Server Platform
Selection and Positioning

Version 2.1
February, 2009

This whitepaper started with Mark Dixon taking the web lecture from the Server Computing Community of Practice and putting words around it. The team of Rick Lebsack, Darryl Van Nort and Gord Palin built on Mark's work to develop this whitepaper based on the continuing work of the Fit for Purpose Platform Selection team.

The original web lecture can be accessed at http://w3.tap.ibm.com/w3ki/display/SERVERCOP/Lecture+Series

The Platform Selection work is the result of the insight and contribution of many people

- Sally J Eystad – Atlanta
- Satya Sharma – Austin
- Jim Mitchell – Austin
- Joel Tendler – Austin
- William Starke – Austin
- Ray Williams - Calgary
- Kevin D Dixon – Charlotte
- Darryl Van Nort – Chicago
- Susan Grabinski - Chicago
- Todd Engen – Columbus
- Susan Schreitmueller – Dallas
- Rick Lebsack – Denver
- Gord Palin – Markham
- John Schlosser – Milwaukee
- Gary Crupi – Milwaukee
- John Banchy – Minneapolis
- Randy Holmes – Minneapolis
- Mickey Sparks – Minneapolis
- Scott Lundell – Omaha
- Brian Gagnon – Pittsburgh
- Chris Meurer - Pittsburgh
- Joseph Temple – Poughkeepsie
- Kathy Jackson - Poughkeepsie
- Colleen Ashe - Poughkeepsie
- Pak-kin Mak – Poughkeepsie
- Frank DeGilio – Poughkeepsie
- Hilon Potter – Poughkeepsie
- William Clarke – Poughkeepsie
- Steve Kunkel – Rochester
- Craig Wilcox - Rochester
- Mark Dixon - Sacramento
- Dale Martin - San Francisco
- Jeff Josten - Santa Teresa
- Danny Williams – UK
- Wakaki Thompson – Wayne
- James Huston - White Plains

Trademarks

The following terms are registered trademarks of International Business Machines Corporation in the United States and/or other countries: AIX, AS/400, DB2, IBM, Micro Channel, MQSeries, Netfinity, NUMA-Q, OS/390, OS/400, Parallel Sysplex, PartnerLink, POWERparallel, RS/6000, S/390, Scalable POWERparallel Systems, Sequent, SP2, System/390, ThinkPad, WebSphere.

The following terms are trademarks of International Business Machines Corporation in the United States and/or other countries: DB2 Universal Database, DEEP BLUE, e-business (logo), ~, GigaProcessor, HACMP/6000, Intelligent Miner, iSeries, Network Station, NUMACenter, POWER2 Architecture, PowerPC 604,pSeries, Sequent (logo), SmoothStart, SP, xSeries, zSeries. A full list of U.S. trademarks owned by IBM may be found at http://iplswww.nas.ibm.com/wpts/trademarks/trademar.htm. NetView, Tivoli and TME are registered trademarks and TME Enterprise is a trademark of Tivoli Systems, Inc. in the United States and/or other countries.
Oracle, MetaLink are registered trademarks of Oracle Corporation in the USA and/or other countries.
Microsoft, Windows, Windows NT and the Windows logo are registered trademarks of Microsoft Corporation in the United States and/or other countries.
UNIX is a registered trademark in the United States and other countries licensed exclusively through The Open Group.
LINUX is a registered trademark of Linus Torvalds.
Intel and Pentium are registered trademarks and MMX, Pentium II Xeon and Pentium III Xeon are trademarks of Intel Corporation in the United States and/or other countries.
Java and all Java-based trademarks and logos are trademarks of Sun Microsystems, Inc. in the United States and/or other countries.
Other company, product and service names may be trademarks or service marks of others.

Platform Selection

There is a difference between positioning and selection. One of the first things to discuss is what we are referring to when describing a platform. It is entirely possible that a platform could include more than just a combination of hardware, operating system and virtualization.

For example each of these listed environments could be considered a target of platform selection either individually or combined together:

- System Infrastructure -- Hardware systems, Operating Systems and virtualization
- Application Support – the middleware and system support environment (CICS, WebSphere, WebLogic, etc)
- Data Platform – where the data will or must be hosted (DB2, UDB, Oracle, etc)

For the purposes of this discussion:

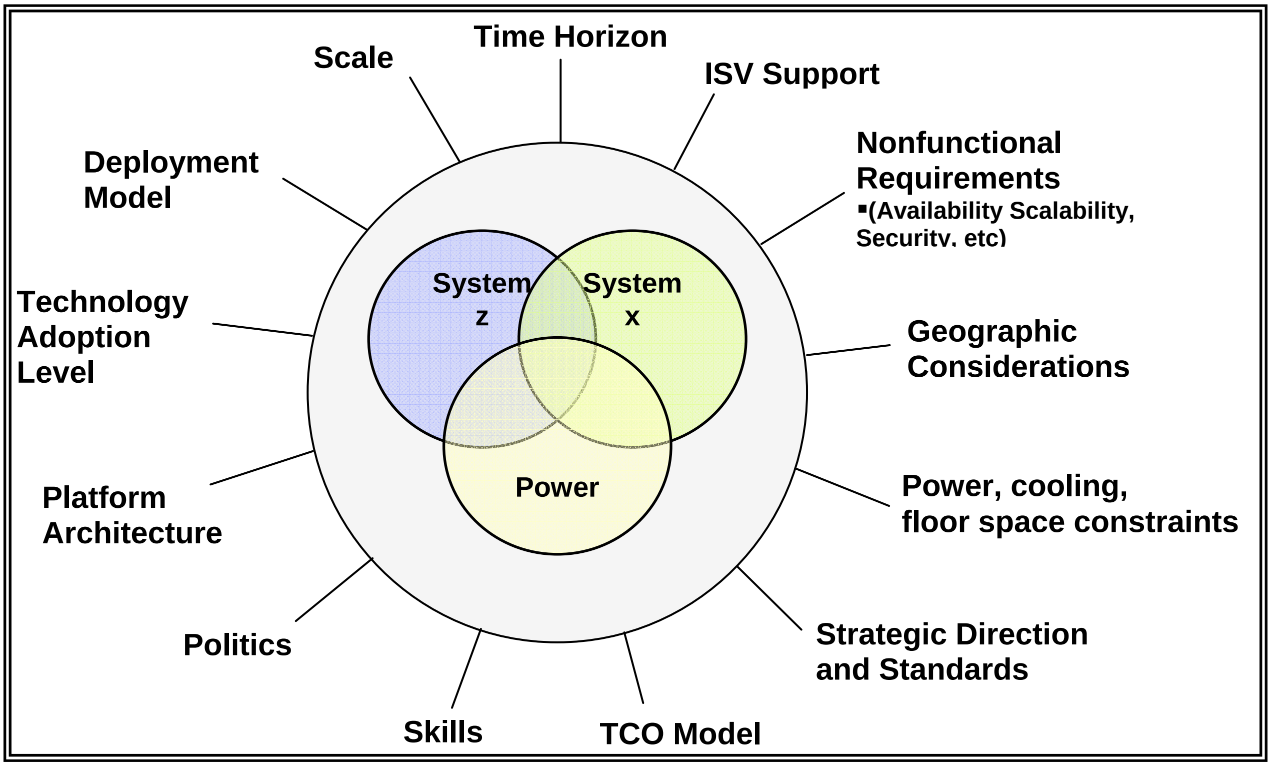
Positioning
Positioning is the fact-based discussion of different platforms and architectures including features and functions in an attempt to understand the tradeoffs that apply for specific applications.

Selection
Selection is the process of choosing a platform to host an application. The selection will either be between architectures (i.e., System p or System z or System x) or between vendors within a given architecture (i.e., Sun/Solaris vs. IBM/AIX).

If the OS and middleware are hardware agnostic, then the scope of the analysis can become much larger.

