An Oracle White Paper
April 2009

Application Grid: The Ideal Platform for IT Consolidation

Introduction

Today's economic environment is driving many enterprise IT organizations to *consolidate*, that is, to reduce the numbers of vendors of different technologies and to reduce the resources used by those technologies within their data centers. Consolidation means different things at different levels: it can be reducing the number of physical servers to run a given workload; it can be combining multiple data stores into a single larger storage facility; it can be the replacement of multiple applications of redundant functionality with a single all-encompassing solution. We look here at the opportunity for consolidation of *middleware*—Java application servers and related technologies.

Consolidating or reducing the number of vendors of a given technology can significantly improve efficiency by streamlining operations. For example, if an IT shop has more than one type of application server, each will have its own update schedule, patching procedures, and management practices. Consolidating to a single application server will bring simplification and economies of scale to each of these activities.

Consolidating or reducing the amount of resources used by a given technology brings near-linear reduction along many dimensions: the labor associated with management activities mentioned above, problem diagnosis, energy usage, etc. Such capacity consolidation can be achieved in a number of ways, including use of technologies that more efficiently utilize underlying resources as well as well as technologies that can be dynamically scaled with need rather than statically provisioned for the expected worst case.

Application grid is an emerging architecture for application server-level infrastructure that is ideally suited for consolidation. In particular, Oracle Fusion Middleware application grid technologies, including WebLogic Server, Tuxedo, Coherence, and JRockit, provide an excellent foundation for IT consolidation efforts.

"We have a lot of VIP users, and they expect our applications to always be up. We brought in the WebLogic stack—we were really trying to consolidate and gain some efficiencies. [The WebLogic platform] runs almost every core business within our studio. It's really helped us in terms of cost, efficiency, manageability, and performance."

David Buckholtz, VP Planning, Enterprise Architecture, and Quality, Sony Pictures Entertainment

Application Grid Overview

For the last decade or so, foundation software for enterprise applications has been dominated by the notion of a "container", a prepackaged set of low-level functions and services that applications need to run. For Java applications the container is the *application server*; for C, C++, and COBOL applications the equivalent is referred to as a *transaction processing monitor* (TPM) for historical reasons. The container is a "run time" entity—supportive software that an application or component requires when it is started up. This is in contrast to earlier practices where the functionality equivalent to today's containers was packaged into libraries and compiled into the program itself at "build time", resulting in a monolithic entity that was run directly on an operating system (OS) at run time.

Traditional World: Dedicated Stacks

Factoring low-level services into a run-time container was a massive improvement over earlier compilation practices because it decoupled the program logic from the lower-level services, allowing them to be changed and managed separately. However, one potential area for improvement that wasn't immediately exploited was consolidation or sharing at the container level. For the most part, applications and application components continued to be hosted on their own dedicated stacks, where each component ran on its own instance of a container with its own respective instances of OS, hardware, etc.

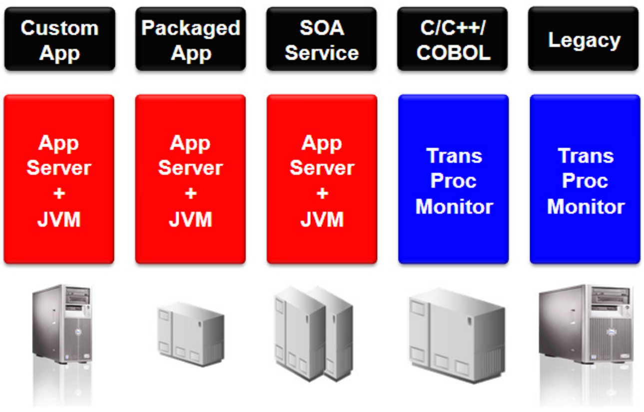

Figure 1. Traditional application infrastructure with dedicated stacks.

Grid Computing

Grid computing is a general architectural approach in which compute resources are decoupled from the demands on those resources. The resources are pooled and shared, and their allocation across demands can be adjusted dynamically. In addition, emulating the structure and behavior of the electricity power grid at the heart of the grid metaphor, a grid computing architecture typically employs redundancy and fail-over such that failures in individual resources or allocation adjustments are hidden from the resource consumers. A grid computing architecture can be created at various levels in computing infrastructure—a grid of physical servers, a grid of databases, a grid of storage facilities, etc.

