

Flexible configuration and concurrent upgrade for the IBM eServer z900

by J. Probst
B. D. Valentine
C. Axnix
K. Kuehl

Flexible, nondisruptive upgrade capabilities and self-healing characteristics in IBM eServer systems are currently being addressed. The IBM eServer z900 and preceding S/390® servers have been delivering leading-edge self-configuring, self-optimizing, self-protecting, self-managing, and self-healing capabilities that provide a solid foundation for those efforts. Current processors, memory, and input/output (I/O) cards are produced with dense physical packaging at high volumes without requiring fine configuration granularity. The granularity is provided under the control of Licensed Internal Code (LIC), which enables hardware entities to fulfill specific requirements based on encrypted product data. Disabled hardware entities are “dormant” and are reserved for capacity upgrades or self-healing. Concurrent capacity upgrade functions enable dormant hardware entities to reflect the new configuration. In case of hardware failure, healthy dormant hardware can be enabled, without disruption, to replace failing hardware. This paper describes the high

level of concurrent configuration-change flexibility provided in the IBM eServer z900 and the asset protection approach on which it is based.

1. Introduction

The IBM eServer z900 provides 26 models, corresponding to different central processing unit (CPU) capacities and multiple different feature combinations, which can exploit many specialized processor types and be utilized for many different purposes. This granular structure supports specific customer demands at appropriate software license fees for IBM and vendor software. Such a large number of different hardware configurations, however, is not affordable because of the high cost of manufacturing and field-part stocking. To address this conflict, the z900 utilizes only two types of multiple-chip modules (MCMs), one with 12 processors and another with 20 processors. Every processor configuration for all of the various models contains just one of these two MCM types. During system start-up, Licensed Internal Code (LIC) allocates physical processors, designated as processing units (PUs), and initializes them as regular zSeries® CPUs, system assist

©Copyright 2002 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.

0018-8646/02/\$5.00 © 2002 IBM

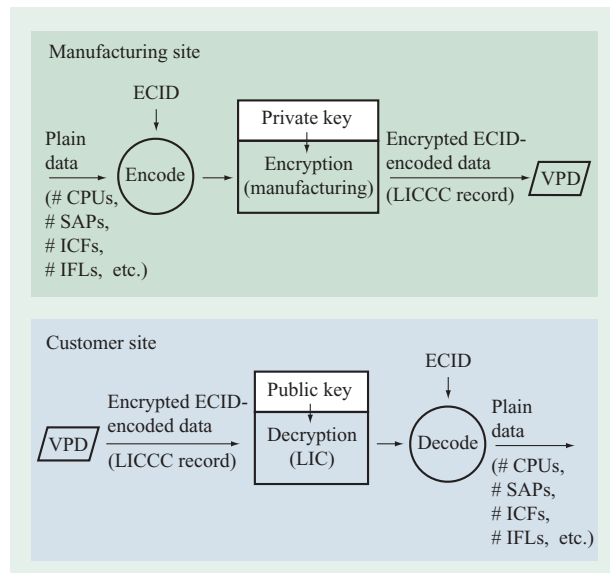


Figure 1

Basic LICCC concept.

processors (SAPs) for I/O, internal coupling facilities (ICFs), or integrated facilities for Linux** (IFLs). SAPs exclusively execute LIC for I/O and other internal purposes; ICFs and IFLs are special kinds of CPUs limited respectively to coupling and Linux workloads. The allocation is performed according to the types and quantities specified in an encrypted configuration record. PUs that are not required for the specified configuration remain in a dormant state. As spare PUs, they can be utilized for concurrent replacement of failing PUs and for concurrent capacity upgrades of CPUs, ICFs, and IFLs.

