Linux on IBM System z

A Strange but Powerful Pairing

White Paper
Prepared by

Kevin Gates
Linux Solutions Specialist
Sirius Computer Solutions

Table of Contents

1.0 Linux on System z: A Strange Pairing?

1.1 A Reversal of Fortune for Mainframe Computing

Seven years ago, I was taken from my comfortable world working with distributed systems, and brought into the strange and—to a degree—mysterious world of mainframe computing. I use the word 'mainframe' carefully; back then, S/390® was the hardware architecture and MVS the operating system. Coming from a distributed background and busy with the dot.com boom, the S/390 seemed antiquated and closed. Indeed, many IT magazines were declaring that the mainframe was dead.

Given all that, why did I make the move to what looked like a dying platform? The answer is pretty simple: I saw a hint of light in the darkness, a seed about to flower. In other words, I saw the potential for new life for the S/390, and that potential came in the form of Linux®.

IBM officially started supporting Linux on S/390 in 1999. At the time Linux was not mature enough for most enterprises to consider it a production platform option. However, IBM realized that by introducing Linux early on S/390 they were making more of a statement of direction than providing an enterprise class solution. Linux needed time to mature, and Linux on S/390 would need time to iron any kinks out.

And there were kinks! In the beginning, the main problem was that Linux wanted a box that provided plenty of CPU and memory. Let's just say that S/390 boxes did not fully deliver on this requirement; hence applications that could run on S/390 were limited. However, when they did run, they ran very well. Samba and Apache servers began cropping up on S/390 around the world as many early adopters took this path.

I'll not be giving the details of how Linux on System z® matured over the next 10 years; suffice it to say that as both Linux and S/390 (then zSeries®, now System z) progressed, so did the possibilities. Customers began migrating more robust applications and shared their experiences with others. The result? IBM `new workload' (as they call it) MIPS growth on System z is greater than MIPS for traditional workload. (See Figure 1)

