Reliability, availability, and serviceability (RAS) of the IBM eServer z990

The IBM eServer™ zSeries® Model z990 offers customers significant new opportunity for server growth while preserving and enhancing server availability. The z990 provides vertical growth capability by introducing the concurrent addition of processor/memory books and horizontal growth in channels by the use of extended virtualization technology. In order to continue to support the zSeries legacy for high availability and continuous reliable operation, the z990 delivers significant new features for reliability, availability, and serviceability (RAS). This paper describes these new capabilities, in each case presenting the value of the feature, both in terms of enhancing the self-management capability of the server and its availability.

M. L. Fair
C. R. Conklin
S. B. Swaney
P. J. Meaney
W. J. Clarke
L. C. Alves
I. N. Modi
F. Freier
W. Fischer
N. E. Weber

Introduction

IBM eServer* zSeries* Model z990 introduces a new structure for zSeries servers. At the same time, the design ensures the required high availability and ease of growth characteristics. To increase availability, the z990 provides extensive "concurrent" service and growth capability. This means that the service action, configuration change, or hardware/microcode enhancement can take place while the server continues to process the customer's workload. The new structure includes up to four "books," interconnected by two active redundant Level 2 (L2) cache rings as shown in Figure 1. Each book is independently powered and contains a multichip logic module (MCM) with facilities for cooling, two memory cards, and three I/O (input/output) connectivity control chips. In addition, the z990 introduces a new multiple logical channel subsystem (MLCS) architecture, breaking through the previous 256-channel boundary.

The z990 server offers up to 32 customer-configurable processors, eight dedicated system assist processors (SAPs), and eight dedicated spare processors distributed across four books. Each book has eight processor chips, and each chip contains either one or two processor cores. The multibook structure consists of four book slots, as shown in Figure 1(a), numbered and populated in sequence B0, B1, B2, and B3. When "jumper" cards are used, their slots are designated, in later figures, with a leading J rather than a B. A jumper is a passive card with no active electronic devices, providing connectivity to the book structure. Figure 1(a) shows the front and rear views of the processor cage, indicating the locations of the

MCMs, distribution converter assemblies (DCAs), cage controllers (CCs), also called flexible service processors (FSPs), oscillators, and optional external time reference (ETR) cards. The z990 Model A08 contains a book in slot B0 and three airflow cards in the other slots. The Model B16 contains books in slots B0 and B1, with jumper cards in slots J2 and J3. The Model C24 contains books in slots B0, B1, and B2, and a jumper in J3. The Model D32 contains a book in each slot.

The RAS design features, such as *concurrent book add* (defined below), the fault-tolerant book-interconnect facility, processor and SAP sparing in both single-book and multibook environments, memory key array sparing, MLCS flexibility and I/O sharing, power/hybrid cooling, technology enhancements, and quality improvement measures for Licensed Internal Code (LIC), are described in the following pages. Also described is the IBM service and support structure, which has been enhanced to address this new design for repair, upgrade, and problem determination.

Processor subsystem

Concurrent book add

The z990 provides the capability of concurrently upgrading the server model by adding the second, third, or fourth processor book, which increases physical processors,

Note: For the reader's convenience, acronyms used in this paper are expanded in an appendix at the end of the paper.

Copyright 2004 by International Business Machines Corporation. Copying in printed form for private use is permitted without payment of royalty provided that (1) each reproduction is done without alteration and (2) the Journal reference and IBM copyright notice are included on the first page. The title and abstract, but no other portions, of this paper may be copied or distributed royalty free without further permission by computer-based and other information-service systems. Permission to republish any other portion of this paper must be obtained from the Editor.

0018-8646/04/\$5.00 © 2004 IBM

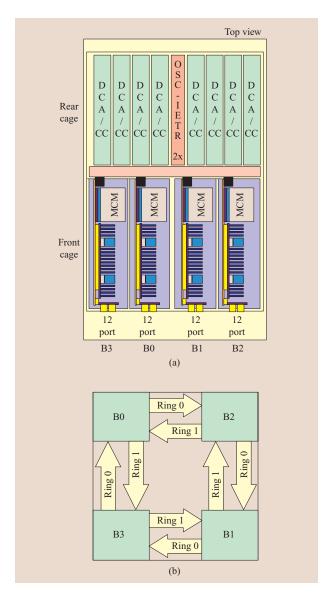


Figure 1

zSeries z990 multibook structure: (a) Physical layout; (b) logical layout.

memory, and enhanced self-timed interface (eSTI) high-speed I/O links.

The concurrent book add (CBA) function allows new physical hardware to be integrated into the system without affecting ongoing customer workload. This capability is an extension of the concurrent permanent upgrade function introduced in previous-generation servers [1] to add the new processors, memory resources, and high-speed I/O links seamlessly to the system. This new capacity is supported by the Licensed Internal Code configuration control (LICCC) technology [1].

Hardware verification tests such as logic built-in self-test (LBIST), array built-in self-test (ABIST), and memory self-test are performed during the CBA process to verify that the new hardware is properly installed and operating correctly before it is integrated into the configuration. Voltages, supplied by the book DCAs, are applied to the logic, and verification tests are executed while the new book remains fenced (electrically isolated) from the running configuration. Only when all verification tests are completed is the new book unfenced and added to the configuration. If a problem is found with the new hardware, the model upgrade is terminated, and the server is restored to its original configuration without affecting the customer's running workload. The model upgrade is rescheduled after the problem is understood and corrected.

The LICCC record describes the amount of new memory to be added, along with the number of new central processors (CPs), internal coupling facilities (ICFs), integrated facilities for Linux** (IFLs), and system assist processors (SAPs) that the customer has defined and that are to be allocated when the new processor book is concurrently added. The minimum SAPs (two) are allocated first to available processing units (PUs) on the new book, and any remaining new SAPs are balanced across the existing books. New CPs, ICFs, and IFLs are allocated using the same PU assignment algorithm used during the initial microcode load (IML) sequence, where CPs are clustered together separately from ICFs and IFLs in order to maximize system performance. However, since no rebalancing of processor resources occurs as part of a CBA operation, it is possible that the processor allocation assignment will be different after a subsequent IML.

Once the hardware is installed and integrated into the configuration and the new memory and processor resources have been allocated, the Processor Resource/Systems Manager* (PR/SM*) is informed of the new physical memory increments and newly available processors. These resources are then added to existing workloads or used to start new workloads. Some level of preplanning must be done by the customer to ensure that the new resources are used effectively.

The CBA function supports concurrent upgrade from one model to the next (e.g., A08 to B16, or B16 to C24) in the vast majority of cases. Upgrades to three- and four-book models may be disruptive depending upon the levels of the book hardware. An upgrade that would add two or more processor books can be performed as a series of concurrent single-book add operations; otherwise, the upgrade must occur disruptively.

The sequence of steps involved in the CBA function can be represented by a number of distinct concurrent operations.

1. Physically install the new hardware.

The new hardware is installed as described in **Table 1**. All of the upgrades require the second modular refrigeration unit (MRU) to be installed prior to the start of the CBA operation. This is also done concurrently.