The final selection process is more than “speeds and feeds” and incorporates an evaluation model which includes many other factors where a selection is the end result.

Platform selection is not a simple matter. Many issues, both technical and non-technical, can affect the selection of a hardware platform.



Issues Affecting the Platform Selection Process

Every client will feel the influence of these items differently depending on their "local factors". Local factors are what personalize the selection process to what matters and is important to them. Local factors may include:

- Skills
- Technology adoption levels
- Platform management practices
- Application specific platform requirements
- Number of servers / scope of implementation

Technology is rapidly changing, and one of the keys to lowering IT costs is to take advantage of functions offered by newer technologies (may be more than just platforms). Realization of a platform's ultimate capabilities is the intersection of the inherent platform capabilities and local/unique client factors to deliver the required services. Therefore it is generally best to develop qualities of service and cost metrics for each platform under consideration or in use.

Within the realm of the platform itself, it's best to develop and adopt best practices to exploit the platform capabilities to move toward platform "best of breed" operations.

The following table represents some of the newer capabilities offered by key IBM hardware platforms that help to drive computing costs down.

System z	Power Systems	System x and BladeCenter
▪ Parallel Sysplex ▪ DB2 data sharing ▪ zIIP, zAAP, IFL, ICF ▪ Execute in place filesystems ▪ Shared z/VM binaries ▪ z/VM shared memory ▪ Workload license charges ▪ 4.4 GHz Processors (z10)	▪ Virtualized I/O ▪ Multiple shared processor pools ▪ AIX Workload Partitions ▪ 5.0 GHz Processors ▪ IBM i - Integration ▪ Virtualized IBM i storage ▪ DB2 web query for IBM i ▪ IBM i – run mixed workload	▪ X4 Architecture ▪ x3950 scalable servers ▪ iDataPlex ▪ Server/desktop virtualization ▪ IBM BladeCenter integration ▪ Cell and power blades ▪ IBM Director

Strategic and Tactical Selection Decisions

Whenever an organization makes a platform selection decision, there is always tension between the business/technical strategy and the legacy environment. The following table is an example of some of the contrasting elements of this tension:

Strategy	**Legacy**
Long-term advantage	Skills
Competitiveness	Capital
Problem Solution	Transformation Costs
Issue Resolution	Culture
Cost Savings	Momentum
Optimization	Risk

Platform selection activities often mix strategic and tactical viewpoints and elements. From a strategic standpoint, there is no single ideal platform for all workloads. Selecting a platform from a strategic standpoint presents many challenges, such as:

- Interoperability – vendor certification of OS and or release levels
- Mergers & acquisitions can introduce nonstrategic solutions
- Strict adherence to the strategic view can run afoul of legacy
- Vendors will attempt to make their platform cover the waterfront
- Technology changes will shift what is the best-fit strategic platform

Tactical decisions have challenges, too. Some key attributes of tactical selection are:

- Largely based upon convention, organizational momentum, and previous decisions
- Can allow for sub-optimal decisions leading to increased costs and “server sprawl”
- Date-challenged projects allow insufficient time to choose the *right* solution

One interesting result of strategic initiatives is that once they are implemented successfully they become part of the legacy. For example: as new skills required for a strategic initiative are learned and assimilated throughout the enterprise they become business-as-usual and as such are part of the "new legacy." As a result of many factors, an organization's IT decisions will tend to be some combination of both the tactical and the strategic.

Deployment Models

There are three basic deployment models involved in platform selection. We will refer to these as: Centralized, Virtualized Distributed, and Dedicated. (See Table below).

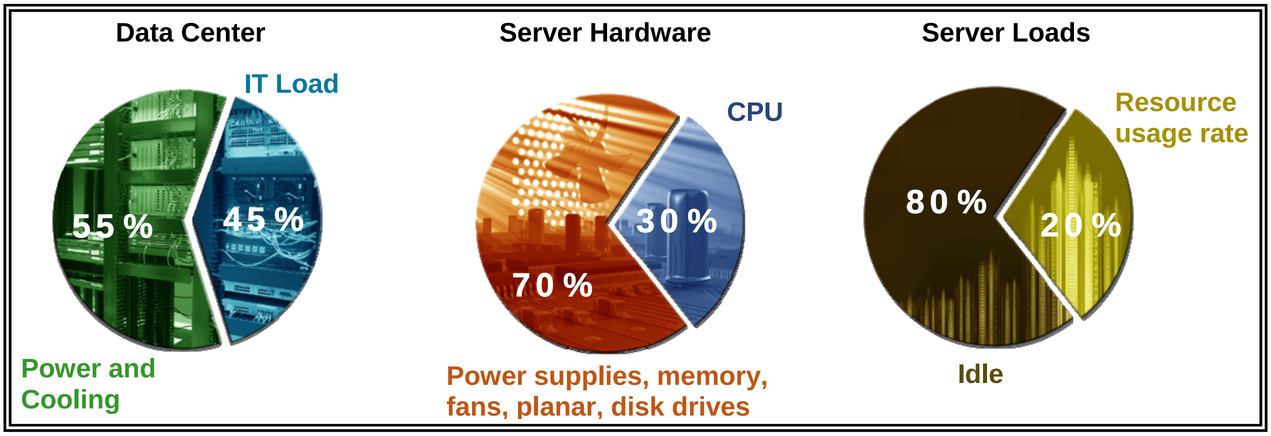
In terms of capacity and performance, a centralized deployment will eliminate network path lengths and the resulting latency between software components. By sharing all CPU, Memory and I/O resources, it allows for transaction prioritization and extremely fluid movement of resources between various components. These characteristics result in lower total resource requirements, higher average utilization and easier performance tuning, capacity planning and disaster recovery capabilities. This model may also allow streamlining of logging, security and compliance issues.

Model Name	Software Layers	Resource Sharing	Workload Mgmt
Centralized	All SW layers run in the same operating system instance	All resources (CPU Memory, I/O) are shared at a very granular level	Fine-grained workload management is possible
Virtualized Distributed	SW layers run in logical partitions	Resources may be shared, but at a more coarse-grained level than in the centralized option	Workload management is more coarse grained than is possible with a centralized deployment
Dedicated	SW layers run in separate physical systems	Resources cannot be shared	Workload management is accomplished by over-configuration

Reliability of a system is inversely related to the number of components in that system. A centralized system tends to have the fewest components of the three deployment models, leading to inherent higher availability. A distributed deployment depends on many different components, such as servers, network and SAN switches and storage devices. Clustering servers and mirroring storage (by subsystem or OS function) can improve a distributed application's availability, but introduces added complexity and costs. The inherent complexity of a distributed deployment can also be magnified by the number of additional staff and increase in skills required. In addition, the larger number of components and their interconnections in distributed environments increases the cost of outage management, both in terms of human capital and time to recover.

Data integrity is harder to ensure in a distributed deployment model. Operational data stores must be in sync and transaction commitment must be made on that in-sync data. The more commit phases, the more complexity and thus a greater integrity exposure is created. Latency becomes an issue as resources are held or locked to allow for the commitment to occur, and can introduce significant issues, especially in High Performance, High Availability and Disaster Recovery realms. The increased latency increases the data synchronization issues during any recovery scenario.

The choice of deployment model can affect the environmental costs of a particular platform choice. Maximum power efficiency is achieved when all the resources are driven to the highest levels of utilization.



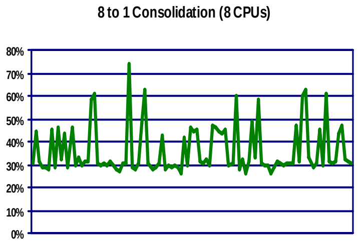
Power and Cooling Efficiency Value Chain for Servers

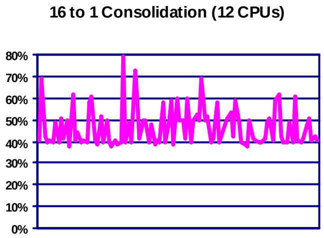
Of the three areas above, IT has direct control over how the servers are loaded and their accompanying utilization. IT can purchase equipment with energy efficient components and capabilities to maximize both utilization and how these resources are deployed to deliver service. A centralized deployment minimizes infrastructure complexity and maximizes utilization. Virtualization capabilities allow for resource sharing between logically distributed instances, helping to drive up utilization and reduce overall power and cooling requirements.

