

# **Migrating from Linux Kernel 2.4 to 2.6 on iSeries and pSeries**



**Matthew Davis  
Chakarat Skawratananond  
Nikolay Yevik**

**IBM (e) Server Enablement**

## Special Notices

This publication/presentation was produced in the United States. IBM may not offer the products, programs, services or features discussed herein in other countries, and the information may be subject to change without notice. Consult your local IBM business contact for information on the products, programs, services, and features available in your area. Any reference to an IBM product, program, service, or feature is not intended to state or imply that only IBM's product, program, service, or feature may be used. Any functionally equivalent product, program, service, or feature that does not infringe on IBM's intellectual property rights may be used instead.

Information in this presentation concerning non-IBM products was obtained from a supplier of these products, published announcement material, or other publicly available sources and does not constitute an endorsement of such products by IBM. Sources for non-IBM list prices and performance numbers are taken from publicly available information, including vendor announcements and vendor worldwide homepages. IBM has not tested these products and cannot confirm the accuracy of performance, capability, or any other claims related to non-IBM products. Questions on the capability of non-IBM products should be addressed to the supplier of those products.


Information is provided "AS IS" without warranty of any kind. The information contained in this document represents the current views of IBM on the issues discussed as of the date of publication. IBM cannot guarantee the accuracy of any information presented after the date of publication.

All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics may vary by customer.

All statements regarding IBM future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only. Contact your local IBM office or IBM authorized reseller for the full text of the specific Statement of Direction.

Some information in this presentation addresses anticipated future capabilities. Such information is not intended as a definitive statement of a commitment to specific levels of performance, function or delivery schedules with respect to any future products. Such commitments are only made in IBM product announcements. The information is presented here to communicate IBM's current investment and development activities as a good faith effort to help with our customers' future planning.

The following terms are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both:

AS/400	IBM	pSeries	POWER4
AS/400e	IBM (logo)	RS/6000	
eServer	iSeries	AIX	
	OS/400	VisualAge	

Lotus and SmartSuite are trademarks of Lotus Development Corporation and/or IBM Corporation in the United States, other countries, or both.

MMX, Pentium, and ProShare are trademarks or registered trademarks of Intel Corporation in the United States, other countries, or both.

Microsoft and Windows NT are registered trademarks of Microsoft Corporation in the United States, other countries, or both.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

SET and the SET Logo are trademarks owned by SET Secure Electronic Transaction LLC.

C-bus is a trademark of Corollary, Inc. in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product or service names may be trademarks or service marks of others.

## Table of Contents

Table of Contents .....	3
Main differences between 2.4 and 2.6 Linux kernels on POWER.....	4
New features of Linux Distributions for POWER5 .....	6
Dynamic Logical Partitioning (Dynamic LPAR).....	7
Sub-Processor Partition .....	7
Simultaneous Multi-Threading.....	9
Storage Options .....	9
CDROM, Tape, and DVD-ROM.....	10
Large Page Support .....	12
PCI Hot Plug .....	12
SUE Machine Check Handling .....	12
Development Toolchain Changes.....	12
About the Authors .....	14

## **Main differences between 2.4 and 2.6 Linux kernels on POWER**

### **Module Subsystem, Unified Device Model, and PnP support**

Module subsystem has been significantly changed.

#### **Improved Stability**

The process for loading kernel modules in and out of the kernel was improved to prevent modules from being used during this process altogether or at least to reduce cases when it is possible to use modules while they are being loaded or unloaded, sometimes leading to system crash.

#### **Unified Device Model**

Creation of Unified Device Model is one of the most important changes to 2.6 kernels. It promotes standardization of the module interfaces thus allowing for better control and management of devices, for example:

- Better determination of system devices;
- Power management and power state of a device;
- Improved system bus structure management.

Unified Device Model allows hardware management utilities, such as Red Hat's "kudzu" to make better choices of available drivers for device. Overall, more information about hardware a module supports will be available outside of that module.

#### **Plug-and-Play (PnP) Support**

Changes mentioned in sections 1.1.1 and 1.1.2 combined make Linux running kernel 2.6 a real Plug-and-Play OS. For example, PnP support for ISA PnP extensions, legacy MCA and EISA buses, and hot-plug PCI devices.

