HRG
Harvard Research Group
Experience - Expertise - Insight - Results

HRG Assessment:

Enhancing Business Agility in Mixed Server Environments

Executive Summary

Services-oriented architectures (SOAs) are receiving a great deal of attention as enablers of business agility. Today, many companies are considering or implementing SOA, but are also dealing with IT complexity in the forms of multiple technologies, languages, and infrastructures that may or may not be standards-based. In addition, heterogeneous mixtures of production environments can make even the most straightforward application integration tasks daunting and unwieldy. IT success today may therefore be tied to enabling organizations to fully leverage all of their key IT assets for competitive advantage.

IBM is attempting to help its customers address these challenges with products designed to extend the functionality of a typical application server environment. These products can help users create messaging-centric, cross-environment implementations with high levels of performance, reliability, and security. They also support the virtualization of both workloads and information assets to allow more effective use of existing systems and networks. The result is lower costs and increased flexibility to meet continuously changing resource demands. Perhaps most importantly, all of these products support "mixed server" environments, with support for a wide variety of application servers and other deployment options.

Business Agility: Key Challenges and Success Factors

A variety of factors have empowered organizations to leverage their IT assets more aggressively to gain and maintain competitive advantage. IT resources are now tied more directly than ever before to the bottom line performance of an enterprise. However, complexity and heterogeneity have increased, business environments have grown more competitive, and the end of the internet "boom" has forced organizations to put greater emphasis once again on traditional business metrics and modes of accountability. Intensifying cost control pressures fly in the face of the growing resource levels required to support IT's greater role in business decisions and operations.

Service-oriented architecture (SOA) has received a great deal of visibility in the past few years as the architectural context for building flexible business systems. The standards and methodologies established via the Web services model provide a basis for more effective integration of disparate systems. Yet today, many organizations are by necessity focused on dealing with a mixture of heterogeneous servers and deployment environments across which applications must interoperate and communicate in order to effectively support business objectives.

The high level IT challenge for organizations today is to achieve the levels of agility and flexibility required to meet committed business goals in a secure and cost-effective manner. While approaches that address this overall

As part of, or in addition to, evaluating SOA as an architectural direction, many organizations today are focused on leveraging existing resources and assets with high levels of efficiency and agility. These assets may include a heterogeneous set of server and deployment environments, and solutions that effectively meet business needs must support all of them regardless of their source. This "mixed server" reality is being addressed by IBM with a set of WebSphere products enhanced both to address key solution requirements and to support a wide variety of deployment environments.

Copyright © 2007 Harvard Research Group, Inc.

challenge may vary across organizations, a number of common requirements and metrics for success apply:

- ***Providing support for mixed server environments.*** Customers need to support all of their IT assets to meet their business goals. They may need to run applications across a wide array of systems and application servers. Merger/acquisition activities, as well as internal consolidation of IT assets, often serve to intensify this requirement. Customers must therefore place deployment platform independence high on their lists of requirements.
- ***Building a flexible infrastructure that can easily and quickly adapt to changing business needs.*** This infrastructure must allow for a more cost-effective and efficient utilization of existing resources to keep costs and new expenditures low. Perhaps more importantly, customer and business partner satisfaction levels depend on the ability to allocate compute power where and when it is needed to meet dynamic utilization demands with sufficient performance. Virtualization of a variety of IT asset types is a key enabler of this flexibility, including the ability to virtualize assets across a variety of deployment environments.
- ***Meeting quality of service (QoS) levels for transaction-based systems.*** Agile businesses require support for handling large and often unpredictable transaction loads in order to meet service level requirements of customers and business partners. Standards-based messaging systems are a viable solution for increasing interoperability across disparate systems and providing persistence mechanisms to ensure transactional integrity and reliability.
- ***Providing Robust Security.*** High volumes of external transactions serve to accentuate the security challenges already facing IT managers today. Organizations need robust intrusion protection to prevent the exposure of sensitive information through improper access or other means.

While many vendors offer products designed to address some or all of the above requirements, the specific messaging and cross-environment challenges inherent to these requirements necessitate a more focused approach than generally available via a typical infrastructure software “stack”.