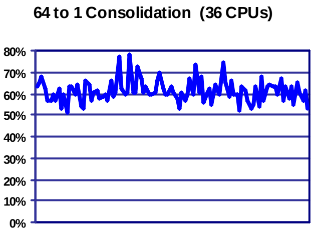
From an efficiency standpoint, the higher number of applications one is able to virtualize in a single system, the more the average workload will approach the peak workload, indicating a higher level of utilization.

The following three charts are illustration of the effect of combining multiple workloads while maintaining service level (response time).

- Single workload model assumptions:
    - Average Utilization: 20.7%, Peak: 79%
    - Random Arrival Rate
- As more copies of this workload are added, average utilization approaches peak
    - 8:1 39% Average, peak 76%
    - 16:1 48% Average, peak 78%
    - 64:1 61% Average, peak 78%


8 to 1 Consolidation (8 CPUs)


16 to 1 Consolidation (12 CPUs)


64 to 1 Consolidation (36 CPUs)

Note that as workload is added, the need for additional CPUs grows at a smaller rate.

When implementing the Virtualized Distributed model, it is generally best to use fewer larger servers than a greater number of smaller servers for several reasons. From the hardware standpoint, larger servers show the following benefits:

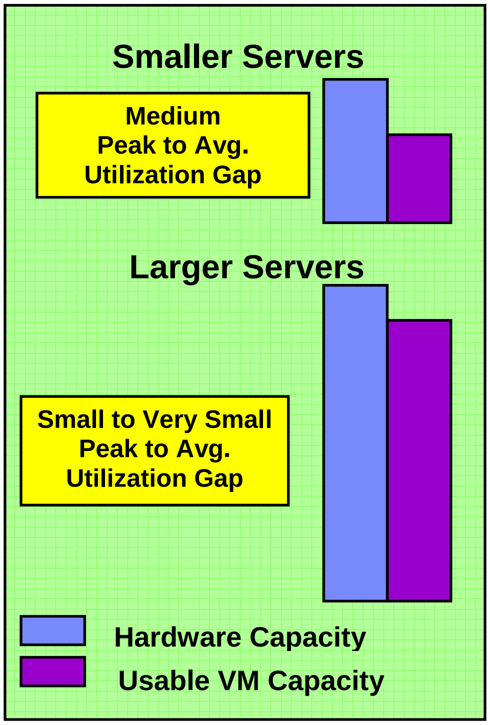
- Higher utilization due to shared “headroom”
- Each application shares increase in internal bandwidth for better performance
- Fewer disk drives, adapters and ports needed
- Able to share memory more effectively
- More fault tolerant features

From the administration standpoint, larger servers show the following benefits:

- Fewer servers to order, install, maintain, retire
- Fewer Hypervisor instances to manage
- Fewer firmware patches to apply
- Less time spent cabling

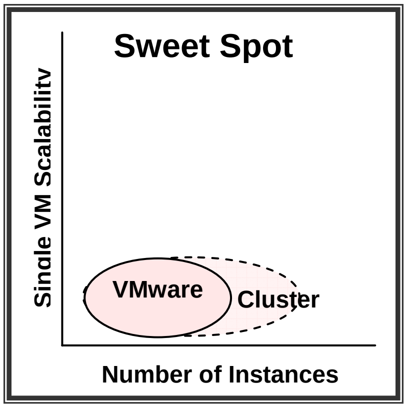
From the Data Center perspective, larger servers allow for:

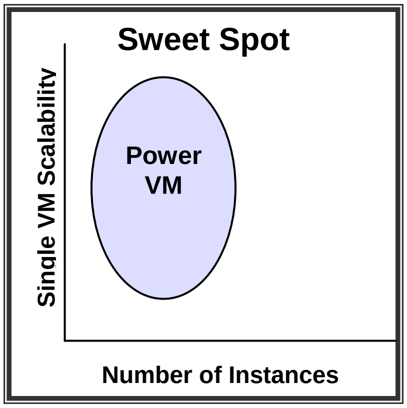
- Better power utilization
- Reduced floor space

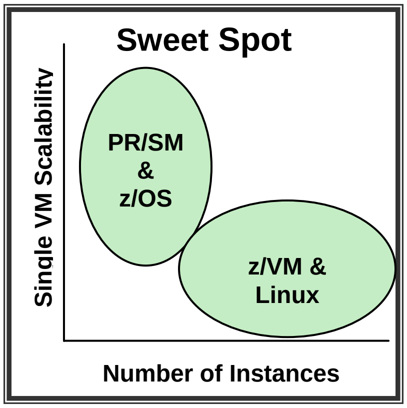


Virtualization “Sweet Spots”

Every hardware platform has its “sweet spots” in terms of virtualization, expressed as functions of the scalability of a single Virtual Machine and the number of instances in question.







Some of the issues and attributes of single VM scalability and number of instances thereof include:

Single VM Scalability

- Efficient Hypervisor scheduling model
- Run time efficiency – assists for locks, spin loops, etc
- Number of supported virtual cores
- I/O virtualization paradigm
- Quantity of adapters/memory
- Support for dedicated memory
- Tight integration of entire virtualization stack
- Isolation of guests/versions
- OS Scalability

