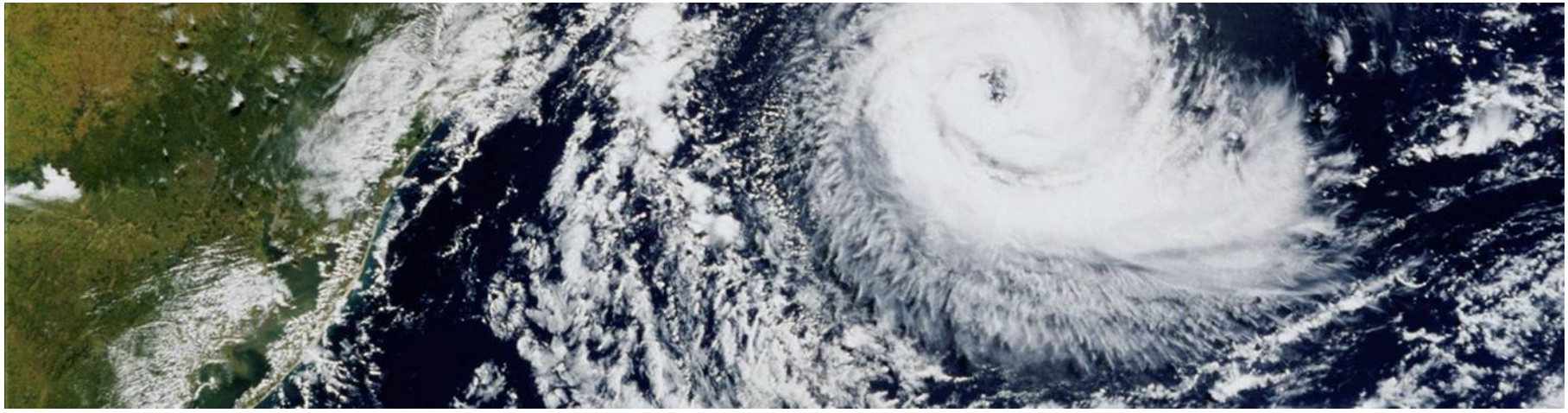
Linux on System z
Reliability, Availability and Serviceability

Dr. Reinhard Bündgen (buendgen@de.ibm.com)



Trademarks

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

/ marks used by IBM are listed on this page. Failure of a mark to appear does not mean that IBM does not use the mark nor does it mean that the product is not actively marketed or is not significant within
ollowed by ® are registered trademarks of IBM in the United States; all others are trademarks or common law marks of IBM in the United States.

ee www.ibm.com/legal/copytrade.shtml:

N, IBM®, IBM (logo)®, iSeries®, MVS, OS/390®, pSeries®, RS/6000®, S/30, VM/ESA®, VSE/ESA, WebSphere®, xSeries®, z/OS®, zSeries®, z/VM®, System i, System i5, System p, System p5, System

The following are trademarks or registered trademarks of other companies.

PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.
s a trademark of Sony Computer Entertainment, Inc. in the United States, other countries, or both and is used under license therefrom.
trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.
dows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.
de, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the U
lemark of The Open Group in the United States and other countries.
lemark of Linus Torvalds in the United States, other countries, or both.
mark, and a registered community trademark of the Office of Government Commerce, and is registered in the U.S. Patent and Trademark Office.
s a registered trademark of the Central Computer and Telecommunications Agency, which is now part of the Office of Government Commerce.

* All other products may be trademarks or registered trademarks of their respective companies.

The actual throughput that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload proces

and the results they may have achieved. Actual environmental costs and performance characteristics will vary depending on individual customer configurations and conditions.
ıe information may be subject to change without notice. Consult your local IBM business contact for information on the product or services available in your area.

ıcts and cannot confirm the performance, compatibility, or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Abstract

The System z platform is designed for premium Reliability, Availability and Serviceability (RAS). This presentation describes how customers running Linux on System will benefit from the outstanding RAS features of System z hardware and z/VM. In addition it discusses the special efforts of the Linux on System z team to make Linux on System the most reliable, available and serviceable Linux by ensuring the highest levels of data integrity, minimizing the occurrences and extends of outages and optimizing the means to help those customers that encounter problems.