The initial marketing requirement was for the ability to create permanent upgrades by enabling dormant processor resources. Later, new requirements were added to support temporary upgrades, which provide additional capacity above the existing permanent capacity. The duration of temporary additional capacity can be limited by an expiration date or by a predefined number of days after activation. The eServer z900 uses the latter method, designated as capacity backup (CBU). Most of the applications deal with fast activation of additional capacity in cases of real emergency and tests for emergency backup. Depending upon their contracts, customers may perform the CBU upgrade by themselves without the involvement of a service representative. In Geographically Dispersed Parallel Sysplex* CBU (GDPS*-CBU), upgrades can be enabled automatically in case of a site failure or disaster. Special short-duration CBU enabling supports planned disaster-recovery tests. Concurrent CBU

“undo” returns the configuration to permanent capacity without disruption. The same facilities are used to provide finer granularity for memory size, Enterprise Systems Connection (ESCON*) ports, and ISC-3 ports. This facility also provides for spare I/O ports on an ESCON-16 card. The eServer z900 also supports permanent concurrent upgrades of pre-installed dormant memory, dormant I/O, and ISC-3 ports.

The basis for achieving these advantages for our customers is the shipment of dormant capacity that can be activated immediately without installing new hardware. Since processor capacity is a major determinant of software licensing cost, it is in the best interest of a customer to have processor capacity adjusted to need (i.e., not to pay for unused capacity in a backup computer center). To prevent mistrust, including that of independent software vendors, it is important to protect the dormant hardware with appropriate security.

Section 2 of this paper introduces the basic protection approach. Section 3 describes the various flexible configuration features that employ the protection approach and discusses the related activity at the customer’s site. Section 4 provides an overview of the design of the concurrent upgrade functions that are implemented in LIC.

2. Basic protection approach

The z900 hardware business model and vendor software license charge methodology require a secure protection mechanism to ensure the integrity of the enabling configuration data. In addition, this mechanism must be flexible and simple, and incur minimal added costs. Various known approaches were investigated. Some were very complex, and others needed dedicated special hardware and consequently could not provide the required flexibility. Another used unprotected vital product data (VPD) and flexible enabling via LIC but could not prevent cloning of the enabling configuration data. The “Licensed Internal Code-controlled configuration” (LICCC) approach overcomes the shortcomings of unprotected VPD by using

1. An intrinsic and unchangeable identifier incorporated in a chip on each respective field-replaceable unit (FRU). Modern IBM CMOS technology fabrication provides an “electronic chip ID” (ECID) as a standard service.
2. A nonsymmetric encryption method with a private/public key pair to prevent counterfeiting and protect the VPD against misuse.

Figure 1 shows the basic LICCC concept, which is based on the patent GE-995-039 (U.S. 5,982,899) [1]. The IBM manufacturing site generates the encrypted configuration

data. The plain configuration data is encoded with the ECID of a proprietary chip contained in each respective FRU. This encoding links the data to that specific hardware sample and prevents the cloning of other samples of the same FRU. To prevent manipulation, the encoded data is encrypted with a private key known only to the restricted manufacturing processes. The output of the encrypting, ECID-encoding data process is a LICCC record which is shipped as part of the VPD with the respective FRU. It may reside in flash memory or on hard disk. At the customer site, the LIC retrieves the LICCC record from the FRU VPD and performs the decryption utilizing the public key. Then the LIC retrieves the ECID from the respective FRU and verifies whether the LICCC record matches the sample to ensure that it has not been cloned from the LICCC record of a different hardware sample.

3. LICCC features

Initially, LICCC was provided only for the function of permanent increase in the number of processors. This upgrade required a service representative to be on-site with a diskette containing the new LICCC record. Over time, support has been added for temporary upgrades, electronic upgrades, and even customer control of applying upgrades. Upgrades can now be done more quickly than in the past, since a service representative does not have to be on-site with physical hardware, and in some cases customers can actually apply the change themselves.

LICCC also plays a fundamental role in dynamic processor sparing and replacement of a defective multi-chip module (MCM). On the z900, the use of LICCC has been expanded. It includes concurrent upgrade on demand (CUoD) for memory and for I/O (ESCON-16, ISC-3). The LICCC also supports ESCON port sparing (introduced in the z900) and replacement of defective ESCON-16 and ISC-3 cards.

