LOAD BALANCING TECHNIQUES FOR RELEASE 11i AND RELEASE 12 E-BUSINESS ENVIRONMENTS

Venkat Perumal

IT Convergence

Introduction

Any application server based on a certain CPU, memory and other configurations has a limit. Once the server reaches its capacity, performance is impacted. In order to provide better performance, a server can be replaced by a more efficient server to handle the load. A better option is to implement cost effective solutions such as load balancing to share the load with additional servers of similar configurations. Load balancing can be achieved based on response time, least connections, least bandwidth or response time. The load can be balanced at different levels such as server, network or operating system. One of the primary goals of load balancing is to distribute the load among many servers, so that the overall throughput of the system is improved.

The purpose of this paper is to explore Oracle's E-Business suite 11i load balancing options. Although there are number of ways to achieve load balancing, this paper explores five primary load balancing techniques associated with this application.

The five load balancing techniques are:
1. Domain Name Server Load Balancing
2. HTTP Server Load Balancing
3. Forms Metric Load Balancing
4. Apache Jserv Load Balancing
5. Parallel Concurrent Processing (PCP) Load Balancing

The first four techniques are associated with middle tier, and Parallel Concurrent Processing (PCP) focuses on concurrent manager tier.

Domain Name Server Load Balancing

As the name indicates a Domain Name Server (DNS) is utilized to load balance Oracle's E-Business suite 11i applications. Domain Name Server translates a domain name into an IP address. In this technique, single domain name is mapped tot multiple IP addresses of web servers. This technique is widely used to load balance web servers and is used by web services including Google and Yahoo.

Depending upon the number of users, the technique used in DNS load balancing varies. For an 11i application environment, a DNS technique called Round Robin Domain Name Service (RRDNS) is widely used, where the user requests are forwarded on a cyclical fashion.

As shown in Diagram 1, when a user requests the URL, DNS first translates the domain name to IP address and routes back the request to webnode1, the second user request onto webnode2 and the third user request onto webnode3, and thus in a cyclical fashion the user load is balanced among the web servers. This is quite inexpensive and easy to setup.

Oracle's E-Business suite 11i applications' URL is based on hostname and domain name, for example http://itcerp.itconvergence.com, where itcerp is the hostname where the web server is installed. Since there is more than one web server involved and the URL remains static for all users, the virtual host naming technique is used to map the different web servers to an assigned virtual host. A virtual host name is mapped to multiple IP addresses in such a way that when the URL is accessed, it is translated and the request is passed on to the particular web server.

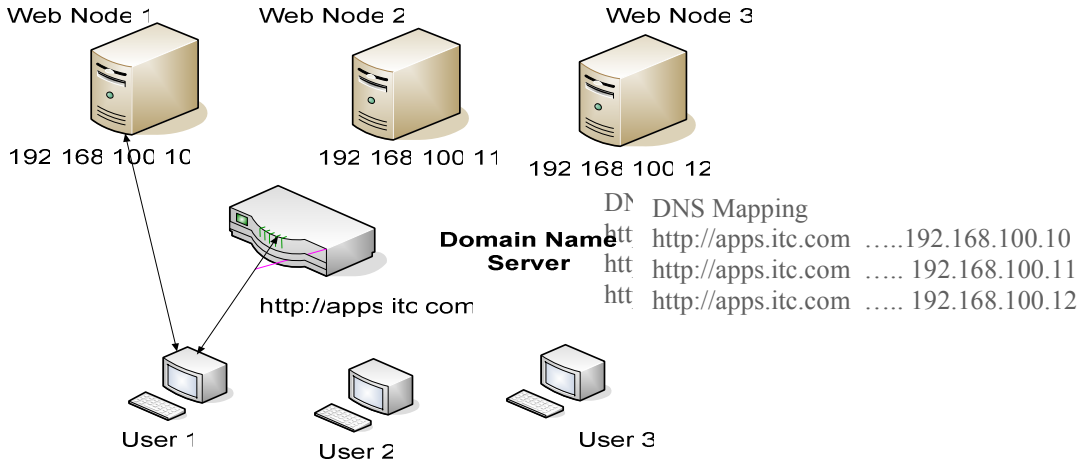

Diagram 1: Domain Name Server Load Balancing – Architecture

DNS Load Balancing Configuration
This configuration does not require additional patches, but virtual host name and DNS configuration are required. The following autoconfig parameters should be configured with virtual host name:
1. S_LOGIN_URL
2. FORMS60_MAPPING
3. PROXY SERVER
4. WEBSERVER ENTRY

Scenario
Medium range business clients with 300 to 500 concurrent users would be ideal for DNS load balancing. In most cases the web server and forms server will be installed on the same host, and DNS load balancing can be used to load balance both self-service and forms requests.

