Linux RAS for IBM pSeries

Reliability, Availability, Serviceability (RAS)

Version: 1.0

September 30, 2003

Syed Iggy Haiderzaidi - RAS Architecture and Engineering
IBM eServer
Austin, Texas

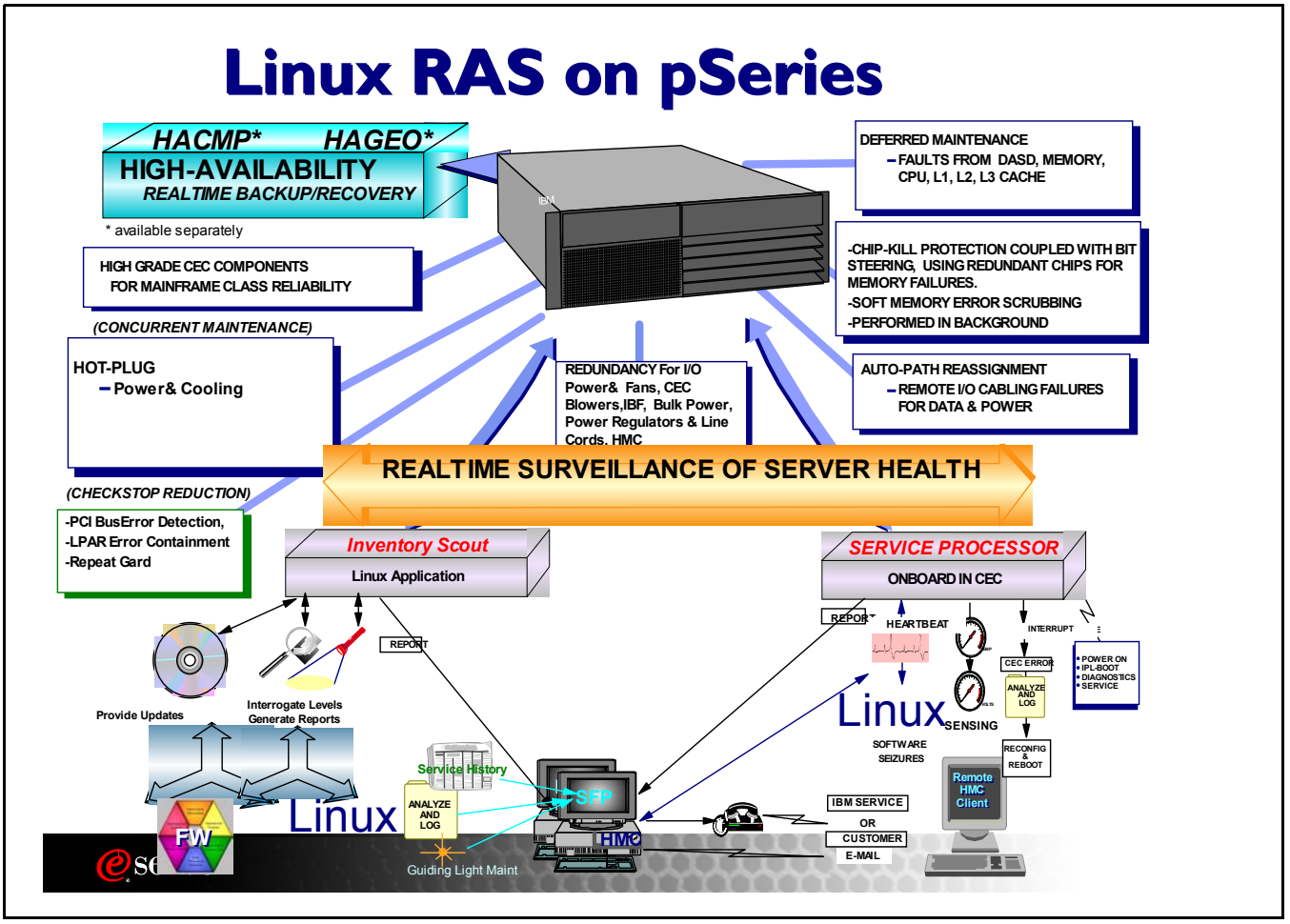
Advanced RAS Features and Functions

IBM has spent years developing RAS capabilities for mainframes and mission-critical servers. The IBM @server™ pSeries™ hardware has been able to take advantage of this knowledge and experience with customer requirements.

The following features provide the pSeries hardware with Linux industry-leading RAS features:

- Automatic First Failure Data Capture and diagnostic fault isolation capabilities
- Self-healing internal POWER4™ processor array redundancy
- Scrubbing and redundant bit-steering for self-healing in main storage
- ECC and Chipkill™ correction in main storage
- Fault tolerance with N+1 redundancy of power and cooling, dual line cords, and concurrent maintenance for power and cooling
- Predictive failure analysis on processors, caches, and memory
- Processor and memory boot-time deallocation (Persistent Processor Deallocation)
- Fault avoidance through highly reliable component selection, component minimization and error mitigation technology internal to chips
- Concurrent run-time diagnostics for power and cooling

Excellent quality and reliability are inherent in all facets of the pSeries servers. These capabilities are designed to help ensure that the each server operates when required, performs reliably, efficiently handles infrequent failures in a nondisruptive fashion, and provides timely and competent repair in many cases either concurrently or on a deferred basis to allow operational resumption with minimal inconvenience. Mainframe-inspired diagnostic capability based on internal error checkers, First Failure Data Capture of all internal error check states is provided for all CPU, memory, I/O, power and cooling components are designed to eliminate the need for recreating failures.



Reliability is one of the most significant factors in the design of all IBM products. RAS is an integral part of the pSeries - and Linux philosophy which is based on our high-end server design. It begins with the development of architectures, where RAS innovations are of paramount importance. It flows through design and product development stages, where RAS designs are reviewed, assessed, developed, evaluated, and perfected. It continues through the manufacturing and release processes, where the manufacturing quality is extensively measured and is under continual evaluation. It culminates in service and support; where the reliability is consistently monitored for deviation from the criteria, where warranty and maintenance have high priority, and where significant customer problems are assigned to and addressed by an expert team.

All of the development processes, from the architectural and concept phases of development, through the manufacturing process, and culminating in the provision of

service and support are ISO-certified and audited periodically for ISO compliance by representatives of Underwriters Laboratories Inc.