Permanent upgrade

Processors

LICCC permanent processor upgrades, introduced in 1996 on Generation 3 of the IBM S/390*, are used to permanently upgrade the number of processors in the system without adding any hardware [2]. The dormant processors in the MCM are activated by applying a new LICCC record. When a customer orders additional processors (CPUs, SAPs, ICFs, or IFLs), IBM manufacturing checks the VPD of the MCM for which the order was placed. If there are sufficient dormant processing units (PUs) on that MCM to accommodate the order, IBM manufacturing creates a diskette containing the new LICCC record indicating the total number of CPUs, SAPs, ICFs, and IFLs. This diskette is shipped as a

“miscellaneous equipment specification” (MES) upgrade to the customer’s site. A service representative uses the support element (SE) to import the new LICCC configuration diskette and performs the upgrade.

The processor upgrade, except for SAPs, is done concurrently with zSeries operation. If the server is running in logically partitioned (LPAR) mode, the additional processors will generally be added to the shared pool of processors. However, if the customer has planned ahead and defined “reserved” processors in addition to the “initial” processor pool in an image profile, these processors can be added as dedicated processors for that image.

This LICCC processor upgrade can also be done when the server is not running. In this case, the LICCC data is imported by the SE, and the new processor configuration is defined during the next power-on reset (POR). This is sometimes referred to as a disruptive LICCC upgrade.

Memory

Similarly, both concurrent and disruptive permanent upgrades can be done for memory. If there is sufficient dormant memory available, the memory upgrade can be performed concurrently; otherwise, the upgrade requires the disruptive replacement of memory cards. The current product offering does not provide plan-ahead for memory. Memory cards are always shipped at the smallest size that physically contains sufficient storage to allow for the LICCC-defined enabling.

I/O

The z900 also supports both concurrent and disruptive permanent LICCC upgrades for ESCON-16 and ISC-3 cards. The ESCON-16 card contains a dormant port dedicated as a “spare” port, which is not available for LICCC upgrade. Depending on the number of ESCON ports ordered, in most cases additional dormant ports are available and can be used for LICCC upgrades or port sparing. Similarly, depending on the number of ISC-3 links ordered, dormant ports are available on ISC-3 cards for LICCC upgrade. Planning ahead for LICCC I/O growth is not as critical as it is for processors, since ESCON-16 and ISC-3 cards can be “hot-plugged” concurrently into system operations.

A concurrent LICCC I/O upgrade requires more than just importing a LICCC record from a diskette. The service representative must plug the cable(s) into the port(s) which are being LICCC-upgraded. The service representative or customer may also use the channel path ID (CHPID) assignment function on the SE (via single-object operations on the hardware management console) if there is a need to reassign the CHPID number(s) from their default setting(s).

Temporary processor upgrade

LICCC temporary processor upgrades, introduced on the S/390 G4, are used to temporarily upgrade the number of processors during an emergency or disaster situation [3]. For example, if a server in a sysplex experiences a problem, another server in that same sysplex is temporarily upgraded for up to 90 days, and then restored to the original, permanent LICCC record level. This capability is implemented by overlaying the permanent LICCC record with an emergency backup (EBU) record, which increases the processor capacity. However, this requires the customer to contact the IBM support center, which in turn contacts product engineering (PE) to generate the EBU LICCC record. This record is transferred electronically to a service representative to generate a LICCC diskette, or the LICCC record is read to the service representative over the telephone, to be entered on the SE. This is a time-consuming process, especially considering the time required for the service representative to arrive at the customer's site and contact PE to generate the EBU record.

A new capacity backup (CBU) function was introduced on G5. With this function, a separate CBU record is loaded onto the SE hard disk in addition to the permanent LICCC record. This allows the support center to obtain the LICCC password directly from RETAIN¹ to activate the CBU. They can remotely activate the CBU, or have the customer or service representative enter the password to activate the CBU. On z900, a "customer-activated CBU with RETAIN validation" feature, which eliminates any telephone calls to IBM support personnel, was introduced. A service representative, the customer, or an automatic application triggers a function that automatically dials into RETAIN to validate CBU activation. This method is predominantly used today except in customer locations that do not have a RETAIN connection. In that case, the support center is called to obtain the password for CBU activation.

