Oracle Application Server 10g
J2EE and Web Services

An Oracle White Paper
September 2003
INTRODUCTION

Service-oriented architecture has fast become the cornerstone design principle upon which organizations are building and integrating modern business applications. This, in turn, has created a demand for a new class of middleware software infrastructure that is engineered from the ground up with a service-oriented perspective.

The business drivers behind the service oriented architecture approach are simple and compelling:

1. Reduce costs by adopting a standards-based software built on top of commodity hardware and operating systems
2. Increase business responsiveness by focusing software development and integration on re-usable services rather than re-development efforts.

Organizations can expect the build out of service-oriented architectures to intensify as Web service standards in the security, reliability, transaction and business process space continue to mature. The recent emergence of grid computing, a natural next step in service-oriented architectures, will also create more impetus in this direction.

The new release of Oracle Application Server 10g is designed to provide a standards-based, mission critical platform for organizations planning their futures around such architectures. At the core of this infrastructure is unparalleled support for the key standards used in service-oriented architectures – the Java 2 Enterprise Edition (J2EE) component model and Web services based on XML, SOAP, WSDL and UDDI.

The underlying runtime engine for this infrastructure is the Oracle Application Server Containers for J2EE 10g (OC4J). OC4J has a proven heritage of being lightweight, easy-to-use and highly productive for developers and continues to lead industry benchmarks for performance
and cost-effectiveness – the most recent top results in SPECjAppServer2002 being certified in September 2003.¹

However, it is not just J2EE and Web service standards, productivity and performance alone that make for robust enterprise systems. These must be complemented by an infrastructure designed for unbreakable reliability, high availability and predictable scalability. Oracle Application Server 10g does exactly this - tying together all these elements in an integrated service oriented architecture. Organizations will find that OC4J 10g gives them the option to seamlessly grow from the smallest pilot projects to the largest, most distributed computing environments.

J2EE IN SERVICE ORIENTED ARCHITECTURES

J2EE Foundation

The release of the J2EE standard in 1998 was the harbinger of today’s emphasis on service-oriented architectures. Before one can adopt service-oriented approaches for development and integration, one must have a proven implementation model to work from.

Service oriented architectures are fundamentally built upon infrastructures that enable developers to encapsulate business functionality into re-usable components that are made public through well-defined interfaces. Not only does the Java language encourage this approach, J2EE itself facilitates it by clearly separating concerns in the architecture itself – a set of standards for developing the application presentation layer, another set for developing and integrating the business layer and, lastly, a well-defined set of infrastructure services including transactions, security, connectivity and lifecycle services.

With nearly 3 million Java developers active today, recently predicted to grow to 10 million by 2006, most mainstream developers have become conversant with J2EE.² Indicative of this ubiquity is how best practices, well-recognized J2EE architectural design patterns and popular J2EE frameworks have become widely understood and implemented across the industry. Now, it is common to see business applications of all sizes – small departmental ones to the largest most highly trafficked Internet sites – built on J2EE.

Oracle has been a driving force in the J2EE community from the start and has established a reputation as a leading implementer of the J2EE specification. Oracle Application Server 10g is currently used by over 16,000 organizations as a mission critical platform for J2EE applications. A large part of that commitment to J2EE manifests itself in Oracle’s

¹ SpecJAppServer Benchmark information can be found at http://www.spec.org/jAppServer2002/
participation in the Java Community Process defining the J2EE standard – Oracle engineers participate in over 100 different Java Specification Requests defining the next generation of J2EE.³

**Oracle Application Server Containers for J2EE 10g**

The result of Oracle’s investment in J2EE is the Oracle Application Server Containers for J2EE 10g (OC4J). OC4J is fully J2EE 1.3 compliant container that runs on a standard file based JDK 1.4 Java Virtual Machine and provides complete support for JSPs, Servlets, Enterprise JavaBeans (EJBs), Web services and all J2EE services. Figure 1 provides an architectural summary of the J2EE 1.3 standards supported within the OC4J environment.

<table>
<thead>
<tr>
<th>JavaServer Pages (JSP) 1.2</th>
<th>Servlets 2.3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enterprise Java Beans 2.0</td>
<td></td>
</tr>
<tr>
<td>Session EJB</td>
<td>Entity EJB</td>
</tr>
<tr>
<td>Java Message Service (JMPS)</td>
<td>JTA 1.0</td>
</tr>
<tr>
<td>Java Transaction API</td>
<td>Java Database Connectivity (JDBC) 2.0 + Extensions</td>
</tr>
<tr>
<td>Java Naming and Directory Interface (JNDI) 1.2</td>
<td>J2EE Connector (JCC) 1.0</td>
</tr>
<tr>
<td>Java Authentication and Authorization Service (JAAS) 1.0</td>
<td></td>
</tr>
</tbody>
</table>

**Table 1: Oracle Application Server Containers for J2EE - J2EE 1.3 Support**

OC4J is extremely lightweight – requiring approximately 27 MB disk and 20MB of memory. The result is a that it is fast to install – typically taking less than 15 minutes – and it is very easy to use with simple management and configuration, capabilities supporting standard Java development and profiling tools. Because it is written in Java it is available on 32-bit and 64-bit versions of standard operating system and hardware platforms including Solaris, HP-UX, AIX, Tru64, Windows NT/2000/2003 and Linux.