If the third (B16 to C24 upgrade) or fourth (C24 to D32 upgrade) book is being added concurrently, before the new hardware is installed, the machine configuration is temporarily changed from a closed L2 ring structure to an open L2 ring structure in preparation for the concurrent removal of the installed jumper card, which will be replaced with the new processor book.

The service element (SE) code guides the system service representative (SSR) through a series of instruction panels describing the locations of the new hardware, and also turns on slot indicator light-emitting diodes (LEDs) on the light strip. If a problem occurs during the hardware installation, the backout and restore procedures are invoked. These procedures are also documented through panels on the SE and through lighting of LEDs on the light strips.

- 2. Apply power to the new processor book hardware.

 This operation starts after all new hardware is physically plugged into the server. The newly installed DCAs are brought up to standby power, the flexible service processors (FSPs) contained within the DCAs are booted, and logic power is applied to the new processor book.
- 3. Perform hardware verification tests on the new processor book.

The same hardware verification tests performed during a normal IML sequence (i.e., ABIST, LBIST, memory self-test, intrabook interface tests, proper cooling levels) are performed on the newly installed book hardware. These tests are performed while the new book remains fenced from the others. If any problems are found during these tests, the upgrade is terminated and the SSR is guided by panels on the SE to remove the defective hardware.

4. LICCC record validation.

After the successful completion of the verification tests on the new book hardware, an asset protection check is performed to verify that the new LICCC record, which describes the customer-purchased resources, has not been corrupted and that it matches the new book hardware. If a problem is detected during this phase, the concurrent book add operation is terminated.

5. Concurrently integrate the new processor book hardware into the system.

Once the integrity of the new book hardware has been verified and the LICCC record has been validated, the interfaces between the clock chips on the new and

Table 1 CBA hardware installation.

	Model upgrade (from/to)		
	A08/B16	B16/C24	C24/D32
New processor book	P.4	D 2	D2
installed in slot no. Jumper cards installed	B1	B2	В3
in slot no.	J2 and J3		
Jumper cards removed			
from slot no.		J2	J3
DCAs (2) installed for book slot no.	B1	B2	В3

existing books and the L2 ring interfaces between the new book and existing adjacent books are unfenced, and the new book is added to the configuration.

6. Concurrent CP/SAP/ICF/IFL upgrade.

Once the new book hardware has been integrated into the server, all of the physical processors on the new book will have been initialized as spare processors. These spare processors are now part of the pool that can be used for concurrent upgrade operations. Using the number of CPs, SAPs, ICFs, and IFLs defined in the LICCC record for the new book, the processor resources are now concurrently allocated.

7. Concurrent memory upgrade.

At the time the new book is integrated into the server, the physical memory on the new book will have been tested and initialized. PR/SM code, which manages the memory allocations, is now informed of the available new physical memory increments. Using the LICCC record for the new book to determine how much of the installed physical memory is available for use, a concurrent memory upgrade operation is started to add the newly defined physical memory from the new book to the available physical memory increment pool.

8. Concurrent I/O upgrade.

After the new book is integrated into the server, the eSTI ports on the book are made available for use in connection with I/O devices.

The CBA function extends the z900 concurrent upgrade strategy for processors and memory, utilizing the LICCC technology to support concurrently adding physical hardware in the form of a book, where the upgrade capability was previously limited to enabling the use of pre-existing nonconfigured hardware resources. The function allows customers to add capacity to their z990 without taking an outage to install physical hardware.

The z990 has the capability to add SAPs concurrently with a concurrent book add, a concurrent temporary capacity upgrade, or a concurrent permanent processor

upgrade. With the latest level of z990 code, these new SAPs will handle I/O activity from existing I/O or from newly added I/O.

Central processing unit (CPU) sparing

The z990 processor chip contains one or two processor cores. Because of chip pin limitations, the interface to the off-chip L2 cache is shared between the two processor cores. The sharing of the L2 interface requires special handling for the processor recovery and CPU sparing functions.

The recovery strategy for zSeries processors is to maintain an architectural checkpoint on every hardware instruction boundary so that in the event of an error, in-flight operations can be purged, the last checkpoint restored, and the instruction processing resumed from the last checkpoint. The clearing of in-flight operations includes requests to the L2 cache. This clearing of requests to the L2 requires a reset of the interface. For the single-core processor chip, this is handled in the same manner as in the z900 server [2]. However, since both processor cores share the interface, when either processor requires recovery, the other processor core must also go through recovery.

Successful processor recovery implies that the architectural checkpoint was restored, and the processor is able to resume instruction execution. When a hard logic failure in a processor core prevents it from recovering successfully, it must *checkstop* (i.e., halt all processing). In most cases, the architectural checkpoint is still intact. In this case, the checkpoint is extracted from the defective processor and "transplanted" into a dormant, spare processor, and instruction execution is resumed on the spare processor. This method of recovery on a spare processor is called dynamic CPU sparing (DCS).

When a processor has a known logic failure, the zSeries reliability strategy is to stop using the entire processor, which includes the interface with the L2 cache. For the single-core processor chip, this is handled in the same manner as for the z900 server [2]. For the dual-core processor chip, since both cores share an interface with the L2 cache, both cores must checkstop. There are very few processor states which are not fully recoverable. Such states generally involve pending updates to storage, as storage updates must be coherent across the entire server. For example, if a processor has pending storage updates that have not yet been sent to the coherent L2 cache and that processor is checkstopped, the storage update and storage coherency will be lost. In these cases, the higher levels of firmware and operating system recovery can sometimes still recover from this bounded storage corruption.

The DCS implementation for multiple processor cores involves a significant volume of data which must be extracted from the checkstopped cores and transplanted into the spare processor cores. The DCS design objective is to be completely transparent to higher levels of firmware and the operating system. Since instruction processing is stopped until it can be resumed on the spare processors, the time it takes to complete the sparing must be short, so that the higher-level firmware and operating system are not affected by the temporary loss of computing capacity.

In a checkstopped processor, the only access to the register checkpoint data is via a scan operation. Scanning is normally done by the SE code, which controls a scan engine on the clock chip in the MCM. Because the service processor is very slow compared with the z990 processors, a significant amount of time would be required to log out data from multiple processor cores. To overcome this delay, the z990 processor chips implement an interface to the clock chip which allows millicode running on a healthy processor chip to control the scan engine on the clock chip. Thus, millicode on a spare processor can directly access the checkpoint data from a checkstopped processor without relying on SE code. This firmware implementation is significantly faster than that of previous generations, even with the additional processor core being spared.

This "dual-core" DCS implementation has other implications with respect to the system. Since two processor cores must be spared on a checkstop condition, there must be two spare processor cores available in the initial configuration in order to avoid any loss of capacity after a sparing action. Thus, every z990 initial configuration includes two spare processor cores per book. These two spare processor cores are not necessarily configured on the same chip. Although a checkstop affects the cores on the same chip, the transplanting of the checkpoint into a healthy core is on an individual processor core basis so that the spare cores can be on different chips.

In the event of processor failure, sparing takes place according to the following algorithm:

- 1. If one is available, use a spare PU on the same book as the checkstopped PU.
- 2. If a spare PU is not available on the same book as the checkstopped PU, use a spare PU on a book adjacent on the ring interface to the book containing the checkstopped PU.
- 3. Use any available spare PU in the server.
- 4. If no spares are available, invoke the processor availability feature (PAF) to recover the application (running at the time of the failure) on another running CP and invoke a service call for book replacement.

