JBoss by Red Hat

RED HAT CONSULTING
STRATEGIC MIGRATION PLANNING GUIDE
WEBSPHERE TO JBOSS ENTERPRISE APPLICATION PLATFORM

CONTENTS

1. EXECUTIVE SUMMARY 3
Pre-planning 4
The migration planning process 4

2. MIGRATION CONSIDERATIONS 4
Migration drivers 4
Potential migration scenarios 5
Configured services 7
Migration deployment scenarios 9
Hardware migration 14
Other considerations 19

3. THE STRATEGIC MIGRATION PROCESS 20
Migration process overview 20
Phase I: Server architecture and deployment model 21
Phase II: Application migration assessment 21
Phase III: Effort and risk assessment 23
Phase IV: Server and application migration plan 25
Phase V: Migration implementation 26

4. ENTERPRISE SERVICES 26
Middleware consulting services 26
Platform consulting services 27
Training 30

5. SUCCESSFULLY MIGRATED CUSTOMERS 32
Union Bank 32
GEICO (Government Employees Insurance Company) 35
Glöckle 37
iGATE 39
Allianz 43

6. SUMMARY 46

1. EXECUTIVE SUMMARY

IT organizations are constantly faced with the challenge to produce high-quality solutions with a lower total cost of ownership (TCO). With the growing recognition that open source software provides quality, stable solutions, migrations of existing enterprise applications to products such as JBoss® Enterprise Application Platform have become increasingly popular.

JBoss Enterprise Application Platform provides a holistic solution that includes tools for such business challenges as business process management, enterprise application integration, and service-oriented architectures (SOA), enterprise portals, and data services solutions.

Migrating from proprietary technologies to JBoss Enterprise Application Platform offers many other advantages to IT organizations, including cost reduction versus proprietary JEE implementations, scalability, and the ability to extensively customize the server. Other advantages include:

Are IBM licensing costs a disproportionately large part of your IT budget? Do your development and operations teams spend an ever-increasing percentage of their time dealing with the complexity of IBM software? Are business solutions and innovation taking a back seat to navigating the maze that is IBM installation and configuration?

Many former WebSphere clients have found that by migrating to JBoss® Enterprise Middleware, they can spend less time and money on their server platforms, allowing them to focus resources on business problems, where they truly belong. Integrated, simplified, and delivered by the leader in open source software, JBoss Enterprise Application Platform includes leading open source technologies for building, deploying, and hosting enterprise Java applications and services.

Migrating to JBoss Enterprise Application Platform offers many advantages to IT organizations. With JBoss you can:

- **Innovate with stability.** Leverage market-leading technologies for building and hosting next-generation Java applications and web services.
- **Improve developer productivity.** Use integrated Web 2.0 technologies that work with the leading application server from the start, remove configuration complexities, and simplify Java development.
- **Enjoy enterprise-class performance.** Take advantage of the integrated clustering and high-availability features, and utilize a proven foundation for mission-critical applications with failover, load balancing, and distributed deployment features used to deploy large, scalable enterprise applications.
- **Extend budgets.** Improve application investment returns and savings by leveraging an affordable middleware solution that includes everything needed for enterprise Java applications.
- **Use open standards.** Avoid vendor lock-in while obtaining the transparency that allows employees to see exactly what's occurring in the platform on which they are developing.

A successful and smooth migration to JBoss requires a strategic level of up-front planning and a proper understanding of the issues and challenges involved in order to minimize migration risks and costs. In this document, we attempt to cover the most important items and provide readers with some of the resources and references necessary to be successful.

JBoss uses the community version of its projects, following a "release early, release often" model that promotes innovation and is most responsive to user requirements. Appropriate versions of open source projects are subsequently integrated and put through a rigorous certification process to produce enterprise-class solutions, including JBoss Enterprise Application Platform.

A successful and smooth migration to JBoss requires a strategic level of up-front planning and a proper understanding of the issues and challenges involved in order to minimize migration risks and costs. In this document, we attempt to cover the most important items and provide readers with some of the resources and references necessary to be successful.

PRE-PLANNING

A thorough understanding of your migration environment is the critical first step to start realizing the benefits of moving to JBoss. Your organization's motivations for undertaking a software platform migration should be carefully considered, as these may influence choices, opportunities, and trade-offs. Likewise, understanding your potential deployment scenarios will help you proactively identify any roadblocks and anticipate future needs.

THE MIGRATION PLANNING PROCESS

Red Hat has established a proven five-step process designed to identify migration opportunities, examine the risks associated with various migration scenarios, create a standard enterprise environment, and develop a comprehensive and strategic migration plan for the enterprise.

Through this process, your organization will:

1. Examine the existing application server and platform and determine the equivalent capabilities in a JBoss EAP ecosystem.
2. Examine proprietary features and native capabilities and determine the equivalent capabilities in JBoss EAP.
3. Measure organizational readiness and overall migration risk.
4. Develop a strategic migration plan, including a detailed roadmap and cost estimate.
5. Implement the strategic migration plan and employ implementation support strategies.

The details that follow are intended to provide insight into the considerations and processes required to move to the JBoss Enterprise Application Platform. We encourage you to share this with your team as you embark on your migration planning. Through these insights, we hope to arm you with the knowledge to successfully plan and execute your migration.

2. MIGRATION CONSIDERATIONS

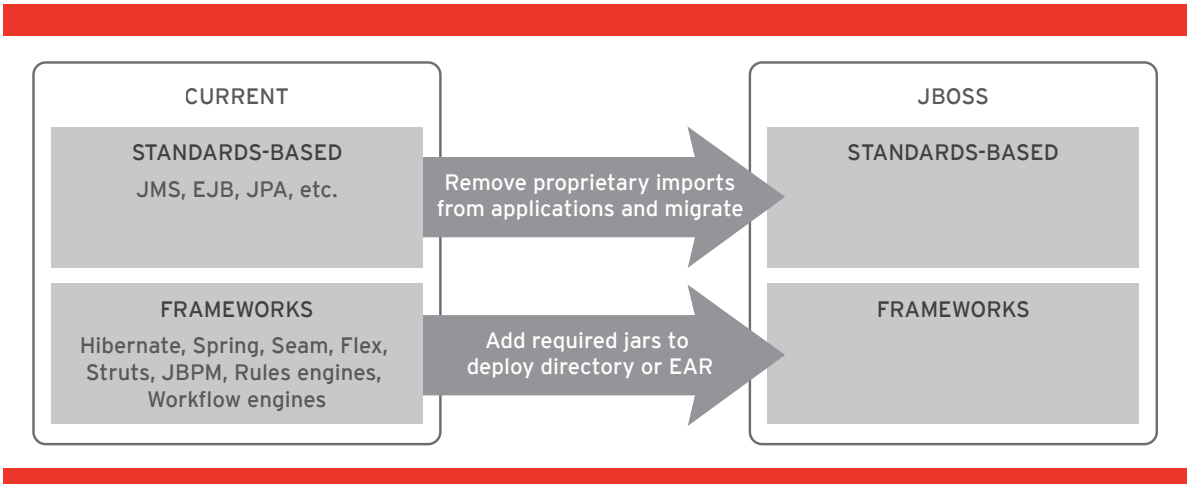
When a company decides to migrate from one application server to another, there may be several possible drivers involved. Common drivers deal with cost, increasing development innovation, or combining datacenters with different technologies.

MIGRATION DRIVERS

- Cost
  - Gain the ability to distribute server with application to customers
  - Eliminates per-CPU licensing
  - Allows the use of cheaper hardware and more robust clusters
  - Lowers support costs
- Expansion of server farms
- Conformance to standards
- Integration and customization

- Flexibility
- Failure of outsourcing
- Mergers and acquisitions
- Datacenter consolidation
- Adoption of rapid development techniques
- Desire for latest features and capabilities
- Performance
- Scalability

Whatever the particular drivers are, it is important to bear them in mind when planning migrations. Drivers help dictate priorities and methods when migrating. They are key to determining which migration scenario applies and how it should be deployed. For example, if a major driver is the desire to benefit from the latest features and capabilities, the migration project will need to delineate the features desired and determine which can be achieved by changing configurations and which will require updating source code.



POTENTIAL MIGRATION SCENARIOS

BASIC WEB APPLICATION MIGRATION

Migrating basic web applications that were developed according to the servlet specification and Java EE specifications is the easiest and most cost-effective way to start migrations. It can be an excellent measure of the effort involved. If the applications were developed with a standards-based Integrated Development Environment (IDE) such as Eclipse, this will go quite smoothly. Many proprietary IDEs embed links to proprietary libraries and thus can add additional challenges to migration.

Standard web applications include a deployment descriptor called "web.xml." Forgiving behavior by application servers often leads to the use of syntax and notations that are not portable. Red Hat tries to document such examples and has a ready list of changes that are often needed when migrating from specific application servers to JBoss solutions.

Most application servers also include a second proprietary deployment descriptor for web applications to support value-added features that are not included in the standard specifications. This second descriptor is called jboss-web.xml for JBoss, while WebSphere Application Server uses two files called "ibm-web-ext.xmi" and "ibm-web-bnd.xmi" for this purpose. Red Hat often maintains XML style sheets that can be used to automatically transform other vendor descriptors to the JBoss-supported "jboss-web.xml," where features and capabilities overlap.

Migrating a web application that is developed and deployed to WebSphere can be very simple or somewhat complicated depending on how proprietary the application architecture and dependencies are. Many web applications can simply be copied over and directly deployed to JBoss with no changes at all. Some of the common barriers to migrating web applications can include:

1. When deployed as an exploded directory, a .war extension for the directory may be required when deployed to JBoss. A similar naming convention in WebSphere is uncommon.
2. WebSphere web applications often bundle their proprietary secondary deployment descriptor. This descriptor can often include no relevant configuration information whatsoever and would, therefore, be ignored and not cause any harm under JBoss. When it provides a relevant configuration, it is very often comprised of simple items such as the context root of the web application or security settings for it, which are similarly provided in a jboss-web.xml file under JBoss.
3. Web applications may use libraries provided by WebSphere, including but not limited to JSP tag libraries and WebSphere helper classes. This is where the migration effort can potentially become substantial, depending on the exact nature of the library being used and what it takes to replace it with a standard open source alternative.

JAVA EE MIGRATION

Migrating Java EE applications is in theory no more difficult than migrating pure web applications. However, the purpose of Java EE specifications is to facilitate the integration of multiple frameworks with diverse functionalities. This means that there will in practice be many mini-migrations involved. We will discuss these more in the JSR migration section.If proprietary IDEs are used in the development of Java EE applications, these will often have the generated classes extend and implement proprietary classes and interfaces. The JBoss MASS project provides a migration analysis tool called MAT, which scans the application's source code for proprietary libraries and imports. This provides invaluable information for estimating the amount of work that needs to be done for a Java EE migration.

The tool is hosted at **http://www. jboss.org/mass/MAT.html**.

JBoss MAT also produces detailed HTML reports covering server configuration, deployed applications, and class dependencies.

ADVANCED JSR MIGRATIONS

Java Specification Requests (JSRs) represent the Java community's effort to provide standards in additional areas such as:

- Portal
- ESB
- Rules engine
- Workflow engine
- Injection

		APPLICATIONS TO MIGRATE				
		App 1	App 2	App 3	App 4	App 5
Applications	App 1		✓	✓		
	App 2				✓	
	App 3		✓		✓	
	App 4			✓		✓
	App 5					
Libraries	Jakarta Commons 3.2	✓	✓	✓	✓	
	Apache	✓		✓		
	etc.					
Framework	Hibernate 3.1				✓	
	Hibernate 3.4	✓	✓	✓	✓	
	Spring 2.1		✓			
	Oracle DB 10g	✓	✓	✓	✓	
	etc.					

There are different migration paths for each separate JSR. Depending on how complete the specification is, migration can be fairly simple or very involved. Generally, annotations are more standard than XML implementations.

CONFIGURATION MIGRATION

In broad strokes, a WebSphere Application Server (WAS) environment is organized into a series of cells and nodes. A node is defined by a common configuration repository and a single profile. Multiple WAS instances can belong to a single node; however, a WAS instance cannot belong to multiple nodes. They can be managed simply by an administrative agent, which a node registers itself with. In a more sophisticated environment, a deployment manager is used to efficiently administer, manage, and provision nodes. Lastly, a job manager can be used to asynchronously submit administrative jobs to deployment managers and administrative agents.

CONFIGURED SERVICES

WebSphere Application Server has the concept of installed applications and configured resources, scoped at an applicable level. Resources in the WebSphere terminology can include JDBC data sources and JMS destinations, which are considered deployable artifacts in JBoss. This leads to two very distinct models.

> With **WAS**, services such as JDBC and JMS are configured through the administration console by the WebSphere administrator. This is true in both production and development, though the developer may also be acting as the WAS administrator in a developer environment. While such resource configuration ultimately depends on application needs, it is both physically and practically separate from the application itself. Application developers often document their service dependencies and administrators follow the document to set up the required services for the application to function properly.

With **JBoss**, each service (for example, a JDBC connection pool or a JMS queue) is configured in its own XML file. Two or more services of the same type can often be configured in the same XML file, if that is preferable. This gives users flexibility in their configuration choices. The same configuration practices administrators would use with WebSphere can be followed—configuring services based on provided documentation. But it is also possible for the development team to include these deployable XML artifacts with their application or even bundle them in the same folder or archive. This second model leads to a self-contained application that includes its own configuration dependencies.

