

**POWER5 Architecture**

**Unix-Linux Solutions LLC**

*By Ken Milberg*

**Overview ---- POWER5 architecture**

As we all already know, computer operating systems communicate with the hardware that eventually allows functional programs to run on computer systems. AIX (Advanced Interactive eXecutive) is IBM's home grown Unix Operating System. Unix has evolved into one of the most successful operating systems to date, since its introduction in 1969. AIX was developed in the mid 70s, and in recent years with the introduction of POWER5 architecture at IBM, the combination of both has evolved what is commonly referred to as the midrange into mainframe scale capabilities (reliability, virtualization and performance). IBM has made substantial improvements throughout the years on their IBM proprietary RISC based hardware, where additional mainframe-type components are actually needed today to utilize the new architecture. Systems like the HMC (hardware management console) and the Hypervisor (software which runs on hardware machines and manages one or more operating systems) are important elements of the p5 architecture and are necessary to build systems and take advantage of systems such as Advanced Power Virtualization (APV) on the midrange.

Though most companies are still running AIX on the p5 platform, perhaps the most impressive part of the POWER5 architecture is that it has been optimized to run Linux. Unlike other RISC based hardware, IBM has fully implemented most of the functionality of the p5 into its Linux support. This is largely due to the recent developments of the Linux 2.6 kernel, which has brought Linux into the Enterprise. IBM added its own code to SUSE and Red Hat variants to provide support to the POWER5 architecture.

## **Table of Contents**

---

Abstract .....	2
Table of Contents .....	3
Introduction.....	4
Power History.....	5
POWER5.....	7
SMT.....	10
HYPERVISOR.....	12
Advanced Power Virtualization.....	17
AIX.....	19
Linux.....	21
Linux on Power.....	24
Summary and Conclusion.....	26
References .....	28

## Introduction

The Power5 processor is the latest implementation of the PowerPC AS Architecture (IBM DeveloperWorks, 2005). It consists of a dual core processor with simultaneous multi-threading capabilities across the new architectural design of its core. What sets apart the technology from others, are capabilities that this architecture provides, as well as the operating systems that are supported by this new architecture. Advanced Power Virtualization might be its most important asset. Unlike its predecessor, the POWER4 (sometimes referred to by its codename Regatta), p5 servers have strong virtualization capabilities. The IBM virtualization engine is comprised of an entire suite of services and forms the key element of IBM's on-demand computing model (APV on IBM System p5, 2004). It lets users take advantage of capabilities such:

- Resource sharing
- Shared Ethernet and Virtual SCSI
- Workload management (Partition Load Manager)
- Micro partitioning (APV on IBM System p5, 2004).

The p5 architecture provides support for both Unix and Linux operating systems. The Unix operating system is IBM's propriety Unix system, AIX, and the Linux operating systems are either Red Hat or SUSE's. Recent versions that contain the 2.6 kernel need to be installed to take full advantage of the capabilities of the system.

### **Power History**

The Power architecture stands for *Power Optimization with Enhanced Risc* and is the processor used by many IBM Servers today (Power to the People, 2005). It is a descendent from the 801 CPU and is a 2<sup>nd</sup> generation RISC processor. It was introduced in 1990 to support Unix RS6000 Systems. Power architectures incorporated characteristics which were common in most RISC architectures. The instructions were fixed length (4 bytes) and had consistent formats. The architecture provided a set of floating point registers for floating point computation. Basically, all computations retrieved source operands from one register set and placed the results in the same set. The instructions themselves performed one basic operation. What made the architecture unique among existing RISC architectures, was that it was functionally partitioned, which separated the functions of program flow control, fixed point computation and floating point computation (Power to the People, 2005). The objective of most RISC architectures were to be extremely simple so that implementations would have an extremely short cycle time. This would result in processors that could execute instructions at the fastest possible clock rate. The Power architecture designers chose to minimize the total time spent to complete a task. This time was a bi-product of three different components. They were; the path length, number of cycles needed to complete an instruction and its cycle time (Power to the People, 2005).

During the early 90's, there were 5 different RISC architectures that were actively competing with one another.

IBM partnered with Apple and Motorola to come up with a common architecture, which would meet the standards of the alliance (A High-Performance Architecture with a History, 2006). Its first design was very simple and all instructions were completed in one clock cycle. It lacked floating point and parallel processing ability. The Power architecture was an attempt to correct this flaw. It consisted of over 100 instructions and was known as a complex RISC system. The Power1 chip consisted of 800,000 transistors per chip and was functional partitioned. It had separate floating point registers and could scale from the low to the high end workstations. The first chip actually had several chips on one single motherboard, but was refined to one RISC chip with more than 1 million transistors. It was used as the CPU for the Mars Pathfinder mission (Power to the People, 2005).

The Power2 chip was released in 1993 and was their standard-bearer for almost 5 years. It contained 15 million transistors per chip. It also added a second floating point unit (FPU) and extra cache. This chip was known for powering the IBM Deep Blue supercomputer that would beat Garry Kasparov at Chess in 1997 (Power to the People, 2005).