#### **Kernel Infrastructure Changes**

- Kernel modules now have .ko extension to differentiate from regular object files with .o extension.
- New 'sysfs' filesystem has been created, that represents device tree as kernel sees it.

## **Memory Support, NUMA**

### **Greater Amounts of RAM Supported**

2.6 kernel supports greater amounts of RAM, up to 64GB in paged mode.

### **NUMA**

Support for Non-Uniform Memory Access - NUMA systems is new in 2.6 kernels.

## **Threading Models, NPTL**

New in v2.6 is NPTL (Native POSIX Threading Library) in comparison to v2.4's LinuxThreads. NPTL brings enterprise-class threading support to Linux, far surpassing the performance offered by LinuxThreads. It is based on 1:1 ratio between user and kernel threads.

As of October 2003, NPTL support was merged into the GNU C library, glibc, and Red Hat first implemented NPTL within Red Hat Linux 9 and Red Hat Enterprise Linux using a customized v2.4 kernel.

## **Performance Improvements**

### **New Scheduler Algorithm**

New  $O(1)$  algorithm has been introduced to 2.6 Linux kernels. It performs especially well under high loads. The new scheduler improves performance by distributing timeslices on a per-CPU basis and thus eliminating the global synchronization and recalculation loop.

### **Kernel Preemption**

New 2.6 kernels are preemptive. This will significantly improve performance of interactive and multimedia applications.

### **I/O Performance Improvements**

Linux's I/O subsystem has also undergone major changes to allow I/O operations to be more responsive by changing I/O scheduler to ensure that no process is stuck in the queue for too long waiting to perform input/output operation.

### **Fast User-Space Mutexes**

Responsiveness is also improved by introducing "futexes" (Fast User-Space Mutexes), that allow threads serializing to avoid race conditions. Improvement is achieved by implementing "futexes" in part in kernel space to allow prioritizing waiting tasks on a basis of contention.

## **Scalability Improvements**

### **Higher Processor Count**

Linux kernel 2.6 can support up to 64 CPUs.

### **Larger Memory Support**

On 32-bit systems due to PAE (Physical Address Extensions) memory support in paged mode was increased to 64GB.

### **Users and Groups**

Number of unique users and groups has been increased from 65,000 to over 4 billion, that is from 16-bit to 32-bit.

### **Number of PIDs**

Maximum Number PIDs was increased from 32,000 to 1 billion

### **Number of Open File Descriptors**

Number of open file descriptors was not increased, but this parameter is no longer required to be set up in advance, it will self-scale.

### **Greater Number of Devices Supported**

Previous to Linux kernels 2.6 there were limits within the kernel that could constrain large systems, such as 256 devices per chain. The v2.6 kernel moves well beyond these limitations, not only supporting more types of devices, but also more devices of the same type. Under Linux 2.6 system can allow for 4095 major device types and more than a million of subdevices per a single type.

### File systems Size

Linux kernel 2.6 allows addressing file systems sizes of up to 16TB.

### File systems

Traditional Linux file systems such as ext2, ext3, and ReiserFS were significantly improved. Most notable improvement is introduction of extended attributes, or file metadata. Of the major importance is implementation of POSIX ACL, an add-on to usual UNIX permissions that allows for more fine-grained user access control.

In addition to improved support for traditional Linux filesystems, the new kernel includes full support for relatively new on Linux XFS filesystem.

Linux 2.6 kernels now also features improved support for NTFS filesystem, now allowing mount NTFS filesystem in read/write mode.

### New features of Linux Distributions for POWER5

Linux distributions that will run on POWER5-based systems are SUSE LINUX Enterprise Server 9 (SLES 9), and Red Hat Enterprise Linux Advanced Server 3 with the third service pack (RHEL AS 3 Update 3). Both distributions will be generally available in 2004. SLES 9 is based on Linux kernel 2.6. RHEL AS 3 Update 3 is based on Linux kernel 2.4. Both SLES 9 and RHEL AS 3 Update 3 will run on POWER4 hardware as well. The following table highlights POWER5 features supported in the two distributions.