Overview

- Introduction to RAS
- System z HW RAS
- z/VM RAS
- Linux on System z RAS
- (Platform Solutions for Business Continuity)

What is RAS?

- **Wikipedia** **(http://en.wikipedia.org/wiki/Reliability,_Availability_and_Serviceability):**

  * **Reliability** means features that help avoid and detect such faults. A reliable system does not silently continue and deliver results that include uncorrected corrupted data, instead it corrects the corruption when possible or else stops and reports the corruption.

  * **Availability** is the amount of time a device is actually operating as the percentage of total time it should be operating. Availability may be shown as minutes or hours of downtime per year. Availability features allow the system to stay operational even when faults do occur. A highly available system would disable the malfunctioning portion and continue operating at a reduced capacity. In contrast, a less capable system might crash and become totally nonoperational.

  * **Serviceability** takes the form of various methods of easily diagnosing the system when problems arise. Early detection of faults can decrease or avoid system downtime. For example, some of IBM's systems could automatically call an IBM service center (without human intervention) when the system experiences a system fault. Traditional focus has been on making the correct repairs with as little disruption to normal operations as possible.

What is RAS?

by RB

- **Reliability**
  - whatever my computer computes is correct
- **Availability**
  - my computer works whenever I need it
- **Serviceability**
  - if there is a problem with my computer it can be fixed soon

Concepts for Implementing RAS

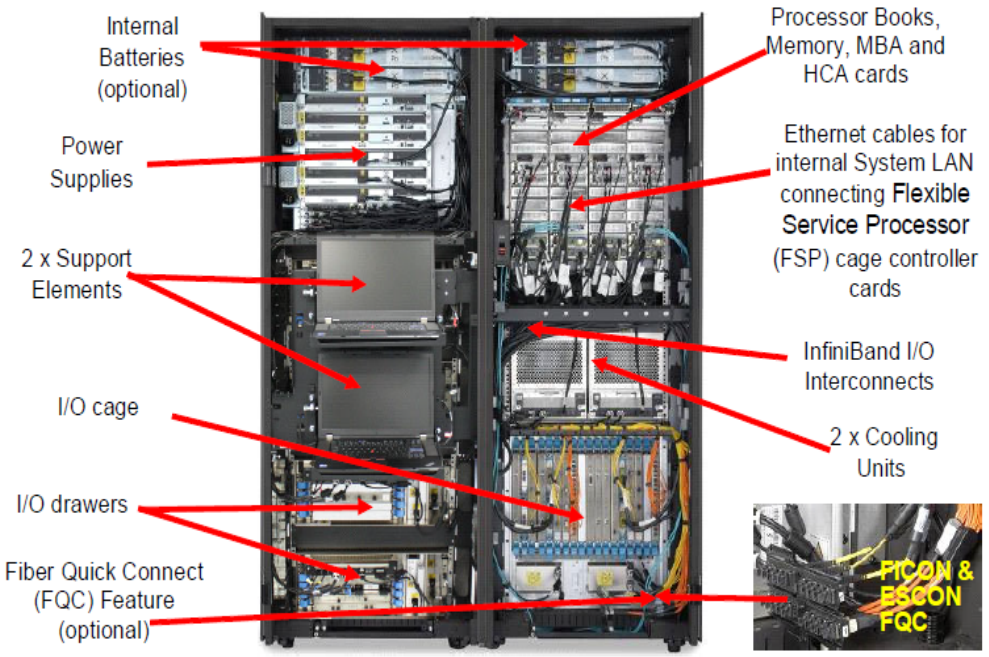
- **Reliability**:
  - quality
  - error detection, isolation, indication, correction
- **Availability**
  - quality
  - robustness
  - redundancy
- **Serviceability**
  - problem/status reporting
  - problem analysis tooling
  - service process