The Power3 Architecture was the first 64-bit symmetric multiprocessor. It was designed to work on both scientific and technical computer applications. It included a data prefetch engine, dual floating point execution units and non-blocked interleaved data cached. It used copper interconnect, which delivered double the performance at the same price (Power to the People, 2005).

The Power4 architecture was released in 2001 with 174 million transistors per processor. It incorporated micron copper and silicon based technology.

When introduced, it was the most powerful chip on the market. It also inherited the characteristics of the POWER3 design, with the reinvention of the new design. Each processor had 64-bit 1 GHz PowerPC cores and could execute as many as 200 instructions simultaneously. It became the driving force behind the IBM Regatta Servers, which were the first Unix servers that could deliver logical partitioning capabilities, previously only available on the mainframe. Logical partitioning enables one to configure logical hosts on one server, each with its own kernel and operating system (Power to the People, 2005).

### **POWER5**

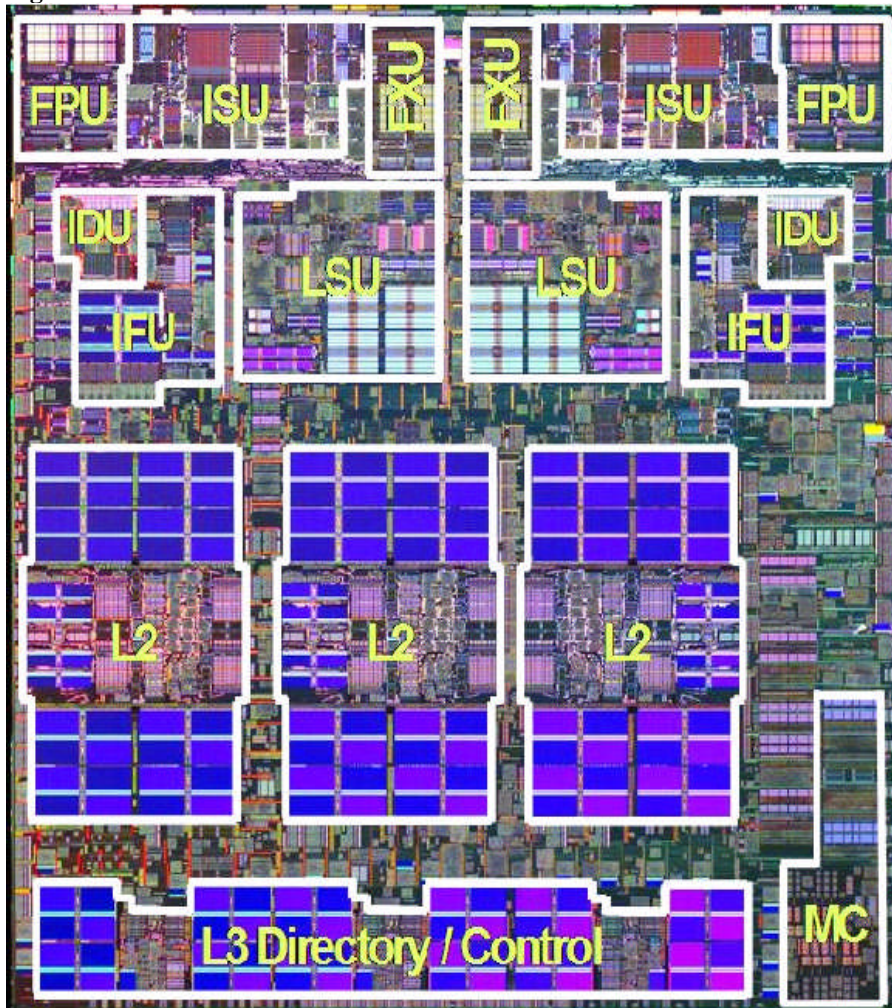
The design objectives of the p5 technology were:

- To maintain binary and structural compatibility with older POWER4 systems
- Enhance and extend SMP scalability
- Improve performance
- Provide additional server flexibility
- Improve power-efficiency
- Enhance reliability and availability.

The Power 5 architecture, introduced in 2003, contained 276 million transistors per processor (Power to the People, 2005). It was based on the 130 nanometer copper/SOI Process and featured chip multiprocessing, a larger cache, a memory controller on the chip, simultaneous multi-threading (SMT), advanced power management and improved hypervisor technology.

The Power5 was built to allow up to 256 logical partitions and was available on both its pSeries and iSeries servers. The processor itself exhibits a superscalar inner organization, which contains an aggressive branch prediction, out-of-order issues, register renaming, a large number of instructions in flight and fast selective flush of incorrect speculative fetched instructions and results. Each POWER5 core is designed to support SMT and single threaded modes. The software (the Hypervisor) switches the processor from SMT to single threaded mode.

