

Reducing planned outages for book hardware maintenance with concurrent book replacement

C. R. Conklin
C. J. Hollenback
C. Mayer
A. Winter

The IBM System z9™ introduces the enhanced book availability (EBA) feature to reduce the number of planned system outages. Included as part of the EBA feature is the concurrent book replacement (CBR) function, which allows a single book in a multi-book server to be concurrently removed from the system in order for service personnel to perform a repair or to physically upgrade the hardware on the book. This repaired or upgraded book is then concurrently replaced and reintegrated into the server configuration. In this paper, we describe the benefits that the concurrent book replacement function offers a customer during a planned repair or upgrade of the System z9. We also describe a tool, developed to analyze the server in order to determine whether the server is “prepared” for the concurrent book replacement operation, and we provide an overview of the actions performed by this tool. The paper also contains a description of the concurrent book replacement operation, with an emphasis on the unique functions developed as part of this design.

Introduction

High-availability computing system solutions are desirable throughout the computing industry [1, 2]. Continuous availability is also a very important characteristic that IBM System z* customers seek in a mainframe [3]. Customers expect their servers to be operating nearly 24 hours a day, every day of the year. A scheduled outage, whether for the purposes of hardware repair, hardware upgrade, or software upgrade, costs customers time and money. To increase availability, the IBM System z9* provides extensive “concurrent” service and growth capability. This means that a service action, configuration change, or hardware/microcode enhancement can take place while the system continues to process the customer’s workload.

The concurrent book add (CBA) function was introduced in the previous eServer* zSeries* model z990 [4]. A processor book contains multiple processor chips, physical memory cards, and multiple I/O hub cards. Both the eServer z990 and the System z9 are available in models with one to four processor books. The CBA

function has enabled customers to concurrently upgrade the eServer z990 by adding new book hardware including processors, physical memory, and I/O connectivity. This availability enhancement allowed customers to perform significant hardware upgrades without requiring a costly scheduled outage to complete this action.

The System z9 server improves upon this availability concept with the introduction of the concurrent book replacement (CBR) function. This function allows a customer with a System z9 equipped with two or more books to concurrently repair or upgrade the processor book hardware. Before the advent of the System z9, this kind of repair or upgrade would have required a costly, disruptive system outage. The new CBR function provides the customer with the option of performing this repair or upgrade either concurrently with the customer’s running workload, or disruptively, as was done in previous System z models.

Prior to the introduction of the CBR function, the following scenarios would have required a disruptive customer outage. As noted, with CBR these upgrades and

©Copyright 2007 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.

0018-8646/07/\$5.00 © 2007 IBM

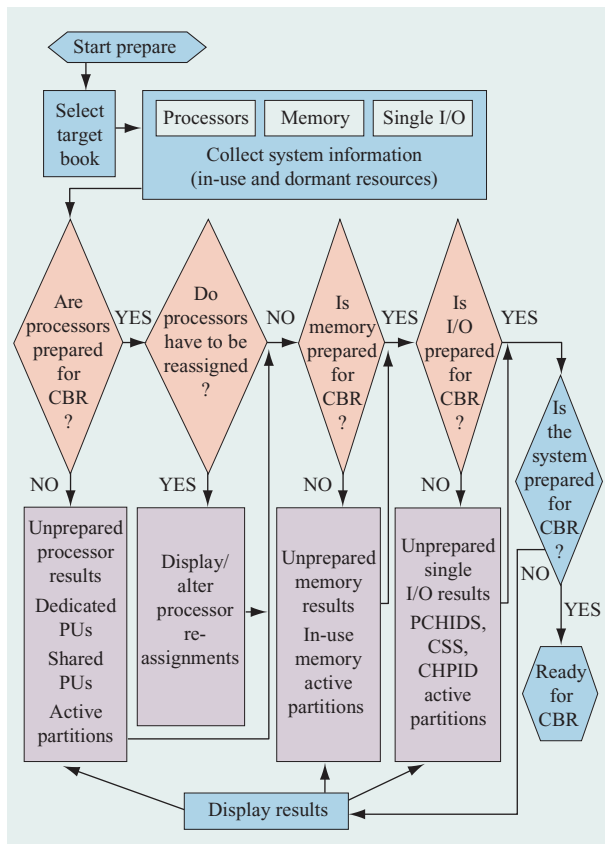


Figure 1

Flow chart for the Prepare for CBR function. (PU: processing unit; PCHIDS: physical channel identifiers; CSS: channel subsystem; CHPID: channel path identifier.)

repair procedures can be performed concurrently without interfering with customer operations.

- *Concurrent physical memory upgrade*—allows one or more physical memory cards on a single book to be added, or an existing card to be upgraded, increasing the amount of physical memory in the system.
- *Concurrent physical memory replacement*—allows one or more defective memory cards on a single book to be replaced concurrently with the operation of the system.
- *Concurrent defective book replacement*—allows the concurrent repair of a defective book when that book is operating in a degraded manner because of errors such as those caused by multiple defective processors.
- *Concurrent evaporator replacement*—allows the concurrent repair of a defective multichip module (MCM) evaporator, restoring proper cooling to the MCM.

- *Concurrent I/O fan-out cage replacement*—Each book of a System z9 contains an I/O fan-out cage that holds from one to eight I/O hub cards. A defective I/O fan-out cage may prevent the operation of one or more of the plugged I/O hub cards. With CBR, the complete I/O fan-out cage can be repaired in a single concurrent operation, restoring I/O connectivity.

To utilize the CBR function for the purpose of concurrently repairing or upgrading hardware residing on the processor book, the server must be “prepared” so that sufficient dormant resources are available on the books that will remain in the system to accommodate resources that are in use on the targeted book. If sufficient dormant resources are not available, the customer must reduce the workload running on the server to meet the requirements before the CBR operation can be initiated. The CBR function provides a powerful tool that analyzes the System z9 and determines whether the system is prepared for the concurrent removal of a specified book. If the server is not prepared, the tool informs the user about the required actions that must be accomplished prior to initiating the CBR operation. Such actions might include deconfiguring (i.e., disabling) of single-path I/O connections, or requiring a reduction of workload to free memory or processor resources.

Once the System z9 is “prepared,” the concurrent book replacement operation can begin. During this phase, the use of resources (such as processor, memory, and I/O) is moved from the book targeted to be replaced to dormant resources physically resident on the remaining book or books. The targeted book is “fenced” (i.e., isolated) from the rest of the system, and powered off. At this point, the book can be physically removed from the system, upgraded, repaired, or replaced and physically reinstalled. As a final step, the book is powered on, initialized, and concurrently integrated back into the system configuration.

Organization of the paper

The remainder of this paper describes the different steps, procedures, and components of the CBR function and how they interact to offer the enhanced book availability feature of the System z9. Two major phases of the CBR function exist. The first and second phases are respectively referred to as the *Prepare for concurrent book replacement* and *Perform concurrent book replacement* phases. The Prepare for CBR phase analyzes the System z9 and determines whether sufficient dormant (unutilized) resources are available on the remaining books that can replace the resources in use on the book targeted for CBR. Once the system verifies that a CBR operation can be performed on a designated book in the system, the Perform CBR phase can be executed.

The Perform CBR section of this paper begins with a description of the design that allows the system to either move the workload off these resources to previously dormant resources or add workload to already active resources. As mentioned in the Introduction, before the targeted book can be powered off and physically removed from the system, it is necessary to isolate and deactivate all of its currently unused resources. This process is described in the subsection of this paper on the book fencing operation. The remaining portions of the Perform CBR section describe the steps that are necessary to physically replace and activate the hardware, and we describe the design needed to concurrently redistribute the previously evacuated resources across all of the books in order to restore the availability and performance characteristics of the system.

The measurements that we took in order to simulate and test these complex functions are explained in the section on verification techniques for CBR. Finally, our paper concludes with a short description of the continued capacity with fenced book function and the cold concurrent book repair function that complement the CBR function and together provide the enhanced book availability feature.

Prepare for CBR

Invoking the Prepare for CBR procedure, which targets a single book of a multi-book System z9, is the first phase of the CBR function. This is a prerequisite to performing the actual CBR operation. **Figure 1** illustrates the high-level decision flow for the Prepare for CBR procedure.

