WHITEPAPER

Architecting High Performance eCommerce Platforms

CONTENTS

EXECUTIVE OVERVIEW

eCommerce applications have been a strategic asset for many companies since the late 1990's. During that period, the nature of customer interactions and their associated expectations have been rapidly evolving causing business executives and their architects to scramble to keep up with higher demands and a massive proliferation of users. Unfortunately, these requirements strain the infrastructures and applications that were designed and developed using traditional approaches established a decade ago.

eCommerce applications are developed for different purposes, each with their own distinctive business drivers. Business-to-consumer (B2C) sites are becoming strategic marketing channels for almost any product or service imaginable. At the same time, businesses depend on business-to-business (B2B) connections more than ever to streamline their supply chains and reduce operating costs. While they have different purposes, both B2C and B2B drive differentiation and are having a profound impact on company growth and profitability.

These business drivers translate to a wide range of IT challenges to which businesses must respond to by aligning IT with business goals in order to establish a fit-for-purpose approach to the IT strategy. The business drivers also initiate this process that in turn motivates the development of an IT architecture and infrastructure that meets the business objectives of the organization.

This paper outlines the primary steps to building successful eCommerce architectures which include (1) understanding the business drivers and IT challenges, (2) creating a demand model, and (3) translating and incorporating both into the eCommerce architecture. The first two sections will cover the business drivers and how these translate into IT challenges. We then outline a model for business demand. The remainder of the paper focuses on designing and creating the eCommerce architecture.

THE BUSINESS CHALLENGES

eCommerce applications are developed for two different purposes, each with their own distinctive business drivers: business-to-consumer (B2C) and business-to-business (B2B).

Business-to-Consumer (B2C)

This class of applications includes all retail and auction sites. Today, consumers have a wide range of site choices. Consequently, product and/or service pricing has become very transparent so that incentives are usually necessary to entice site traffic and to increase consumer loyalty. Consumers have high expectations for their overall experience and are attracted to rich content, including animated displays, videos, and more sophisticated browsing experiences.

In fact, the user experience is becoming as key a differentiator as the items or service for sale. As a result, sites are becoming strategic channels motivating companies to evolve their sites to include new features and themes that:

- Promote personalization for ease-of-use and comfort on the site
- Establish differentiated Qualities of Service for various classes of customers to reward loyalty, for example, gold, silver and bronze
- Provide a forum for community experience sharing
- Provide consistently fast response times allowing visitors to focus on products and brand messages and not the technology
- Entice customers with information that is dynamic and interactive, utilizing rich media delivered in real time
- Provide multi-channel content for laptops, desktops, phones and PDA's over wireless and broadband networks

Business-to-Business (B2B)

Today, businesses depend on B2B connections more than ever to streamline their supply chains, reduce operating costs, and scale back inventory levels. B2B interactions have grown tremendously in the last ten years and most of the existing systems to manage these connections are relatively new. However, many of today's systems have to coordinate transactions between legacy mainframe systems that have very stringent rules about transactional commitments. This makes fulfilling end-to-end transactions in a tight timeframe a challenging task and one of the most important infrastructure concerns. By definition, this is the user experience service level in B2B and how that is achieved feasibly is the Quality of Experience.

Growth in B2B capabilities can have a profound effect on a company's agility and cost efficiency. However, managing this growth effectively hinges on the ability to:

- Provide secure transactional integrity to multiple business partners employing a wide variety of data formats
- Verify, validate and commit/fulfill a transaction in tight pre-defined timeframe
- Manage wide fluctuations in transaction volumes
- Add new partners and assign customer service levels based on business arrangements.

Overall Business Drivers

Both B2C and B2B applications share the following overall factors that drive differentiation and sustainable growth:

- **User Experience** - Guarantee that the right information is given to the right customer at the right time
- **Price/Performance** - Ability to transact business in the most optimal manner possible

- **Efficiency** - Guarantee a compelling User Experience at the best price performance possible while reducing and optimizing the data center footprint—power, cooling, and space

Businesses must meet these challenges by aligning IT with business leadership in order to establish a fit-for-purpose approach to the IT strategy. The business drivers initiate this process that in turns motivates the development of the IT architecture and infrastructure that meets the business objectives of the organization.

Creating this fit-for-purpose design is essential to avoiding a mismatch with provisioning and efficiency. Conversely, eCommerce businesses cannot grow, differentiate and win without a fit-for-purpose strategy that delivers on the three key factors of user experience, price/performance and efficiency.

THE IT CHALLENGES

The foregoing business drivers translate into a number of IT challenges, usually as a result of reacting to compute capacity constraints. For example, many businesses that have a strong B2C presence also have a comprehensive B2B environment that aids in fulfillment. However, large scale increases in B2C traffic will often trigger a corresponding increase in B2B volume. For many IT organizations, the only way they know how to ensure that there is sufficient capacity required to meet the user experience service levels is to constantly provision more servers.

