retrieval engine. Generic information retrieval system server represents a multiple open database that is selected at start-up time. All retrieval operations take place at the generic information retrieval system server. There are three types of generic information retrieval system servers: query, document, and integrated. A query server only performs query operations, a document server only performs document retrieval operations, and an integrated server performs both document retrieval and query evaluation operations. The ESB is the administrator between the clients and generic information retrieval system servers. Multiple clients and generic information retrieval system servers may be connected to the same connection server.

The connection server is a loosely coupled and lightweight process that manages the work between the user interface and retrieval engine. All messages passed between the clients and generic information retrieval system servers must go through the web services. Messages are placed in a queue until the generic information retrieval system server is available to process new operations. The ESB also maintains a limited amount of state information about current operations and available databases.

International Journal of Information and Communication Technology Research

http://www.esjournals.org

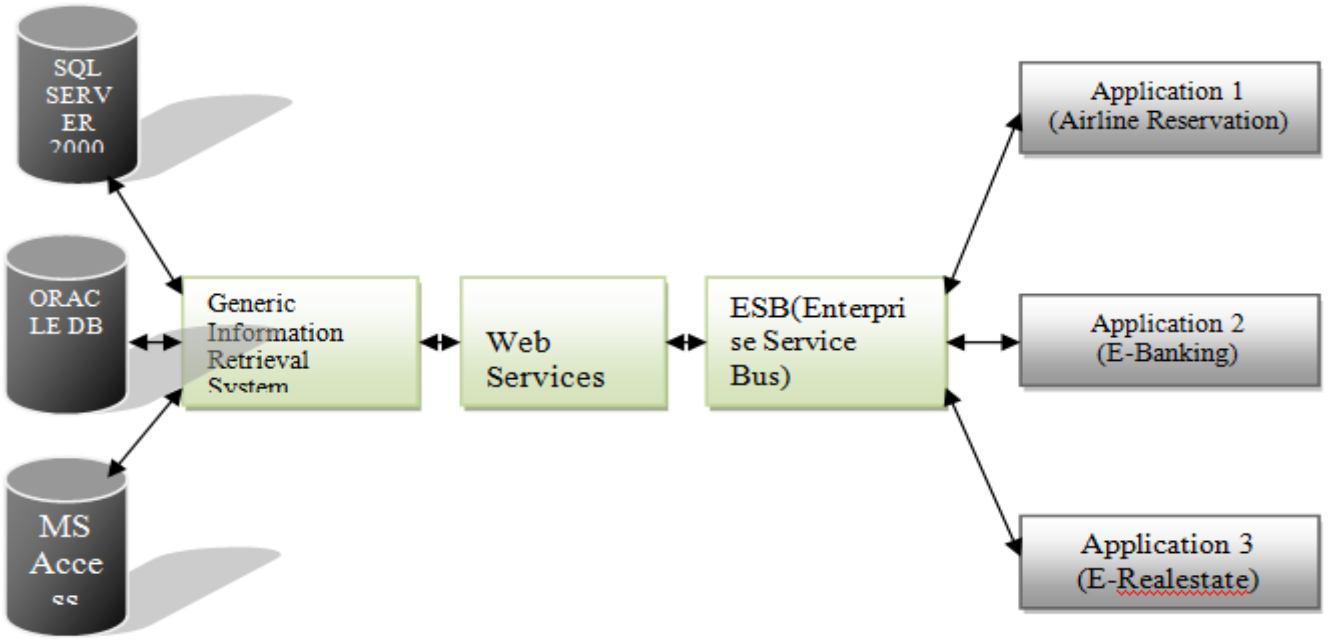

Figure 1: Information Exchange and Query Retrieval System

2. SIMULATION OF INFORMATION RETRIEVAL SYSTEM PARAMETERS

To accurately model an IR system, we analyzed the generic information retrieval system and measured the time the resources used for each operation. Empirical measurements rather than an analytical model drive the activities performed in the simulator. Creating a full analytical model requires too many simplifying assumptions to make an accurate model of a complex information retrieval system.

In this section, we first describe the collections of applications used to obtain resource measurements. We also describe the measurements for each of the activities performed by the prototype. The measurements include:

- Query evaluation time.
- Document retrieval time.
- Summary retrieval time.
- Connection server time.
- Time to merge results.
- Network time.

We obtained measurements using aDECsystem-5000/240 (MIPSR3000 clocked at 40MHz) workstation running Ultrix V4.2A (Rev. 47) with 64MB of memory and 300MB of swap space. To minimize the effects of other users we ran all tests during the evening when the system load was very light.