Reliability - Fault Avoidance

Major design efforts have contributed to the development of the pSeries servers to analyze single points-of-failure within the Central Electronic Complex (CEC) to either eliminate them or to provide hardening capabilities to significantly reduce their probability of failure. The best way to harden a system is to prevent the errors from occurring in the first place. Components within the CEC are designed to provide "mainframe" levels of reliability. These components provide the superior levels of reliability and undergo additional stress testing and screening above and beyond the industry-standard components.

Fault avoidance is also served by minimizing the total number of components, and this is inherent in POWER4™ chip technology, with two processors per chip. In addition, internal array soft errors throughout the POWER4 chipset have been systematically masked using internal error checking and correcting (ECC) and recovery techniques. Going beyond ECC in the memory subsystem, the basic memory DIMM technology has significantly improved reliability features through the use of more reliable soldered connection to the memory card.

Single Chip Modules (SCM) and Multi Chip Modules (MCM) utilize the high density, copper technology mounted on a "silicon-on-insulator" substrate to provide highly dense, high-performance chips running at reduced temperatures which also is designed to increases reliability.

Variable speed fans in the system unit are designed to maintain proper cooling levels in case of a fan fault.

This packaging provides for electromagnetic compatibility (EMC) shielding designed to minimize errors induced by electrical noise, and provides positive retention seating to help prevent shocks or vibrations from loosening critical system connections. These packaging features eliminate many of the intermittent errors experienced in UNIX servers containing less robust packaging.

During the design and development process, subsystems go through rigorous verification and integration testing processes. During system manufacturing, the pSeries systems go through a thorough testing process to help ensure high product quality level. Extensive error detecting and checking circuitry helps maintain the integrity of data stored and transported in the system. The system design facilitates the recognition of component errors that are either corrected dynamically, or properly reported for isolation and repair. Parity on the system bus, cyclic redundancy checking (CRC) on the Remote I/O (RIO)

bus, and the extensive use of ECC on memory and arrays provide some of these capabilities.

BIST (Built-In Self-Test) and POST (Power-On Self-Test) are designed to check the processors, caches, and associated hardware that are required for proper booting of the Linux operating system every time the server is powered on. Additional testing can be selected at power-up time to fully verify the system memory and the chip interconnect wiring as an added reliability measure.

The system automatically reboots in the extended test mode following a failure to check that all components are thoroughly tested and verified. If a non-critical error is detected, or if an error occurs in resources that can be deconfigured from the system, or if a processor has been marked for deconfiguration by Persistent Processor Deallocation, the boot process will attempt to proceed to completion with the faulty device automatically deconfigured. Detected errors are logged in the system non-volatile RAM (NVRAM). Run-time algorithms then gather the information from the NVRAM, and log it to the Linux Sys log facility.

The Linux log facility is where hardware and software failures are recorded and analyzed by Linux Error Log Analysis (LELA) routines to provide warnings to the system administrator on the possible causes of system problems. If the system concludes that service is required, an update is made to the Diagnostic Event Log which is monitored by Remote Management Control (RMC). RMC forwards a service action event to the Service Focal Point application running on the Hardware Management Console (HMC). If service is required, the Service Focal Point (SFP) will provide any filtering of duplicate service requests which may have been received from multiple sources or operating system partitions and initiate the call home for service. Data on the nature of the failure, the parts to be replaced and additional data on the specific machine configuration is gathered and transmitted to the service provider. This enables the service representative to bring along needed replacement hardware components when a service call is placed, thus minimizing system repair time.

Surveillance of system operation is one of many functions provided by the service processor, which is a separate microprocessor subsystem whose many additional functions will be described in more detail in a later section. During boot time, a surveillance monitor in the service processor is automatically enabled to check for “heartbeats” from the boot firmware. If a heartbeat is not detected within a default period, the service processor is designed to cycle the system power and attempt to reboot until the system either boots successfully, or a retry threshold is reached. If the threshold is reached, the service processor logs the error, leaves the system powered on, and provides the user with various options to assist in diagnosing the error. The error logs can be interrogated and various options exist for attempting to reboot the system. The service processor is designed to report the error to the Service Focal Point. This capability will be described in more detail in the later section on Serviceability.

In symmetric multiprocessor (SMP) mode, the service processor can also be optionally configured to monitor for operating system hangs or failures. If enabled, the service processor can log operating system hangs or fails and report them the next time that the system is successfully booted. This function is disabled in the logical partitioning (LPAR) mode of operation.

Similar to the firmware surveillance scenario, the service processor can be enabled to notify the Service Focal Point and report the operating system surveillance failure condition.

Availability - Redundancy and Recovery

Power and Cooling Subsystem

The pSeries systems running Linux bring new levels of availability features and functions. Within the CEC, the N+1 power and cooling subsystem is designed to complete redundancy in case of failures in the power supplies, the power controllers and the cooling units as well as the power distribution cables. Concurrent repair is supported on all of the power and cooling components when optional power and cooling features have been installed.

The I/O drawer extends availability by providing N+1 power supplies and fans. The power supplies can be repaired concurrently, while the fans can be repaired on a deferred basis.

The interface from the processor to the I/O is through the Remote I/O (RIO) link. This link, in a similar method to the SSA interface, uses a loop interconnect technology to provide redundant paths to I/O drawers. RIO is a proven and robust interconnect technology. RIO availability features include CRC checking on the RIO bus with packet retry on bus time outs. In addition, if a RIO link fails, the hardware is designed to automatically initiate a RIO bus reassignment to route the data through the alternate path to its intended destination.

Power to the drawers is controlled from the power controller in the CEC through the System Power Control Network (SPCN) link. This link is implemented in a point-to-point technology. Any break in the loop is recoverable via alternate routing through the other link path and can be reported to the service provider for a deferred repair.

The pSeries systems use Linux Error Log Analysis and Service Aids that allow administrators or IBM service representatives to diagnose potential system malfunctions. Auto-restart (reboot) options, when enabled, can automatically reboot the system following an unrecoverable software error, software hang, hardware failure or environmentally-induced (AC power) failure as described in the reliability section. These standard high-availability features, coupled with the IBM High Availability

Cluster Multiprocessing (HACMP) for Linux program product offering, offer outstanding server availability.

