

# Exploring Oracle E-Business Suite Load Balancing Options

Venkat Perumal  
*IT Convergence*

# Objectives

- Overview of 11i load balancing techniques
- Load balancing architecture
- Scenarios to implement

# Load Balancing

- What is load balancing?
  - Improve Performance
  - Scalability

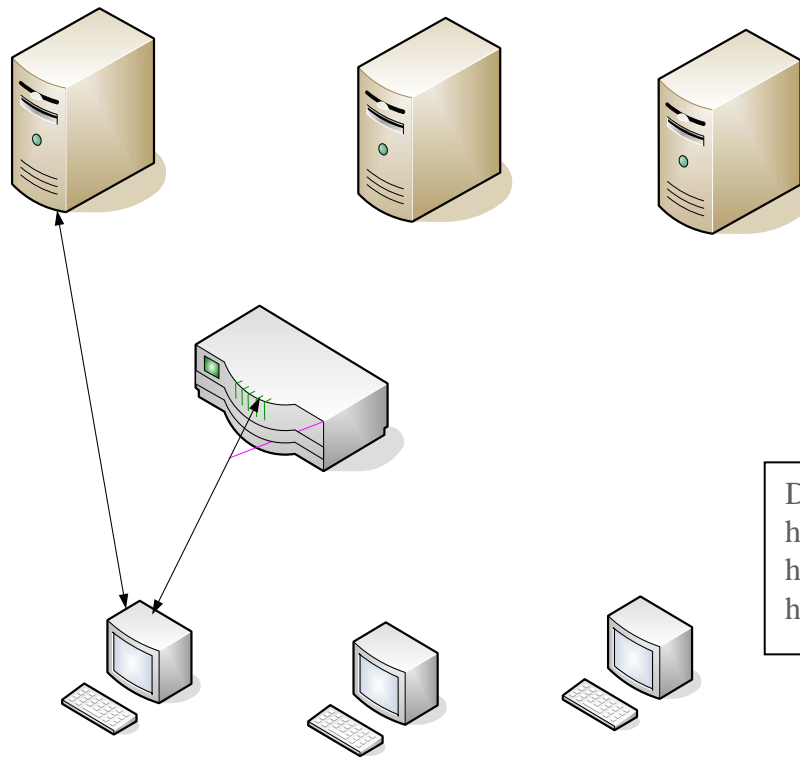
# Load Balancing Options in 11i

- DNS Based Load Balancing
- HTTP Layer Load Balancing
- Forms Metric Server Load Balancing
- Apache Jserv Layer Load Balancing
- Parallel Concurrent Processing

# DNS Load Balancing

- Independent of Oracle EBusiness Suite Techstack
- Domain Name Server (DNS) Layer load balancing solution distribute end-user requests across multiple servers based on dynamic assignments of IP addresses to a fully qualified domain name
- Associated with Round Robin DNS technique
- Inexpensive and easy to setup

# DNS Load Balancing



## DNS Mapping

```
http://apps.itc.com .....192.168.100.10  
http://apps.itc.com ..... 192.168.100.10  
http://apps.itc.com ..... 192.168.100.10
```



# DNS Load Balancing

- Load balance multiple web/forms servers
- Widely used to load balance Self Service Applications
- Virtual Host Naming option in Oracle Applications is used to configure DNS Load balancing among multiple web/forms server nodes

# DNS Load Balancing

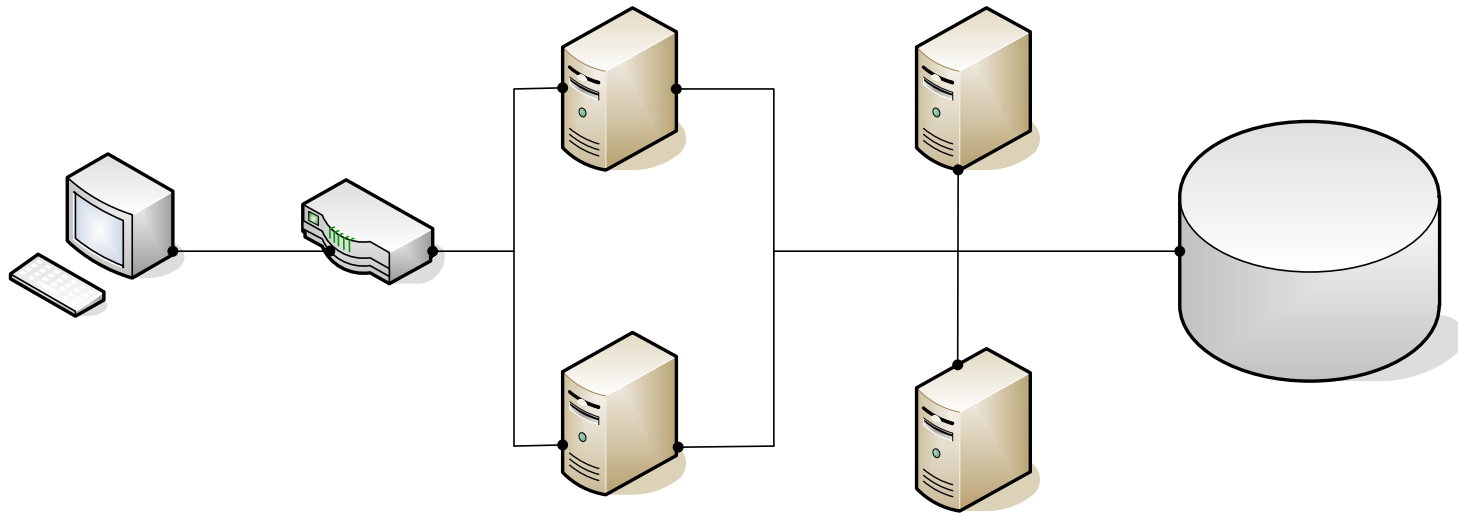
- High availability is not guaranteed
- Doesn't consider "actual" server load - balances number of users on each server, it doesn't necessarily balance the server load
- In case of server failure, DNS needs to be reconfigured - Bigger problem when removing a node than when adding one. When a node is dropped, a user may be trying to hit a non-existing server



# HTTP Layer Load Balancing

- Two types – Hardware and Software HTTP load balancers
- Load balancers accepts communication and forwards the requests to Application web nodes
- Primarily used to load balance Web nodes
- Yet to be certified by Oracle

# HTTP Layer Load Balancing



# HTTP Layer Load Balancing

- Requires additional patches and context file configuration.
- Hardware based HTTP Load-Balancers must be configured to ensure persistent session connections between clients and Web Server Nodes.
- Need to make sure that load-balancer sends all the requests from one client session to the same Web Server Node.
- Sophisticated HTTP load balancer algorithm guarantees fail over.