Disaster-recovery feature

On z900, the CBU LICCC records are installed with a disaster-recovery feature, which ensures that there are sufficient dormant processors to satisfy the CBU capacity requirement. In case of a processor failure, LIC verifies that the processors remaining dormant after a dynamic CP sparing (DCS) are sufficient to fulfill CBU activation. Otherwise, a "call-home" for MCM replacement is generated, even though the permanent LICCC record processor configuration is still fully satisfied.

For example, a z900 using the 20-processor MCM with five processors activated by the permanent LICCC record

(15 dormant processors) and a CBU LICCC record requiring an additional 13 processors does not call home for MCM replacement after a single DCS event, since the disaster-recovery feature can still be fulfilled. However, when additional DCS events reduce the number of dormant processors and prevent the full 18-processor CBU activation, a call-home for MCM replacement is made.

CBU activation types

The CBU record provides support for

- One real CBU activate for up to 90 days.
- Five test CBU activates, each test for up to 10 days.

A customer need not wait for a disaster to occur in order to activate the increased capacity of the CBU. To demonstrate disaster-recovery readiness, customers use the "test" CBU activation to verify that the software correctly utilizes the increased capacity. The test activation allows the customer ten days of testing. If the customer does not "undo" or back off the CBU after this time, the server's performance is automatically decreased. The customer can execute the test CBU activation up to five times. If additional tests are necessary, the customer must purchase another CBU record. The "real" CBU activation can only be used once and lasts for 90 days. Once this is used, the customer must order another CBU record.

Activation techniques

The real disaster-and-test CBUs can be activated by the customer using one of the following techniques:

- CBU activation via password from the support center.
- CBU activation via RETAIN validation.
- Geographically Dispersed Parallel Sysplex (GDPS).
- Hardware management console/support element simple network management protocol application programming interface (HMC/SE SNMP API).
- SE panel (via HMC single-object operations).

These actions can be done concurrently with z900 operating system execution or disruptively, depending on the state of the system.

CBU activation via password from the IBM support center can be used by a customer that does not have a RETAIN connection or experiences a problem connecting to RETAIN. The customer contacts the IBM support center to obtain the CBU password. The customer then uses the HMC and single-object operations to connect to an SE panel and enter the password for the CBU activation.

CBU activation via RETAIN validation is the most used activation technique. There are three different ways to

¹ RETAIN: Remote Technical Assistance Information Network, an IBM-internal database of technical-support resource information.

invoke it, but the underlying design works the same way in all cases. GDPS provides support to monitor and control a sysplex of local servers, multi-site and potentially spread around the world. GDPS can be configured to monitor for a server failure within the sysplex, and if one is detected, to activate the CBU of another server in the sysplex. GDPS uses the HMC simple network management protocol (SNMP) application programming interface (API) to activate the CBU. This activation API is sent over the network to the HMC, which in turn signals the SE to activate the CBU. The SE requests a connection to RETAIN to validate that the requested server is enabled to allow CBU activation. If so, the CBU password is downloaded to the SE. At that point, the SE performs the CBU activation, and an API-completion status message is sent back to GDPS. The HMC/SE SNMP API allows customers to write their own automation programs for general monitoring and controlling of their machines. The HMC/SE SNMP API for CBU activation can be issued directly from a program written by the customer.

Some customers prefer to allow manual initiation of the CBU activation. These customers use the HMC to connect to the SE panel to initiate the activation with RETAIN validation. The customer invokes the function without entering any password, and just as in the case of HMC/SE SNMP API activation of the CBU, the SE automatically obtains the password from RETAIN. The SE panel can also be used if the customer perceives or anticipates a problem which may not be detected by automation.

Undo CBU

Prior to the expiration of the CBU, the customer must perform the “undo CBU” function to restore the capacity to the permanent LICCC record level. If this is not done despite multiple automatic warnings, server performance will be significantly decreased.

The undo CBU function can be carried out using GDPS, HMC/SE SNMP API, or the SE panel. This can be performed concurrently with z900 operations² or on a disruptive basis, depending on the state of the system and the state of the processors being removed. In order for it to be done concurrently, the customer (or automation application) must configure offline the processors that are being removed. After the request has been made via GDPS, API, or the SE panel, the SE initiates a request to the CEC LIC to undo the CBU.