Figure 1



*FXU refers to the fixed point integer unit  
ISU – Instruction Sequencing Unit  
LSU Load Store Unit  
L2 Level 2 cache  
MC Memory Controller.*

Figure 2

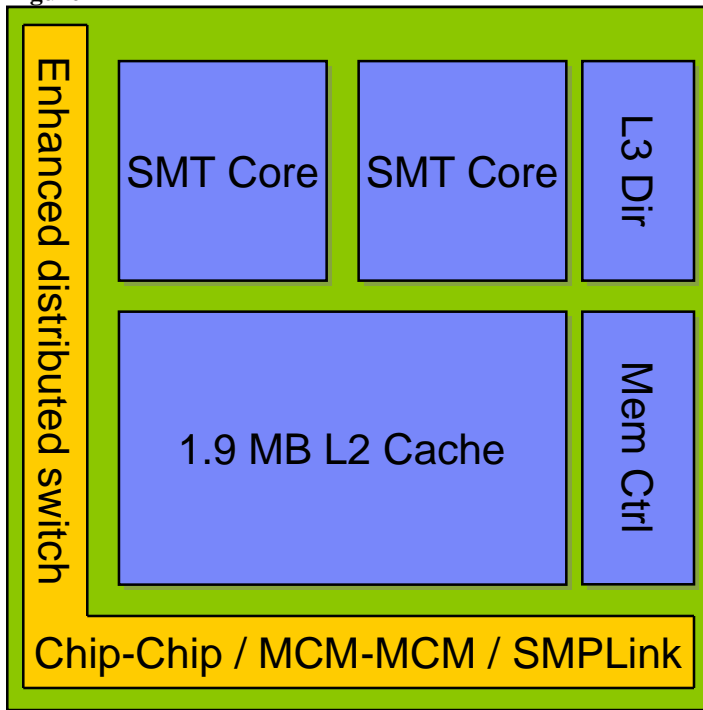


Figure 2 more clearly illustrates the interrelationships of the Chip and SMT. The most powerful improvements on the p5 are:

- Enhanced memory subsystem
  - Improved L1 cache design
    - 2-way set associative i-cache
    - 4-way set associative d-cache
    - New replacement algorithm (LRU vs. FIFO)
  - Larger L2 cache
    - 1.9 MB, 10-way set associative
  - Improved L3 cache design
    - 36 MB, 12-way set associative
    - L3 on the processor side of the fabric
    - Satisfies L2 cache misses more frequently

- Avoids traffic on the inter-chip fabric
- On-chip L3 directory and memory controller
  - L3 directory on the chip reduces off-chip delays after an L2 miss
  - Reduced memory latencies
- Improved pre-fetch algorithms
- Improved performance
- Simultaneous Multi-Threading
- Hardware support for Shared Processor Partitions --Micro-Partitioning (Advanced POWER Virtualization on IBM System p5, 2004).

### **SMT**

Because of its dual-core design and support for simultaneous multithreading, one POWER5 chip actually appears as a four-way microprocessor to the operating system. Processors using SMT, can issue multiple instructions from different code paths in one single cycle. Multiple instructions from both hardware threads can be issues from one processor cycle on the p5. The way it works, is that the core uses two separate Instruction Fetch Address Registers, which stores the program counter for the 2 threads from the programs. They are fetched every alternative cycle for each hardware thread. In a single threaded mode, the instructions are fetched from the active thread cycle and the program counter corresponding to the actual hardware thread is uses. The two threads essentially share the instruction cache and the instruction address translation facility. This is the POWER5 concept of Instruction fetching (APV on p5Servers: Architecture and Performance Considerations, 2005).

The POWER5 simultaneous multithreading implementation is an extension to the eight instruction pipeline superscalar POWER4 design. When the chip is in SMT mode, instructions from either thread can use the 8 instruction pipelines on any given clock cycle. The POWER5 can essentially duplicate portions of logic in the instruction pipeline by increasing the capacity of the register rename pool. This allows the processor to execute two instruction streams or threads concurrently. It also features dynamic resource balancing and adjustable thread priorities for more efficient utilization of the threads. Each core appears to the operating system as a two way SMP (symmetric multiprocessor). They are both supported as a separate logical processor by AIX V5.3. In this sense a partition defined with one logical processor is a logical two way by default. Each hardware thread is supported as a separate logical processor by AIX 5L V5.3. So, a dedicated partition that is created with one physical processor is configured by AIX 5L V5.3 as a logical two-way by default. This is independent of the partition type, so a shared partition with two virtual processors is configured by AIX 5L V5.3 as a logical four-way by default. When simultaneous multithreading is disabled, at least half of the logical processors will be offline (APV on p5Servers: Architecture and Performance Considerations, 2005).

Characteristics of the POWER5 simultaneous multithreading implementation are as follows:

- Eight priority levels for each thread that can be raised or lowered by the POWER Hypervisor, operating system, or application.

- Processor resources optimized for best simultaneous multithreading performance, providing the ability to reduce priority of a thread that is consuming maximum resources or hold decode of a thread with long latency events
- Dynamic feedback of shared resources, enabling balanced thread execution
- Software-controlled thread priority
- Dynamic thread switching capabilities (APV on p5Servers: Architecture and Performance Considerations, 2005).