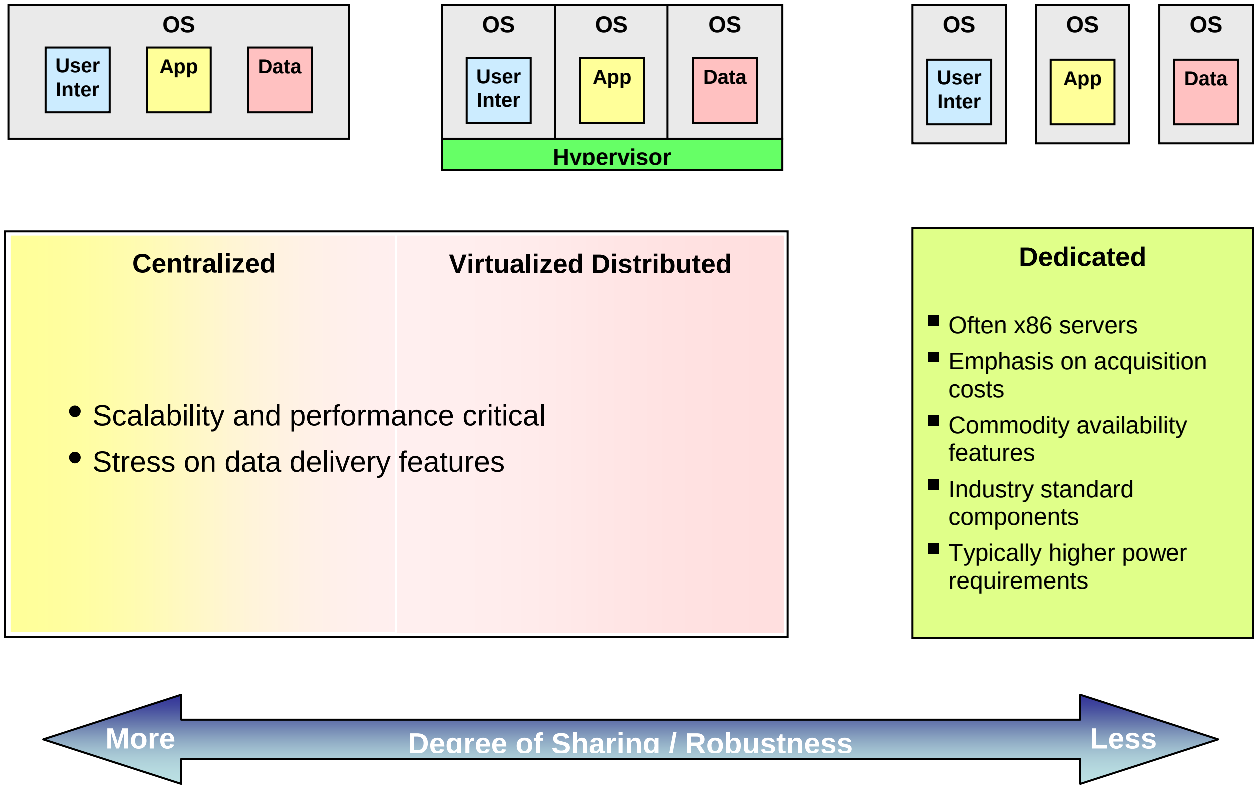
Number of Instances

- Efficient Hypervisor scheduling model
- Ease of creation, cloning, migration, deletion and disaster recovery
- Virtualized memory / memory de-duplication
- VM context switch efficiency
- Cluster level tooling
- Software licensing flexibility
- Isolation of guests/versions
- Uniformity of virtual machines
- Light weight performance monitoring

A virtualization model does not dictate any specific deployment model. Depending on the customer environment and workload, it is just as likely for a large p595 to be deployed Centralized as Virtually Distributed. For example, from the “sweet spot” diagrams and the attributes described above, the Dedicated deployment model could be an IBM BladeCenter or an iDataPlex machine. Both have small, discrete units of resource that cannot be shared across the physical machine/blade boundaries.

Platform Architectures – Scalability and Throughput

Deployment models are based upon and exploit the characteristics of a particular platform's architecture. Looking across the deployment model spectrum, it is apparent both the Centralized and Virtually Distributed models put a relatively high degree of stress on the underlying hardware so a corresponding degree of robustness is required to share resources.

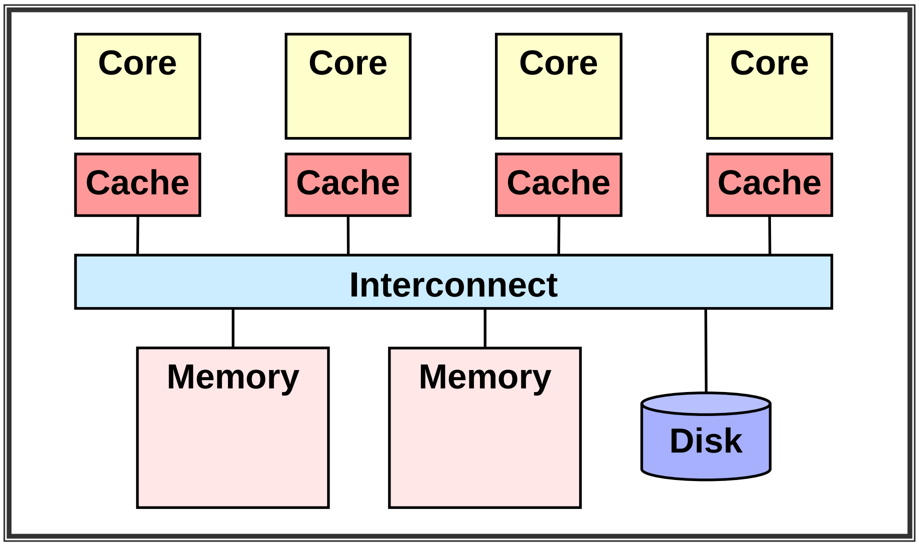


- Scalability and performance critical
- Stress on data delivery features

- Often x86 servers
- Emphasis on acquisition costs
- Commodity availability features
- Industry standard components
- Typically higher power requirements

How these stresses are handled is reflected in the chip and machine architectures. Large Symmetric Multi-Processing (SMP) machines make it much easier to share resources and provide more inherent reliability.

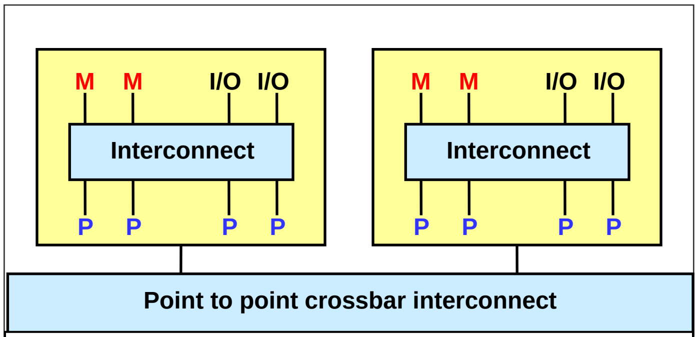
There are limitations and constraints in all chip/machine architectures, particularly in terms of interconnections between CPUs and memory. Raw CPU cycles (clock speed) are but one factor of overall performance, there are many other factors to consider when evaluating a machine's throughput such as I/O offload processing, hardware-assisted encryption/compression and other performance-enhancing technologies including symmetric multi-threading.



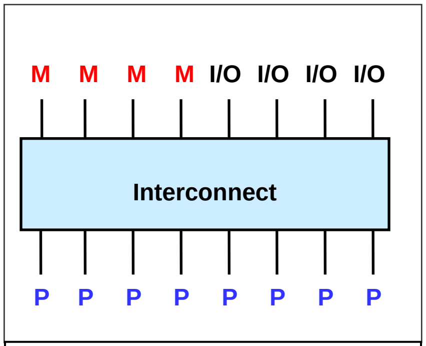
Generic Computer Resource Interconnect Model

The key issue is that memory operates slowly compared to CPUs. As particular hardware designs add CPUs to gain scalability, issues such as locking and serialization become more important in managing the flow of information inside the machine. At some point, adding the additional CPUs can actually reduce performance – this called the n-way SMP effect. The most challenging factor in gaining performance is keeping the CPUs supplied with data and instructions, so cache and interconnections become key factors.

The memory model is also an important factor, as a Non-Uniform Memory Access (NUMA) model will introduce additional latency into the equation as information has to navigate from one module to another. This is referred to as ‘close’ and ‘distant’ memory. A ‘flat’ or ‘uniform’ memory model normalizes the path length that information has to travel for consistent performance.



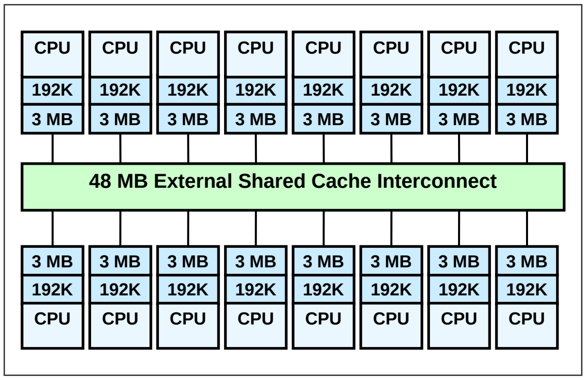
Non-Uniform Memory Access Model



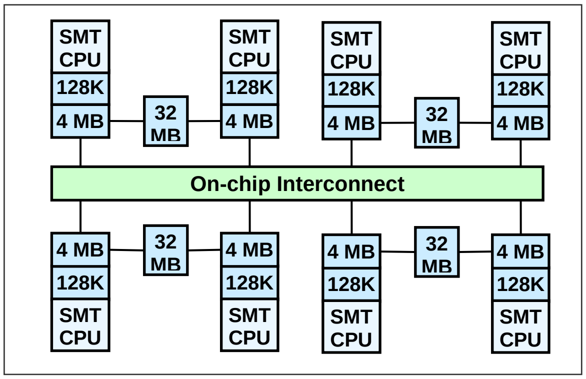
Flat Memory Model

NUMA architecture is useful for workloads that can be physically partitioned. It tends to support a larger quantity of components, but memory latencies limit the overall machine scalability. NUMA architecture is used in IBM’s high-end System x machines and by HP and Sun across their product lines. The coupling facility in a Parallel Sysplex is a specialized NUMA implementation when clustering System z machines.