The physical book resources and the total used resources of the system serve as important information for the Prepare for CBR calculations. The system can be viewed as having two physical entities. The first is the targeted book that is to be serviced. The second entity is the set of the remaining books in the system configuration. As previously stated, the Prepare for CBR procedure analyzes the system in order to determine whether the dormant resources on the remaining books of a system are capable of handling the system's current in-use processors, memory, and I/O from a targeted book when its resources are evacuated.

The Prepare for CBR procedure can be invoked from the hardware management console (HMC) through the customer interface or directly from the support element (SE). The SE is a separate laptop computer supplied with each System z9 that executes certain support functions for the System z9 and is used by service personnel to perform maintenance operations on the system. This Prepare for CBR procedure must be invoked prior to actually performing a scheduled CBR on a targeted book for hardware updates or repairs. The Prepare for CBR procedure is invoked by selecting the *Prepare for*

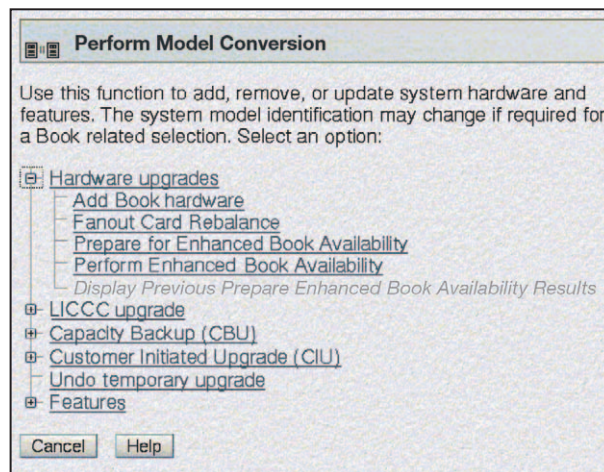


Figure 2

Support element Perform model conversion panel. (LICCC: Licensed Internal Code configuration control.)

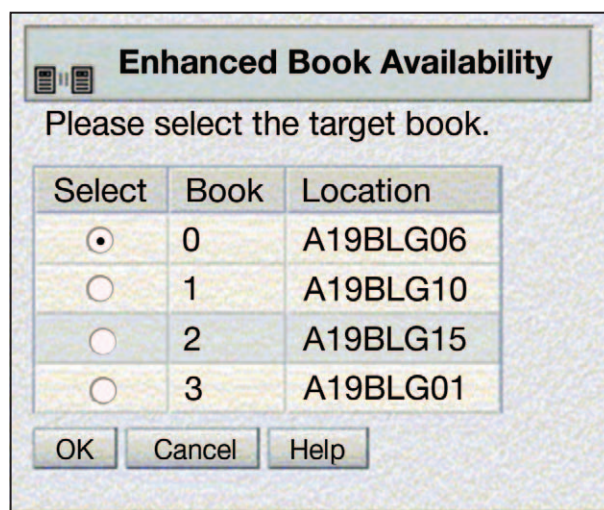


Figure 3

Target book selection panel.

enhanced book availability option listed in the perform model conversion panel of the SE (**Figure 2**). Once this option is selected, the user interface then allows for the book of interest to be selected (**Figure 3**).

Although it is not required, we recommend that the Prepare for CBR procedure be completed under the guidance of a system programmer staff person. If the server resources must be taken offline or reassigned, it is the responsibility of the system programmer to direct any configuration modifications prior to continuing with the CBR procedure.

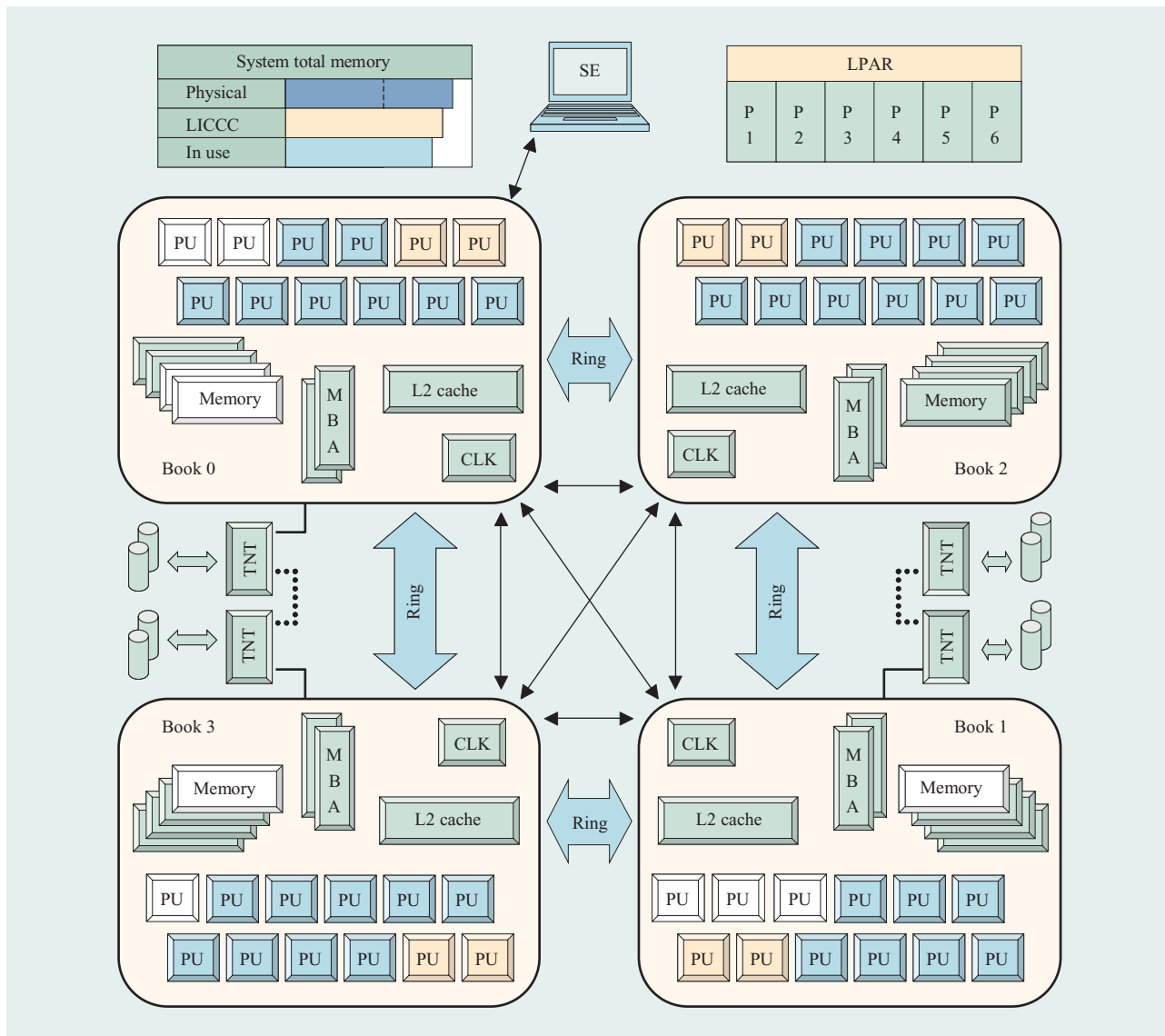


Figure 4

Sample four-book system configuration. (PU: processing unit; LICCC: Licensed Internal Code configuration control; LPAR: logical partition; MBA: memory bus adapter; TNT: Triton-T, a chip that is part of a new redundant I/O interconnect feature.) Colors for memory: white (dormant resources), green (in-use resources). Colors for processors: white (dormant resources), blue (in-use GPPs), yellow (system assist processors). The uppermost bar in the bar graph at the upper left indicates the total amount of physical memory in the system, and the dashed line indicates the amount of physical memory in the three books that will remain in the system. The LICCC graph shows the amount of memory purchased by the customer, which never changes throughout CBR. The in-use bar shows the amount of memory actually being used in the system. “Ring” indicates the ring interface, which is the interface between the L2 cache hardware elements in each book.