Advantages and Disadvantages of DNS
1. DNS load balancing technique is primarily used to load balance multiple web servers.
2. High availability and failover is not guaranteed.
3. Does not consider "actual" server load - balances number of users on each server, it does not necessarily balance the server load.
4. In the event of server failure, DNS needs to be reconfigured. This may be a bigger problem when removing a node then when adding one. When a node is dropped, a user may be trying to access a non-existing server.

HTTP Server Load Balancing

In HTTP server load balancing, an HTTP load balancer is used to redirect requests onto multiple web servers. As shown in Diagram 2, there are two web nodes. When a user request arrives the HTTP load balancer redirects the requests to one of the least loaded web nodes. The HTTP load balancer is the single point of entry for all the user requests and redirects the requests onto the web application running on either of the web nodes. Any number of web servers can be included in this architecture. The HTTP load balancer may be hardware or software based.

In order to configure the HTTP load balancer with Oracle's E-Business suite 11i, there are specific patches and configurations on the application techstack that need to be performed.

Session Persistence
One of the important considerations for HTTP load balancer is session persistence. There are two types of applications with respect to session persistence – stateful and stateless. A stateful application maintains session state information within its runtime environment between successive client calls, where as a stateless application maintains no such information within its environment. A stateless application may persist state information in a common store such as a database or in the client browser. The Oracle E-Business suite is a stateful application, so while configuring the HTTP load balancer, session persistence should be configured, such that all client requests from one client session are sent to the same web node. Otherwise, there is a high possibility that user requests may be not be served as expected.

Some HTTP load balancers use a sticky round robin algorithm to load balance incoming HTTP and HTTPS requests. When a new HTTP request is sent to the load balancer, it is forwarded to an application server instance based on a simple round robin scheme. Subsequently, this request is "stuck" to this particular application's server instance. From the sticky information, the load balancer first determines the instance to which the request was previously forwarded. If that instance is found to be alive, the load balancer plug-in forwards the request to the same application server instance. Therefore, all requests for a given session are sent to the same application server instance.

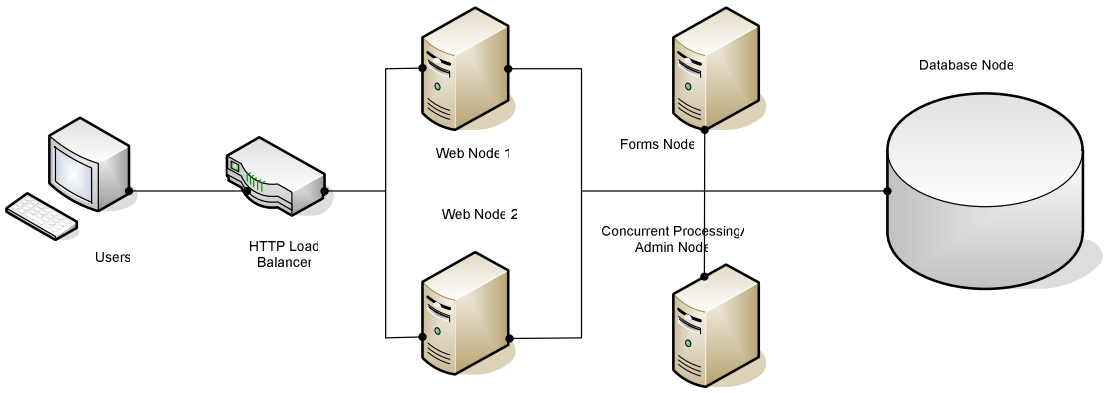

Diagram 2: HTTP Server Load Balancing – Architecture

HTTP Configuration
In order to configure the HTTP load balancer with Oracle's E-Business suite 11i, there are specific patches and configurations on the application techstack that need to be performed. With respect to the application context file, the following parameters should be configured in order to achieve HTTP load balancing:

1. Web entry point Host - HTTP load-balancer machine name
2. Web entry point Domain - HTTP load-balancer domain name
3. Web entry protocol - HTTP load-balancer protocol e.g. "http" or "https"
4. Active Web Port to the value of the HTTP load-balancer's external port
5. Login Page

Depending upon the type of HTTP load balancer, vendor specific additional configuration steps may be required to configure the HTTP load balancer.