The result of this reactive approach is that multiple instances of eCommerce applications are deployed in an attempt to achieve scalability and address throughput needs, with the following consequences:

- **Synchronization issues** – Complexity is increased due to the need to keep multiple instances in synch with application and infrastructure updates
- **Cycle of chronic inefficiency** – Given the isolated deployments, the overall process is inefficient due to not having the right amount of capacity for the right application at the right time. Further, the lack of predictability makes it difficult to determine which component of the overall infrastructure needs a certain level of capacity to fulfill service levels.
- **QoS Risk -** This inefficiency can have a profound effect on differentiated Quality of Service (QoS)for different classes of customers for both B2C and B2B applications. In either of the traditional architectural approaches—scale up or scale out—the only way to achieve better allocation of resources to higher classes of customers is by dedicating processing and memory resources to them. However, this ties up resources that will not be available to other classes of customers even when the gold class customers are not using all the resources.
- **Lost value -** It is harder to justify new expenditures because the typical utilization reporting shows only a 15% server utilization over a 24-hour period across hundreds of servers. Consequently, the Quality of Experience (QoE)

is not met because the costs—servers, power, cooling, and personnel—do not justify the level of service or performance. This is exacerbated by the complex mechanisms to provide failure recovery across and within data centers.

Memory Constraint Challenges

Other IT challenges relate to memory constraints. Often, there is limited scalability of each application instance due to transaction processing operating under a memory-bound model. Further, the memory requirements often exceed the limitations of a single traditional JVM. This creates additional challenges given that today's B2C user interaction model is more memory bound due to richer content and the need to maintain the current state for lengthy transactions. In addition, B2B interactions stress traditional JVM memory limits given they contain lengthy XML content and exercise considerable processing power due to complex parsing.

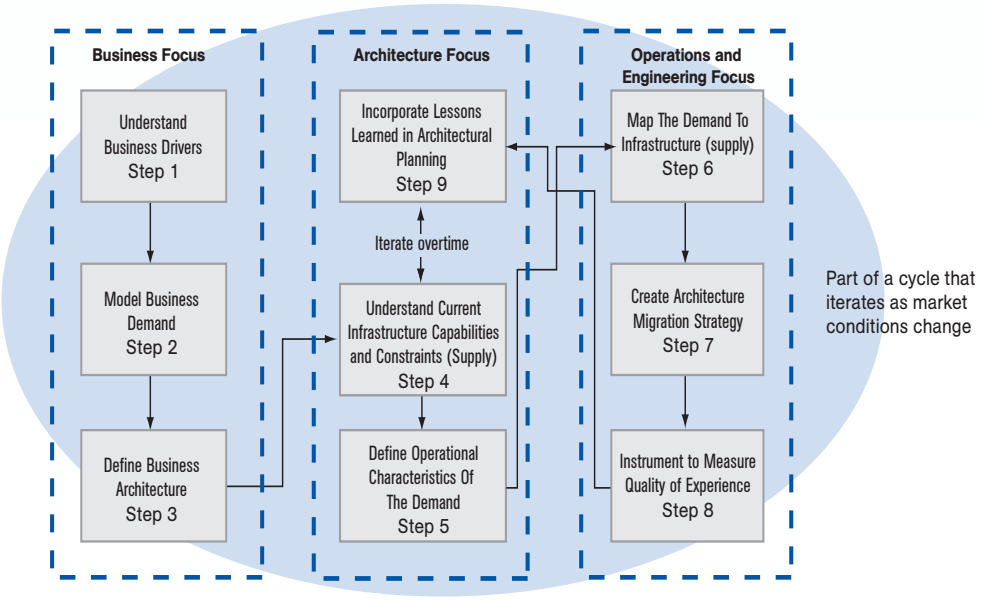
Unpredictable memory usage can also negatively impact user experience and service levels. The random and unpredictable arrival of requests makes it difficult to depend on a stable cache of memory. Thus, garbage collection occurs often because of the volatile nature of memory usage, thereby severely impacting the user's experience quality if response time is excessive. Adding further unpredictability is that the fact that garbage collection can occur at any time and cannot be easily mitigated.

These are all serious, complex issues that will affect the brand image of the enterprise, as well as the relationship between business, IT and operational functions of the organization. Taking a reactive approach to this problem will continue to degrade the Quality of Experience and the ability of the organization to adapt quickly to IT issues. The role that the architecture discipline takes is critical for ensuring success in an eCommerce environment.

BUSINESS DEMAND MODEL CREATION

This section presents a model for business demand. Understanding the characteristics of demand from a business point of view is a important step in any successful architectural design process because demand shapes the behavioral characteristics and processing patterns of the applications. If demand is not understood, then designing a "fit for purpose" infrastructure will not be possible and the operational team will be forced between over-provisioning–which is inefficient and impedes competitiveness–or "just in time" additions that create instability and risk damage to the brand.

The following diagram illustrates the entire architectural design process in formulating a fit for purpose infrastructure.



Note – While there are three areas of focus, there needs to be cross collaboration with the Architecture Group facilitating these discussions.

Understanding the complete process will assist the architecture teams responsible for the delivery of a high quality experience in their eCommerce infrastructure and application environment. These teams will face considerable challenges in not only making functional changes but also keeping up with business requirements which drive operational demands.