³ See http://www.jcp.org
The importance of standards compliance is core to the J2EE value proposition. Not only do organizations choosing J2EE get the built in benefit of application portability by adhering to J2EE as well as the ability to pull from a large pool of knowledgeable developers and system integrators, they get the benefit of a thriving market of third party J2EE applications provided by independent software vendors. Oracle Application Server 10g, for example, has over 1,500 ISV’s certified on the OC4J J2EE containers and over 5,000 consulting organizations taking advantage of its J2EE compliance.

**Presentation Tier**

A key design goal of service-oriented architectures is re-usability of the business services to provide functionality to any client – whether the clients be other applications or a human facing business applications. J2EE works hand in hand with the rest of the Java platform, Java 2 Standard Edition (J2SE) and Java 2 Micro Edition (J2ME), to offer developers a wide variety of client implementation choices. Required by within J2EE specification is support for Servlets and JavaServer Pages, which is fully implemented in OC4J.

**JavaServer Pages**

OC4J provides a JSP translator and runtime engine compliant with version 1.2 of Sun’s JavaServer Pages specification. It includes full support for JSP directives and all core and standard JSP tags, iterator support and buffer management. OC4J installs the JSP Standard Tag Library (JSTL) out-of-the-box to allow JSP developers to add custom actions and make use of the rich expression language provided in JSTL.

In addition to JSTL, a number of additional JSP custom tag libraries are also provided. These additional custom tag libraries enable JSP developers to make easy and consistent use of the advanced features of the Oracle Application Server 10g technology stack. The range of functionality provided with the tag libraries includes looking up and using both local and remote EJBs, calling Web services, obtaining JDBC connections for database operations which use connection pooling, sending and receiving e-mail, accessing files on the local file system or Oracle Internet File System including uploading and downloading, embedding XML result sets into JSP pages, accessing a range of multi-media content, accessing the personalization engine, executing web searches and queries, embedding SQL within pages using SQLJ and using national language support (NLS) to support the internationalization of applications.

**Servlets**

OC4J provides a complete Servlet 2.3 implementation, which provides support for simple, complex and parameterized filters and chaining. The
Servlet container also supports application life cycle events, advanced globalization/national language support, and deployment descriptors. It includes complete WAR file-based deployment, auto-compile, auto-deployment and stateful fail over of clustered deployment of Servlets.

The OC4J Servlet container is 100% application code compatible with the Tomcat Servlet Engine delivered by the Apache consortium, enabling applications that are developed for Tomcat to be easily deployed and executed on OC4J. All future work on the OC4J Servlet container will remain compatible with subsequent Apache consortium efforts including Jakarta and Catalina.

**Business Tier**

At the heart of service-oriented architectures will be the business services themselves. For this J2EE provides rich component model, Enterprise Java Beans, along with supporting services such as messaging, transactions, database and numerous legacy and database connectivity approaches. In OC4J, the EJB implementation is a mature implementation proven to deliver high performance in benchmarks like SPECjAppServer2002.\(^4\)

**EJB**

OC4J 10g provides complete support for the EJB 2.0 specification. It provides full support for session beans, entity beans, and message driven beans. The entity bean implementation provides Bean Managed Persistence (BMP), Container Managed Persistence (CMP), local interfaces, container managed relationships and queryability via the EJB query language.

Within the entity bean implementation, a basic persistence manager supplies both simple (1:1) mapping and complex relationship (1:n) mapping. It supports both 1:1, many:1, and 1:many object-relational mappings. It automatically maps fields of an entity bean to corresponding database table. Additionally, it allows users to specify object-relational mappings between EJBs allowing developers to use 1:1 mappings in EJBs with zero work.

To facilitate application maintenance and deployment, Oracle Application Server 10g provides a number of features including dynamic EJB stub generation, full EAR file-based deployment, automatic deployment. and simplified XML custom configuration. It also supports clustered deployments and fail over of EJBs. CORBA interoperability provides the capability to build EJBs and access them as CORBA services from CORBA clients.

Persistence Architecture

A forgotten component in many service-oriented architecture is the topic of persistence. This area can be extremely time and resource intensive as it typically bridges an object oriented world from which the services have been constructed and a relational backend database such as Oracle Database 10g.

Oracle Application Server TopLink (OracleAS TopLink) integrates the object and relational data worlds, allowing applications to transparently store and retrieve Java objects & entity beans using relational databases. OracleAS TopLink makes application developers more productive by offering an easy to use mapping workbench to map Java objects to relational database schemas simplifying one of the most difficult aspects of developing service oriented applications.