CBU query information

The customer (SE panel) or automation program (GDPS or customer-provided program using a HMC/SE SNMP API) can query how much time is left on the CBU

activation. This information assists in determining when to invoke the undo CBU function and whether a new CBU feature should be ordered.

Spare hardware utilized for hardware failures

Dormant processors and ESCON ports can be used for upgrades or to replace defective ports. If an active processor fails, it is concurrently spared while the z900 operating system is executing. ESCON ports (channel path IDs, or CHPIDs) are also concurrently spared, but require the service representative to remove the cable from the defective port and plug it into a spare port. The CHPID is configured offline while this operation occurs, but the rest of the z900 operating system will continue to run. Once the sparing is complete, the CHPID is configured back online.

After a processor or ESCON port-sparing event, the hardware VPD is updated and transmitted to IBM. This information is used by order processing to assess whether sufficient dormant hardware is available to fulfill an upgrade request. If not, a new part is shipped, together with its LICCC diskette. The disaster-recovery feature is also validated after a sparing action to ensure that there are enough spares left to satisfy a potential CBU activation.

Hardware repair and replacement

When a LICCC-controlled FRU (MCM, memory, or channel card) is replaced as part of a repair action, the SE repair and verify (R&V) function obtains the LICCC record for the new FRU from RETAIN. RETAIN contains a database of all permutations of all possible enablements for each field-stocked LICCC FRU. During a repair action, R&V connects to RETAIN, sending the plain data with the serial number of the FRU being replaced and the serial number of the new FRU. RETAIN returns the correct LICCC record for the new FRU matching that of the defective FRU.

This is generally done automatically. However, in some cases a RETAIN connection cannot be successfully established, or the customer does not allow connection to RETAIN for security reasons. In these cases, the service representative creates the LICCC diskette and brings it to the customer's location during the R&V process. As a last resort, the service representative can type into an SE panel the 192-byte LICCC record.

4. Technical designs for concurrent up/downgrades

The following section describes some of the most challenging new designs required to execute upgrades and downgrades concurrently. It focuses on processors and memory. (Upgrading of LICCC-controlled I/O ports did not require new methods to bring such resources

² C. Axniz, K. Kühn, A. Mühlbach, J. Probst, C. J. Hollenback, and J. Kubala, “Flexible Temporary Capacity Upgrade/Downgrade in a Computer System Without Involvement of the Operating System,” U.S. patent pending.