Storage subsystem - Cache ring interconnect

Notice from the logical layout in Figure 1 that the books are interconnected via the redundant L2 rings to form a cache-coherent symmetrical multiprocessor (SMP). One ring passes through books $B0 \rightarrow B2 \rightarrow B1 \rightarrow B3 \rightarrow B0$ and the other through books $B0 \rightarrow B3 \rightarrow B1 \rightarrow B2 \rightarrow B0$. As described earlier, Models B16, C24, and D32 are respectively configured with two, three, and four books, along with their necessary jumper cards, to form "closed" systems. They are considered closed because both ring 0 and ring 1 form a complete loop in these configurations. Along with Model A08 [Part (a)], these closed configurations for Models B16, C24, and D32 [Parts (b)-(d)] are shown in Figure 2. Each of these configurations operates as a tightly coupled SMP. However, there are protocols to help reduce the amount of ring control and data traffic for performance optimization.

Interface communications

Each book contains, on its MCM, an L2 cache structure, the system control element (SCE), consisting of a storage controller control (SCC) and four storage controller data (SCD) chips. For interbook L2 ring communication (Figure 3), each SCC communicates with the other SCCs, while each SCD communicates with corresponding SCDs.

Each ring segment on each book-to-book interface has an "elastic interface." This means that the interface compensates, at startup time, for the latency of the logic circuits and packaging so that the controls and data lines are calibrated to the topology of the interface. Once calibrated, the interface protocols operate on the basis of the calibrated latencies.

Interface ECC

For each ring segment on each book-to-book interface, the controls and data bus have single-error correction/double-error detection (SEC/DED) error-correction codes (ECCs). This means that all single-bit errors are detected and corrected on-the-fly, without loss of latency or performance. Uncorrectable errors (UEs) are detected but cannot be corrected. In this case, a tag bit is set and sent along with the data to indicate that the data is not usable.

Threshold/call home

An error indicator is logged out and causes a service "call home" if the number of CEs on an interface is greater than the predefined acceptable rate. (A *call home* is the automatic call from server to IBM support structure with error data, server status, or other service-related information.) For the z990, the interface threshold call takes place if the rate is at least one CE in each of three consecutive minutes. Therefore, if a bit fails permanently, the threshold is exceeded quickly, the call for scheduled

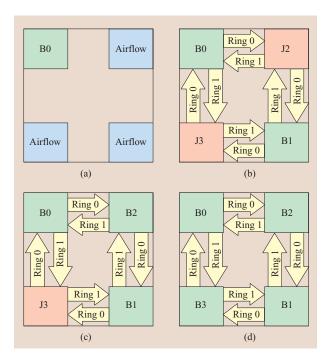


Figure 2

zSeries z990 standard configurations: (a) One-book; (b) two-book closed; (c) three-book closed; (d) four-book closed.

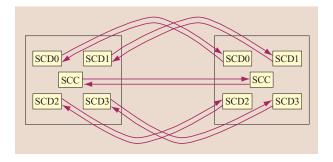


Figure 3

Chip-to-chip interbook communication.

service is placed, and a replacement book is ordered. Meanwhile, the CEs continue to be corrected.

If a data UE occurs on the L2 ring interface, it may be recoverable. Upon being received by a processor or SAP, a UE is purged out of the cache and refetched from memory. It is likely that the retry data will be successful if the memory copy is not corrupted and the L2 ring segment is not a solid 2-bit breakage. It is expected, since single-bit failures tend to occur long before a second bit fails, that a call home will have occurred and the part will have been replaced well before an L2 ring UE can occur.

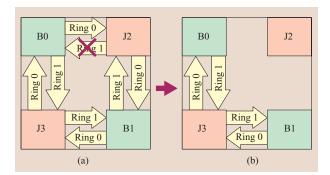


Figure 4

Example of two-book configuration degrade: (a) Closed; (b) open.

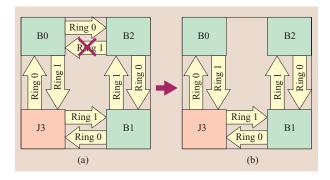


Figure 5

Example of three-book configuration degrade: (a) Closed; (b) open.

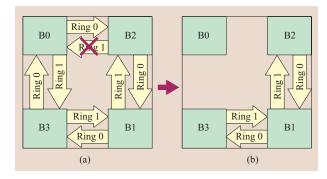


Figure 6

Example of four-book configuration degrade: (a) Closed; (b) open.

Control UEs, checkstop/degrade

Uncorrectable errors on the control signals result in an immediate system checkstop. Error indications help determine which book has experienced the failure. During

the next IML after the system checkstop, the server is configured in an "open L2 ring" configuration. This means that the ring 0 and ring 1 segments between the two books that experienced the checkstop detection are not configured, but since the books remain in the configuration, there is no loss of resources.

The next two sections describe, in more detail, the various scenarios of ring degrade capabilities.

Two-book degrade

In a two-book or three-book configuration with jumper card(s), the entire path from the driving book, through the jumper, and into the receiving book is covered by the same error checker. When a segment is deemed to be broken, the hardware is reconfigured to exclude the broken segment. Also, when a ring segment is taken out of the configuration, the corresponding segment for the other ring is also removed to allow for consistent architectural operation.

Using the two-book configuration as an example [Figure 4(a)], assume that an error has occurred on the segment from jumper card 2 into book 0, on L2 ring 1 (red X). The L2 ring segments from jumper card 2 to book 0 (L2 ring 1) and book 1 to jumper card 2 (L2 ring 1), as well as their corresponding L2 ring 0 segments, book 1 to jumper card 2 (L2 ring 0) and jumper card 2 to book 1 (L2 ring 0), are removed from the configuration. The resulting configuration [Figure 4(b)] has book 0 and book 1 as "end books" with jumper card 3 between. When a similar approach is used on a three-book configuration [Figure 5(a)], the resulting configuration [Figure 5(b)] would have book 0 and book 2 as "end books" with jumper card 3 and book 1 between. In each case, there is no loss of resources. This degrade capability exists for the majority of configuration cases, depending on the level of the book hardware.

Four-book degrade

In a four-book configuration [Figure 6(a)], a similar L2 ring 1 error on the segment from book 2 into book 0 removes the L2 ring segments from book 2 to book 0 (L2 ring 1) and book 0 to book 2 (L2 ring 0) from the configuration. However, because of performance and operability issues, a four-book open-ring structure is not supported. Therefore, the book most likely to fail (book 0 in this case) is also removed from the configuration. The resulting configuration [Figure 6(b)] has book 3 and book 2 as "end books," with book 1 between. In contrast to the two-node/three-node scenarios, resources are lost in this case.

This detailed high-availability design for ring and book degrade modes extends the zSeries RAS recovery strategy to provide the customer the flexibility of operating the server, in the presence of partial failures, in degraded

mode while awaiting the appropriate service action. Again, prior planning is required of the customer, in selecting reduced workloads and defining profiles and priorities, in order to best take advantage of this capability.