JDBC

In WebSphere version 7.x, each JDBC resource is configured in the resource's XML file corresponding to the Profile and the scoping level. The typical pattern is that a folder will be created for a given profile under WebSphere/AppServer/profiles with the chosen profile name. This folder will contain a config directory, which in turn will have a hierarchy of directories corresponding to the scope that the data source has been targeted to. For example, if the data source is targeted at the cell level, a directory named "cells" will contain a directory for that cell, which will in turn have a resources.xml file describing the data source. A data source defined under a JDBC provider will appear in the resources file as a "factories" element having the type "resources.jdbc:DataSource", itself embedded in a "resources.jdbc:JDBCProvider" element. Further configuration properties are included in the XML as resource properties in a property set, which the connection pooling described in a separate connection pool element.

In JBoss, it is enough to create a datasource XML snippet for each connection pool which may be provided in its own poolName-ds.xml file or combined collectively in the same *-ds.xml file. The content of such a datasource XML file in JBoss will depend on the transactional behavior of the datasource as well as the connection details. Most—if not all—of these details can be found in the matching *-jdbc.xml file under the config/jbdc directory of the WebSphere domain. The database password will be encrypted and therefore such a migration cannot be fully automated. When the proper *-ds.xml files are prepared, they can simply be deployed to the matching JBoss server or cluster to provide an equivalent JBoss environment.

JMS

In WebSphere 7.x, various JMS components are configured separately in the administrative console, with each specifying its dependency at the time of creation. For example, JMS providers are created and configured in a separate section from destinations such as queues and topics, yet in the XML file where this configuration is reflected, destinations appear under the applicable provider. This information is stored in a resources XML file corresponding to the Profile and the scoping level. The typical pattern is that a folder will be created for a given profile under WebSphere/AppServer/profiles with the chosen profile name. This folder will contain a config directory, which in turn will have a hierarchy of directories corresponding to the scope that the JMS component has been targeted to. For example, if a queue is targeted at the cell level, a directory named "cells" will contain a directory for that cell, which will in turn have a resources.xml file describing the JMS queue. A queue tied to a JMS provider will appear in the resources file as a "j2cAdminObjects" element under the JMS provider, itself described as a J2CResourceAdapter element.

In JBoss, JMS connection factories are typically configured in the deploy/jboss-messaging.sar/connection-factories-service.xml file of a JBoss server but can generally be configured as part of any service archive deployment. It is more typically the destinations themselves, the queues and topics, which need to be configured and deployed to support each application. In JBoss, it is enough to create a JMS Destination XML snippet for each queue or topic, which may be provided in its own destination-service.xml file or combined collectively in the same *-service.xml file. The content of such a destination XML file in JBoss will resemble

that of any MBean deployment file, and the MBean code will be QueueService or TopicService as the case may require. Such an XML file would be deployed to the JBoss server or cluster, possibly by dropping it in the server's deploy directory or one of the cluster members' farm directory.

CLUSTERING

While the concepts and the goal remain very similar, there can be significant differences in the way clustering is configured in WebSphere and JBoss.

WebSphere provides a very static view of clustering, both in terms of cluster membership and configuration details. There are limited configuration options and clustering largely remains a black box. This leads to a level of simplicity that works well for some clients but can cost others with more exceptional requirements. The cluster itself is statically defined on the administration server with provided addresses of each member.

In JBoss, a cluster is a very dynamic concept. It is defined by a cluster name and an address, which may be a multicast or unicast address. Any JBoss instance declaring an intention to join the cluster can do so by communication with that address and by providing the cluster name. This results in a very fluid cluster definition that works very well with dynamic provisioning scenarios. Furthermore, clustering in JBoss is built on top of JGroups, which itself is highly configurable and has configuration parameters that are exposed to cluster users. A wide variety of network topologies can be supported and the ability to fine-tune the cluster performance is virtually endless.

CACHE REPLICATION

In a cluster, a number of caches are provided and must be kept in sync to avoid data integrity issues. The most prominent ones are the HTTP session and entity bean caches. Once again, WebSphere takes a black-box approach to cache configuration while JBoss builds them on top of open stacks that are exposed and highly configurable.

JBoss Cache is used to provide a distributed transactional cache for entity beans and the HTTP sessions. A wide array of configuration options can help tailor the cache behavior to user requirements. Cutting-edge technology such as non-serialization-based replication through AOP is available through the POJO cache features. The communication protocol is also configurable, and synchronous replication, asynchronous replication, and invalidation are all supported.

MIGRATION DEPLOYMENT SCENARIOS

HIGH THROUGHPUT OR LOW LATENCY

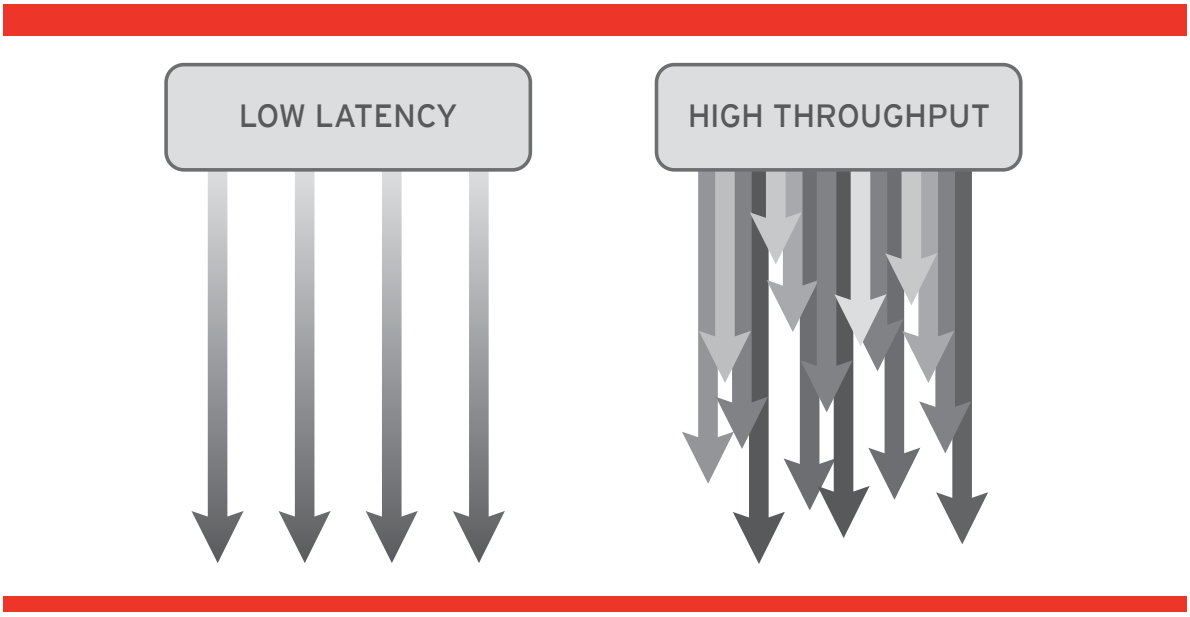
Servers deployed for high throughput are configured very differently than low-latency application servers. It is important to separate applications that require low latency from those that require high throughput. The configuration of timeouts, max threads, and other basic options are at the opposite end of the spectrum from each other and will not mix well.

When targeting high throughput, it is the overall performance of the system that is prioritized—client requests are not given individual attention. As an example, for a system with only four processor cores, one might determine that 200 concurrent requests mark the optimal point of parallel processing with an average processing time of one second. Anything below this number would be under-utilizing the system, while a higher number of concurrent requests results in wasteful context switching and hurts the overall throughput.

Configuration based on throughput would, therefore, limit concurrent requests to 200. In the event of 300 concurrent requests, we would queue up the remaining concurrent requests and process them as a batch consecutive to the first group. In this oversimplified analysis, requests that are part of the second batch will take almost twice as long to process as the first batch.

This runs counter to prioritizing low latency and the requests from the second batch may well be violating our service-level agreement (SLA). Should low latency be our priority, we could configure the system to accept 300 concurrent requests.

In our hypothetical example, it is conceivable that average processing time would then grow to 1.7 seconds. Requests from the second batch would complete in 1.7 seconds instead of 2 seconds and would therefore meet the SLA. The throughput, however, has gone down from 200 tps to 176.5 tps and even with a constant flow of 300 concurrent requests, the system will process fewer requests in the long-term.



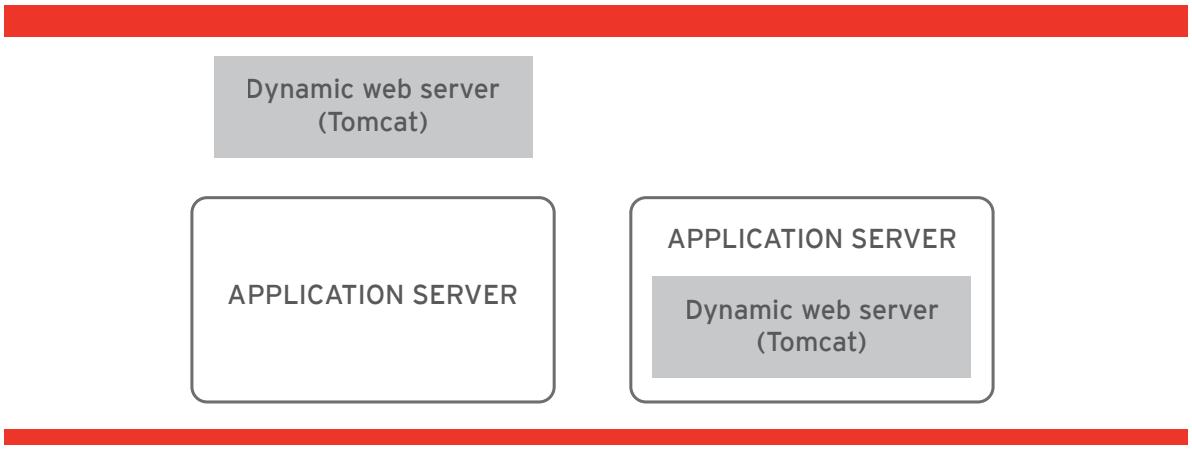
JBoss provides a robust platform for building out large-scale server farms that can support tens of thousands of concurrent requests.

PHYSICAL DEPLOYMENT MODELS

- **Physically separate machines for web and applications servers**

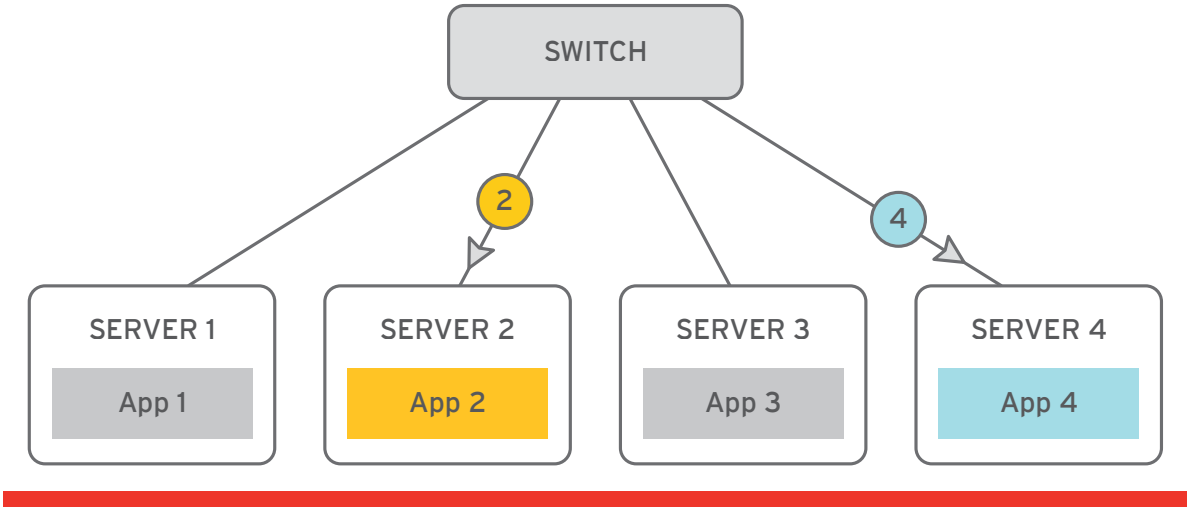
  Some middleware server companies advocate using separate servers for web servers (servlet containers) and application servers. High license costs and a desire to avoid them in favor or simpler and cheaper servlet containers have often been at the root of such concerns.

  JBoss embeds the web server in the application server, making for a much more efficient use of hardware while benefiting from an open source model that avoids many licensing concerns. As such, decisions to implement two physically distinct layers for presentation and business tiers should be reviewed in light of the lack of such licensing restrictions.



This enables JBoss to make more efficient use of hardware resources. Users have found that they can sometimes reduce their hardware requirements by up to 50 percent for the same load. This offers greater flexibility in selecting hardware and matching load. Of course there will still be occasions where separate physical tiers are recommended for various reasons, including security, isolated configuration, and better reliability.