# HTTP Layer Load Balancing

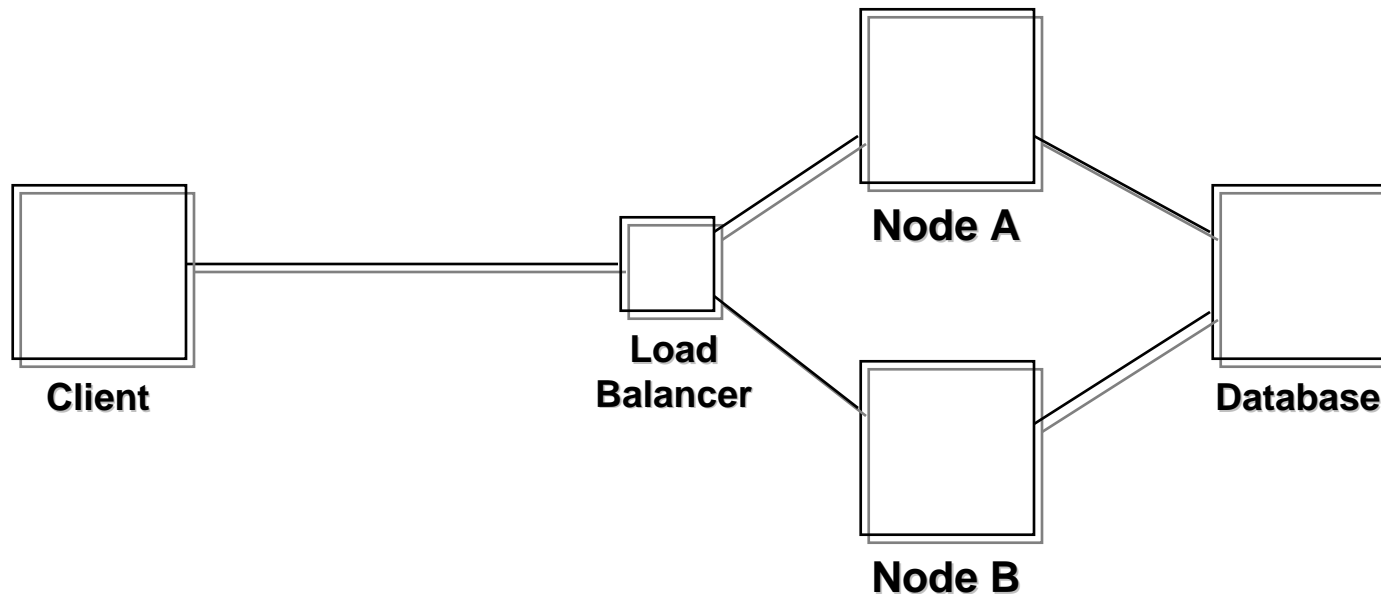
- Primarily used for load balancing high volume of Self Service Application users
- Compared to DNS based – requires expertise to setup HTTP layer load balance
- Guarantees high availability against failures

# DNS and HTTP – connectivity.

- EBusiness 11i is a stateful application
- Difference between STATEFUL and STATELESS
- A *stateful* application maintains session state information within its runtime environment between successive client calls
- A *stateless* application maintains no such information within its environment. It may persist state information in a common store such as a database or in the client browser