Storage subsystem—Cache/memory

The z990 greatly increases the size and flexibility of the storage subsystem while maintaining the strong reliability, availability, and serviceability characteristics of its predecessors. Logically, the storage structure is one large pool; that is, the cache and main memory are shared across the server, such that the z990 is one large server like its predecessor, the z900. Physically, the storage structure is hierarchical and spread across all the books.

The robust design for the Level 1 and Level 2 caches is carried over from the z900 server [2]. The Level 1 (L1) cache is a parity-protected, store-through cache on the processor chip featuring purge, delete, and relocate capability to recover from cache errors. The Level 2 (L2) cache is the shared, ECC-protected store-in cache on dedicated chips on the MCM, also featuring purge, delete, and relocate.

Each book contains two memory cards for the Level 3 (L3) arrays and the store-protect arrays. All processors share the L3 storage. The data is ECC-protected. Special UE codes exist for L2-detected failures and L3-detected failures; these codes aid in fault isolation. Memory is continuously "scrubbed" to remove correctable errors and detect uncorrectable errors. Each card is organized into four rows; each row has two spare dynamic random-access memory (DRAM) chips. These spare DRAMs are used during scrubbing and during IML to concurrently replace other DRAMs in the row that are over threshold on correctable errors. All DRAMs are scrubbed, including unused spare DRAMs. Unused spares that are over threshold are spared. The available storage is determined by summing the memory LICCC values from each book. Logical partition (LPAR) code then determines which physical storage is to be mapped to the available storage. All physical storage is available to LPAR to be used at its discretion and therefore is continuously scrubbed and checked. Concurrent memory upgrades are simplified compared with previous servers, since all physical storage is always available to LPAR. The L3 storage supports ABIST during power-on IML. Special power governor circuitry slows down accesses to L3 storage when the current drawn is too high. Typically, over-current occurs during the self-test of arrays. Having a power governor allows the design to have a higher nominal array access rate while protecting the arrays during extreme conditions. CBA performs an IML of the book which includes ABIST. CEs are spared, and UEs prevent the book from becoming available.

Store-protect arrays

The store-protect keys, which are distributed evenly across the two memory cards, form an architected facility which provides access control for each 4K block of storage. Each store-protect key is parity-protected with three-way voting and sparing. The array is accessed with normal key fetches. Each logical key has four physical copies; three copies are actively involved in the three-way voting, and the other is on standby as a spare. Four logical keys are fetched in one operation.

Store-protect cache

The store-protect cache is new in the z990 server. It is located in the L2 SCC chip and contains 128K store-protect entries for the local memory. Each 7-bit key entry is protected by five ECC bits. Two special UE codes are supported for fault isolation. The cache supports ABIST during power-on IML. A cache failure during ABIST causes the book to be called for repair. CBA performs an IML of the book, including ABIST, to detect defective caches.

I/O subsystem

While the z990 book structure provides for expanding processing capacity by adding new books to the configuration, the challenge faced by the I/O subsystem to move beyond the 256-channel limit had to be handled solely through the use of firmware and virtualization technology. Clearly, the sheer increase in the amount of hardware required to double or triple the number of I/O cages to support the I/O traffic and maintain redundant path connectivity would be detrimental to overall server reliability. In addition, there would be an increase in the power required and a significant increase in the z990 footprint. These issues were addressed by introducing higher-speed I/O interconnect (eSTI) and I/O cards, extending the internal high-speed HiperSockets* recently introduced in the z900 server [2], increasing the number of channel subsystems (CSS), extending the sharing concept of the Enterprise Systems Connection (ESCON*) multiple image facility (EMIF) to share I/O cards across the logical channel subsystems, and using only the high-density I/O cage introduced in the z900 server [1].

Hardware view

The channel subsystem structure comprises CPs, SAPs, MBAs, enhanced self-timed interface (eSTI) ports, and the high-density I/O cage. In the z990 server, the memory bus adapter (MBA) chips are no longer located within the MCM, as in the z900 server [2]. The three MBA chips, each providing four high-speed eSTI links, are separately mounted on the front-facing edge of the book, closer to

Figure 7

Channel subsystem—hardware view.

the eSTI connectors to improve the bandwidth. In addition, a single-MBA clockstop design was introduced in the z990 server to enable the isolation and fencing of a single failed MBA while the remaining MBAs continue operation over their high-speed links. In a single-book configuration, the eSTI link connections to the I/O domains are optimized for high availability by spreading them across the MBAs. In the multibook configurations, the high availability is extended such that the I/O domains are first spread across all available books and then across the MBAs within each of the books. This ensures that there are virtually no single points of failure which would result in the loss of I/O connectivity to a device in a properly configured environment.

The high-density I/O cages (maximum of three) introduced with the z900 provide up to 84 I/O slots

supporting ESCON, fiber connection (FICON*), Fibre Channel protocol (FCP), open system adapter (OSA) and cryptographic coprocessor cards (Figure 7).

Logical view

The z900 channel subsystem contains 256 channel path identifiers (CHPIDs) supporting 15 logical partitions [2]. Since, in the z990 server, the processor/memory books are logically viewed as one SMP, with up to 60 logical partitions, the multiple logical channel subsystems (MLCS) concept was introduced. The MLCS feature supports up to four logical channel subsystems (LCSSs), each supporting 256 CHPIDs. This allows the z990 system to maintain the same average ratio of CHPIDs to logical partitions as the z900 server and, for a partition requiring high connectivity, all 256 CHPIDs can be dedicated to it.

526

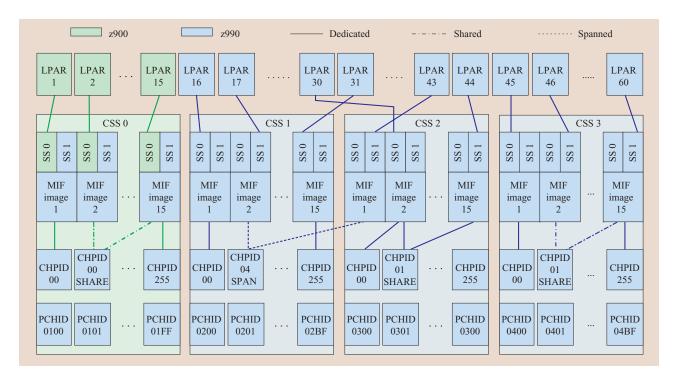


Figure 8

Channel subsystem—logical view.

The logical aspect of MLCS allows it not to be tied to any specific processor book. Thus, even in a single-book configuration, four LCSSs can still run, but with fewer physical channels, since it would be constrained by the number of eSTI ports.

Another feature introduced in the z990 server is the concept of spanning. Spanning is an extension of the EMIF shared-channel architecture used in previous generations; it allows a single FICON, coupling, or networking adapter to be shared among all 60 partitions if desired. This reduces the amount of hardware required to support the logical partitions and also reduces the hardware single points of failure among a finite number of channels. For example, 60 partitions would require a minimum of just two spanning channels to provide connectivity to all 60 partitions and still avoid a single point of failure, whereas, with EMIF, eight physical channels would be required, since EMIF supports sharing only within a single CSS (Figure 8). The spanning feature is not supported for ESCON. The ESCON channels can continue to be shared across 15 logical partitions within a single logical channel subsystem.

