lustratus

Research

Choosing the right SOA platform on System z™

Understanding the SOA workload trade-offs between z/OS and Linux on System z

Author: *Steve Craggs*

September 2008

Version 1.00

Table of Contents

Executive Summary

Many IBM System z users are actively looking to drive greater returns from their mainframe investments through the use of service-oriented architecture (SOA), bringing the mainframe with all its applications and data into full collaboration with distributed, enterprise-wide investments. Not only does this make System z applications accessible to other systems, and vice versa, but SOA also promises greater agility and a closer alignment of IT with business goals. While some System z users concentrate their SOA workloads in the distributed world, effectively turning the mainframe into a service-provider, many others have realized that System z can provide considerably greater value by taking a more active or even lead role in running SOA-based applications and activities, whether consolidating existing Linux servers or simply providing a SOA 'super-server'. This allows the System z strengths such as availability, reliability, scalability and efficiency to be leveraged by the enterprise SOA workload.

But there is a problem. Many companies interested in SOA have started to embrace the Linux operating environment as a key part of IT strategy, largely due to the ability to standardize on a single environment across multiple different hardware platforms. System z is no exception here - IBM offers Linux on System z as a way of providing the same environment on the mainframe, as an alternative to the proprietary z/OS environment that is native to System z. However this introduces a key decision – should the SOA workload on System z go with the standardized approach and use Linux on System z, or would it be better to make an exception in the case of the mainframe and go with a z/OS environment?

Today, companies are often making this decision without all the relevant information, often dependent on who shouts the loudest. For those users with a powerful Linux organization, the argument will be that System z is no different to any other platform and should fall in line and use Linux on System z. Users where the mainframe organization holds the power will probably throw up their hands in horror at the idea of Linux on the precious mainframe, and will demand a z/OS approach.

However, this decision has major ramifications, and it is imperative that users take the time to consider all the facts before making this key platform selection decision; important influencing factors include quality of service requirements, cost implications and the make-up of the SOA workload. Getting the decision right will greatly increase the value IT can deliver to the business. This paper looks at the factors to consider in order to make a balanced judgement, based on the relative importance of these factors to the enterprise.

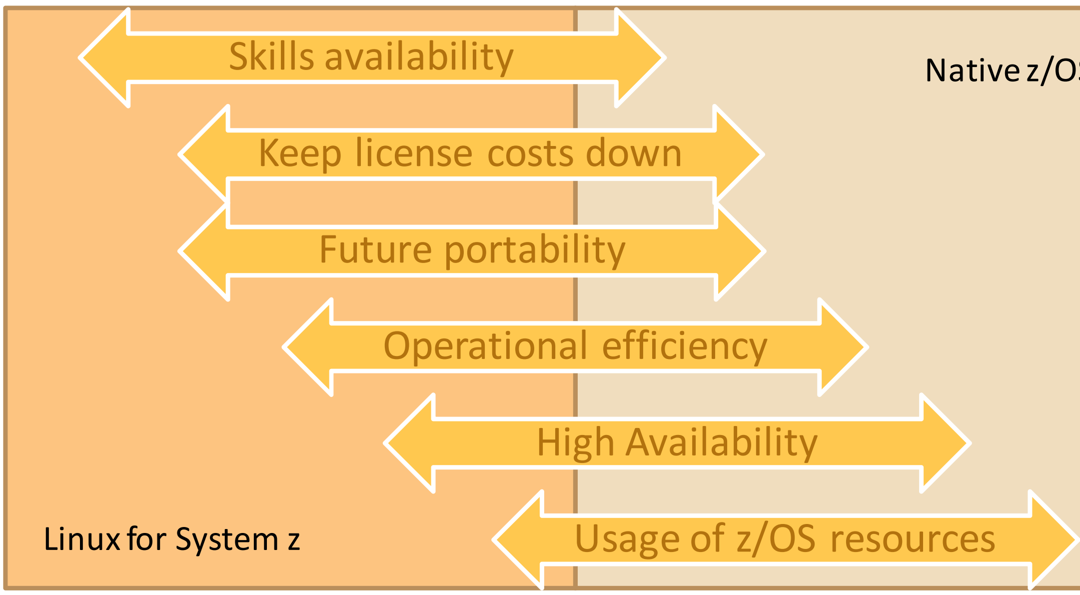

Figure 1: Balancing the platform implications against enterprise needs

Introduction

As more and more companies opt for a service-oriented strategy for their IT implementations, based around SOA, it has become clear that there is considerable confusion over the decision of where best to host the various parts of the SOA workload. There are a number of different elements to this decision. For a start, a company with multiple hardware platforms must decide how to spread the load across the different options. These could include System z mainframes, distributed servers and even Intel-based Windows workstations. However, even when the hardware platform decision is made, there is often a question of which operating system to use – for instance, on distributed servers there could be the option of running LINUX or some proprietary operating system, while on System z there is a choice between z/OS or Linux on System z.