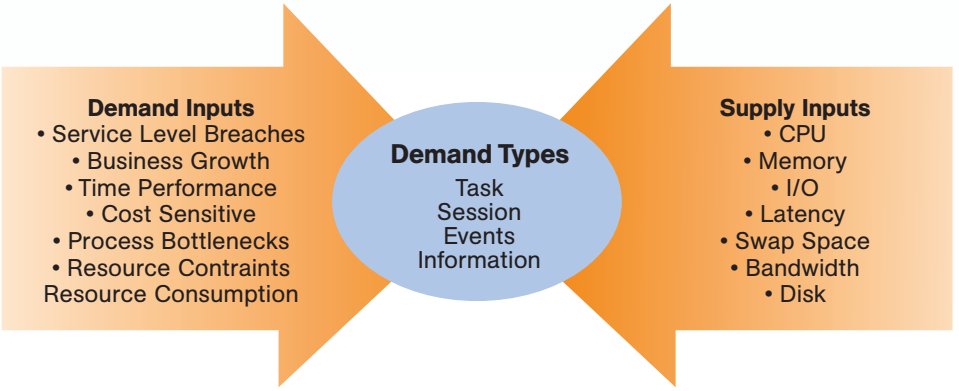
The following table is a subset of tasks that need to be performed by business and IT managers to build a Service Oriented Enterprise (SOE) that achieves efficient Quality of Experience for all stakeholders in terms of user experience, total cost, and performance.

STEPS FOR SUCCESSFULLY ARCHITECTING SCALABLE "FIT FOR PURPOSE" E-COMMERCE INFRASTRUCTURE AND APPLICATIONS

STEP	DESCRIPTION	RESULT
Understand The Business Drivers	What drives revenue, risk and cost? Understand what the Quality of Experience (QoE) will be for the clients and executives	Balance the economic model of the firm by using the fulfillment of business goals to align IT with business, taking the first step towards making IT a strategic asset.
Model The Business Demand	Determine the impact of QoE- Understand the high value traffic, alerts, constraints, bottlenecks, anticipated demand, time sensitivities	Determine HOW demand shapes revenue, cost and risk, providing a strategy for measurement, a first step in dynamic adaptation
Define The Business Architecture	Build on a stable foundation- The business architecture should ground any technical architecture because it represents the most stable factors of the business (how does it make money, adhere to regulations, report its state, avoid risk). This is the best chance IT has to align with the Business to make it a relevant asset instead of just a cost center	Defining the major entities, use cases, functions and data that a business uses provides a stable base from which to build the technical architectures required to create a "fit for purpose" infrastructure. Any business will change its models or risk profiles as the market demands (e.g. new channels, new processes), but the fundamental business model (e.g. a bank, a retail store) is static compared to the changes in the technical environment.
Understand The Existing Infrastructure Capabilities And Constraints	Determine what is feasible short and long term. - It is vital that the supply within the current infrastructure is known and that a strategy for managing that supply is created. Any new architectural changes need to take this into account	The capabilities (server types, bandwidth, growth) and constraints (power, cooling, network, square feet) are managed properly to drive a better user experience (price/ performance, efficiency)
Define The Operational Characteristics Of The Demand	Identify Operational Qualities- How do the requirements affect the operational qualities of the infrastructure and applications. What are the latency sensitive transactions, how severe is the change in throughput in a given time period, what is the maximum time to recover for key applications, how will load balancing be achieved.	Understand the technical driving forces that will shape solutions that are "fit for purpose". Knowing how the infrastructure must behave to fulfill the throughput, latency and memory dependent requirements is critical to perform Architecture Tradeoff Analysis.
Map The Demand To Infrastructure	Link User Experience with Business & IT. Perform Architecture Tradeoff Analysis to make the most feasible decisions given user experience, cost, and efficiency.	Map the demands to the supply of CPU, I/O, network bandwidth, disk, latency bottlenecks, swap and page space
Create An Architecture Migration Strategy	Determine the impact of change. This is crucial to ensure that high value customer facing applications get priority, while the effort also enhances stability.	A Blueprint of the migration based upon business needs and current capabilities, addressing highest pain points first
Instrument To Measure The QOE	Provide a feedback mechanism to gauge success, impact and next steps, and strengthen the alignment between business and IT.	Measuring performance comprehensively is critical to success, Measures of (CPU, transaction latency, Memory, Network, Disk, Cache) within an application [on all servers] and across application classes. Measures must roll up to give the total view of Quality of Experience (Cost, Performance, User Experience
Incorporate Lessons Learned In Architectural Planning	Create an Adaptive Infrastructure- that uses the metrics and changes in business needs to influence architecture changes and future migration	Ensure that all future migration of the infrastructure is a strategic one, so that IT becomes a competitive edge instead of a cost to be endured.

MAPPING DEMAND TO INFRASTRUCTURE

Once the demand model has been created and the tradeoff analysis is complete, the next step is mapping the demand to the infrastructure. Completing the tradeoff analysis will enables managers to gain a realistic view of what the current infrastructure can provide, allowing them to determine how to best fit applications to the infrastructure based on behaviors and qualities. It is important to remember that any large scale B2C or B2B environment has a significant infrastructure investment with many types of processing behavioral patterns. These behaviors need to be viewed holistically. For example, imagine a change that triggered a 1,000% increase in B2C transactions without any change to the B2B infrastructure. The fulfillment engine would be overwhelmed and the company would suffer irreparable brand damage.



The Demand Mapping exercise allows the business drivers to be aligned with the IT constraints, giving all stakeholders an opportunity to make feasible investment decisions. For example, it is clear that an Azul Network Attached Processor can provide significant benefit in response time if the transaction rate exceeds 2,500 transactions per second. If the target is much less, say 1,000 per second, Azul still might have a major impact on consolidation efforts. If the problem is not framed correctly, the business will not see a result that solves their business need and this choice could be construed as a failure.

Fit for Purpose is a critical principle in the Service Oriented Enterprise. The term means that the investment in an IT solution is measured and considered against numerous business and technical driving forces. Technical driving forces are addressed through tactics that will enhance or augment technical capabilities, as briefly reviewed later in the paper.