Scenario
The best case scenario to implement HTTP load balancer in an E-Business Suite 11i environment is for a high number of self-service application users. If forms server is installed on the same node as the web server, this technique also load balances forms users as well.

Advantages and Disadvantages of the HTTP Load Balancer

1. Sophisticated HTTP load balancer algorithm enables detection of failed servers.
2. Some HTTP load balancers take into account server parameters such as CPU load and memory consumption before redirecting the users request onto a specific web node.
3. Compared to DNS load balancing, HTTP load balancing requires expertise to configure and setup.
4. Depending upon the HTTP load balancer selected, it might be expensive to setup.

Forms Metric Server Load Balancing

Forms metric server load balancing is available since Release 11. This is one of the mature load balancing techniques, which can be achieved by configuring multiple forms servers. Forms server is an application server optimized to deploy Oracle Forms applications in a multi-tiered environment. Forms metric load balancing is implemented based on number of processes running on a load balancer client machine. Load balancing only checks the number of processes but it does not take into account actual resources on the server like CPU load or memory consumed by the forms process. Distinction between primary server and secondary server is a must as the metric server will always be on the primary server, with clients running on secondary servers. A Forms server can also load balanced by configuring forms cgi-bin.

In order to configure Forms metric server load balancing, at least 2 Forms servers are required. One of the Forms servers acts as the primary server and other one as a secondary server.

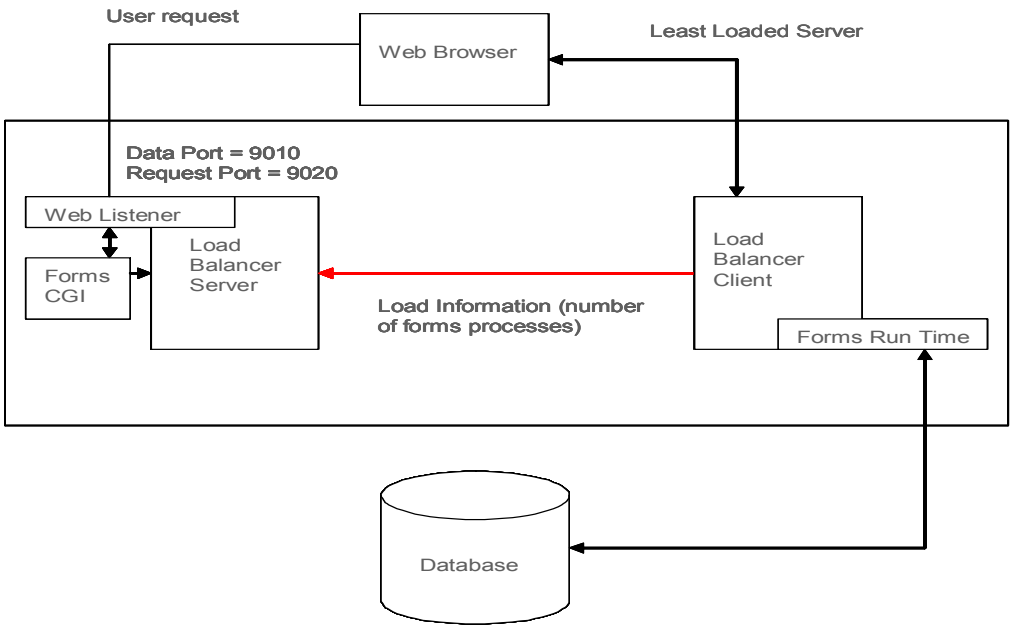

Diagram 3: Forms Server Load Balancing – Architecture

In diagram 3, the load balancer client periodically sends load information to the load balancer server. When user request comes in, the Forms cgi-bin executable requests the load balancer server for the name of the least-loaded system that is available. The Forms cgi-bin executable dynamically creates an HTML page with the name of the least-loaded system on which to run the Forms server, and returns that HTML page to the user's web browser. The web browser then requests the Java applet to be downloaded from the host specified in the HTML page which is the least loaded forms server. The Java applet sends a request to the Forms server requesting for a particular Form