IBM’s WebSphere Products for Mixed Server Environments

IBM has recognized that in order to more effectively address the requirements outlined above, it needed to provide products that address QoS, flexibility, and security issues. IBM has established itself as a leading provider of flexible application infrastructures. The foundation of this leadership is IBM’s WebSphere software platform. Through its SOA related announcements in October, 2006, IBM has clearly demonstrated its commitment to SOA and enhancing the comprehensive nature of its infrastructure solutions. IBM has also made these products available on non-WebSphere platforms, making it possible for those organizations that use application servers from a wide range of vendors and sources (or who deploy without application servers) to leverage their capabilities. In doing so, IBM is offering its customers the ability to extend their existing application server environments to more effectively meet the service level and interoperability challenges they face.

WebSphere Extended Deployment: Boosting QoS through Application and Information Virtualization

It is very often difficult to meet QoS requirements when the dynamic nature of transaction levels and interactions with various parties make workloads unpredictable. Upgrading production applications can also degrade service levels when systems must be brought down. It can be expensive and time-consuming to maintain non-production environments for staging new application versions. Finally, network bottlenecks can also cause poor service levels; the bottlenecks arise when applications repeatedly access data and information stores.

Virtualization is often utilized to dynamically allocate servers and storage platforms in order to apply these compute resources where and when needed. However, applying virtualization requires that issues relating to potential loss of service be directly addressed. As an example, sensing and responding to hardware platform problems in order to drive virtualization activities is an important way to maintain QoS levels. Doing so means monitoring all systems and applying policies reflective of QoS requirements. These policies need to either manually or automatically take action to prevent service outages.

A potential way to better address QoS issues is to extend the virtualization concept to include both applications and information sources. Any solution designed to do so, however, must incorporate the ability to virtualize assets across a heterogeneous collection of platforms on which they may be deployed. This "mixed server" approach ensures that all deployment scenarios (including applications servers) can be extended as part of a virtualization plan that optimizes asset usage and performance.

IBM's WebSphere Extended Deployment (XD) is designed to help users implement effective workload and service virtualization that complements hardware virtualization. WebSphere XD's dynamic response to changes in complex workloads reallocates virtual resources, allowing organizations to be more flexible and therefore to more consistently meet required service levels. Complex workloads that can most effectively leverage WebSphere XD's virtualization capabilities are those which contain multiple applications whose peak periods of usage can be substantially different, or those that employ large monolithic applications utilizing resources across many servers in order to address high transaction volumes.

WebSphere XD can also provide a centralized point for workload management and operational control. The service and health policies of all application servers can be managed by the WebSphere XD environment in which they operate. This capability enables the management of service and health policies for all application servers included in the environment served by WebSphere XD. For instance, WebSphere XD could be used to front-end a server farm to which it distributes work requests to application servers (WebSphere and non-WebSphere). Applications running on non-WebSphere application servers can run as-is and would not need to be migrated to a WebSphere deployment environment. In addition, a feature of WebSphere XD would run on non-WebSphere application servers that would communicate workload status back to WebSphere XD as input to service and health policy decisions. In this way, an existing, heterogeneous application server environment can be extended in ways that permit more effective workload management.

WebSphere XD is therefore focused on the virtualization of an organization's software infrastructure with the goal of consistently meeting service levels, regardless of how activity levels may fluctuate. Two types of virtualization are possible with WebSphere XD:

1. *Workload Virtualization*: WebSphere XD can dynamically allocate and manage a pool of application server resources based on the types and volume of workloads using 3 different approaches.

   Service Level Management - Users can create policies that represent service level goals. These policies are then used by WebSphere XD to classify and prioritize workloads and route them intelligently to the proper resources.

   Application Edition Management – This extends the notion of application management to include multiple versions of applications and services. Deploying new application versions in traditional environments can cause loss of service. While administrators can get around this by creating duplicate staging systems and custom routing algorithms to transition these new versions into production, this solution is not ideal. WebSphere XD's Application Edition Manager enables different application editions to coordinate their request routing and other activities. In this way, multiple deployments of a given application can be supported, thus making it possible to keep multiple versions in production at the same time. Requests can be allocated to different versions based on pre-defined routing rules. Updates can be rolled out seamlessly. It is also possible to orchestrate reversion to older versions when necessary.

   Health Management – This permits administrators to identify and mitigate server health problems before they cause production outages. Health policies are defined and conditions monitored, and problems can be handled in three ways. Actions in response to problems can take place "manually", in "supervised mode" and "autonomically".

   - *Manual mode* - An operator is informed of the problem via an alert that identifies which application server needs to be restarted.