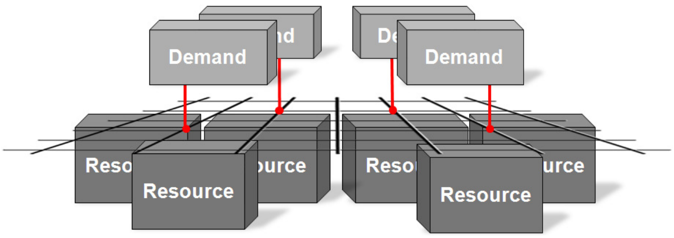

Figure 2. Grid computing as pooled, shared resources with dynamically adjustable allocation across demands.

Application Grid

Application grid refers to applying grid computing at the level of middleware, specifically, at the level of the container. By pooling and sharing resources at the container level and dynamically adjusting their allocation, infrastructure can be utilized more efficiently by applications while at the same time providing higher reliability, performance, and scalability than traditional dedicated stack architectures.

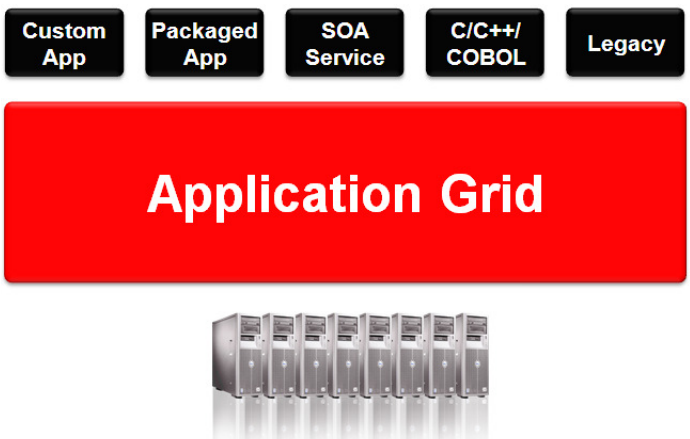

Figure 3. Application grid as the new architecture for enterprise application infrastructure.

Container Clustering as the Basis for Application Grid

The primary mechanism that enables grid-like sharing and adjustment at the container level is *clustering*. Application servers such as Oracle WebLogic Server and TPMs such as Oracle Tuxedo have clustering capabilities that allow a single application component to run on multiple instances of the container, grouped in a cluster. Work is load-balanced across cluster instances. "Nodes", or instances in a cluster, may be added to or removed from the cluster. A particular container's clustering capabilities determine how dynamically and how automatically clusters can be adjusted.

The figure below shows how clustering of multiple applications can be used to achieve grid efficiency, reliability, scalability, and performance. Two applications share five servers, one in a cluster of three nodes and the other with two. Load on the first application subsides, so its cluster is reduced to two nodes, freeing up a server resource. Load on the second application then rises, so a new application server instance in started on the free machine and added to the cluster as a new node.

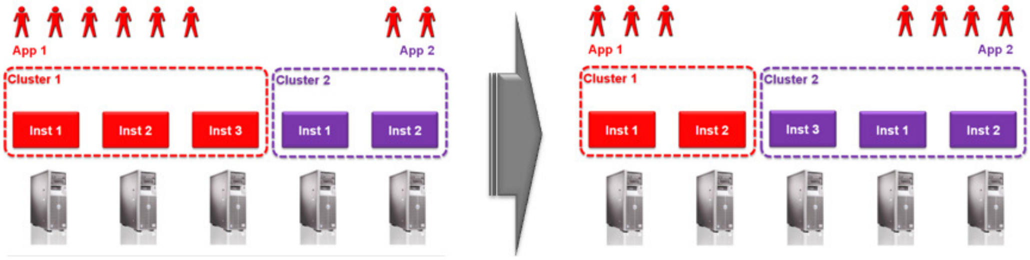

Figure 4. Container clustering as the core mechanism for efficient hardware utilization in an application grid architecture.