Choosing the right platform, that is the most appropriate hardware and operating system combination, for a particular workload can be a tricky decision. A number of factors must be taken into account as part of the evaluation process before any final decision is made.

Platform selection

There are generally three key requirement areas that will to a large extent govern the decision of the best fit platform, although in specific circumstances there may be a range of additional considerations to take into account that could also be very influential. The three most important requirement areas are:

- Quality of Service
- Cost of Ownership
- Enterprise considerations

Quality of service and cost of ownership tend to be at opposite ends of a balance. Delivering a higher quality of service generally results in higher investment costs, while lower service level requirements allow more options in platform selection and configuration with a corresponding reduction in costs. A workload requiring a high degree of availability might need redundant resources, for instance, increasing costs accordingly. However, platform selection will still be affected by the relevant weightings of these two requirement areas. Top of the range performance or scalability needs might require specialized hardware, while platform flexibility and adaptability might result in a lower overall cost of ownership.

Enterprise considerations reflect relevant aspects of enterprise strategy, policies and investments. For a start, specific technical requirements may end up dictating platform selection. For example, a company might have a policy that all data transferred between applications must be hardware encrypted. In this case, only hardware and software combinations that support the desired form of hardware encryption are eligible for selection. Or perhaps a workload might be dependent on support for an application environment or package that only runs on a specific platform. Another enterprise consideration might be the desire for a high degree of operational flexibility, such as the ability to port the workload onto a range of other system types. Some companies have an open systems policy, for instance, where all workloads must be transferrable to any other platform, thereby restricting platform choice. Other companies might have a policy of limiting the number of IT suppliers, again reducing the options. Available skills could be a further determining factor.

System z considerations

This paper is aimed at operating system selection for those who have already chosen System z as a hardware platform to serve SOA needs. However, before proceeding it is worth reflecting on some of the general System z characteristics in terms of general platform selection factors. It almost goes without saying that, for SOA workloads involving mainframe environments such as CICS and IMS, System z is the logical hardware platform to satisfy these workload needs. Any other choice would insert an extra overhead since these systems are mainframe-based. However the major contributions from the System z platform lie in the Quality of Service area, and to a lesser extent Cost of Ownership.

The IBM mainframe has always had a strong reputation for delivering high levels of service. If asked to name mainframe strengths, most users will talk about reliability, scalability, performance, security, integrity, availability and consistency. IBM is constantly driving these factors to higher and higher levels, with the most recent IBM System z10 EC platform offering substantially greater performance and capacity than previous models. Therefore, if these characteristics are critical IBM System z is likely to be a key candidate hardware platform.

On the cost of ownership front, the combination of increased processing capacity with virtualization technology makes System z a logical choice if server consolidation is desired. For example, IBM claims the new System z10 EC can consolidate the workloads of 1500 PC servers, with major savings in power consumption, heat generation and space requirements as well as a higher level of resource utilization. With the rapidly increasing costs of energy, heat dispersal and data centre space, these benefits can significantly bring cost of ownership down, while also adding a higher degree of flexibility in resource usage to smooth out peaks and troughs of demand.

Finally, manageability can be a major factor affecting both quality of service and cost of ownership. System z management tools can leverage the concentration of resources that the mainframe model offers, delivering a productive and cost-effective environment for implementing corporate policy in areas such as security, audit, data management and business continuity. The result is a substantial shift in personnel deployment from maintenance to delivering new business solutions, as confirmed recently by a major US bank which consolidated its LINUX and Intel platforms onto System z and saw the ratio of personnel working on support versus new projects move from 40/60 to 30/70.

SOA Infrastructure Components

Before looking at specific differences between handling SOA workloads on z/OS or Linux on System z, it is necessary to review which SOA components are likely to be relevant to the platform selection question. This makes it possible to work through the various selection criteria based on the characteristics of the two operating systems and their corresponding SOA component versions.

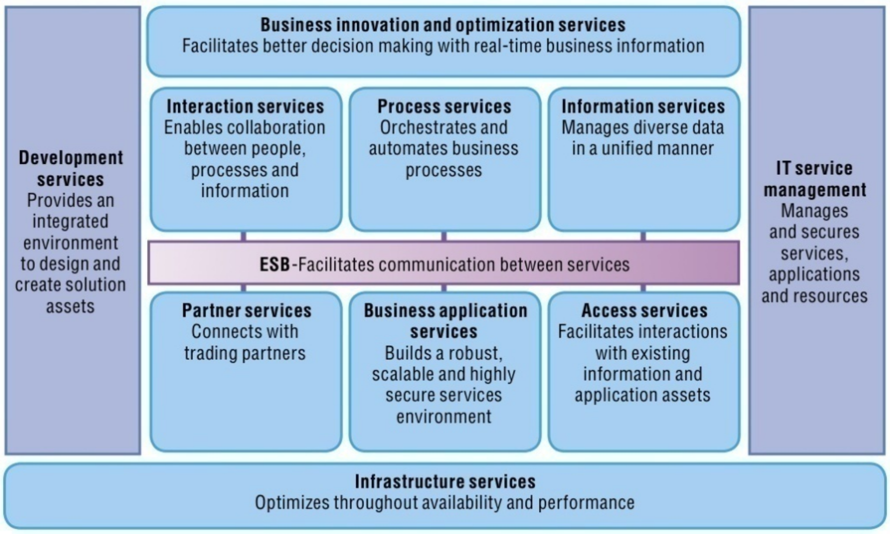

Figure 2:- IBM SOA Reference Architecture (Source:- IBM)