In addition to the SE panel option for preparing the system for CBR, the Prepare for CBR procedure is automatically invoked at the start of the Perform CBR operation to ensure that the server is still prepared for the concurrent removal of the specific targeted book. The Perform CBR operation can be invoked from the perform model conversion panel by selecting the *Perform enhanced book availability* option (Figure 2) for

upgrading memory hardware on a book or through the serviceability maintenance package for repairing book hardware.

As mentioned, depending on the model, the System z9 can contain one to four books, each populated with processors, memory, and input/output (I/O) hub cards. **Figure 4** illustrates a sample four-book system configuration. The Prepare for CBR procedure assesses

the physical and logical aspects of all of the system resources in preparation for a possible CBR action.

The Perform CBR operation is prohibited until all of the pertinent conditions identified from the Prepare for CBR procedure are satisfied. All conditions preventing the server from being prepared are presented to the user with instructions that describe how to continue. The results of the Prepare for CBR procedure that reflect a server-unprepared state remain available on the SE and can be redisplayed as needed until another Prepare for CBR procedure is executed. The results panel (Figure 5) provides detailed information for processors, memory, and the various types of single-path I/O conditions blocking the CBR prepared state.

The sample panel shown in Figure 5 illustrates a CBR unprepared state. In this example, the analyses of processors, memory, single I/O, and single alternate Path I/O conditions all failed to meet the criteria required for performing CBR on the targeted book (Book 0). The selected processor information displayed is described in detail in the section on preparing processors later in this paper.

Prepare for CBR results

Three states can result from the Prepare for CBR procedure:

1. The system is ready to perform the CBR operation for the targeted book. Sufficient dormant resources are available on the remaining books to replace resources that are in use on the targeted book. This is referred to as a GO state.
2. The system is not ready to perform the CBR operation because of unsatisfied conditions identified during the Prepare for CBR procedure. Whenever a CBR not-prepared state exists, the customer is provided with detailed information to help determine how to reduce system resources. The customer may need to deconfigure logical processors, release the use of storage within a partition, deconfigure channel paths, or deactivate partitions in order to successfully complete the Prepare for CBR phase. This is referred to as a NO GO state. When we use the term *deconfigure* with respect to logical processors, we refer to the fact that the work running on certain logical processors must be moved to different logical processors, and the association between the logical processor and certain shared physical processors is removed.
3. The system is ready to perform the CBR operation for the targeted book. However, processors have to be reassigned from the original configuration to meet the criteria required to continue the CBR. This is referred to as a GO state.

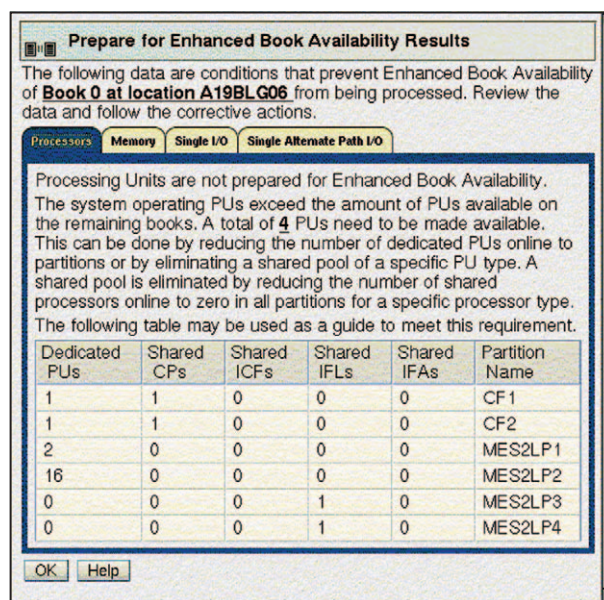


Figure 5

CBR unprepared state. (PU: processing unit; CP: central processor; ICF: integrated coupling facility; IFL: Integrated Facility for Linux; IFA: integrated facility for applications.)

Processors

To understand the concepts involved with preparing the processors for CBR, some basic knowledge of the physical and logical representations of System z9 processors is necessary.

The IBM System z9 offers six types of processors: the general-purpose central processor (CP), system assist processor (SAP), internal coupling facility processor (ICF), Integrated Facility for Linux** processor (IFL), application assist processor (IFA/zAAP), and integrated instruction processor (zIIP). Collectively, the five processor types (excluding the SAP processor type) are referred to as general-purpose processors (GPPs). (Note that “IFA” stands for “ingrated facility for applications,” but it is also common to refer to IFA as the zAAP, which stands for “eServer zSeries application assist processor.”)

Figure 6 shows 24 physical processors, of which 18 (in the first three columns) are actually purchased by the customer (eight CPs, four SAPs, one ICF, three IFLs, two zIIPs). The numbers and types of processors purchased by the customer are controlled by processor LICCC (LIC customization code) and customized to the customer’s individual needs. The customer defines how the various processor types are used to maximize performance and operations either by dedicating them to a single logical partition or by sharing them across multiple partitions. In this illustration there are a total of four dedicated processors (two CPs, one ICF, one IFL)

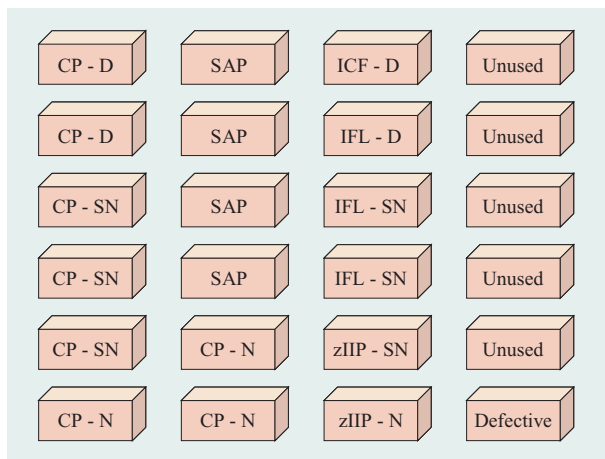


Figure 6

Sample processor distribution. (D: dedicated; S: shared; N: non-dedicated; CP: central processor; SAP: system assist processor; ICF: internal coupling facility processor; IFL: Integrated Facility for Linux processor; zIIP: integrated instruction processor.)

and six shared processors (three CPs, two IFLs, one zIIP). Also shown is one defective processor. Defective processors may or may not be part of the prepare processor calculations, depending upon the number that are defective and the number of dormant resources that are available. The number of nondedicated processors is represented by the total number of processors defined by the LICCC less the number of dedicated processors. Therefore, nondedicated processors can include shared processors and undefined processors. There are 12 nondedicated processors (six CPs, two ICFs, two IFLs, two zIIPs) in this example.

Preparing processors

Processor information is collected according to the processor entities, as described above. The number of physical processors that are available on the remaining books determines whether the current processor definitions and usage meet the criteria for a successful preparation for CBR.

Physical and LICCC processor information is collected from the system's vital product data (VPD). The LICCC processor information includes both the permanent configuration (which defines the processors that the customer purchased and has available at any time) and any additional processors that may be active because of a temporary processor upgrade. This could be a result of a capacity backup feature (CBU) or an on/off capacity on demand (OOCOD) being active. (CBU and OOCOD are two different types of temporary processor upgrade.)

Each of these features can bring in a new temporary LICCC processor configuration.

Once the physical processor information has been determined, the logical processor information is collected. This logical processor information provides details for each online processor for every active partition in the system. It includes the processor LICCC type and information as to whether it is a dedicated or shared processor.

The number of nondedicated processors may be reduced temporarily while the Perform CBR operation is being executed to meet the needs of the available processor resources. The minimum number of nondedicated processors is defined by the shared pool count. (A shared pool includes physical processors of the same type that are assigned to a given type of logical processor.) At least one nondedicated processor of each type must remain active during the Perform CBR operation if any shared processors of that type are currently in use. The minimum number of GPPs is the number of dedicated processors plus the minimum number of nondedicated processors. In this example there is a minimum of seven GPPs (four dedicated processors plus three in the shared pool count).