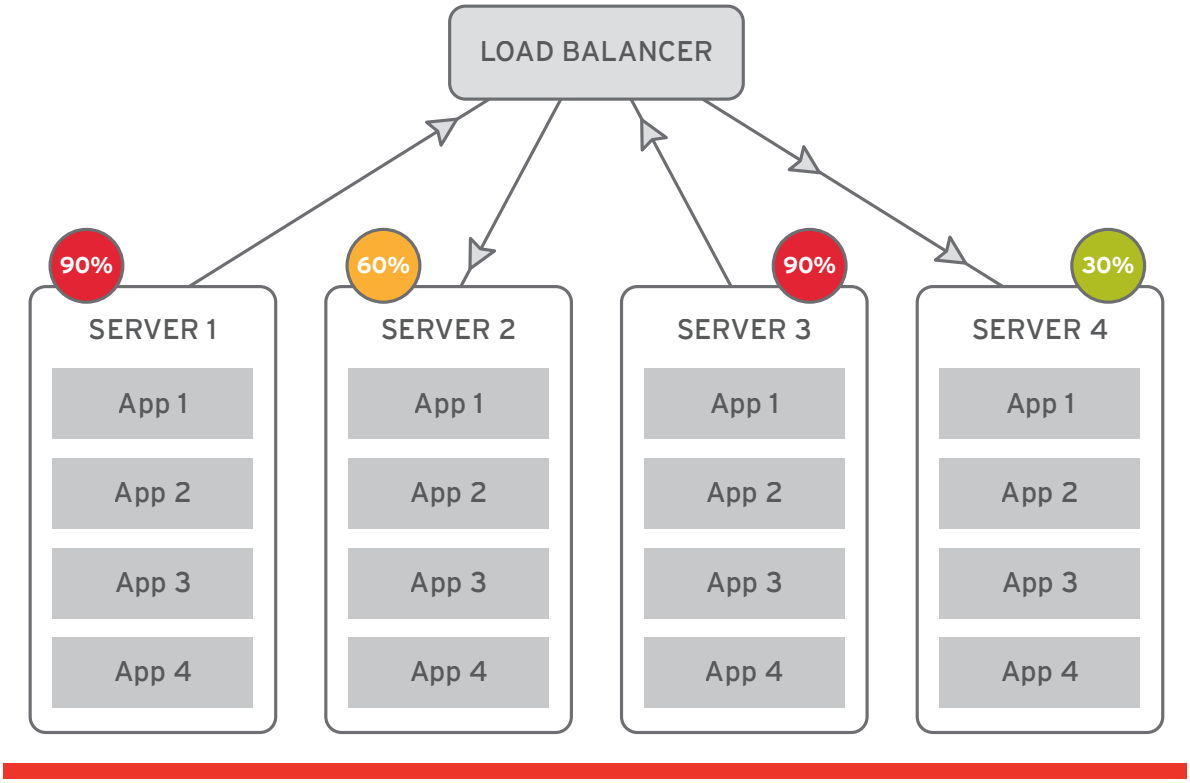
- **Single application per application server**



This deployment is used when:

- An application requires special configuration of the application server
- You are using a 32-bit operating system or JVM and need the maximum possible memory (less than 4GB per 32-bit JVM) for an application
- With 64-bit operating systems, JVMs, and more isolated configuration capabilities, the preferred deployment is multiple applications to a single server
- Governance or organizational issues that lead to preferring the single application per server model

- **Multiple applications per application server**



This is a more efficient deployment in many respects. Each application benefits from unused memory that would otherwise be allocated to each separate instance, with sufficient memory to minimize the latency at the application's peak usage. This would not necessarily coincide with peak usage time for other applications.

- **Multiple application servers per machine**

  Multiple instances of JBoss can be deployed to a single machine. This is a common deployment model if you are using a 32-bit operating system on a multi-CPU system with more memory than can be allocated to a single JVM. It can also be used when there is limited large scale hardware, multiple applications requiring special configurations, or the need for failover.

Stateful versus stateless web applications

Stateful web applications use the container-managed HTTP session to store their state and as such require HTTP session replication to be configured on the application server.

JBoss offers various options for state replication. A straight-forward configuration of the clustered cache would result in replication of the state in all participating nodes. More advanced features would enable buddy replication, where nodes are joined with one or more buddies to keep a backup of their state in case of failover. This is particularly useful in large clusters where the replication of state from each node to all other nodes results in a ballooning of memory footprint and can become difficult to manage.

State replication can be configured to take place either synchronously or asynchronously. The former strategy offers reliability and the latter favors performance.

Stateless applications can reside on the same server as stateful applications and simply do not use the replication features. The replication includes constant overhead plus a variable portion that grows with each application state, but is independent of the number of stateless applications.

HARDWARE MIGRATION

There are times when an application server is in some way tied to the operating system or even hardware. The client may have happily decided or have been compelled to use an application platform and operating system or hardware from the same vendor. Under such circumstances, hardware migration will often accompany the software migration to JBoss.

Even when a hardware or operating system migration is not necessary, it may well be an option worth considering. There are arguments both in favor and against migrating the operating system and/or hardware along with the application platform:

ADVANTAGES:

- Share resources between the software and hardware migration, including management and technical personnel, QA environments and network resources used for testing.
- Save time and money by avoiding duplicating the overlapping testing effort. Platform migration already requires thorough testing and evaluation of the environment, tasks that need to be performed for a hardware migration as well.

DISADVANTAGES:

- While it is arguable whether the risk of the combined effort will be less or greater than the sum of the risk of the two separate efforts, it is undoubtedly larger than that of a single type of migration and may therefore surpass a desired maximum threshold.
- When two types of migrations are combined, there is less isolation in changes. Therefore, the root cause of a potential problem will be harder to discover. By the same token, It will be more difficult to attribute any performance gain/loss to one migration versus the other.

Regardless of whether the hardware migration is required or optional, when and if it is performed there is an opportunity to redesign the physical deployment architecture of the servers in a way that can lead to cost savings and various improvements.

There are four primary deployment scenarios that are common to migrations: consolidation, dispersion, aggregation, and cloud migration. These scenarios are not mutually exclusive and can be combined in a large-scale migration to achieve the right balance of functional and operational characteristics for specific workloads.

HARDWARE REQUIREMENTS

Whether it is because of a necessary hardware migration or changes in the software requirements, it may be required to step back and take a look at application constraints to determine the proper hardware. Some questions that should be asked include:

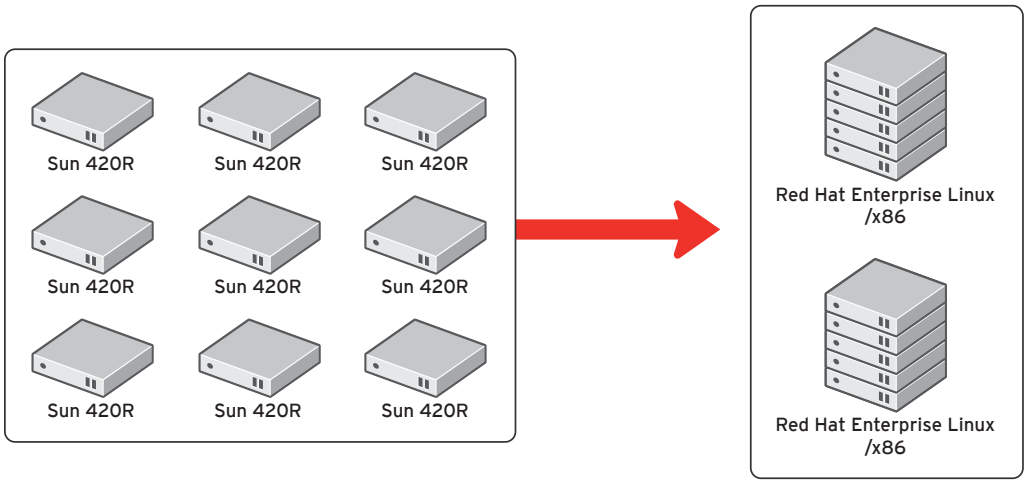
- Is the application mission-critical?
- What service-level agreement (SLA) must be met?
- How critical is the stored data and what type of storage and/or redundancy is required?
- What network topography suits the application and its communication needs?
- How much network bandwidth will be needed?
- What type of caching should be implemented on JBoss?
- What other optimization and tuning is appropriate for JBoss?
- Where is the optimal point in terms of latency versus throughput?
- How large is the required memory footprint and is it too large for a single JVM?

Certain performance expectations will have a direct impact on the choice of hardware and its deployment. For example, virtualization will often increase latency as a sacrifice to gain higher throughput and/or utilization. The same is also true with very large heaps that result in longer garbage collection. Certain advanced garbage collection strategies may alleviate that problem.

CONSOLIDATION

In the consolidation scenario, workloads on a large number of under-utilized systems are consolidated onto fewer systems. These new systems may use virtual machines running open standards-based operating systems to contain each workload. This type of scenario is common in environments where customers have made virtualization of systems a strategic directive. In this scenario, the customer utilizes the chosen virtualization technology to control access to system resources.

CONSOLIDATION SCENARIO



Advantages:

- Reduced hardware operational costs
- Reduced datacenter footprint
- Greater return on investment (ROI) from the chosen virtualization strategy

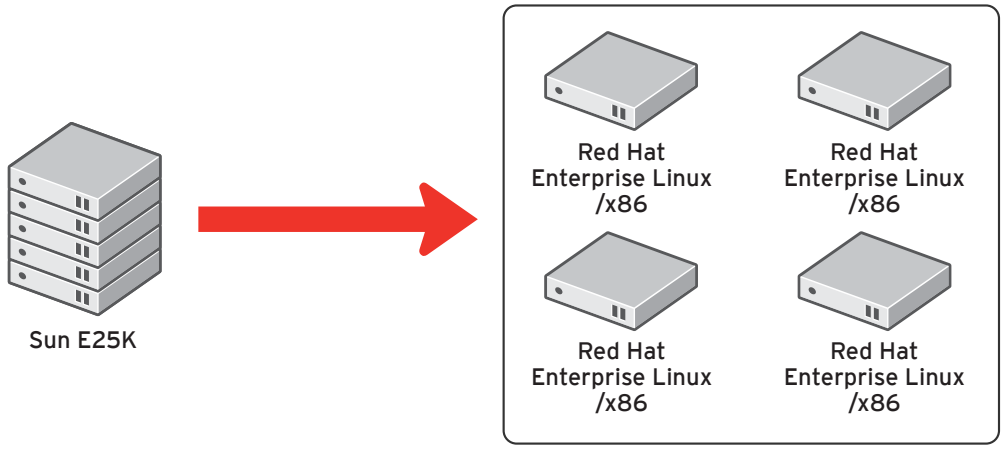
Disadvantages:

- Proprietary virtualization technologies can increase capital costs and create a new kind of vendor lock-in for the customer

DISPERSION

In the dispersion scenario, workloads on one or more large systems are distributed among a number of smaller systems running open standards-based operating systems. This type of scenario is common in environments where Enterprise Linux® has a growing footprint. Customers can distribute and scale hardware resources in smaller units across multiple datacenters. While 1U to 4U individual rackmount systems have traditionally been common in this scenario, the use of blades has been growing in recent years. Blade servers provide the customer similar advantages with lower operational costs.

DISPERSION SCENARIO



Advantages:

- High performance from newer x86 hardware technologies
- Low capital cost to scale hardware resources
- High flexibility with deployment and re-deployment of resources

Disadvantages:

- When not properly planned, this scenario can result in high operational costs

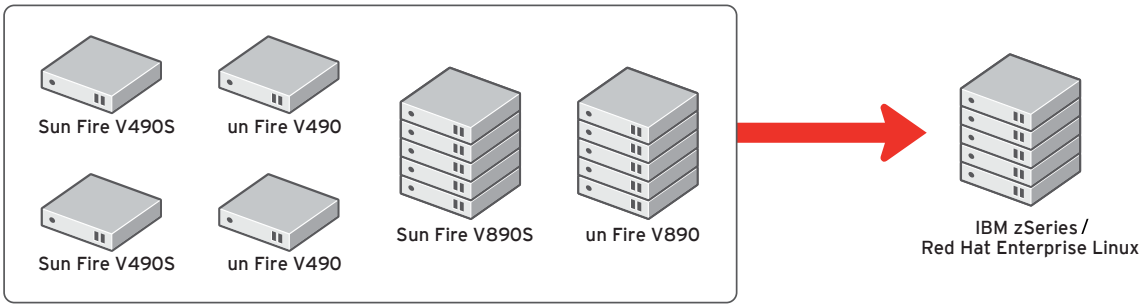
AGGREGATION

In the aggregation scenario, workloads for a large number of systems of various sizes are migrated into a single, large, fault-tolerant hardware platform where Enterprise Linux can be run. This type of scenario is common in environments where the customer already has a high investment in the specific hardware platform, and wishes to further leverage the platform to aggregate legacy platforms using Enterprise Linux. Customers have a choice of using hardware (LPARs, partitioning) or software (zVM, Xen virtualization) to control access to system resources.

Examples of these platforms include:

- IBM® System z® using Integrated Facilities for Linux (IFL) central processors
- HP® Superdome® (Intel Itanium-based)
- Fujitsu® Primequest® (Intel Itanium-based)

AGGREGATION SCENARIO



Advantages:

- Reduced hardware operational costs
- Reduced datacenter footprint
- Greater ROI derived from existing hardware platform

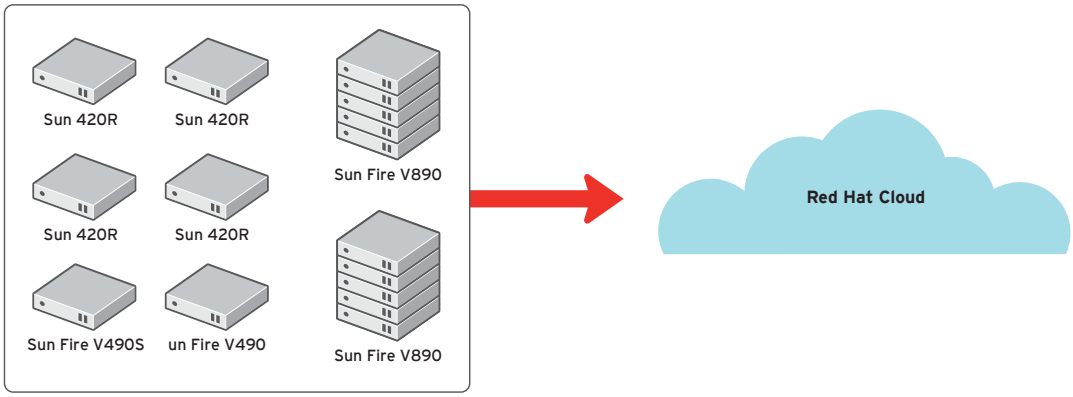
Disadvantages:

- Without prior investment in the platform, customer will incur a high capital hardware cost

CLOUD MIGRATION

In the cloud migration scenario, workloads on any number of systems are migrated to run on open standards-based operating systems in a cloud computing environment. This may be an internal cloud created by the customer, or an external cloud like those offered from Amazon or Rackspace. This type of scenario is very new to most customers, though a small number of customers are moving or have moved their entire operations into a cloud computing environment. Within the cloud, customers have a very high level of control over resources provided to individual workloads.