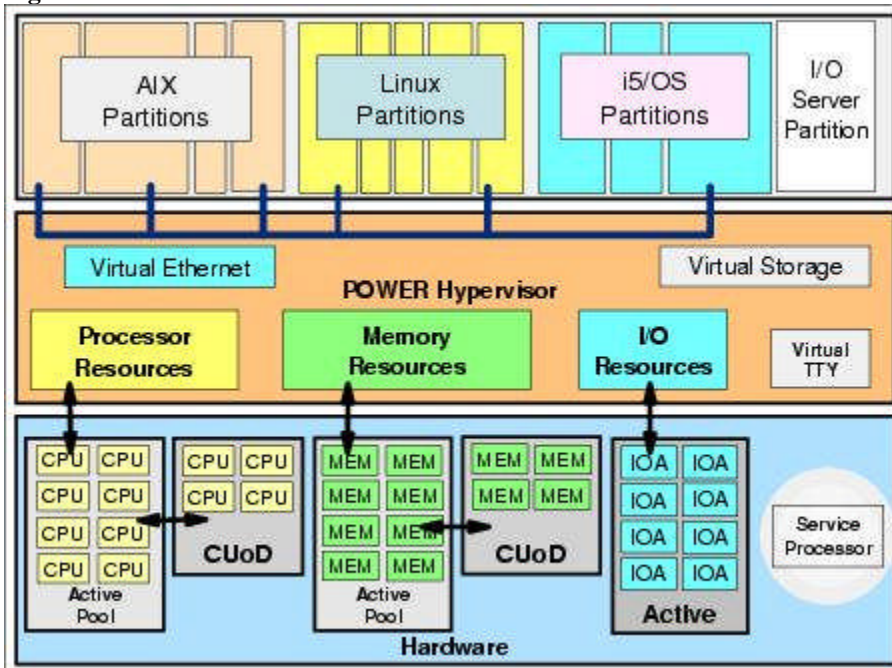
IBM has estimated the performance benefit of simultaneous multithreading at 30% for commercial transaction processing workloads. Not all applications benefit from simultaneous multithreading. Having two threads executing on the same processor will not increase the performance of applications with execution-unit-limited performance or applications that consume all of the processor's memory bandwidth. For this reason, the POWER5 supports single-threaded execution mode. In this mode, the POWER5 gives all physical resources to the active thread, enabling it to achieve higher performance than a POWER4 system at equivalent frequencies. In single-threaded mode, the POWER5 uses only one instruction fetch address register and fetches instructions for one thread every cycle (APV on p5Servers: Architecture and Performance Considerations, 2005).

### **HYPERVISOR**

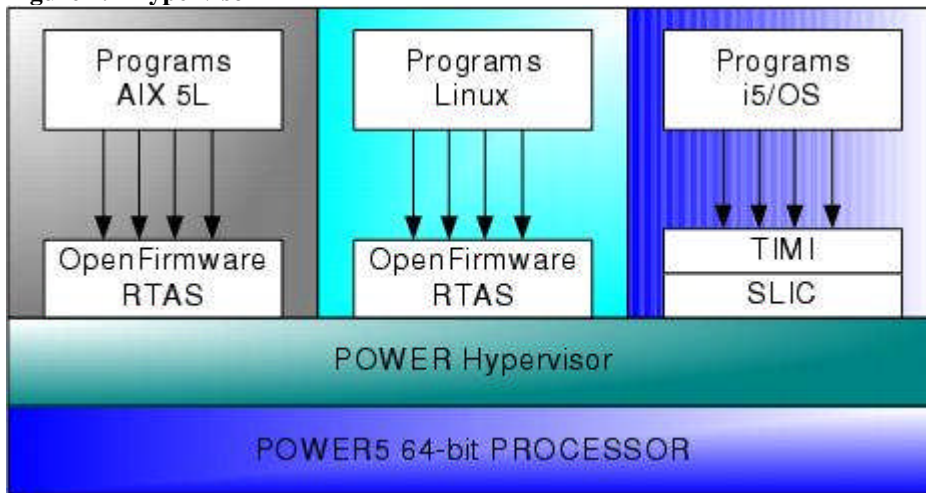
The technology behind the virtualization of IBM eServer p5 systems is provided by a piece of firmware known as the POWER Hypervisor. The Hypervisor supports partitioning and dynamic resource movement across multiple operating systems.

It is the foundation of the IBM virtualization engine, which is implemented as part of the POWER5 architecture. The POWER Hypervisor supports many advanced functions when compared to the previous version found in POWER4 processor-based systems. This includes sharing of processors, virtual I/O, and high-speed communications among partitions using a virtual LAN, and it enables multiple operating systems to run on the single system, including AIX, Linux, and i5/OS operating systems (APV on p5Servers: Architecture and Performance Considerations, 2005). With support for dynamic resource movement across multiple environments, clients can move processors, memory, and I/O between partitions on the system as workloads are moved between the partitions.

**Figure 3**



As one can see by the illustration above, the hypervisor stands between the operating systems and the hardware. It is the system that provides for Advanced Power Virtualization.

**Figure 4.1 Hypervisor**

As we examine the architecture more closely, layers above the POWER Hypervisor are similar but the contents are characterized by each operating system. The layers of code supporting AIX 5L and Linux consist of system firmware and Run-Time Abstraction Services (RTAS). System firmware is composed of low level firmware type code that perform server unique input/output (I/O) configurations and the Open Firmware that contains the boot-time drivers, boot manager, and the device drivers required to initialize the PCI adapters and attached devices. RTAS consists of code that supply platform-dependent accesses (APV on p5Servers: Architecture and Performance Considerations, 2005). They can be called from the operating system. These calls are all passed to the POWER Hypervisor that handle all I/O interrupts.

