FAULT TOLERANT SYSTEM IN WEB SERVICES

Suhaila Mohd Zain
Department of Information Technology and Communication,
Kuala Terengganu Polytechnic
suhaila_mdzain@pkkt.edu.my

ABSTRACT

The web technology has been increasingly used for critical applications and services. However quality of services (QoS) must remain in priority. Qualities such as high availability, performance, reliability and security have been an issue. Aiming to solve this limitation, this research will proposed an architecture providing dependability for web services in an e-commerce. The architecture is designed using NCAPS mechanism which apply replication and redundancy components mechanism for server side execution. All the components involved will be a renew version of previous work done. Firstly the client server physical design is proposed and then the detailed component in the structure of server is suggested. The discussion is divided into several side which includes network, middleware, duplex manager, and linking. Then the strength and weakness of the proposed architecture is outlined.

Keywords: Fault tolerance, web services, QoS.

1. INTRODUCTION

Uses of networked computers are becoming great things to improve the complexity of human activities nowadays. Many applications have arisen in accordance with the technologies advancement and some of it requires non-stop accessibility which means it still can function although there exist an irregular error or hindered by very short delays.

Web services as can be defined any services available on the Internet commonly has a set of protocols and standards adopted for service discovery, interface description and message exchange (Souza & Siqueira, 2008). Designing a fault tolerant system requires finding a way to prevent the logical fault that arises from physical failures from causing an error. A fault can be caused by the physical failure, an inadequacy in the design, an environment influence or operation of the system (Siewiorek, 1991).

Since its almost impossible to remove software faults, fault-tolerance is an essential approach to building highly reliable systems. Although developing redundant software components for tolerating faults is too expensive, the cost of developing multiple component versions is greatly reduce in modern era of service-oriented computing (Zheng & Lyu). Replication
mechanism is commonly used technique for fault tolerance system (Reiser, Kapitza, Domaschka, & Hauck, 2006).

One of the most obvious is in the web services application, where E-Commerce has played a major concern in contributing the development of Fault-Tolerant Web Server. This is due to some inconvenience such as long response times or temporary system downtime that occurred in web services and will effect crucial online transaction such as in E-Commerce. Therefore the service need to overcome the problems stated in order to process quick HTTP requests, transfer important data, and to use any built-in program as to support the use of high-performance machines.

This paper will focus on the HP’s Tandem Non-Stop Clusters Application Protection System (NCAPS), a mechanism involving protective software redundancy to permit correct operation after the occurrence of a specified fault (Laranjeira, 1998). It was reported that Tandem are originally created under Tandem Computers, Inc (1974) to support commercial transaction processing in high availability computer systems and the technology development is still in progress.

NCAPS is a fault tolerant system for high availability systems that is being used to improve the continuity of programs operation on a Unix computer environment. It fundamental based on primary/backup concept where two instance of an application are executed concurrently. In state of normal primary application service, the back-up will remain in idle, ready to takeover if failure occurred to primary instance. It will recover the application if anything happen to hardware, software or operating system, such as failure with total failover is expected to be approximately 10 seconds. Thus give better result compared to existing high availability solutions which provide recovery times in minutes. This is because they assigned the task to another node in case of hardware or operating system failure meaning that, full initialization from the start need to be done making it as original. Details of NCAPS functionality and techniques employed are explained in further sections.

2. LITERATURE REVIEW

Research into fault-tolerant control has attracted many investigators mainly for safety reason such as in hazardous chemical plants, the control of nuclear power plant reactors, space craft or aircraft system. However, research into fault tolerant control is largely motivated by the control problems encountered in aircraft system design. The goal is to provide a "self-repairing" capability to enable the pilot to land the aircraft safely in the event of a serious fault. The interest has been specially stimulated since two commercial aircraft accidents in the late 1970's (Patton, 1997). As time goes by, the research is then turn into fault-tolerant system as majority of millions web sites developed offers various web services for almost everyone in the world. The load on these sites are usually small and the services offered are based on background task on general-purpose workstation therefore concerned for quality of services (QoS) generally is not primary (Ingham, Panzieri, & Shrivastava, 2000). Ingham et. all stated that QoS most important factors is availability in web services. Their research focus on issues in supporting high-volume and high-reliability Web Services which includes the hardware, software and network-based approaches regarding fault tolerance. Many techniques and method have been proposed and applied to various environments and for many reason. For example, Chen and Dayal focus on managing database transaction for
transaction failures in hierarchies (Chen & Dayal, 1999). Their techniques consider transaction on in blocks. Such that the transaction has parents and child concept. A block can itself be another hierarchy. Failure recovery is achieved by taking advantage of the ability to trace any transaction in a process in its corresponding history tree, which represents the execution history of the transaction.