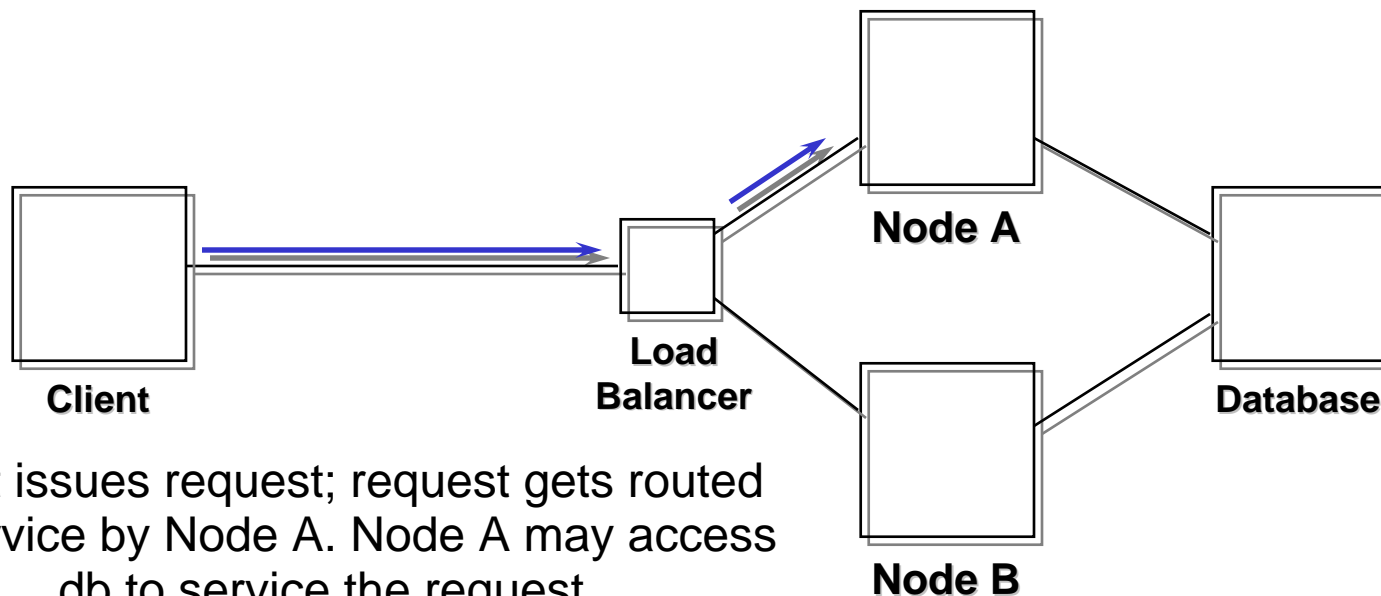
# STATEFUL Application Example

- Step-by-step example of client requests to a STATEFUL application



# STATEFUL Application Example

- Step-by-step example of client requests to a STATEFUL application

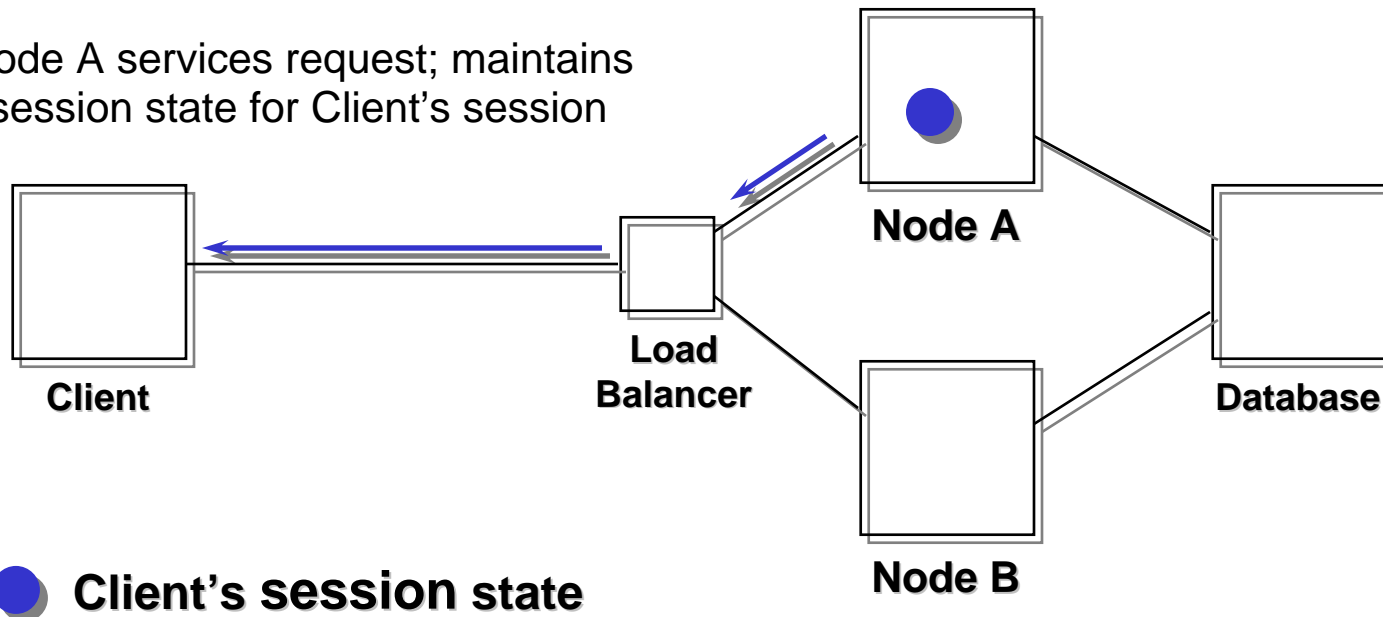


Client issues request; request gets routed for service by Node A. Node A may access db to service the request

# STATEFUL Application Example

- Step-by-step example of client requests to a STATEFUL application

Node A services request; maintains session state for Client's session

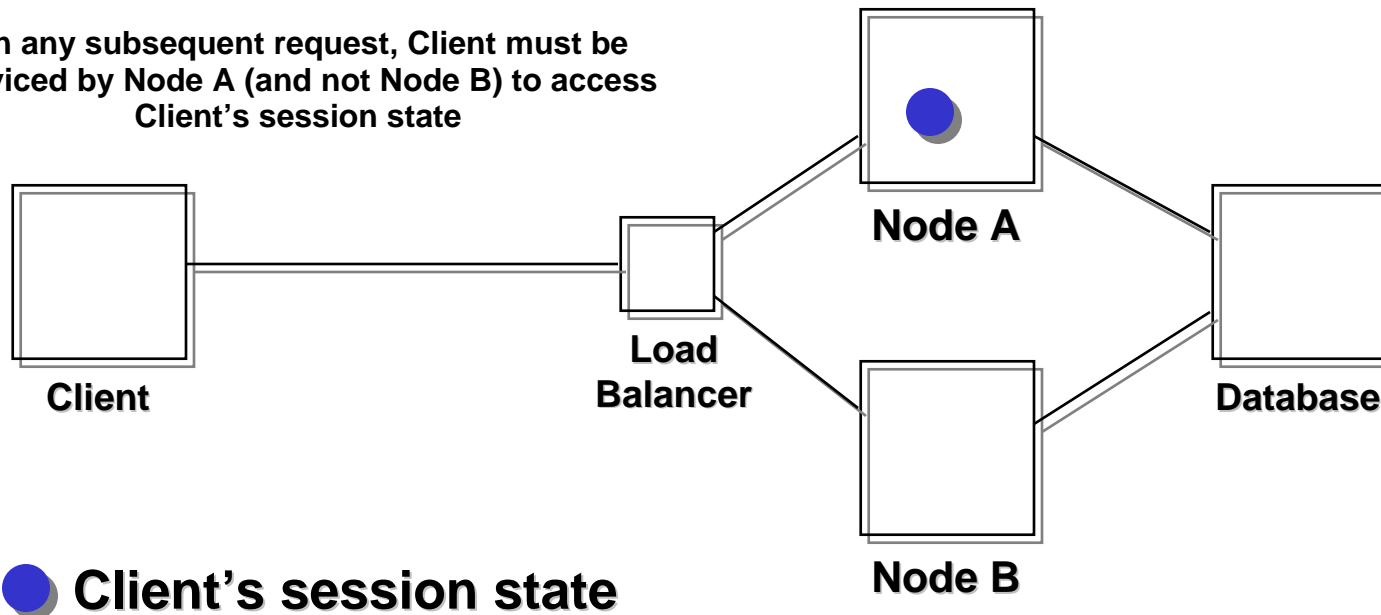




# STATEFUL Application Example

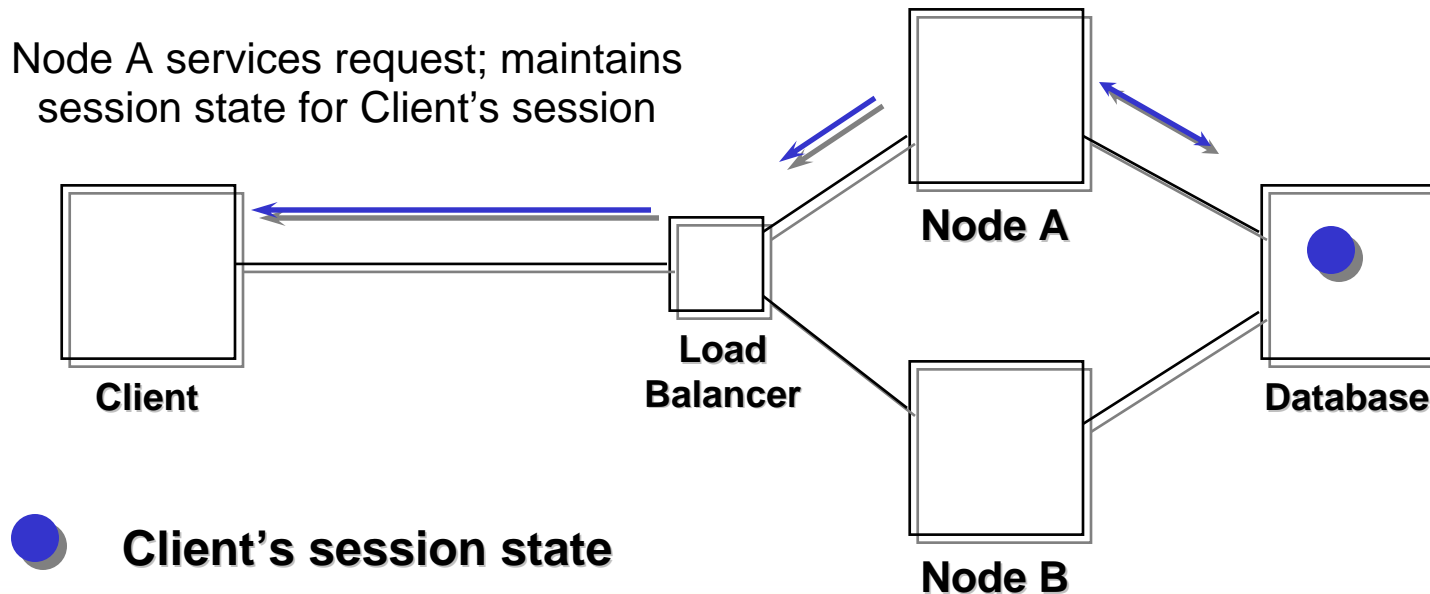
- Step-by-step example of client requests to a STATEFUL application