THE OPERATIONAL CHARACTERISTICS OF E-COMMERCE SYSTEMS

The remainder of the paper will shift to the work required to incorporating the business drivers and demand model into the eCommerce architecture and infrastructure. To get the process started, we first drill down to describe the operational characteristics of today's eCommerce systems. Given the demand model, drivers and business architecture, it is essential to examine operational characteristics against the current infrastructure and determine the feasibility of meeting the behavioral characteristics.

Business-to-Consumer Operating Characteristics

Typical large-scale B2C environments exhibit the following processing patterns:

- Throughput is unpredictable, hard to control and can spike quickly in a short amount of time
- Response time must be perceived as consistent by the user, despite large fluctuations in traffic
- The environments that exhibit these high throughput patterns have grown exponentially, primarily due to the removal of speed barriers by broadband land-line based access. In addition, wireless networks have provided more frequent access with a wider range of users
- The demand for rich user experiences has transformed interactions to evolve to highly stateful, memory-bound processes. As a result, memory requirements for a typical transaction have increased significantly
- Predictable response times are the critical measure of success, but will be hampered by J2EE architectures due to memory limitations in the traditional JVM, which will also cause arbitrary invocations of garbage collection.

Business-to-Business Operating Characteristics

Typical large-scale B2B environments exhibit the following processing patterns

- Large spikes in B2C traffic will cause a corresponding spike in fulfillment traffic for B2B partners
- Multiple diverse message formats need to be processed quickly as part of an overall SLA requirement. These messages can be quite large as they are in XML format, further taxing memory requirements.
- Transactions occur between two databases that might view the transaction as multi-phase, and thus commitments in a tight timeframe are imperative.

The following table compares a scale-out architecture approach (that is, the proliferation of blade racks) with a scale-up architecture (building bigger servers). For comparative purposes, we also show the unit-of-work offload architecture that an Azul Networked Attached Processor would provide.

OPERATIONAL PERFORMANCE QUALITIES AND ARCHITECTURAL SOLUTIONS

Operational Quality	Definition	eCommerce Context	Scale Up Architecture Tactics & issues	Scale out Architecture Tactics & issues	Azul Systems Appliance Differentiation
Throughput	The amount of discrete transactions (unit of work) that must be processed through the entire system in critical intervals	Note: In most eCommerce applications the number of input messages are not 1:1 with the numbers of output messages generated, due to alerts	As transactions per second reach the 1000's , Java memory limitations will ensue	Individual instances will be slowed handling high 100's of transactions/ second, dependent on massive scale out in the 1000's	If greater than 2500 transactions/ second is required, more of a consolidation issue if less. Minimal 5x- 19x throughput is achieved
Throughput Peak Characteristics	The Duration, Extent and Frequency of the peak transaction load. It is important to capture all three, as they can have profound effects on utilization	•B2C eCommerce Applications will receive unbridled input, with unpredictable spikes. •B2B eCommerce Applications might be able to throttle input depending on the protocols invoked	Sustained High Peak will cause accelerated memory usage forcing more frequent garbage collection , causing cascading delays after a certain threshold. Scale up and Scale out tactics are essentially equivalent in Java applications, due to JVM scaling limitations of ~2GB.		Sustained high peak is maintained because garbage collection is handled discretely by the hardware, minimizing the GC cycle. The result is garbage collection pauses are practically eliminated.
Latency	The maximum acceptable time it takes to process a transaction.	Since this category is usually relevant for LOW Latency applications such as an eCommerce Portal the expectations are that answers would be in milliseconds	Scale out and Scale up are essentially equivalent: Latency is unpredictable in practice, because the transactions are memory bound , and memory use per transaction is high. Since each traditional JVM instance is limited to 2-3 GB the amount you can cache in such small footprints is not very effective.. This is a critical drawback because the incoming traffic is not uniform enough, making overlapping data reads less likely, making the cache to be highly volatile, which causes frequent GC cycles. GC cycles and the relatively small number of CPUs have a cascading effect as they cause transaction queuing., which cannot be easily drained by offloading more processing resources to applications with the queue, because there is no variable based workload scheduling		While the transfer through the proxy will add latency on the round trip to and from the NAP it is predictable and can be accounted for. Latency within the NAP will be less because GC pauses are eliminated. The Worst case is still better than the other 2 choices due to the lack of a big GC pause. The reason GC pauses are eliminated is because Azul gives almost unlimited ability to cache (100's GB), as long as there is data that can be cached. The other main reason for maintaining low latency with Azul is the consistency of processing transaction throughput at peak levels , which will prevent large queues from forming. With Azul, all of the spare CPU's can be given to the applications under load and hence for MOST parallel transactions there will be no necessity to wait in queues for processing