Using OracleAS TopLink, developers gain the flexibility to map objects and Enterprise JavaBeans to a relational database schema with minimal impact on ideal application design or database integrity. As a result, developers can remain focused on addressing business needs rather than building infrastructure. OracleAS TopLink is built on JDBC and is portable across any JDBC compliant database, including Oracle, IBM DB2, Microsoft SQL Server, Sybase, Informix, and MS Access.

OracleAS TopLink provides a rich set of performance optimization and scalability features. Performance is accelerated with caching techniques that minimize database and network traffic while always leveraging optimizations provided by JDBC and the databases.

J2EE Services

The supporting services in J2EE are the low level plumbing that nearly every application based on service-oriented architecture requires. The power of J2EE is how mature the underlying services of JMS, JCA, JNDI, JTA, JAAS and other supporting API’s are. Developers are able to focus on writing high value business services rather than architectural plumbing – the J2EE platform provides this out-of-the-box.

Java Message Service

OC4J gives the user the flexibility of choosing between a number of JMS providers, depending on their integration and quality-of-service requirements.

- **Oracle Application Server JMS (OracleAS JMS)** – The new OracleAS JMS provider is fully JMS 1.0.2b compatible, supports an in-memory persistence model for optimized performance, and provides for durable messaging support through a file based persistence mechanism.
• **Oracle JMS (OJMS)** - OJMS is the Java front-end for the Oracle database integrated Advanced Queuing (AQ), which offers secure, transactional, recoverable, guaranteed delivery of messages. OJMS leverages the Oracle database robustness, query-ability and DML operations, scalability and high availability, and support for all data types in message payload, including relational data, text, XML, and multimedia.

• **Pluggable JMS Providers** – OC4J defines a ResourceProvider interface for plugging in message providers and provides the implementation classes for Oracle's Advanced Queuing and for third-party messaging systems such as MQSeries, SonicMQ and SwiftMQ.

**J2EE Connector Architecture**

OC4J provides complete J2EE Connector support including support for both stand-alone and embedded resource adapters; full support for the J2EE Connector client API; and support for the three "Quality of Service" elements of J2EE Connector - transactions, security, and connection pooling. Oracle also provides J2EE Connector adapters certified for accessing Oracle and non-Oracle Databases, and is in the process of certifying J2EE Connector adapters for legacy systems and packaged applications.

**Java Transaction API**

OC4J provides a complete implementation of the JTA specification for highly scalable 1PC and 2PC. It includes optimizations such as auto-detection of 1PC transactions to improve performance and the support of three concurrency control options for greater performance and scalability: read-only locking, pessimistic locking and optimistic locking.

**JDBC**

Oracle Application Server 10g provides Type 2 and Type 4 JDBC drivers to access Oracle8i Databases, Oracle9i Databases and the latest Oracle Database 10g. In addition to full compliance with JDBC 2.0 there are several JDBC enhancements:

• **Complete Data Type Support** - Support for advanced data types, such as BLOBs, CLOBs, Character Streams; Abstract Data Types; Collections; and with the Oracle Database 10g Release support for Abstract Data Types with Inheritance

• **JDBC 2.0 Connection Pooling** - Full support for JDBC 2.0 Connection Pools.
- **Advanced Features** - Support for Transparent Application Failover, which allows the mid-tier to redirect connections to a “failed-over” node when an Oracle Database fails; Real Application Cluster support, scrollable result sets; batch updates; and, Unicode support.

- **DataDirect JDBC Drivers** - To access non-Oracle Databases from OC4J, Oracle distributes the Type 4 JDBC Drivers from DataDirect (formerly Merant). The DataDirect JDBC drivers provide access to Informix, Sybase, Microsoft SQL-Server, and IBM DB/2 Databases from Oracle Application Server 10g.

**Java Authentication and Authorization Service**

OC4J offers a complete JAAS implementation with the following features:

- **Comprehensive Security** - JAAS can be used as a mechanism for authentication (identifying users), authorization (limiting what they can do), and delegation (enabling code to run securely, with privileges of other users).

- **PKI, Single Signon (SSO), or Custom Authentication** - The JAAS provider provides a flexible authentication framework which offers specific authentication mechanisms based on SSL and SSO but also allows developers to integrate custom authentication modules through the standard JAAS Login Module API.

- **Role-Based Access Control** - The JAAS Provider provides secure, centralized, and customizable role-based access control. The JAAS Provider does not explicitly define user communities but uses a concept called realms, which provide access to users and roles.

- **JAAS Delegation** - The JAAS Provider has support for privilege delegation allowing a J2EE Application to run with the privileges of a specific user. Both RunAsClient (where an application runs with the privileges of the current client user) and RunAsID (where an application runs with the privileges of a specific user) are supported.