2.1 Text Collections Used to Obtain Measurements

We examined several different text collections and query sets to obtain measurements used in the simulation. The text collections are:

- Airline Reservation
- E-RealState
- E-Banking

Table 1 lists statistics about the text collections and query sets. Airline Reservation is a large heterogeneous collection of full-text databases and abstracts used in the reservation system. The documents in the collection come from a variety of sources including online reservation, broker reservation center, and authorized counters. The average document size is not large but the sizes of the documents vary from 100 bytes to 1 MB. The Airline Reservation collection includes a set of 50 queries created automatically from reservation topics 51-100. The queries consist of English text and do not contain any structured operators.

The information collection contains the text of the E-Banking. The documents in the collection summarize of the day's events from E-Banking. The size of the documents range from 1K to 700K and the average document size is large. We included this collection since it is a database that is accessible over the Internet and we were able to obtain the user query traces. We examined

the query logs one year data to obtain realistic query statistics. Interestingly, people searching the database typically enter small queries, usually only 1 or 2 words and never more than 8 words. This contrasts with the other query sets that contain long queries. The E-Real estate document collection consists of small abstracts from the Communications of the E-Real estate. This collection is an older test collection consisting of a small number of homogeneous documents. The collection is only 2MB and the average size of the documents is small. The E-Real estate collection also includes a set of 50 English text queries. Again, no structured operators appear in the queries.

Table1: Collection and Query Set Statistics

Collection	Collection Statistics					Query Statistics	
	Size (MB)	No. of Documents	Avg. Doc Size (KB)	No. of Postings	Max Term Frequency	No. of Queries	Avg. No. of Terms
E-Banking	1980	520887	2.9	159571494	654658	80	47.1
E-Real estate	2.8	3904	0.8	185967	5208	70	30.7
Airline Reservation	5.6	45378	18.7	58078407	814849	5141	6

2.2 Query Evaluation Measurements

In information retrieval system, a query operation consists of creating a query network, evaluating the query network on the document network, and ranking the documents that match the query. Since the process is quite complicated, we empirically measured the time required to evaluate a query using information retrieval system rather than creating a complex analytical model. We found that the evaluation time is very strongly related to the number of terms in the query and the frequency of each of the terms. Figure 2 shows a scatter plot which compares query length versus evaluation time for each query in the Airline Reservation query set. The correlation is very high, .96, indicating a strong linear association between query length and query evaluation time. Figure 3 shows the relation between term frequency and evaluation time. Again, the correlation coefficient, .95, indicates a very strong association. To collect this data, we measured evaluation times for individual terms with different term frequencies.

The simulation model contains a distribution of evaluation times based upon the term frequency. We measured the time to evaluate the individual terms with information retrieval system. Given a query that is internally represented as a list of term frequency values, the evaluation time is the sum of the times for evaluating the individual terms in the query. The data in Figure 2 indicates that this simple model reflects the time to perform query evaluation.

We validated the query model used by the simulation against the actual system. Figure 4 shows that the simulation model is close but does not exactly reflect the actual system, especially for large queries. However, there is a strong correlation between the simulation times and actual times. The difference between the times is due to the simple model used by the simulation. The actual retrieval process implemented by information retrieval system is quite complex and difficult to describe using a simple model. Although it is not perfect, the general trend of the model is accurate. (Using a prototype and a model enables us to do this type of validation. We need to examine the actual retrieval process in greater detail to add more features to the simulation model.

The following Table 2 indicates the Term Frequency and Evaluation Time that has been simulated with sufficient set of queries. Evaluation time varies from minimum to maximum.

Table 2: Term Frequency and Evaluation Time

Sl No.	Term Frequency	Evaluation Time
1	10000	0.2
2	20000	0.4
3	30000	0.5
4	40000	0.5
5	50000	0.6
6	50000	0.8
7	55000	0.9
8	30000	1
9	40000	1.4
10	50000	1.5
11	55000	1.7
12	60000	2.3
13	66000	2.5
14	70000	2
15	80000	2
16	90000	2.8

17	100000	3
18	110000	3.5
19	150000	4
20	170000	2
21	100000	2
22	125500	6
23	130000	3
24	134000	2.5
25	150000	5
26	157000	5.4
27	160000	4.3
28	170000	5
29	180000	4
30	185000	4.5
31	190000	3
32	200000	6
33	250000	6.5
34	275000	7
35	280000	7.5
36	290000	6.5
37	300000	8
38	350000	8.5
39	400000	12
40	425000	11
41	450000	10
42	470000	14
43	500000	15
44	600000	18
45	600000	10