Builder application and the server contacts a Forms server runtime engine. A dedicated runtime engine is allocated for each forms session. Forms run time, through a Net 8 or ODBC connection, contacts the database to fulfill the user request.

Forms Metric Configuration
The following, load balancer server and client parameters are necessary in order to achieve Forms metric server load balancing. These parameters are configured in formsweb.cfg file. Additional configuration in the autoconfig file is also required.
1. Data Port – port number for the load balancer server must match the Data Port values for ALL load balancer clients.
2. Request Port – port number for the load balancer requests.
3. Server Host - parameter should be set to least loaded server name.
4. Data Host – hostname of the primary forms server.

Scenario
Forms metric server load balancing is widely implemented to address core forms load. If there are a high number of core forms users, this technique will be highly effective to counter the load. In some cases, the web server will be part of the primary Forms metric server in that case primary server will handle the requests from the self service application, as well as core forms user.

Advantages and Disadvantages of Forms Metric Server Load Balancer
1. Since load balancing is achieved based on number of forms processes, this technique doesn't take into account OS resources like CPU load and memory.
2. In case of Forms server failure, user request doesn't get processed, thus it doesn't provide high availability.
3. Since forms metric server load balancer is a part of the E-Business Suite 11i, it is quite inexpensive and easy to setup.

Apache Jserv Load Balancing

Apache Jserv configuration is made up of an Apache module called mod_jserv, which runs in the httpd process, and a servlet engine, which runs in a Java process. Mod_jserv functions as a dispatcher, routing each servlet request to a JServ process for execution. The servlet engine runs in its own JVM (Java Virtual Machine) and is solely responsible for parsing the request and generating a response. Multiple JServs can service requests. The HTTP server process and the JServ process communicate using the Apache JServ Protocol (AJP) 1.2.

It is beneficial to spread the servlet application load among multiple JServ processes, especially when the application is run on a multiprocessor server or if the servlets and HTTP server are run on separate nodes. Running multiple Apache JServ processes generally results in higher through put and shorter response time, even on a single-processor host.

As shown in diagram 4, by configuring multiple Web nodes and multiple Java Virtual Machines across different servers, Oracle Application requests are load balanced. In 11i applications, a user request will be processed by one of the following predefined JVM groups.
Core Group - Default group
Forms Group - Forms Servlet request
Discoverer Group - Discoverer 4i request
Web Services Group - XML Gateway, Web Services and SOAP requests

It is common to increase the number of JVM defined per group even in a single-node installation. Error messages like 'java.lang.Out Of Memory Error' indicate depleted JVM, which suggest a need for multiple Java Virtual Machines.

Apache Jserv Configuration
Apache Jserv load balancer configuration requires Oracle 9i Application Server version 1.0.2.2 and techstack specific patches. Apache-Jserv load balancing also requires extensive configuration in the application context file.

The following parameters are relevant to Apache Jserv load balancing and for more details refer to Oracle Metalink document 217368.1.
1. OPROC Manager Port
2. Local Domain Name
3. Multi Web Node
4. OA Core zone