On any subsequent request, Client must be serviced by Node A (and not Node B) to access Client's session state



# STATELESS Application Example

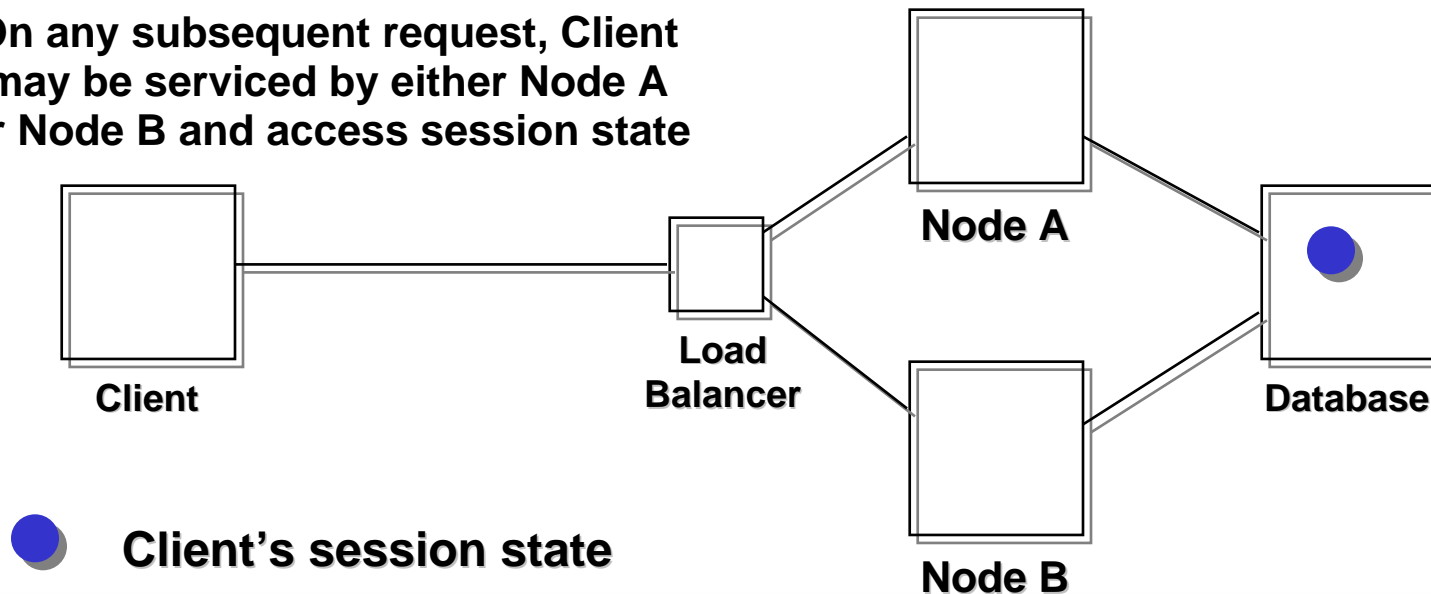
- Step-by-step example of client requests to a STATELESS application



# STATELESS Application Example

- Step-by-step example of client requests to a STATELESS application

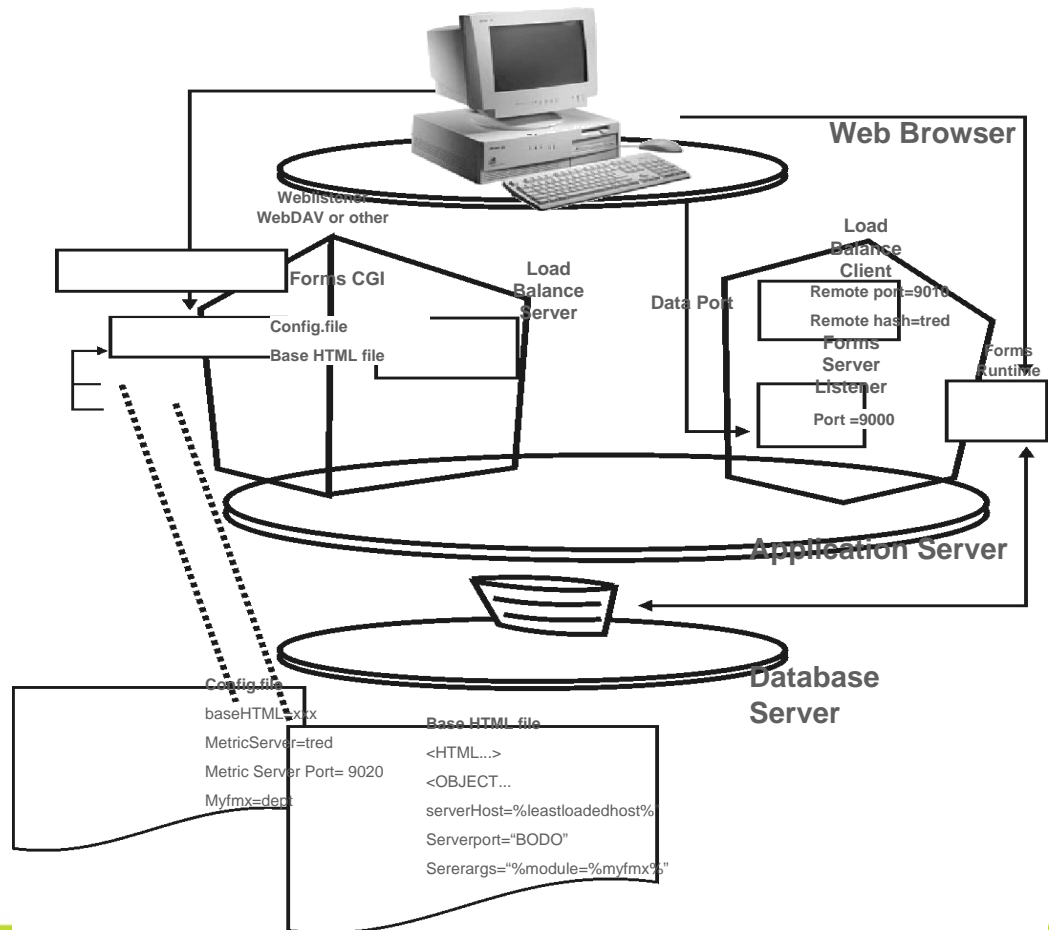
On any subsequent request, Client may be serviced by either Node A or Node B and access session state