Function	SLES 9	RHEL AS 3 Update 3
<b>Dynamic LPAR</b>		
-- Processors	Y	N
-- Memory	N	N
-- I/O	Y	N
-- Max 254 Partitions	Y	Y
<b>Sub-Processor partition with 0.1 granularity</b>	Y	Y
-- Capped and Uncapped partitions	Y	Y
<b>Simultaneous Multi-Threading</b>	Y	Y
<b>Storage Options</b>		
--Virtual SCSI Server	N	N
--Virtual SCSI Client	Y iSeries Initially pSeries with AIX 5.3	Y iSeries Initially pSeries with AIX 5.3
<b>Communication Options</b>		
-- Virtual LAN	Y	Y
<b>Large page support</b>	Y	N
<b>PCI Hot Plug</b>	Y	N
<b>SUE machine check handling</b>	Y	Y

In the following, we provide detailed description for those features.

## Dynamic Logical Partitioning (Dynamic LPAR)

Logical partitioning allows multiple operating systems to reside on a hardware platform simultaneously. System resources are divided so that partitions cannot interfere with each other. Managing LPARs in the system is made possible by the hardware management console (HMC). With Dynamic LPAR, resources can be dynamically added and removed without requiring a partition reboot. When these resources need to be added, administrators can reconfigure the system to recognize these additional resources. The maximum number of logical partitions supported depends on the number of processors in the server model and the system limit is 254. Adoption of Dynamic LPAR is ultimately determined by the Linux distributor and the use of the 2.6 kernels. SLES 9 supports the dynamically movement of processors and I/O. RHEL AS 3 Update 3 will not support Dynamic LPAR.

## Sub-Processor Partition

A minimum of 0.10 processing units can be configured for any partition using shared processors. A group of physical processors that can be shared among multiple logical partitions is called a *shared processing pool*. The shared processor function allows you to assign partial processors to a logical partition.

Consider Figure 1 as an example of an environment using the shared processor pool. This figure represents a fictional setup of a 4-way machine running either i5/OS or AIX. It also has three additional logical partitions. Assume the second partition is a transactional server that processes financial transactions; furthermore, assume that this transaction application interacts with either AIX or i5/OS to store and retrieve its information in a database. The partition labeled "Report" is the sister-application to the transactional server and it generates financial reports. For the purpose of load balancing, the company has separated the transactional and report partitions because the transactional server is time and response sensitive while the report generation can be done at offpeak times. In the last partition is the company's development and test partition. This partition serves as a development space for their engineers. Notice how the processors have been divided up between the four partitions based on workload.