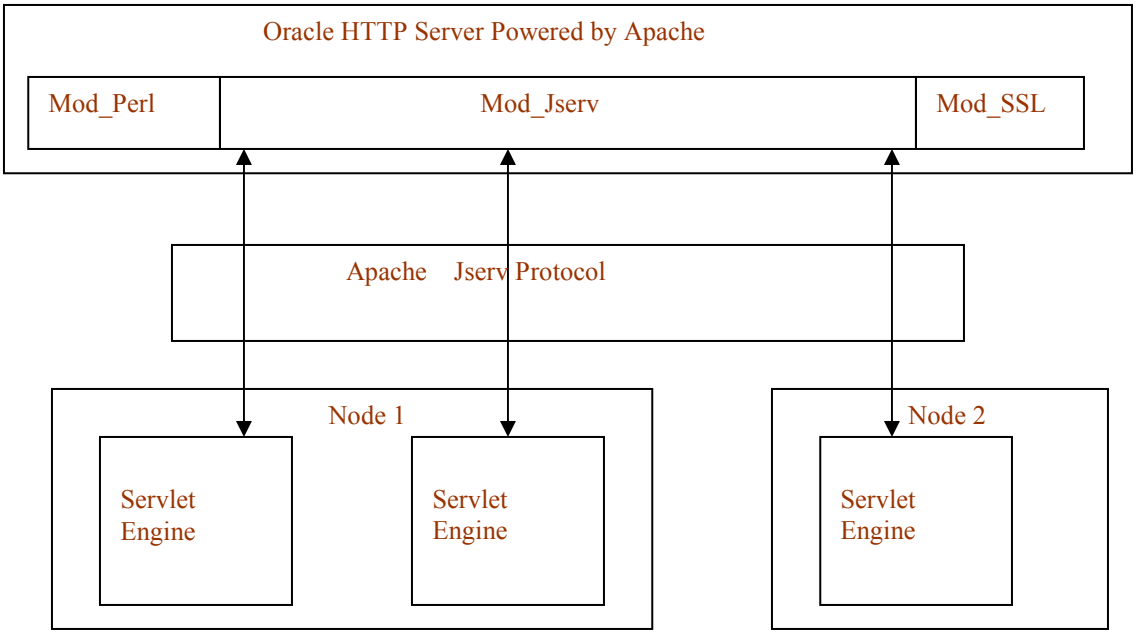

Diagram 4: Apache-Jserv Load Balancing – Architecture

Scenario
Apache Jserv load balancing will be ideal for an E-Business suite 11i environment with a very high number of self service and core forms users.

Advantages and Disadvantages of Apache Jserv Load Balancer
1. Because of multiple Apache Jserv configurations on different servers, it provides increased fault tolerance and increased scalability.
2. Multiple Apache Jserv configurations provide increased through-put for user requests.
3. Requires extensive configuration on the application teckstack when compared to previous load balancing techniques
4. Compared to previous load balancing options, Apache Jserv load balancing requires more expertise to setup and maintain Oracle E-Business 11i environment.

Parallel Concurrent Processing (PCP) Load Balancing

Parallel concurrent processing is the only technique with which concurrent managers can be deployed across multiple nodes. In 11i applications, one Internal Concurrent Manager (ICM) can update the concurrent manager tables. By implementing PCP, which allows a single ICM on one of the nodes, and by distributing other managers across multiple nodes, concurrent manager load balancing is achieved.

In a single-tier configuration, non PCP environment, a node failure will impact concurrent processing operations due to any of these failure conditions. In a multi-node, configuration the impact of any these types of failures will be dependent upon what type of failure is experienced, and how concurrent processing is distributed among the nodes in the configuration. Parallel Concurrent Processing provides seamless failover for a concurrent processing environment in the event that any of these types of failures takes place.

The Internal Monitor (IM) monitors the Internal Concurrent Manager, and restarts any failed ICM on the local node. During a node failure in a PCP environment the IM will restart the ICM on a surviving node (multiple ICM's may be started on multiple nodes, but only the first ICM started will remain active, all others will be terminated). There should be an Internal Monitor defined on each node where the ICM may migrate.