Another research focused on network filesystem (NFS) data serving, where server system and mirroring internally is set in redundantly. A dedicated error detection control subsystem is provided typically at the CPU base level to identify any operational problems in the programs executed synchronously by the redundant CPUs. A malfunctioning CPU can thus be identified and functionally eliminated from further participation in providing computer system services. The remaining CPU will then continue to provide all pending services.

Load balancing technique in web cluster is another solution proposed for Internet server system in order to achieve stable and fault tolerant service to clients (Myung-Sup, Mi-Jeong, & Hong, 2002). A load cluster management system (LCMS) is designed based on Simple Network Management Protocol (SNMP) and Web technology. Meanwhile, in another research, dependability and security are highlighted in multi-cloud computing. A model is designed for increasing system reliability and security dimensions which cover data integrity, data confidentiality (AlZain, Soh, & Pardede, 2013).

Load balancing technique in web cluster is another solution proposed for Internet server system in order to achieve stable and fault tolerant service to clients (Myung-Sup, Mi-Jeong, & Hong, 2002). A load cluster management system (LCMS) is designed based on Simple Network Management Protocol (SNMP) and Web technology. Meanwhile, in another research, dependability and security are highlighted in multi-cloud computing. A model is designed for increasing system reliability and security dimensions which cover data integrity, data confidentiality (AlZain, Soh, & Pardede, 2013).

Recent research involves the manipulation of server reboot in which there are machining system supported by a repair facility and mixed standby units operating system (Jain & Meena, 2017). Another research, applied rerouting for link/path failures (Subedi, Nguyen, & Cheriet, 2017). This technique is designed for bandwidth reservation to improve transfers of data from data centers which are distributed geographically.

3. METHODOLOGY

This section will further discuss the proposed architecture in banking transaction. The concept of bank transaction applied is considered from Zhao in his research (Zhao, 2007). Meanwhile the mechanism for model proposed is a renew version of previous work done by Laranjeira (Laranjeira, 1998). Techniques such as transaction processing, active replication and checkpointing as employed in NT-SwiFT had been also referred (Liang et al., 2004).

3.1 DESIGN GOALS

My proposed fault tolerant system architecture will be implemented in a distributed environment consists of advanced duplicate nodes in a cluster for banking web services. Analyzing the situation and problems of Web Services technology is important in creating distributed systems in high availability attempt before designing it goals. There are many reasons to relate of online system downtime because it resides on top of dynamic platform which include database server, software system, computer technology and network facility, means failure in any part, will affect the systems. Problems can due to service, software processing, devices integration or network design. Hence, fault tolerant techniques proposed must be applied with goals to cater all the possibilities as mention above. The goals are:
- Try to tolerate service failures with degraded functions in order to fulfill customer request thus met high-availability requirements which will enhance quality of service.
- Minimize response time to the fastest achievable result in present of outnumber customer accessing the system concurrently with aim time of waiting no longer than eight seconds.
- Attempted to automatically retry of failed-transaction or execute redundant tries to avoid losing any transaction including data loss in transfer process thus maintain data integrity and consistency.
- Guarantee that the message sent between individuals are in same domain with correct conclusion.

3.2 OVERVIEW OF PRESENT SYSTEM ARCHITECTURE

This subsection will discuss on the overall system architecture of NCAPS (Laranjeira, 1998). The architecture can be viewed using modularity concept introduced in System Reference Model which consisted of five main components. The model has four layers, communication links, communication manager, system executive and application process. Their elements and function are simplified in Table 1 below.

Table 1: NCAPS System Reference Model (Laranjeira, 1998)

<table>
<thead>
<tr>
<th>LAYERS</th>
<th>COMPONENTS INVOLVED</th>
</tr>
</thead>
<tbody>
<tr>
<td>COMMUNICATION LINKS</td>
<td>Open Fault Tolerant Library (OftLib)</td>
</tr>
<tr>
<td></td>
<td>- provide functions as listed:</td>
</tr>
<tr>
<td></td>
<td>- Application Management (AM)</td>
</tr>
<tr>
<td></td>
<td>- Checkpointing (CKP)</td>
</tr>
<tr>
<td></td>
<td>- Hang Detection Services (HDS)</td>
</tr>
<tr>
<td></td>
<td>- File Descriptor Storage and retrieval (FDS)</td>
</tr>
<tr>
<td>COMMUNICATION MANAGER</td>
<td>Process Pairs Managers (PPM)</td>
</tr>
<tr>
<td></td>
<td>- being the subscribers to NSM</td>
</tr>
<tr>
<td></td>
<td>- exchange heartbeats with another PPM</td>
</tr>
<tr>
<td></td>
<td>- contains:</td>
</tr>
<tr>
<td></td>
<td>- Application Administration (AAD)</td>
</tr>
<tr>
<td></td>
<td>- Inter-Application Communication (IAC)</td>
</tr>
<tr>
<td></td>
<td>- Application State Model (ASM)</td>
</tr>
<tr>
<td>SYSTEM EXECUTIVE</td>
<td>Keep Alive (KpA)</td>
</tr>
<tr>
<td></td>
<td>- spawns and watches the PPM, NSM</td>
</tr>
<tr>
<td>[EMBEDDED SYSTEM]</td>
<td>Node Status Monitor (NSM)</td>
</tr>
<tr>
<td></td>
<td>- exchange heartbeats with another NSM</td>
</tr>
<tr>
<td>APPLICATION PROCESS</td>
<td>Command Line Interface (CLI)</td>
</tr>
<tr>
<td></td>
<td>- it is to manually change the system</td>
</tr>
</tbody>
</table>
- consisted of commands: query, swap, init, primary, inhibit, uninhibit, enable, disable

