Applying Mainframe Experience to the Distributed Environment
Table of Contents

**Executive Summary** ..................................................................................................................... 1

**Different Mindsets** ..................................................................................................................... 2  
  > Conserve versus Consume ........................................................................................................... 2  
  > Diversify versus Standardize ........................................................................................................ 2  
  > Leveraging Hindsight .................................................................................................................. 2

**Change Management** .................................................................................................................. 2  
  > Balancing Stability and Agility ..................................................................................................... 3

**Virtualization** ............................................................................................................................... 3  
  > The Challenge of Virtualization in the Distributed Environment ............................................... 3

**Clustering** .................................................................................................................................... 4  
  > Distributed Clustering Approach ............................................................................................... 4  
  > Mainframe Clustering Approach ............................................................................................... 5

**Resource Standardization** .......................................................................................................... 6  
  > Freedom in the Distributed Environment .................................................................................. 6  
  > Standardization in the Mainframe Environment ......................................................................... 6

**Conclusion** .................................................................................................................................... 7
Executive Summary

“Those who cannot learn from history are doomed to repeat it.”
George Santayana, philosopher

Today, IT organizations are striving to increase the maturity level of their IT service management. They are trying to run IT as a business. In pursuing this goal, people in the distributed environment face many of the same challenges that those in the mainframe environment have faced and successfully addressed in the past.

Mainframes and distributed systems, previously looked at as separate entities, are now rapidly coming together into a single, distributed IT environment. As a result, people from two previously separate and very different cultures — mainframe and distributed — are coming together to architect, develop, and manage the resulting unified infrastructure.

Each of the two groups has arrived at this point through two very different evolutionary paths. In evolving along those paths, both groups have gained considerable, albeit different, experiences along the way and have learned valuable lessons from these experiences.

This paper examines some of the major differences between the mainframe culture and the distributed culture. It also discusses how people in the distributed environment can learn from the experiences of those in the mainframe environment. The paper focuses on four major areas:
> Change management
> Virtualization
> Clustering
> Resource standardization
Different Mindsets
The mindsets of the mainframe group and the distributed group differ substantially due to their very different evolutionary paths. There are two areas in which the differences are especially profound: resource consumption and resource diversity.

Conserve versus Consume
First, there is the difference concerning resource consumption. People in the mainframe community have grown up in an environment in which resources, such as CPU cycles, have been expensive. As a result, conservation of computing resources has been a key issue for the mainframe application developer.

Compare this to the distributed environment, in which resources have been relatively inexpensive. People in that environment have grown up with a perspective of simply adding resources, rather than spending development time on minimizing resource consumption. This has resulted in applications that are not usually designed with resource conservation in mind. Applications typically have a whole server dedicated to them, so application developers tend to fully use all server resources.

Diversify versus Standardize
A second difference in mindset has to do with resource diversity. People who have grown up in the distributed environment have always enjoyed wide freedom in selecting hardware and software. That’s because they have always had a wide choice of hardware and software from a large number of vendors.

Freedom of choice has resulted in wide diversity in the distributed infrastructure. We find multiple server and PC hardware architectures, such as Intel-based and RISC-based. There are multiple operating platforms that might include UNIX (multiple flavors), Windows, and Linux. In addition, there is a wide variety of commercial software from multiple vendors — enterprise applications, databases, and personal productivity software.

Compare this to the mainframe community. Here, people have grown up with only limited choices in hardware and software. In fact, many of the mainframe installations early on were single-vendor. As a result, people in the mainframe community have typically leaned toward standardization of resources. They deploy standardized hardware, usually from a small number of vendors or even a single vendor, and they deploy standardized operating platforms, programming languages, and database management systems.

Leveraging Hindsight
Today, IT professionals are under intense pressure to improve service levels while at the same time cutting costs. In response, people in the distributed community are evolving their approaches to system architecture and management.

Driven primarily by its major role in mission-critical applications, the mainframe group has evolved particularly strong practices and processes in system architecture and management, especially in change management, virtualization, clustering, and resource standardization. People in the distributed community can learn from the different mindset of the people in the mainframe community in these areas.

In fact, mainframe hindsight has already proven valuable in the creation of the IT Infrastructure Library (ITIL®). Some of the ITIL processes have their origins in mainframe practices. In the early 1980s, original systems management concepts, many of which had been developed in the mainframe environment, were documented in a four-volume series, often called the “yellow books.” These books define and describe a generic model of 42 processes called the Information Systems Management Architecture.

The yellow books were key inputs to the original set of ITIL books. The original processes specified in the yellow books have been adapted and enhanced in ITIL to meet the more demanding requirements of the distributed environment.