The z990 server also introduces the concept of the physical channel identifier (PCHID), which serves as a "handle" to map the relationship between the channel subsystem identifier (CSSID) and its CHPIDs to a physical

location in the I/O cage. The PCHID is a 2-byte value with a leading zero followed by an "XYZ" value (Figure 9). For example, PCHID 325 would refer to the front of the second I/O cage (located in the lower half of the I/O frame), slot no. 2 (the third physical slot from the left), and port no. 5 (the sixth port down from the top position) on that card. Although the PCHID number is a physical entity, it can change as a result of a service action. For example, an ESCON port-sparing action will cause a permanent change to the PCHID location. The PCHID is part of the information required by the service functions for problem resolution to the field-replaceable unit (FRU). The PCHID-to-CSSID.CHPID relationship is defined by the customer in the I/O configuration data set (IOCDS).

Concurrent upgrade

In the z900 server, the SAPs could not concurrently be added to the configuration [2]. However, in the z990 server the I/O subsystem can grow horizontally as part of the CBA function described earlier. Since the SAP affinity is defined during IML, the newly introduced SAPs can handle only those requests from new I/O added as part of the upgrade. In order to re-optimize the I/O distribution, a new function called *workload rebalancing*

527

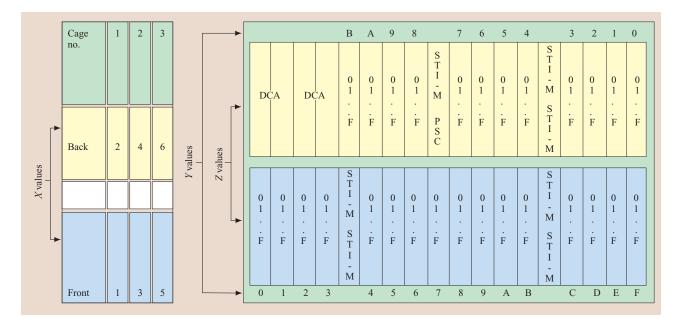


Figure 9

PCHIP-I/O cage definition.

enables the new SAPs to handle operations from existing I/O (available with the latest z990 code level).

The z990 server maintains concurrent I/O card upgrade capabilities similar to those of the z900 [2]. However, with the introduction of the MLCS, some additional preplanning is required. The following must be addressed prior to future growth, since these changes cannot be defined dynamically:

- 1. Number of LCSSs.
- Number of devices per LCSS. This is specified in the IOCDS.
- 3. Number of logical partitions per LCSS. The names of the logical partitions can be assigned dynamically. This latter capability is new for the z990 server.

It should be noted that preplanning other MLCS features, such as whether or not a running channel is spanned, is not required on the z990. As long as the channel is capable of spanning and is defined as shared, adding the spanning property to the channel via the MLCS-enhanced dynamic I/O feature does not require the channel to be varied offline.

Concurrent service

The z990 server maintains concurrent I/O card repair capabilities similar to those of the z900 [2]. However, the introduction of the MLCS has affected some functions. In order to maintain the problem determination for channel

card failures in the fields, the PCHID reassignment tool is introduced. This tool allows the PCHID-to-physical card location mapping to be changed by the SSR via the SE console. Thus, should an entire ESCON card have to be relocated in the I/O cage due to some failure which prevents that card slot from being used, those PCHIDs can be remapped to a usable card slot without having to redo the IOCDS.

Power/cooling subsystem

For the z990 server, a significant amount of the robust N+1 power system is retained from the z900 [2], but certain new elements are introduced to support up to a four-book server in a single frame. There is an increase of power for cages as well as at the total server level.

There are two system oscillator cards and two ETR (external time reference for coupling) cards, dual SEs, and a new DCA-X regulator to support additional current for each voltage level in the I/O cage. There are two FSPs per book and two each for the I/O cages. There is also an optional high-capacity internal battery facility (IBF) unit to serve as temporary backup if primary power is lost.

A new hybrid cooling system has been introduced for the z990 which operates above the dew point, eliminating the need for desiccant bags, dry air system, heaters, and humidity sensors as used in the z900. The evaporator on the MCM is eliminated as an FRU, which also eliminates the pressure check valve. There are one or two modular refrigeration units (MRUs) to cool the MCMs, depending

528

on the z990 model. The A08 has one MRU, and the other three models each have two MRUs. Each MRU controls up to two MCMs. The B16 model has a second MRU; thus, the cage is configured to grow to a three- or four-book model without adding MRU hardware. The MRU is controlled by the MCM evaporator thermistor. Standby redundant air-moving devices (AMDs) in the processor cage are used for mitigating MRU failures. The MCM temperature is sensed by three thermistors mounted on the MCM copper hat. In the event of an MRU failure, service is called, and the MCM temperature begins to rise. When the temperature threshold is reached, the high-speed backup AMD is activated. The service action on the MRU is performed concurrently, and the MRU resumes the cooling responsibility.

If the MRU fails, the backup cooling is invoked, and the processor cycle time is slightly increased in order to reduce MCM power. This phenomenon is known as *cycle steering*. It is invoked solely for MRU cooling failures, not for logic failures, and is triggered only by the temperature value sensed from the MCM hat thermistors. Two levels of performance degradation, 4% and 8%, are determined by the cooling requirements. All books are cycle-steered to the same cycle time.

Technology reliability

For maximum reliability, the z990 technology undergoes stress testing, burn-in, and system-level run-in. The stress test conditions include 1) a 20°C temperature test; 2) a port test to reduce defective single cells; and 3) a low-voltage test to screen cell fails. A low-frequency burn-in is done at 10 MHz at 1.8 V and 140°C temperature for 48 hours. A high-frequency burn-in is done at 640 MHz at 1.8 V and 140°C for six hours. System-level run-in is done at 1.45 V at 90°C for three hours. The burn-in and run-in screen most of the ac and dc defects that would otherwise escape to the field.

Service subsystem

Enhancements to the service subsystem address the new and unique aspects of the z990. These include the multibook structure, the expanded I/O subsystem, the coupling link changes, and the associated error reporting, problem analysis, error data handling, and call-home capabilities.

In the z990 server, the number of physical channels supported is increased to exceed the 256-channel boundary. The multiple-channel subsystem is introduced to support multiple sets of 256 channels addressed via CHPIDs.

Integrated cluster bus (ICB) link connections for legacy ICB links are moved from the processor cage to the I/O cages. The eSTI link from a book is connected to an eSTI2 or eSTI3 card plugged into an I/O cage, which enables support of ICB-2 and ICB-3 channel links. All of

these changes are supported by the maintenance package and, for service simplicity and consistency, provide the same appearance to service personnel as earlier zSeries servers.

FRU resolution is critical in the new multichannel subsystem (MCSS), especially in the repair scenario. To resolve generic FRU names such as "ANY-CHANNEL," addressing information is provided within the extension of the system reference code (SRC). To address a book within a processor cage, a new qualifier for addressing purposes was needed. The MCSS uses the new PCHID and CSS.CHPID information to address channel paths. The SRC header (64 bytes wide) is introduced. The SRC extension and a new second extension are part of the header to support the new hardware structures. The SRC header provides information for SRC control and navigation, such as

- 1. The time stamp when the SRC is created instead of when the event is logged.
- 2. Information for organizing the error event within the log file.
- 3. Information used by the display function to interpret the attached error data, known as first error data capture, or FEDC.