PCI Bus Parity Error Recovery

EEH is an acronym for Enhanced I/O Error Handling. Hardware-level PCI fault isolation is achieved by isolating each IOA (I/O Adapter) on a secondary PCI bus. This is implemented in the EADS (or it's PCI-X follow-on, EADS-X) chip. The primary PCI bus runs between the PCI Host Bridge chip (such as Speedwagon or Winnipeg) and EADS.

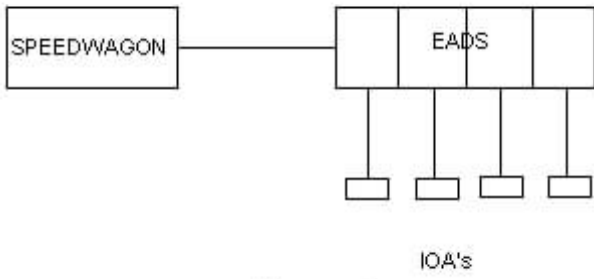


Figure 1

EADS provides up to 4 PCI slots, with each of these slots being behind a PCI-to-PCI bridge. Thus when EADS detects an error, it can "freeze" the slot, such that PCI transactions cannot be forwarded in either direction across the PCI-to-PCI bridge. This "freeze on error" capability is the foundation of EEH.

Both firmware and the Linux code are involved in using EEH. The operating system never interacts directly with the EADS chip; instead, it calls firmware to perform the desired EEH-related operation. First, the "freeze-on-error" capability must be turned on. Details of how this works in pLinux will be described later.

The next event to consider is when an error occurs. The EADS chip detects the error on the basis of PCI protocol or internal chip logic, and freezes the slot. While the slot is in the frozen state, any transactions initiated by the IOA (i.e.. DMA operations) are blocked, they will not be forwarded to the primary bus. Load/store operations generated by the operating system that are directed to the IOA are also blocked. Store operations (writes to the IOA) are simply thrown away by EADS (from the operating system's perspective,

they fail silently). Load operations (reads from the IOA) return a value of all one's to the operating system. Instead, the detection actually occurs in the Linux kernel code. The kernel provides a set of functions to load and store to IOAs. On ppc64, the functions that implement the load calls have been made EEH-aware, so that after every load from an IOA, if the value returned is all one's, the kernel will check (via a firmware call) to see if the slot is frozen. More details on this implementation appear later, the important point here is that on Linux the slot freeze is detected when a Linux device driver calls certain kernel code to read from its device.

The open source community needs to define and accept the Linux device drivers behavior which have not been explicitly modified for EEH, so they do not have any of the logic to drive the EEH recovery model. Instead, when a slot freeze is detected (i.e.. EEH-aware kernel load call receives all one's, and firmware callback indicates the slot is frozen), the unconditional kernel response is to call panic(). The behavior then depends on how kernel has been configured, typically system will either drop into kdb or the it will automatically reboot after three minutes. Enhancements are being made so that some detailed data on the slot freeze will be collected via a firmware callback and then placed in syslog after the operating system is rebooted. But from the endures perspective, detection of a slot freeze will result in a system (SMP) or partition (LPAR) crash.

Since Linux is not currently capable of driving the EEH recovery model, the recovery of the slot then becomes a function of hardware and firmware capabilities. The kind of error experienced is a key parameter. For an error that was isolated to a given EADS slot, other partitions will be unaffected. When an error instead places the PHB in the error state, any partitions owning EADS slots or other devices underneath the same PHB will be impacted as well. The net is that with EEH enabled, recovery occurs either by rebooting the partition or rebooting the entire CEC.

Slot Freeze Detection

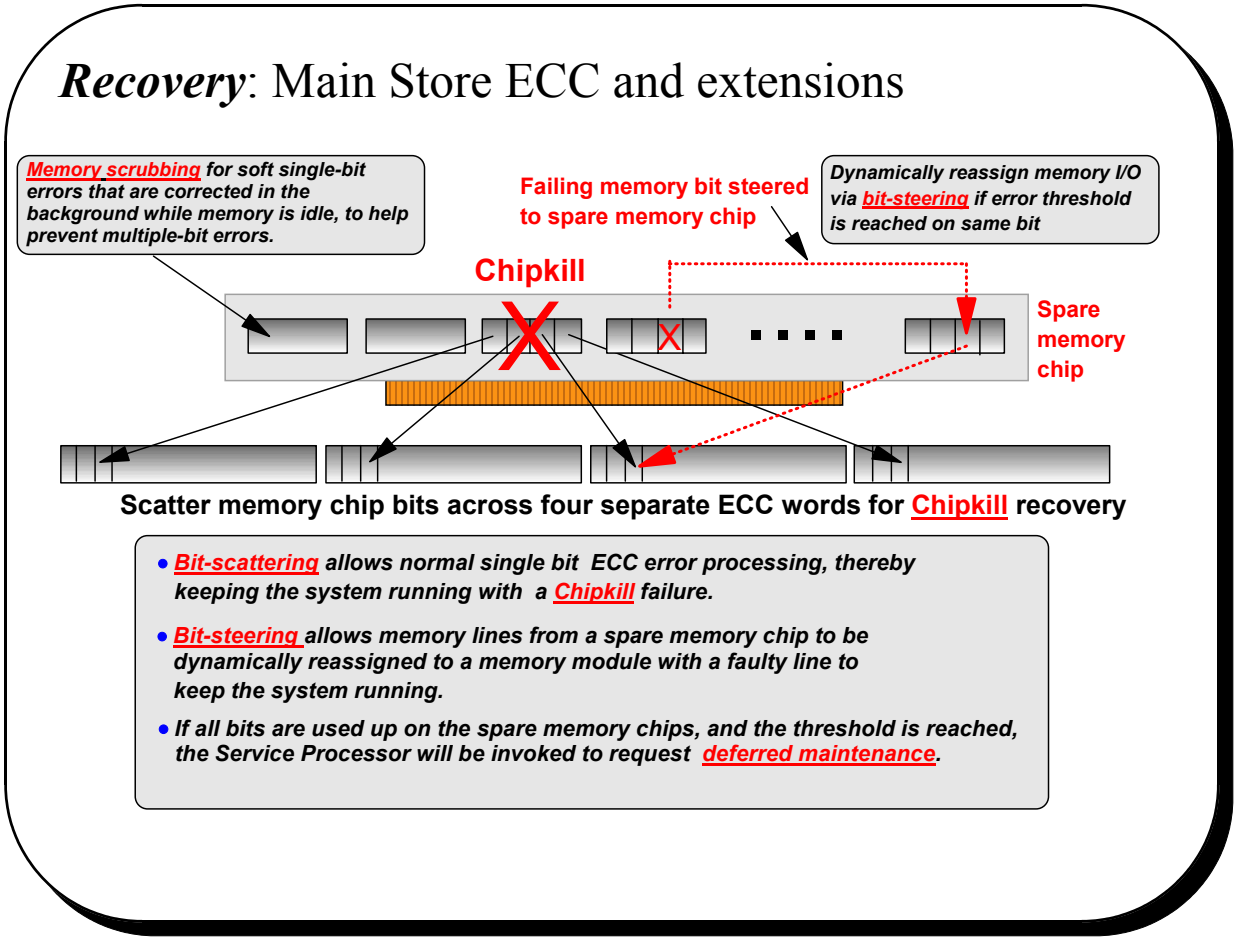
The heart of the slot freeze detection strategy is the use of certain kernel functions to access device registers and device memory. Detection actually only occurs by the functions that perform loads (inb, inw, inl, readb, readw, and readl at present). The insb, insw, and insl functions are also being made EEH-aware. On every call to one of these functions, if the value read from the device is all one's (and EEH slot freeze detection has been enabled), then firmware is called to determine is the slot is actually frozen.