In contrast, a flat memory model provides consistent access to resources and is ideal for shared workloads. A flat/uniform memory model generally uses fewer components which improves reliability and provides for lower average transaction latency. Both IBM System z and POWER Systems implement a flat memory model.



System z10 CPU Design Schematic



POWER 6 CPU Design Schematic

The two above charts illustrate the current flat memory model of the System z10 and Power 6 architectures. While they are both flat models they do have differences based on the design requirements for each. One of the key requirements was support for their respective legacy workloads.

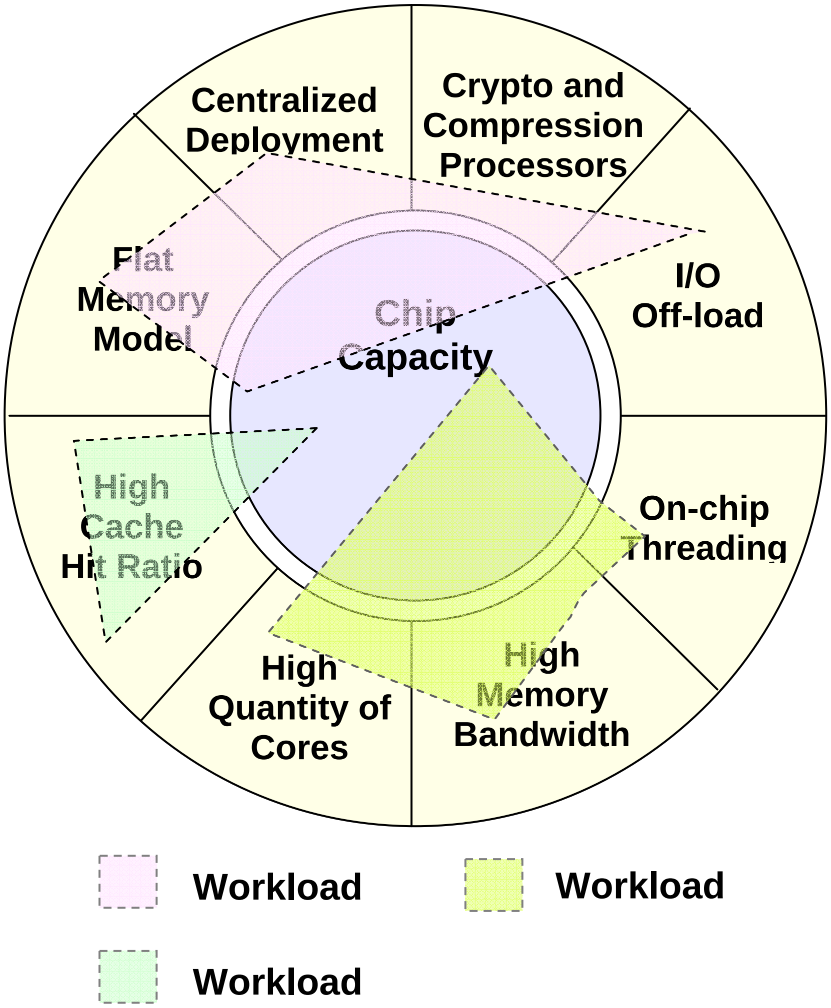
The System z10 was designed to handle varying sizes of workloads and arrival rates within an LPAR. It was expected there would be frequent context switching and significant sharing of data between threads. A key design objective for System z running zOS was that performance be consistent at high utilization and that all workloads see similar performance gains.

Conversely, Power 6 was designed with a wide variety of form factors in mind – from blades to high end SMPs. The design point for the workload was different than System z and it was crucial that it span commercial and HPC workloads, support highly threaded/CPU intensive work and applications requiring large memory bandwidth and/or size.

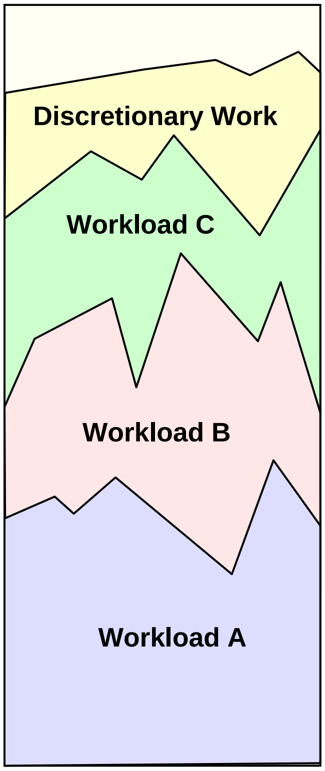
Just as certain workloads tend to gravitate to particular deployment models, the overall machine capacity to do work varies with the type of workload being run. All architectures have characteristics that workloads can leverage. Effective capacity of a given architecture can be thought of more as a ‘cloud’ as opposed to a single data point. Given the differences in machine architectures, there is no way to effectively do a one-to-one mapping of the effectiveness of one machine versus another. This is illustrated in following chart where three different workloads each exploit different hardware characteristics.

Even within a given workload, capacity can vary. For example, database I/O can vary from a transaction mix of light read to heavy update and from random to sequential. Memory bandwidth and reference patterns can range from light to heavy and there are differences in batch workloads as compared to interactive workloads.

Given the workload variability discussed above, the effort to analyze workload factors should be undertaken in any platform selection decision. Note that servers with a broad set of capabilities can adjust to varying workloads better than mono-architecture servers which may run a particular benchmark well.



In addition to cache, interconnect and memory architectures, any server's throughput is dependent upon transaction sizes and arrival rates of those transactions. Throughput is also affected by schedulers – PR/SM, z/VM, and PowerVM hipervisors schedule ready-to-run virtual processors independently with consideration towards reuse and affinity, hence System z (Centralized deployment model) and System p (Virtualized deployment model) can lower workload switching costs.



Driving consistently high utilization rates requires the presence of discretionary workloads to fill in variations in the total aggregate workload. Centralized deployment models can identify discretionary work at very granular levels, while a virtualized distributed deployment model requires a mechanism to identify discretionary work (i.e. a separate variable LPAR for discretionary work) and therefore is usually at a much coarser granularity.

Non-Functional Requirements

Every application has both functional and non-functional requirements. Functional requirements answer the question "What does this application do?" Non-functional requirements answer the question "How well does this application do what it does?" Non-functional requirements answer such questions as:

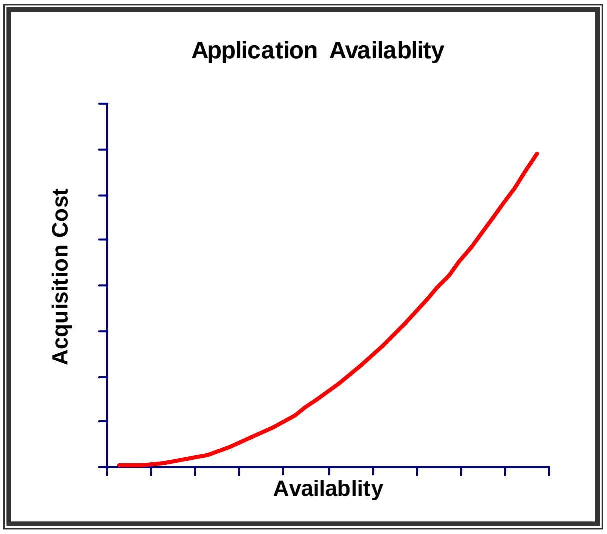
- What are my availability requirements?
- How many transactions a minute do I need to sustain?
- What are my security requirements?
- How easy is it to provision and support?
- What are my disaster recovery requirements?
- Do platform features accommodate rapid growth?