In addition, Business Service Management (BSM) solutions are available that leverage mainframe hindsight. These solutions support best practices for service management processes, of which many originated in the mainframe environment. BSM is an approach to service management that combines best practice IT processes (including support for ITIL), automated technology management, and a shared view of how IT resources directly support the business.

The following sections will discuss, for each of the four areas, how hindsight from experience gained from the mainframe community can help the distributed community avoid pitfalls.

Change Management
Because of the major implications of a mainframe outage on the business, the reliability and stability of the mainframe are fundamental tenets of the mainframe mindset. One of the major threats to stability is change. That’s why those responsible for mainframe systems have developed strong control over changes to ensure a high level of stability. Over the years, these people have evolved rigorous change management processes to ensure that changes to the
mainframe are properly approved, tested prior to production, and implemented in a way that minimizes business disruption.

Some organizations have taken mainframe change rigidity to the extreme. A large bank, when asked how often their IT organization implements changes to their database management system, replied, “March 30.” They deploy changes to the database only once a year on that date. Needless to say, although lacking in agility, that database is extremely stable.

Balancing Stability and Agility

Mainframes have long been the backbone of mission-critical systems. As such, mainframe change processes have evolved over time with a focus on stability.

Today, business-critical applications often involve distributed system components, as well as mainframes. In many cases, these are complex, multi-tier applications. Because of their importance to the business, it’s essential to ensure the overall stability of these applications. That means ensuring the stability of the entire application stack — not only the mainframe, but also the distributed system components. As a result, it’s essential that those in the distributed community evolve their change management processes to the level of mainframe processes.

In the distributed environment, people have usually opted for agility, often at the expense of stability. This is evidenced by the fact that change to the distributed IT infrastructure continues to be one of the main sources of system outages. An inadequately planned or improperly executed change can bring down the component being changed as well as the other components dependent upon it. Even a change successfully implemented on one component can have disastrous consequences on other related components.

For example, due to inadequate planning, a change to an application running on an application server, although implemented successfully, can inadvertently lock out all the clients that need access to that application.

Those responsible for distributed systems can improve stability by applying some of the more rigorous change management processes that have evolved in the mainframe environment. These include processes for hardware changes and software changes, such as patches, software updates, and new software deployments.

The main objective of effective change management is to achieve the optimum balance between agility and stability. In determining the level of stability required, and hence the rigidity of change control necessary, it is important to know the business services and processes supported by the various infrastructure components and the business criticality of those services. That’s because the more critical the business services or processes supported, the higher the level of stability required, and hence the tighter the change control on the components that support those services or processes.

> In summary, develop and use a change management system to record every type of change. Even if the system is a spreadsheet, it will save you hours, if not days, when tracking down problems.

> Look at the business requirements when defining a platform for application deployment.

Virtualization

Due to the mindset in the distributed environment that hardware is relatively inexpensive, many data centers are now suffering from “server sprawl,” with lots of servers to maintain and manage. This is not only driving up costs, but also has resulted in low server utilization rates, and that means a low return on investment.

Today, IT is under intense pressure to reduce costs. In an attempt to reduce the number of distributed servers and increase server utilization rates, those in the distributed environment are looking to server virtualization. Virtualization provides the ability to run multiple server images — called virtual servers — on a single, physical host server. Each virtual server appears to the environment as a separate physical server.

The Challenge of Virtualization in the Distributed Environment

In virtualizing distributed servers, the IT staff faces two major challenges: the application challenge and the management challenge. First, we’ll examine the application challenge.

Applications in the distributed environment have typically been designed for the dedicated server environment; that is, the application developer assumes that the entire server is dedicated to the application. As a result, applications tend to consume resources based on the gas volume law, which states that the volume of a gas expands to occupy the full volume of its container. In this case, it means that the applications tend to freely consume all the resources of the server. If the application runs out of resources, the tendency has been to simply move it to a bigger machine.

Merely virtualizing these resource-hungry applications and smashing them together into physical host servers does not result in optimum utilization of server resources. These applications will not be “good citizens” in the virtualized environment.

With respect to the management challenge, virtualization significantly increases the complexity of the infrastructure,
transforming it from a relatively static environment to a highly dynamic environment. Traditional distributed system management tools, however, were designed for a static server environment and not for the dynamic virtualized environment.

For example, with some event management products, the IT operations staff can define a set of rules that help an administrator understand the implications of server events on applications and jobs. These rules are usually based on the assumption that the server environment is static; that is, each physical server is dedicated to running certain applications and jobs. With virtualized servers, however, the environment is dynamic. A physical host server can play multiple roles from an operating system perspective, an application perspective, and a workload perspective, and those roles change rapidly. Operating systems and applications may bounce around among different physical servers as workloads change. In this highly dynamic environment, rules based on a static environment may no longer apply.