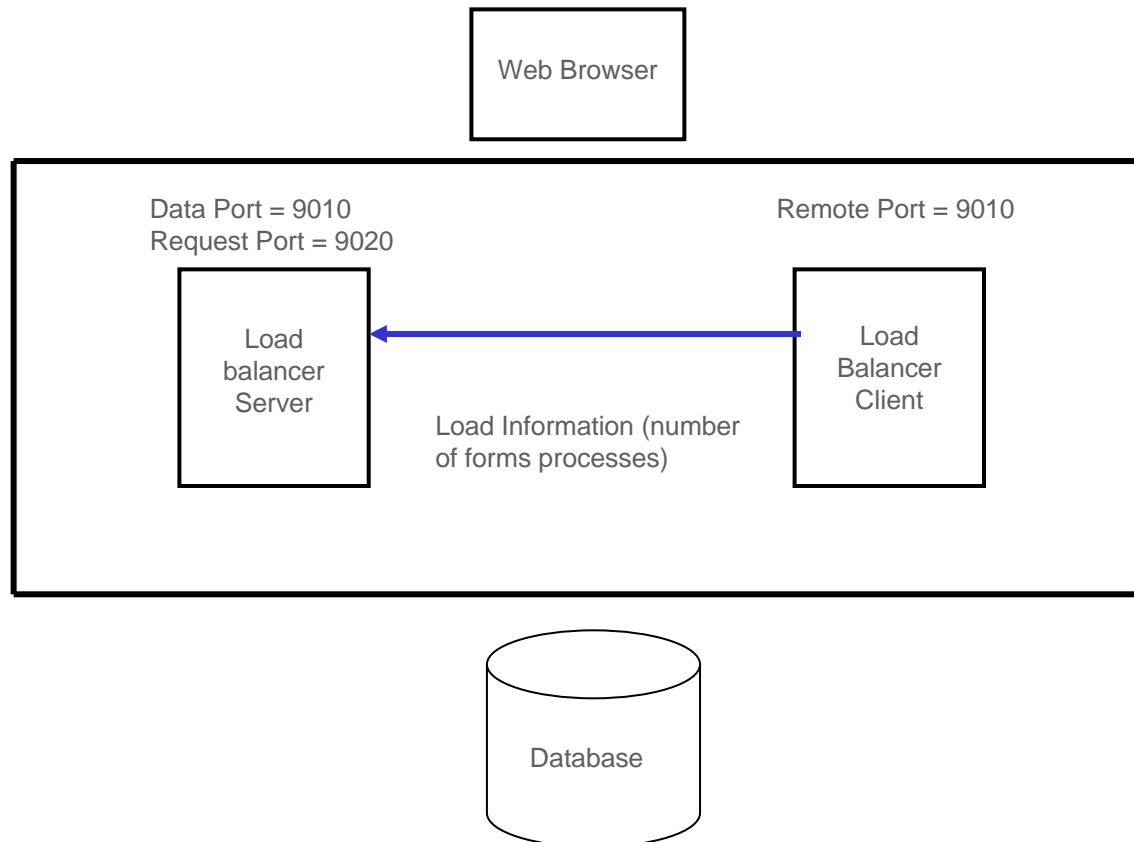
# Forms Metric Server Load Balancing

- Available from R11
- Matured load balancing technique for Forms servers
- Load balances multiple forms servers
- Requires minimum of two nodes – Primary and Secondary
- Primary acts as Metric Server and Secondary as Metric Client