CLOUD MIGRATION SCENARIO



Advantages:

- Resources can be easily scaled up or down as needed for each workload
- Zero hardware costs (when using a public cloud)
- Low investment cost results in fast ROI (when using a public cloud)
- Higher hardware utilization providing better hardware ROI
- Simplified cloud environment provides for a lower operational cost

Disadvantages:

- Severe outage of cloud or connectivity can cause total loss of access to the operating environment (when using a public cloud)
- Critical data is stored and processed on systems not owned by the customer, so there may be issues of compliance and record-keeping (when using a public cloud)

OTHER CONSIDERATIONS

DIFFERENCES IN JAVA VIRTUAL MACHINE IMPLEMENTATIONS

In theory, various implementations of the JVM follow the same specifications and therefore an application or an application server that has been developed and tested against one JVM seamlessly works with other JVM implementations as well. While this is largely true, there are exceptions and circumstances where differences in implementations cause problems. These problems are sometimes the result of misinterpretation of the specification by a vendor, or at other times simply a defect that has not been uncovered and resolved. Most relate to assumptions made by developers that are not backed by the specification. When faced with uncertainty about the correct JVM behavior, more often than not, developers test the actual behavior and make assumptions based on their observations. Unfortunately such assumptions are only valid for the current JVM vendor and version and may not be part of the explicitly stated specifications. As such, a change in the JVM implementation may often negate such assumptions and lead to the misbehavior of the application.

CLASSLOADER STRATEGY

The JEE specifications have traditionally had little to say about classloading strategy, an unfortunate omission that has led to many challenges in JEE portability. Implementers of the JEE specification have made autonomous decisions on the default class loading behavior for applications that they host, though most have recently made such behavior configurable. Still, the difference in the default behavior alone is enough to cause much confusion and require days, if not weeks, of troubleshooting in search of the culprit.

WebSphere Application Server provides a hierarchical classloader model with the standard behavior to first delegate to the parent. After the JVM, extended libraries and WAS libraries, the hierarchy eventually reaches the application layer in which each application (including its JAR files) is assigned a classloader that is a child of the Application Server classloader. When Web Applications exist inside Enterprise Applications, they are by default given a classloader of their own, which serves as a child to the enterprise application classloader. In contrast, JBoss opts for a much simpler model where most of the artifacts reside in a singular flat classloader by default but are highly configurable and can be customized to use a desired and possibly shared classloader.

For an example of a classloader issue resulting from the mismatch of the default behavior of the two different application servers, consider that a flat classloader model in JBoss can mean that two different web applications with the same class bundled in their library would actually share the same instance of that class. This is in contrast with the default classloader behavior in WebSphere where that identical class will load in two different classloaders and be considered two different classes for all intents and purposes. So in JBoss, the static context between the two classes of the two web applications will be shared, whereas in WebSphere they will be distinct. This is a significant behavioral difference that can have a large impact but be difficult to trace.

A related issue is the version mismatch of third-party libraries; applications often bundle their third-party libraries that are already shipped with the application server, but depending on the class loading behavior, may or may not actually be using their bundled version. It is often easy to forget that with the exception of a few cases, "parent-first" is the most common classloader delegation strategy. This means that developers will often bundle a certain version of Hibernate, Struts or other commonly used library and use it with a certain expected behavior, unaware that a different version embedded in the application server may actually be the one in effect and overriding that of the child classloader. This can cause issues resulting from subtle functional differences in various versions of libraries that can be hard to troubleshoot.

3. THE STRATEGIC MIGRATION PROCESS

MIGRATION PROCESS OVERVIEW

PHASE	DESCRIPTION	DELIVERABLES	TYPICAL DURATION
Phase I: Server architecture & deployment model	This phase looks at the inventory of existing servers, network infrastructure, and applications. This creates a baseline of existing functionality and load capacity.	Inventory of application servers describing their purpose and general configuration; network configuration; hardware Inventory	2–4 weeks
Phase II: Application migration assessment	This phase assesses the individual applications for migration. This will examine dependencies on proprietary libraries and adherence to specifications.	Prioritized list of applications suitable for migration. Proprietary libraries used by applications. Specifications involved in migration.	2–12 weeks (highly variable, depending on number and complexity of applications)
Phase III: Effort and risk assessment	This phase looks at the application servers, their configuration, and services deployed. This will determine what functionality will need to be replaced on the servers. It will also assess the amount of effort required for each application and the risks involved in migrating them.	Listing of services configured by server. Clusters defined and the configuration of each cluster. Estimated effort in migrating each application. Identify major risk factors for each application.	4–6 weeks
Phase IV: Server and application migration plan	This phase will plan out the migration for the servers and the applications. This will use the information discovered in the previous phases to produce a roadmap for migrating the servers and the applications. This includes defining the open source alternatives to proprietary libraries and services.	Server migration plan detailing the services and configurations that must be migrated for each server/cluster. Application migration plan detailing library changes and code changes that will be required to migrate each application.	5–8 weeks
Phase V: Migration implementation	This phase involves the actual migration and successful deployment in the new environment. Red Hat Consulting offers a wide variety of workshops, training, and service offerings designed to help customers implement their strategic migration plan.	Successful migration	TBD

PHASE I: SERVER ARCHITECTURE AND DEPLOYMENT MODEL

This phase looks at the inventory of existing servers, network infrastructure, and applications. This creates a baseline of existing functionality and load capacity. This is a critical phase.

INFRASTRUCTURE ANALYSIS

The infrastructure includes the full ecosystem in which the application server operates.

- Datacenters: How many are there? What are their missions? Are they interconnected? Is there a failover strategy between datacenters?
- Hardware: Servers, load balancers, network routers (if clustering), database servers, file servers.
- Software: Load balancing, static web server, dynamic web server, GUI, etc.
- Ancillary services: This will include messaging, portal, AOP, injection, caching, etc.
- Development environment: What IDEs, how are they used, what methodology are they using?
- Applications: What applications run on the servers and which ones are business critical?

INFRASTRUCTURE ELEMENT	
Datacenter	one or multiple
Network	protocols, routers, DMZ,
Hardware servers	load balancers, database, application, web
Applications	web, Java EE, SOA
Development platforms	Eclipse, JBoss Developer Studio, Rational Application Developer, etc.
Source control	CVS, SVN, GIT, ClearCase
Build	Ant, Maven

PHASE II: APPLICATION MIGRATION ASSESSMENT

In order to be aware of the full scope of migration, it is important to assess what technologies are currently used by the applications and their migration targets. The typical enterprise application makes use of several technologies for security, caching, clustering, and injection. These technologies are largely hidden from the average application developer and they might not be aware of them, but they can have a major impact on the performance and scalability of the application.

Failing to plan for the migration of these technologies can create serious challenges. These will generally crop up at the end of a migration project when performance testing is carried out. So it is vital to assess not only the technologies, but also the specific features used, in order to determine the most appropriate alternative.

COMMON APPLICATION SERVER MAPPING

COMPONENT	FEATURES	JBOSS EAP
Monitoring and management	Remote control and configuration, threshold notification, etc.	JBoss admin console, standard JMX interface, threshold alerts
Web server	Servlet container, state replication	Embedded Tomcat, JSP and servlet spec implementation, coarse-grained and fine-grained in-memory replication
Messaging	Participate in transactions, persistence of messages, bridges outside cluster, JMS specifications	JBoss messaging, message distribution and rebalancing, persistent and transactional
Caching	Transactional, distributed, object graph support	JBoss cache, serialized and AOP-based replication
Clustering and replication	Layered replication, transaction-safe replication, fine-grained	JBoss cluster, configuration replication support using JGroups
Persistence	Supported databases, support for flat file systems, hybrid data stores, transaction support, and database feature support	JDBC connection pooling, Hibernate, security data stores, and transactional data sources
Security	Authentication, authorization, single sign-on, JAAS/JACC support, certificate support	JAAS compliant log-in modules, flat-file, database and LDAP support, SSO support
Aspect-oriented programming	Cross-cutting concerns	JBoss AOP
Injection	Standard injection support	Support for JEE injection standards
Presentation layer: JSP, JSF, Facelets, tag libraries	Compatibility and support for presentation standards	Support and implementation of common standards
Transaction manager	Reliable transactional behavior, distributed, support for latest technology	JBoss transactions, reliable transaction support, distributed, support for web service transaction standards

A key element of application migration will be the identification of proprietary libraries that may have been used in the development of the application. JBoss provides a Migration Analysis Tool (MAT) for assessing the amount of proprietary code that will have to be switched out for open source libraries. Details for using it can be found at:

http://community.jboss.org/wiki/JBossMASS-MigrationAssesmentTool-GettingStarted

More information is available from:

http://www.jboss.org/mass/MAT.html

The next step will be to identify required configurations and potential gaps between existing feature sets and open source capabilities. There are nearly always open source equivalents, or equivalents can be produced with relatively minor customizations.

Finally, an assessment should be made of the difficulty of migration for each application. One should plan to order the migration process from easiest to most difficult. Starting with the easiest applications will build knowledge and confidence in your team.

PHASE III: EFFORT AND RISK ASSESSMENT

After studying the deployment architecture and inventorying the applications and technologies that will be involved in the migration, we are in a position to estimate the effort and risk of the migration. Risk and effort assessment will include both technical and organizational components.

Using the technical analysis from the previous sections, we can begin the technical effort and risk assessment. This will focus on:

1. Technical analysis
   - Scope
   - Number of servers, datacenters, and applications to be migrated
   - Technology gap analysis
   - Features that don't immediately map to open source alternatives
   - Conflicting configuration requirements
   - Adherence to specifications
   - Use of proprietary IDEs that embed proprietary libraries
   - Library selection
2. Organizational analysis

   Organizational factors will generally prevail over technical factors. Technical factors can be easily delineated and addressed. Organizational factors lie beneath the surface and can derail efforts. Seemingly small hurdles can become insurmountable if the organization is unprepared or unwilling to address them. Without a plan for addressing issues, the organization can quickly become focused on minor shortcomings and overlook major opportunities for migrating to open source.

   The first step in addressing organizational risk factors is to analyze organizational issues and risks. This will provide a roadmap for preparing the organization for the migration. Organizational readiness factors include.

   - Training and knowledge gaps
     - Is the staff knowledgeable about the technology or merely comfortable with existing tools?
     - Has the organization adopted a process for developing and deploying software?
   - Workload factors
     - Will the current staff have time to perform current workload tasks and also participate in training and migration work?
     - Will there be sufficient hardware to deploy and test new servers before putting them into production?
   - Cultural factors
     - Bottom-up versus top-down decision making
     - High quality versus low cost
     - Leading versus trailing edge
   - Budget
     - CAPEX versus OPEX
     - TCO versus ROI

Taking these considerations into account for the migration plan will avoid unpleasant surprises and produce a successful migration.

A strength, weakness, opportunity and threats (SWOT) analysis can help define the organizational readiness. It basically helps the organization match up strengths and weakness, opportunities and threats, and then develop a plan to leverage strengths to overcome weaknesses.

SWOT ANALYSIS

STRENGTHS	WEAKNESSES
• IT staff trained in relevant standards and technologies • IT staff growing specific JBoss knowledge • Applications follow WAR and EAR formats • Use of open IDEs	• Reduced budget • Reliance on proprietary integrated development environments
OPPORTUNITIES	**THREATS**
• Budget cuts • Licensing coming up for renewal • Need to expand capacity beyond the current licenses • Integration of multiple platforms due to acquisition or standardization	• Non-standards based deployments • No budget for training • All resources are currently fully loaded • No budget for expanding capacity for migration

3. Migration risk assessment

There are risks common to all migrations. By understanding them and preparing for them upfront, we can minimize the risk posed to the migration effort. JBoss offers training, consulting, and tools for migrations.

Migration is a specialized skill that may not be worth developing internally. Even if the migration effort will be extensive enough to justify training an internal team to do it, the company may wish to utilize outside expertise.

RISK	LIKELIHOOD OF OCCURRENCE	POTENTIAL IMPACT	STRATEGY
Training budget	High	High	Formal classes and workshops can be provided to support migration efforts.
Staging hardware	Medium	Medium	It will be important to have hardware similar to the production environment to fully test configurations, applications, and installation.
Use of proprietary IDE tools	Medium	High	Proprietary tools from WebLogic and WebSphere will make heavy use of proprietary libraries. In their code generation, they will generate classes that extend or implement proprietary classes. We can use the JBoss MASS project and MAT in particular for identifying potential migration issues.