Open Firmware and RTAS are both platform-specific firmware and both are tailored by the platform developer to manipulate the specific platform hardware. RTAS encapsulates some of the machine-dependent operations of the IBM eServer p5 systems into a machine-independent package. For example, AIX can call RTAS to do things such as start and stop processors in an SMP configuration, display status indicators (I.E.

LEDs), and read/write NVRAM without having to know the intricate details of how the low-level functions are implemented. On the other hand, Open Firmware, does not have to be present when the operating system is running. It is a specification for machine-independent BIOS that are capable of probing and initializing devices that have IEEE-1275 compliant Forth code in their ROMs. The device tree produced by Open Firmware is passed to the operating system when control is passed to the operating system during boot (APV on p5Servers: Architecture and Performance Considerations, 2005). The POWER Hypervisor uses some system processor and memory resources, a small percentage of overhead. These resources are associated with virtual memory management (VMM), the POWER Hypervisor dispatcher, virtual processor data structures (including save areas for virtual processor), and for queuing of interrupts. The impact on performance should be minor for most workloads, but the impact increases with extensive amounts of page-mapping activity. Partitioning may actually help performance in some cases for applications that do not scale well on large SMP systems by enforcing strong separation between workloads running in the separate partitions. The POWER4 processor introduced support for logical partitioning with a new privileged processor state called POWER Hypervisor mode. It is accessed using POWER Hypervisor calls, which are generated by the operating system's kernel running in a partition. POWER Hypervisor mode allows for a secure mode of operation that is required for various system functions where logical partition integrity and security are required. The POWER Hypervisor validates that the partition has ownership of the resources it is attempting to access, such as processor, memory, and I/O, then completes the function.

This mechanism allows for complete isolation of partition resources (APV on p5Servers: Architecture and Performance Considerations, 2005).

In the POWER5 processor, further design enhancements are introduced that enable the sharing of processors by multiple partitions. The POWER Hypervisor Decrementer (HDEC) is a new hardware facility in the POWER5 design that is programmed to provide the POWER Hypervisor with a timed interrupt independent of partition activity. The HDEC is described in POWER Hypervisor Decrementer. HDEC interrupts are routed directly to the POWER Hypervisor, and use only POWER Hypervisor resources to capture state information from the partition. The HDEC is used for fine-grained dispatching of multiple partitions on shared processors.

It also provides a means for the POWER Hypervisor to dispatch physical processor resources for its own execution. The POWER5 processor supports special machine instructions and are exclusively used by the POWER Hypervisor. If an operating system instance in a partition requires access to hardware, it first invokes the POWER Hypervisor by using POWER Hypervisor calls. The POWER Hypervisor allows privileged access to the operating system for dedicated hardware facilities and includes protection for those facilities in the processor and memory locations (APV on p5Servers: Architecture and Performance Considerations, 2005).

### *Advanced Power Virtualization*

Advanced Power Virtualization (APV) is a combination of hardware and software that supports and manages the virtual I/O environment on POWER5 systems.

It provides for the following technology:

- Micro-Partitioning
- SMT multi-threading support
- Virtual SCSI Server
- Shared Ethernet
- Partition Load Manager (Power to the People, 2005).

The key ingredients to making it work are optimized operating systems (AIX 5.3 and Linux supported versions, including Red Hat and SUSE that include support for the 2.6 kernel).

Micro partitioning, is mainframe inspired technology, which just arrived on the midrange with the introduction of POWER5 systems. It allows one to virtualize CPUs shared by multiple partitions. One CPU can be split into 10 logical partitions, each of which can be given as low as 1/10 of a logical partition. This allows for a finer grained resource allocation, as well as higher resource utilization within the managed system. Historically, Unix systems are seen as being CPU bound if they are more than 50% utilized. The POWER5 technology allows one to actually utilize all the resources in your environment and not worry about partitions which are 90% busy.

One can even *uncap* their partition to allow for other partitions to make use of their shared partition resources (Power to the People, 2005). This allows for logical partitions to utilize even more than their entitled capacity, when the workload gets heavier.

Virtual I/O (VIO) provides the capability for a single I/O adapter to be used by multiple logical partitions on the same frame (managed system). This enables consolidation of I/O resources and minimizes the number of required I/O adapters. Essentially, the use of VIO also provides a more economic I/O model by using physical resources more efficiently through sharing (Power to the People, 2005). With each partition typically requiring one I/O slot for disk attachment and another one for network attachment, this puts a constraint on the number of partitions. To overcome these physical limitations, I/O resources can be shared. Virtual SCSI provides the means to do this for SCSI storage devices. Furthermore, VIO enables attachment of previously unsupported storage solutions. As long as the VIO supports the attachment of a storage resource, any client partition can access this storage by using virtual SCSI adapters. Typically, a small operating system instance needs at least one slot for a Network Interface Connector (NIC) and one slot for a disk adapter (SCSI, Fiber Channel, and so on), but more robust configurations often consist of two redundant NIC adapters and two disk adapters. Virtual I/O devices are intended as a complement to physical I/O adapters (also known as dedicated or local I/O devices). A partition can have any combination of local and virtual I/O adapters. The IBM VIO Server is the link between the virtual resources and physical resources. It is a specialized partition that owns the physical I/O resources, and is supported only on POWER5 processor-based servers. This server runs in a special partition that cannot be used for execution of application code. It mainly provides two

functions, serving SCSI devices to client partition and also shared Ethernet adapters for VLANs. When implementing VIO Servers, it is recommended to have at least 2 on a frame, so that there is redundancy in case the VIO Server goes down.