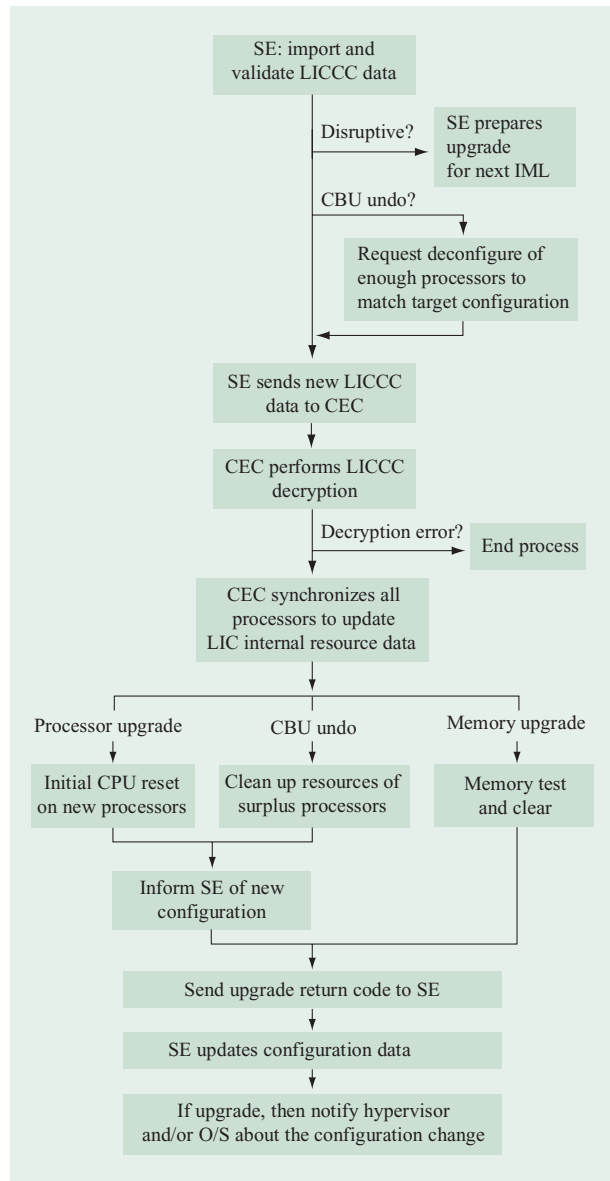


Figure 2

General control flow of concurrent upgrades or downgrades.

concurrently online. The existing methods for concurrent I/O upgrade by plugging additional I/O cards could be reused, e.g., dynamic I/O and flexible CHPID mapping. The concurrent protection validation for I/O is not different from the design for processors and memory.)

Figure 2 shows the general control flow used for the functions concurrent processor upgrade, concurrent CBU undo, and concurrent memory upgrade. For all of these functions, the process begins with the SE importing and validating the new LICCC record. During the validation

process, it may turn out that the upgrade cannot be done concurrently, e.g., due to invalid configuration change requests. In this case, the SE prepares everything for an automatic upgrade or downgrade with the next power-on reset (POR). Otherwise, the request for up/downgrade is sent to the central electronic complex (CEC) for processing, and, depending on the type of up/downgrade, different actions are performed on the CEC. After these actions are completed, the SE receives the result of the operation and performs the finishing actions, including updating of internal data areas and the VPD.

Concurrent processor upgrade

As previously described, concurrent processor upgrade provides additional processors (CPUs, ICFs, IFLs) to a running server without requiring an outage. During the concurrent processor upgrade process, dormant processors on the MCM are converted to CPUs, ICFs, and/or IFLs as specified by the new LICCC record. The concurrent upgrade can be permanent ("capacity upgrade on demand," or CUoD) or temporary ("capacity backup," or CBU), but the underlying design to convert the dormant processor to full operation is the same.

To accommodate concurrent processor upgrade, some preparation is done during POR. Mainly, LIC data areas are allocated for all dormant processors, since the respective storage area is frozen at POR time. On the CEC, the upgrade process is controlled and monitored by one of the system assist processors (SAPs). A SAP is a processor that has no CPU function assigned but exclusively executes LIC for I/O and service functions. One of the SAPs acts as the controlling processor for the upgrade. It receives the new LICCC record from the SE and decrypts it. After a successful decryption, the requested processor configuration is compared against the current configuration, and certain checks are performed to determine whether the upgrade request can be fulfilled. If sufficient dormant processors are available to fully satisfy the concurrent processor upgrade request, the upgrade is completed successfully. If not (possibly due to a processor-sparing event which occurred between ordering the upgrade and activating it), as many processors as possible are added, and the upgrade request is completed with partially successful status. This way, the customer can exploit at least part of the new configuration until the MCM is replaced.

To update the processor configuration, all running processors must be quiesced prior to the configuration update and released afterward. This is to ensure that no processor works on only partially updated configuration information. After the processor configuration, an "initial CPU reset" is triggered on the new processors, be they CPUs, IFLs, or ICFs. This forces the new processors to initialize their internal structures and the z900 architected

facilities required for CPU/IFL/ICF operation. Once initialized, the processors are placed in “standby,” and are thus prepared to accept “configure on” commands from the LPAR hypervisor or from the operating system. As part of the initial CPU reset, the SE is also notified of the existence and status of the new processors.

The controlling processor for the upgrade monitors the new processors and waits until they complete their initial CPU reset. When all of the new processors have completed the reset or the wait time-out has been reached, the SE is notified that the concurrent processor upgrade process has finished, and a return code is sent indicating the success of the operation.

When the concurrent processor upgrade is fully or partially successful, the SE triggers a notification to the LPAR hypervisor and/or the operating system indicating that a change in the processor configuration has occurred. The LPAR hypervisor and/or the operating system use standard z900 architecture facilities to retrieve the new processor configuration information and to bring the new processors into the active processor configuration.