There are numerous versions of SOA reference architectures, all saying roughly the same thing but in different ways. However, given the decision to look at SOA on IBM hardware, in the shape of the System z, it seems appropriate to use the IBM version for the purposes of this discussion shown in Figure 2 above.

Similarly, although other SOA tools can run in the System z environment, for the purposes of this paper the IBM product line will be used in the forthcoming discussions about platform selection. The IBM SOA infrastructure products that map against the relevant parts of the reference architecture are listed below. Although any of these products might have an influence on platform selection, the most relevant ones are highlighted. These products form the engine room for SOA processing, together with the connectivity to the z/OS specific application environments.

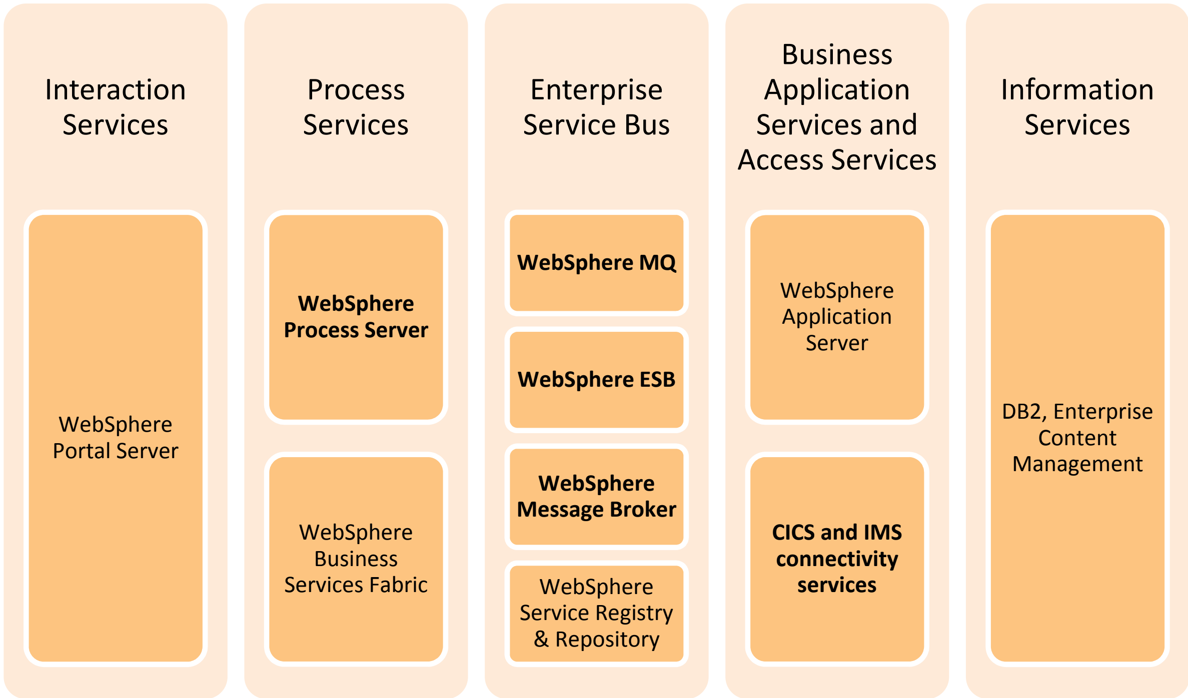

Figure 3: Key members of the IBM SOA product portfolio

The highlighted products have most relevance to the platform selection decision on System z. WebSphere MQ, WebSphere ESB and WebSphere Message Broker carry out the information exchanges between the SOA components and provide associated mediation services such as data transformation or intelligent routing. WebSphere Process Server enhances the linkage between the business processes being implemented and the underlying IT implementation. The CICS Transaction Server and IMS connectivity services, provided by CICS Transaction Gateway and the IMS product itself respectively, enable the SOA workload to encompass applications running within these z/OS transaction processing environments.

Platform selection for SOA on System z

Having looked briefly at the key criteria for a typical platform selection process, and some of the general characteristics offered by the System z hardware platform, it is now time to focus on which System z operating system to use to support the mainframe SOA workload – z/OS or Linux on System z.