PHASE IV: SERVER AND APPLICATION MIGRATION PAN

Once the risk and effort analysis has taken place, it is critical to begin thinking about how to design a standard operating environment (SOE) upon which to migrate. An SOE is an organization's standard implementation of the core operating system and middleware components. It can include the base operating system, the JEE container, custom configurations, standard applications used within an organization, software updates, and service packs.

Once an application set has been identified, a standardized configuration based on an SOE approach will be created for rapid and consistent deployment. An SOE configuration consists of a set of tested hardware, tested software, and configurations deployed within a JBoss environment. The SOE configuration will be fully aligned to your technical and business requirements, dramatically reduce deployment time, simplify maintenance, increase stability, and reduce support and management costs.

In some cases, multiple SOEs will be required and the next step in the migration plan is to determine how many different server configurations will be necessary to support all the applications after the migration. Multiple server configurations may be necessary because different applications require conflicting configurations, throughput versus latency, special deployments, or because the expected load requires special hardware for some applications. The key is to keep the number of configurations as small as possible while meeting the functional and non-functional goals of your environment.

Once the number of configurations has been determined, it will be important to know which applications will be deployed to which servers. From this point, configuration of the individual applications can begin. Individual applications can be configured to use different class loaders, interceptor stacks, caches, URLs, and many other configuration options.

Plan to provide training for the team to compensate for knowledge gaps that were identified in the earlier risk and effort assessment. With the migration of applications scheduled, training can be planned to coincide with the need for that knowledge in the migration effort. Arranging the training schedule so that the migration team first encounters the problem, works with it, and then gets training in the relevant technology can be a very efficient way of advancing the migration project.

Cost estimation and timeline finalization will be the next items to address. Cost estimation will focus on the following areas:

- Cost of staging hardware that will be used for testing servers and configurations
- Cost of developing extensions to existing open source software to fill gaps when moving off proprietary software
- Migrating applications can be grouped for costestimation based on whether they need source code changes or just simple configuration changes
- Training costs
- Savings from reduced software costs
- Savings from re-deployed hardware

PHASE V: MIGRATION IMPLEMENTATION

Once a migration plan has been formed, one or more corresponding project plans can be designed and implemented. It is important for major milestones in the project plans to follow those of the migration plan and that training and resource availability conform to the migration plan.

Red Hat resources are available to ensure success in migration and include public and private training as well as development and production support and consulting.

4. ENTERPRISE SERVICES

In the current economic climate, it's critical to make the most of the technologies currently deployed while still looking for opportunities to carve out costs. Red Hat Enterprise Consulting Services provides the expertise and knowledge transfer to help your organization realize a faster time to value and improved migration experience.

ENTERPRISE-CLASS CONSULTING DELIVERED BY SUBJECT MATTER EXPERTS

Partnering with Red Hat Consulting to plan a middleware migration ensures success by combining proven best practices and methodologies with the experience and expertise of Red Hat consultants. With Red Hat, risks are mitigated better, implementation time is reduced, and as a result, the cost of the migration itself is lower. A Red Hat consultant will ensure that the migration team has the knowledge and support needed to complete the job with minimal disruption to IT operations.

Red Hat Consulting has a proven track record helping customers do more with less by fully utilizing the value of their subscriptions. Our global team of consultants includes architects and engineers who are JBoss product experts. Cumulatively, they have years of experience integrating JBoss Application Server into unique and varying environments—always ensuring maximum performance and value.

TRAINING TO IMPROVE PRODUCTIVITY AND PERFORMANCE

By investing in the expertise of your IT staff, you can help ensure optimal system performance, enhance productivity, and mitigate risk. Award-winning training offerings from Red Hat give your team the skills and confidence needed to maximize your open source implementation.

MIDDLEWARE CONSULTING SERVICES

By engaging Red Hat Consulting Services, you benefit from years of experience in the design and deployment of Java-based middleware on both open-source and closed-source software platforms. This expertise in closed-source middleware platforms helps Red Hat Consulting quickly identify and overcome the potential risks and hidden costs of porting applications within your portfolio to the JBoss platform. Red Hat Consulting has unique expertise with the JBoss Platform that helps ensure that your migration results in applications that are designed to work with future versions of the JBoss platform, as well as the current version.

Red Hat Consulting provides a wide range of services around migration to the JBoss Platform, including:

- **Migration assessments** – Review some or all of your current applications to identify costs, risks, and detailed estimates of the effort required to migrate your applications to the JBoss platform.
- **Application migrations** – Convert your existing applications from closed-source middleware systems to the open JBoss Platform to help reduce licensing costs, minimize vendor lock-in, and increase open standards compliance, where possible and cost-effective.
- **Architecture reviews and recommendations** – Provide ongoing guidance for your JBoss use and future application development. Help align your technical roadmap with the standards and features planned for future versions of the JBoss platform.

And because each project is unique, Red Hat Consulting will work with you to understand your current environment and the most appropriate scope of work to successfully migrate your applications to JBoss and maintain and enhance them in the future.

Red Hat Consulting uses developed tools, proven methodologies and experience gained in the field through numerous migrations to help guide clients through a smooth and safe process.

PLATFORM CONSULTING SERVICES

With all migration efforts, having a solid infrastructure that provides a scalable foundation is the first step. Red Hat infrastructure migration planning services provide a detailed evaluation of your IT environment and deliver strategic recommendations for simplifying your IT infrastructure as you migrate. The result? You can reduce IT costs while creating a scalable IT infrastructure.

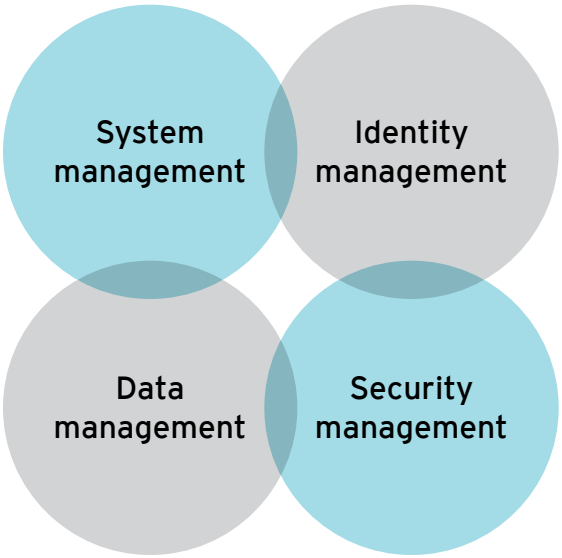
Red Hat provides a foundation based on the standard operating environment (SOE) approach, in order to ensure a successful migration and a solid foundation for your organization's continued growth.

BENEFITS OF AN SOE

- Simplified architecture: One codebase that can be deployed on different branches and services. Support different platforms (workstations, servers, or mainframes) from the same build process.
- Flexible and rapid deployment: Grants the ability to take a server from bare-metal to fully-configured in less than ten minutes. Ensures identical configuration and the ability to compare machines from a centralized GUI interface, which is useful when searching for anomalies.
- Security: Enforce security policy across different machines and distributed datacenters.
- Centralized management: Manage different types of machines with different functionality remotely. Also includes the ability to delegate responsibility to regional or provincial management.

Centralized configuration management: Enforce configuration, schedule configuration updates, compare configurations, and query current configuration.

STANDARD OPERATING ENVIRONMENT DIMENSIONS



Systems management: Evaluates and documents current systems management infrastructure. Recommendations will be provided regarding the management of systems and software post-migration and how to incorporate Red Hat Enterprise Linux into existing change management processes and systems.

Service includes:

- Bare-metal and virtual platform provisioning
- Linux software build and deployment
- Monitoring and performance management

Identity management: Determines and documents current identity management policy. Recommendations are provided for integrating Enterprise Linux systems into existing authentication and authorization infrastructures or for migrating existing directory solutions to open source software.

Service includes:

- User and group management
- PKI infrastructure
- Policy creation and enforcement

Data management: Determines and documents availability requirements for migrated services. The architect will design a strategy for meeting those requirements with a mixture of storage and clustering technologies.

Service includes:

- High-availability clusters
- Distributed file systems
- Load-balancing solutions
- Disaster recovery
- Systems and data backup
- Data recovery
- Bare-metal recovery

Security management: Identifies and documents current corporate security practices and procedures for Linux as well as requirements for migrated services. A thorough understanding of the end-user requirements is necessary.

Service includes:

- Operating system hardening
- Emergency security errata patching
- Security auditing and reporting
- Compliance requirements and remedial action

Within each of the above areas, a gap analysis is performed to assess the existing infrastructure and processes that support the Enterprise Linux operating system versus the support of other operating systems within your IT environment. This analysis is conducted using industry-standard practices and industry-proven methodologies.

One of the additional benefits of these tasks is that Red Hat works side-by-side with your team members to provide hands-on mentoring, real-time knowledge share, and valuable guidance as your teams encounter issues or have questions.

Red Hat Consulting offers a comprehensive suite of end-to-end solutions to help your business realize the benefits of your investment faster—no matter where you are in your deployment cycle.

RED HAT CONSULTING SOLUTION	DESCRIPTION
Assessment	Combines proven best practices with the expertise of Red Hat Consultants to plan a safe, stable migration.
Quick Start	Accelerates project completion and time to value.
Implementation	Comprehensive installation, configuration, and deployment of new technologies.
Health Check	Validates installation and configuration of the technology to identify issues that impact your business.
Optimization	Troubleshoots and resolves issues, thus increasing business effectiveness and reducing costs.

If you're ready to begin your migration initiative, e-mail or call us, and we'll have a conversation to determine how we can best support you and your organization.

Tel: 1 (866) 273-3428 x44555 or **redhatconsulting@redhat.com**

TRAINING

When migrating platforms, it is critical to ensure that you have a skilled staff who can maximize performance beyond initial deployment. Hands-on training from Red Hat is offered and suggested to teach organizations optimal management techniques, effective troubleshooting, and the ability to maintain improved efficiencies across the entire system. Training leads to rapid, successful deployments and ensures your staff has the skills and knowledge to keep your IT organization running smoothly.

Red Hat Training and Certification programs are widely recognized as a cost-effective, high-impact way to improve operations for the long haul. In fact, for the past four years IDC has ranked Red Hat a leader in IT education. With training, your staff can better manage systems, quickly troubleshoot problems, and improve efficiency across all systems.

Various Red Hat training courses are available, and while some have prerequisites, there are courses available for those without prior Java or JEE experience. Regardless of experience level or training goals, Red Hat Training has the right course and training path that will build on and leverage existing industry experience.

Ways to Train: Whether you need to train a team of IT Professionals—or just yourself—Red Hat offers a variety of learning styles, delivery methods, certifications, savings programs, and customized solutions to maximize return on your Red Hat enterprise investment.

- Classroom training: Robust, six-month schedule of classroom training in more than 40 locations across North America.
- Virtual Training: Same industry-recognized classroom training while maintaining a partial work schedule and achieving cost savings objectives.
- Onsite Training: At offices, our training venues, or hosted in a virtual environment; also provides the option of customizing the course content to meet your specific training requirements.

VALUE OF JBOSS ENTERPRISE MIDDLEWARE TRAINING

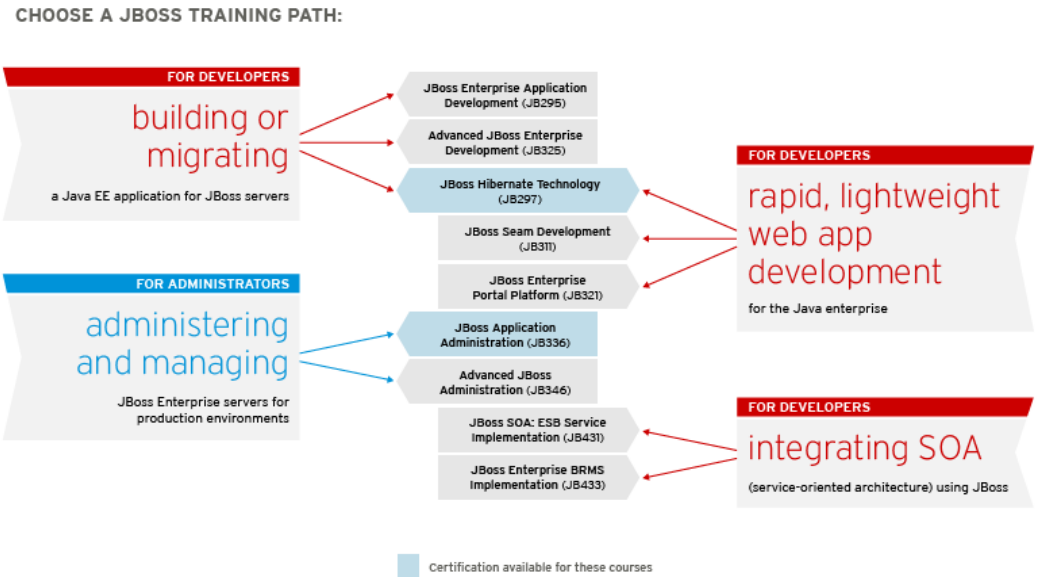
Your team is your most valuable asset. By investing in the expertise of your staff, you can ensure optimal system performance, enhance productivity, and mitigate risk. When you include hands-on JBoss Enterprise Middleware training as part of your deployment and maintenance strategy, you can maximize the value of your JBoss technology investment and reach new productivity heights.

INDUSTRY-LEADING, ROLE-BASED TRAINING

Red Hat uses best-of-breed contemporary teaching approaches to ensure maximum skills transfer and retention. This includes actively engaging students in task-focused activities, lab-based knowledge checks, and facilitative discussions. In addition, our course materials provide job aids, exercise reinforcement, and task-based reference material. Industry-leading, role-based training can increase a company's return on investment (ROI) and an IT professional's productivity.