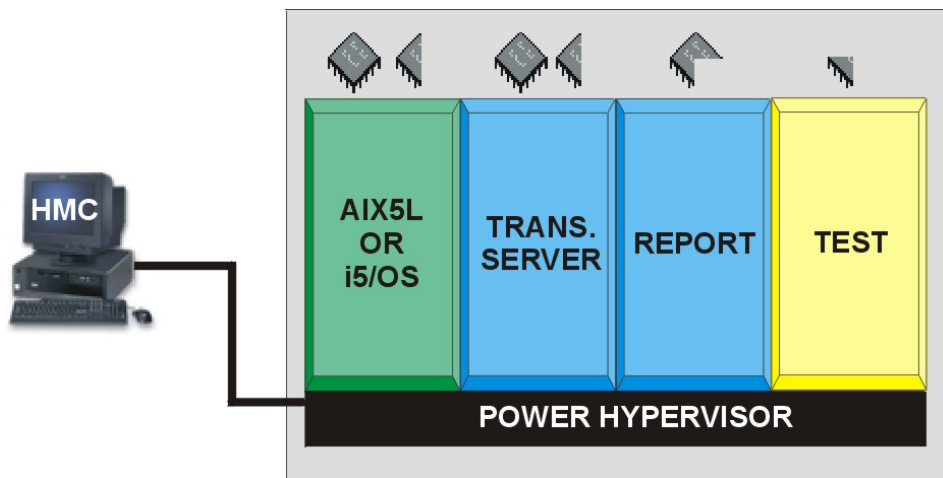
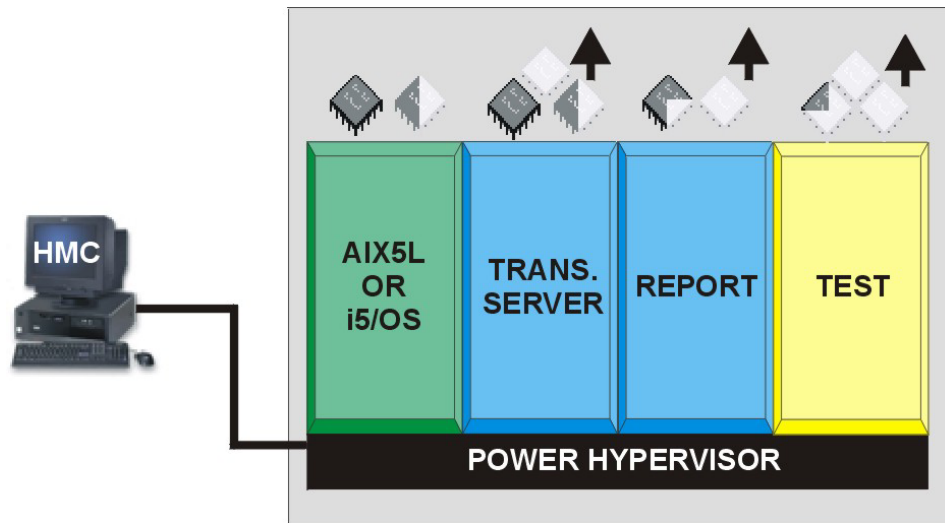


Figure 1 – An example of shared processors

Partitions in the shared processing pool can have a sharing mode of capped or uncapped. A *capped partition* indicates that the logical partition will never exceed its assigned processing capacity. Any unused processing resources will only be used by the *uncapped partitions* in the

shared processing pool. You can specify whether a partition is capped or uncapped when you define the partition's profile. While defining a partition, you can also set a minimum and maximum processor value for number or fractions of processor power. This fits nicely with the example that was discussed earlier. Figure 2 is an evolution of the previous example but now the minimum and maximum values are represented.



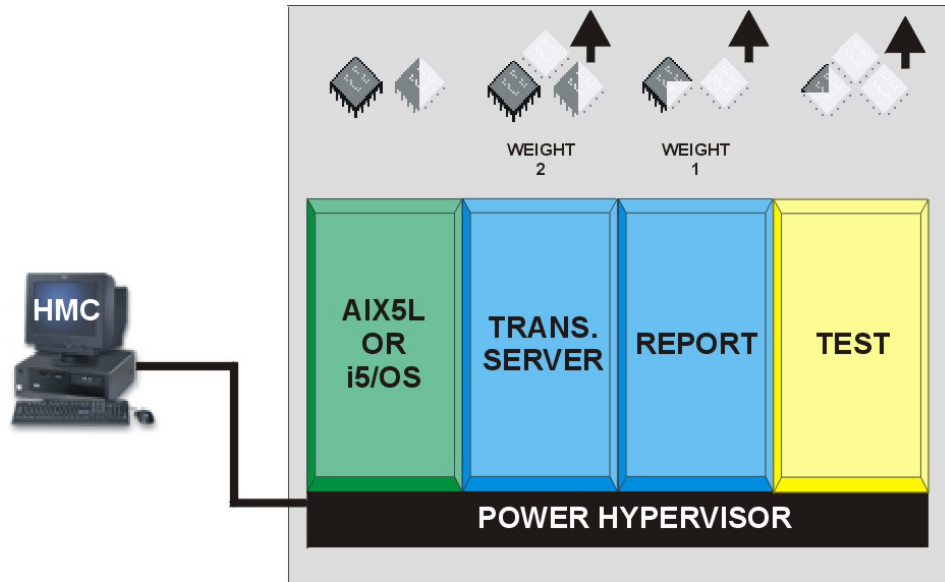
**Figure 2 – Dynamic movement of processor power based on workloads**

The advantage of being able to dynamically move processors based on demand is very evident in the fictitious example. The loads of the transaction and report partitions crystallize the very need for dynamic processor allocation. The transaction server has one and a half processors allocated (this being the minimum). It also has the ability to consume the second half of the second and all of the third virtual processors based on demand. If you assume the reports are run on off-peak hours during which the system may have more idle time, then the report partition and its applications can consume up to two virtual processors but no less than three-quarters of one processor. The same goes for the test partition. Suppose the engineers need to compile their applications. If the compile is done while there is idle processing power, the Test partition can consume up to 3 virtual processors allowing their compiles to complete quicker.