Before going into detail on the various different platform selection factors to consider, it is worth taking a step back. A lot of the forthcoming assessments stem from the fact that z/OS is the native mainframe operating system that has grown over time with the System z and preceding mainframe platforms. As a result, z/OS is the result of a continual process of evolution over the last thirty years or so. In contrast, Linux on System z is an implementation of the Linux distributed operating system that runs on System z. In other words, Linux on System z is to all intents and purposes a Linux platform just like Linux on any other server platform, and it has great affinity with these other Linux implementations at the operational and 'look and feel' levels.

Therefore, as a general rule of thumb, if a user is interested in any of the many advanced System z characteristics, this will tend to push the platform selection decision towards z/OS, while if the user is wanting to maintain consistency with a wider Linux implementation, such as in the case of consolidation of a set of Linux servers, then the decision is likely to lean towards Linux on System z. In the subsequent discussions, this theme will surface frequently.

Now it is time to look at the various capabilities and features that should be taken into account when making the platform selection for mainframe SOA workloads.

Quality of Service

Following on from the generic comments on System z, it will come as no surprise that System z offers the opportunity to deliver extremely high levels of service to the business. However, these capabilities are not exploited equally by the SOA infrastructure components on the z/OS and Linux on System z platforms, and therefore the platform selection decision will depend to a great extent on user needs in this area. Essentially these features are either z/OS only or at their most effective in a z/OS environment.

Key QoS enablers

The five most pertinent System z features to be considered in this area are:

- Parallel Sysplex
    - The ability to run multiple System z mainframes as a single, logical processing platform
- GDPS (Geographically Dispersed Parallel Sysplex)
    - Extending the Sysplex concept to mainframes running in separate locations
- ARM (Automatic Restart Management)
    - Policy-based control over the way applications restart after a failure
- WLM (Workload Manager)
    - Automatic allocation of resources to particular workloads based on pre-defined criteria
- RACF (Security) for z/OS resources
    - High performance, granular security for all z/OS resources

In high availability terms, the Parallel Sysplex concept extends System z reliability to the absolute maximum through the use of redundant hardware combined with a shared storage facility. While this clearly represents a significant additional investment in both hardware and software terms, for businesses that truly require this extra level of availability there are few other options. The key is in the shared storage facility – there are other cluster-based solutions to redundancy to protect availability in the event of one system failing, but without shared storage the workloads on those systems will be bound to suffer disruption due to the inaccessibility of 'live' data.

Closely related to Parallel Sysplex support, GDPS extends availability protection to handle not just local system failures but disaster recovery scenarios. In a GDPS configuration, machines running in geographically remote locations benefit from far more rapid recovery times through the GDPS linkage, turning recovery times from days down to minutes. In the interests of balance, it should be pointed out that Linux on System z also benefits in part from GDPS, supporting the mirroring of disk files, but this is some way from the automated level of switchover offered in a z/OS environment. The ARM facility is a software-based way of controlling how applications are restarted after a failure. This may give availability and integrity benefits, since key workloads can be prioritized and systems holding up others through locking mechanisms can be restarted in order to free up their locks.

Workload manager is a good example of the difference in approach for z/OS and Linux on System z. As an operating system, Linux was never designed with a view to managing many different workloads. The design point for Linux is that it will be running in a server that has been allocated to a particular workload. For other workloads, there will be other Linux servers. In contrast, z/OS was designed to handle IBM mainframes where there are traditionally a whole range of different workloads, such as CICS, IMS, DB2 and batch, with different classes of service and expectations. In order to manage these expectations, z/OS offers a dynamic workload manager (WLM) that effectively allows the user to define a service level contract with the operating system for each workload. Using WLM, workloads can be placed into different classes coupled with goals as to how the work should be performed. The result is better performance for high priority workloads, and a smoother overall response from the System z mainframe as a whole, yielding a closer alignment of IT with business needs.

RACF security is slightly different from the other features, because Linux on System z can use RACF when running under z/VM. However, the functionality offered by RACF in the z/OS environment is more comprehensive and able to deliver higher levels of performance. The key functional difference is that RACF under the combination of Linux on System z and z/VM manages only those resources that are relevant to this environment. So, for example, a 'user' is interpreted as a Linux server or a systems administrator, as opposed to a z/OS end user. Essentially, RACF in the z/OS environment can secure and manage z/OS resources such as CICS transactions and z/OS datasets, which are 'invisible' in the Linux on System z environment. The z/OS implementation of RACF also offers highly optimized RACF interrogation facilities which are not available in other environments. So RACF on z/OS offers the most granular security option, spanning all z/OS resources, with the most highly optimized performance.

SOA portfolio exploitation

All five of these facilities are only available in full when running z/OS on the System z. However, in terms of selecting the appropriate platform for a mainframe SOA workload, the key question is whether the different SOA infrastructure components utilize these capabilities or not. Looking at each of the facilities in turn the picture is somewhat mixed.