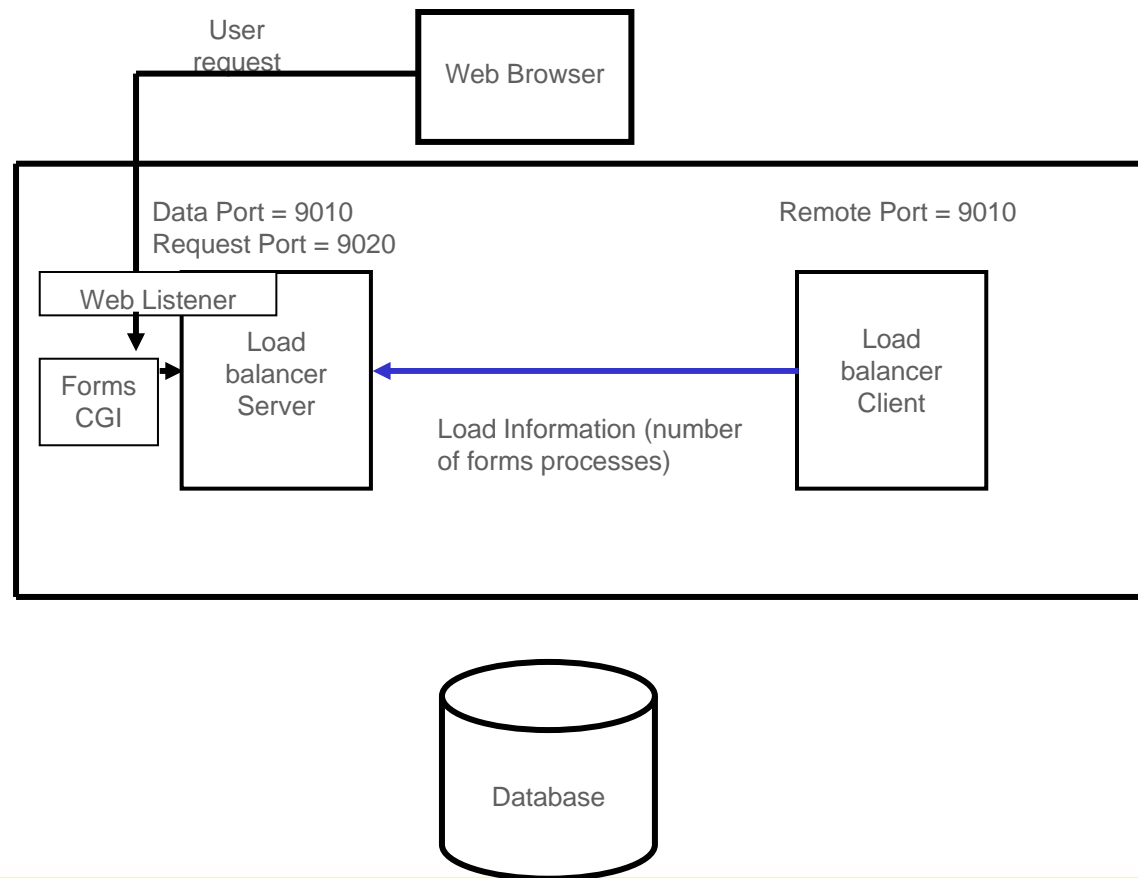
# Forms Metric Server Load Balancing



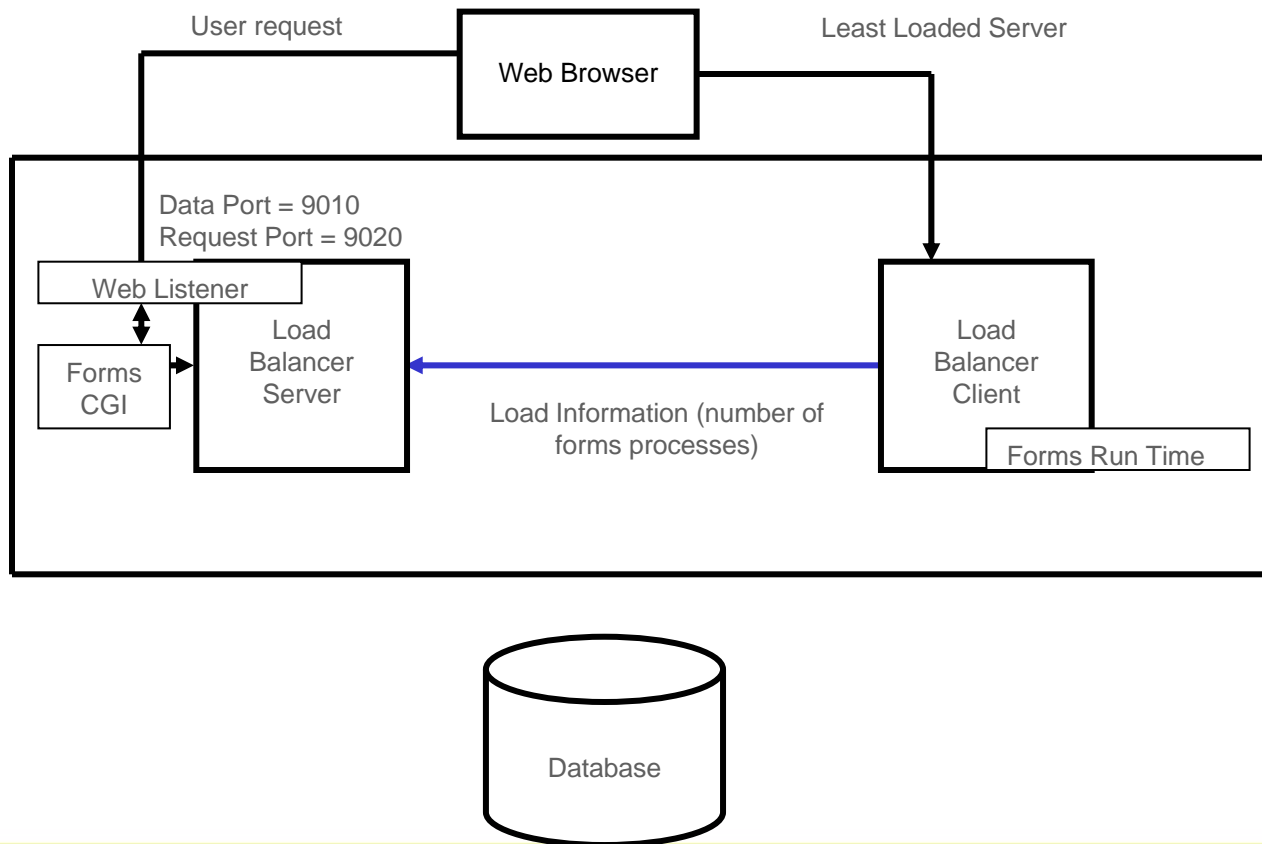
# Forms Metric Server Load Balancing



# Forms Metric Server Load Balancing

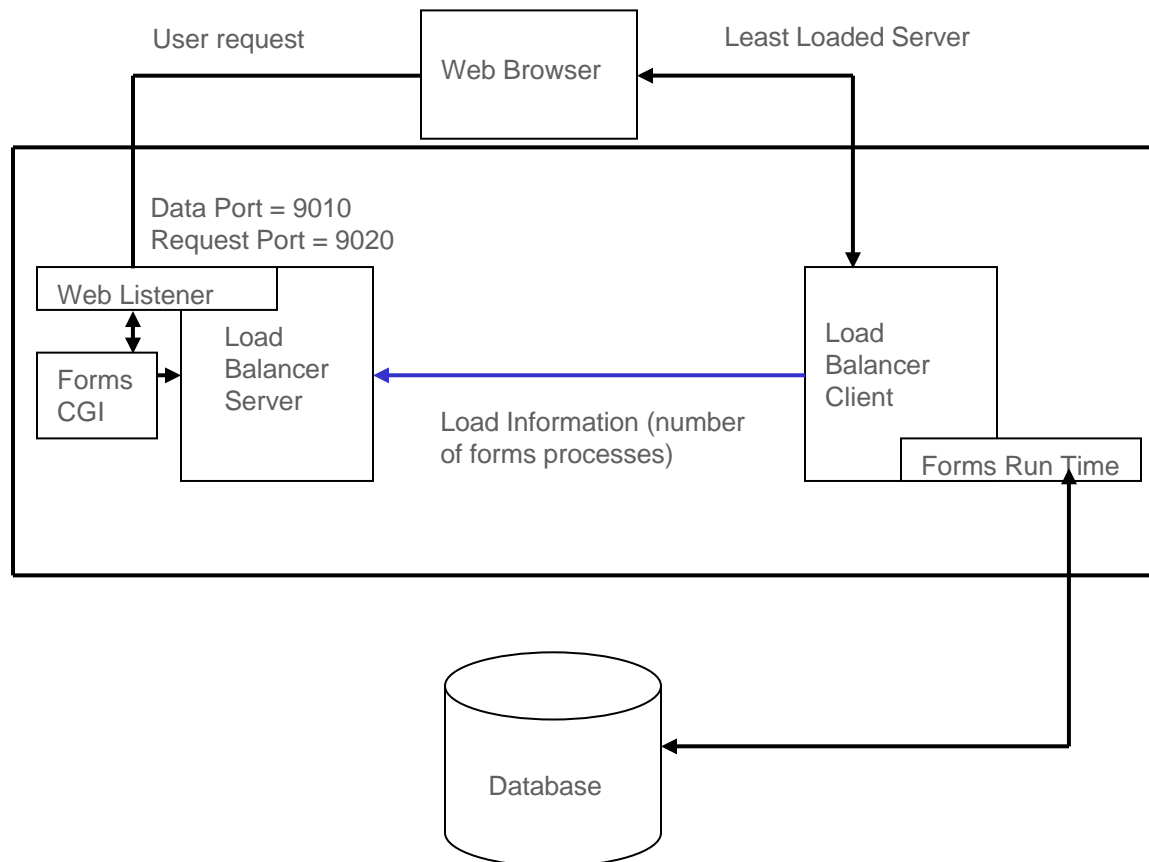


# Forms Metric Server Load Balancing





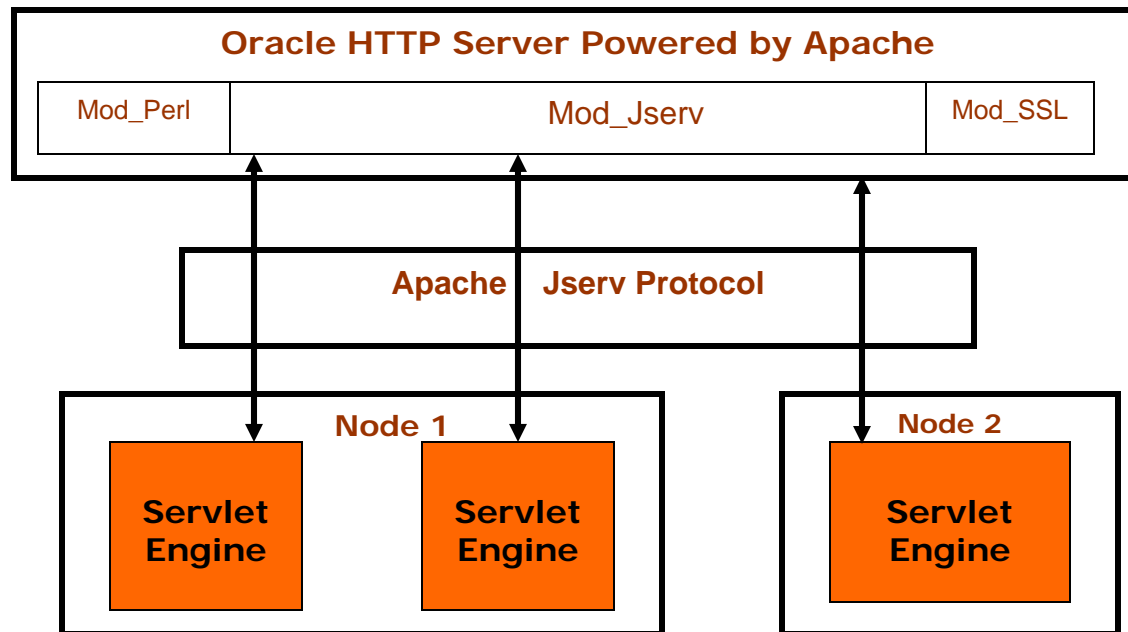
# Forms Metric Server Load Balancing



# Forms Metric Server Load Balancing

- Load balancer based on number of Forms processes running on each node
- Doesn't take into account of actual OS resources like CPU load and memory
- Doesn't guarantee high availability
- Fail over option is not supported
- Web servers are not load balanced as it is running only on Primary Node
- Supported by Oracle

# Apache Jserv Layer Load Balancing



# Apache Jserv Load Balancing

- Poor web page performance
- User requests being timed out
- Java processes taking 100% CPU on your server
- Sample error message in Apache log file:  
    `java.lang.OutOfMemoryError"`

# Apache Jserv Load Balancing

- Default setup – jserv.conf file.

```
ApJServGroup OACoreGroup 2 1 /usr/.../jserv.properties
ApJServGroup DiscoGroup 1 1 /usr/.../viewer4i.properties
ApJServGroup FormsGroup 1 1 /usr/.../forms.properties
ApJServGroup XmlSvcGrp 1 1 /usr/.../xmlsvcs.properties
```

- OACoreGroup is the default group.
- DiscoGroup is only used for Discoverer 4i requests
- FormsGroup is only used for Forms Servlet requests
- XmlSvcGrp is for XML Gateway, Web Services, and SOAP requests

# Apache Jserv Load Balancing

## **Factors affecting JVM performance**

- JServ configuration file tuning (jserv.properties and zone.properties)
- Applications modules being used
- How many active users
- Hardware specification

# Apache Jserv Load Balancing

## As per Oracle Best Practice:

- One JVM per CPU (ideally, there should be 1 JVM per 2 CPUs)
- OACoreGroup
  - 1 JVM per 100 active users
- DiscoGroup