- *Supervised mode* - WebSphere XD recommends corrective action and asks an operator for approval to take the action.
- *Autonomic mode* - WebSphere XD informs an operator it has taken a specific action after the fact.

2. Information Virtualization: This unique capability recognizes that even when application resource allocations are optimized, access bottlenecks to shared information resources can severely limit application performance. One way to alleviate the effects of this problem is to persist application sessions in order to limit the need to access information sources. WebSphere XD includes a feature called **ObjectGrid**, a distributed caching framework for storing and sharing Java object data by multiple applications. Each server can have its own ObjectGrid, thus making the framework highly scalable. Because many or all servers in an ObjectGrid can cache objects redundantly, the effects of a specific server's failure on service levels can be minimized or eliminated. ObjectGrid can be used for HTTP session management, allowing the storage of short-lived session data within the application tier. WebSphere XD also includes a **Partitioning Facility**, which permits the design of applications that can separate logic and data into partitions. These partitions can then be mapped to available servers. Users can utilize "Partition-aware routing" to route requests to the right servers, allowing information to be cached much more efficiently. This approach can result in performance improvements and optimized scalability as the transaction load increases because server resources are used more efficiently.

WebSphere XD's infrastructure virtualization capabilities are designed to enable service levels to be met consistently and predictably. This is accomplished by permitting the fast adaptation of application server resources (including non-WebSphere application servers) to changing workload demands, reducing deployment complexity, and relieving backend data store access loads. In addition, as a result of the service level and health management features as well as the distributed and redundant information caching capabilities of WebSphere XD, scaling of the application environment can be highly efficient.

Previous versions of WebSphere XD supported non-WebSphere J2EE-based application servers as well as .NET, but with only a subset of service policy capabilities and with none of the Health Management features. With the introduction of Version 6.0.2 in October 2006, IBM articulated plans to extend broad service and health policy support for virtualized workload management and new, customizable health policy and action features, to non-WebSphere servers. These servers include (but are not limited to) PHP Server, Apache Tomcat, Ruby on Rails, JBoss Application Server, and BEA WebLogic Server. Plans were announced to introduce this support during the first half of calendar year 2007.

WebSphere MQ: The Backbone for Cross-Environment Messaging

As noted earlier, effective and reliable messaging is a critical requirement to support the transaction-based messaging on which many businesses depend. Messaging systems therefore focus on effective data and information exchange. A couple of factors have combined to intensify the challenges associated with this process. One is the growing requirement to create interoperability among disparate applications that support critical business processes. The second is the increasing emphasis on systems that meet more rigorous compliance requirements. While messaging standards can help in this effort, the application of such standards must first take into account the potential variety of languages and technologies that already exist within an organization's application portfolios.

WebSphere MQ has long been IBM's messaging infrastructure to support the effective exchange of data and information among disparate systems and platforms. IBM continues to work to make WebSphere MQ a versatile "messaging backbone". The goal is to enable organizations to focus on their core competencies and business goals and not on modifying their applications to accommodate messaging requirements. To illustrate this, it is helpful to note that in today's programming environments, where the use of Java for application programming is widespread, the belief may exist that messaging needs must be met through adherence to the Java Message Service (JMS) standard. However, while JMS is probably the most widely used API for messaging, a distinction must be drawn between messaging APIs and the messaging engine that implements them. While a typical JMS engine is adequate to implement messages based on this standard and to achieve required quality of service levels, such an engine is not sufficient in the following cases:

1. Where non-Java applications exist
2. Where other languages and programming styles are preferred
3. Where different quality of service levels may be required by differing application environments.

For this purpose, and to complement the native programming interface for WebSphere MQ (known as MQI), IBM has created **IBM Message Service Clients** (also known as **XMS)**, a set of messaging clients and APIs that permit the rendering of JMS in a variety of different languages (C, C++, C#, Visual Basic, .NET, in addition to Java). Consistent with IBM's focus on enabling its products on non-IBM platforms, XMS permits users who have standardized on other technologies, such as .NET, to leverage WebSphere MQ messaging functionality or to integrate non-Java applications with new J2EE applications and services without needing to learn specific WebSphere MQ APIs. WebSphere MQ's support of XMS and JMS in addition to MQI make it attractive for situations where applications must run across a variety of deployment environments (such as .NET, SAP, Oracle, and any combination of J2EE application servers).