PROVEN COMPETENCIES THROUGH PERFORMANCE-BASED TESTING

Performance-based certifications from Red Hat are among the most highly regarded in the IT industry. Their success stems from a solid track record of benchmarking and hands-on skills assessment. Because of this method of testing, our customers can be assured that certification professionals on their teams have demonstrated the skills needed to perform the required tasks in actual IT environments.



Courses: The following table lists standard courses offered by Red Hat for JBoss Enterprise Middleware. Customized education opportunities are also available on a per-customer basis.

COURSES
JBoss Application Administration (JB336) Red Hat's most popular JBoss course, JBoss Application Administration, focuses on installation and deployment issues with the JBoss Application Server as well as configuring the server for production usage.
JBoss Enterprise Application Development: (JB295) Intended for entry to mid-level Java developers, the JBoss Enterprise Application Development course will expose students to JBoss Java EE frameworks, specifications, and interfaces (APIs).
Advanced JBoss Enterprise Development (JB325) JBoss Enterprise Development dives into the JBoss Enterprise Application Platform (EAP) with an emphasis on advanced Java EE application programming interfaces (APIs).
JBoss Seam Development (JB311) The JBoss Seam Development course teaches experienced Java developers how to efficiently use Seam to intelligently tie components together and manage increasingly complex IT systems.
JBoss Hibernate Technology (JB297) The JBoss Hibernate Technology (JB297) course gives Java developers the knowledge and skills required to leverage the powerful Java Hibernate Application Stack.
JBoss Enterprise SOA (JB341) The JBoss Enterprise SOA course provides developers with real-world examples, integration patterns, and strategies for integrating enterprise systems and legacy applications via a common, services-based architectural style.

To access a complete, interactive, or PDF/printer-friendly version of the complete JBoss Enterprise Middleware course catalog, please visit **https://www.redhat.com/training/catalog/**.

Red Hat training specialists can help identify your staff training requirements and what level of training is needed. Contact Red Hat at **training_sales@redhat.com** to craft a custom corporate training plan to meet the needs of your group.

5. SUCCESSFULLY MIGRATED CUSTOMERS

UNION BANK, N.A.

CATEGORY: Superior Alternatives

INDUSTRY: Financial Services

GEOGRAPHY: Headquarters: San Francisco, CA

BUSINESS CHALLENGE: An aging and costly IT infrastructure was impeding the ability of Union Bank to scale to growth and respond agilely to changing market dynamics

MIGRATION PATH: UNIX™ on high-end RISC machines to Red Hat Enterprise Linux running on Intel Xeon based HP servers; WebSphere to JBoss Enterprise Application Platform.

SOFTWARE: Red Hat Enterprise Linux™, Red Hat Network Satellite, JBoss Enterprise Application Platform™, JBoss Seam, JBoss Hibernate, Red Hat Consulting

HARDWARE: More than 150 Intel™ Xeon™ processor-based HP ProLiant servers

BENEFITS: Improved reliability and scalability, cut costs, and delivered new financial services and products to market faster

BACKGROUND

Union Bank, N.A., headquartered in San Francisco, is a full-service commercial bank providing an array of financial services to individuals, small businesses, middle-market companies, and major corporations. Union Bank is California's fifth-largest bank by deposits. The bank has 335 banking offices in California, Oregon, and Washington and two international offices. Its holding company, UnionBanCal Corporation, is the 16th largest commercial bank holding company in the U.S. based on assets at March 31, 2009.

Union Bank migrated its operating platform from AIX to Red Hat Enterprise Linux and its middleware environment from WebSphere to JBoss to support its mission critical applications at an improved price with greater performance and less up-keep. Union Bank used open source solutions to increase time to market, reliability and return on investment.

BUSINESS CHALLENGE

When Mok Choe joined Union Bank in early 2007 as chief technology officer, the Union Bank IT infrastructure faced a host of challenges similar to those of many other companies at the time, mainly increasing costs and resources associated with the maintenance and upkeep of legacy systems.

Over the years, Union Bank's IT infrastructure had grown increasingly large, cumbersome, and complex. Not only was it costly to operate and maintain, but it couldn't scale to accommodate the bank's rapid expansion into new markets. System availability was also a continuing challenge. And as the financial services industry

expanded into electronic banking products, Union Bank's reliance on IT was increasing. The bank thus required an IT infrastructure that could speed new products to market with rock-solid reliability and availability, and which could also scale as needed.

The hardware environment embraced a "big box" approach with a few massive servers at strategic locations that offered little relief when significant impacts occurred. This environment required tremendous overhead with constant monitoring and management of server problems.

The IT department at Union Bank was also under pressure to reduce the total cost of ownership (TCO) of its overall IT operations. The solution needed to deliver a robust disaster recovery environment with minimal mean-time-to-restore (MTTR) and maximum mean-time-between-failures (MTBF) times. Finally, the solution needed to better leverage Union Bank's most highly skilled IT workers. By enabling valued staff workers to reduce the day-to-day support required by overhead-intensive legacy systems, productivity would improve, and the bank's IT department could move from a reactive to proactive support model.

"First and foremost, we needed to improve system availability," said Choe. "Secondly, we needed to speed time to market of new financial services products. And at the end of the day, we needed to decrease the cost per transaction of delivering services."

SOLUTION

Union Bank immediately focused on the task of establishing a new and innovative technology environment. The first decision: to create a new open source-based enterprise-wide IT platform to obtain improved availability, agility, scalability and reduced TCO (total cost of ownership), while enabling the support of the bank's growing IT needs and better alignment with the bank's overall business plan.

"We did three specific things," said Choe. "First, we migrated our entire Web-based infrastructure over to Red Hat Enterprise Linux so we could go from a scale-up to a scale-out architecture. Next, we ported our teller platform over to JBoss. And third, we wrote a brand new Web-based cash management application built on the entire Red Hat technology stack: Red Hat Enterprise Linux, JBoss, Hibernate, and SEAM."

The strategy started at the operating platform level by replacing the aging UNIX based RISC servers with commodity x86 machines running Red Hat Enterprise Linux, and migrating to JBoss Enterprise Application Platform at the application server level. Union Bank initially utilized Red Hat Network to set up centralized, secure management of its Red Hat Enterprise Linux systems.

Union Bank took advantage of Red Hat Consulting to assist the IT group with the initial design of the first phases of deploying the new architecture and Web-based applications. The bank's infrastructure and application development teams attended Red Hat Training to learn valuable tools and lessons on integration and migration issues.

The new strategy also encompassed building a new data center that leveraged virtualization technology on top of Red Hat Enterprise Linux to dramatically reduce the bank's hardware footprint. "The bank is very serious about its green initiative, and Red Hat Enterprise Linux is a key part of that," said Choe.

One of the most strategic projects was to replace the bank's operating system environment on branch teller systems with JBoss Enterprise Application Platform running on Red Hat Enterprise Linux. Within just months, the Union Bank development staff was able to create a "silent" JBoss deployment package and distribute it remotely to over 330 production branch servers.

"The JBoss-based teller application has been running successfully at the 330 branch sites ever since," said Choe. "The small footprint of JBoss has freed up much needed space on each branch server and has laid the groundwork for future expansion. We plan to migrate other customer-facing web applications from WebSphere to JBoss Enterprise Application Platform."

BENEFITS

Union Bank's innovative approach to its IT re-architecture has resulted in improvements to system availability, scalability and, resiliency, increased ROI, enhanced security, provisioning, configuration management, and improved time to market.

The most significant benefits have been improved system availability, scalability and resiliency, as well as increased ROI. Upon migrating to Red Hat Enterprise Linux, there have been improvements of the bank's hardware infrastructure, as seen by improved mean-time-to restore (MTTR), and mean-time-between-failures (MTBF).

The return on investment (ROI) was also substantial. For example, the large RISC machines were running at less than 50 percent capacity. To ensure redundancy, the bank needed to double its hardware investment to allow for failover. "With Red Hat's commodity model, we were able to spread the load over multiple machines and reduce our overall spend by approximately 80 percent," said Choe. "And these savings don't take into account the reduced maintenance costs of moving to the Red Hat platform, which is easier – and therefore cheaper – to maintain."

Additionally, because application performance increased significantly under the new JBoss and Red Hat architecture, the bank was able to reduce the time-to-market of new products. The bank was also able to improve customer service by boosting the performance of its teller application. "The success of that project gives us confidence to tackle the rest of our browser-based Web applications with a JBoss solution," said Choe.

The move from a vertical to a horizontal architecture and process enhancement have improved both system availability and resiliency, which allows the bank to absorb normal glitches without impacting customer transactions. "The reliability of our Web applications has improved to the point where I can go to our business partners and confidently say we have better than 'four 9s' availability," said Choe.

The Red Hat/JBoss solution requires less maintenance and enables Union Bank IT to reduce their efforts on day-to-day support of legacy systems, allowing for better resource utilization. This also helped the IT group move from a reactive to a proactive model more expediently.

Additionally, the bank's overall cost-per-transaction declined 25 to 40 percent, something that Union Bank's business centers appreciate. "We have a charge-back system in which our departments pay for the IT resources they consume," said Choe. "They've seen their charges go down month by month."

"We benefited greatly from Red Hat consulting services as they provided valuable input and assistance in helping us migrate to Red Hat technology and dramatically improved our ability to achieve our goals," said Choe, "With Red Hat Consulting, we felt there was an immediate knowledge transfer, and we were very satisfied with the level of involvement and quality of knowledge provided to our team."

And ultimately many of the ongoing benefits that Choe expects to reap in coming years as a result of transforming the bank's IT operations come from his expanded technology options. "We've achieved tremendous cost, reliability, and availability benefits, but in the end it all comes back to the fact that we now have choices when it comes to deploying hardware and software," he said. "We're no longer locked into using a particular product or vendor. Open source – and by extension, Red Hat – makes that possible."

"The high costs and overhead associated with legacy proprietary-software and infrastructure led us to the decision to deploy Red Hat and JBoss open source solutions, and this allowed us to provide core infrastructure and development platforms at a significantly lower cost and at a faster rate," said Choe, "Our use of Red Hat and JBoss solutions demonstrate creative business innovation through the use of horizontal architecture and the improvements allow Union Bank to continue to increase our customer experiences."

GEICO (GOVERNMENT EMPLOYEES INSURANCE COMPANY)

Industry: Insurance

Geography: US

Business challenge: Existing proprietary middleware platform was complex to manage, not performing and scaling as expected, and expensive to maintain. The architecture team decided to investigate alternatives that could be deployed that would better meet their needs.

Migration path: Proprietary middleware platform to JBoss Enterprise Middleware

Software: JBoss Enterprise Application Platform: 28 bands (1 band = 32 CPUs), JBoss Technical Account Manager (TAM), Red Hat Consulting, Amentra

Hardware: 50 Dell servers

Benefits: Reduced the TCO by more than 30 percent, throughput gain of 3X with utilization down to a third of the current platform, overall resource utilization went from above 50 percent to under ten percent, which allowed significant room for scalability without having to acquire additional hardware.

BACKGROUND

GEICO (Government Employees Insurance Company) is the third-largest private passenger auto insurer in the United States based on the latest twelve months written premium. GEICO provides auto insurance coverage for nearly nine million policyholders and insures more than 14.4 million vehicles.

In addition to auto insurance, GEICO also offers customers insurance for their motorcycles and homes. Commercial auto insurance, boat, ATV, RV, personal umbrella protection, and life insurance are also available.

GEICO is a wholly-owned subsidiary of the Berkshire Hathaway group of companies, is rated A++ for financial stability by A.M. Best Company, and ranks at the top of several national customer satisfaction surveys. For more information about GEICO, go to www.geico.com.

BUSINESS CHALLENGE

GEICO's enterprise architecture team recognized they were facing several challenges with their existing proprietary middleware platform. The platform was complex to manage, not performing and scaling as expected, and expensive to maintain. The architecture team decided to investigate alternatives that could be deployed that would better meet their needs.

The GIECO IT team identified the following challenges with their existing proprietary solution:

- **Cost** – GEICO's license agreement was a "time bound licensing agreement" related to the number of proprietary application servers deployed during the time frame. Since GEICO experienced significant growth during this timeframe, the cost to "true up" and pay for the additional licenses was significant.

- **Performance** – When GEICO upgraded their standard Java Development Kit (JDK) from version 1.4 to 1.5 on their existing proprietary platform, they did not see any improvements in machine (CPU/Memory) usage or application response time. After eight weeks of performance testing and tuning, they were finally able to configure the upgraded proprietary platform to match the earlier version's performance. The upgrade was not only cumbersome but was also expensive since they had to engage external consultants to accomplish the upgrade.
- **Memory leaks** – The previous proprietary deployment also experienced unexplained memory leak(s). Developer load and memory testing returned misleading results unless the developer knew how to work around the leaks and complete certain types of tests.
- **Documentation and support** – GEICO found it challenging to identify and understand the Java API in the current proprietary environment due to lack of documentation. They also had challenges in acquiring tools to identify memory issues, debug leaks, etc. For every instance of a high-severity issue such as memory leak, external consultants needed to be engaged to identify and fix the problem.
- **Staging** – Due to these challenges, some of the GEICO development teams adopted JBoss technologies for their developer workstations and began building applications using JBoss. This dual-use strategy became complex and redundant for IT Operations as they needed to make configuration changes on both the proprietary and JBoss platforms.