Most of these instances for processor sharing have been based on parts of the system being idle so that other partitions can use the resources, but there will certainly be times where multiple partitions are asking for more processing power. Consider an example where the both the Report and Transaction servers require more processor power because of peaking workloads. Because timely response from the Transaction server is critical to your business, you would prefer the Transaction server get virtual processing power before the Report partition. This is where setting weights for processing power becomes important.

*Uncapped weight* is a number in the range of 0 through 255 that you set for each uncapped partition in the shared processing pool. By setting the uncapped weight (255 being the highest weight), any available unused capacity is distributed to contending logical partitions in proportion to the established value of the uncapped weight. The default uncapped weight value is 128.





**Figure 3 - Weights determine distribution of unused processors**

In the situation where both the Transaction and Report servers are peaking, weights can be set to determine how processors should be allocated. In Figure 3, the weights for the Transaction server are set to two and the Report server was set to one. So for every three processing units that are available during the peak, the hypervisor will assign two processor units to the Transaction server and one to the Report server.

### **Simultaneous Multi-Threading**

The POWER5 architecture features the Simultaneous Multi-Threading technology. The POWER4 microprocessor collects a group of up to five instructions per clock cycle and can complete one group of instructions per clock cycle. The POWER5 microprocessor doubles that throughput by collecting two groups of up to five instructions per clock cycle and completing two groups per clock cycle. Both SLES 9 and RHEL AS 3 Update 3 support this technology.

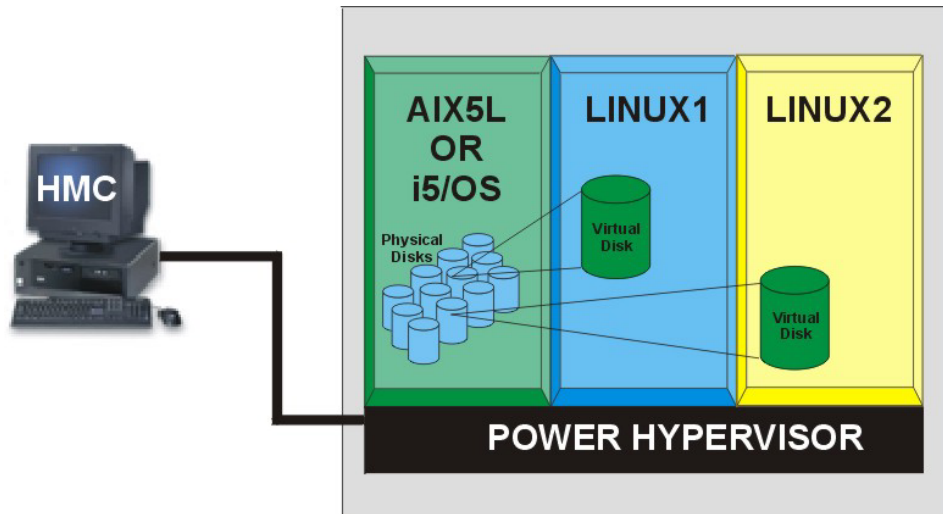
### **Storage Options**

For storage and I/O, Linux can take advantage of a variety of real and virtual devices. This flexibility allows for cost-effective setup of Linux partitions. In the case of disks, Linux logical partitions support three different storage options.

1. Internal storage using SCSI adapters and drives dedicated to the partition.
2. External storage using SAN adapters dedicated to the partition.
3. Virtual storage using a virtual SCSI adapter and storage in a different partition.