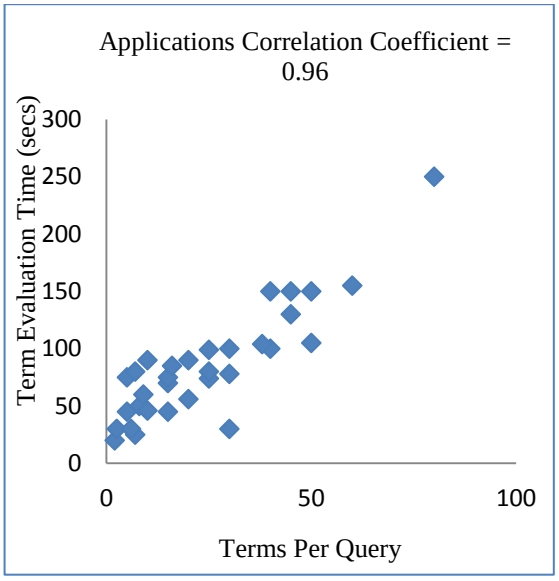

Figure 2: Query Lengths vs. Evaluation Time

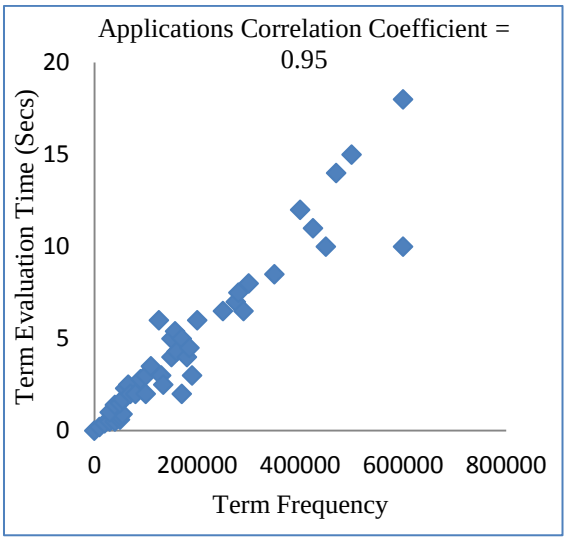

Figure 3: Term Frequency vs. Evaluation Time

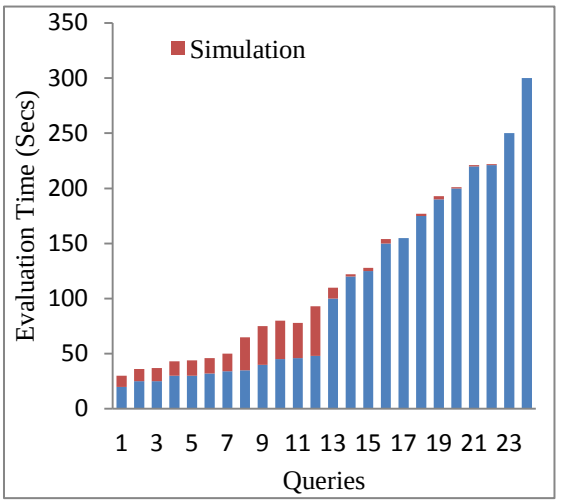

Figure 4: Query Model Validations

3. SIMULATION OF INFORMATION RETRIEVAL SYSTEM CONFIGURATION

A user defines the architecture of the distributed information retrieval system using a simple command language. A configuration file contains the commands and the simulator reads this file at start-up time. Command line options may also be used to define certain aspects of the architecture and these options override any commands in the configuration file. The command line options allow users to easy run the simulator in batch mode by using a single configuration file and changing the command line options for each simulation. For example, if an experiment tests the effect of the number of clients, then a single configuration file defines the architecture and a command

line option specifies the number of clients. The command language supports a rich and flexible set of commands for specifying different distributed architectures. Table 2 describes the set of commands that define the different processes and resources used by the simulation. Each of these commands expects a parameter which is either a constant value or an expression evaluated when the file is read.

Table 3: Definition Commands

Category	Command	Description
Processes	clients conn_servers db_servers	Number of clients Number of connection servers Number of database servers
Resources	Cpus Disks lans	Number of processors Number of disks Number of local area networks
Objects	Machines database	Number of hosts Number of text collections

Table 3 describes the commands that assign properties and attributes to the simulation objects. The parameters depend upon the actual command. The configuration file allows users to provide parameters as constant values or symbolic values evaluated when the simulator reads the file. A user creates system architecture by assigning processes to resources. Also, users may assign various attributes to resources or combine several resources creating a larger object. For examine, a machine object is a combination of the CPU, disk, and network resources. A user creates a complete distributed architecture by defining connections between the different processes.