Business and IT objectives will influence both, but platform selection is generally concerned with the non-functional requirements. A few key non functional requirements will be expanded on.

Non-Functional Requirements: Availability

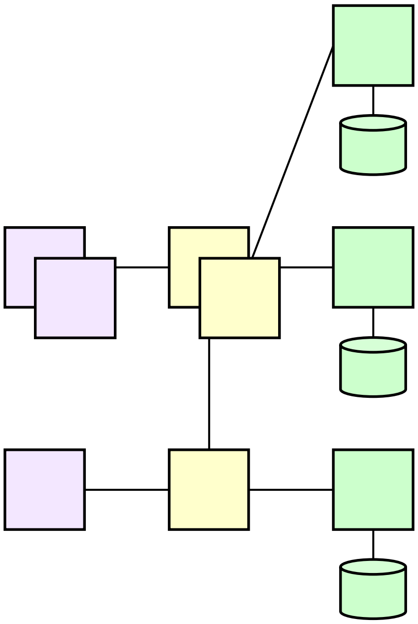
Application availability can be generally correlated to platform acquisition costs. At the low end of the curve, there is extreme focus on acquisition costs, extensive use of commodity RAS solutions and limited, if any, integration of the hardware components and software stack. Most, if not all, of the integration work is done by the client.

In the middle of the curve, there is increased integration across the hardware components and software stack. There is also more usage of non-commodity RAS features on the hardware platform that allows for a wider range of both planned and unplanned outage management capabilities.



At the high end of the curve, there is a holistic approach to availability, characterized by significant automation to reduce human errors and advanced clustering for near-continuous and continuous operations.

Non-Functional Requirements: Disaster Recovery



Disaster recovery capabilities can also follow a similar curve. It is extremely difficult to provide disaster recovery services for a large number of disparate servers. Even Tier 1 servers can often be dependent upon lower tier servers which may not be included in a disaster recovery plan.

Despite good planning and testing, the amount of effort to resynchronize all of the various systems after a disaster will be overwhelming and will stress even the most capable of organizations.

A DR plan to be useful must be tested regularly. As the number of interdependent components increase this becomes harder to accomplish in both testing and if a disaster occurs.

Non-Functional Requirements: Security

Security is a business requirement which is generally expressed as a policy or collection of policies. The IT organization defines appropriate technical controls to meet the business policies. Complexity is the bane of security, yet security is inherently complex.

No system is inherently secure, but any individual server can be made secure with enough time and resources (and wire cutters). As diversity in a server population grows, it becomes increasing difficult to secure a large number of servers, especially when the environment is characterized by employee turnover, rapid application changes, and growth.

Security at the system level is about having the appropriate controls to implement policy, provide audit ability, and do so with a minimum of additional complexity. A security system should be flexible and resilient, not fragile.

Some desirable attributes of a comprehensive security solution include:

- Availability of technical controls to meet the policies
    - Easiest if the controls are and integral part of the system
    - Add-on tools and products increase complexity and personnel workload

- Ease of configuration
    - Control is more likely to be implemented if the solution is easy to set up
    - Reduces the potential for errors (leading to accuracy)

- Consistency
    - Makes traceability and auditing easier
    - May reduce the staffing requirements to support the environment

        - May reduce the possibility of misinterpreting control implementation (relative to policy) across different platform environments

- Auditing
    - Gather the right level of information quickly without undue human intervention
    - Minimal amount of information "scrubbing" to achieve audit reporting

- Separation of duties
    - Limits the control scope of one individual
    - Provides auditable checks / balances on implementations

Non-Functional Requirements: Integration and Ease-of-Use

Administrative costs are frequently one of the largest cost elements in platform deployment. Designing ease-of-use into a system platform can provide significant TCO benefits. By integrating and testing the hardware components and software stack of a particular platform, administrative costs can be substantially reduced and application availability can be significantly increased. The complexity of this integration increases as more components must be included.

The IBM System z and POWER Systems (running IBM i) platforms are focused on providing ease-of-use and availability through integration and testing. POWER Systems running IBM i are often found to have the lowest total cost of ownership by independent consultants. Some examples of design elements that allow the POWER hardware and IBM i software to achieve such an enviable record are:

- Software Integration
    - OS, standard tools, and database are integrated and tested together
    - Reduces administrative effort
- Automatic Storage Management
    - Disk storage usage is automatically balanced based upon usage and capacity
    - No separate storage administration team required
- Object Based Architecture
    - All files are objects resulting in no known viruses
    - Reduces administrative effort and downtime
- Mixed Workload Capabilities
    - Can run multiple workload types on the same OS instance
    - Easily managed by use of subsystems
    - Significantly reduces number of servers to manage
- Integrated Security
    - Leverages the Object Architecture
    - Implements security requirements by object type
    - Easy to add or delete special authority
- Menu Driven Control
    - Complex tasks can be accomplished through simple menu interfaces
    - Significantly reduces human errors and staff training

Geographic Considerations

As enterprises grow or as requirements dictate a multisite deployment for applications geographic considerations will enter into the picture for platform selection. Some of these considerations are related to traditional non-functional requirements, but there are a number that extend into new areas. Each of the items listed below expand the complexity of managing multiple sites. The coordination efforts are significant between geographically separate sites and add requirements for large amounts of bandwidth and inter-application planning. Following is a list of some of these considerations:

- **Network**
    - Latency
    - Bandwidth
    - Impact to application of network outage
    - Dependencies on other applications

- **Economies of Scale**
    - Data center space
    - Multiple power and network feeds
    - Backup generators
    - Skills
    - Backup infrastructure
    - Storage infrastructure

- **Time Zones and Languages**

- **Security**
    - Network exposures
    - Distributed data
    - Physical security / theft

- **Business Continuity**
    - Physical separation of data centers
    - Recovery time

- **Regulatory / Geopolitical**
    - Export regulations
    - Local regulations
    - Political stability
    - Customers

- **Data Consistency & Management**

ISV and Middleware Support

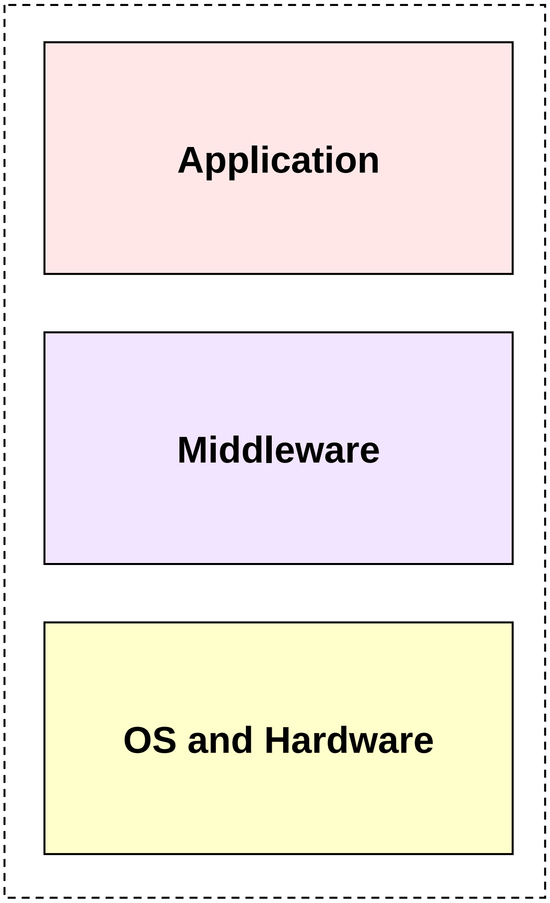
As discussed above, platform selection is generally focused on non-functional requirements. Functional requirements, however, will be a “filter” or constraint which will narrow the choice of platforms under consideration. Functional requirements are often dictated by the choice of a specific ISV software application or middleware on which the application is built.