Three internal schemes address hardware functions and interface with the FRU resolution routine. They are the PCHID, the hardware address (HWA), and the slot number in the cage.

Each physical channel receives a PCHID assignment. Each PCHID number is unique within one configuration. The PCHID assignment is stored in the hardware object model (HOM). If an "ANY-CHANNEL" FRU has to be resolved, the SRC extension field must carry a valid PCHID number. The second extension provides information on the MCSS if applicable. The PCHID is used to query the HOM. The query returns all information to create the location information. It returns the location information within a cage and the unit ID for the cage. A call to the configuration manager with the unit ID returns the cage location.

The hardware address represents the topology of the eSTI links. It contains the book, MBA, primary eSTI link, secondary eSTI link, and I/O information. The HWA is unique. Any generic FRU within the eSTI link topology can be resolved using the HWA or part of the HWA. HWA information is stored within the HOM.

Each slot location within one cage that is able to hold an FRU (full-high, half-high, or quarter-high FRU/card) receives a slot number. This is a fixed assignment, valid for one type of cage. Generic FRU names such as ANY-SLOT require a cage unit ID and a slot number for resolution. This is valid for all cages but the processor

cage. The processor cage contains up to four books. The parameters to address a FRU via slot number are cage unit ID, book ID, and slot number. I/O cages have one book ID assigned, which is always "0." This allows one method of addressing a FRU using the slot number.

Extension fields which carry address information are called trailers. Each trailer identifies itself via the trailer number. The 81-trailer provides HWA information, and all other 8x-trailers provide PCHID information; 9x-trailers provide address information using cage, book, and FRU. Trailers are used as an interface between the FRU analysis function and the code to resolve a FRU to the physical location information.

Errors reported from FRU analysis routines are posted into the log file which resides on the SE. FRU analysis routines are part of the clock stop error handler, clock running error handler, channel error handler, power, FSP network, and other error-detection and error-recovery functions. Additional information is added to error events by the pre-problem analysis (prePA) routine. System status and available hardware resources are investigated, and status flags for PA are maintained.

Problem analysis detects errors or failures which require a service action. PA, with the first error event, opens a time window. All events received within the window participate in the problem analysis routine. PA selects one error event, the root cause of the problem, and reports it to the customer via panels on the HMC. The problem analysis panel provides information about the problem, the recommended corrective action, and the impact of the repair. The subsequent panel for requesting service provides problem details, problem data, and a parts list. With a simple mouse click, it calls home for repair. The normally selected setup enables the call home to be issued automatically, without intervention by the user.

The call home invokes the service organization so that the appropriate service actions may begin. With the call home, FEDC data for all events within the error-event time window are sent to the Remote Technical Assistance Information Network (RETAIN). The FEDC data contain a subset of the log file. This subset is merged from the current and backup log files, has a maximum size of 5 MB, and should contain no "pruned" information (that is, data which has been deleted because of space limitations). For the z990, the log file to store events (system and error) has been increased from 1.4 MB on the z900 to 10 MB of data. This significantly increases the data available to the IBM support structure to perform problem determination and root-cause analysis. Whenever the amount of data is close to the limit, the less necessary records are deleted (pruned) to gain space for new events. Backup files help to maintain system and error events.

Not all customer accounts provide a remote support facility (RSF) connection to call home in case of an error,

or the connection may be unavailable when needed. In all cases the call home is sent to "Virtual RETAIN," which is a resource on the SE to keep/store call-home data until the call home is performed or the data is gathered by the SSR. After successful data transfer, all data for the error event is deleted. The maximum amount of data that can be submitted to RETAIN is 14 MB. This is also an increase for the z990, as the previous server could submit only 8 MB. Not all FEDC data captured by the server at the time of the error can be part of the call home. The less critical data remains in Virtual RETAIN, including, for example, the complete log file, backup log files, and processor data dumps. Should it be required later for problem determination, this data can be collected by the SSR. After successful repair, the remaining FEDC data in Virtual RETAIN is deleted.

Close cooperation between the service organization and product development ensures that escapes from the maintenance package are reported, investigated, and corrected. These are cases in which the package fails to identify the correct FRU, potentially resulting in an extended outage. Updates to the maintenance package are delivered via microcode fix (MCF).

Enhanced data retrieval and analysis

zSeries availability is enhanced by enabling the support teams in product development and Product Engineering to proactively study the field performance of zServers to determine whether potential problems can be avoided by the use of data mining and follow-up action.

Data mining is the process of retrieving data from databases and other sources for use in a number of analyses. These include verifying past designs, determining actual performance for future designs, and executing preventive maintenance based on error log data or data retrieved from other sources. All zSeries data is retrieved during call-home operations. If the call home is for reporting a problem, the data retrieved is specific to that failure, and only that failure data is loaded into the Product Support System (PSS) database (see below). The majority of the PSS data used for data mining comes from the weekly survey call home, which transmits general server information and summary files. Additional files needed for analysis may also be retrieved during the weekly survey connection. This is achieved by modifying the file-selection criteria in RETAIN. The zSeries data mining process uses a number of different sources:

1. PSS database

PSS is an MVS* DB2* database, containing millions of lines of data, that is used as the main database for zSeries tracking. It contains worldwide inventory files, failure data, call home statistics, failure analysis

results, and reference code data not tied to a call-home situation.

2. Product description file

A database file in manufacturing called VPDTRAN stores all current vital product data (VPD) files and the server VPD history since manufacture. The VPD file is a listing of all of the hardware components in the z990 server. Whenever any component changes, a processor is spared, or a part is removed, added, or replaced, the VPD is updated and transmitted to RETAIN. RETAIN then forwards the VDP file to VPDTRAN. VPD is scanned daily for spared CPs and FRU exchanges. VPD is also used to determine minimum, maximum, and average server size. This is key to accurate product tracking and future design requirements.

3. Operational use information

Although the VPD file defines the hardware components in any server, many components serve multiple purposes. The IQZQTSAD.TRM file is a very versatile file that defines exactly how a server is being operated. It is updated every hour with performance data which shows usage capacity.

4. GOFER

GOFER (Get Other Files Easily, Remotely) [2] is used to retrieve other files needed to perform data mining. RETAIN can be set up, for a specific duration, to pull files that are not normally retrieved. These files may then be scanned or mined for any data required for the investigation. This method has been used on earlier zSeries servers, for example, to be proactive in checking memory sparing to avoid unplanned outages.

Serviceability tools

zSeries servers are designed to require a minimum of tools for field usage. To achieve the highest possible server availability, execution of installation, upgrade, and repair activities must be quick and flawless. Where tools are absolutely required to support these tasks, they must be lightweight, efficient, readily available, and where possible, multipurpose. The z990 server, with its unique multibook structure, presented some interesting challenges.

zSeries products have a requirement that installation and service be performed by a single SSR. This requires that each FRU weigh less than forty pounds so that a single person can lift it. Since certain FRUs such as the processor book weigh more than forty pounds, tools are necessary to assist the SSR, and this, in turn, requires that the tool, if it must be lifted, also weigh less than forty pounds. zSeries packaging designers have created a set of tools that meets the requirements.