Application Grid and Consolidation

The application grid architecture is particularly well-suited to be the middleware foundation for consolidation efforts. As outlined above, the grid approach fundamentally reduces the amount of hardware needed to run a given set of applications when compared with the traditional dedicated stack approach. This is because the hardware required for the dedicated stack approach is the sum of all the applications' worst-case needs, whereas grid's shared resource approach means that the hardware is less than that sum. In the example above we have two applications, each with worst cases of three servers, for a total worst case of six; by using an application grid approach we can use 5, representing a 17% reduction. With larger numbers of applications and larger peak/off-peak variance, the reduction can often be greater than 50% and approaching 80% in extreme cases.[1]

Note that reducing the amount of hardware used per application is not simply a matter of lower up-front costs for purchasing hardware. There are also significant implications for ongoing maintenance. The fewer servers in use for a given application load, the lower the ongoing operational costs—the labor involved in upgrading, patching, and diagnosing hardware, OS instances, etc.

Oracle Fusion Middleware as the Foundation of Choice

If application grid is the right approach to structuring middleware for consolidating usage of underlying resources, the next consideration is what technologies to use in implementing application grid. Part of any consolidation effort is likely to be a reduction in the number of vendors for each technology type (ideally to one vendor per type). In choosing a vendor for application grid, facets to consider are: the merits of each technology in its own right, the synergies across the technologies, and how well the products implement application grid.

Foundation Consolidation Requirements

The core technology that forms the middleware foundation is the container—the application server in the Java case, and the TPM in the C/C++/COBOL case. Complementary to container are several other technologies. A Java application server requires a Java virtual machine (JVM), which can be an important differentiator for particular application grid characteristics. An in-memory data grid technology, which is a superset of distributed caching functionality, can significantly enhance both application server- and TPM-based applications.

For most enterprise applications, key requirement areas for container-related technologies include:

- Correct (latest) versions of all relevant APIs (e.g. Java EE APIs)
- Reliability
- Availability
- Scalability

[1] If we have ten applications that each need ten servers at peak load but only one server during off-peak, if peak load only happens 10% of the time for each application, and there is no overlap of peak between any two applications, then dedicated stacks would require 100 machines whereas application grid would only require 19, an 81% improvement.

- Performance
- Clustering
- Management/automation

Oracle Fusion Middleware Application Grid Technologies

Oracle Fusion Middleware includes four main products for implementing application grid. The core container technology for Java is WebLogic Server. For C, C++, and COBOL applications the equivalent is Tuxedo. In-memory data grid technology that can be used with either WebLogic Server or Tuxedo is Coherence. Both WebLogic Server and Coherence run best when run on the JRockit JVM, which has a real time variant called JRockit Real Time for extreme performance applications. Finally, the necessary application grid management and automation functionality is part of Enterprise Manager, Oracle's comprehensive and unified management technology for the entire stack including database, middleware, and applications.

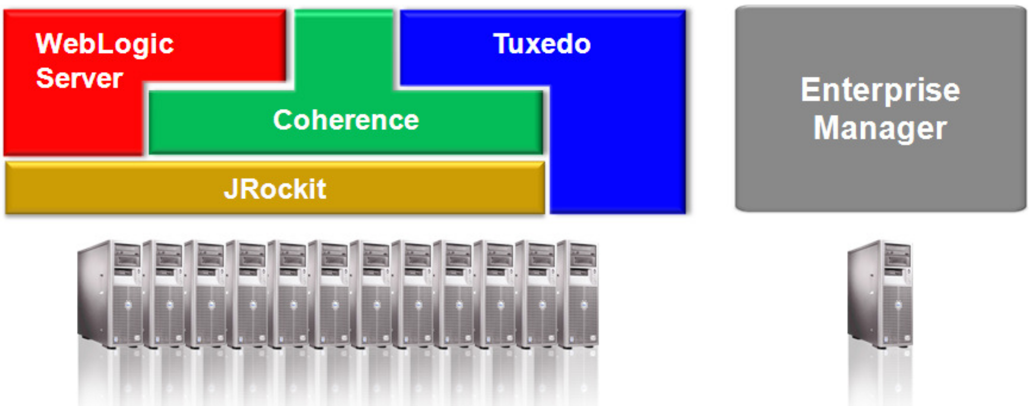

Figure 5. Oracle Fusion Middleware application grid technologies.