Concurrent CBU undo

The concurrent CBU undo feature restores the current CBU configuration to its permanent configuration before the CBU activation. This new z900 feature eliminates the need for an outage every time a real or a test CBU is completed. The concurrent CBU undo behaves as a permanent concurrent processor downgrade request.

A prerequisite for a concurrent CBU undo is that no LPAR hypervisor or O/S workload be running on the surplus processors (the processors that must be removed from the configuration to restore it to the original permanent configuration). The number of surplus processors may not necessarily equal the number of processors that were added during the CBU activation, since a “master SAP reassignment” event could have occurred [4] since the CBU activation. There are several ways to ensure that no workload is running on the surplus processors:

1. With GDPS, the “controlling z/OS* image” deconfigures the surplus processors in the native or LPAR z/OS images.
2. In LPAR mode, the SE triggers the LPAR hypervisor to deconfigure the surplus processors from the pool of shared processors.
3. In native mode without GDPS, a message is displayed on the SE panel, instructing the system operator to deconfigure the appropriate number of processors manually.

After the surplus processors are freed from load and set into standby state, the SE sends the permanent LICCC

information to the CEC, which performs the decryption and checks to ensure that active processor resources do not exceed the target configuration (CPUs, IFLs, ICFs).

After a successful configuration check, the surplus processors can be removed from the configuration. As in the upgrade case, all running processors are quiesced in order to ensure consistent updating of the LIC internal configuration data. In addition, all LIC internal processor resources on the surplus processors are reinitialized to an IML-complete state. Any locks on system global resources held by those processors must be released. As a result of the concurrent CBU undo, the surplus processors are converted to spare processors which can later be used either for another CPU upgrade or for a sparing action in the case of a failing processor.

When the CBU undo process has completed on the CEC, the new configuration information is sent to the SE as well as the return code of the CBU undo operation. A completion notification to the hypervisor or to the operating system is not necessary at this point, because they have already done their part by releasing the requested surplus resources.

Concurrent memory upgrade

The concurrent memory upgrade function allows the customer to add memory to his system without doing an IML. This function is available only in LPAR mode. During IML, the memory self-test is performed on all of the physically installed memory to ensure that the entire memory is good. As with the concurrent processor upgrade and the CBU undo, the concurrent memory upgrade process begins with the SE transferring the new LICCC information to the CEC for decryption. The process stops on any error, e.g., LICCC record “sanity checks” or decryption errors. On successful decryption, all processors are quiesced while the LIC internal memory configuration information is updated. To prepare the additional memory for use, it is tested and cleared. This happens concurrently with system operation, and is done in pieces to minimize the performance impact. Memory failures found during this initialization are handled in the same way as during regular operation. This applies to DRAM sparing, call-home policy, and uncorrectable storage error (UE) handling. The upgrade never ends with partial success; it always completes. During the memory test-and-clear process, the CEC-LIC will update the storage status maps with the UE information, which can later be retrieved by the LPAR hypervisor using the standard z900 architecture facilities.

When the entire new storage area has been tested and cleared, the SE is notified of the completion of the concurrent memory upgrade. The SE then notifies the LPAR hypervisor about the number of added memory increments which are now available for use. Finally, the

SE updates the reset profiles to reflect the new memory configuration.

Concluding remarks

The IBM eServer z900 provides flexible concurrent configuration to an extent never considered when the basics of LICCC were introduced in 1996 with the S/390 CMOS Generation 3 system. Extensive improvements in LICCC make possible concurrent upgrades in processor capacity, memory size, and I/O ports. Flexible initialization of processors for different purposes makes it possible for customers to add specialized processors (e.g., for Linux) without affecting the software license capacity of the system. Dormant hardware, secured by LICCC, is used for sparing in case of hardware failures. The latest versions of CUoD make it possible for customers to upgrade their system capacity immediately, without waiting for involvement by a support representative. Nearly all of those enhancements have been added as a result of customer demand, at a reasonable lead time, demonstrating the flexibility of the base LICCC approach.

Acknowledgments

The authors wish to thank our IBM colleagues for their innovative ideas on translating new customer requirements into design enhancements. Their many contributions over the years helped to extend the initial G3 LICCC base to today's z900 level. The authors would also like to acknowledge L. C. Alves, G. Sandner, and C. Perkins for many helpful discussions regarding this paper.