The new z990 processor book contains the memory cards, the MCM, I/O connectivity modules, and cooling

connectivity. In previous servers, the memory cards were accessible from the front of the server. For the z990, the book must be removed from the cage to replace a memory card, and an ingenious service tray has been designed for this purpose. This service tray consists of two service tray arms that mount to the z990 frame and the service tray (platform) itself that mounts to the two service tray arms. A mechanical stop is positioned in one of four slots (one slot for each of the different books) to ensure that the book is not pulled out too far from the frame. Once the book is pulled out and secured, the memory card is accessible for service

If the entire book must be replaced, the service tray is used as a platform to support the book. Since the book weighs fifty-three pounds, a lifting device must be used to ensure that a single SSR can perform the replacement. Again, the packaging designers came up with a clever solution. An existing hoist that mounts on the top of the frame is used to raise and lower any component that exceeds the weight requirement. zSeries requires that any tools needed to replace critical parts be available within the same parts procurement time as the replacement part itself (that is, within two to four hours for critical parts). The existing hoist tool has been modified to fit the extended z990 frame. This hoist tool is adaptable, with multiple locations for locking pins, to three different frame lengths, eliminating the need for multiple tools for multiple frame lengths. The hoist itself weighs less than forty pounds, so that it can be assembled and installed by a single SSR.

The service tray arms used for book and memory repair are also unique. These arms can be inverted, swapped from left to right, and then mounted in a different location on the frame, to create a service platform for the optional IBF batteries. The IBF batteries weigh more than a hundred pounds and are installed at the top of the frame. The service tray arms are mounted just below the batteries and are used as a platform to pull the batteries out of the frame. The hoist, mounted on the top of the frame, is then used to lower the batteries to the floor and raise replacement batteries onto the service tray. The SSR can then reinstall the replacement batteries in the frame.

Licensed Internal Code subsystem

To improve the quality of the Licensed Internal Code (LIC) of the z990 and to shorten the duration of the initial new-hardware bringup in development, an emphasis was placed on increasing the level of LIC verification using enhanced simulation techniques with system-level models. A key objective was to use the system-level simulation models to verify the LIC operation of a number of complex functions so that these functions,

which previously required actual hardware for a full verification, could be tested during the simulation phase, thus reducing the number of servers necessary as test vehicles during hardware bringup.

For example, a complete multibook IML, including LPAR partition activation, was achieved using the central electronic complex simulation (z/CECSIM) microcode simulator as described in [3], enhanced for the z990 server. Reaching this target using the actual LIC from all subsystems was an important milestone in reducing test effort. This simulation scenario also verified the correct handling of numerous interfaces between various LIC subsystems and allowed efficient troubleshooting and repairs of any problems that were found.

Using the z/CECSIM microcode simulator to perform an early verification of complex functions implemented in LIC proved to be vital in achieving the LIC quality objectives. Because increased testing of these functions could be done in a simulation environment, less verification was necessary on the actual hardware, thus reducing the number of physical hardware systems needed to execute the bringup testing efforts and reducing the time required to complete the bringup objectives.

Summary

The RAS design for the z990 server has been enhanced to support the significant new server structure. This design supports the e-business on demand* and autonomic computing initiatives for the z990 server with new levels of self-management and nondisruptive growth and service. As described in this paper, the major RAS improvements for the z990 include concurrent book add, the ability to add and activate processors, memory, and I/O connectivity while the server is running; a fault-tolerant ring to interconnect books; cross-book PU sparing, which enables processors on other books to be available for activation to maximize recovery capability in the event of individual processor failures; sparing capability in the main storage key array, which allows recovery from key chip failures; multiple logical channel subsystems with channel sharing across those subsystems; a high-speed fan backup for the modular refrigeration unit; enhanced code quality through the extensive use of design simulation; enhancements in error data capture and retention; and innovative and efficient new service tools to support the new book structure.

Appendix: Acronyms used in this paper

ABIST Array Built-In Self-Test
AMD Air-Moving Device
CBA Concurrent Book Add
CC Cage Controller

CE Correctable Error

CECSIM Central Electronic Complex Simulation

CHPID Channel Path Identifier
CP Central Processor
CSS Channel Subsystem

CSSID Channel Subsystem Identifier
DCA Distribution Converter Assembly

DCS Dynamic CPU Sparing

DRAM Dynamic Random-Access Memory

ECC Error-Correction Codes

EMIF ESCON Multiple Image Facility
ESCON Enterprise Systems Connection
ESTI Enhanced Self-Timed Interface
ETR External Time Reference
FCP Fibre Channel Protocol

First Error Data Capture

FICON Fiber Connection
FRU Field-Replaceable Unit
FSP Flexible Service Processor
HOM Hardware Object Module

HWA Hardware Address I/O Input/Output

FEDC

IBF Integrated Battery Facility
ICB Integrated Cluster Bus
ICF Internal Coupling Facility
IFL Integrated Facility for Linux
IML Initial Microcode Load

IOCDS Input/Output Configuration Data Set

LBIST Logic Built-In Self-Test
LCSS Logical Channel Subsystem
LED Light-Emitting Diode
LIC Licensed Internal Code

LICCC Licensed Internal Code Configuration Control

LPAR Logical Partition
L1 Level 1 Cache
L2 Level 2 Cache

L3 Level 3 Storage (Main Memory)

MBA Memory Bus Adapter
MCF Microcode Fix
MCM Multi-Chip Module
MCSS Multi-Channel Subsystem

MLCS Multiple Logical Channel Subsystems

MRU Modular Refrigeration Unit OSA Open System Adapter PA Problem Analysis

PAF Processor Availability Facility PCHID Physical Channel Identifier

PR/SM Processor Resource/Systems Manager

PSS Product Support System

PU Processing Unit

RAS Reliability, Availability, and Serviceability RETAIN Remote Technical Assistance Information

Network

RSF	Remote Support Facility
SAP	System Assist Processor
SCC	Storage Controller Control
SCD	Storage Controller Data
SCE	Storage Control Element

SE Service Element

SEC/DED Single-Error Correction/Double-Error

Detection

SMP Symmetrical Multi-Processor SRC System Reference Code SSR System Service Representative

UE Uncorrectable Error VPD Vital Product Data

References

- 1. J. Probst, B. D. Valentine, C. Axnix, and K. Kuehl, "Flexible Configuration and Concurrent Upgrade for the IBM eServer z900," *IBM J. Res. & Dev.* **46**, No. 4/5, 551–558 (July/September 2002).
- L. C. Alves, M. L. Fair, P. J. Meaney, C. L. Chen, W. J. Clarke, G. C. Wellwood, N. E. Weber, I. N. Modi, B. K. Tolan, and F. Freier, "RAS Design for the IBM eServer z900," *IBM J. Res. & Dev.* 46, No. 4/5, 503–521 (July/September 2002).
- 3. J. von Buttlar, H. Böhm, R. Ernst, A. Horsch, A. Kohler, H. Schein, M. Stetter, and K. Theurich, "z/CECSIM: An Efficient and Comprehensive Microcode Simulator for the IBM eServer z900," *IBM J. Res. & Dev.* 46, No. 4/5, 607–615 (July/September 2002).