On the Parallel Sysplex front, while all of the SOA portfolio of products benefit from the ability to have a hot-standby hardware platform to take control if a system fails, the advantages are somewhat limited through the unavailability of in-flight data. The exceptions here are WebSphere MQ, and by inheritance WebSphere Message Broker. WebSphere MQ has the ability to support the use of the Sysplex shared storage facility to host its queues, ensuring that even if one system goes down there are no 'marooned message' – all message information remains available. As a result, for SOA workloads using WebSphere MQ or WebSphere Message Broker (which uses MQ) as the information bus across the SOA implementation, availability and integrity will be protected through the use of Parallel Sysplex. The same is true of GDPS in business continuity terms, in the event of a disaster.

Just about all of the SOA portfolio benefits from ARM capabilities. Since WebSphere MQ, WebSphere Process Server, WebSphere ESB and WebSphere Message Broker are all seen as individual subsystems, policies can be put in place for each of them. At the workload management level, exploitation is good with the ability to specify different WebSphere Message Broker execution groups to various WLM process classes, for example. As far as WebSphere ESB and WebSphere Process Server are concerned, not only do they benefit from inheriting the benefits of WebSphere Application Server's support for WLM, but also a complete transaction can be monitored and acted upon by WLM. Finally, the RACF security support is usable across the entire SOA portfolio in the z/OS environment.

So as far as these five key z/OS facilities are concerned, an SOA workload would indeed benefit from improved quality of service in terms of availability, reliability, performance, security and integrity. In a Linux on System z environment, these features would not be exploited at anything other than a basic, all-or-nothing level, although it is worth noting that if Linux is being run on the z/VM hypervisor then it will inherit at least some level of more advanced System z functionality.

Additional QoS observations

There are a few other points worth mentioning in the area of quality of service and how it reflects on platform selection. For example, on the performance side, since z/OS was designed with a view to managing multiple subsystems and workloads, it has some extremely efficient ways for different subsystems to communicate. So, when the SOA workload is running on z/OS, it can access mainframe data sources or applications like CICS or IMS with optimal performance. In contrast, if the SOA workload is running in a Linux on System z environment, then it will communicate between systems with TCP/IP which is much less efficient. Even with the use of IBM's HiperSockets facility to speed this communication up, the linkage between a Linux partition and other mainframe subsystems like CICS will be less efficient than the equivalent native z/OS command. While this may sound a small difference, SOA workloads typically involve lots of communication between different SOA components and services. If this is constantly requiring communications with other z/OS systems then this could result in lower levels of performance of the workload on Linux on System z than on z/OS.

This point is just one illustration of how the extent to which z/OS resources are used in the SOA workload in question affects the selection decision. As well as the issue of communicating between application environments, performance will also be affected by the proximity of the workload to the database systems in use. For example, DB2 on z/OS provides optimal data access performance through the use of advanced z/OS facilities. Also, when multiple application components are operating in collaboration, the resource coordination required to maintain integrity in the event of tied operations is at its most efficient in a z/OS environment.

In terms of support productivity, problem determination may be a factor in delivering the desired quality of service, particularly when multiple z/OS subsystems are involved. In a z/OS environment, these subsystems can write problem determination information such as traces and log entries to the system log, and this is something that many of the SOA infrastructure products do. This becomes particularly important when trying to resolve a problem involving how multiple subsystems are interacting. In this case, being able to understand the relative timing of activities across the affected subsystems may be crucial in identifying the problem. This multi-subsystem confusion is also evident in looking at the different approaches to product maintenance, which also affects service quality. The z/OS maintenance environment, SMP/E, makes it easier to manage maintenance involving dependencies spanning span multiple products.

All these issues combine to suggest that the level of z/OS resource usage within the SOA workload to be hosted on System z will be an important consideration in platform selection for that workload.

Cost of Ownership

While quality of service was definitely a focus area for z/OS, cost of ownership is a strong area from a Linux on System z perspective. However, while first impressions might be that Linux on System z will always be a more cost-effective platform for the SOA workload than z/OS, this is not entirely the case.

The three factors to be considered in this section as having the most relevance to cost of ownership are:

- License costs
- Skills implications
- Additional work required

License costs

As might be expected, the area of license costs comes down heavily on the side of Linux on System z. For a start, while most of the products in the IBM SOA portfolio are available on a One Time Charge (OTC) basis on both Linux on System z and z/OS, the z/OS prices tend to be considerably higher. In addition, the traditional third party software market for z/OS generally exhibits similarly high prices, while ancillary Linux tools may be available at much lower price points or even on an open source basis.

However, before looking into the license cost question in more detail, it is important to appreciate that the license side of costs is only one element of the overall cost of ownership situation. In fact, users often get blinded by the 'cost of acquisition' part of the equation without giving enough attention to the ongoing 'cost of ownership' element, which overtime will increasingly outweigh the one-time cost of acquisition. As an example, the cost of acquiring and utilizing the necessary skills over time to support the platform decision must be taken into account, as must implied costs such as the financial implications of system outages or other service level failures.