Web program
- have registered process

3. 3 DESCRIPTION OF PROPOSED FAULT-TOLERANT ARCHITECTURE

This design approach focus to counter backup process for each running processes in different node in order to tolerate any failure in logical machine, therefore to contain physical machine failure. This is due to fault avoidance techniques have been applied in the first place. Main process refers to process pairs of servers in two machine cluster. Figure 1 illustrate it with server, S1, S2 in one site in Australia and while other two server S3, S4 represent other location such as in US. Another important process is checkpointing and states monitoring between the servers and components introduced. There also data such as customer transactions, configuration setting and process checkpoint that are kept in file because they are needed in reconfiguration and recovery method. The system work such that one node acts as a primary server and another as backup. The same design functionality then is applied to other location in which replicates is required. The discussion will look on different part of design implementation separately.

Figure 1: The Client/Server Physical Design
3.3.1 NETWORK

Here there will be two kinds of router they are router in the internet, and router member on site. The many routers in the internet will make sure any request from bank customer can go to any available high bandwidth communication path, this should overcome connection problem. In other words the redirect routing will make the network available all the time.

Then there are redundant on-site routers, this is to tolerate faults in data link, so the request will be routed to the destination on a duplicate route. The redundant connections allow for alternative paths if a device or links fails. The routers will check each other by sending messages and this will keep them informed. In router itself will consisted of firewall setup, this is to further increased security aspect of the banking transaction. Then the gateway will distribute the load accordingly using DNS round robin manner. The technique is to tune DNS entry with small TTL value such that only limited caching effect occurs.

It then contacts the primary server to work on processing transaction. For example the server will accept information from customer to first login into the banks online system before he can transfer money to another person within the same bank. If the server fails another server performing the same functions should be able to pick up the work until repairs are made. This is done as web servers in each node are made to monitor each other by responds to periodically message sent to indicate it states. Consequently hardware approach for solution should tolerate failure in single node.

In this process, the Load Manager (LM) in position 2, introduced will save processes as checkpoint in a log file with some information about routing such as customer host address. It also responsible to gather other information from RPC and RDA process and save them in the file. Basic information includes host name, IP address, CPU type, memory size. LM should communicate to DM informing the situation of networking problems and vice versa. So fails node will be configured by replacing it with a backup node using information read from the log file.
3.3.2 MIDDLEWARE

Simple Object Access Protocol (SOAP) package, a middleware services which reside in client server application is chosen to provide standard for building applications in distributed objects. According to Fang et. al, SOAP is recognized as more promising middleware for electronic commerce applications among other leading candidates such as CORBA (Fang, Liang, Lin, & Lin, 2007). Its main features are to invoked appropriate method from the package to serialize the services between client and server without considering the SOAP vendor, platform or language been used. Therefore, the feature itself has proven to tolerate any failure transaction on distributed objects. This offer an atomic transaction in which failures of any case will guarantee of complete transaction or unoccupied transaction. Data continuity in this regard is thus consistent to be used for other transaction. For example different customers with different transactions, they are treated separately, likewise to isolate the customer relationship. In this sense data error in Customer1 didn’t affect Customer2. Hence this scenario covered the concept of process group in updating transaction in distributed environment.

3.3.3 DUPLEX MANAGER (DM)

This main component will be located at position 3, which act as heart of the process pairs in the system, it will send message to another DM in spare node informing it states. The approach uses self-checking computers where each node is compared to provide error detection. These computers run synchronously so that if one fails, the other component can continue running without delay.

As a data copy server, it receives the data from primary database and transmits it to the backup database. Connected with the Data Extractor (DE) in a directory which shows what data is to be extracted, where it is to be sent, how frequent it is to be sent and whether the sent data is to be applied immediately or delayed.

In addition, DM proposed will embrace GUI-based facility and comprehensive display to monitor the event and status of various components and communicating message between nodes and replicas. It will enable system administrator to define, generate, initialize and customize the various replication server components and also setting the primary and secondary database for the replicated data items. It also allows specification of parameters to govern the operation of various components with configuration file provided.