Table 4: Property Commands

Command	Properties
Conn_server	Machine assigned to connection server
client	Machine assigned to client
DB_server	Machine, database, and type of server
machine	CPU, Disk, and LAN that create a machine
lan	Bandwidth of network in MB/s
database	Size of collection, No. of documents, Avg. size of documents
connect	Define connections between processes
queries	Define attributes of queries

4. EXPERIMENTAL METHODOLOGY OF IR

We describe the experiments we will conduct in order to analyze the performance of our system, identify potential bottlenecks, and to create a scalable system. We first describe our experiments that analyze system utilization. At the end of the section, we briefly discuss several other types of experiments we will perform.

4.1 System Utilization

We designed these initial sets of experiments to measure the utilization of the different resources in the distributed system under varying conditions.

4.2 Fixed Parameters for System Utilization

In the initial experiments, the system architecture closely matches the prototype client-server version of information retrieval system. We must fix several parameters throughout these experiments to match the architecture of prototype system. We fix other parameters to reduce the total number of experiments performed. We will explore the effects of varying these parameters in future experiments.

The fixed parameters are:

- Functionality of processes (e.g., clients, connection server, database servers)
- Connectivity between clients and database servers
- Size of text collections
- Network speed

4.3 Parameters for IR System Utilization

We analyze system utilization by running many experiments which vary several important parameters. The parameters affect system performance and are often variable in actual systems. We will determine the effects of these parameters on system performance by using the simulator. We provide a description of the experiment parameters listed below in the following sections.

- Number of Applications
- Number of Database servers.
- Terms per query.
- Distribution of terms in queries.
- Number of documents that match query.
- Think Rate.
- Documents retrieved.
- Summary information operations.

5. DISTRIBUTION OF TERMS IN QUERIES

Zip documented the widely accepted distribution of term frequencies in text collections based upon empirical measurements. In contrast, the distribution of term frequencies in queries is more difficult to characterize and researchers do not agree on a commonly accepted distribution. Figure 5 shows the query and term frequency distributions for our query sets. The shapes of the distributions for the three query sets are very similar. The difference between the three distributions is due to the size of the text collections.

Comparison and deference's of the different applications is examined in the simulator and obtains the various occurrences over the term query is shown in Table 5.

Table 5: Term frequency and occurrence for different applications

Sl No.	Term Frequency	Occurrences		
		Airline Reservation	E-Real estate	E-Banking
1	1	4	0	0
2	10	6	2	1
3	50	10	3	1
4	100	15	4	1
5	150	18	6	2
6	200	22	8	2
7	250	25	10	2
8	500	30	12	3
9	750	40	14	4
10	1000	60	16	5
11	1500	80	18	6
12	2000	90	20	8
13	3000	90	25	12
14	6000	100	33	16
15	8000	100	40	20
16	10000	100	50	25
17	15000	100	60	35
18	50000	100	70	45
19	100000	100	80	60
20	150000	100	90	80
21	1000000	100	95	90
22	1500000	100	95	95

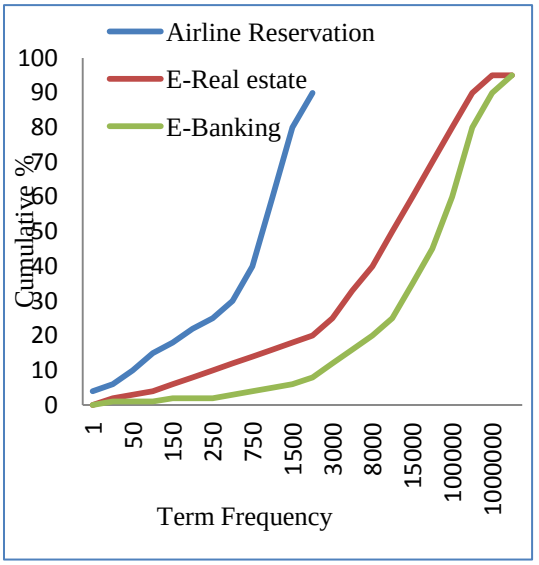

Figure 5: Query Term Frequency Distributions

6. DISCUSSIONS

By reviewing the results and analysis of all the above mentioned existing models, it is found that SOA outperforms all the existing models and proposed models. Hence, it is conclude that the SOA can be used in the design and development of information retrieval system.

7. CONCLUSION

We present the implementation of a prototype information retrieval system. The prototype is based on SOA, an existing and effective standalone IR system. We developed a detailed simulation model to test the performance of the distributed system under varying parameters and configurations. The simulator provides an easy and flexible platform for quickly performing different experiments in a controlled environment. To accurately model the actual system, the simulator uses many measurements obtained from the prototype system.