- **JAAS Repositories** - Oracle JAAS provider can manage user identities in either an encrypted XML file or in Oracle Internet Directory (OID). Integration with OID enables Java-based applications to leverage the centralized user management capabilities of OID, yet allows delegated administration by organization.
WEB SERVICES

Web Services Architecture

Web services are a set of Internet standard messaging protocols, programming standards, and network registration and discovery facilities that expose business applications to other services and clients over the Internet. Oracle Application Server 10g provides a complete infrastructure for developing, deploying and managing Web services supporting SOAP 1.1, WSDL 1.1, and UDDI 2.0.

While Web services are often seen as a new technology the standards behind them have been under development since 2000. Figure 1 provides a synopsis of the key areas Web services standards are tackling. At the lower levels, the core transport, messaging, description and discovery standards have reached a level of stability that many organizations are well underway implementing in production applications. This is reflected by robust implementations like Oracle Application Server 10g.

Over the next year, it is expected that the higher level standards in the area of security, reliability, transactions and business processes will emerge from standards bodies to flesh out remaining key areas of Web services, enabling wide adoption of service oriented architectures.

<table>
<thead>
<tr>
<th>Orchestration/Choreography</th>
<th>Business Processes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reliable Messaging</td>
<td>Security</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>UDDI</td>
<td>Discovery</td>
</tr>
<tr>
<td>WSDL</td>
<td>Description</td>
</tr>
<tr>
<td>SOAP</td>
<td>Message</td>
</tr>
<tr>
<td>XML</td>
<td>Transport</td>
</tr>
<tr>
<td>HTTP, IIOP, JMS, SMTP</td>
<td></td>
</tr>
</tbody>
</table>

Figure 1: Web Services Standards Focus Areas

An important development in the J2EE community has been the standardization of Web services API within the J2EE platform with J2EE 1.4. The primary design intent of this standardization has been first a natural uptake of Web services in the platform so that service oriented
architectures can take full advantage of the foundation provided by J2EE 1.4. Web services in J2EE 1.4 build on the underlying J2EE platform by adding the core standards SOAP, WSDL and UDDI as first class citizens. For J2EE developers, Web services have become not a new concept; but an effortless extension to the programming model they have been using for years.

Like with the Java Community Process for J2EE, Oracle has its Web services engineering staff deeply involved in the Web services standards development process. In particular, the World Wide Web Consortium (W3C), the Organization for the Advancement of Structured Information Standards (OASIS), the Web Services Interoperability Group (WS-I) and Global Grid Forum are key focus groups for Oracle as they are defining these next generation standards.

**Oracle Application Server Web Services 10g**

Oracle Application Server Web Services (OracleAS Web Services) 10g provides a highly scalable runtime infrastructure for developing, deploying and managing Web services. In OracleAS Web Services 10g significant enhancements were made, focusing on improving Web services performance, scalability, interoperability, ease-of-use and testing. A fundamental goal of the OracleAS Web Services 10g release has been to simplify and speed the process of publishing J2EE artifacts as Web services following industry standards.

OracleAS Web Services 10g is built on top of the OC4J J2EE 1.3 infrastructure and takes advantage of its built-in, industry leading security, scalability, reliability, availability and performance capabilities. Built into OC4J is a powerful set of command line utilities that automate the publishing of J2EE artifacts as Web services. These utilities are also available through a visual design time in Oracle JDeveloper and can be integrated into leading 3rd party Java IDE’s.

Publishable Web service artifacts in OracleAS Web Services 10g include stateless and stateful Java classes, stateless session EJB, JMS topics and queues, and PL/SQL stored procedures. Deployment of Web services to Oracle Application Server 10g generates matching WSDL descriptions and client side proxy stubs.

Key features include support for document style Web services, custom type serialization, typed and untyped support, a JSP Web service tag library and generation of an endpoint home page for each deployed Web service. For each Web service deployed to Oracle Application Server 10g the endpoint home page provides a browser based testing harness, a link to the WSDL file and a link to the client side proxy. Figure 3 provides an architectural overview of OracleAS Web Services 10g.
SOAP

OracleAS Web Services 10g provides a SOAP 1.1 message processor which runs as a Servlet on OC4J. It provides a complete implementation of the SOAP 1.1 specification; support for cookies and sessions and SOAP message delivery over HTTP. OracleAS Web Services 10g SOAP processor integrates the Oracle XML parser for maximum performance. Key features include support for RPC and document style requests, typed and untyped support for .NET interoperability, custom serialization, built in support for complex database types such as CLOBs and XMLType and server side debugging support.

WSDL

OracleAS Web Services 10g provides tools to generate WSDL 1.1 compliant stubs and skeletons in two ways:

(i) **Auto-Generation** - OracleAS Web Services 10g runtime automatically generates WSDL, server skeletons, and client stubs on demand as it needs them - this makes application and system maintenance significantly simpler than competitor products.

(ii) **WSDL Generation Tools** - OracleAS Web Services 10g also provides a set of tools to statically generate WSDL and client stubs given a Java class or J2EE application. These tools have also been integrated seamlessly with Oracle JDeveloper and work with other popular Java IDEs.