Operational Quality	Definition	eCommerce Context	Scale Up Architecture Tactics & issues	Scale out Architecture Tactics & issues	Azul Systems Appliance Differentiation
Response Time	This implies that there is an entity (human or other application) that is expecting a response.	• B2C must respond to the end user in 3 seconds or less • B2B will have tight SLA's for end to end transaction commitment	Individual response time for any given transaction is not predictable during peak loads due to latency issues discussed below	Response times will vary even more in a scale out architecture, because of GC issues mentioned above, and the fact that a data center grows in spurts, so scale out architectures might have different aged hardware servicing requests	The round trip through the proxy will add to response time, but the realized decrease in response time has been 10x in many cases. The response time is predictable, which is a critical measure. The Worst case is still better than the other 2 choices due to the lack of a big GC pause.
Distribution Of Compute Load	Does this particular application solve problems that can be decomposed into smaller chunks and thus can be deployed in a distributed computing model?	• B2C will likely have one transaction be processed as one executable thread • B2B might have multiple threads to parse the XML (likely format), and might need to assemble data from multiple data sources	Scale up tactics allow for partitioning of the compute load, with shared memory, but processor assignments tend to be static, peak loads will stress the scale up system unless it is massively over provisioned	The limitation is the number of blade racks, which if large, increases the complexity , heat, square feet usage, and operational maintenance	Azul has capacity to run 200 VM's on over 100's of processors cores and 100's of GB of memory
Calculation Data	The amount of data required to complete a unit of work.	• B2C might have considerable amount of data to stream or hold onto • B2B will process large messages that could contain many sub-records	Scale up architectures will be able to utilize the capacity of a large machine for large data sets, but there are inherent limitations for all traditional JVMs and garbage collection will be a factor because memory usage will be volatile	In scale out architectures there are inherent limits in traditional JVMs that will be stressed by volatile memory usage. Sharing schemes need to be weighed because CPU proximity to memory will be a latency factor in large distributed caches.	Azul processors have a large amount of RAM-100's of GB. Most B2B or B2C applications will be able to be serviced by this memory pool. As mentioned earlier GC will not be an issue with large volatile sets.
Calculation Size	What is a typical unit of work for this system? Size of a unit work until some transaction is accomplished.	• B2C applications would have small units of work, because the emphasis is on massive throughput and low latency, so the amount of data manipulation would be very small. • B2B will tend to have larger calculations to perform than B2C	Scale up architectures will depend upon a partitioning scheme that tends to be static once processing elements have been assigned. Large transformations during peak times will cause bottlenecks	Scale out architectures will be stressed at peak loads if all the transactions have large conversions, the strategy then becomes how far the scale out occurs to handle peak. Most projects over-provision scale out architectures for peak, but rarely make the processing power available for other uses during non-peak	Azul can accelerate repeatable translation tasks such as XML conversions. It is not a floating point numerical processor however, which should NOT be an issue for most B2C or B2B applications

Concurrent Users	The number of expected simultaneous users in the system.	• B2C- would be harder to predict, but there are likely seasonal trends to act as a basis • B2B- would be easier to predict, because it is based on partner traffic. It gets more difficult if B2C drives the B2B	Scale up architectures use partitioned processing tactics, and shared memory resources to achieve a high degree of concurrency. All the factors mentioned above regarding volatile memory, complexity, static processing assignments still apply. Response time will suffer at peak. Scale up and Scale out tactics are essentially equivalent in Java applications.	Scale out architectures achieve concurrency through fan out processing, distributed caching, federated views, but still have inherent VM limitations and cause server rack sprawl which affects operational stability and affect cooling, power and space. Response times will trend to suffer at peak, unless over -provisioning occurs. Scale up and Scale out tactics are essentially equivalent in Java applications.	Azul provides the benefits of the shared memory resource tactic, and enhances cooperative processing by allowing multiple platform environments to off-load work through a proxy, enabling a more sustainable concurrency model with predictable response times.
Scalability Approach	Scale becomes complicated if state needs to be maintained for transactions that are in flight through the various tiers of an application. The typical choices for state maintenance are: Stateful- (Partitioned or Affinity), Stateless, Transparent to the processing (state kept at the client or on edge servers)	• Most B2C eCommerce applications that exploit WEB 2.0 will keep state. The question becomes how the state is managed: •In B2C some of the most popular content could be preloaded (e.g. promotional sales, static content, some catalogs) • In B2B processing rules could be preloaded. The key for either is the size of the state and the degree that the state could be subject to memory limitations	Scale up architectures use a large shared memory pool to make scalability easier, but the model still has limitations as mentioned in previous cells, but applications can view state and scale more transparently in this model.	Scale out architectures will attempt to use various distributed caching models for this, but will have memory volatility issues as mentioned before. Transparency is harder to achieve without supplemental products. All previous limitations on memory usage will still be a factor	As mentioned in previous cells, scalability has many factors, and state is a critical one in B2B and B2C. Application can view state more transparently in this model than in others because the shared memory resource is so large

For additional information, we refer readers to The Software Engineering Institute who advocates the Architectural Tradeoff Analysis Method (ATAM). This methodology explores many facets of a software system, including performance, availability, portability, modifiability and other qualities.

REVISITING TECHNICAL CAPABILITIES IN CONTEXT

The comparison matrix in the previous section highlighted performance. However, other capabilities that should be addressed include operational complexity which can affect operational stability. That is why demand mapping is not just an exercise in aligning inventory, since placing certain application behaviors on a scale-out architecture could force companies to inappropriately increase their investment in scale-out architectures. This section elaborates on performance and complexity in the context of the comparison matrix.