Those in the distributed environment are just beginning to move into virtualization technology. This technology was originally developed for the mainframe more than 30 years ago. With the advent of logical partitioning (LPAR), the mainframe architecture evolved from a physical to a virtual architecture. The IBM VM (Virtual Machine) operating system, released with the IBM System/370 back in the 1970s, exploited this architecture.

People who work with mainframes have learned through experience that it is not prudent to simply jump into virtualization. They must first address application and system management issues.

Mainframe application developers have been building applications in the shared environment for decades. During that time they have developed a good citizen approach. It can be summarized as, “I am on a shared resource and I am not the only application using that resource, so I have to be careful of my resource consumption.” As a result, applications developed for the virtualized mainframe environment are resource efficient. Mainframe developers have learned that spending the extra development effort upfront to achieve resource efficiency pays off later in that there are less computing resources to operate and manage. That keeps costs down and resource utilization up.

Even pre-virtualization mainframe applications have been developed with resource conservation in mind. When these applications are moved to a virtualized environment, they continue to operate as “good citizens,” conserving resources. In addition, mainframe system management tools have evolved to accommodate the virtualized environment.

People in the distributed environment can learn from mainframe experience in two major areas when it comes to virtualization:

> Develop every application as if it were for a virtualized environment. This will keep you out of trouble when using virtualization to balance resources.

> Invest in the tools and education to manage the virtualized environment.

It’s interesting to note the experience of a distributed application developer at a BMC customer who moved over to mainframe application development from the distributed environment. In his new assignment, he has been influenced by the mindset of the mainframe application developers to conserve resources and be good citizens in the shared environment. The developer remarked that with this new approach, he has increased his value as an application developer. He added that this mindset has value, whether he is developing applications for the mainframe or the distributed environment.

**Clustering**

Cluster has usually been employed in the distributed environment for two major purposes: scaling and redundancy. In both of these areas, a mindset of proliferation has resulted in a vertical approach to clustering.

**Distributed Clustering Approach**

With respect to scaling, the mindset has been to simply add more servers when the workload grows beyond the processing capacity of a single server. As a result, many applications have multiple servers dedicated to them. (See Figure 1.)

![Figure 1. Vertical scaling](image)
In redundancy, the system provides automatic failover to a backup server in the event of a failure in the primary server. In the distributed environment, both primary and backup servers are usually dedicated to a single application. Here again, every application has multiple servers dedicated to it resulting in a vertical approach to scaling. (See Figure 2.)

![Figure 2. Vertical redundancy](image)

**Mainframe Clustering Approach**

In the mainframe environment, clustering technology is more highly evolved than in the distributed environment. Parallel Sysplex, introduced in 1997, allows multiple, co-located physical mainframes to be integrated into a cluster, with the cluster appearing as a single system image that is managed from a single point. Geographically Dispersed Parallel Sysplex (GDPS) extended the capabilities of Parallel Sysplex to allow the physical mainframes in the cluster to be geographically dispersed. Parallel Sysplex and GDPS include both mature hardware and software to implement the cluster, and they include mature system management software to permit effective management of the cluster.

An important lesson learned in the mainframe environment is the need to augment virtualization with clustering. Initially, people in the mainframe environment saw virtualization as a great way to reduce the number of physical mainframes. They could run multiple applications in a single physical machine as opposed to having a machine dedicated to each application. This, however, created a problem. When a physical machine went down, all virtualized applications running on that machine also went down; whereas in the non-virtualized, dedicated environment, only the applications running on the failed machine went down.

The mainframe staff learned that combining clustering with virtualization provides a way to enjoy the benefits of virtualization without jeopardizing availability. The combination of clustering with virtualization permits a horizontally sliced architecture in which multiple applications are dispersed across multiple physical mainframes. (See Figure 3.)

This approach permits IT to reduce the number of machines without sacrificing performance and availability. It provides efficient use of the machines through load balancing, which dynamically allocates computing resources to the heaviest workloads, ensuring fast performance, even under heavy workloads. It also permits redundancy to provide backup in the event of a machine failure.

Another benefit gained from the horizontal slicing approach is that it permits rotating maintenance to minimize the impact of maintenance on service delivery. For example, a large

![Figure 3. Combining clustering and virtualization](image)
European company has implemented a cluster architected similarly to the system illustrated in Figure 3. With this architecture, the operations staff can temporarily take down a single machine for maintenance, such as for microcode upgrades. That machine’s workload is taken up by the other machines in the cluster. (Of course, the staff does the maintenance during periods of relatively low workload so the other machines can take up the increased workload without jeopardizing performance.) When the maintenance is completed, the staff returns the machine to service. As a result, the staff can perform maintenance on the machines, one-at-a-time, without disrupting service delivery.