The GPP-to-SAP ratio and current SAP configuration of the current system are determined, and an attempt is made to preserve them during the actual CBR operation when the targeted book is removed from the server configuration. The GPP-to-SAP ratio is determined by dividing the sum of the total number of dedicated processors plus the total number of nondedicated processors by the current number of SAPs. In this example, the GPP-to-SAP ratio is equal to 4 [(four dedicated processors plus 12 nondedicated processors) divided by four SAPs].

The minimum number of SAPs is also calculated using the GPP-to-SAP ratio. In this example, the minimum number of SAPs is 2. This is derived by calculating an initial value of 1 [(seven minimum GPPs)/four GPP-to-SAP ratio], plus one due to a remainder in the calculation.

The number of dedicated processors and shared pool types and the minimum number of SAPs affect the GO/NOGO results of the Prepare for CBR readiness test. Whenever the exact current system configuration cannot be maintained within the targeted system in order to achieve a GO status, the shared processor types and SAP quantities are adjusted. The Prepare-for-CBR procedure displays an initial selection that indicates how the number of nondedicated processors can be temporarily reduced. Users can accept them as shown or make modifications to best suit their needs. The number of SAPs in the targeted system is calculated by the prepare tool, which maintains the initial GPP-to-SAP ratio. This is described in more detail in the section on reassigning processors.

For the cases in which the server's processors are not ready to perform CBR, the Prepare for CBR tool collects and displays all of the appropriate current workload information associated with partitions and processors. An example of this display is shown in Figure 5. This panel displays the corrective actions required to adjust the processor configuration conditions that are preventing the Perform CBR operation for the targeted book. Logical partitions may have to be deactivated or processors deconfigured in order to meet requirements as indicated by the panel information. In this example, the panel notifies the user that the number of in-use processors must be reduced by four. This user instruction was determined by the prepare processor algorithm and is required in order for the processors to be ready for the concurrent removal of the targeted book.

In general, removing any dedicated processor or all of one type of shared processors reduces the total in-use processor count by one. However, manipulating this processor configuration may also change the target system GPP-to-SAP ratio or minimum number of SAPs on the next execution of the Prepare for CBR procedure.

Reassigning processors

The processor reassignment panel is displayed only when the criteria required for the processor prepare step can be met. This panel allows the customer to specify reassignments to current nondedicated processors that take effect during the actual Perform CBR action.

Figure 7 is a sample nondedicated processor, reassignment panel. It is used to change or accept the system processor assignments that are generated during the processing of the Prepare for CBR procedure. These processor values set by the system programmer are preserved and utilized for the running system during the Perform CBR operation.

Preparing memory

Two factors are used to determine whether the system memory is prepared for CBR. The first requires a calculation to determine the amount of physical memory contained on the remaining books within the configuration. The second is the current in-use memory for the running system. This in-use memory includes the hardware system area (HSA) memory as well as the memory used from each active logical partition.

In order for the memory to be prepared for CBR, the in-use memory must not exceed what is physically available when the targeted book is removed. If the criteria cannot be met, all of the pertinent memory information is collected and provided to the system programmer for evaluation. The memory information is collected on a logical partition basis, which includes the identity of the partition and its associated memory

Processor Type	Dedicated Count	Non-Dedicated Count	Processor Totals	LICCC Count
CPU	1	14	15	18
ICF	2	0	2	4
IFL	0	1	1	2
IFA	1	0	1	2
SAP	5	0	5	6
Available to use	0	0	0	0
Remaining Book Totals	9	15	24	0

Figure 7

Nondedicated processor reassignment panel.

consumption. This information is sorted from highest to lowest memory consumption when it is displayed to the user. On the basis of this information, the system programmer can decide what actions to take in order to meet the memory requirements. Memory may be freed by releasing storage within a partition, by deactivating partitions, or both.

The panel in **Figure 8** illustrates the corrective actions required to address the memory configuration conditions that prevent the Perform CBR operation from being executed for the targeted book. Logical partitions must be deactivated in order to satisfy requirements as indicated by the panel information. The in-use memory must be less than or equal to the available memory on the remaining books within the server.

Preparing I/O

All pertinent I/O information for the targeted book is collected and evaluated during the Prepare for CBR phase to ensure that I/O connectivity is maintained during the Perform CBR operation. The I/O information gathered is also used during the perform step. The state and status of every physical channel path identifier (PCHID) associated with the targeted book is collected, evaluated, and processed accordingly. The PCHID is used to map the channel subsystem identifier to a physical location in the I/O cage.

The information for those PCHIDs that are defined in the configuration but are not currently online is saved and then later used during the Perform CBR operation to ensure that the channel paths are placed in the correct service state at that time. Such PCHIDs are not included in the single-path I/O checks during the prepare I/O step.

Ideally, every I/O connection from a System z9 book should have an associated alternate I/O connection from a different book within the server configuration. During the prepare I/O step, every online PCHID associated with

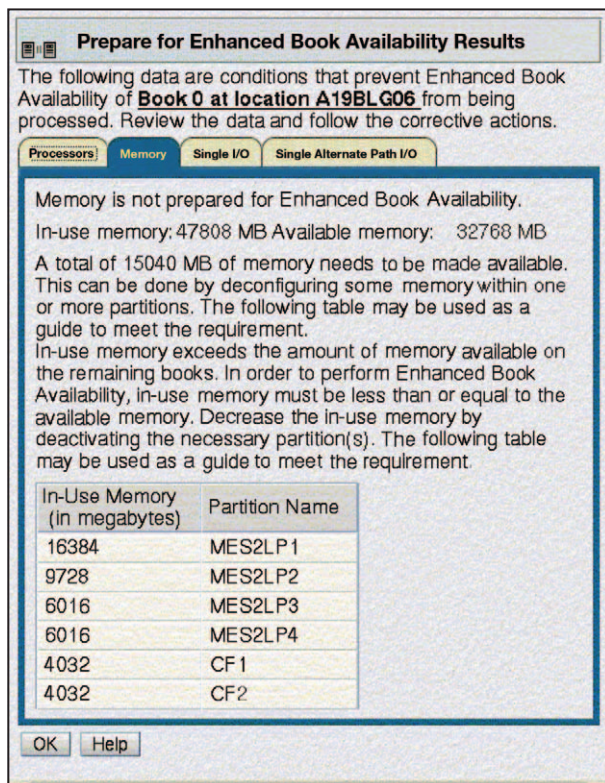


Figure 8

SE panel that highlights the memory-prepare requirements.

the targeted book is checked for any possible conditions leading to *single I/O connectivity*, a phrase that is explained shortly. The prepare tool checks several conditions to determine whether a single I/O connection exists. These conditions include

- I/O connections that do not have associated alternate paths.
- I/O connections that have alternate paths; however, the paths are determined to be faulty.
- I/O connections with alternate paths that are connected to the same book.

In addition to these checks, other tests ensure that any alternate paths associated with the target book are not active. The single I/O path connections must be determined during the Prepare for CBR phase so that the system programmer can deconfigure such channel paths prior to starting the Perform CBR operation.

The prepare tool collects additional information for the PCHIDs that are online, such as all of the associated channel subsystems (CSSs), channel path identifiers (CHPIDs), and associated partition information. If any

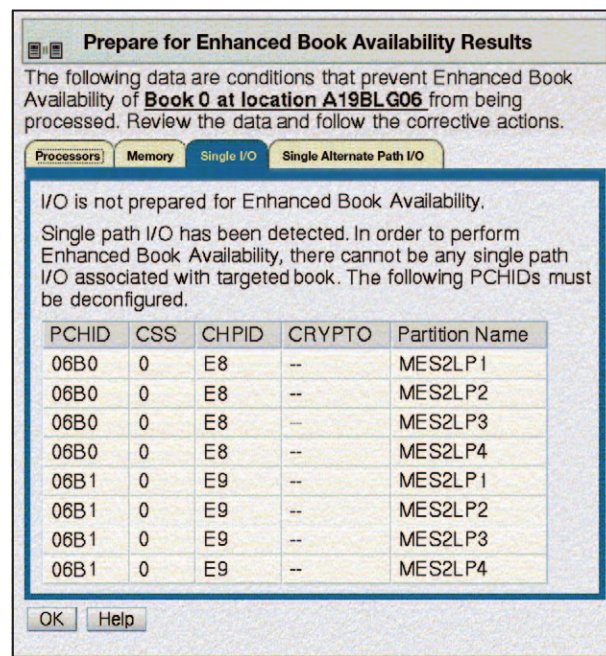


Figure 9

SE panel highlighting the single-path I/O requirements.

single-I/O connectivity condition is detected during the Prepare for CBR procedure, the results are displayed.