The Scale-Up Approach

The ideal benefits of the scale-up approach are higher processing utilization and fewer "moving parts", which should lead to increased operational stability. In principle, the use of Java should produce fewer instances of the application because of its threading model. In this multi-threaded model the number of logical instances goes up to accommodate the workload. However, when implemented in real world environments, this idealistic model has proven to be flawed due to the following:

- **Memory limitations** – For Java applications, the operating result ends up being no different than the scale-out tactic from a logical application deployment viewpoint, even though the applications may be running on fewer but larger machines. The gating factor is the size of an instance of a Java application which is limited by the amount of practical usable memory (typically ~ 2 GB). That limitation is the same whether the system has 100 processors and 1 TB of memory or 2 processors and 8 GB of memory. Further, the heap of ~ 2 GB gets heavily exercised causing frequent garbage collection pauses due to the memory bound nature of the transactions. These pauses negatively impact the user experience.
- **Multi-threading consequences** – The need for more execution threads increases complexity, making it as complex as the scale- out tactic. Java applications with small memory requirements need tens of threads to be effective. So, many such instances drive up the overall number of threads in a system to the point where conventional operating systems do not scale well, especially when approaching thousands of threads.
- **Partition limitation consequences** – Partitioning is used to reduce complexity and multi-thread overload. It often requires segmenting the input based upon rules to reduce overlap. The Partitioning tactic employs the running of Virtual or Logical Partitions (VPARs or LPARs), that segment the processing load. The problem with this tactic is the loss of primitive resource sharing capability of deploying on a single shared OS, thus defeating the purpose of scale-up. Most scale up tactics allow for execution partitions, based upon assigning a fixed amount of processing to an application. As the number of applications increase, the complexity remains basically the same. However, difficulties arise because IT must create its own "virtual

platforms" within the larger frame to house the different versions of system and infrastructure software.

In summary, deployments on large-scale machines are not logically that different from the scale-out model. In other words, large numbers of small instances, even if they are hosted on the same physical machine, are still hard to manage and share resources effectively.

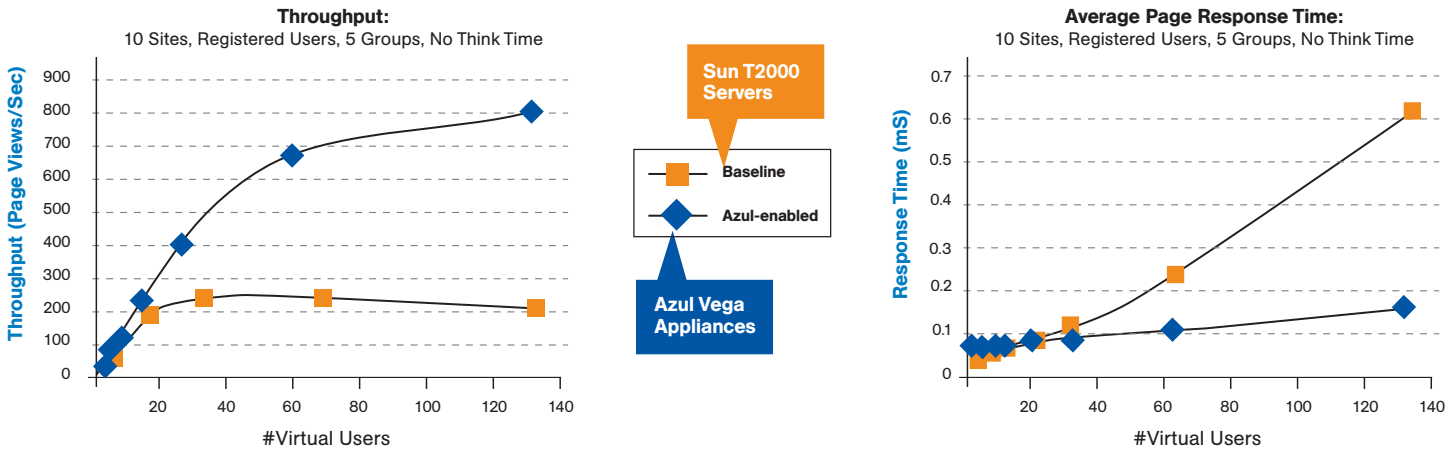
The Scale-Out Approach

Scale-out tactics will cause processing silos to occur, with very little sharing. This has many negative consequences, including:

- Low utilization – Scale-out tactics tends to produce less utilization over a 24-hour basis which is very inefficient, making it difficult to justify expansion.
- Datacenter expansion – This tactic creates datacenter footprint sprawl, increasing complexity and costs.
- Higher licensing costs – Scale-out typically forces a company to buy excessive amounts of software licenses, further driving up costs
- Increased complexity – Complexity increases which invariably leads to component failure due to operational or configuration errors, consequently negatively impacting customer service.

Azul's Appliance Approach: Scale without Sprawl

Azul achieves the benefits of scale-up without the configuration issues. Azul emphasizes cooperative processing which is based on sharing common system resources in a controlled, priority-based methodology. In addition, Azul allows many different types of machines (x86 or RISC) to utilize the same network appliance. In this manner, workloads from different platforms can be off-loaded, thereby avoiding the need to invest in additional legacy equipment to meet computing requirements.