In summary, there are two ways that clustering can be improved when deployed in a distributed environment:

> Combine virtualization with a clustering strategy to get the most out of both technologies.
> Rotate maintenance to minimize service disruption.

**Resource Standardization**

People in the distributed environment differ markedly from those in the mainframe environment with respect to the diversity of their computing resources.

**Freedom in the Distributed Environment**

In the distributed environment, freedom of choice has always been a way of life, resulting in a wide diversity of hardware, operating platforms, application programming languages, databases, and packaged applications. This has created an IT infrastructure that is extremely difficult to manage and maintain. For example, there are diverse application stacks that require multiple skill sets to manage and support. That means multiple application development groups and multiple operations groups using different system management tools. Stability suffers in that IT may not have expertise in all areas to effectively maintain all systems. What’s more, inordinately long problem resolution times may result, reducing availability.

**Standardization in the Mainframe Environment**

Early on, mainframe application developers operated in an environment in which they had a very limited choice of programming languages and databases on which to build their applications. The choice then widened as new programming languages and databases emerged.

The mainframe operations staff soon discovered, however, that freedom of choice resulted in a highly complex and difficult to manage environment. As a result, the people in the mainframe environment opted to impose standardization, limiting the choices of programming languages and databases for application development. Some, for example, have standardized on COBOL for business applications (PL1 has just about disappeared), and some have standardized on DB2®. Mainframe application developers also are considering Java, but they are looking at its adoption from a strategic perspective rather than just jumping into it.

In some large enterprises, the mainframe staff is using its substantial buying power to encourage application software vendors to provide applications on the organization’s preferred platforms.

Convergence in standardization across platforms offers benefits to distributed and mainframe organizations:

> Numerous technologies are available on both platforms (e.g. J2EE, Linux, Websphere, Oracle and DB2) allowing independent selection of development environments.
> Platform selection can be driven by business requirements for stability and performance.
Conclusion

As the mainframe and distributed environments merge, the distributed staff and the mainframe staff can learn much from each other in the architecture, development, and management of enterprise systems. This paper focuses on those lessons that people in the mainframe environment have learned over the long history of the mainframe — lessons learned the hard way, by making mistakes. By applying these lessons, people in the distributed environment can accelerate their progress toward meeting management’s mandate to improve service quality while lowering costs. They can further accelerate their progress through the deployment of BSM solutions that implement, enforce, and foster the use of processes that are based on mainframe lessons.

Both the mainframe and the distributed communities benefit, because as their two environments merge, their dependence on each other’s success increases. By combining their collective expertise, they can ensure mutual success, and that goes right to their organization’s bottom line.

For more information, visit www.bmc.com/mainframe.
About BMC Software
BMC Software delivers the solutions IT needs to increase business value through better management of technology and IT processes. Our industry-leading Business Service Management solutions help you reduce cost, lower risk of business disruption, and benefit from an IT infrastructure built to support business growth and flexibility. Only BMC provides best-practice IT processes, automated technology management, and award-winning BMC Atrium technologies that offer a shared view into how IT services support business priorities. Known for enterprise solutions that span mainframe, distributed systems, and end-user devices, BMC also delivers solutions that address the unique challenges of the midsized business. Founded in 1980, BMC has offices worldwide and fiscal 2007 revenues of $1.58 billion. Activate your business with the power of IT. www.bmc.com.

About the Author
Bronna Shapiro, director of Solutions Marketing for Identity Management and Services for BMC Software, manages a team that is responsible for marketing and strategy for identity management solutions. In addition, she also leads a team responsible for marketing for consulting services, educational services, and customer support. Her prior industry experience is in marketing leadership for distributed systems solutions and in product management of key mainframe products. She has also been an instructor for courses on IMS and DB2.

To learn more about how BMC can help activate your business, visit www.bmc.com or call (800) 841-2031. BMC, BMC Software, and the BMC Software logo are the exclusive properties of BMC Software, Inc., are registered with the U.S. Patent and Trademark Office, and may be registered or pending registration in other countries. All other BMC trademarks, service marks, and logos may be registered or pending registration in the U.S. or in other countries. DB2 is a registered trademark of International Business Machines Corporation. All other trademarks or registered trademarks are the property of their respective owners. © 2007 BMC Software, Inc. All rights reserved.