GEICO conducted extensive research and identified Sun's GlassFish and JBoss Enterprise Middleware from Red Hat as potential solutions that were suitable for GEICO's applications and infrastructure. JBoss Enterprise Middleware was selected based on its market share and extensive support from Red Hat. GEICO conducted a proof-of-concept, installing JBoss Enterprise Application Platform on a cluster of servers (POC environment). Performance and load tests were conducted using various tools for a selected business application on both platforms.

The JBoss results from these tests were astonishing. A few highlights include:

- User page transition time decreased as much as 19 seconds using JBoss
- During the proof-of-concept, 1,749 additional business processes were created on the JBoss platform
- On the same hardware and environment, JBoss required 70 percent less CPU resources than the current platform
- Performance tuning with JBoss was accomplished in 40 man hours versus 1,440 man hours for the existing proprietary platform

GEICO also conducted multiple reference checks with organizations that were of similar size and industry. The reference checks were extremely positive about Red Hat and JBoss Enterprise Middleware.

SOLUTION

The solution consisted of subscriptions for JBoss Enterprise Application Platform (EAP) and the initial deployment environment consisted of 540 processors. An additional 350 were added at a later date. A plan was put together to aggressively migrate two out of three mission-critical applications in a time span of three months. GEICO also utilized a JBoss Technical Account Manager (TAM) who was dedicated to supporting GEICO's specific needs during their switch to JBoss.

BENEFITS

By implementing JBoss Enterprise Application Platform, GEICO was able to reduce the total cost of ownership by more than 30 percent.

When compared to the previous proprietary platform, GEICO also experienced throughput gains of 3x, and a two-thirds reduction in utilization. The overall resource utilization went from above 50 percent to under ten percent, which allowed significant room for scalability without having to acquire additional hardware.

One of the challenges for GEICO was the time-bound migration process. GEICO's middleware team was trained on JBoss for a week. With the support of Red Hat and Amentra (a Red Hat company), they successfully migrated the initial two applications and were able to migrate the third application as well. This was a clear demonstration of expertise in Red Hat Consulting services and the ability of GEICO's middleware team to adapt rapidly to the new JBoss environment.

GEICO had initial concerns about support, stability, and deploying open source software for its mission-critical applications, but the market maturity and the premium level of enterprise support offered by Red Hat made it very easy to make the change to an open source environment.

GLÖCKLE DIRECT

Industry: Marketing and call center services

Geography: Stuttgart, Germany

Business Challenge: To build a scalable and highly flexible middleware infrastructure for a new Java-based customer care and marketing application based on proven database server hardware

Software: JBoss Enterprise Application Platform, JBoss EJB (Versions 1.1 and 3)

Hardware: IBM AS/400 midrange servers, HP RX6600

Migration Path: Legacy IBM AS/400 client-server based application to a browser-based system developed in Java, using JBoss Enterprise Application Platform

Benefits: Built a modern and scalable infrastructure; Realised considerable cost savings; Secured its investment in a mission-critical, future-proof infrastructure

"What's great about JBoss Enterprise Middleware is that all the components just work together nicely. We can pick just what we need and keep the total cost of ownership much lower than with proprietary software. That also saves us a lot of time and effort when architecting the system."

–Uwe Wagensommer, head of IT at Glöckle direct

BACKGROUND

Glöckle direct GmbH is a marketing and call centre services company, based in Stuttgart, Germany, that provides technical infrastructure and various management functions to clients within the Glöckle Group. Companies in the Glöckle Group, a family-owned group of companies with a history stretching back more than one hundred years, include lottery companies in Germany and Austria, an electricity reseller, and an e-commerce company that sells luxury chocolates.

All of these subsidiaries have central business functions provided by Glöckle direct, including technical and IT infrastructure, human resources, and marketing. Glöckle direct's IT and application development department is also responsible for creating and delivering the software tools required for successful operations within the Glöckle Group.

Because state-licensed lotteries account for the lion's share of Glöckle Group's business, a large part of its application infrastructure is managed by Glöckle direct's IT department, supporting marketing and customer care functions. Throughout the past decade, Glöckle Group has gradually expanded its operations beyond managing and marketing certified lotteries in southern Germany and added lottery organisations in Austria as well as northern Germany.

The company's IT philosophy is "build our own," meaning that the entire infrastructure and application environment is custom developed, managed, and operated by Glöckle direct's staff to meet the company's specific needs.

BUSINESS CHALLENGE

One of the key business processes for Glöckle Group's lottery businesses is the collection and management of customer data through its call centres. Call centre agents service existing customer contracts and follow up with potential up- and cross-selling opportunities. For many years, the central application used by call centre agents was based on a legacy "green screen" system, which used an IBM AS/400 system as the database server and deployed the application written in IBM RPG directly to the call centre agents' computers.

"Our old call centre application was extremely robust and reliable, but its user interface with keyboard-centric data entry didn't meet today's usability requirements," said Frank Breckle, Glöckle direct application developer. "Also, the company's expansion into alternative industries brought about new requirements that the previous software, with its traditional client-server computing principles and RPG development, could no longer meet."

When planning a new application development and operations environment, Glöckle direct's IT team defined two key requirements for the new platform. "First, we wanted to maintain the 'build-our-own' principle in application development as it would continue to retain process knowledge in-house and keep our dependencies on external consultants at an absolute minimum," explained Uwe Wagensommer, head of IT at Glöckle direct. "Second, from a technology perspective, the new platform should enable us to continue using our reliable AS/400 database server along with a state-of-the-art framework for application development and deployment."

Finally, the new system was required to provide enough openness and flexibility to quickly and efficiently accommodate additional business requirements as Glöckle Group's operations expanded to new areas.

SOLUTION

Early on in the technology evaluation phase, Glöckle direct decided to implement a Java-based environment for application development, as it would enable staff to create all applications and retain expert knowledge about the company's business processes.

Initially, the IT team considered IBM WebSphere as the application server and middleware foundation for the new system. "We quickly realised that IBM WebSphere would be too complex and bulky for our requirements," said Breckle. "Luckily, a Red Hat Business Partner advised us to consider JBoss Enterprise Application Platform."

Glöckle direct now has a number of new applications running on JBoss Enterprise Application Platform as well as other open source and JBoss Enterprise Middleware platforms and frameworks. The main call centre application is still based on the IBM AS/400 database server, but now JBoss Enterprise Application Platform channels user requests and sends the responses to a browser-based application that runs on a Tomcat web server within the architecture.

To meet the business requirements of the marketing department, the IT team also developed a new marketing and lottery application, Marketing- und Glücksspiel-Anwendung (MGA). Designed to accommodate new products and processes very quickly, MGA helps call centre agents serve customers efficiently and with very targeted offers. MGA was built on JBoss Enterprise Middleware platforms and frameworks. The Oracle database component of MGA runs on HP-UX.

"What's great about JBoss Enterprise Middleware is that all the components just work together nicely," said Wagensommer. "We can pick just what we need and keep the total cost of ownership much lower than with proprietary software. That also saves us a lot of time and effort when architecting the system."

BENEFITS

With the strategic choice of JBoss Enterprise Application Platform as the foundation for its new application development and deployment framework, Glöckle direct developed a highly flexible, cost-effective solution that meets the growing business requirements of the customers within the Glöckle Group.

The EJB layer of the JBoss Enterprise Middleware within the system architecture was instrumental in joining the proven and reliable IBM AS/400 database server with state-of-the-art, browser-based Java applications at the front-end. In addition, it simplified the integration of the new MGA application with more than 100 million sets of customer data.

"Compared to alternative solutions such as IBM WebSphere, we have probably saved a high five-digit Euro figure in license costs per year," said Wagensommer. "In addition, we were able to develop and maintain our infrastructure predominantly in-house."

Over the past three years, Glöckle direct has relied on Red Hat Global Support Services to help resolve operational issues. "You can do a lot on your own because it's all open source software and there's a very vibrant JBoss Community out there," said Breckle. "However, when it comes to mission-critical stages of a project, speed matters and we were certainly able to rely on fast response times from Red Hat Global Support Services. Even the lead developer of a Red Hat product has called us to help, which is really impressive.

IGATE CORPORATION

Industry: Information Technology

Geography: Pan India

Business Challenge: To replace its outdated iQuest management tool and design a more reliable, stable, high-performance, cost-effective project management system, which could scale to meet the organization's needs and free it from proprietary vendor lock-in.

Migration Path: IBM WebSphere to JBoss Enterprise Application Platform

Solution: Designed ManageMe application running on Red Hat Enterprise Linux and JBoss Enterprise Application Platform

Software: Red Hat Enterprise Linux Advanced Platform with integrated virtualization, Oracle Database, JBoss Enterprise Application Platform, JBoss Developer Studio, JBoss jBPM

Hardware: Intel processor-based x86 servers from Dell

Benefits: Increased flexibility and choice by freeing the organization from vendor lock-in, decreased IT costs, simplified management, reduced systems maintenance, increased scalability and performance, and obtained the ability to achieve 100 percent system availability

"The success of this initiative has helped us to innovate further and choose the most appropriate set of technologies for our business needs. Open source solutions in general, and Red Hat in particular, will now be adopted as the foundation of our long-term IT strategy. We are confident that Red Hat solutions will enable us to serve our customers effectively every day."

- Chella Namasivayam M., vice president and chief information officer, iGATE

BACKGROUND

iGATE Corporation, a global IT company, provides niche IT Services in enterprise solutions, business intelligence and data warehousing, application development and maintenance, infrastructure management and consulting services in IT strategy, enterprise performance management, business intelligence and analytics and process outsourcing services. The company specializes in business data processing and uses a structure known as iTOPS (integrated Technology and Operations Systems) to meet customer demands.

The company focuses more on productivity, quality, and knowledge management. Customers have benefited from the synergy between investment strategy, technology and business outsourcing provided by iGATE. By 2012, the company wants to be a billion-dollar organization associating itself with 100 of the global 1000 clients.

BUSINESS CHALLENGE

As a business that prides itself on customer service, iGATE needed to give its developers the right tools to ensure that they could service customers both quickly and efficiently. The company decided that it was necessary to replace its outdated Project Management tool, iQuest, due to performance issues. These issues were impacting multiple business functions, including operations, the organization excellence group, finance, human relations, and training.

"There were frequent server shutdowns leading to only 80 percent uptime of the system," said Sandeep Digambar, group project manager, iGATE. As a result, a new mission-critical project management tool that would alleviate these problems was planned for in-house development under the project name ManageMe.

The cost of acquisition for the new application, ManageMe, and the cost of maintenance for iQuest was also a major concern. "We didn't want to spend too much on the procurement of the proprietary licenses for building the new system," said Digambar. "Additionally, the current iQuest system was also utilizing dedicated hardware servers, which translated into added cost for the organization." iGATE wanted to build a system that could rely on a commodity hardware stack, which could be shared between multiple applications.

To achieve optimal performance from the system, it was also essential for iGATE's internal IT team to customize the new system. "We wanted to build the system on an open architecture with no vendor lock-in so that we could customize the system to fit our needs," said Digambar.

As a tracking application, ManageMe requires managing workflows. "It is cumbersome, but the requirement is common with most of the tracking applications," said Digambar. "Instead of writing our own workflow engine, we decided to use a commercial Business Process Management (BPM) tool." iGATE's internal IT team set out to evaluate the right BPM tool for the system requirements from various proprietary and open source options.

SOLUTION

iGATE's internal IT team decided to adopt open source solutions for its infrastructure due to the high availability, performance, and reliability features of open source, as well as for their lower cost compared to proprietary competitors. After evaluating various enterprise-class solutions, iGATE adopted Red Hat Enterprise Linux and JBoss Enterprise Application Platform because the solutions not only met the organization's criteria, but also provided huge cost benefits. In addition, it allowed iGATE to take advantage of the rapid innovation from the open source community and avoid proprietary lock-in.

"We devised a plan to use supported open source solutions to build the new management tool because Red Hat does not have a complex licensing structure and the entire product set, along with Red Hat's reliable support, is provided as one single package through the company's subscription model," said Digambar. iGATE's internal IT team then evaluated various Business Process Management tools including proprietary ones such as OpenLogic and BPLE from Oracle, and open source solutions such as JBoss jBPM. These tools were tested thoroughly on parameters such as application redundancy, clustering, high availability, performance, fault tolerant behavior, and capability of session replication.

"We tested JBoss in the Proof of Concept (POC) for features and performance. The results were encouraging. The cost benefit offered by JBoss jBPM made our decision easy," said Deepu George, divisional head, iGATE.

"Implementation of JBoss Enterprise Application Platform on Red Hat Enterprise Linux was one of the key milestones in our internal IT deployments as it was the first time iGATE was implementing Red Hat's open source solution internally," said Digambar.

ManageMe is a Java-based application running on JBoss Enterprise Application Platform, Oracle 10g as the database server, and Red Hat Enterprise Linux as the base operating system. It took just six months for developing the system. "The system has been tested with 250 GB and 600 concurrent users and there was no noticeable drop in performance," said Digambar. Current database size is approximately 30 GB, but the IT team hopes it plans to increase the usage to 120 GB in a year's time.

