

Win in the flat world

Fusion Administrator: The Governor of SOA Implementation?

– Arun Batra, Rizwan MK

Abstract

Service Oriented Architecture (SOA) -based systems, from a business point of view, enable low cost integration of heterogeneous systems – Intranet or Internet-based and hence enabling real time systems that allows enterprises to rapidly deploy and manage critical Business Solutions.

However, as we move ahead with SOA- based systems, gains in Return on Investment and business flexibilities come with an increase in the complexity of the resultant architecture and software. With new challenges across the SOA-based platform, the administrator designs and manages the environments for better implementation and administration of SOA-based systems

For more information, Contact askus@infosys.com



Background

Oracle Fusion Middleware is a portfolio of the industry leading suite of products, including J2EE services, developer and integration tools, Business Intelligence (BI), and the transaction management tool. Essentially, all these tools help in building the SOA-based infrastructure across various geographies.

Some of the tools and framework included in Oracle Fusion are:

- Oracle Application Server
- Oracle BPEL Process Manager
- Oracle ESB
- Oracle Service Registry
- Oracle Business Activity Monitoring
- Oracle B2B

As the SOA-based system grows across various functional modules and geographies, complexities of web services and business processes increase and various non-functional and technical challenges come into the picture for its efficient management.

Some of these challenges include:

- Proper Environment Design
- 24x7 High Availability of Systems
- Performance Tuning
- Environment Security
- System Flexibility for New Changes
- New Versions Compatibilities
- Integration Challenges.
- Deployment Challenges.
- Health Check and Reporting Challenges

Foremost among the set new challenges is the spiraling cost of management as complexity outstrips available skills, tools and processes required for effective SOA. The fusion administrator should not only be doing management and administration, but should also be involved in almost every part of the project life cycle.

It's a multi-skill requirement, involved in-depth knowledge of product and various different technologies described in the next section.

Fusion Administrator

Fusion administration is a combination of various technologies and framework. At times it can be very tricky as it not only involves management of the system, but also the knowledge of various supporting products. The fusion administrator may find him/herself dealing with different technologies including –

- Oracle Database
- Java Programming
- XML and XSLT Programming
- Unix Shell Scripting
- Web Services and SOAP Programming.
- Network Protocol Knowledge
- Application Server In-depth Knowledge
- Products Supporting High Availability e.g. Load Balancer and Web Broker.
- Oracle Fusion Components In-depth knowledge

It's not an understatement, when we say that the fusion administrator is the governor of SOA implementation. It's a combination of people management, technology expertise and in-depth product knowledge. In the following sections, various challenges and typical responsibilities of fusion administrators are described.

Environment Design Challenges

In any typical fusion-based integration, at least 3 environments exist - development, testing and production. At most times, there are more environments as different parallel projects are running or Client needs separate Prod Bug Fix environments or Clone Production Environments. One of the key challenges for the fusion administrator is to design a proper environment for all the instances involved in Software Development Life Cycle. Major influencers of these decisions are:

- Current Business Needs
- Time Frame to Go Live
- Future Business Needs
- Availability of the System
- Geographies of the System – Location of Dev, Test and User's Teams.
- Third Party Products Requirements
- Database Geographies
- Platform of Different Instances.
- Hardware Resources of Each Environment.
- Opening and Restricting Network for Each Instance, Based on Requirement of Each Team i.e. Dev, Test and Users.

Success factors for the Proper Design based on the Administrator's Skill Set, Past Experience, Support from all the teams (Development, Network, System, and Testing), Clear Client requirements (or rather asking the right question) and people management skills.

Load Balancing

Load balancing is a technique to spread work between two or more computers, network links, CPUs, hard drives, or other resources to get optimal resource utilization, throughput or response time. Using multiple components with load balancing, instead of a single component, may increase reliability through redundancy. The balancing service is usually provided by a dedicated program or hardware device.

Load balancing is one way to implement failover - the continuation of service despite the failure of one or more pieces of equipment. Failover without load balancing means that there is a single "live" component that is replaced by a single "backup" component in the event of a failure. Load balancing supports multiple simultaneous "live" components. Disk mirroring is one example of the "traditional" type of failover.

Among the better known commercial load balancers are Cisco Load Director, Central Dispatch, Web Server Director, ACEdirector, Big/ ip and Equalizer.

To load balance the SOA-based application:

- Create Multiple OC4J processes, i.e. Multiple Oracle Application Server (OAS) Instances
- Use HTTP Server as the Load Balancer
- Use a dedicated Load Balancer product

High Availability Challenges

Clustering is a concept that, when implemented, gives the solution for high availability. Clustering can be implemented at different levels of the system, including hardware, operating systems, middleware, systems management, and applications.

High availability is on top of the list of client requirements. Clustering is the foundation for achieving it in the OAS middleware. Clustering the components of a system together allows them to be viewed functionally as a single entity. A cluster is a set of processes running on single or multiple computers that share the same workload.

A clustered group of components often need to have the same configuration so as to ensure that the components provide the same reply to the same incoming request.

If any configuration changes are done in one instance, then the same need to be done in all other instances within the cluster. This is known as redundancy. A cluster, therefore, provides redundancy for the system.