Even the license cost question is not as clear as it might seem at first glance, as a closer examination of pricing specifics shows. For example, System z speciality engines such as zAAP and zIIP represent a way to achieve considerable savings, since workloads offloaded to them do not contribute to software costs. In addition, there is a usage pricing mechanism called 'Sub-capacity pricing' that ensures users only pay for the extent of usage rather than based on the overall processing power available. Indeed, if the product usage for the SOA portfolio is small enough, as is likely when the SOA project gets started, there is yet another pricing mechanism available to reduce costs even further. On the Linux front, offerings will generally have a pricing grid based purely on the size and number of processors as might be expected from a server-based market offering. Also, it is worth checking on how development and test licensing is implemented – typically on z/OS a single license provides unlimited production, development and test usage.

Skills

Skills availability is a key area of concern in the area of cost of ownership and how it affects the platform selection decision. Obviously the best result from a cost point of view is to be able to use existing resources to service the SOA workload needs, and to be able to do this as productively as possible. However, the problems stem from the fact that the Linux and System z worlds are poles apart in terms of common understanding. System z specialists understand System z and z/OS, while Linux experts have a skills base around Linux on different distributed server platforms. Running Linux on System z brings the two worlds together, and this can cause issues.

This skills problem and its effect become immediately apparent when looking at team ownership of the SOA infrastructure and workload on System z. If the SOA workload is placed on z/OS, then the infrastructure products at least will be owned and controlled by the System z team. There may be some debate about the SOA workload itself, particularly if System z is only one component of a much wider SOA implementation, but certainly the infrastructure products and their installation, configuration and administration will be handled by the System z specialists. However, in the Linux on System z case, the problem becomes apparent. Typically, the Linux products will follow normal Linux install and administration procedures, but these will be very unfamiliar to the System z team. But any Linux specialist will find the System z environment very unfamiliar too. A further problem is that Linux on System z is at its best when run on z/VM, which throws yet another skills issue in the mix since it may be unfamiliar to the z/OS team too. Therefore, it is not clear how the ownership of these tasks should be allocated. What is clear is that there will need to be a certain amount of education and training involved, or use of expensive third-party skills with the combined skills set. Both of these options will incur additional running costs.

If additional skilled resources are needed, then this also has a bearing on the platform selection decision. Over time, the overall pool of System z skills has reduced in size, but in contrast LINUX skills are relatively easy to acquire based on the fact that new programmers often learn their trade in a LINUX environment today. As a result, acquiring additional skills will incur considerably lower costs than finding z/OS ones.

These running cost implications will probably extend into ongoing operational support of the SOA workload and its environment. Security, maintenance, problem determination and general management will all differ between the traditional z/OS environment and the Linux on System z one. The implication is that evaluating the skills and productivity impact of the platform selection decision will be important in assessing the effect on overall cost of ownership.

Additional work required

From a product point of view, installation, configuration and administration of the SOA infrastructure portfolio products will be very different in the two environments. Installing and administering products in Linux is a relatively simple affair, largely because of the assumption mentioned numerous times previously that each Linux instance is responsible for one workload. In comparison, the same tasks in a z/OS environment require a lot more effort and communication. In the z/OS case, there are stricter controls. For example, taking its normal multi-system view, a z/OS installation and configuration exercise will need to be validated by the DBA team, the security team and so on. Therefore there is more work required, which clearly has cost implications. However, the point about skills made above should not be forgotten – installing and administering the SOA infrastructure products in Linux on System z will need skills that understand Linux installation processes but can handle them in a System z environment.

Another area that may contribute additional work is integrating a Linux on System z implementation of the SOA workload with existing z/OS environment components. A clear example is that of z/OS based CICS and IMS applications, which are highly likely to be involved in any SOA workload since this allows these investments to be leveraged across the wider distributed server network. While the IBM SOA portfolio does offer specific tools to help with this task, it will be easier and more efficient if the SOA workload is also running on z/OS.

Enterprise considerations

Although quality of service and cost of ownership are the two main areas of consideration for deciding on which operating system to use for the SOA workload on System z, there are other factors based on the needs of the enterprise that should be taken into account before making any final decision. This includes the technical considerations relating to the workload itself, and areas such as enterprise policy.

Specific technical requirements may affect the platform selection decision quite dramatically. For instance, the user may have mainframe data stored in VSAM, QSAM or CICS files that is required for the SOA workload. IBM's WebSphere Message Broker for z/OS has specific processing nodes to handle the use of these specialist data sources. However, the Linux version of the product does not have these nodes, and therefore if support for these data sources is needed the user would have to build a specialized mechanism to achieve the desired access. There may also be third-party products running on z/OS that need to be involved too, and once again this will be easier if the SOA workload and infrastructure is running on z/OS too.

A big area to be considered is portability, or to be more precise deployment flexibility. The assumption behind the platform selection decision as far as this paper is concerned is that the SOA workload, or at least a portion of it, is to be placed on System z. However, it may be that this will not always be the case. Some companies may want to have the flexibility to move this SOA workload to a distributed Linux server at a later date, for example if the System z becomes overcommitted. This will be a critical factor in the decision of which environment to use on the System z.