Partition Load Manager (PLM) is designed to automate the administration of CPU and memory across logical partitions within a managed systems. PLM will automate the migration of resources based on partition load and priorities. High demand partitions will resource more resources. User defined policies govern how the resources are moved around. PLM is not supported with Linux, only on AIX. It should be noted that many of the features of PLM can already be provided automatically, by uncapping logical partitions and taking advantage of unused clock cycles (Power to the People, 2005).

SMT, as we discussed previously, is a POWER5 enhancement that allows multiple threads to execute concurrently on a single processor. It requires either AIX 5.3 or an enabled version of Linux, and can lead to approximately a 30 percent improvement in throughput.

## AIX

AIX (*Advanced Interactive eXecutive*) is a Unix based IBM proprietary operating system, introduced in 1986. It is based on Unix System V, but also has BSD roots as well. The roots of Unix go as far back as the 1960s. This is when AT&T's Bell Labs partnered with MIT and GE to develop a multi-user operating systems called Multics. Dennis Ritchie and Ken Thompson worked on this project until AT&T withdrew from it.

In their spare time, Ken and Dennis played with a game called Space Travel, which ran under Multics. Because they could no longer play this game on the system they were using, they decided to port the game to another system running on a DEC PDP-7 computer. They would need to create another operating system for this game, which they did, and named it Unics, which they wrote in C. People started to hear about both the new OS and the game, and wanted both. Unix would start to get distributed to universities and eventually grow into one of the most important operating systems (McCarty,2000). IBM ported AIX to the RS6000 platform of products in 1989 and since then it has served as the operating system of choice for the IBM platform. The release of AIX version 3 coincided with the announcement of the first RS/6000 models. At the time, it was considered unique, in that it not only outperformed all other machines in integer compute performance, but also beat the competition by a factor of 10 in floating-point performance (Wikipedia, n.d.). It was the first operating system to introduce the idea of a journaling file system, JFS, which allowed for fast boot times by avoiding the need to perform file system checking (*fsck*) for disks on every reboot. Another innovation was the introduction of shared libraries, which avoided the need for an applications to statically link to libraries that it used. The resulting smaller binaries used less of the hardware RAM, to run, and used less of the disk space to install. AIX also has a strong built in Logical Volume Manager (LVM) – which helps to partition and administrate groups of disks . LVM is flexible, powerful, and easy to use and has been borrowed by most of the other hardware vendors in some fashion. The latest update to AIX 5L (AIX 5.3) provides innovative new features for security, mainframe-inspired reliability, systems management, distributed filesystems and virtualization

It also includes a host of other enhancements while maintaining full binary compatibility with previous releases of AIX 5L. Most importantly, AIX 5.3 fully supports the advanced virtualization capabilities of the POWER5 architecture. It can support up to 64 CPUs and two terabytes of RAM (Wikipedia, n.d.).

The JFS2 filesystem, introduced by IBM as part of AIX, supports files and partitions up to 16 TB in size. It is important to note that while AIX 5.2 will run on POWER5 systems, it does not contain the enhancements necessary to run any of the virtualization features that AIX 5.3 allows.

### Linux

The Linux operating system is a multi-user, multi-tasking operating system that runs on many platforms. This includes PCs, mid-range servers and mainframes, though it was initially developed on PCs, for PCs. Linux has its roots in Unix, and its inventor, Linus Torvalds, is very quick to give credit to the early inventors of Unix and C, Brian Kernighan, Dennis Ritchie and Ken Thompson. (Rooney, 2004). Recently voted the #1 top executive of the year by CRN magazine, 35 yr old Linus Torvalds has without a doubt left a huge mark in the industry.

To understand how Linux really came to be, we need to explore in more detail where it came from. As stated previously, it has its roots in Unix. In the early 1980's, AT&T started to recognize the true value of Unix (that it could make money from it). They started selling license fees that were substantial, and allowed others to sell it. Many people felt that they contributed code to Unix and that AT&T had stolen their contributions to make money.