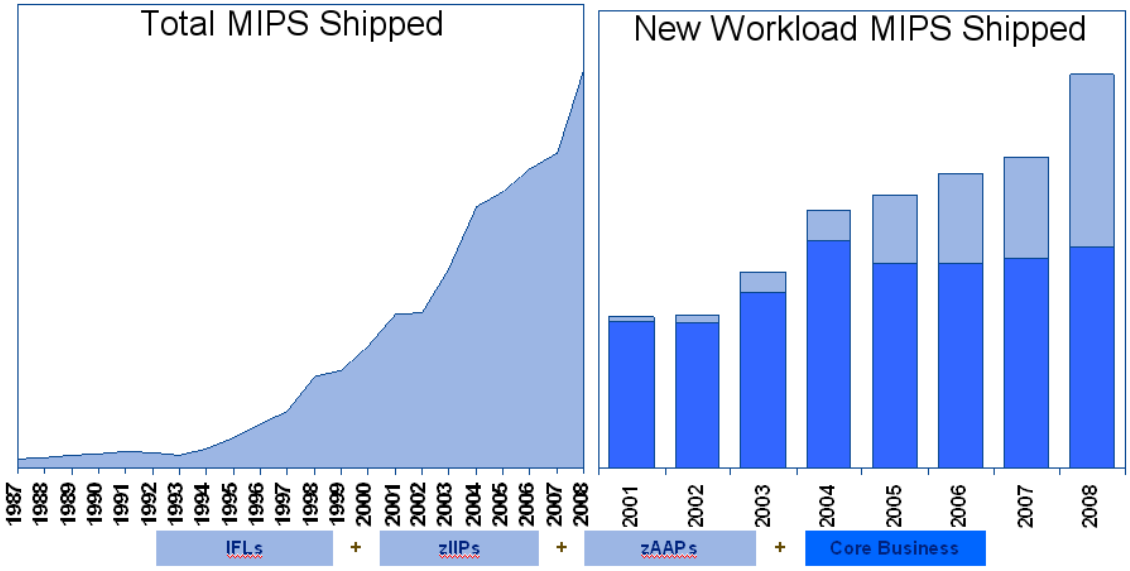

Figure 1: Courtesy of IBM

1.2 A Powerful Platform for Open Source Computing

As I expect no one will be reading this article with the expectation of running Linux on a 12 year old S/390, I will focus on the IBM System z10™. Starting with the z9® and further realized with the z10 is the ability to run significant workloads using technologies available up to 30 years ago. Virtualization, high percentage utilization of resources, unmatched reliability and advanced workload management are all part of the line of products dubbed mainframes. What z10 brings to the plate is hardware that is much faster, has bigger 'pipes,' and is highly flexible and more expandable than previous models. One example would be the new powerful CPU that is rated at 4.4 GHz on a z10 EC (Enterprise Class) system, and at 3.8 GHz on a z10 BC (Business Class) system. Another would be the 6 GBps internal network called Hipersockets. These obvious hardware advances are coupled with not-so-obvious advances such as large memory page support for efficient task swapping, a dedicated hardware decimal-point CPU, dedicated crypto cards, and system-assist processors. This combination creates a hardware environment that is literally unequalled in power and scalability. To use a technical phrase, you can consolidate like crazy on these boxes.

So if Linux on System z is so great, why hasn't it caught on more? Some would give you a drawn-out answer perhaps based on System z market share or some techno-babble. Once it is agreed to that Linux on System z is more

pervasive than what is evidenced in public testimony, I think the answer is fairly straightforward for remaining customers: the combination of Linux and System z is strange. When most people think of System z, they are probably thinking of mainframes—huge systems, filling up rooms, being attended to by old men wearing lab coats. Who would think of those dusty systems running modern applications on a new upstart operating system? Unless you are already in the world of System z, you probably haven't—the same way you have probably never considered mixing peanut butter and mayonnaise. It just doesn't make sense. Or does it? Let's find out.

First things first: the term 'mainframe' should no longer be used with regard to the z10. Mainframes are based on old, 31-bit, closed system technology. System z is 64-bit, completely open and shiny-bright new. Some call it an enterprise server, some call it z10, some System z. See how flexible System z is? You already have choices!

Now that we have that out of the way, I am going to tackle one other subject that is brought up all the time by clients. In the past, the CPU found in older models was not the fastest in the industry. For instance, the predecessor to z10, the z9, had a CPU that ran at approximately 1.6 GHz. Not exactly blazing-fast performance! People who did not understand the System z architecture thought this was a significant problem, many to the point that it closed their minds to System z. What they did not understand is that, in System z, the frequency of the CPU is not as important as it is in distributed systems. In a distributed system the CPU handles all instruction sets. In System z this is not true. Memory and I/O instructions are offloaded to other processors called system-assist processors. These SAPs remove workload from the CPU and provide efficiencies for software execution. So, although having a faster CPU can't hurt (up to a point), it is not the deal-breaker people thought it would be. At the time, applications that were less CPU-intensive and more memory- and I/O-intensive were recommended as 'best fit' applications for System z. With the advent of the new z10 4.4 GHz CPU, we can now run applications that are more CPU intensive and still take advantage of system-assist processors. Z10s provides the best of both worlds, speed of processors and optimizing their use. So again, you can consolidate like crazy.

In order to understand what Linux on System z is, you need to see what it looks like. In figure 2, I have created a simple example of what a System z box running z/OS® and Linux might look like:

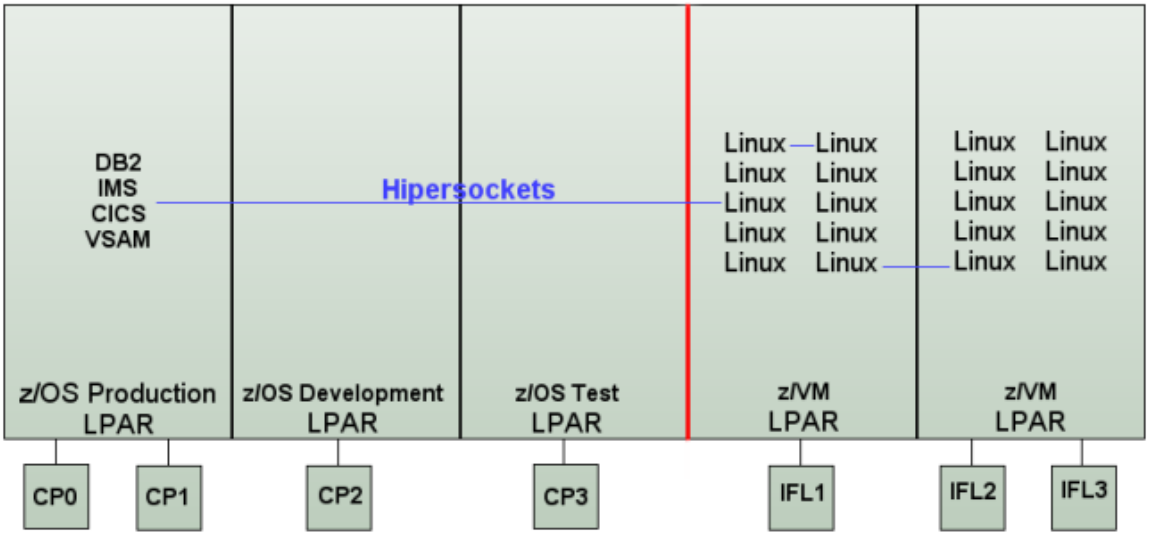

Figure 2 - System z with z/OS and Linux

In the diagram above, we see a typical z/OS environment on a System z box. z/OS is running in three logical partitions (LPARs) and has four general-purpose engines running production, development and test workloads. A red line separates the Linux environment to demonstrate how the Linux side is discrete from the z/OS side. This is important in the z world, as when you add CPUs to the box, software costs increase. For this reason, IBM created specialty engines. In the case of Linux they are called Integrated Facilities for Linux, or IFLs. IFLs run at full capacity (general purpose engines can be tuned down), and because they are specialty engines they do not effect z/OS software pricing.

z/VM® is used to create a virtualized environment. z/VM is the current version of VM (short for Virtual Machine), an IBM virtualization technology that's more than 30 years old. It is fast, efficient, rock-solid stable and highly scalable. To administer z/VM for Linux guests requires approximately 5% of the operating system. This is good news, as education requirements for staff is not extreme. Although z/VM does not have all the fancy graphics of other virtualization technologies (yet), the mature operating system offers much higher scalability

and reliability. Ask a z/VM systems programmer the last time he saw a z/VM guest go down. He will probably have to think for awhile!

The last major pieces the diagram depicts are the networking options available. Linux guests can communicate with other guests in the same LPAR or across LPARs. This is useful for clients who would like to provide failover for applications across LPARs. Additionally, a feature called HiperSockets™ allows Linux guests to communicate with traditional workload. This provides a high-speed (remember 6 GBps?) network transport that is secure (no wires to sniff!), removes network traffic in the data center, and shortens the data path between applications. For example, a WebSphere® Application Server in Linux could serve up DB2® data running on z/OS. Network options are both robust and flexible. Layer 2 and Layer 3 networking is available along with VSwitches, Guest LANs, and just about anything else you would expect from a physical network.

In the diagram we use a System z box with a z/OS environment and a Linux environment. This suggests that the System z box was already installed and Linux was added to the box. What happens if you don't have System z in house today? Fortunately, there is a great option. Take the diagram and remove everything to the left of the red line. Voila! You now have a Linux only System z box. Think of it as a robust enterprise server that runs Linux (we support both SUSE® and Red Hat®) and scales like no other platform can. The icing on the cake is that Linux-only System z boxes cost a good deal less than their z/OS hosting counterparts, making the cost case more compelling. In fact, Sirius is seeing more clients than ever before purchasing System z boxes to run Linux, even ones that previously did not have any System z deployed at all.

1.3 A Cost-Effective Solution for Enterprise Computing

Have you ever heard the phrase 'Mainframes are expensive!'? Well, that is 100% true—if you live in 1984. Luckily, we live in 2009, and the cost case for System z is vastly different from what it was back then. In fact, the Linux side of System z is different than the z/OS side. For instance, you have one-time-charge (OTC) software instead of monthly license charges. It does require some upfront costs that can make the cost of acquisition more than other

platforms. Let's face it: an Intel® processor-based server is going to cost less than an IFL. But what if that IFL can replace 5, 10. 20 or more Intel servers? What if you could consolidate software licenses along with reducing power consumption cooling requirements, floor space and administration costs? Suddenly the cost of ownership for Linux on System z versus other platforms can look very advantageous.