UDDI

OracleAS Web Services 10g provides a UDDI v 2.0 compliant registry to publish and discover Web services. It has several important elements:
(i) **Database-backed** – can be set up to use an Oracle Database, Microsoft SQL Server Database or IBM DB2 database as a backend data store.

(ii) **Browsing** - The UDDI registry can be browsed using any standard UDDI browser and is certified with Microsoft and IBM's UDDI browsers.

(iii) **Compliant programmer API** - It completely implements the UDDI Programmer API specification which allows users to browse, query, and publish Web services. It can be operated both as a private UDDI registry within organizations and as a public UDDI node supporting the standard UDDI synchronization mechanisms.

(iv) **Managing UDDI entries** - The UDDI registry supports a number of standard UDDI classification taxonomies including NAICS, ISO 3166 and UNSPSC. Web services can be published to the UDDI registry, and existing services can be both browsed and deleted using Oracle Enterprise Manager.

**Web Services Interoperability**

In a service-oriented infrastructure it is not enough just to have core Web service functionality. The infrastructure itself must be engineered to facilitate interoperability and should help architects implement interoperable business services. Driving this interoperability requirement is the natural expectation of architects that business services built on one vendor’s platform, be they developed with J2EE, .NET, legacy environments or open source, should be able to seamlessly interact with business services built on another vendor’s platform.

In an effort to satisfy the broadest number of users and platforms, OracleAS Web Services 10g has approached interoperability from two perspectives: (i) Oracle engineering participation in the interoperability organizations SOAPBuilders and Web Services Interoperability (WS-I) Group; and, (ii) engineering specific interoperability characteristics into the OracleAS Web Services infrastructure. The design goal is to make it easy to develop Web services that conform strictly to standards for interoperability purposes but also to be very open to idiosyncracies and non-standard behaviour from other vendor platforms.

SOAPBuilders is a grassroots organization dedicated to identifying interoperability issues between shipping SOAP implementations with engineers from vendors like Oracle, IBM, Sun and Microsoft participating. A key deliverable of SOAPBuilders is a set of SOAP  

---

5 The SOAPBuilders mailing list and further information on the groups deliverables can be found at [http://groups.yahoo.com/group/soapbuilders/](http://groups.yahoo.com/group/soapbuilders/)
interoperability tests that vendors can run showing SOAP interoperability. Oracle ships the SOAPBuilders Round III test bed with Oracle Application Server 10g to demonstrate SOAP interoperability.

In contrast to SOAPBuilders, the WS-I approach to interoperability is prescriptive, through a specification called the WS-I Basic Profile. This document defines best practices and recommended approaches for using the base standards of SOAP 1.1, WSDL 1.1 and UDDI 2.0 to build interoperable business services. Like SOAPBuilders, WS-I is a broad consortium of key vendors including Oracle, IBM, Microsoft and Sun. The WS-I Profile also comes with a sample application illustrating its practices in action. Oracle provides this application, runnable on Oracle Application Server 10g, for download from the Oracle Technology Network to illustrate how its infrastructure seamlessly supports WS-I. In addition to the WS-Basic Profile, WS-I is also developing similar profiles for higher-level Web services standards such as security, orchestration and reliability.

OracleAS Web Services 10g is also designed to seamlessly deal with external vendor Web services extensions and platform idiosyncrasies. A simple example is how OracleAS Web Services 10g interoperates seamlessly between a common usage pattern within the Microsoft .NET platform called untyped SOAP messages and an equally common usage within J2EE platforms called typed SOAP messages. Correspondingly, OracleAS Web Services 10g transparently handles another Microsoft .NET SOAP and WSDL usage pattern called doc-literal-wrapped where an remote procedure call from .NET is represented as an XML document following a Microsoft specific convention.

SERVICE ORIENTED DEVELOPER PRODUCTIVITY

One of the important characteristics of service-oriented architectures is the wide variety of tools used by developers and system administrators. This is partly due to the standardization of the services layer but also due to the standardization of many underlying infrastructures to J2EE. A large, competitive tools industry has evolved around this standardization.

To improve service-oriented developer productivity, Oracle Application Server 10g provides a number of new enhancements. These include the following capabilities:

• **JDeveloper Integration** - JDeveloper provides a rapid application development environment based on industry standards.

  (i) For Java development, the latest J2EE APIs are supported including graphical tools for building Enterprise JavaBeans, JavaServer Pages

---

6 WS-I information can be found at http://www.ws-i.org
and Servlets.

(ii) For XML development, an visual XML Schema driven code editor offers guided XML editing and XML processing tools include an XML parser, XSLT Processor and XML Schema Processor.

(iii) For SQL development, a database browser enables inspection of any SQL92 compliant database and development of PL/SQL in Oracle databases.

(iv) For Web services development, a declarative design time provides publishing and consumption of J2EE applications as Web services using SOAP, WSDL and UDDI.