The problem is not so much the SOA infrastructure, which will to a large extent shield applications from any differences in operating system environments, but the application content of the workload. If the SOA workload is put onto z/OS, it will naturally tend to be tuned to the z/OS environment – for example, it might start utilizing Parallel Sysplex capabilities as discussed in the section on quality of service. It may use one of the specific z/OS features like the WebSphere Message Broker node for VSAM for instance. The result is that if, at a later date, it is desired to move this workload onto a distributed Linux platform, there will be work required to achieve this, quite possibly impacting service levels in the process. In other words, with the workload running on z/OS there is a risk that the workload may become 'contaminated' with proprietary technology exploitation that does not exist outside of the System z environment. However, if the SOA workload is put onto Linux on System z, this would have the effect of keeping it isolated from these z/OS-only facilities, ensuring that it remains 'vanilla' Linux-based and hence easier to port to Linux on a distributed server at some point in the future.

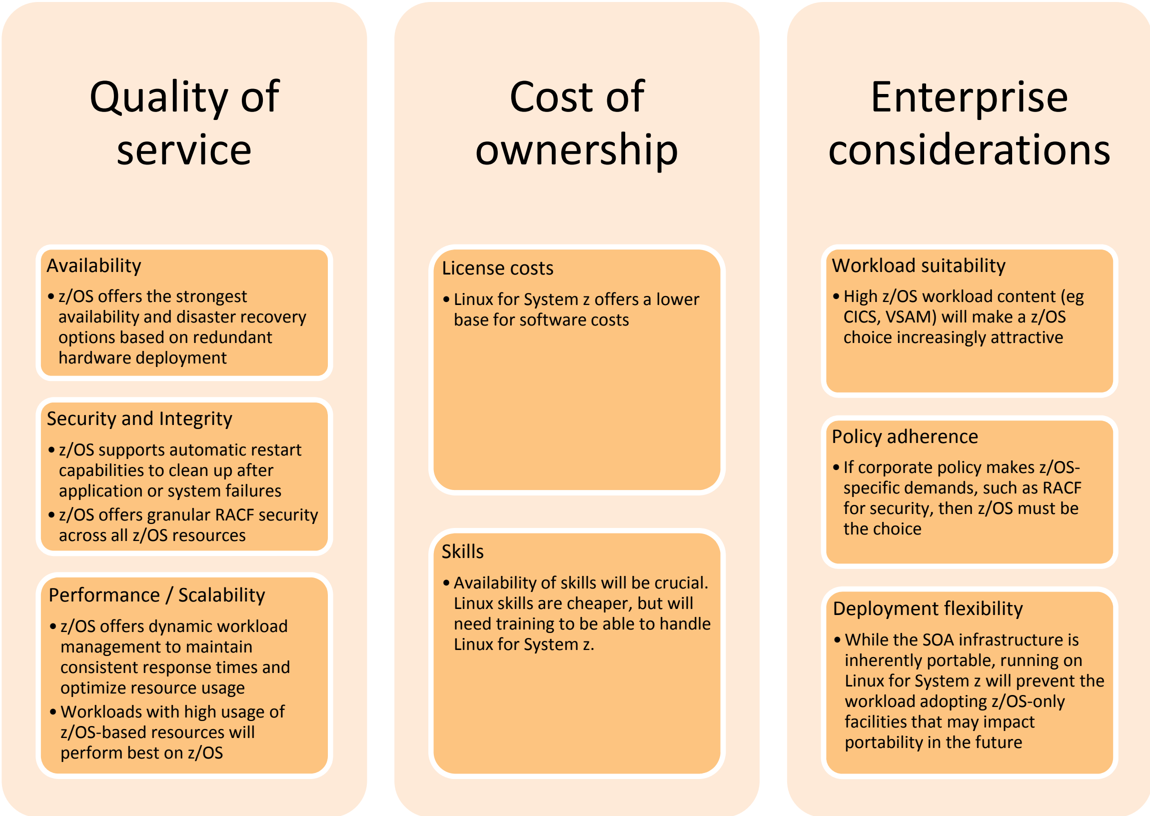

Figure 4: Selection criteria summary for SOA on System z

Sample scenarios

In order to emphasize some of the points made throughout this paper, it is helpful to look at some sample scenarios. These scenarios are highly simplified and idealized, but serve the purpose of illustrating how the platform selection decision might differ based on certain criteria.

Scenario A

Company A has a large investment in IBM mainframes, but outside of the glasshouse it has standardized on Linux across its server and desktop platforms as well as a strategic direction to move to a service-oriented architecture (SOA). However, as economic conditions have tightened the company is looking to save money by increasing utilization and optimizing the increasingly expensive resource consumptions like power and space. Linux servers are all running at well below individual capacity, and there is spare mainframe capacity available too. The decision has therefore been taken to consolidate a subset of Linux servers onto the System z platform, with SOA providing enterprise-wide connectivity, and now the platform selection decision needs to be made between moving the workload to Linux on System z or z/OS.