Figure 9 is a sample panel that is displayed when single-I/O connectivity conditions are detected. Although different graphical tabs are generated for the various types of single-I/O connectivity failure conditions (single I/O, alternate I/O, domain I/O), the panel information is similar for the different failure conditions. Corrective actions require the PCHID(s) to be configured offline; otherwise, all of the associated partitions must be deactivated.

Perform CBR

The Perform CBR operation is executed in order to carry out the actual removal and replacement of the physical book hardware and is the second phase of the CBR function. **Figure 10** illustrates the high-level process flow that occurs during the operation.

Perform CBR operation initiation

The Perform CBR operation is initiated through a panel interface on the support element (SE) where the book targeted for repair or upgrade is specified (see Figures 2 and 3). The SE code verifies that the server is ready to perform the CBR action by calling the Prepare for CBR procedure. This operation, described in the previous section, also determines the number of processor resources to which the system is reduced while operating

with the targeted book removed from the server. Once the verification test completes, the reduced system processor results are passed to the logical partition hypervisor (LPAR), which begins the resource evacuation phase of the Perform CBR operation.

Figure 11 illustrates an example of a four-book server that is ready to perform a CBR operation, targeting Book 2 for replacement. In this example, the system programmer had to deactivate two LPAR partitions (P5 and P6) during the Prepare for CBR phase in order to reduce the memory requirements of the system to meet the CBR requirements. Sufficient dormant processor resources are physically available on Books 0, 1, and 3 to satisfy the processor requirements. All I/O with connectivity to Book 2 has an available alternate path to Book 1 through the linked Triton-Ts (TNTs) [5], chips that are part of a new redundant I/O interconnect feature.

Resource evacuation

The LPAR hypervisor directs the resource evacuation step of the Perform CBR operation. This step begins when the SE sends a request to the LPAR hypervisor to begin the resource evacuation, and the SE passes to the LPAR code the book number of the targeted book and the number of processors of each type to which the server must be reduced. These processor counts were determined during the Prepare for CBR procedure on the basis of the physical configuration, the currently running workload, and the choices made by the system programmer in order to reassign nondedicated processors.

Upon receiving the request to begin the resource evacuation, the LPAR hypervisor code verifies that the requested parameters (the book number and the processor counts by type for the reduced system) represent a valid request based on the server configuration. Once the validity checks are satisfied, the LPAR hypervisor stops any further physical memory allocation requests targeted for the book that is to be removed, in preparation for initiating the start of the concurrent physical memory evacuation procedure. In addition, the LPAR hypervisor responds to the SE, which indicates that the resource evacuation request has been accepted and that this process has started in the server.

Concurrent physical memory evacuation

The first step in the resource evacuation procedure is the concurrent movement of physical memory increments that are in use on the node being targeted for removal to available physical memory increments (each increment 64 MB in size) on one or more of the books that will remain in the system. This movement of physical memory increments is performed concurrently with the operation of the system without involvement of the operating system or application software.

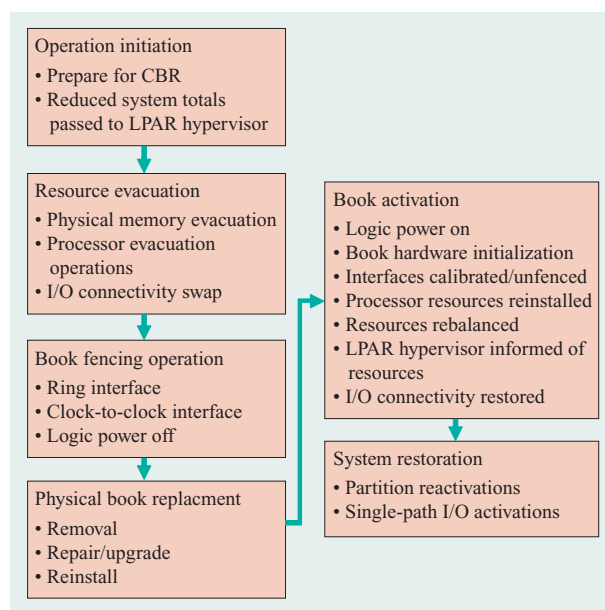


Figure 10

Flow chart for Perform CBR. (LPAR: logical partition). The ring interface connects the L2 cache chips on each book.

The concurrent physical memory evacuation uses the new dynamic memory move function to perform the actual movement of storage from one physical memory increment to another. This function utilizes unique firmware and specific hardware to concurrently change the physical memory backing of an absolute storage increment. A system has an absolute storage space that may be larger than the physical storage space. Any absolute storage increment that is in use must be assigned to, or backed by, a unique physical storage increment.

As previously stated, the storage increment size for the System z9 server is 64 MB. During the concurrent physical memory evacuation step, the physical memory must be moved with a storage-increment granularity. For this operation to be performed concurrently, the server must be paused, and no memory activity can occur during the time during which the physical memory storage increment is moved. Pausing the system for the time it would take to move a full 64 MB of memory would be too time-consuming and would have noticeable effects on the operating system. To overcome this, the hardware and firmware combination developed for the dynamic memory move function breaks up a 64-MB storage increment into 1-MB sub-increments during the time period in which the dynamic memory move function is operating. This 1-MB sub-increment can be moved from one physical memory location to another without affecting the server. Specific