WebSphere addresses the issue of transactional reliability through a mechanism that ensures that all operations associated with a transaction are completed as an integral part of delivering a message to its target system. Very often, transactions will involve a single unit of work that must include updates to multiple resources and operations (which in turn may be tied to "suboperations" that need to be executed as well). WebSphere MQ has been designed to support transactions "natively", i.e., a message can be defined as a transaction that includes all operations and updates to resources such as databases. In this way, all operations associated with a transaction are completed before a message is finally committed on the target system. This capability is provided through a combination of the control provided by WebSphere MQ's built-in transaction manager, as well as the asynchronous nature of the messaging environment itself. Messages are routed between queues by the product's Queue Manager. Many Queue Managers are typically deployed across an organization, which could then be linked together, if required, to form a single "virtual" Queue Manager. This "clustering" of Queue Managers can increase overall reliability and contribute to the scalability of a server which must handle large numbers of messages.

Writing messages to non-volatile storage such as hard drives – "persisting" them - provides a means of ensuring that messages are not lost in case systems fail with transactions only partially completed. While a vast majority of messages do not have to be persisted (because they are transient and therefore non-critical in nature), in some cases it is not appropriate for a query to be reissued. In other cases, the message may involve a specific update or key information which must not be lost. To satisfy this need, WebSphere MQ is capable of writing messages out to a queue persistently (via hard disk storage). It is important to note that IBM distinguishes the different methods by which WebSphere MQ persists messages from other approaches available today. For example, the JMS specification permits implementations to perform what are called "lazy writes". These involve the persisting of a message to memory before letting the message go, without confirming first that the message had been persisted to disk. While this can provide a net performance boost over a more robust persistence approach, it introduces risk if the system fails. WebSphere MQ allows for this mechanism to be selected, but in those situations where message and transaction reliability are non-negotiable (such as with many financial service transactions), full, disk-based persistence is required, and is natively supported by WebSphere MQ.

The requirement to keep messages secure is also becoming more important over time as organizations use SOAs and other designs to leverage their IT assets for use beyond their office walls. WebSphere MQ has always been able to encrypt messages for transmission to another Queue Manager. This is important, because when a message is sent to a destination system, it sits on a queue before being read by an application and must therefore be protected from unauthorized access. Today, an edition of the product, WebSphere MQ Extended Security Edition, combines WebSphere MQ and Tivoli Access Manager for Business Integration into a single package. This combination allows messages to be encrypted end-to-end between applications, rather than just between Queue Managers. This adds greater security and privacy for messages throughout the business. Additional features include centralized, browser-based, remote security policy administration, and the use of a message-level audit function with records that can support compliance efforts.

The ability for WebSphere MQ to work with a variety of IT infrastructures cannot be overstated. In fact, one of the major benefits of WebSphere MQ is that it can support environments that include multiple application server brands as well as other internally-developed or commercially-available deployment platforms. While many of these have their own messaging infrastructures, WebSphere MQ can provide the cross-environment functionality required to enable messaging and data sharing among, and independent of, all systems and platforms. WebSphere MQ supports over 80 platform configurations, and, as noted earlier, is not dependent on any one language or technology in order to create or execute messaging calls from any application.

WebSphere DataPower SOA Appliances: Specialized Hardware for SOA

Service connectivity is top of mind for many organizations today. It's all about quickly and securely connecting systems, users, business partners and channels in a way that leverages open standards to reconcile disparate platforms and technologies. In many cases, connectivity can involve the integration of legacy applications. These applications may be written in a variety of languages that employ disparate and sometimes non-standard messaging formats. As Web services standards and SOAs are fast becoming the norm, XML - more accurately, XML messaging - is considered the technology of choice to resolve connectivity issues.

In October, 2005, IBM acquired DataPower, which offers a variety of appliance products aimed at improving XML Web Services integration, security, performance, and scalability. What is most interesting is IBM's decision to place DataPower organizationally within the WebSphere organization despite the fact that DataPower products are rack-mountable hardware appliances, or "network devices". This decision may be easier to understand when considering these products' focus on middleware technology and "any-to-any" messaging which ties into IBM's overall Web service and SOA software strategy. In addition to handling non-XML messages with its integration appliance, the appliances employ broad support for a wide variety of standards, including XML/SOAP, WS-Security, SAML, WS-Trust, Kerberos, WSDM, WS-Policy, XKMS, LDAP, RADIUS, and numerous others.