3.3.4 LINKING

Triple bus line in position 4, is provided in order to connect from server to server and also database communication on a site. The redundant of SCSI line bus will make sure one link fails will have backup to continue forwarding data of message passing and process pairs. The process of detecting problem in bus line is handled and managed by LM.
3.3.5 DATABASE

Server managed two complete copies of databases. It will process read/write requests from customer. In the proposed environment, banking activities require read and write access to database. Viewing account transaction in a particular month just employed reading the database while transferring money to other account will require updates to account balance. Such activity will write access to database. It starts with primary server access its local database image; this is done with uncommitted writes process. In the meantime it will contact the backup server in the cluster as well as servers distributed over the net. As the customer done his transaction, then the fixed committed order writes will propagated to backup database and also its replicates.

However the commit action may cause inconsistency when some replicated server system is down. In the situation depicted, component named Data Extractor is responsible to capture needed data from the primary database, means data that have changes. Techniques used involved capture the data changes from the logs. The capture program continuously monitors the logs and extracts the needed changes. The incremental updates are suitable for large database that require frequent synchronizations. In the meantime, Data Synchronizer located in position 5 is responsible for transmission of captured data from the primary database to the secondary. However in case of database corrupt due to hardware failure, which means permanent fault, DM in the system will reach memory to activate the backup database. Strict consistency is maintained within two machine cluster but lazy consistency is applied to the replicas.

4. RESULT

4.1 DISCUSSION OF NCAPS TECHNOLOGY INCORPORATED INTO THE PROPOSED DESIGN

- **Redundancy** concepts in NCAPS are applied in the proposed design. However it is expanded in internet environment. The additional devices include routers and gateways and also the middleware to successfully reach agreement between protocols of customer and the system. Other obvious redundant are server, components within each server and communication between servers, where each exchange message is to inform their states.

- Configuration file for the system administrator and to be used by DM are similar to configuration files used by PPM.

- The LM in proposed design also save checkpoint of every process in a log file. Further it store information in transaction process. The information is important in order to rollback if failure occurs. This step is the same with checkpointing functionality by OftLib in NCAPS.

- DM functions to reconfigure the system in case of failure by activating the spare as primary. This can be set automatically by the system or done manually by system
administrator for example in maintenance attempt, so the system will be available at all cost. The component roles have similarity with PPM in NCAPS.

4.2 STRENGTHS AND WEAKNESS OF FAULT-TOLERANT IN THE PROPOSED ARCHITECTURE.

By referring to design goals mentioned, description of strengths and weakness of fault-tolerant approaches in the proposed design can be listed as the following:

- The proposed design will give the highest availability, 24/7 because it has duplicated server with duplicated database on a site and also replicated in the Internet which is distributed geographically.

- The design caters all the ACID properties to create dependable web services in banking application by configuring two backend database in primary server and backup server.

- However the design is very complex and it need very high performance computer in order to communicate distributed server with many transaction of many customer concurrently. The data and message must be sent and updated in nanosecond. Very high cost is needed to setup the best and advanced computers to achieve the goals.

- System administrator also must have high understanding and knowledge on the process happen in the running system in able to administer the whole process.

- In terms of data integrity, the main challenge in data replication is to keep the replicated data consistent with each other.

5. CONCLUSION

This paper presents software architecture for fault tolerant banking web services. It provides explanation of components involved to achieve availability and dependability services. It specifically suited recent standards and technologies by employing SOAP middleware concurrently with object group transaction primarily for server and database management. In other words redundancy and technology applied has shown a complete set to tolerate any faults in any levels of execution by allowing the possibility of allocating duplex in other networks setting which provides a higher degree of availability. This is different with the NCAPS software architecture, which is not specific to any application or specific field. The original work also does not include security dimension.

The proposed architecture also succeeds in serving better confinement of network traffic within the internet. Communication which happened between primary server and other replicated primary server within different location has minimize traffic in the internet, thus reduce the load of transferring messages. Therefore reduce response times, which is an important factor for distributed applications. The overall does provide 24/7 QoS as needed.
However, hardware redundancy brings a number of consequences: increase in weight, size, power consumption, cost, as well as time to design, and test. Therefore, the design still need improvement as to determine the best way to incorporate redundancy into a system.

6. FUTURE WORK

Few issues related to the fault tolerant system have been identified, which need to be address in order to enhance the architecture proposed, such as to qualify the load balancing mechanism between replicas which incurred minimum time response. The nearest server with light burden must be recognized to provide excellent performance to the customer. Thus will improve the QoS with added functionality that comply with advance e-commerce functionality. In distributed environment which also consists of redundant servers on site, dynamic priority setting should be set in order to give precedence to whom should accept the message. Message should just be simplified using code instead of normal message. Therefore security measure from this execution can be upgraded.

7. REFERENCES