### **Virtual Disk**

Virtual storage allows multiple partitions within a POWER5-based system to share storage. One partition, the I/O server partition, owns the physical adapters and storage (which may be internal or external). Virtual adapters allow other partitions, I/O client partitions, to use storage from the I/O server partition. I/O server partitions can be AIX and i5/OS. Both SLES 9 and RHEL AS 3 Update 3 support this.



**Figure 4 – AIX or i5/OS can provide virtual disk to Linux partitions**

Figure 4 graphically describes how a hosting partition can provide virtual disks to Linux partitions. The benefits virtual disk include more than saving expense on disk drives. On smaller machines, adding disks and controllers may be a challenge and may also require the purchase of an expansion unit. Also, the virtual disks can be managed, backed-up, and quickly replicated by the hosting system.

#### **CDROM, Tape, and DVD-ROM**

You can also share SCSI devices owned by AIX 5.3 or i5/OS with Linux partitions. It works very similar to the virtual disk function. If AIX or i5/OS owns a CDROM, tape device, or DVDRAM drive, Linux can use those devices as if it were physically attached to the Linux partition provided that the hosting partition is not actively using the device. The benefits of virtual SCSI devices are much the same as virtual disk; less hardware expense and not needing to dedicate a device to each partition.

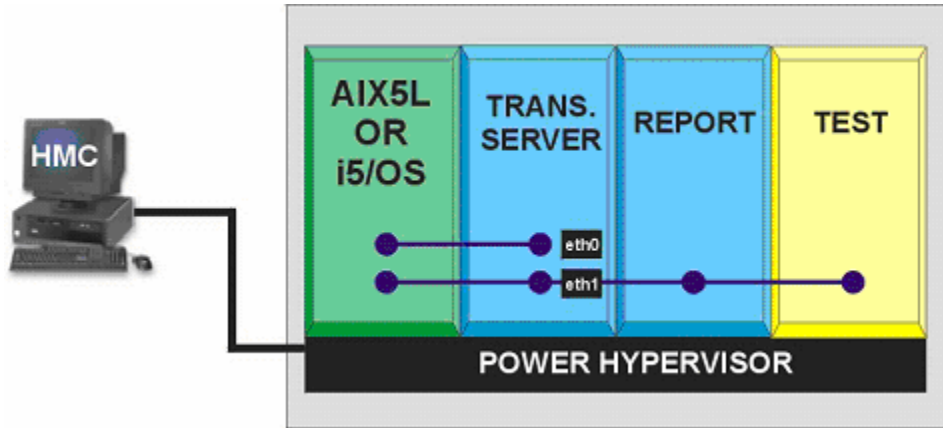
#### **Communication Options.**

Linux on POWER5-based systems can establish a TCP/IP connection through either a directly attached network interface or through a virtual Ethernet interface. Virtual Ethernet provides roughly the same function as a 1 Gigabit Ethernet adapter. Partitions in POWER5-based servers can communicate with each other using TCP/IP over the virtual Ethernet communication ports.

You can define up to 4,094 separate virtual Ethernet LANs (VLANs). Each partition can have up to 65,534 virtual Ethernet adapters connected to the virtual switch. Each adapter can be connected to 21 VLANs. The enablement and setup of virtual Ethernet does not require any special hardware or software. After you enable a specific virtual Ethernet for a partition, a network device named ethXX is created in the partition. The user can then set up TCP/IP configuration to communicate with other partitions.