- **Command Line Tools and Utilities** - Oracle Application Server 10g offers a number of command line tools to configure the server, data sources, Web services and J2EE skeleton generation tools, a CMP mapping utility, application packaging and deployment tools, WAR and EAR file validation tools.

- **Apache Frameworks** - For developers who want to use J2EE Frameworks from the Apache community, Oracle Application Server 10g supports Apache Struts, Axis, Cocoon, Turbine, and Avalon

- **Integration with Third Party Tools** – Oracle Application Server 10g is integrated with best of breed tools and frameworks including the following:
  
  - **HTML Design Tools** - Macromedia UltraDev, Adobe GoLive, and Microsoft Frontpage
  
  - **UML Modeling Tools** – IBM Rational Rose, Borland Together ControlCenter
  
  - **Java Development Tools** - Borland JBuilder, Sun Forte, Quest JProbe, Empirix BeanTest, Pramati Studio, Computer Associates Cool:Joe, Blaze Advisor
  
  - **Monitoring Tools** - Mercury Interactive LoadRunner, Borland Optimizeit, Compuware OptimalJ, Quest JProbe

**GRID COMPUTING – MANAGING CAPACITY AND QUALITY OF SERVICE**

With one of the underlying intents of building out a service-oriented architecture being to increase the utilization of software assets, the emphasis on reliability, availability and scalability continues to increase in importance. New technologies, particularly in the area of grid computing are emerging to facilitate the ability to manage computing resource utilization while minimizing the management burden.

Oracle Application Server 10g, the next generation of Oracle's integrated software infrastructure for enterprise applications, has been designed to
enable grid computing. It has been designed to effectively pool together large numbers of low cost servers to create a virtual computing resource across which enterprise applications can be transparently distributed to use capacity very efficiently, at low cost, and with very high availability.

Any existing application that runs on Oracle Application Server 10g can transparently take advantage of grid computing without any changes. Service-oriented applications will find additional benefits when deployed in a grid. Oracle Application Server 10g provides a number of grid computing features, most importantly:

- *Reduce and eliminate excess computing capacity* through policy-based resource management; metrics-based workload management; and a variety of advanced back-up, disaster recovery, and clustered fail-over solutions to provide maximum availability in a grid.

- *Enable modular, inexpensive capacity growth* through automated installation, configuration, and software provisioning (including both software cloning and patch management) across hundreds of nodes in a grid.

- *Lower cost of management* and eliminating human errors in management through centralized systems monitoring, unified application server cluster management (including cluster monitoring, cluster optimization, and cluster-wide application deployment), and centralized identity management across a grid.

Figure 4 provides an overview of a large-scale deployment of the Oracle Application Server 10g which typically includes clusters of web caches, clusters of HTTP servers, clusters of J2EE servers and clusters of database servers (real application clusters – RAC) – all acting as a logical application server with quality of service unmatched in the industry. Managing this in a transparent manner where new capacity can be seamlessly added, removed and provisioned through a consolidated management infrastructure is a core design principle in the grid capabilities of Oracle Application Server 10g.
Figure 4: Oracle Application Server 10g Deployment Infrastructure

Capacity and Scalability

Oracle Application Server 10g provides the most scalable application server in the industry. The scalability of a system refers to how well it can respond as user demands increase and how efficiently it uses system resources to handle large volumes of users. With application servers, the primary scalability bottlenecks are CPU-constraints (where the system becomes constrained because it is CPU-bound) and memory-constraints (where the system becomes constrained because it is memory bound). These issues can be addressed by building into the infrastructure tools to manage and distribute resource utilization across existing hardware assets or by making it seamless to add hardware capacity.

Resource Pooling

Oracle Application Server 10g provides very efficient scalability supporting large numbers of users on a single CPU, supporting more users on a single Java VM, and allowing clusters of Java VMs to be started to handle more users or greater loads. There are four important ways in which it provides such scalability:

- **Single JVM - Threading Model** - Oracle Application Server 10g can maintain a hierarchy of J2EE containers in the following manner:

  (i) A single hardware instance can run multiple Oracle Application Server 10g instances;

  (ii) A single Oracle Application Server 10g instance can run multiple Java Virtual Machines (JDKs);

  (iii) A single Java Virtual Machine or JDK can start and maintain multiple threads;

  (iv) Each thread can run a single J2EE application module (JSP, Servlet or EJB).
Inbound Request/Connection Pooling - Oracle Application Server 10g maintains a configurable pool of socket connections to handle requests coming from a J2EE client, a Web server, a load balancer, or another J2EE application module.

Outbound Request/Connection Pooling - To eliminate database access overhead, OC4J pre-starts and maintains a pool of JDBC connections to a database. On specific requests, it selects/reuses an idle JDBC connection from the pool. OC4J provides fully JDBC 2.0 compliant connection pooling facilities.