When a slot freeze is detected, panic is called and the operating system crashes. There are some different external behaviors depending on the level of kernel code is running, as well as the configuration. A message may be displayed on leds/HMC. The latest code is capable of calling firmware to collect additional information about the nature of the slot freeze, and to log this error information into NVRAM. If kdb is enabled, stack trace can be viewed.

Without kdb, the default panic behavior is to reboot in three minutes. Finally, with the latest code, any error information cached into NVRAM prior to the crash will be extracted and placed into syslog during the next boot.

Memory Error Correction Extensions

The standard memory has single-bit error correct and double-bit error detect ECC circuitry to correct single-bit memory failures. The double-bit detection helps maintain data integrity by detecting and reporting multiple errors beyond what the ECC circuitry can correct. The memory chips are organized such that the failure of any specific memory module only affects a single bit within an ECC word (bit-scattering) thus allowing for error correction and continued operation in the presence of a complete chip failure (Chipkill recovery).



This memory also utilizes memory scrubbing and thresholding to determine when spare memory modules within each bank of memory should be used to replace ones that have exceeded their threshold value (dynamic bit-steering). Memory scrubbing is the process of reading the contents of memory during idle time and checking and correcting any single-bit errors that have accumulated. These single-bit failures could be either solid

(technology failures) or soft failures (intermittent errors caused by noise or other cosmic effects). If an error is detected, the system hardware is designed to correct it by passing the data through the ECC logic that corrects the fault and then writing the corrected contents back to its memory address location. This function is also used to restore correct memory data after bit-steering takes place. Scrubbing is a hardware function on the memory controller chips that takes place during memory idle time, and does not influence normal system memory reference performance.

In order to prevent an uncorrectable memory error from causing a system outage, the service processor is designed to initiate a deferred maintenance request on memory cards that have used their spare bits and are experiencing additional correctable errors (memory predictive failure analysis).

Unrecoverable Error Handling

All levels of storage protected by ECC are capable of failures whose resulting errors are uncorrectable because the failure corrupted more than one bit of an ECC word. In the I/O domain, there is also the potential of unrecoverable errors between a PCI Host bridge and PCI to PCI bridges. While Chipkill ECC formatting helps eliminate unrecoverable errors in main store, for many of the others, the pSeries servers are capable of presenting a synchronous machine check interrupt to the processor with the hardware state indicating the address of the referring instruction. This permits localizing the error to the software or firmware involved. In the case of logical partitioning (LPAR), the effect will be a software reboot of the partition. The ultimate capability of this error handling will be software process terminate rather than partition reboot.

Availability - First Failure Diagnostics and Reconfiguration

The ability to correctly diagnose problems in a computer is the bedrock capability upon which availability is based, and without this pervasive capability, even simple problems which behave intermittently can be a cause for serious and prolonged outages. pSeries systems provide unmatched capability in both IPL and initialization diagnostics, based on internal test procedures, and in run-time first failure diagnostics based on strategic error checkers operating full time to detect and capture precise error signatures with predetermined hardware fault domains.

The diagnostics goal for the pSeries systems running Linux is to isolate **96% of the failures to a single Field Replaceable Unit (FRU).** For the estimated 4% failures, two FRUs plus any boards or wires that interconnect the FRUs are candidates for fault identification. In this 4% of the cases, manual isolation procedures (MAPs) may be employed by the service person. In order to attain these isolation goals based on error checkers alone, the entire system logic design must contain strategically placed error checkers.