Factors to take into account for Company A's scenario are likely to be the availability/cost of Linux and z/OS skills, together with license cost implications. Company A also needs to think about its future plans and the likelihood of wanting to redistribute the workload at a later date. However in this scenario the KEY is that the workload move is not being done for service level reasons but for economic ones. In this case, it is highly likely the company will find Linux on System z the most appropriate platform choice.

Scenario B

Company B is similar to Company A, in that it has major System z and distributed Linux investments. However, rather than looking for economic gains from consolidation, Company B is finding that it needs much higher service levels for one part of the SOA-based workload. By delivering 'always on' availability and more predictable response times, even in an environment of sudden load spikes, Company B believes it can compete more effectively and achieve increased revenue and profit levels. But investigations have concluded that the Linux servers will be unable to deliver these service level increases, and therefore this critical part of the workload is to be moved to System z.

The System z platform selection decision is more tricky in this scenario. If it is assumed that skills are available in both z/OS and Linux disciplines, then the decision seems to hang on the new SLA requirements. As discussed previously, z/OS offers advanced capabilities such as Parallel Sysplex which are not available on Linux on System z. This may lead to a simple conclusion that if the SLA requirements are at such high levels that Linux on System z cannot deliver them, then choose, z/OS, and otherwise go with Linux on System z. However, this may well be too simplistic. While it is true to say that moving the workload to System z will enable higher service levels to be delivered regardless of the operating system selection, even if these new levels are high enough this may well not remain the case over time. Obviously this part of the business workload must be sensitive to service levels, and therefore there is the distinct possibility that further increases will be likely over time as competitors upgrade their own capabilities. This could well lead to the choice being to go with z/OS as the platform of choice in order to avoid requiring a second workload migration, from Linux on System z to z/OS, at a later date.

Scenario C

Company C has investments in both System z and distributed Linux, but the main driver for its SOA adoption is to leverage its extensive System z investments in CICS applications and VSAM data within a wider enterprise context. The company's Linux experience and skills base is relatively limited, since it only embarked on Linux as a strategic direction a couple of years ago.

The key issue here is the fact that the workload is going to be spending most of its time driving CICS applications and accessing VSAM data. While this is possible from a Linux on System z environment, perhaps by using WebSphere MQ as a communications mechanism to the z/OS environment, there will clearly be an overhead in achieving this. The most sensible option is to run the SOA workload in the closest possible proximity to the applications and data it will be using, and hence the likely choice will be to put the SOA workload on z/OS.

Summary

The preceding discussions make it clear that the decision of which platform to use for the SOA workload on System z needs to be based on many different factors. It also puts to rest some popular misconceptions, such as the belief that running Linux will 'obviously be cheaper'. While license costs may be less with the Linux on System z selection, overall costs must take into account skills availability, training requirements and additional work effort to interact with z/OS resources. Even license costs themselves deserve a closer look, since IBM has introduced a number of pricing mechanisms to bring headline prices down substantially. On the other hand, perceived advantages of running z/OS in combination with System z technology may well in reality depend on the extent to which the SOA infrastructure portfolio exploits rather than just tolerates these capabilities – for example, the shared storage offered by Parallel Sysplex to ensure seamless switchover in the event of a failure is only used by the WebSphere MQ component.

Fundamentally, however, a relatively simple guideline emerges to help users make the best decision for their own circumstances. In essence, z/OS is the home-owner of the System z house, while Linux on System z is the guest. In other words, Linux performs perfectly respectably on System z, but in a slightly 'arms-length' fashion. In contrast, z/OS takes every advantage of years of an intimate familiarity with the System z environment. Therefore, the more that the specialized capabilities of the System z environment such as the high availability Sysplex-based environment are needed, or even might be needed at some point in the future, then the more the decision will lean to using the native power of z/OS. But if the major consideration is to deliver a Linux-based enterprise solution, albeit taking advantage of the attractive resource consolidation options of System z, then Linux on System z will be the favoured approach.

About Lustratus Research

Lustratus Research Limited, founded in 2006, aims to deliver independent and unbiased analysis of global software technology trends for senior IT and business unit management, shedding light on the latest developments and best practices and interpreting them into business value and impact. Lustratus analysts include some of the top thought leaders worldwide in infrastructure software.

Lustratus offers a unique structure of materials, consisting of three categories—Insights, Reports and Research. The Insight offers concise analysis and opinion, while the Report offers more comprehensive breadth and depth. Research documents provide the results of practical investigations and experiences. Lustratus prides itself on bringing the technical and business aspects of technology and best practices together, in order to clearly address the business impacts. Each Lustratus document is graded based on its technical or business orientation, as a guide to readers.

Terms and Conditions

Lustratus Research Limited
St. David's, 5 Elsfield Way, Oxford OX2 8EW, UK
Tel: +44 (0)1865 559040

www.lustratus.com

Ref STC/LR/96497938/V1.0