WebLogic Server

WebLogic Server is the industry's leading Java Enterprise Edition (Java EE) application server, holding numerous performance world records, commanding a large share of the market, and running many of the Fortune 500's most mission-critical applications. WebLogic Server is frequently the first commercial application server to bring to market updates to any of the Java EE APIs. As Oracle's strategic application server technology within Fusion Middleware, WebLogic Server has an extremely strong roadmap and future. For all these reasons, it is an application server technology that any IT decision maker can be confident in choosing as a single platform on which to consolidate.

"What really distinguished Oracle WebLogic Server for us was its reliability, fault tolerance, and failover support. Every minute of downtime represents a monetary value, so that's extraordinarily important to us."

"After converting to WebLogic, we saw a reduction in CPU footprint of 40%, allowing us to postpone new hardware acquisition by 9-12 months."

Director of Engineering, Consumer-facing Web service in top 1% of U.S. sites by volume

Specifically as a platform for application grid, WebLogic Server offers numerous advantages over alternatives. Its clustering capabilities are second to none, with unique features such as rolling upgrade, automatic service and whole server migration, and the ability to add and remove nodes to and from live clusters. Extensive scripting capabilities as well as integration with Enterprise Manager mean that WebLogic Server clustering is not only extremely dynamic but also highly automatable—key enabling features for application grid.

JRockit

The JRockit JVM plays a significant role in helping WebLogic Server achieve its world record benchmarks, and it also in its own right holds numerous world records for pure JVM performance. A variant of the JRockit JVM called JRockit Real Time addresses the unpredictability of response times for typical JVMs caused by nondeterministic memory clean-up. JRockit Real Time can guarantee clean-up ("garbage collection") pause times to be under one millisecond. This real-time operation requires no code changes—simply swap JRockit Real Time in place of a standard JVM and get deterministic, predictable behavior. Both the standard JRockit JVM and JRockit Real Time have unmatched instrumentation capabilities, allowing significant metrics related to memory usage and responsiveness to be captured and transmitted with near-zero overhead, thus supporting applications running live in production. These metrics are important input to application grid management, enabling optimum performance and service level maintenance as well as efficient resource utilization.

Coherence

The Coherence in-memory data grid technology allows the memory of multiple physical servers to be used as single, "seamless" memory space for storing data objects. Data objects may be replicated across multiple nodes in the grid and/or distributed ("partitioned") among nodes, depending on particular data characteristics and performance and availability requirements. Both performance *and* reliability can be significantly enhanced by holding objects "in memory" in a Coherence distributed cache. Coherence is unique among data grid technologies in that it has a pure peer-to-peer architecture, with no single point of failure.

"We saw an increase in responsiveness of 100% for search queries."

Ian Robertson, Chief Architect, Overstock.com

Coherence is also uniquely enabling for application grid in that nodes can be added to or removed from a live Coherence data grid (cluster) with near-instant automatic repartitioning and re-optimization of the data objects across the cluster. Because the capacity constraint on many applications is memory rather than processing, scaling of the Coherence cluster rather than the application server or TPM cluster can be a highly dynamic and resource-efficient way to adjust capacity within the application grid.

Tuxedo

Tuxedo can be thought of as the C/C++/COBOL equivalent to Java's application server and JVM combination. Tuxedo was originally developed at Bell Laboratories over twenty years ago and runs many of the world's most mission-critical, high-volume transactional systems, ranging from electronic funds transfer to travel reservations to package tracking. Tuxedo is often used in mainframe modernization projects where legacy application code is migrated in a very straightforward way from a costly mainframe to Tuxedo clusters running on distributed, mainstream servers. Tuxedo provides an outstanding platform for consolidating legacy mainframe applications as well as other C, C++, or COBOL components.

For application grid, Tuxedo provides highly dynamic and automatable clustering. A comprehensive scripting environment as well as extensive management capabilities that are integrated with Enterprise Manage complete Tuxedo's application grid picture.

Enterprise Manager

Spanning applications, middleware, and database management, Oracle Enterprise Manager's unique "top-down" approach enables IT departments to focus on what matters to the business—greater agility, better service quality and lower operational costs. Oracle Enterprise Manager enables its customers to manage their applications from a top-down perspective—from monitoring service levels to proactively isolating business exceptions before they escalate, and remediating issues at any level of the IT stack.