  - 2 JVM per 75 active Discoverer users
- FormsGroup
  - 1 JVM per 50 active forms users
- XmlSvcGrp
  - 1 JVM

# Apache Jserv Load Balancing

- How to configure multiple JVM's:  
Through OAM - > Autoconfig edit wizard

The screenshot shows the Oracle Applications Manager interface. The breadcrumb trail is: Applications System:prod1158 > AutoConfig > Edit Parameters > Context File Parameters:/sanb/prod1158\_bld7/prodappl/1158/admin/prod1158\_upibld7.xml:prod1158\_upibld7. The table below lists various JVM parameters.

Title	OA_VAR	Status	Value	Description		
AD JVM Program	s_adjvprg	Servlet JVM Options	s_forms_jvm_options	Not Changed	-Xmx256M -Xms64M -XX:NewSize=60M -XX:MaxPermSiz	Servlet JVM Options
AF JVM Program	s_afjvprg	Forms Servlet JVM Processes	s_forms_servlet_nprocs	Not Changed	1	Number of JVM processes to serve the Forms Group
Amount of time given to a JVM to startup and wait to ping the JVM to see if it is alive	s_apjserv_vmt	Servlet JVM Options	s_jvm_options	Not Changed	-verbose:gc -Xmx512M -Xms128M -XX:MaxPermSize=128M	Servlet JVM Options
		Servlet JVM Startup Executable	s_jvm_startup	Not Changed	java	Path to the executable for running the Servlet JVM
		Long Running JVM	s_long_running_jvm	Not Changed	true	Indicates that this JVM is expected to have a long lifespan if this value is set to true
		MSCA Compatibility Setting for JVM (see docs)	s_mwaJVmb	Not Changed	FALSE	MSCA Compatibility Setting for JVM (see docs)
		OACORE JVM Processes	s_oacore_nprocs	Not Changed	1	Number of JVM processes to serve the OACore Group
		xmlparservc for xmllsvcs zone	s_xmlparservc	Not Changed	/sanb/prod1158_bld7/prodcomm/java/xmlparservc2-904.zip	This is the xml parser used by the xmllsvcs jvm group
		XML Services JVM Processes	s_xmllsvcs_nprocs	Not Changed	1	Number of JVM processes to serve the XML Services Group

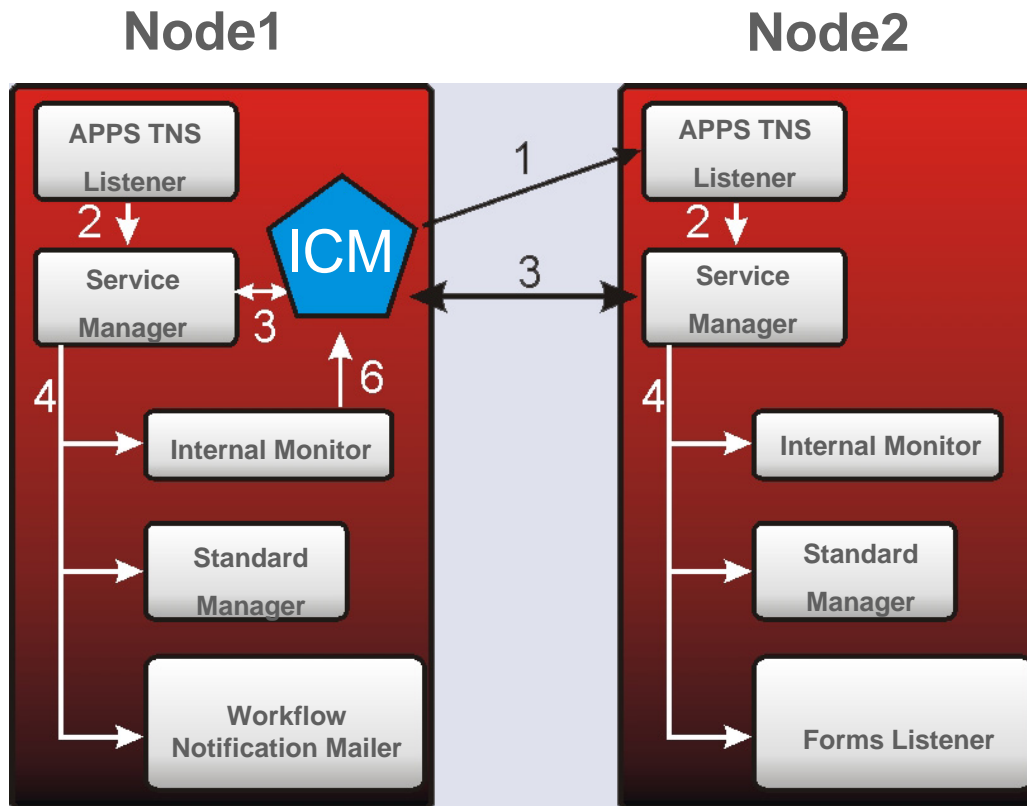
(Add to Support Cart) (Save)