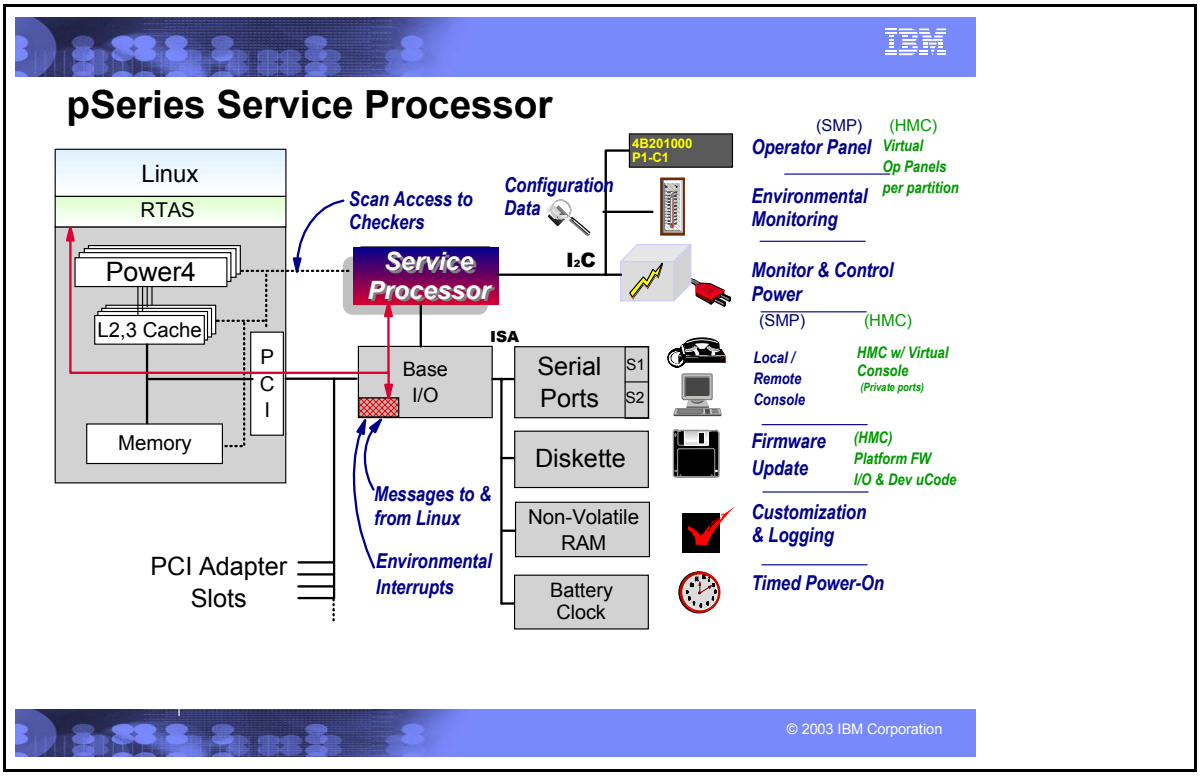
In pSeries systems, all error checking mechanisms, including parity, ECC, and control checks, have three distinct but related attributes. First, checkers are designed to provide data integrity. Second, checkers initiate appropriate recovery mechanisms, like ECC correction based on hardware detection of a non-zero syndrome in the ECC logic, to firmware executing recovery routine based on parity detection. Third, and equally important, all error check stations have been placed in the data and control paths of pSeries systems to deterministically isolate physical faults based on run-time detection of each unique failure that may occur.

All error checkers are instrumented with software readable error capture Fault Isolation Registers (FIRs) and blocking logic so that for every detected error the error is recorded only by the first checker that encounters it. This form of instantaneous run-time diagnostics greatly enhances other forms of diagnostic testing, such as BIST, which relies on reproducible defects, rather than intermittent ones often present or evident only at run-time. Run-time error diagnostics are deterministic, in that for every check station, the unique error domain for that checker is defined and documented. Diagnostic validation consists of dynamic run-time injection of intermittent error conditions, to determine that the correct physical component is called out by the diagnostic.

Role of Service Processor in run-time diagnostics

The role of the service processor in FRU isolation is similar to that in IBM 308X, 3090, and 9021 machines.

The service processor is a separate, independent processor that provides hardware initialization during system IPL, operational monitoring of environmental and error events, and maintenance support for the pSeries systems. For run-time diagnostic purposes, the communication between the service processor and the system consists of (1) Attention signals from the system hardware and (2) Read/Write communication between the service processor and all hardware internal FIRs, using specialized JTAG ports between the service processor and all system chips. This diagnostic Read/Write capability of hardware error registers is simultaneous, asynchronous, and transparent to any system activity running on system. These FIRs are known only to the service processor, and are not accessible by system software.



The system is designed to generate an appropriate Attention signal to the service processor when an error is detected in hardware. The ultimate response of the service processor is to Read the appropriate FIR, based on analysis of the "Who's On First" (WOF) structure, and to examine the active FIR bits, and post the FRU callout in the system NVRAM. The NVRAM acts as a mailbox between service processor, system firmware and the Linux running on the system. The defined FRU callout is moved by system firmware into the Linux System Error Log, along with notification about the nature of the event, usually a deferred repair. Following the analysis of a recovered event, the service processor resets the FIRs so they can accurately record any future error events.

Redundancy for array self-healing

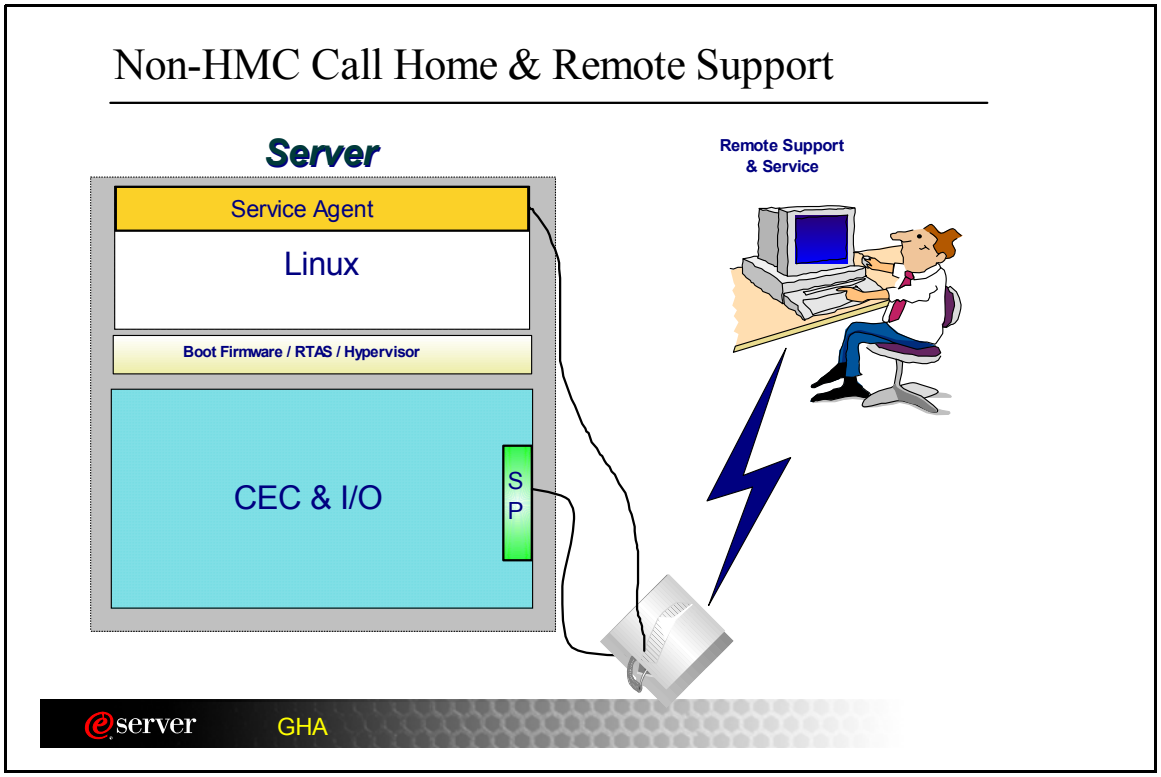
While the most likely failure event in a processor is a soft single bit error in one of its caches, there are other events which can occur, and which need to be distinguished from one another. For the L1, L2, L3 caches and their directories, hardware and firmware keeps track of whether permanent errors are being corrected beyond a threshold. If exceeded, a deferred repair error log is created. L1, L2 caches and L2, L3 directories on the POWER4 chip are manufactured with spare bits in their arrays which can be accessed via programmable steering logic to replace faulty bits in the respective arrays. This is analogous to the redundant bit-steering employed in main store as a mechanism to avoid physical repair that is also implemented in POWER4 systems. The steering logic is