Enterprise Manager Diagnostics Pack for Oracle Middleware

Most application performance problems surface during peak loads. Often times, these problems are time- and resource-intensive, if not impossible, to reproduce in test environments. Application administrators need solutions that not only monitor production applications but also provide intelligence to help diagnose problems early and avert emergencies. Oracle Diagnostics Pack for Oracle Middleware provides proactive monitoring and advanced diagnostic capabilities that empower administrators to prevent crashes and other undesirable outcomes in high load production environments.

Benefits of the Diagnostics Pack include:

- Manage and monitor multiple Oracle WebLogic Server domains and Oracle Application Server farms from a single console
- Optimize the performance of Java applications using real-time events, notifications, reports and historical trends of metrics
- Improve availability and performance of Oracle JRockit JVM environments with low-overhead application monitoring and diagnostics
- Reduce performance problem resolution for Java environments without instrumentation overhead and no server restarts
- Simplify the determination of root causes between database and application servers using transaction tracing
- Rapidly detect memory leaks using differential heap analysis

Enterprise Manager Configuration Pack for Oracle Middleware

As IT infrastructures continue to expand, configuration management has become one of the most critical components of day-to-day IT operations. Put simply, failure to effectively control application and system architecture erodes the value of technology investments. Oracle Configuration Management Pack for Oracle Middleware provides comprehensive functionality to manage configurations and automate IT processes. A key component of this solution is the Configuration Change Console, which reduces cost and mitigates risk by automatically detecting, validating, and reporting authorized and unauthorized configuration changes in real time.

Benefits of the Configuration Pack include:

- Automate discover and asset tracking of middleware configuration and its underlying host and operating system
- Optimize configuration searches using out-of-the-box and customizable search-and-compare features
- Improve configuration management through historical change tracking
- Quickly and accurately assess policies and compliance initiatives
- Streamline configuration management through integration with change management systems
- Detect configuration changes in real time to confirm anticipated changes and minimize the impact of unauthorized actions
- Ensure compliance with regulatory and industry standards such as Sarbanes-Oxley

Customer Example: Sony Pictures

Sony Pictures Entertainment is a subsidiary of Sony Corporation of America, a subsidiary of Tokyo-based Sony Corporation. Sony Pictures produces and distributes film and other types of digital content in 67 countries around the world.

Business Challenge

Because of the specialized requirements of the entertainment industry and the limited range of packaged applications, Sony Pictures custom-builds most of their solutions. The use of multiple tool-sets and technologies, including open source, across different development teams and departments led to a spiral of increasing complexity and costs to operate the expanding set of solutions. In addition, applications must meet stringent performance and availability criteria in order to satisfy the Sony Pictures' highly demanding VIP user community.

Solution

Sony Pictures now standardizes on WebLogic Server as the common platform for Java application development projects and centralizes IT operations for their application server, database, and content management environments. This allows Sony Pictures to operate the solutions much more efficiently and at much lower cost, without compromising the high service levels demanded by users.

Results

Over fifty business-critical Java applications are currently deployed on hundreds of Oracle WebLogic Server instances. They operate in a centralized environment administered by a five-person team. The standardization on WebLogic Server has allowed Sony Pictures to achieve the economies of scale required to release and integrate new applications faster, with better quality, and at lower total cost.

Conclusion

Many enterprises are undertaking substantial IT consolidation efforts to increase efficiency and reduce costs. There is much opportunity for consolidation of middleware, particularly at the foundation level of application servers, TPMs, and related technologies. By employing an application grid approach and architecture, not only can the highest possible consolidation of hardware resources be achieved, but performance and reliability can also be significantly improved. Oracle Fusion Middleware foundation technologies, including WebLogic Server, JRockit, Coherence, and Tuxedo enable the most effective application grid and provide the best overall platform for middleware consolidation.

Appliclation Grid: The Ideal Platform for IT Consolidation
April 2009
Author: Mike Piech

Oracle Corporation
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

Worldwide Inquiries:
Phone: +1.650.506.7000
Fax: +1.650.506.7200
oracle.com

Oracle is committed to developing practices and products that help protect the environment

0109