Load Balancing

Load balancing essentially means how requests from clients can be distributed across multiple Oracle Application Server 10g instances on a single CPU or on multiple CPUs. Oracle Application Server 10g provides a number of advanced capabilities including:

- **Load balancing at HTTP server** - The Web server can load balance between HTTP server processes within a single instance of the service. Additionally, Oracle HTTP servers can be run on multiple nodes where client requests can be load balanced over the separate host instances using a variety of techniques including DNS round-robin or a dedicated hardware load balancer.

- **Load balancing at OC4J** - Web and EJB container instances load balance requests either across instances on a single node or across multiple nodes using a variety of load balancing algorithms.

- **Dynamic registration and load balancing** – Oracle Application Server 10g also provides a “dynamic clustering” architecture where components dynamically register themselves with the appropriate load balancing point when they are started or restarted and requests are dynamically routed to those instances via configurable load balancing algorithms.

- **Integration with third party load balancing products** – Oracle Application Server 10g has been certified with IP load balancing Appliances - Cisco Local Director, BigIP, and Alteon - for stateless and stateful load balancing. Best practices on how to configure these devices with Oracle Application Server 10g can be obtained from Oracle Technology Network.

- **Connection re-direction and node affinity** – Oracle Application Server 10g supports standard facilities such as cookies and dynamic URL-rewriting to bind and redirect clients to an existing session on a specific instance.

Clustering

Clustering is the coordinated usage of a group of Oracle Application Server 10g server instances to provide scalable, highly available services in a transparent manner. Clusters enforce homogeneity between member instances so that a cluster of application server instances function as a single virtual instance.
In the Oracle Application Server 10g clustering is provided at multiple levels – the HTTP Server, Web Cache, J2EE containers and backend database infrastructure. For J2EE and Web services clustering is configurable for JSP, Servlets, Web services, stateful session EJB, entity EJB, JMS topics and queues and JNDI.

In the case of stateful HTTP session objects for JSP, Servlets or Web services, or in the case of stateful business services based on EJB, clients are automatically routed to another instance in the cluster to which the session state has already been propagated to when an instance fails. In the case of clustered JNDI the JNDI tree structure is maintained cluster wide. In the case of JMS, clustering is available through the usage of Oracle Database as a backend JMS provider.

Figure 5 gives a conceptual perspective of the Oracle Application Server 10g clustering architecture for J2EE and Web services. Application server clusters represent groups of application server instances. Application server instances contain OC4J instances, which in turn contain OC4J processes. To share state, OC4J processes can be grouped into islands. This taxonomy gives Oracle Application Server 10g administrators great flexibility configuring their clustering environment.

Oracle Application Server 10g Cluster

Oracle Application Server 10g provides a set of clustering management services - Oracle Process Monitoring and Notification (OPMN) and Distributed Configuration Manager (DCM) - which are responsible for three important facilities:

(i) Maintaining a cluster-wide topology of live and failed instances for load balancing purposes;
(ii) Maintaining cluster-wide configuration information so that all cluster instances can be configured from one place and are configured identically; and,

(iii) Automatically restarting and synchronizing failed instances in a cluster.

The Oracle Application Server 10g clustering environment can be maintained either through command line configuration tools or through Oracle Enterprise Manager 10g Application Server Control.

**Hardware Capacity and Scalability**

When loads on the system grow beyond the capabilities of a single CPU, Oracle Application 10g supports two different hardware scalability models - horizontal scalability and vertical scalability.

- **Horizontal Scalability** - When loads grow beyond the capacity of a single CPU, multiple CPUs can be added. Multiple Java VMs can be started up on these CPUs, each JVM with one or more Oracle Application Server 10g instances. These instances can be clustered together. Requests can then be transparently distributed to across these CPUs, JVMs, and Oracle Application Server 10g instances. Oracle Application Server 10g can be deployed in a variety of different “horizontal scalability” architectures including commodity 1-4 CPU boxes being added in racks, blades and standalone servers.

- **Vertical Scalability** - Administrators can choose not to add more CPUs but to transparently migrate the application server to a higher end hardware configuration (if a system is CPU-bound) or add more memory (if a system is memory-bound). To support such vertical scalability, Oracle Application Server 10g is available on a broad range of hardware platforms scaling from low-powered desktop systems, low-end uniprocessor machines to high-end SMP clusters, and on 32-bit and 64-bit versions of major operating systems.

**Reliability and Availability**

Availability of an overall system or of an individual service is defined as the percentage of time that it works normally (or alternatively the mean-time-to-failure for the system) and how quickly and efficiently the system can recover from failures (or alternatively the mean-time-to-recovery for the system).

Clearly availability is important in service-oriented architectures and is a driving factor behind the emergence of grid computing infrastructures like Oracle Application Server 10g. To take enterprises to a higher level of reliability and availability, Oracle Application Server 10g is designed
around two key concepts: zero planned downtime and zero unplanned downtime.