activated during processor initialization and is initiated by the Built-In-System-Test (BIST) at Power On time. L3 cache redundancy is implemented at the cache line granularity level. Exceeding correctable error thresholds while running causes invocation of a **dynamic L3 cache line delete function**, capable of up to 2 deletes per cache. In the rare event of solid bit errors exceeding this quantity, the cache continues to run, but a message calling for deferred repair is issued.

Serviceability - Effective Problem Resolution

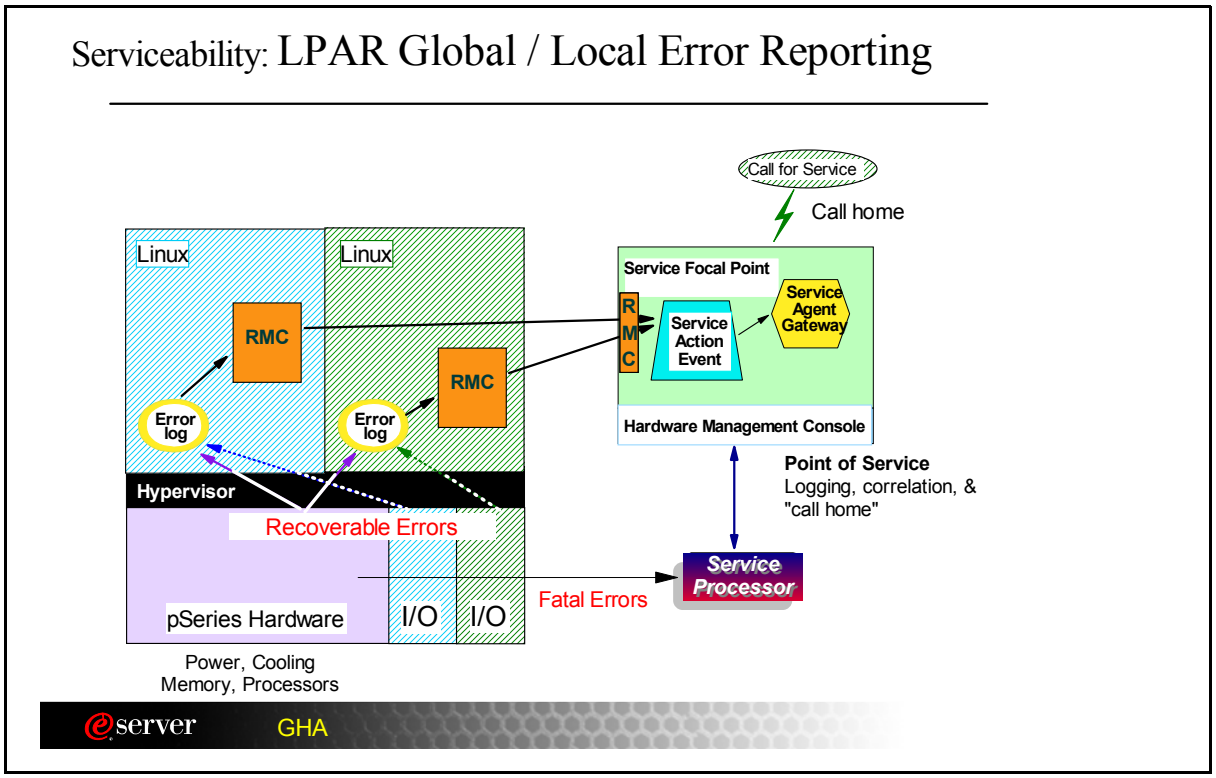
The pSeries systems are designed to be installed and maintained by both customers and trained service representative. Mid to entry level systems do support Customer Replaceable Units (CRU) which can be repaired by the customer.

Error Reporting: There are two main components of the pSeries error reporting strategy. The first component is the service processor and the second is Service Agent (planned to be available in 2H'04). These two components provide reporting capabilities on unexpected changes in the system environment. In a non-HMC system environment, a modem attached to the service processor and accessible to the Service Agent application running on the operating system is used to report errors to IBM for service. See the non-HMC environment diagram below:



In the environment where the system is attached to a Hardware Management Console, the errors are reported to the Service Focal Point (SFP) application running on the HMC. In the HMC-attached environment, the SFP application is the third component in the error reporting strategy and provides the error filtering, extended error data gathering and call home capabilities to report the service action request to the service provider as well as notifying the customer. A high level overview will be provided first with more detailed specifics on each component to follow.

System failures that prevent the system from coming back to an operational state (ie. operating system inoperative) will be handled and reported by the service processor. If running in the non-HMC environment, the service processor will report the errors directly using the attached modem. In the HMC environment, the errors are reported to the SFP application by the service processor via RMC.