An MIT researcher named Richard Stallman launched the GNU (GNU is not Unix) project, which was established to create a Unix-like operating system, which could be openly distributed, free of charge. The Free Software federation was created in 1984 (FSF), which supported GNU. The GNU project helped establish the GNU Public License (GPL), which is a form of copyright, which allows the user rights to use, study, copy and otherwise distribute software freely. Before AT&T started selling licensing fees to Unix, Universities were able to use Unix as a vehicle for teaching students computer science. Since Universities could no longer freely use Unix for this purpose, they needed something else. Andrew Tannenbaum created a Unix like system, called Minix, which became a teaching tool (McCarty,2000). While it was a good teaching tool, it was not a product that performed well. In 1990, Linus Torvalds, at the age of 20, started working on a memory manager for PCs. An astute follower of the open source movement, he would start to lose his patience waiting for the Minix kernel, which would never come in the form it was supposed to. So he started to build on his work, as he thought it could operate as a Unix kernel. He called his code, Linux, because it was his version of Minix and he asked the developer community for help in building on his kernel. Stallman's GNU project would ultimately use Linus' kernel and a marriage would be built. Four years later, Linux 1.0 was released, with a built-in user base of 100,000 people. Linux may be one of the first real innovations that was made without thinking about how much money could be made from it. It was all about the technology and not about the selling or marketing.

From an OS perspective, traditionally operating systems were developed in a very highly structured way, with carefully engineered sub-projects which involved much analysis, design, and many stages of testing and debugging, filled with highly

complicated source code control mechanisms. Micro-management was the way to do this kind of development. The Linux model, followed a completely different path, which became known as the open source model. With the help of the movement, an entire operating system based upon Unix, was essentially developed from scratch, using this method.

Far from reaching its maturity in 2005, it arguably had just started to reach its maturity stage in the product life-cycle, with the introduction of version 2.6 of its kernel. Traditionally used only in the back rooms by Internet & web hosting companies, Linux has evolved today where it is used by Fortune 100 companies in the most complex infrastructure areas, including firewalls, mainframes and cluster based hardware solutions. Corporations are even using Linux to run their mission critical financial and ERP (Enterprise Resource Applications) applications. The latest release of the kernel, which has been branded an Enterprise ready release, contains major improvements in scalability, manageability and performance. Some of the important innovations include a new scheduler, kernel preemption, improved threading models and support for NPTL, VM changes, memory management changes, workqueue interface, interrupt routine changes and also support for 16 way processors (Santhanam, 2003). At this time it's a viable alternative operating system that poses a serious challenge in the marketplace, to competitors such as Unix and Windows. Both government and big business are moving to Linux, and one would be hard pressed to find any organization without a real Linux strategy. Today, it now powers everything from the smallest of handheld devices to S/390s.

Unlike Unix, anyone can distribute Linux, and one is not held hostage by large corporate vendors that rule that Unix world, such as IBM, HP and Sun. Furthermore, Linux is multi-platform, and can run on virtually any type of hardware.

### **Linux on Power**

Linux is also supported on the POWER5 architecture. IBM has invested over a billion dollars in Linux through the years, and unlike other hardware companies, have made their Linux strategy an important part of their offerings. Virtually all the capabilities of the POWER5 Architecture are extended to Linux. Even their HMC runs Linux. The HMC is a necessary part of the POWER5 Server architecture. It is a pre-installed Linux workstation, which is used as a local console to configure and administrate the partitioned environment.

But how does Linux play into p5 architecture? Why install Linux, rather than AIX, a more mature, robust operating system on your POWER5 partition. One might want to install Linux on a pSeries server for any of the following reasons:

- You have existing IBM Unix hardware and a large Linux PC based infrastructure, and are looking at consolidation. pSeries servers let you create Unix, Linux or even AS/400 logical partitions (LPARs), and IBM offers considerable Linux support options that may benefit large IBM shops.
- You need massive horsepower to run your Linux applications. With the 2.6 kernel, Linux can run on a 32-way system.

- You want to run Linux with minimal resources and do not wish to purchase hardware that will rarely be utilized. With the micro-partitioning abilities of Advanced Power Virtualization (APV), you can assign as little as a tenth of a CPU to an LPAR.

As with AIX, one can run multiple Linux installations simultaneously on a single eServer p5 server's LPAR. You can even share Linux and Unix partitions on the same CPU with APV's micro-partitioning features. Shared Ethernet, an APV feature that lets you use virtual adapters on your partitions using VIO servers, further lowers total cost of ownership (TCO) by removing the need for dedicated adapters in environments that may not need a lot of network bandwidth. The following table compares the Virtualization capabilities of AIX vs. Linux, and whether its release is supported by the functionality.