The JBoss Developer Studio and JBoss jBPM helped iGATE tremendously in reducing its development and training costs. "JBoss jBPM is less complicated with a visual designer to model the workflow, which means there is much less overhead on our team in terms of its learning and usage," said Digambar.

iGATE is pleased with the ease of deployment and reliability of the Red Hat-based solution in its production environment. "The deployment went smoothly and the performance of the system is top-class with no downtime. The system has performed exceedingly well in production on parameters such as robustness, stability, reliability, and performance," said Amit Shekhar, Project Lead, iGATE.

iGATE's internal IT team has also used the power of virtualization provided as part of Red Hat Enterprise Linux Advanced Platform. With Red Hat virtualization technology, iGATE is able to use the same host hardware for running various operating systems, which enables the organization to reduce costs and optimize resources.

iGATE's IT team is highly satisfied with the performance of the whole system. "We now have an IT policy that supports the usage of enterprise open source software wherever possible," said Digambar.

iGATE's IT team is investigating the opportunity to migrate more of its web-based applications to JBoss Enterprise Application Platform. The first application that is currently planned for migration is its Internal Audit Tracker application, which the IT team is planning to migrate from TomCat to JBoss Enterprise Application Platform. "We intend to move all applications developed and managed by Information systems from proprietary systems to JBoss Enterprise Application Platform," said Amit Goyal, Divisional Head,

Information Systems, iGATE. "We are treating the ManageMe project as a pilot to showcase the power of open source solutions to our end-user customers and proposing its use in their respective projects," Digambar added.

BENEFITS

"With JBoss Enterprise Middleware, we were able to extract tremendous value from our systems and applications," said Digambar. "JBoss platforms deployed on Red Hat Enterprise Linux freed iGATE from vendor lock-in, and enabled the organization to choose hardware platforms that delivered flexibility and high performance."

Usage of JBoss Enterprise Application Platform for ManageMe has provided a stable system and iGATE is experiencing fewer errors, easier ongoing maintenance, and reduced support costs. "The cost savings by our adoption of Red Hat solutions are significant as compared to proprietary available options and this came without compromising performance, scalability, or other features," said Amit Goyal.

Red Hat solutions helped iGATE not only with lower IT costs, but also with simplified management, reduced systems maintenance, and increased scalability and performance from the combination of JBoss Enterprise Application Platform and the Red Hat Enterprise Linux operating system. "Red Hat's expanding presence in the market is proof of its reliability, and we know that Red Hat solutions are viable for a long lifecycle," said Digambar.

The Red Hat platform and middleware solution has also given a huge boost to the overall productivity of the iGATE workforce. "With the implementation of this tool, we are expecting to free the efforts of our employees by 400-500 thousand hours annually," said Digambar.

IT costs have also been reduced significantly. The software licensing costs have been reduced significantly due to the open source nature of the Red Hat solutions. Apart from the lower cost of acquisition, the other reason for lowered software costs is due to the reduced number of application server instances needed for ManageMe, as compared to iQuest.

"The application server instances needed to run the system have been reduced significantly from six IBM WebSphere 6.0 instances to only two JBoss Enterprise Application Platform instances," said Digambar. Additionally the hardware requirement has also come down significantly. "Our hardware requirements have come down by 50 percent as compared to the old iQuest application," said Digambar.

In addition, iGate benefits from Red Hat's reliable subscription model. "Updates can be obtained very easily with our Red Hat subscription and through the Red Hat Network management solution. With Red Hat, we created a more flexible, reliable environment that is easy to maintain. This leaves us free to concentrate on new projects. We're very happy with the flexibility and reliability we've achieved with Red Hat," said Chella Namasivayam M., chief information officer, iGATE.

Today, given the successful proof of concept that iGATE completed with Red Hat, in addition to the organization's ability to resolve its development and production problems through open source solutions, senior management has begun to understand and support open source as a development concept for iGATE's strategic needs. "We are showcasing the success of the project to our end customers and urging them to migrate from a proprietary software stack to a Red Hat open source software stack," said Digambar.

In summary, Red Hat provided iGATE with enterprise-class solutions at an affordable price and without any compromise on performance to ensure 100 percent availability. "The success of this initiative has helped us to innovate further and choose the most appropriate set of technologies for our business needs. Open

source solutions in general, and Red Hat in particular, will now be adopted as the foundation of our long-term IT strategy. We are confident that Red Hat solutions will enable us to serve our customers effectively every day," said Namasivayam.

ALLIANZ AUSTRALIA INSURANCE LIMITED (ALLIANZ)

CATEGORY: Carved Out Costs

INDUSTRY: Insurance

GEOGRAPHY: Australia

BUSINESS CHALLENGE: Rebuild Allianz's IT infrastructure based on a flexible and scalable platform that could leverage new virtualisation technology to generate hardware and support savings, and reduce its underlying software and operations costs for several strategic business application projects

MIGRATION PATH: Windows and Intel-based infrastructure to Red Hat Enterprise Linux on IBM System z10 mainframe and from WebSphere to JBoss Enterprise Middleware on Intel processor based HP ProLiant servers.

SOFTWARE: Red Hat Enterprise Linux, Red Hat Network Satellite, JBoss Enterprise Middleware, JBoss Operations Network, Red Hat Consulting

HARDWARE: IBM System z10 mainframe, HP ProLiant servers

BENEFITS: Significant reduction in middleware software and support costs, reallocation of IT budget from software licensing to staff and resources, resolution of data centre power limitations with new capacity for growth, superior workload management and operational efficiency, reduced carbon footprint, increased flexibility, scalability, and freedom from vendor lock-in

BACKGROUND

Operating across Australia and New Zealand with approximately 3,300 staff, Allianz offers a wide range of insurance and risk management products and services, including personal insurance, industrial and commercial insurance, corporate insurance, public and products liability and workers' compensation insurance.

Allianz is Australia's fourth largest general insurer with over two million policyholders. It also provides some form of insurance cover for more than half of Australia's top 50 BRW-listed companies (2005).

The organisation has been a member of the Australian Government's Greenhouse Challenge Plus program and the wider Global Allianz Group has a target to reduce its greenhouse gas emissions by 20 per cent by 2012.

BUSINESS CHALLENGE

In 2007, Allianz Australia's IT department reached a crossroads as it faced a data centre that was at capacity and network equipment that was at end of life. As a long-standing Windows shop, Allianz saw the situation as an opportunity to build a completely new infrastructure from the network right down to the back-up devices.

"After running WebSphere applications on an Intel platform using Windows for some time, we realised we couldn't go any further with the current set-up," said Peter Rowe Head of Infrastructure and Operations, Allianz Australia Limited.

In addition, Allianz had also reached the limitations of its Windows based operating platform.

"We needed a platform that could give us the flexibility and scalability to enable us to grow and expand for the future," said Rowe.

In line with Allianz's goal to reduce emissions by 20 percent by 2012, the IT department was also interested in examining how it could take advantage of new virtualisation technology to save on power usage.

SOLUTION

After comprehensive analysis, Allianz concluded that the most viable option for the business going forward would be to:

- Combine the performance and reliability of the IBM System z10 mainframe with the flexibility and efficiency of Red Hat Enterprise Linux; and
- Deploy JBoss Enterprise Middleware on the Intel platform – implementing a common system across diverse hardware

In September 2008, Allianz received project approval to commence both infrastructure refreshes concurrently. "IBM System z10 mainframe running Red Hat Enterprise Linux was undoubtedly the best fit-for-purpose solution for us," said Rowe.

"The mainframe offered us the best option for consolidation - Our business took the view that the mainframe was essentially another commodotised piece of hardware and for us, the value really lies in the interface between the server and the applications. Our target was to employ a Red Hat Enterprise Linux based platform that could combine agility with low support costs.

"When assessing operating systems for the IBM System z10, Allianz found that open source based Linux would deliver the best outcome and as a result, selected Red Hat Enterprise Linux, to host all Internet-facing applications, including home and motor insurance quoting, broker and agency pages, premium funding pages, and other broker tools for policy servicing.

"A major part of the decision to use Red Hat Enterprise Linux was its tight integration with the IBM platform and its impressive support structure," said Rowe, "Red Hat's relationship with IBM meant that if we were going to have any issues, Red Hat could essentially replicate the scenario on their own IBM z10." For ongoing, centralised systems management, Allianz also implemented Red Hat Network Satellite to complement the management of Red Hat Enterprise Linux and minimise daily administrative demands.

As a centralised tool, Red Hat Network Satellite can help boost productivity by creating a single template for managing multiple servers more efficiently. It creates a system for grouping machines together and mapping out how to manage them. Having this template also means that one administrator can run the same updates on a greater number of systems simultaneously, and can build a new and completely configured machine within a couple of minutes.

In addition, it has the capability to manage virtualised instances of Red Hat Enterprise Linux through the same interface regardless of underlying hardware platform. Red Hat Network Satellite manages Red Hat on the mainframe in the same manner and interface as Red Hat on a distributed machine.

As an alternative to WebSphere, which Allianz had been using as part of its existing framework, the organisation implemented JBoss Enterprise Application Platform for content management, document management, business process automation, and client portals, along with JBoss Operations Network Management to reduce systems management and resource costs.

"We had three new workloads - BPM, content management, and print services - that had begun on the old system, and which had to be redeployed in order for us to achieve the projected capacity needed for these workloads over the next two to three years," said Rowe.

"JBoss has enabled us to cost-effectively leverage the recent advancement in Intel chipsets in HP hardware, without the costs traditionally associated with multi-core software licensing models," said Rowe.

Allianz enlisted Red Hat and JBoss Consulting services to assist in building automated provisioning of new Red Hat Enterprise Linux and JBoss Enterprise Middleware deployments through Red Hat Network Satellite and JBoss Operations Network in a matter of minutes rather than days.

BENEFITS

Commencing production in late April 2009, Allianz has already observed a number of solution benefits during the testing phase.

"The combination of Red Hat Enterprise Linux, Red Hat Network Satellite, JBoss Enterprise Application Platform and JBoss Operations Network was, and will continue to be, valuable for Allianz, as the solutions enabled the organisation to streamline operations and free up resources for allocation to other high-value tasks," said Rowe.

Allianz will save more than $500,000 in middleware licensing costs, as it deploys JBoss Enterprise Middleware on the Intel platform.

Plus, by using Red Hat and JBoss Consulting and Training, the Allianz team has been able to get up-to-speed in a very short period of time, once again freeing up funds to invest in areas such as staff development, rather than expensive proprietary infrastructure software.

"We were looking for an operating system that we could simply drop down into the network, that was independent of the hardware, and provided a higher level of service with centralised management for patching," said Rowe.

"So far with Red Hat Enterprise Linux, Red Hat Network Satellite, and JBoss Enterprise Middleware, we've begun to significantly reduce day-to-day security administration and now have more strings in our bow in terms of virtualisation and what we can do to minimise both the investment in hardware and ongoing production costs – particularly those associated with escalating licensing and support costs," said Rowe.

Allianz believes this is just the beginning, with expectations that the move from a Windows-based environment to a virtualised Red Hat Enterprise Linux environment, together with the JBoss Enterprise Middleware deployment will save the organisation over one million dollars a year in hardware and support costs.

In addition to resolving data centre capacity issues, the new system is expected to accommodate significant future growth whilst enabling superior workload management and operational efficiencies, and helping to reduce the organisation's carbon footprint.

Allianz also believes it will continue to benefit from the fact that Red Hat has more applications certified to run on Red Hat Enterprise Linux than any other open source operating system platform.

Once the production system is up and running, Allianz will shift its focus toward the continued reduction of proprietary software in favour of more widespread use of Red Hat Enterprise Linux and JBoss Enterprise Middleware.

6. SUMMARY

Every migration project, no matter the size or scope, requires detailed planning to ensure success. Understanding the risks, savings, and cost structure of a migration project is critical if you are to accurately project net improvements and realize actual return on your IT investment.

The considerations and processes detailed in this guide are designed to help you identify migration opportunities, examine the risks associated with various migration scenarios, create a standard build, and help develop a comprehensive strategic migration plan.

Prior to formal planning, an organization must acknowledge the motivations behind the migration, as well as understand the advantages and disadvantages to each potential migration scenario. Lacking this understanding, organizations may be unprepared for decisions and trade-offs that must be made throughout the planning process. Once motivations are clear, organizations should step through each of the five phases of the strategic migration process detailed in this guide. Those phases are:

1. Examine existing middleware architecture and determine the equivalent capabilities in the JBoss Enterprise Application Platform.
2. Examine third-party functional and business applications and determine compatibility or alternatives for the JBoss platform.
3. Measure organizational readiness and overall migration risk.
4. Develop a strategic migration plan, including a detailed road map and cost estimate.
5. Implement the strategic migration plan and employ implementation support strategies.

With this guide and additional Red Hat services, any organization will be armed with the necessary tools for planning and implementing a successful migration. And by combining the technology, training, and mentoring from one source, you will experience reduced development complexity and risk and see the value of your investment faster.

When you are ready to embark on your middleware migration, we encourage you to give us a call to discuss how Red Hat can help you make the right decisions from the start, reduce risk, and accelerate the impact of your deployed technology.


JBoss
by Red Hat

JBoss by Red Hat

CONTACT RED HAT TO GET A PERSONALIZED QUOTE FOR THIS RED HAT SOLUTION

www.redhat.com/consulting
redhatconsulting@redhat.com

NORTH AMERICA
1-888-REDHAT1
www.redhat.com

LATIN AMERICA
+54 11 4341 6200
www.latam.redhat.com
info-latam@redhat.com

EUROPE, MIDDLE EAST AND AFRICA
00800 7334 2835
www.europe.redhat.com
europe@redhat.com

Turkey: 00800 448 820 640
Israel: 1809 449 548
UAE: 80004449549

ASIA PACIFIC
+65 6490 4200
www.apac.redhat.com
apac@redhat.com

ASEAN: 800 448 1430
Australia and New Zealand: 1800 733 428
Greater China: 800 810 2100
India: +91 22 3987 8888
Japan: 0120 266 086
Korea: 080 708 0880

www.jboss.com
#7297537_0811