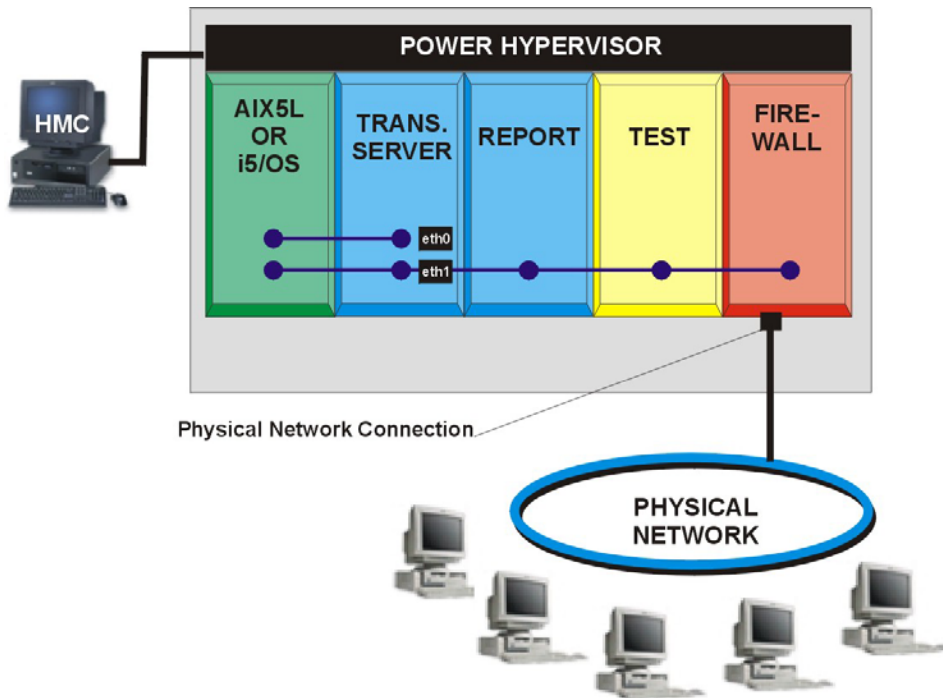
Let's reconsider the example from the Sub-Processor Partition section on page 7. In the description of the scenario, the Transaction server uses a database in the AIX or i5/OS partition for storing and retrieving information. This is a very typical use of virtual Ethernet, because the communication is

very fast and no additional hardware is required. Figure 5 depicts the example with the addition of a two virtual Ethernet LANs.



**Figure 5 – Virtual Ethernet LANs are fast and cost effective ways for partitions to communication with each other.**

In most cases, however, you will want to allow partitions connected to a virtual Ethernet to also communicate with the physical network. That requires that at least one partition have both a physical Ethernet adapter, as well as a virtual Ethernet adapter that is connected to the other partitions. The partition owning both adapters can route traffic between the physical and virtual Ethernets.



**Figure 6 – A partition that has both a physical and virtual network connection, can route traffic to a physical LAN.**

A common way to connect your partitions to a physical network is to run a firewall in one of your partitions. In the firewall partition, you can have a network interface card that connects directly to the physical network as seen in Figure 6. The other partitions can then communicate with the physical network by passing traffic through the virtual LAN and the firewall.

At the time of this writing, Virtual LAN is available for both SLES 9 and RHEL AS 3 Update 3.

## **Large Page Support**

In the 2.6 kernel, there will be support for two virtual page sizes: the traditional 4KB page size and the 16MB page size. Large page usage is primarily intended to provide performance improvements to memory access intensive applications. With large page support, applications are able to run with text and data segments backed by large pages (16MB) with no changes to the application code. The performance improvements are due to the reduced translation lookaside buffer (TLB) misses. This is because TLB is able to map a larger virtual memory range. Large pages also improve memory prefetching by eliminating the need to restart prefetch operations on 4KB boundaries. Large page is supported in SLES 9, but not in RHEL AS 3 Update 3.

## **PCI Hot Plug**

With this capability, you can insert a new PCI hot plug adapter into an available PCI slot while the operating system is running. This can be another adapter of the same type that is currently installed or a different type of PCI adapter. New resources become available to the operating system and applications without having to restart. You can also replace a defective PCI hot plug adapter with another of the same type without shutting down the system. When you exchange the adapter, the existing device driver supports the adapter because it is of the same type. Device configuration and configuration information about devices below the adapter are retained for the replacement device. PCI Hot Plug is supported in SLES 9, but not in RHEL AS 3 Update 3.

## **SUE Machine Check Handling**

This is the capability that allows the system to mark the Special Uncorrectable Errors (SUE) and kill any dependent processes that reference this resource while the system continues to run without requiring reboot to recover from error. Both SLES 9 and RHEL AS 3 Update 3 support this feature.