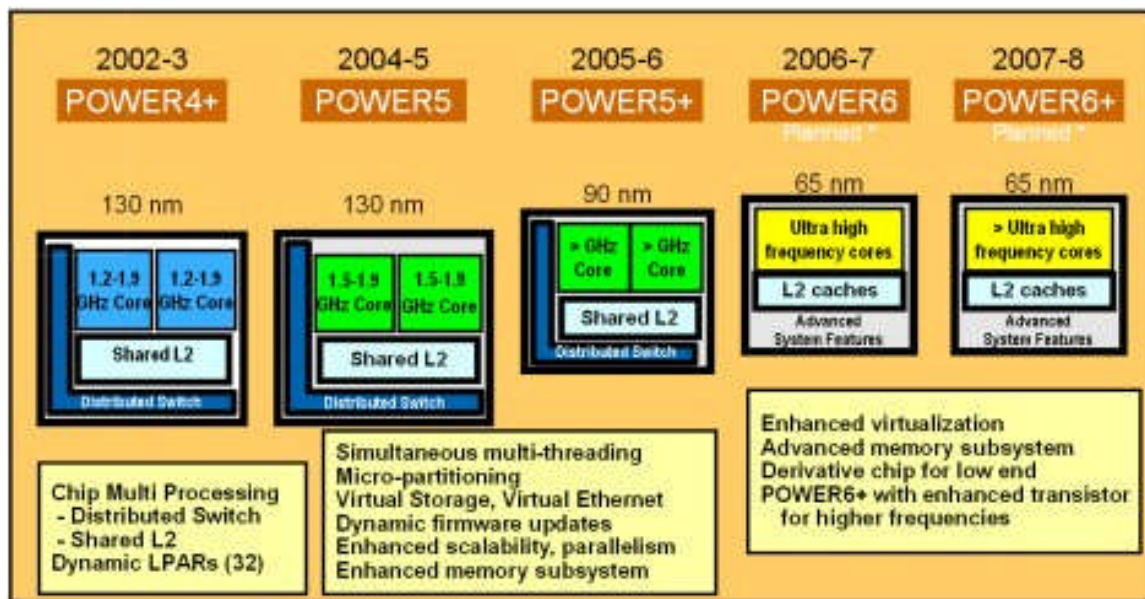
Function	AIX 5.2F	AIX 5.3	Linux SLES9	Linux RHEL AS4
DLPAR				
Processor	Y	Y	Y	Y
Memory	Y	Y	N	N
I/O	Y	Y	Y	Y
Virtual LAN	N	Y	Y	Y
Micro-partitions	N	Y	Y	Y
Virtual Storage	N	Y	Y	Y
Virtual Ethernet	N	Y	Y	Y
Partition Load Manager	Y	Y	N	N

## **Summary and Conclusion.**

The POWER5 Architecture clearly is one of the most powerful midrange systems available today. The flexibility that logical partitioning offers to the midrange environment is a 'killer app' type attribute, which allows companies the capability to rapidly add and change environments. Advanced Power Virtualization, helps decrease TCO, while allowing the use of shared I/O resources where applicable. Through micro-partitioning of its LPARs, it allows one to take advantage of unused clock cycles in an attempt to mirror the mainframe world by utilizing as much of CPU capacity as possible. A major enhancement to pSeries through POWER5 technology is micro-partitioning, which can enable the creation of multiple virtual partitions within a processor, each one tailored to the resource requirements of a particular application based on business needs and priorities. Without partitioning, processing resources are typically underutilized. SMP partitioning today traditionally requires allocation of one or more entire microprocessors to each partition supported. Depending on the nature of the application supported, partitioned resources (processor cycles, memory, I/O) may be underutilized, resulting in unnecessarily high total cost of ownership. With POWER5 SMP micro-partitioning, the ability exists to allocate partial microprocessors to better match workload and increase utilization. Micro-partitions can be tailored to the demands of individual applications, in increments of 1/10th of a processor. The results: increased productive use of system resources, higher system productivity and lower TCO.

The flexibility that IBM provides which allows different types of operating systems to run on its pSeries servers (Linux, AIX, AS400 partitions) provides companies with even more consolidation options. IBM has also published their roadmap of systems that clearly define next steps through the next several years, which is extremely important to partners who need to know the direction that their technology company is taking.

Figure 5



The POWER5 Architecture is clearly a winner.

## References

AIX operating system. (n.d.). In Wikipedia, The Free Encyclopedia. Retrieved February July 18th, 2006, from [http://en.wikipedia.org/wiki/AIX\\_operating\\_system](http://en.wikipedia.org/wiki/AIX_operating_system)

Advanced POWER Virtualization on IBM System p5. IBM Redbook, Document Number SG24-7940-01, 2004. Available at <http://www.redbooks.ibm.com/redbooks/SG247940/wwhelp/wwhimpl/js/html/wwhelp.htm>

Advanced POWER Virtualization on p5Servers: Architecture and Performance Considerations. IBM Redbook, Document Number SG24-5768-01, 2005. Available at <http://www.redbooks.ibm.com/abstracts/sg245768.html?Open>

A High-Performance Architecture with a History. (n.d.) IBM developerworks. Retrieved July 17<sup>th</sup>, 2006 from [http://www-03.ibm.com/servers/eserver/pseries/hardware/whitepapers/power/ppc\\_arch.html](http://www-03.ibm.com/servers/eserver/pseries/hardware/whitepapers/power/ppc_arch.html)

McCarty, B.(2000). Red Hat Certified Engineer. San Francisco:Sybex.

Power to the People; A history of chip making at IBM. (12,2005) IBM DeveloperWorks. Retrieved July 17<sup>th</sup> 2006 from <http://www-128.ibm.com/developerworks/power/library/pa-powerppl/>

Rooney, P. (2004, November 11). Linus Torvalds. CRN, p 23-26.

Santhanam, A. (2003). Towards Linux 2.6.Retrieved November 30' 2004 from <http://www-106.ibm.com/developerworks/linux/library/l-inside.html#h1>