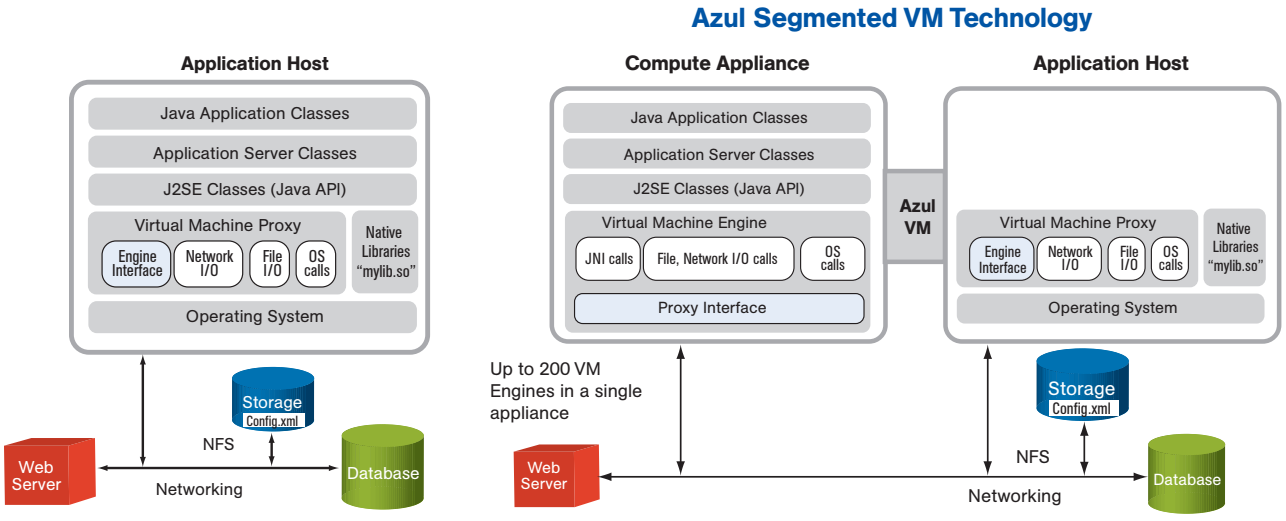


ARCHITECTURE MIGRATION STRATEGY

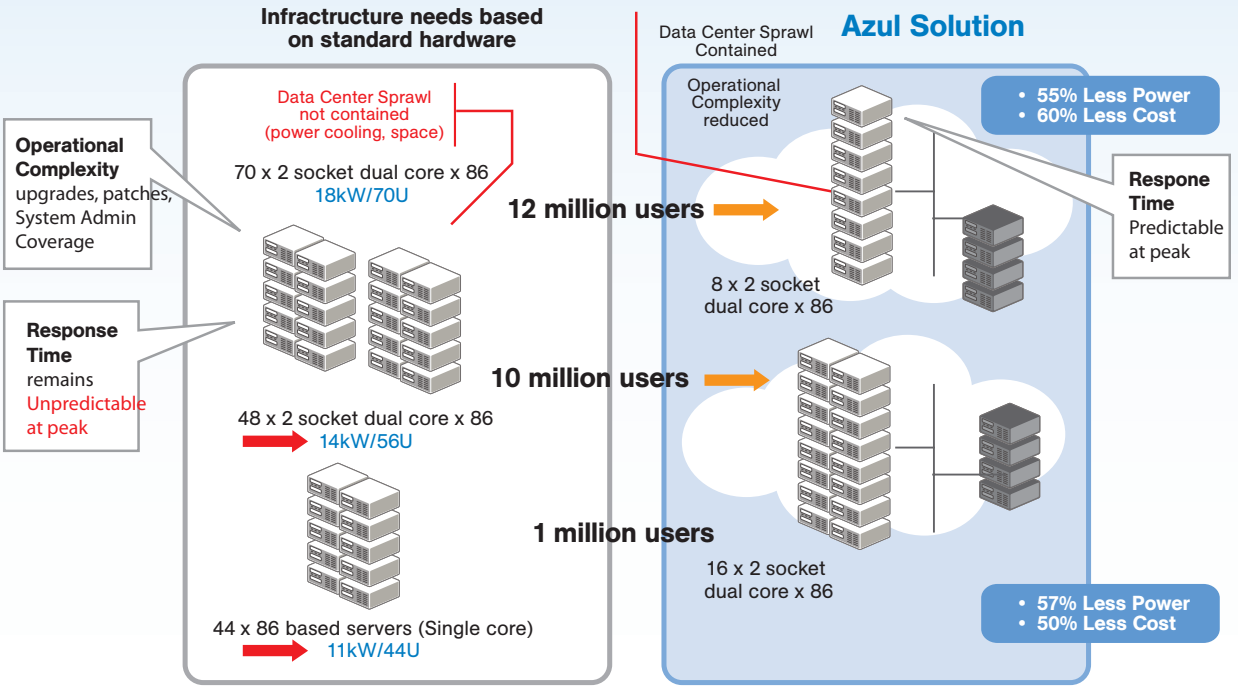
Any introduction of new technology within the organization requires investment in migration and integration of the technology. It is essential to introduce new technology in a phased manner, allowing the opportunity for adoption and maximum success. New technology presents a learning curve and is often viewed by operations as a risk. Therefore, scenario-based prototypes performed in a lab environment are an ideal way to initiate the implementation. Integration should be featured in this phase. It is also important to define and design how the technology will be introduced into production, with details of how it will run on day one and the backup plan to be deployed if necessary. All these aspects are an integral part of the Quality of Experience metrics mentioned in the next section.

Migration to Azul Systems

In the case of Azul, the migration strategy is easier than most new products. It was designed to be transparent to J2EE applications. There is an Azul proxy that runs on the server machine which acts as conduit between the application server and the Azul processor. This proxy will offload all of the work to the Azul processor. The only change to the code is in the application server configuration file, which requires a one-line change. Azul will also need power and network cabling, but that is offset by the number of servers that will be decommissioned.