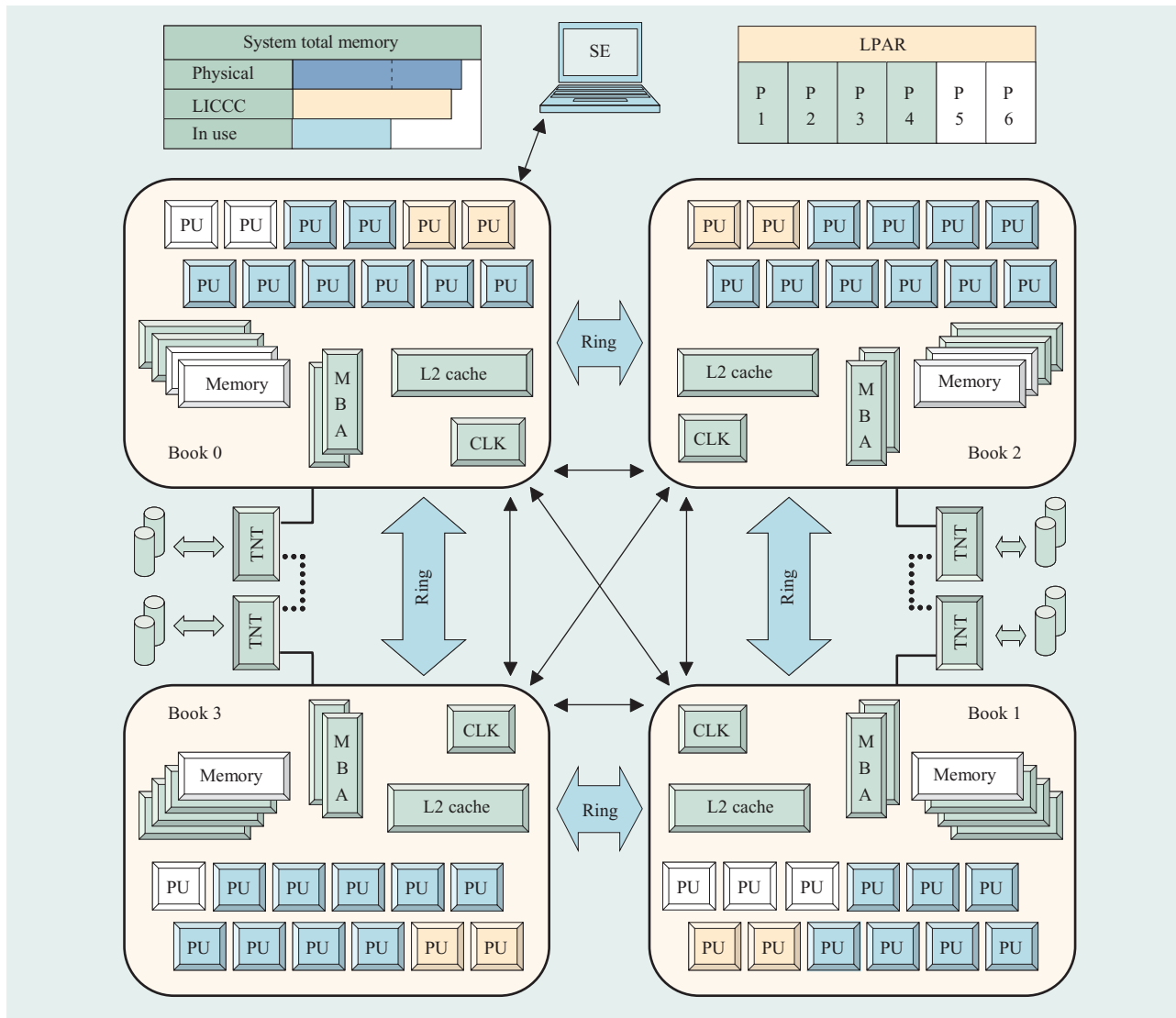


Figure 11

CBR example illustrating Prepare for CBR. Note that sufficient dormant memory (white) exists on the books remaining in the system (0, 1, 3) to take the place of the memory that is in use and physically located on Book 2 (green). Sufficient dormant processors are available on the remaining books (white) to take the place of processors physically located on targeted Book 2. The system total memory graphs at the upper left indicate that the system is prepared, in terms of memory, for the CBR operation. The total amount of in-use memory is equal to the amount of physical memory available in the remaining books of the server (dashed line).

delays are intentionally introduced between the 1-MB moves that are long enough to allow the server to run to prevent I/O timeout issues, but short enough to allow all of the physical memory on a fully populated book (128 GB) to be completely moved within 20 minutes.

Processor evacuation operations

After the in-use physical memory increments on the targeted book have been concurrently moved off that

book, the next step of the resource evacuation procedure is to stop using processors that are physically located on the targeted book. The following actions take place during the processor evacuation step of the resource evacuation procedure.

- On the basis of the reduced system processor resource requirements that were determined during the Prepare for CBR phase, certain nondedicated processors may

be downgraded to spare (dormant) processors to free physical processor resources.

- The workload that runs on any processors that are physically allocated on the book targeted for replacement is concurrently moved to a dormant processor that is physically located on another book.
- SAP processors that are physically allocated on the targeted book may be downgraded to spare processors if it is determined that they are unnecessary in the reduced system, or they may be concurrently reassigned to dormant processor resources on another book if they are required to remain operating in order to maintain a constant GPP/SAP ratio.

The LPAR hypervisor begins the processor evacuation step by deconfiguring shared processors in preparation for them to be downgraded to spare processors. The number and type of shared processors that must be deconfigured is determined by the reduced system processor counts that were passed to the LPAR hypervisor at the beginning of the resource evacuation procedure. Once the LPAR hypervisor deconfigures the required number of shared processors, control is passed to the i390/millicode firmware for the remaining actions in order to complete the processor evacuation step.

Using the reduced system processor counts, the i390/millicode firmware begins its role in the processor resource evacuation step by converting the required number of SAP and/or shared processors to spare processors. If a SAP was chosen to be downgraded to a spare processor, any functional or error affinities to I/O hub devices are reassigned to other SAPs in the server prior to the conversion. To help understand this concept, note that the I/O hub devices communicate with SAP processors and that each I/O hub device in the system is assigned to a certain SAP for handling normal operations (functional affinity). Each I/O hub device is also assigned to a SAP to handle error-type operations (error affinity). If a SAP processor is to be removed from the system, these assignments must be made to another SAP that is to remain in the system.

Any logical processors that are to remain in the server and that are physically located on the targeted book must be relocated to a physical processor on another book. This is accomplished using the new concurrent physical processor reassignment operation, which changes the physical assignment of one or more logical processors in the system. The state of the source operating processor is captured and copied into the target physical processor. The operation utilizes the z/Series transparent sparing hardware and is performed transparently with respect to the operating system or application program.

Once the processor evacuation step has been completed, the SE code is informed of the completion of the memory and processor evacuation steps so that the remaining steps of the resource evacuation procedure can be initiated.

I/O alternate path swap operation

After the completion of the memory and processor evacuation steps, the I/O information collected during the Prepare for CBR phase for the targeted book is used to exploit the System z9 redundant I/O interconnect (RII) feature [5]. At this stage, all single-path I/O [i.e., Integrated Cluster Bus (ICB) channels attached to the targeted book] were deconfigured during the Prepare for CBR phase. The residual I/O attached to the targeted book has an associated, functional alternate I/O connection. Therefore, this I/O can remain operational, without stopping or interrupting the traffic to and from the I/O units, throughout the entire Perform CBR operation. For each redundant I/O connection, the SE requests the CEC firmware to perform a controlled swap to the an alternate path. These swaps are completely transparent to active I/O operations. Once completed, all of the I/O attached to the targeted book is accessed from a book that remains in the server via the alternate path. In the unexpected event of failing to swap to the alternate path, the affected I/O domain must be deconfigured, as in the other single-path I/O discussed previously, in order to continue with the Perform CBR operation.

Book-fencing operation

After the memory, processor, and I/O resource evacuation has completed, the targeted book is ready to be fenced (logically disconnected) from the server and finally primed to be physically removed. The following actions take place during the book-fencing operation:

- The SE code de-registers all necessary resources on the targeted book from the clock stop error handler. This action is performed so that when the resources are intentionally fenced, this will not be viewed by the clock stop error handler as a unit check-stop caused by defective hardware. A unit check-stop occurs when a certain piece of hardware immediately stops running. In most cases, this occurs when hardware, such as a processor or I/O hub, detects an error and stops. When a unit check-stop occurs in a running system, this is interpreted as hardware failure. However, in the case of a controlled shutdown, this should not be interpreted as a failure.
- The SE requests the i390/millicode firmware to fence the memory bus adapter (MBA) fan-out cards on the targeted book. The MBAs are check-stopped and made unavailable for further use by the firmware.

- The i390 firmware, after receiving the request from the SE and verifying that it is safe to fence the book, invokes the steps required to fence the targeted book from the server. All of the hardware resources within the book, and the book itself, are logically removed from the server configuration.
- After all of the resources on the book are fenced, the SE code initiates the fencing of the clock-to-clock interfaces to the clock chip on the targeted book and disables all interrupts on the book.
- Next, the SE deactivates the modular refrigerator unit (MRU) temperature sense cable so that the power firmware does not incorrectly detect and report cooling errors while the book is physically removed from the system.
- Finally, the SE requests the firmware on the associated flexible support processor (FSP) to turn off the logic power for the targeted book and all contained field-replaceable units (FRUs). The FSPs associated with the targeted book remain operating.

Figure 12 illustrates the state of the server after the resource evacuation and book-fencing operations have completed. The physical memory increments that were being used on Book 2 have been concurrently moved to memory increments that were available on one or more of the other books. All physical processors on the targeted book have first been converted to spare processors and then removed from the server configuration. The logical processors that were physically allocated on Book 2 prior to the start of the resource evacuation operation have been either relocated to dormant physical processors on one of the other books or downgraded to spare processors. All I/O that was attached to the targeted book (Book 2) is now accessed through a book that will be remaining in the server (Book 1).

Physical book replacement

At this point in the book-replacement process, the physical hardware associated with the targeted book has been fenced from the rest of the server, and logic power has been turned off in preparation for physically removing the book hardware from the server to perform the required repair or upgrade.

Next, all cabling (such as self-timed interface cables and power/thermal sensor cables) is physically removed from the targeted book. The I/O fan-out cage containing the I/O hub cards is removed from the book, and the physical book is removed from the server.