When multiple instances of identical server components are available, then a load balancer can be used to ensure that the client requests to these components can be routed to the server

components with less work load. If any of the instances fail, requests of the failed instance can be sent to the instances that are alive

Everything in clustering is based on Topologies. There are two types of topologies by which redundancy can be achieved, they are:

- Active-Active
- Active-Passive

There are four ways in which we can implement this configuration, they are:

- Multi-casting
- Chain of Discovery Servers
- Configuring Cross Topology Gateways
- Node to Node Communication

Performance Tuning Challenges

Performance has always been and will continue to remain the most important requirement for any software development. SOA-based development is no exception. Apart from the best practices for development, SOA fusion administrator plays an important role in the tuning of the performance. Some of the areas of involvement for the fusion administrator are:

- Sufficient Hardware Resources
- Application Server Tuning
- Web Server Tuning
- Memory Leak Problems
- Sufficient Heap Size for all Application
- Proper Garbage Collection
- Efficient Connection Pooling
- Sufficient HTTP Connections
- JDBC Tuning and Statement Caching
- Logging levels
- Load balancing
- Clustering
- JVM Setting

Deployment Challenges

Deployment is always a challenge in the high changing SOA-based system. An important factor to determine a successful deployment strategy is to understand the change management in SDLC and the Release Plan of the Client. Essentially, SOA projects deployment can be done in 2 ways – manual deployment and automated deployment.

Fusion administrators should create the reusable automated deployment scripts, which should do the following actions:

- Should go to version control folder defined in the properties file (VSS or CVS or SVN)
- Removes all the .vss or .svn files or folders
- As a developer makes codes only for dev, this tool should create the code ready to be deployed on dev, test and prod server by changing the environment specific details (ex-url for fault handler, BAM server, etc.), depending on the environment that the user wants to deploy the code, which is passed as a parameter
- All these details should be in the properties file, either in txt or in xml file.
- Compiles all the processes at one go or only specified process one at a time, passed as a parameter
- Deploys the process on a specified environment and domain passed as parameters
- If the domain and version number are not both passed, then it will deploy the process to default domain with 1.0 version
- If the version alone is not passed, then it will deploy the process with version 1.0
- Can deploy in the local server or the remote server, defined in the properties file. Clustered environments should be supported
- Should compile the Java code for the project and make the jar file.
- Should be able to make a war file for the project if custom JSP or HTMLs are developed.
- Should be able to compile Enterprise JavaBeans (EJBs) and make the EAR file with BPEL/ ESB code.
- Should make the validations. For e.g. some of the WSDLs are not present.

Application Server Administration

Application servers are basically used for some complex transaction-based applications. Whenever we design an application in that we need transaction handling, database connectivity, integration, connection pooling, transaction management, distributed computing, security, monitoring, etc.

An application server exposes business logic to client applications through various protocols; possibly including HTTP also provides access to business logic for use by client application programs. The application program can use this logic just as it would call a method on an object (or a function in the procedural world). In most cases, the server exposes this business logic through a component API, such as the EJB component model found on Java 2 Platform,

Enterprise Edition (J2EE) application servers. Moreover, the application server manages its own resources. Such gate-keeping duties include security, transaction processing, resource pooling, and messaging. Like a Web server, an application server may also employ various scalability and fault-tolerance techniques. Some of the key features of an application server are:

- Scalability** Designed to build and deploy the most demanding applications
- Availability** Provides the most flexible, resilient and fault-tolerant application server platform for grid computing
- Security** Optimized for the Oracle Database for tight integration into database security capabilities
- Manageability** Provides industry-leading automation for dramatically lower operational costs

However, by using the application server, all this can be done and handled without writing any kind of code. In the application server, this is possible with the use of plug-ins.

Among the better known commercial J2EE application servers are WebLogic Server (BEA), Jboss (Red Hat), WebSphere (IBM), Jrun (Adobe), Geronimo Application Server (Apache Foundation), and Oracle OC4J (Oracle Corporation).

Conclusion

If Oracle SOA Middleware is the answer for organizations that are looking for a highly flexible and manageable solution, fusion administrator is the person. They should first, be contacted for infrastructure-related options and proper environment design to ensure ROI.

It's a cross and multi-skill level job, but perhaps the most important skill to a fusion administrator is problem solving, which is frequently done under various sorts of constraints.

The fusion administrator helps the SOA system run smoothly with almost 24x7 availability, ensuring that the business runs according to client expectations.

About Authors:

Arun Batra is a technical architect at Infosys Technologies Limited. He has 11 years of experience in J2EE and Web Services related technologies. Arun is providing in-depth Oracle SOA Administration/ Fusion Middleware strategy for various clients and also responsible for defining best practices for the Oracle SOA/ Fusion technologies administration. He has also been involved in developing new tools for the Oracle SOA Technologies.

Rizwan M.K. is a senior principal at Infosys Technologies Limited. He has 15 years of experience in Oracle Applications and Technologies and has played a key role in defining and delivering strategic business and technology solutions to Infosys clients. At Infosys, Rizwan is a key member of Oracle Technology Architecture Group, the Technology wing of the Infosys Oracle Practice. As a senior principal, Rizwan is a leader in developing strategic initiatives and handling technology critical projects for organizations in the Oracle domain.

Read more on this at Infosys Blog

http://www.infosysblogs.com/oracle/2008/06/having_confusion_around_fusion.html

© 2008 Infosys Technologies Limited.

ALL RIGHTS RESERVED

Copyright in whole and in part of this document "**Fusion Administrator: The Governor of SOA Implementation?**" belongs to Infosys Technologies Limited. This work may not be used, sold, transferred, adapted, abridged, copied or reproduced in whole or in part in any manner or form or in any media without the prior written consent of Infosys Technologies Limited.