In the case of a hard functional requirement for a specific ISV application, this becomes a straightforward discussion. For any given deployment, you would eliminate platforms that don’t support the ISV application. If there is a strategic requirement to target specific platforms not supported by the ISV or the non-functional requirements can’t be supported by the platforms supported, then you would have to choose a different ISV stack.

In the case of in house developed applications built on middleware that is available on a number of platforms, there are some additional considerations:

- Select platforms based upon application requirements not on middleware brand
    - Non-functional requirements can vary across applications using the same middleware

- Consider platform agnostic middleware
    - Non-functional requirements, such as transaction volume, may change over time
    - Agnostic middleware offers better flexibility



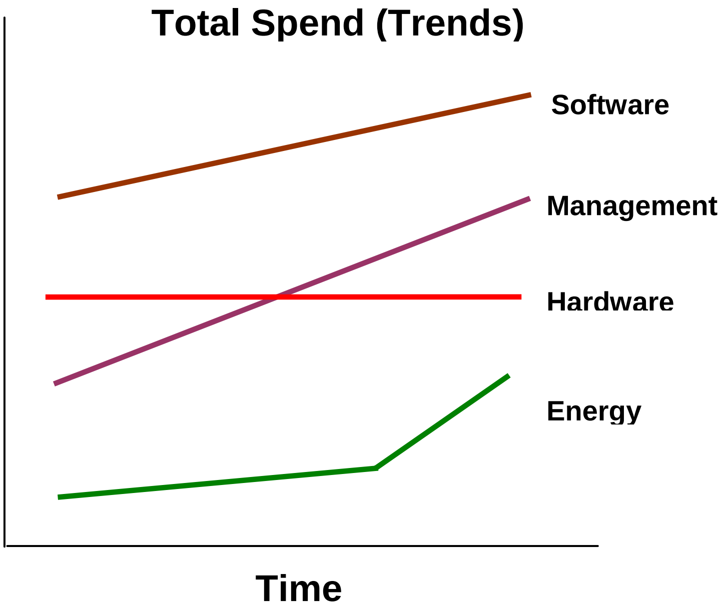
Cost Trends: Developing an Accurate Cost Model

IT Cost Trends

The cost “mix” in IT has been changing rapidly over the last years. In fact, over time, the management and administration cost has switched places with the hardware/software cost as the single largest IT expenditure. Unfortunately, the energy costs have ramped up rapidly in the last few years and show no sign of slowing down.

As the graph to the right shows, management costs are significant and growing rapidly while energy costs are rising at an ever increasing rate. Hardware spend has been flat, despite the significant growth in unit volume. Software costs, not shown, have grown linearly.

The unmanaged growth in unit volumes has many IT organizations "hitting the wall" in space or power capacity as "server sprawl" has overwhelmed them.



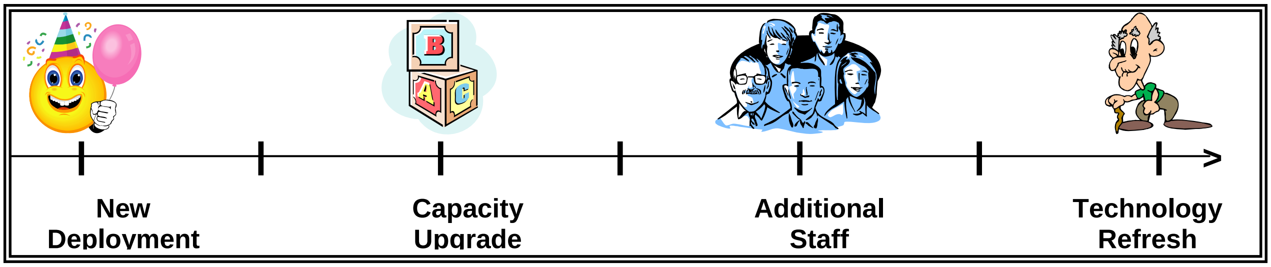
Unfortunately, they are finding out the hard way that new datacenters take time and are very expensive to build. Often many municipalities prevent the building due to the lack of available power.

There are two major points to note relative to this changing cost landscape:

- A cost model that focuses primarily on hardware spend (TCA) will ignore most of the significant and growing costs in the datacenter.
- IT Accounting systems must be able to accommodate the changes that are driving changes in infrastructure deployment (e.g. data center consolidation, virtualization, and cloud computing).

Planning Horizon

The planning horizon is the most critical part of the cost case as it is a significant factor in both the capacity required and the corresponding growth rates from which the total capacity is derived. It also guides the cost accounting for any chargeback systems. A relevant chargeback system for today's changing world must account for how overages or shortfalls of cost recovery are accounted for and it must address or eliminate the "penalty" that can occur for incremental growth that requires acquisition of new capacity (e.g. new CPUs or storage system) that is larger than the requirement that drove the acquisition in the first place.



A good Planning Horizon includes:

- A well-defined starting point
- A well-defined end point
- The growth required within the initial platform buildout
- The granularity and frequency of upgrades
- Any residual values or scrap costs along the way
- Accounting issues with serial number changes

Developing a Cost/Value Model

Cost and Evaluation models are highly interrelated. The cost model is where the items that were developed in the evaluation model get quantified. Note that the choice of evaluation and related cost elements often dictate which platforms are considered the "lowest" cost, therefore it is imperative that accurate models be developed.

In fact, costs go way beyond the acquisition cost of hardware and software as well as the projected life-cycle costs of maintenance for both. In a comprehensive cost model, often the indirect costs are the inverse of the hardware, software and maintenance direct costs. To accurately assess the value of any solution organizations should take the time to develop accurate metrics to quantify the indirect costs (e.g. outages, security breaches, etc.).

Sample Direct Costs

- Hardware: Prod and non-prod
- Hardware maintenance
- Internal FTEs and consultants
- Software
- Software maintenance
- Power and cooling
- Floor Space
- Network and FC ports
- Residual value
- Disaster recovery
- High Availability
- Asset management

Recovering Costs over Time: Chargeback

A chargeback system should not be the basis for a Platform Selection or Cost/Value analysis. The fundamental purpose of a chargeback system is for the IT organization to recover their costs.

To implement the Virtualized Distributed deployment model is it critical that the existing chargeback system be able to deal with a shared infrastructure. This normally means that infrastructure components should be owned by the IT department and the chargeback system should be implemented to encourage behavioral changes in terms of managing and encouraging demand for IT services. It is likely that multiple charging mechanisms may be required to handle different IT consumers.

Sample Indirect Costs

- Cost of an outage
- Security breach
- Cluster complexity
- Business flexibility
- Risk
- Time to market

A relevant chargeback system links the IT organization to the business accounting system. A relevant chargeback system:

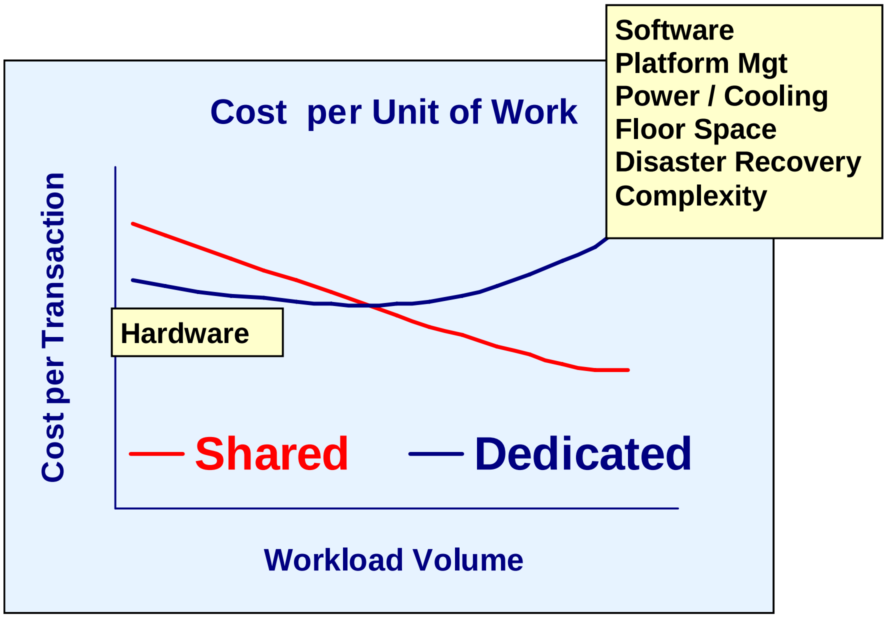
- Should not be based on efficiency of resource utilization, just on the fact that the resource was utilized
- May not include all relevant cost factors, or may not apportion such factors equitably
- Does not reflect return on investment, or the value of what software system is being run
- May not reflect Service Level Agreement or resiliency attainment
- May not attribute all costs recovered to the platform(s) that accrues them, but rather to where they can be easily recovered

Cost per Unit of Work

Inevitably, the Cost per Unit of work comes up in any discussion of relevant chargeback systems.