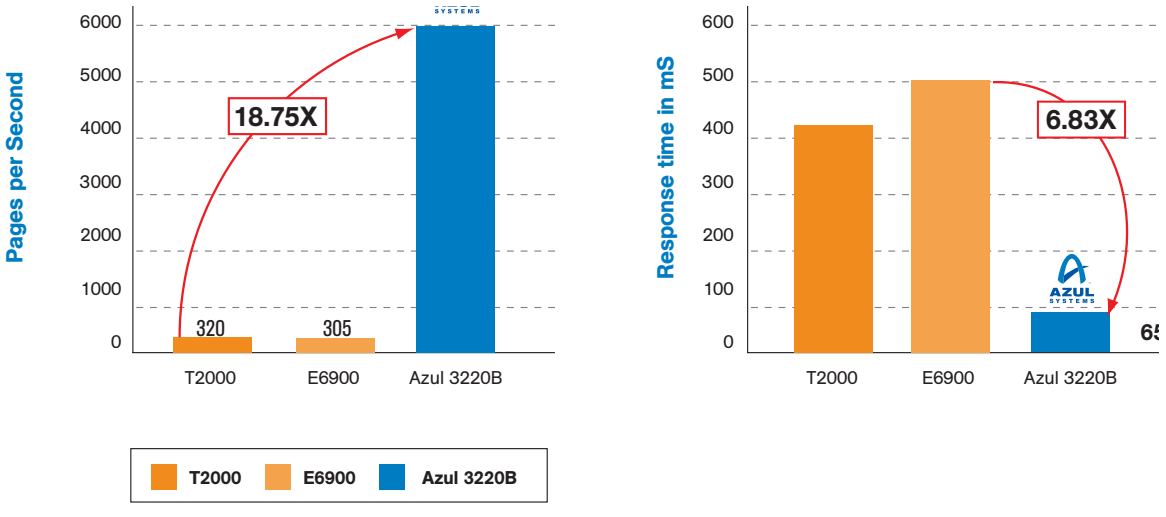


The above diagram illustrates the impact of transferring the heavy workload from the application server to a network appliance that can run 200 VMs simultaneously with up to 864 processors and 768 GB of memory. This configuration is designed to eliminate disruptive garbage collection pauses. With the Azul appliance, the Web portal architecture will benefit from significantly improved through scalability and consistent fast response times; with capabilities to sustain over 5000 page views per second while delivering low consistent response times.



Benefits of Placing Azul in an Existing Environment

Once it has been decided to place an Azul Appliance in an existing environment, clear benefits can be demonstrated where Azul can substantially boost performance and reduce complexity. The following diagram illustrates the difference in architecture between a conventional scale-out architecture and an Azul implementation.



Azul in the Real World

The below diagram of a real world portal deployed in a major retailer illustrates the power of the Azul approach as it enables significant improvements in the user experience, response time and throughput. These results lead us to the next step in the SOE process, which is how to track progress and measure success.

INSTRUMENTATION STRATEGY DEVELOPMENT

The Quality of Experience (QoE) is about optimizing the user experience while balancing performance, cost and efficiency. It means that the applications and the supporting infrastructure must be highly adaptive in order to respond to unexpected demand and deliver consistent performance. That is why it is imperative to measure the results from different perspectives. This includes all of the performance criteria mentioned in the tradeoff matrix, as well as any other criteria that is used in the decision making. Questions that should be considered include:

- Is traffic increasing?
- Are users satisfied and is the brand in tact?
- In terms of total server utilization, is it going up when viewed from a senior IT executive perspective?
- Are the number of servers being ordered and used going down quarterly?
- Is the operating environment more stable?
- Are cooling and power costs being reduced?
- Is the ratio of system administrators to servers improving?
- Is the square footage available in the data center increasing?

The adaptive SOE requires commitment to agility. With the above questions as a starting point, the operating environment must be measured, architected, designed and redesigned as necessary to meet the needs of the business. It requires discipline and a desire to be responsive to the business. The end result is relevance, where IT is an asset and strategic partner, not another cost center. In a fast global paced world, the choice is becoming binary—either relevance through a highly adaptive model, or irrelevance and marginalization.

CONCLUSION

With the mainstream adoption of e-commerce applications for Internet banking, online insurance, e-auctioning, online travel booking and e-trading, enterprises are facing extremely demanding levels of performance, scalability and availability. These transaction-intensive applications require fast, reliable service levels and demand more capacity and scale than what is currently available from today's traditional systems architectures.

Forward-thinking companies are moving beyond the limits of today's traditional servers by strategically designing architectures and supporting infrastructures to meet the unique scalability, reliability and memory requirements of extreme Java™ deployments, ensuring guaranteed service levels that drive top-line revenue growth.

ABOUT AZUL SYSTEMS

Azul Systems is a global provider of enterprise server appliances that deliver compute and memory resources as a shared network service to transaction-intensive applications, such as those built on the Java™ platform. Azul Compute Appliances enable transparent, massively scalable infrastructure to support the business priorities of today's most demanding enterprise environments and deliver increased capabilities, capacity, and utilization at a fraction of the cost of traditional computing models. More information on Azul Systems can be found at www.azulsystems.com.

1600 Plymouth Street, Mountain View, CA 94043 T 650.230.6500 | F 650.230.6600 | www.azulsystems.com