**Zero Planned Downtime**

To reduce the need to take down a business service deployed on Oracle Application Server 10g in order to perform planned maintenance operations such as patching, upgrade, and maintenance on the application server itself, Oracle Application Server 10g provides a number of facilities:

- **Hot Deployment of Applications** - First, Oracle Application Server allows for hot deployment of applications - JSPs, Servlets, EJBs, and Web services - to an instance of the Application Server that is already running.

- **Dynamic Instance Reconfiguration** - Oracle Application Server 10g also supports the ability to dynamically reconfigure J2EE container instances - all configuration information is specified in XML files and application server instances do not need to be bounced when configuration changes are made.

- **Dynamic Cluster Reconfiguration** - When multiple instances belong to a cluster, configuration changes made to one instance are automatically replicated by DCM across the cluster.

- **Rolling Upgrade** - When an application is deployed to a cluster of Oracle Application Server 10g instances, any specific instance in the cluster can be taken off-line, modified, upgraded, and rejoined to the cluster without any application downtime - we call this “rolling upgrade of instances.” This allows planned maintenance operations to be performed on any application server instance without needing to shut down the application itself.

**Zero Unplanned Downtime**

To reduce unplanned downtime for business services, Oracle Application Server 10g provides two important facilities:

(i) **Zero Application Downtime - Longer Mean-Time-to-Failure** - Prevent applications from having any downtime even if any specific Oracle Application Server 10g instance or underlying system resource (CPU, JVM, network) has a failure. This significantly increases the Mean-Time-to-Failure for applications.

(ii) **Faster Failure Recovery - Shorter Mean-Time-to-Recovery** - They also ensure that in the event of a failure, the Oracle Application Server 10g instance can be recovered much more
quickly and efficiently decreasing the Mean Time-to-Recovery from faults.

The features bringing this capability to Oracle Application Server 10g are:

- **No single point of failure** - Oracle Application Server 10g can be deployed in an architecture that exposes no single point of failure. A hardware load balancer can send requests to any of multiple Oracle Application Server Web Cache or Oracle HTTP Servers. The Oracle HTTP Servers, in turn, can dispatch requests to any Web container instance. The Web container can in turn dispatch requests to any EJB instance. The EJB instances can in turn access the database across any database connection.

- **Automatic Connection Re-routing and Load Balancing** - For stateful applications, every tier of Oracle Application Server 10g can re-route connections from a “stateful” client to a pre-existing session on a specific Oracle Application Server 10g instance; or in case of failure, to another instance belonging to the same cluster.

- **Session Failover** – Oracle Application Server 10g provides failover of Servlet and EJB session state across Oracle Application Server 10g instances that belong to the same cluster through simple server side configuration. The session state for clustered components is replicated via IP-Multicast across instances in the cluster, enabling session recovery from unavailable server side instances.

- **Automatic Death Detection - Fast Restart Architecture** - Oracle Application Server 10g has integrated fault monitoring facilities to both detect the death or failure of a specific instance and restart it in order to minimize Mean-Time-to-Recovery for that instance. This facility is provided by Oracle Process Manager and Notification Service (OPMN).

- **Dynamic registration and transparent load balancing** - When a failed instance is recovered, the instance dynamically registers itself with the appropriate load balancing mount point and requests are automatically routed to it.

- **Transparent Application Failover (TAF) - database state management** - When used with an Oracle Database Server, Oracle Application Server 10g provides advanced state recovery facilities called Transparent Application Failover (TAF).

TAF essentially means that when an Oracle Database is deployed with a set of Oracle Application Server 10g instances, and an Oracle Database fails, the mid-tier Application Servers receive a notification from the database and can automatically fail over the JDBC
connection and route database requests to the “failed-over” database node.

Management and Monitoring

The entire Oracle Application Server 10g environment can be managed and monitored through a web-based administration interface, Oracle Enterprise Manager 10g. This consists of two components: the Oracle Enterprise Manager 10g Grid Control central console provides a single interface for monitoring distributed application servers and Oracle Enterprise Manager 10g Application Server Control for application server administration.

This graphical HTML-based environment uses a Servlet-based administration service to gather configuration data and performance metrics from all clusters, instances and components of Oracle Application Server 10g for presentation through an HTML-based graphical interface. By making all the management facilities accessible from a browser, administrators can manage Oracle Application Server 10g either locally or remotely. Oracle Application Server 10g Monitoring

Through integration with Oracle Enterprise Manager and the Dynamic Monitoring Service (DMS), Oracle Application Server 10g provides comprehensive monitoring metrics to administrators. These include resource metrics on CPU and memory utilization, JDBC metrics on connections and transactions, JSP/Servlet metrics on active sessions, requests being processed and request throughput and EJB metrics on number of methods active, number of requests and method throughput.

SUMMARY

Choosing a software infrastructure for a service-oriented architecture has been made dramatically easier by the emergence widely adopted, open industry standards like J2EE and Web services. Through a consistent dedicated vision, Oracle has developed and released Oracle Application Server 10g designed expressly to be the foundation of such an architecture.