In centralized environments, initial costs tend to be higher than in non-centralized environments, but the costs tend to decline rapidly on a per transaction basis as the volume grows.

In dedicated environments, initial costs tend to be low and the costs per transaction incurred early in the lifecycle decline as tooling and other common resource factors are shared across the environment. However, as workload continues to increase, complexity, administration, systems management and other factors rise, increasing costs.



Virtualized environments can affect the cross-over point for unit of work costs in positive or negative ways depending upon factors such as cloning, automation and leading-edge execution in systems and service management practices, etc.

As server environments grow, the dominant costs shift from hardware and software acquisition and maintenance thereof to other factors. As expected, the Deployment Model matters: Dedicated servers may offer the lowest cost for small environments

- Virtualized Distributed servers will most likely dominate the middle of the curve
- Centralized servers often become the platform of choice with scale

Clearly, deployment decisions based upon individual application or departmental costs will most likely be sub-optimal for the collective organization.

Summary and Top Recommendations

This paper shows that many factors influence platform selection – and that a simple matrix or method for making platform decisions does not exist. The characteristics of the organization can influence the platform selection process - platform choices have to balance strategy and legacy.

Each deployment model has its place – but given cost projections for energy and administration personnel, where scale permits, it is best to virtualize and/or centralize when possible. Larger SMP machines with flat memory models can be driven to higher levels of overall utilization and provide better availability. In addition, larger servers offer significant hardware, people, and environmental advantages for virtualization.

There is no single platform or middleware capacity metric. IBM high-end System z and Power System servers are built to handle the varying requirements found in real life applications and virtual environments. Workload characteristics should be matched to the sweet spots to optimize resource consumption. When sizing, make sure to adjust relative capacity with workload factors and beware of using tuned single workload benchmarks (which are often optimized for one characteristic of the hardware and are not representative of actual workload to be supported).

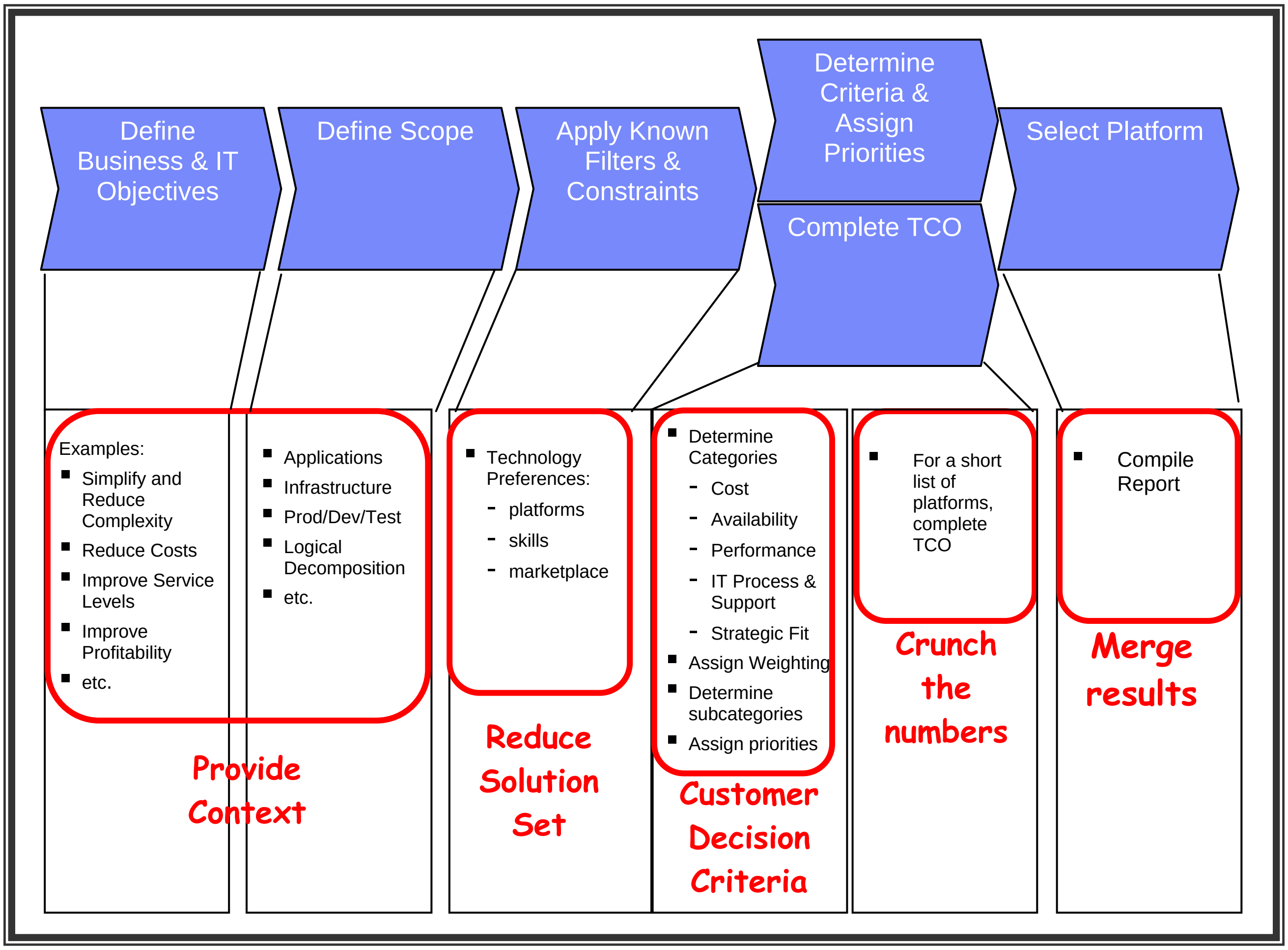
From a business standpoint, it is apparent that non-functional requirements are a significant element of platform selection. Ignore them and you will increase cost and complexity, especially in disaster recovery environments.

The impact of key cost elements are influenced by scope and time horizon. The most cost effective solution is built by exploiting new platform capabilities (both hardware and software) and taking an enterprise scope for platform selection. With an enterprise scope platforms will be selected with the objective of maximizing the optimization potential.

A cost and value model should be used for platform selection, not a dated and possibly irrelevant chargeback system.

Appendix 1: IBM Systems Architect Platform Selection Process

The IBM Systems Architect community has developed this document and an accompanying slide deck to assist clients in making appropriate and fully informed platform decisions. The community has also outlined a general process to follow when conducting a platform selection study.



We are interested in your feedback regarding this document, the slide deck (if seen) and the workshop process (if conducted). Send any feedback to Rick Lebsack ralebsa@us.ibm.com, Darryl Van Nort devanno@us.ibm.com or your local Systems Architect.

Please contact your IBM Systems Architect if you would like do discuss this process, provide feedback or schedule a workshop for your organization.