There are currently three major products that make up the WebSphere DataPower product line. These appliances have broad interoperability and integration capabilities with many IBM products including WebSphere and Tivoli products, as well as with heterogeneous environments such as BEA, Microsoft, Sun, TIBCO, and others:

- *WebSphere DataPower Integration Appliance XI50*: The device is essentially a middleware appliance purpose-built for application integration and designed to transform disparate messaging formats (e.g., binary, flat files, XML, and various legacy formats including COBOL Copybook, CORBA, and others), and to then provide secure message routing, transport mediation and connectivity with support for MQ, HTTP, FTP, and JMS. This "any-to-any" messaging support is particularly useful to "SOA-enable" existing systems, and to accomplish the often difficult task (especially when legacy applications are involved) of application integration in an easy to deploy appliance.
- *WebSphere DataPower XML Security Gateway XS40*: This hardware appliance serves as a security-enforcement point and Web services firewall for SOA transactions. It is an effective means of mitigating both incoming and outgoing attacks that can threaten both transactional integrity and data/information security. The product supports XML firewall filtering, digital signatures, encryption, schema validation, and other security mechanisms and standards (including WS-* and SAML). Appliances can be configured via a Web-based administrative console, command line interface, and various IDE tools. The console allows users to drag and drop filters and transform into XML pipelines. This simplifies configuration and administration. Both incoming attacks and outgoing threats (which may originate from backend servers and adversely affect business partners) are all handled by the XS40.
- *WebSphere DataPower XML Accelerator XA35*: This product was created specifically to raise service levels by improving XML processing performance (and ultimately, QoS). It offloads XML, XPath, and XSLT processing from general purpose servers, and employs XML pipeline processing to help accelerate XML message processing for better performance and higher scalability. Capacity can be shared across multiple applications, making the creation of a standardized, unified, and shared XML infrastructure possible.

WebSphere DataPower SOA Appliances have many functions integrated into a single simple and secure device with non-programmatic interfaces which are easy to configure. They leverage XML Web services and a host of security and messaging standards to achieve interoperability and integration across disparate systems. While ease of configuration, use, and maintenance are key features of these appliances, the bottom line is that these devices enable customers to achieve high service and security levels for their transaction-based applications across heterogeneous IT environments.

Summary and Conclusions

Many organizations today are facing increasing application integration demands, as well as the need to scale capacity in a secure, reliable, and cost effective way. A number of vendors provide comprehensive suites of software that address many or all of the functional requirements for developing, deploying, and managing SOA-based systems. However, they may still not meet the needs of many customers today. The reality in many organizations is that heterogeneous systems and infrastructures are an integral part of their IT environment and will stay that way for some time. These may include applications, messaging methodologies, application servers and various other technologies. Solutions must therefore accommodate all of these assets and approaches, thus creating special implementation and performance challenges requiring more than just a highly functional SOA software “stack”.

IBM has recognized the need to extend its software offerings with a number of products that can address these significant challenges. The products are specifically focused on enabling application integration more easily and reliably, and with greater performance, using standards that make the resulting solutions forward-looking in terms of conforming to evolving best practices. One of the most significant characteristics of IBM’s approach is the company’s apparent steadfast commitment to supporting mixed server environments, with emphasis on configurations in which a variety of application servers and other deployment options exist together. The ability to work with standardized technologies such as JMS, while at the same time maintaining and abstracting technologies that underlie legacy applications and messaging middleware, is particularly appealing as well.

HRG believes that IBM will earn its stripes by being the best in the market at delivering all claimed benefits specifically where a mix of non-IBM platforms and application servers are the norm. IBM is therefore highly motivated to deliver messages to the marketplace that emphasize its ability to improve the performance, reliability, and security of integrated application environments regardless of the infrastructures on which they are deployed and the technologies used to build them. While IBM’s success here will serve to reinforce customers’ use of existing non-IBM deployment platforms, IBM is placing its bets on its ability to resolve customer challenges with quality and neutrality, regardless of the customers’ IT environment.

Harvard Research Group is an information technology market research and consulting company. The company provides highly focused market research and consulting services to vendors and users of computer hardware, software, and services. For more information please contact Harvard Research Group:

Harvard Research Group™
PO Box 297
Harvard, MA 01451 USA

Tel. (978) 456-3939
Tel. (978) 263-3399

e-mail: hrg@hrgresearch.com

http://www.hrgresearch.com

Copyright © 2007 Harvard Research Group, Inc.