System failures that do not prevent the system from coming back to an operational state (recovered through hardware, firmware or software techniques) will be reported by Service Agent (in non-HMC environment) directly using the modem or through the RMC component in an HMC environment since the operating system is operational.

In either case, if the modem is attached and configured appropriately then system failures will be reported to the service provider for service action (automated call home). Otherwise manual intervention is required.

Service Processor

The service processor provides for excellent RAS service features such as First Failure Data Capture analysis explained in the prior availability section and surveillance monitoring described previously. It also provides functions such as; power-on/off of the system, reading the service processor and POST error logs, reading vital product data (VPD), changing the boot list, viewing boot sequence history, and changing service processor configuration parameters, all of which can be performed remotely. Customers can enable console mirroring on the system console so they can monitor all remote console activity. For this option to work, a modem must be attached to one of the serial ports and configured appropriately.

Service Agent

There are two main components of the Service Agent application. The single partition mode version resides on the operating system and monitors the system while Linux is running. The Service Agent application monitors and analyzes all recoverable system failures, and, if needed, can automatically forward a service action event request to IBM through the serially attached modem or through the HMC-attached modem.

The second component of Service Agent is the gateway function and resides on the HMC as part of the SFP application and provides the consolidated service focal point to place a service call to the service provider (automated call home). Automated call home for maintenance can reduce the amount of downtime experienced in the event of a system component failure by giving the service provider the ability to view the error report entry, and if needed, order any necessary replacement parts, prior to arriving on-site. The opportunity for human misinterpretation or miscommunication in problem determination is mitigated.

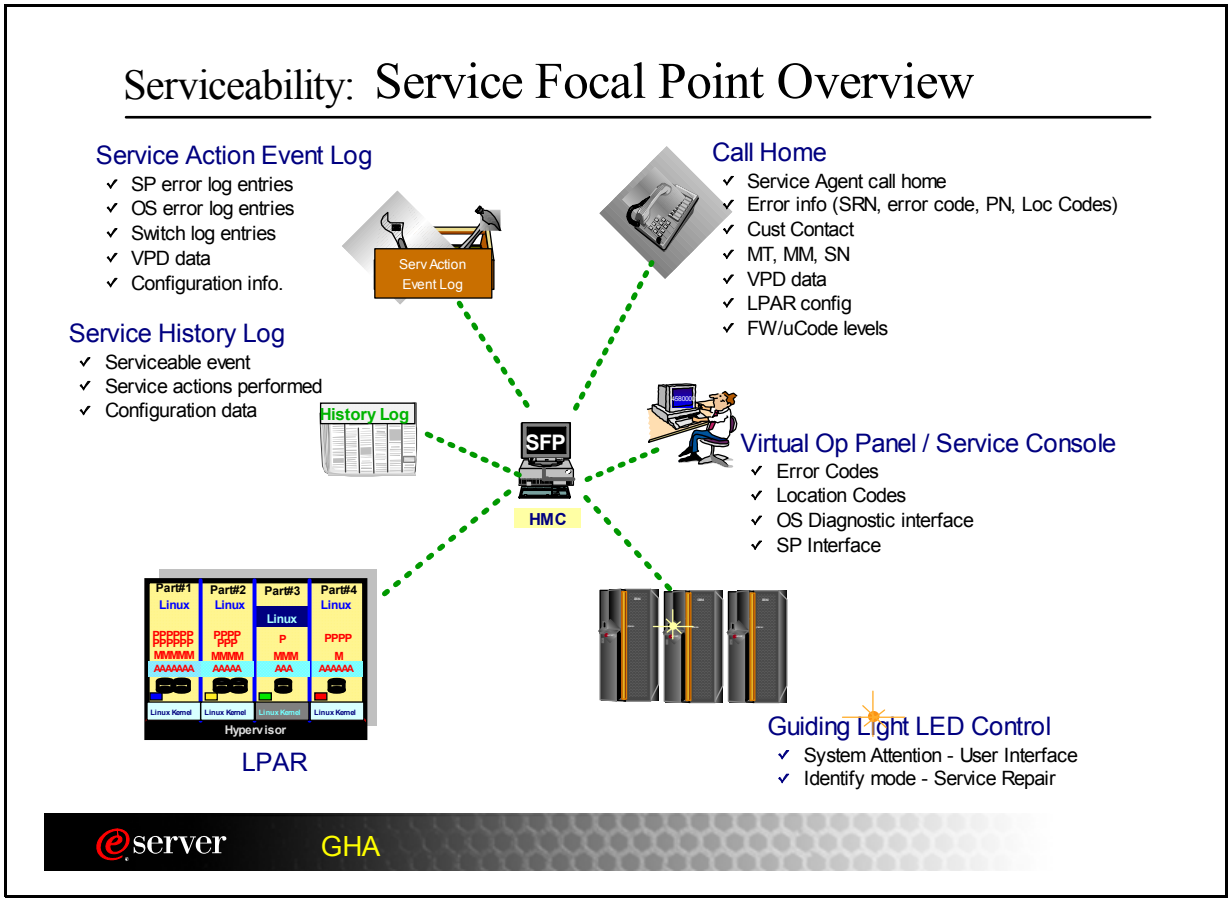
Service Agent is designed to automatically report problems based on default settings. The customer may modify the default values to prevent Service Agent from placing a service call during hardware upgrades, testing, or in the event that the failed component is not covered by an IBM Service Agreement (e.g., a third-party disk subsystem). The customer can also configure the product to only alert personnel within the customer's IT department via e-mail/telephone/pager. This function can be configured instead of, or in combination with, the ability to automatically place a service call to the IBM Service Center.

Service Agent is driven on a timed cycle (by default, set to one hour). At the completion of the cycle, the Service Director awakens to check diagnostic results, and failed or pending transmission events. In the event of an entry in the Linux Error Report, however, the Service Agent is designed to automatically start actions to prepare and send a request for service.

If the customer uses IBM for call out, the IBM Service Center receives the machine type, serial number, host name, Service Request Number (SRN), extended error data such as

configuration information and a problem description (taken directly from the failing machine's error report). The Service Center analyzes the problem report and determines whether service action is necessary. The Service Center will also determine if any hardware components need to be ordered to complete the service action.