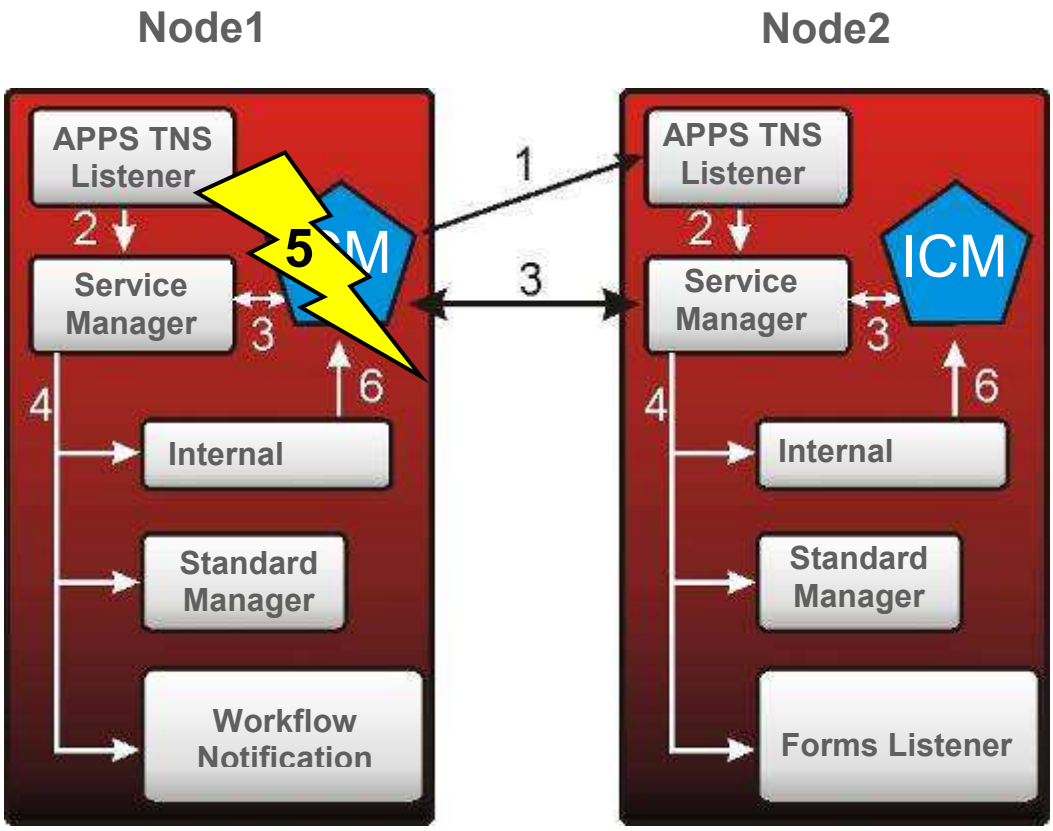

Diagram 5: Concurrent Manager Load Balancing – Architecture

PCP uses an operating system ping command and checks database dictionary views for the availability of Internal Concurrent Manager and also to check for any failures. If a failure occurs on the primary node, all concurrent queues will be migrated onto the secondary nodes. Once the primary node recovers, fail back occurs. During the fail over scenario, concurrent requests running on the failed node will be terminated, so the terminated requests must be rescheduled. Rescheduling is a manual process that should be done during a failover.

PCP Configuration
At least two nodes are required to setup Parallel Concurrent Processing. Node configuration as primary and secondary is setup in the concurrent manager definition screen under system administrator responsibility.

In Oracle 11i application file system, listener and PCP specific environment setups must be configured. One important prerequisite for PCP setup is that, at OS level, both the nodes should be able to communicate via rsh.

Advantages and Disadvantages of Parallel Concurrent Processing

1. High performance – PCP provides the ability to run concurrent processes on multiple nodes to improve concurrent processing throughput.
2. Specialization rules can be written in Oracle to execute particular managers on a specific node. For example PO Document manager can be deployed to a certain node.
3. Fault Tolerance – PCP provides the ability to continue running concurrent processes on available nodes even when one or more nodes fail.
4. During fail over, concurrent requests gets terminated, and these requests should be manually rescheduled.

Conclusion

Oracle E-Business load balancing options can be effectively used to improve performance and scalability. One or more techniques can also be combined to design a robust E-Business environment. Most of the load balancing options discussed in this paper is certified by Oracle.

Reference

1. Metalink Note:217368.1 Advanced Configurations and Topologies for Enterprise Deployments of E-Business Suite 11i.
2. Metalink Note 279956.1 Oracle E-Business Suite Release 11i with 9i RAC: Installation and Configuration using AutoConfig.
3. Oracle Part No A73071-01 Forms Server Release 6i.
4. Metalink Note 148155.1 Load balancing implementation and trouble shooting in 11.5.x using metric server.
5. Steven Blog http://blogs.oracle.com/schan/