REFERENCES

[1] Kevin J. M, Web Services: What's Real and What's Not, IEEE Computer Society, 1520-9202, 2005, 14-2.

[2] Kunal Verma, Amit Sheth, Semantically Annotating a Web Service, IEEE Computer Society, 1089-7801, 2007, 83-85.

[3] David Martin, John Domingue, Semantic Web Services, Part 1, IEEE Intelligent Systems, 1541-1672, 2007, 13-17.

[4] Michael A. Casey, Remco Veltkamp, Masataka Goto, Marc Leman,Christophe Rhodes, and Malcolm Slaney, Content-BasedMusic Information Retrieval: Current Directions and Future Challenges, Proceedings of the IEEE, Vol. 96, 2008, 668-696.

[5] Fabrizio Lamberti, Andrea Sanna, and Claudio Demartini, A Relation-Based Page rank Algorithm for Semantic Web Search Engines, IEEE transactions on knowledge and data engineering, VOL. 21, 2009, 123-136.

[6] Liang-Jie Zhang, A Services University, IEEE Computer Society, 1520-9202, 2009, 60-62.

[7] Carolina Fortuna And Mihael Mohorcic, Dynamic Composition Of Services For End-To-End Information Transport, IEEE Wireless Communications, 1536-1284, 2009, 56-62.

[8] Dimitrios Skoutas, Dimitris Sacharidis, Alkis Simitsis, Ranking and Clustering Web Services Using Eria Multicrit Dominance Relationships, IEEE Transactions On Services Computing, VOL. 3, 2010, 163-177.

[9] Javier Gobernado, Carlos Baladrón, and Javier M. Aguiar, Alejandro Cadenas, Management of Service Sessions in an NGN-SOA Execution Environment, IEEE Magazine Communications, 0163-6804, 2010, 103-109.

[10] Tristan Lavarack1 and Marijke Coetzee2, Considering web services security policy compatibility, IEEE, 978-1-4244-5495-2, 2010.

ACKNOWLEDGEMENT

I would like to thank the Department of Information Science & Engineering, R.L.Jalappa Institute of Technology, Department of Computer Science, Mangalore University, Mangalore for providing support to conduct this research work.

AUTHORS

Mr.K.Thippeswamy has received his Bachelor of Engineering Degree in computer science and engineering from University B.D.T College of Engineering, Davanagere, Karnataka(S), India(C), which is affiliated to Kuvempu University, Shimoga in the year 1998 and Master of Engineering in CS & Engineering from Bangalore University, Bangalore, Karnataka(S), India (C) in the year 2004. He is currently pursuing his doctoral program in Javaharlal Neharu Technolgical Univeristy Ananthpur, Andrapradesh, on entitled "Design and Development of SOA for information retrieval using web services" .
He is currently working as Professor, Dept. of Information Science & Engineering, R.L.Jalappa Institute of Technology. Kodiehalli, Doddaballapura, Bangalore rural district, he is authored 10 research papers in National and International Conferences in Distributed Data mining & Data Warehousing
He is Life member of CSI, ISTE, IAENG

Dr.MANJAIAH D.H. is currently Professor and Chairman of the Dept. of Computer Science., Mangalore University, and Mangalore. He is also the BoE and BoS Member of all Universities of Karnataka and other reputed universities in India. He received PhD degree from University of Mangalore, M.Tech. from NITK, Surathkal and B.E. from Mysore University. Dr.Manjaiah D.H has more than 15 - years extensive academic, Industry and Research experience. He has worked at many technical bodies like CSI [AM IND 00002429], ISTE [LM - 24985], ACS, IAENG, WASET, IACSIT and ISOC.
He has authored more than - 50 research papers in International / National reputed journals and conferences. He is the recipient of the several talks for his area of interest in many public occasions. He had written Kannada text book, with an entitled, "COMPUTER PARICHAYA", for the benefits of all teaching and Students Community of Karnataka. Dr.Manjaiah D.H 's areas interest are Advanced Computer Networking, Mobile / Wireless Communication, Wireless Sensor Networks, Operations Research, E-commerce, Internet Technology and Web Programming.
He is the expert committee member of various technical bodies like AICTE, various technical Institutions and Universities in INDIA. He had been actively involving in chairing technical sessions of various International & National Conference and reviewer of the Journals. He is working with Major Research project on " Design Tool for IPv6 Mobility for 4G – Networks " , around Rs.12 lakhs worth funded by UGC, New Delhi from year 2009 -20 12.