Received October 15, 2003; accepted for publication February 9, 2004; Internet publication May 13, 2004

Myron L. Fair IBM Systems and Technology Group, 2455 South Road, Poughkeepsie, New York 12601 (mlfair@us.ibm.com). Mr. Fair is a Senior Engineer in the zSeries systems RAS group. He graduated from the University of Illinois in 1967 with a B.S. degree in mathematics, joining IBM that same year in the Systems Development Division, where he held various technical positions related to field and development RAS, and in special contracts technical assessment. In 1980, he transferred to the Group Staff Headquarters and became manager of product RAS analysis in 1984. Mr. Fair is currently the team leader for RAS requirements and objectives for zSeries servers.

Christopher R. Conklin IBM Systems and Technology Group, 2455 South Road, Poughkeepsie, New York 12601 (smokey@us.ibm.com). Mr. Conklin received a B.S. degree in electrical engineering from the Rochester Institute of Technology in 1984. He joined IBM that same year in the Communication Products Division as a VLSI test engineer. He is currently a Senior Engineer working on the development of LIC for the IBM zSeries processors. Mr. Conklin has received four IBM Outstanding Technical Achievement Awards and has also co-authored one patent and one technical paper.

Scott B. Swaney IBM Systems and Technology Group, 2455 South Road, Poughkeepsie, New York 12601 (sswaney@us.ibm.com). Mr. Swaney received a B.S. degree in electrical engineering from the Pennsylvania State University in 1988. He joined IBM that same year in the Enterprise Systems Division as a VLSI design engineer. He is currently a Senior Technical Staff Member working on hardware and system design for IBM eServers. Mr. Swaney specializes in design for high availability in microprocessors; he holds multiple patents related to processor recovery.

Patrick J. Meaney IBM Systems and Technology Group, 2455 South Road, Poughkeepsie, New York 12601 (meaney@us.ibm.com). Mr. Meaney is a Senior Engineer in eServer zSeries custom hardware design. He is the SCE RAS and recovery leader responsible for design for fault tolerance, error avoidance, error correction, recovery, and design for test and debugging. He was also the SCE timing leader for the G4, G5, G6, and z900 servers. He was the SCE RAS and technology leader for the z990 machine; and he is responsible for the definition of SCE RAS and recovery features of future zSeries CMOS servers. He is also a member of the Future Systems Pervasive Team. Since joining IBM in Poughkeepsie, New York, in 1986, he has held design and timing leadership positions on the ES/9021 bipolar-based machines as well as the S/390/zSeries G4, G5, G6, and z900 CMOS systems. Mr. Meaney holds seventeen U.S. patents and has ten patents pending. He has received several awards, including IBM Outstanding Technical Achievement Awards for H5, G4, G6, and z900 and an IBM Outstanding Innovation Award for the G5 design. He holds a B.S. degree in electrical and computer engineering from Clarkson University (1986) and an M.S. degree in computer engineering from Syracuse University (1991).

^{*} Trademark or registered trademark of International Business Machines Corporation.

^{**} Trademark or registered trademark of Linus Torvalds.

William J. Clarke IBM Systems and Technology Group, 2455 South Road, Poughkeepsie, New York 12601 (wjclarke@us.ibm.com). Mr. Clarke is an Advisory Engineer in the zSeries product engineering group. He graduated from Rutgers University in 1982 and joined IBM in Poughkeepsie, New York, that same year. He worked on various chip designs for the system control element for the 3090 processor family. He also held development responsibility for recovery design, error detection, fault isolation, and the service interface. Mr. Clarke worked in chip design on the 9021 family in the Level 2 cache. He worked on server bringup for the 9021 and continued to hold RAS responsibility for the storage subsystem. He worked in engineering systems test in recovery and serviceability, joining product engineering in 1994 on the processors and storage team.

Luiz C. Alves IBM Systems and Technology Group, 2455 South Road, Poughkeepsie, New York 12601 (alves@us.ibm.com). Mr. Alves is a Senior Engineer working in the zSeries systems design group. He graduated from New York University in 1975 with a B.S. degree in electrical engineering and received his M.S. degree in electrical engineering in 1977 from the Polytechnic Institute of New York. He joined IBM that same year, working in the advanced system manufacturing engineering organization, where he held various technical and managerial positions. In 1985 Mr. Alves was named 3090 field quality assurance manager, and in 1987 he became the RAS manager for the 9021 processor families. He is currently responsible for defining the RAS requirements for future products.

Indravadan (Dan) N. Modi IBM Systems and Technology Group, 2455 South Road, Poughkeepsie, New York 12601 (modidan@us.ibm.com). Mr. Modi is an Advisory Engineer in the zSeries systems RAS group. He graduated from Gujarat University, India, in 1966 with a B.S. degree in electrical engineering and received his M.S. degree in electrical engineering in 1968 from Utah State University. He joined IBM in Poughkeepsie, New York, that same year, working in the power system design area, where he held various technical positions. He transferred to IBM East Fishkill in 1982 as a manager of electrical analysis in the MLC area, responsible for substrate electrical characteristics along with ΔI and coupled noise analysis. He returned to Poughkeepsie in 1986 to join the system assurance group, where he was responsible for system technology assurance. In 1990, he joined the RAS group, where he was responsible for technology reliability, SPQL, and development of system test and acceptance specifications. In 1998 he joined the custom microprocessor design group, where he was responsible for chip/MCM reliability, SPQL, and the system test and acceptance specifications for zSeries. In 1999, he returned to the systems RAS group, with his original responsibilities.

Fritz Freier IBM Systems and Technology Group, IBM Deutschland Entwicklung GmbH, Schoenaicherstrasse 220, 71032 Boeblingen, Germany (freier@de.ibm.com). Mr. Freier is an Advisory Engineer working in the support element application development group. He joined IBM in 1973, working as a Customer Engineer. He has held various positions in product assurance test and microcode development and is currently the team leader for system serviceability engineering in Boeblingen.

Wolfgang Fischer IBM Systems and Technology Group, IBM Deutschland Entwicklung GmbH, Schoenaicherstrasse 220, 71032 Boeblingen, Germany (FischerW@de.ibm.com). Mr. Fischer studied electrical engineering at the University of Siegen and received his master's (Dipl. Ing.) degree in 1985. He joined IBM in January 1986 in the hardware development department for S/370 in Boeblingen. He holds a patent on serializing bus grants in a multiprocessor system. Mr. Fischer has held various positions in hardware and microcode development; he is currently in microcode development, with RAS and FEDC responsibilities.

Norman E. Weber IBM Systems and Technology Group, 2455 South Road, Poughkeepsie, New York 12601 (nweber@us.ibm.com). Mr. Weber joined IBM as a Customer Engineer in Chicago in 1967, working on mid-range 360 and 370 processors. In 1980 he transferred to Endicott, New York, to work in service planning supporting 4300 processors and developing service strategies for 9370 systems. He was promoted to his current position as an Advisory Engineer in the zSeries RAS organization in 1990 and moved to Poughkeepsie in 1995.