If the goal is to provide a physical memory upgrade, the new memory cards are added to the original book, or the new memory cards replace one or more of the existing memory cards. If this is a book repair operation, the

memory cards from the original book are removed and reinstalled in the replacement book.

After the required updates have been made to the original book, or the replacement book has been populated with the non-defective hardware from the original book, the book is physically replaced in the server, the I/O fan-out cage is reinstalled, and all cabling is reinstalled in its original location. After this is completed, the customer engineer (CE) continues with the activation of the repaired or upgraded book.

Book activation (concurrent book add)

Once the book hardware has been physically reinstalled in the server and recabled, the book activation sequence can be initiated. This sequence is essentially the same as the concurrent book add (CBA) operation introduced in the prior System z, with a few enhancements added for the System z9 [4]. The process for the book activation sequence is described as follows:

- Power is applied to the book. The FSPs for this book are still operating and do not have to be rebooted.
- Hardware initialization is performed on the newly reinstalled book, and the hardware verification tests are run. This verification is performed while the book is still fenced from the rest of the system so that a possible failure at this point does not disturb the running system.
- Once the hardware is verified, the book is reintegrated into the server. The clock-to-clock interfaces and the ring interfaces to the newly reinstalled book are calibrated and unfenced.
- The book LICCC record is reapplied in order to bring back any processor resources that were removed during the resource evacuation step. If our example had been a book-repair scenario and a new book was used to repair a defective book, a new LICCC record would be obtained prior to being applied.
- The processor resources are rebalanced to match the original processor allocations prior to the start of the resource evacuation procedure.
- The LPAR hypervisor is informed of the newly reinstalled processor and memory resources.
- If the I/O connectivity was swapped to an alternate path during the resource evacuation procedure, the primary path I/O connectivity is restored to the book.

System restoration

A notification panel is displayed on the SE at the completion of the book activation sequence. At this point, the CBR operation is finished. The customer can

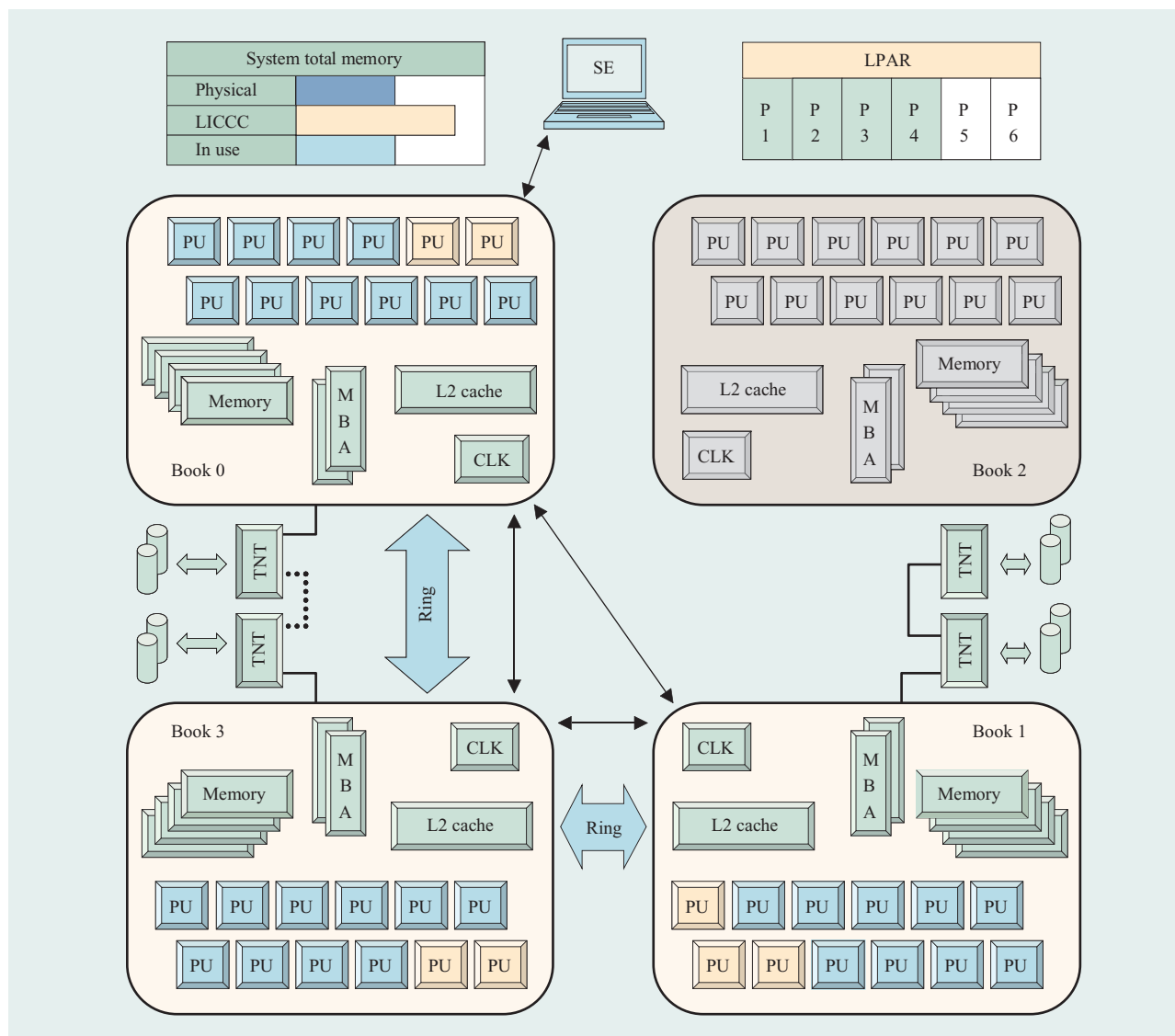


Figure 12

CBR example (Book 2 is fenced). All active resources on the targeted book have been evacuated to dormant resources on the remaining books. The physical memory on the targeted book is no longer available, as indicated in the total system memory bar charts.

now reactivate any partitions that were deactivated in order to free processor or memory resources. Also, any single-path I/O connectivity that had to be deconfigured prior to the start of the CBR action can now be configured and turned on.

Verification techniques for CBR

One of the challenges of developing firmware for a new System z stems from the fact that the new hardware is developed in parallel with firmware on a very stringent schedule. The limited access to early user hardware and

the high cost of such hardware for firmware testing constitute another challenge.

This section describes the innovative design verification techniques that we implemented to ensure that all of the complex components of EBA were designed, verified, and delivered with the high degree of quality and reliability that is expected from a System z.

Some of the functions that are needed for CBR rely on special hardware support that is built into the System z9. This includes support for the concurrent moving of large blocks of memory, moving a snapshot of a processor state

onto another physical processor, and opening and closing the ring interface that connects the books without affecting communication between the books. These functions were verified independently of the whole CBR process as soon as the first hardware was available, so that feedback could be given to the hardware development team as early as possible. This helped to ensure that the final design was robust and met its functional objectives.

Several enhancements to the z/CECSIM (Central Electronic Complex Simulator) [6] verification tool were introduced to support the CBR development. This simulator was used to verify the processor firmware that implemented the different CBR steps inside the server. In particular, the processor evacuation, adding of resources on the new book, and rebalancing of processors across the books could be verified as soon as they were implemented.

To simulate the fencing of the processors, support was added to z/CECSIM so that it would tolerate the stopping of multiple processors. (Earlier versions of z/CECSIM considered the stopping of the clocks to be a severe error and immediately halted the simulation to allow debugging.)

Traditionally, the hardware configuration of the machine to be simulated was defined before starting the z/CECSIM simulator. Special support was added to z/CECSIM to provide the capability to power off, remove, add, and power on a book while the simulation was running. This made it possible to simulate the whole CBR sequence and thus verify the interactions between the different firmware subsystems. The support element firmware was verified in parallel in a standalone support element environment.