The Service Agent Gateway code on the SFP application also gives the customer the option to establish a particular SFP and HMC system as the problem reporting server. A single SFP/HMC , accessible over the customer network, can be used as the central server for all other machines on the local LAN who are running the Service Agent client



application (a second system can be configured as a backup Service Director to the central server). If the Service Agent application on a remote client decides a service request needs to be placed, it forwards the pertinent information to the Service Agent gateway server who dials the service provider from its locally attached modem.

Service Focal Point

LPAR environments usually add complexity to servicing, but the pSeries systems ship with software to reduce this complexity. In order to accommodate error reporting, analysis and repair in the LPAR environment, a new application was developed to run on the HMC. This application is called the Service Focal Point (SFP) and leverages the design capabilities of the HMC to provide equivalent "virtual" function to the current capabilities presented by physical operator panels, service processor TTY menu interfaces and system firmware interfaces as well as capability for configuring/reconfiguring system resources into partitions.

The Service Focal Point is a system infrastructure which manages serviceable event information for the system building blocks. It consists of resource managers that monitor and record information about different objects in the system. It is designed to filter and correlate events from the resource managers and initiate a call to the service provider when appropriate. It also provides a user interface which allows a user to view the events and perform problem determination. When a problem is corrected the user can record actions that have been taken to resolve the hardware problem. This stored data can then be accessed by service representatives on future calls to determine what actions have already been taken on the system and adjust the service action plan accordingly. These features of SFP support the overall problem management strategy in a complex system.

The SFP application receives service action events from the service processor for critical system down situations, and from the RMC components running on the individual logical partitions for system recoverable or predictive events as well as operating system or device driver detected events.

Service Action Event Log

The SFP is designed to collect the serviceable events from different building blocks together in a Service Action Event (SAE) log. The log entries are generated by analysis routines that run once an error has occurred in a building block. The resource manager for the building block forwards information about the event to the service focal point and the information is placed in the SAE log. The particular content of the error data depends upon the type of the error and on the system configuration itself.

The SAE log on the SFP also contains pointers to extended information that may have been recorded at the time of a serviceable event by the building block. Extended error collection includes not only the collection of First Failure Data Capture, but also vital product data, partition information, operating system error logs, service processor error logs, error register data, etc.

When the SFP receives a new log entry, filtering is done to determine if this is a unique event. The filtering is done because sometimes an event notification can come from more than one resource manager for the same event or a resource manager may forward a notification for an event which previously occurred but has not yet been corrected.

Service Agent Component

When a service action event is logged in the SFP, the system needs to communicate the failure back to the service provider. During this "call-home" function, particular error data and system configuration information needs to be sent to the service provider to drive the service delivery infrastructure. The SFP utilizes the Service Agent Gateway application residing on the HMC along with the HMC modem to initiate the call home and transfer the pertinent error information to the service provider. When a call home is required, Service Agent manages the connection to the service provider which is used to open a problem record. The problem record is used by the service delivery team to determine whether or not to dispatch a customer engineer (CE) with the appropriate service parts to perform a repair.

When a CE does perform a repair on the system, the SFP is used to identify the source of the problem and record information relating to the repair. When the CE has performed a repair, the SAE log entry will be updated with FRU replacement information and any comments that the CE has. The information stored by the SFP represents the system's service history and is used to help ensure proper maintenance over the life of the system.

Guiding Light Maintenance

To assist the CE in locating the correct system unit and I/O drawer that contain the fault requiring repair, the SFP will enable the capability to flash LEDs on the respective system unit, drawer and FRU that contain the fault.

Microcode Discovery Service

Microcode Discovery Service provides the capability to determine whether the system is at the latest microcode and firmware levels. Using a secure Internet connection and a Web browser, Microcode Discovery Service captures the machine data and generates a real-time comparison report showing subsystems that may need to be updated.

The Inventory Scout application will run as a daemon on the server to accomplish this function. The tool will create a file containing the current level of all microcode (adapters, devices, system, and support processor) levels in the system. This file will be used to compare the system level codes against the latest available levels on the IBM Web site (http://techsupport.services.ibm.com/server/mdownload/). A report is then generated identifying any new updates available along with a link from which the updates can be downloaded.

Flash updates for firmware have been designed to be installed by a customer. The microcode updates are available from a support page on the Internet. From there, the code is downloaded to the server and installed. There are several options for downloading the updates, which include downloading to Linux workstations as well as DOS, OS/2®, or Windows® operating system-based PC workstations. Also, the update can be downloaded directly to the system and then installed. This capability allows the update to be performed remotely. Service aids or OS command line options can be used to install the updates.

RAS and Performance

The pSeries systems have many RAS features and functions (as specified in this white paper and *"IBM @server pSeries 630" and or "IBM @server pSeries 690 Availability Best Practices"* white paper. (http://www-1.ibm.com/servers/eserver/pseries/hardware/whitepapers/) Depending on customer selectable RAS feature enablement, deconfiguration of certain resources during boot or runtime may affect overall system performance. Notification is provided to the customer, and if optionally configured, to IBM Service, when such deconfiguration is invoked, but the customer can choose at any time to interrogate the hardware status of their system by running diagnostics either concurrently or in a stand-alone manner to review any outstanding service requests. On HMC controlled systems, they can also interrogate the Service Focal Point to see if there are any open service requests. More detailed information on how to perform these functions can be found in the pSeries User's Guide for their specific product.

Service Support

The service and technical support structure for pSeries systems reflect the importance that each product offering will play a key role in any business.

Hardware service requests will go to IBM's remote support center for initial problem diagnosis. This approach provides more direct access to skilled specialists. These specialists can either solve the problem over the phone or help get it resolved as quickly as possible by identifying the failing part or component and the specific skills required to resolve the problem. Service specialists are backed up by a Product Engineering team that has been highly trained and provided with additional tools to assist in problem identification and resolution.

RAS Conclusion

The RAS features and functions designed into the base system, extended by features such as RAID controllers, the HACMP program product, and remote service capabilities, combine to make the pSeries systems that will meet the needs of a mission-critical marketplace, and provide the growth, expansion, and performance required by our customers.

IBM Corporation
Marketing Communication
Server Group
Route 100
Somers, New York 10589

Published in the United States of America
11-02