# Parallel Concurrent Processing

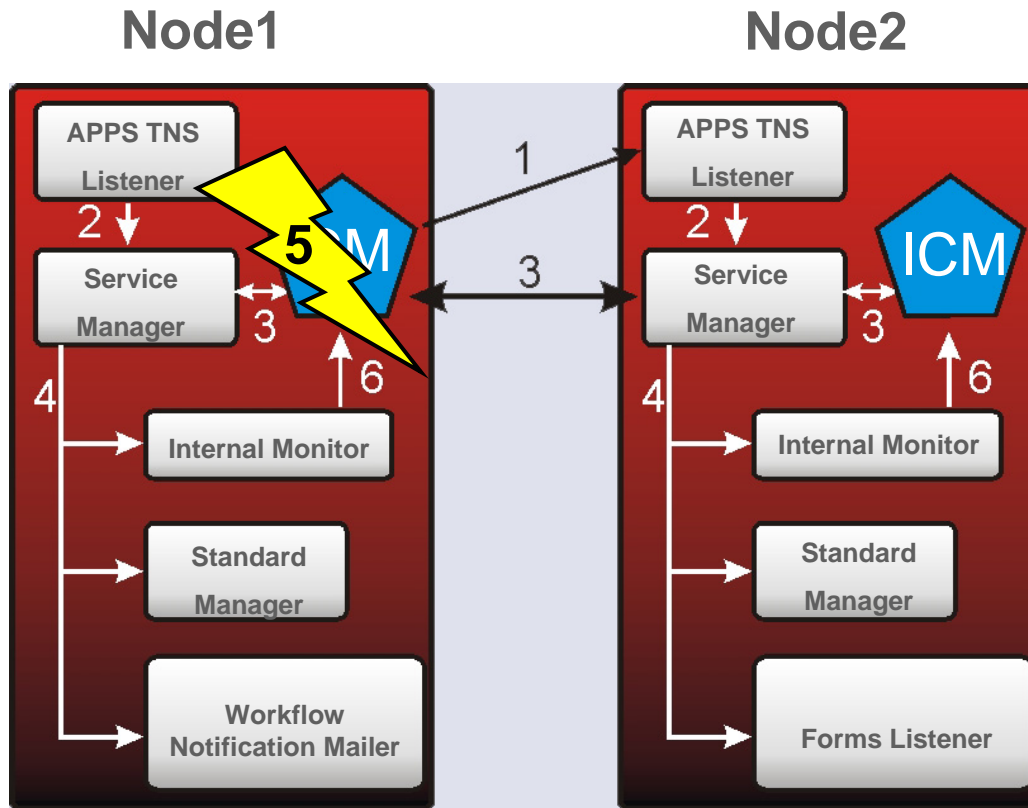
- Only load balancing option for Concurrent Manager tier
- Supported and available from R11
- Different from distributed concurrent processing
- Against one database with multiple concurrent manager nodes
- Widely used in RAC architecture

# Concurrent Manager Load Balancing



- 1- ICM contacts TNS listener
- 2- TNS listener spawn Service Manager
- 3- ICM communicates with Service Manager
- 4- Service Manager spawns Manager and Services processes

# Concurrent Manager Load Balancing



- 1- ICM contacts TNS listener
- 2- TNS listener spawn Service Manager
- 3- ICM communicates with Service Manager
- 4- Service Manager spawns Manager and Services processes
- 5- ICM Crashes
- 6- Internal monitor spawns a new ICM on Node 2

# Parallel Concurrent Processing

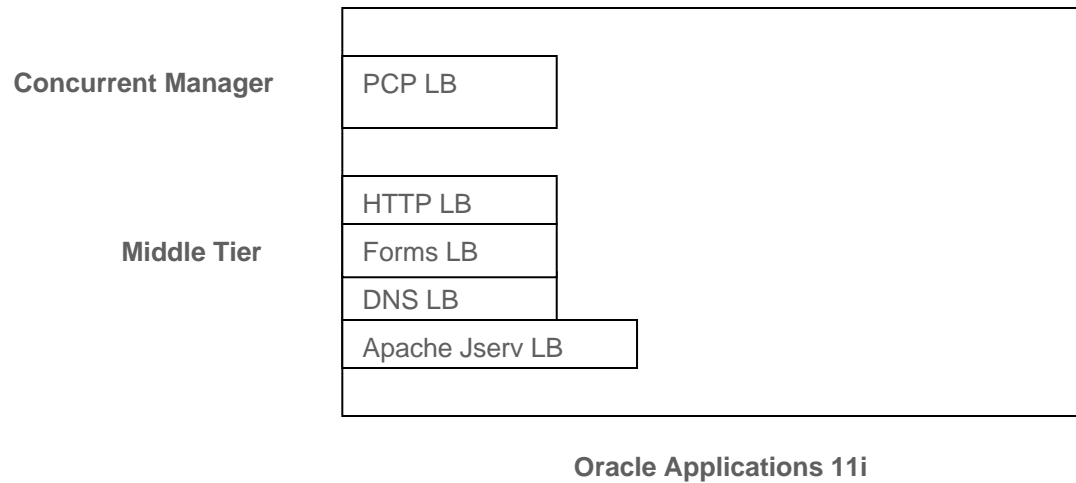
- Each node has its own memory that is not shared with other nodes
- Each node operates independently of other nodes, except when sharing a resource such as a disk
- Specialization rules can be written in Oracle to execute a particular managers on a node. For example PO Document manager can be deployed a certain node
- Managers can be deployed on multiple nodes, thus load balancing among concurrent managers

# Parallel Concurrent Processing

- **High performance** - the ability to run concurrent processes on multiple nodes to improve concurrent processing throughput
- **Fault Tolerance** - the ability to continue running concurrent processes on available nodes even when one or more nodes fails
- **Adaptability** - the ability to integrate with platform-specific batch queue and load-balancing systems to maximize concurrent processing performance on a particular platform
- **Single Point of Control** - the ability to administer concurrent managers running on multiple nodes from any node in a cluster, massively parallel, or networked environment

# Application 11i Load Balancing

## Summary:



# Questions