The complexity of the CBR function required early simulation efforts in order to start the testing of CBR on the server hardware with a very high-quality code base. The team understood that this was absolutely necessary because the time available for testing on the machine was limited. (CBR testing could not start before the majority of the base machine functions were verified, because CBR makes use of so many base components.) Additionally, a large variety of machine configurations and upgrade or repair scenarios had to be tested. Because of the nature of CBR, a significant amount of time and manual intervention is required to perform one CBR operation. The early system tests were conducted as a joint development test, in which a group of developers from Endicott, Boeblingen, and Poughkeepsie worked together on the test floor with the test team, while other developers supported them remotely from the various locations. As a result of the dedication and skill of the teams and the extraordinary teamwork, short turnaround times for problem analysis and problem fixes were made

possible in spite of the high complexity and workload associated with verifying the EBA functions.

Additional functions supported with the EBA feature

Along with the concurrent book replacement function described in this paper, the EBA feature includes two other functions that support concurrent operations and are new for the System z9. The *continued capacity with fenced book* function allows a multi-book server that has had a catastrophic book hardware failure to be restarted with the defective book hardware fenced. The restarted system uses all available physical processor and memory resources on the remaining books in the system to allocate as much of the customers' purchased resources as physically possible. In prior System z machines, if a book was fenced from the server and the server restarted, the resources defined in the LICCC record for the fenced book would not be allocated when the server was restarted with the fenced book. Now, with the continued capacity with fenced book function, the LICCC resources associated with the fenced book will be used and allocated as allowed by the remaining physical resources. Also included with this function is the ability for the customer to preplan for a book hardware failure by establishing profiles that specify how the resources of the server should be allocated in the event that a book failure occurs.

The *"cold" concurrent book repair* function allows the concurrent repair of a book when that book has previously been fenced due to a hardware failure. This function allows the repair and verify operation to utilize the CBA operation during a repair scenario in order to replace the defective book concurrently with the operation of the server.

Summary

The IBM System z9 and its predecessors have always been industry leaders with respect to system reliability, availability, and serviceability (RAS). Many features have been introduced over several generations to reduce planned outages by allowing non-disruptive maintenance and non-disruptive upgrades.

The CBR development team built upon their skills and experiences gained with the mainframe's famous "always on" features such as concurrent processor sparing, concurrent book add, capacity upgrade on demand, I/O hot plug, I/O alternate path swap, and many more. Experience from every area of mainframe design was necessary to transform the enhanced book availability feature into a reality.

What seems like "technological open heart surgery" can now actually be performed on an IBM System z9. This surgery includes repairing, replacing or upgrading

processors and memory of a running system, without having an impact on the operating systems and active applications on the system. The enhanced book availability feature allows customers to adapt their System z9 servers to the rapidly changing requirements of today's business world and to perform maintenance tasks while the backbone of their business, the System z9 they rely on, is continuously operational and performing their most critical business tasks.

Acknowledgments

The authors would like to thank the System z9 design and test teams for their efforts and contributions that led to the release of this highly desired function. We especially thank the core team that was involved throughout the complete development and test cycle, which was key to the success of this project. From the test organization, our thanks go out to Dave Cole, Doug Heuvel, and Jim Brown. From the Product Engineering organization, we would like to thank Mike Gerhart, and from the Development community we would like to thank Ira Siegel, Kim Hanson, Dennis Weston, Steve Fellenz, Judy Johnson, Randy Philley, Joe Turic, Martin Stock, Christine Axnix, Martin Taubert, Andreas Muehlbach, Torsten Hendel, Victor Lourenco, Mike Gregor, Marty Bartoy, Russ Martin, Leigh Van Woert, and Ralf Schaeffler for their continuous efforts.

*Trademark, service mark, or registered trademark of International Business Machines Corporation in the United States, other countries, or both.

**Trademark, service mark, or registered trademark of Linus Torvalds in the United States, other countries, or both.

References

1. P. S. Weygant, *Clusters for High Availability: A Primer of HP Solutions*, Prentice Hall, Upper Saddle River, NJ, 2001.
2. M. R. Barber, "Increased Server Availability and Flexibility Through Failover Capability," *Proceedings of the Eleventh Systems Administration Conference (LISA '97)*, San Diego, CA, 1997, pp. 89–98.
3. M. Mueller, L. C. Alves, W. Fischer, M. L. Fair, and I. Modi, "RAS Strategy for IBM S/390* G5 and G6," *IBM J. Res. & Dev.* **43**, No. 5/6, 875–888 (1999).
4. 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, and N. E. Weber, "Reliability, Availability, and Serviceability (RAS) of the IBM eServer z990," *IBM J. Res. & Dev.* **48**, No. 3/4, 519–534 (2004).
5. U. Helmich, M. Becht, M. J. Becht, J. R. Easton, R. K. Errickson, T. Gehrmann, S. G. Glassen, S. R. Greenspan, F. Koeble, H. Lehmann, C. Mayer, J. S. Nikfarjam, F. A. Schumacher, and W. Storz, "Redundant I/O Interconnect," *IBM J. Res. & Dev.* **51**, No. 1/2, 173–184 (2007, this issue).
6. K. Theurich, A. Albus, F. Eickhoff, D. Immel, A. Kohler, E. Lange, and J. von Buttlar, "Advanced Firmware Verification Using a Code Simulator for the IBM System z9," *IBM J. Res. & Dev.* **51**, No. 1/2, 207–216 (2007, this issue).

Received March 22, 2006; accepted for publication April 25, 2006; Internet publication December 6, 2006

Christopher R. Conklin IBM Systems and Technology Group, 2455 South Road, Poughkeepsie, New York 12601 (smokey@us.ibm.com). In 1984, Mr. Conklin received a B.S. degree in electrical engineering from Rochester Institute of Technology. That same year, he joined the IBM Communications Product Division as a VLSI Test Engineer. He is currently a Senior Engineer working on the development of LIC for the IBM System z processors and was the focal point for the concurrent book add function for the IBM eServer z990 and the concurrent book replacement function for the System z9. Mr. Conklin has received one IBM Outstanding Innovation Award and four Outstanding Technical Achievement Awards; he has also coauthored one patent and two technical papers.

Carl J. Hollenback IBM Systems and Technology Group, 1701 North Street, Endicott, New York 13760 (chollenb@us.ibm.com). Mr. Hollenback is a Senior Software Engineer in the IBM System z support element (SE) and hardware management console (HMC) development group. He received an A.A.S. degree in electrical engineering technology from Pennsylvania State University in 1977, a B.S. degree in industrial engineering/computer science from the State University of New York in 1983, and an M.S. degree in information resource management from Syracuse University in 1993. He is currently the team leader for the development of the configuration management component that supports vital product data for System z, and he leads the support element contributions for the development of concurrent functions such as LICCC capacity on demand, capacity backup feature, on/off capacity on demand, concurrent book add, and enhanced book availability.

Carl Mayer IBM Systems and Technology Group, IBM Deutschland Entwicklung GmbH, Schoenaicherstrasse 220, 71032 Boeblingen, Germany (carl@de.ibm.com). In 1999, Mr. Mayer received a Graduate Engineer degree in software engineering from the University of Applied Sciences Esslingen, Germany. In 2000, he joined the IBM Development Laboratory in Boeblingen, Germany. He is currently the team leader for the group working on the hardware object model; he was the focal point for the hot-plug function for the System z9. He specializes in high-availability functions and serviceability. Mr. Mayer has received three IBM Outstanding Technical Achievement Awards and has also coauthored one patent and two technical papers.

Alexandra Winter IBM Systems and Technology Group, IBM Deutschland Entwicklung GmbH, Schoenaicherstrasse 220, 71032 Boeblingen, Germany (wintera@de.ibm.com). In 1995, Mrs. Winter received an M.S. degree in electrical engineering from the University of Karlsruhe. She worked as a research assistant at the research center for information technology (Forschungszentrum Informatik) in Karlsruhe before joining the IBM development laboratories in Boeblingen in 1997. As part of the processor firmware department, Mrs. Winter has worked on various System z and S/390 projects, including the cryptographic coprocessor, external time reference, capacity on demand, concurrent book add, and enhanced book availability. She has received an IBM Outstanding Technical Achievement Award and was the focal point for the capacity on demand functions of the IBM eServer z990 and the System z9.