*Trademark or registered trademark of International Business Machines Corporation.

**Trademark or registered trademark of Linus Torvalds.

References

1. J. Probst, "Method for Verifying the Configuration of a Computer System," U.S. Patent 5,982,899, 2000.
2. G. Doettling, K. J. Getzlaff, B. Leppla, W. Lipponer, T. Pflueger, T. Schlipf, D. Schmunkamp, and U. Wille, "S/390 Parallel Enterprise Server Generation 3: A Balanced System and Cache Structure," *IBM J. Res. & Dev.* **41**, No. 4/5, 405–428 (1997).
3. C. F. Webb and J. S. Liptay, "A High-Frequency Custom CMOS S/390 Microprocessor," *IBM J. Res. & Dev.* **41**, No. 4/5, 463–474 (1997).
4. 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 (2002, this issue).

Received August 27, 2001; accepted for publication February 11, 2002

Juergen Probst *IBM Deutschland Entwicklung GmbH, Schoenaicherstrasse 220, 71032 Boeblingen, Germany (jprobst@de.ibm.com).* Dr. Probst is a Senior Engineer working on IBM eServer z900 system design implementation. He studied electrical engineering at the Technical University of Aachen (RWTH) and received an M.S. (Dipl. Ing.) degree in 1978. Until 1984 he worked with the Institute of Automatic Control of the RWTH, where he received a D.Sc. (Dr. Ing.) degree. He joined IBM in 1984 to work on system initialization and configuration. He held various positions in S/390 CMOS LIC development and led the transfer of LPAR to the first IBM Boeblingen LPAR machine. In 1998 Dr. Probst founded the S/390 Global System Design Board with members from IBM Poughkeepsie, Endicott, and Boeblingen, and he still chairs the Board. He is currently working on system design implementation for the z900 follow-on product. To retain customer contacts, Dr. Probst supports the Boeblingen Briefing Center with z900 presentations.

Brian D. Valentine *IBM Server Group, 1701 North Street, Endicott, New York 13760 (bdvalent@us.ibm.com).* Mr. Valentine is a Senior Programmer working in the HMC (hardware management console) and SE (support element) Licensed Internal Code Development Group. He graduated from Pennsylvania State University in 1983 with a B.S. degree in computer science, joining IBM that same year to work in the 9370 Support Processor Licensed Internal Code Development Group. He has also worked in the Support Processor Licensed Internal Code Development groups for the IBM 9371, 9221, CMOS G1 to G6 processors, and the z900.

Christine Axnix *IBM Deutschland Entwicklung GmbH, Schoenaicherstrasse 220, 71032 Boeblingen, Germany (caxnix@de.ibm.com).* Mrs. Axnix received the Dipl. Ing. degree in electrical engineering from the Berufsakademie in Stuttgart in 1989. In 1989 she joined the IBM development laboratories in Boeblingen in the Product Assurance organization, where she was responsible for the development and maintenance of S/390 processor and coupling architecture verification test programs from the development laboratory in Poughkeepsie, New York. In 1993 she joined the processor LIC development department, where she has worked on various S/390 projects. Mrs. Axnix led the overall LIC development effort for CuOD and CBU in 1998/1999, and for concurrent memory upgrade and concurrent CBU undo in 2000/2001. She is currently working on the LIC design and code development for the eServer z900 follow-on product.

Klaus Kuehl *IBM Deutschland Entwicklung GmbH, Schoenaicherstrasse 220, 71032 Boeblingen, Germany (kuehlk@de.ibm.com).* Mr. Kuehl joined IBM in 1973 as a Telecommunications Engineer in the IBM branch office in Stuttgart. In 1980, after seven years working on field engineering for S/370 systems, he joined the IBM laboratories in Boeblingen, where he has worked in different areas as firmware and support element code developer on various S/390 projects such as the IBM 9370, 9221, CMOS G1 to G6 processors, and the z900. In 1994, he joined the support element development team, where, since July 1996, he has been responsible for the design and implementation of LICCC and its features.