It is odd how Linux on System z is completely the opposite of distributed systems. On the distributed side, hardware is less expensive and software is more expensive. For Linux on System z, the hardware is more expensive and the software less. The savvy data center manager will look at how each platform can produce the best technical and cost case, and go with that platform. Sounds easy, right? Okay, maybe the real world (read: politics) will make this a bit more complex than that, but you get the idea.

Linux on System z can provide a very compelling cost case by consolidating lots of low- to medium-utilized servers. For instance, a government agency consolidated over 265 Oracle instances to just five IFLs. In engine-based Oracle terms, that reduced the number of licenses they needed from hundreds to five. Given that Oracle licenses average $27,000 per engine, the cost savings can be significant. How did they do it? They migrated production, test and development servers that did not all have to run at the same time. Because z/VM is so good at task-switching, they could swap Oracle instances out very quickly. In addition, Oracle *screams* when running on Linux on System z. One Sirius client went from a 4-way p570 running an Oracle application in 6 hours 44 minutes at 85% utilization, to running the same application on two z9 IFLs at 50% utilization in only two hours. (A z10 test ran it in 1 hour 5 minutes!) With workload execution times being reduced, more workload can be introduced into the environment.

So, the cost case for Linux on System z is the cost of ownership. I'll add one more detail. As a System z customer upgrades their model, say from a z9 with traditional engines and some IFLs to a z10, the IFLs on the z9 are transferred to the z10 at no cost. At the same time, the IFLs became more powerful. In the case of z9 to z10, the IFLs became 50% more powerful with no additional cost. Where else can you do this but with System z?

2.0 The Verdict

A Powerful Pairing. But Not Necessarily for Everyone

The purpose of this document is not to say that Linux on System z is the solution for everything. In fact, it's not. However, it has been a great platform that, because it is unfamiliar or sounds a bit odd, has not been examined closely enough by companies who might be able to take great advantage of it.

For the sake of brevity I have not covered many topics. Application availability and best fits, more about the technical side of the hardware, how administering Linux on System z is the same as on Intel or RISC, the full story of z/VM and so on. I don't think anyone is going to read a 50-page article! Suffice it to say there is a lot more where this came from. We at Sirius would be more than glad to help you determine if Linux on System z is a fit for your organization. We can assist with application assessment, cost cases, even running a proof-of-concept. We are proud to say that we keep our message even-handed. If Linux on System z is a fit, that's great. If it's not, we'll be the first to tell you. We can provide client examples, white papers or technical documentation if you would like and run IBM provided tools to provide cost justification.

Linux running on System z might sound like a strange pairing, but the fact is they are a powerful pairing. The platform is no longer on the bleeding edge. There are great stories out there from the financial institutions, government agencies, utility companies, and everything in between. If it makes sense for them, maybe it will make sense for you, too!

About the author:

Kevin Gates is a Linux Solutions Specialist at Sirius. For the past twenty six years, he has provided employers and clients quite the blend of programming, systems architecture and project management skills on both distributed and centralized systems. Coupling the technical experience with management experience has proven to be useful for working with multiple clients simultaneously. Additionally, understanding how to balance the needs of both technical and managerial staff has led to success in both cooperative environments and where political issues have outweighed technical issues.

Kevin has participated as a speaker in several IBM and Linux oriented events including LinuxWorld, IBM PartnerWorld and various IBM seminars.

Kevin's experience/skills include:

- Member on the IBM Advisory Council for Linux on zSeries
- Member on the Lotus Advisory Councils for Lotus Notes, eSuite, SmartSuite
- Project Manager for many internal applications, intranet and internet web applications
- Architect / Developer or WebSphere 3.x Solutions
- Architect / Developer net.Commerce and NCHS Solutions
- Programming – Cobol, C, Visual Basic
- Administration – Windows NT / Win2k, Linux

About Sirius:

Sirius Computer Solutions is an IBM Premier Business Partner providing advanced infrastructure solutions to clients across the U.S. Sirius is a nationally recognized solution provider with a certified team of sales and technical professionals who have the skills, product knowledge and commitment to help clients develop and implement the right solutions to solve their business needs. With a nationwide consulting, sales and services organization, Sirius provides best-of-breed technologies from across the full spectrum of information technology, including hardware, software, storage, networking, security and voice. Sirius backs it up with post-sale support that ensures you get the maximum benefit and value from your investment. More information can be found at www.siriuscom.com or by calling 800-460-1237.

September 1, 2009