## **Development Toolchain Changes**

With all the innovation in the Linux 2.6 kernel, accommodations must be made in the libraries and user space development tools. This section will address changes in the GNU toolchain, referring to glibc, bintuils, as, ld, and gcc. While this section is far from complete, it should be a good starting reference to be supplemented by additional reading in the freely available source change logs.

### **Glibc**

Glibc, the GNU C library, has been renovated in version 2.3 to support new and improved 2.6 kernel features as well as new and extended function for the POWER architecture. Primarily, the changes have centered around the Native POSIX Threads for Linux model. Changes have also been made for internationalization, network interface addressing, and regular expression usage, among other things.

#### **Internationalization**

Internationalization has been improved by enabling iconv to use the system locale. Also, thread-safe interfaces to locale.h have been implemented. They are not individually reviewed here, but details are kept with the freely available source code documentation.

#### **Network interface**

Network interface addressing has been improved with a BSD-compliant implementation.

### **Regular expression**

Regular expression is now considerably faster after a complete rewrite to be POSIX compliant.

### **Fexecve**

Fexecve used to exec file descriptors, is now enabled in Linux.

### **Malloc**

Malloc has been based on Doug Lea's Malloc 2.7.0.c to be faster and more compatible.

### **Thread-locale storage**

Thread-locale storage has been implemented to allow for faster collection and storage of void objects in threads as this is handled by the compiler now. For more information, see Ulrich Drepper's whitepaper at <http://people.redhat.com/drepper/tls.pdf>

## **GNU binutils**

The GNU binutils include ld, as, and several other minor utilities, such as objcopy and readelf. These minor binutils have not changed specifically to accommodate the 2.6 kernel or the POWER architecture in this release, but have changed in less significant ways. For example, the utility readelf can now be used to display information on files stored in archives. A full changelog is available on the binutils website at <http://sources.redhat.com/binutils/>

## **AS and LD**

AS and LD have changed in several ways for POWER architecture. Though because the POWER architecture is used with a variety of operating systems, these changes are not all for Linux specific needs. Changes that do affect Linux on POWER include better support for POWER opcode, additions for VMX extensions (available on PPC970 based Linux offerings). Additionally, optimization profile are now available for POWER4 and PPC970 chips. The default optimization profile is -maltivec, supporting optimization for the VMX extensions found in the PPC970. For POWER4 optimization, use -mpower4.

## **GCC**

GCC has changed notably to include support for NPTL as well, but other changes should not be overlooked. Numerous improvements for POWER scheduling, optimization, and compliance have been added.

### **DFA Scheduler**

DFA Scheduler for instructions is support in gcc 3.3.3. Learn more about this project at <http://www.gnu.org/software/gcc/news/dfa/html>

### **Directives**

Directives can now be used inside of C macros.

### **Includes**

-I library includes are ignored if the library was already in the path. This avoids unexpected ordering problems with library includes.

### **New support for the POWER4 processor**

New support for the POWER4 processor specific optimizations, i.e. -mpower4

### **Improvements for VMX extensions**

Several function improvements for VMX extensions in PPC970 chips.

### **More ISO C99 compliance.**

For a current status, see this table at <http://www.gnu.org/software/gcc/gcc-3.3/c99status.html>

### **About the Authors**

**Nikolay Yevik** is a Linux Technical Consultant in the IBM eServer Solutions Enablement Team. He has more than 5 years of experience in C/C++ and Java software development, and AIX and Linux System Administration. He holds MS degrees in Petroleum Engineering and Computer Science.

**Chakarat Skawratananond** is a technical consultant in the IBM eServer Solutions Enablement organization, where he assists ISVs in enabling their applications for AIX and Linux on the IBM pSeries platform. You can contact him at [chakarat@us.ibm.com](mailto:chakarat@us.ibm.com). He holds PhD in Electrical Engineering.

**Matt Davis** is a Linux technical consultant in the IBM eServer Solutions Enablement team. As a member of the pSeries Linux project since its inception, he explored and tested emerging technology for pSeries Linux and wrote several reports summarizing his findings. These include papers on journaling file systems, parallel grid computing, open source alternatives to commercial software, and the Linux Solutions Catalog. He came to IBM as an intern during his tenure as a student at the University of Texas at Austin, from which he earned two degrees.