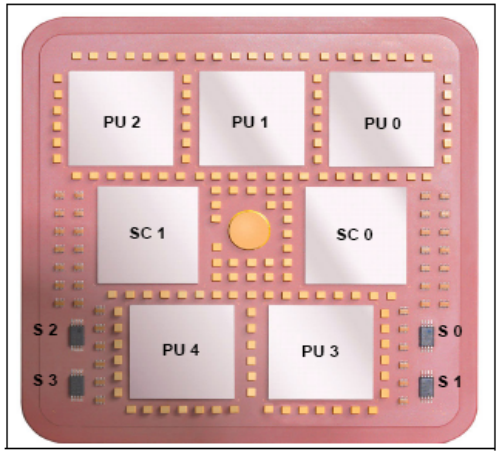
System z HW RAS Features

- HW Redundancy
- Integrity of HW Functions
- Servicing HW

HW Redundancy

- Redundant components
- Spare CPUs
- Redundant paths





Maintenance and Service of System z

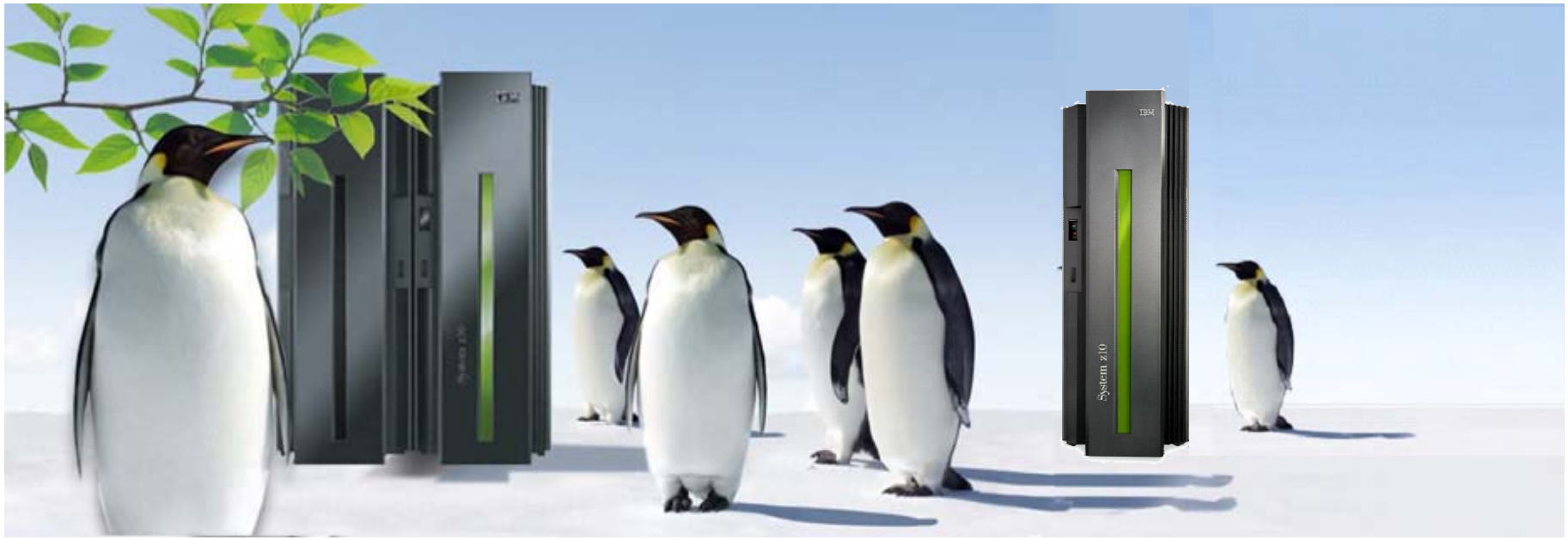
- concurrent maintenance: components & LIC
- concurrent upgrade: components & LIC
- call come
- dynamic capacity: CBU, CoD, CPE

Linux Benefits from RAS Features of z/VM

- Machine Check Handling
- Network availability
- Guest Isolation
- Diagnostic facilities (e.g. TRACE, DUMP)
- SSI / LGR

Linux on System z RAS

- Quality
  - kernel hardening
  - extended QA
- Serviceability
  - documentation
  - minimize maintenance disruption
  - collecting FFDC data, traces, logs
  - tooling for problem analysis

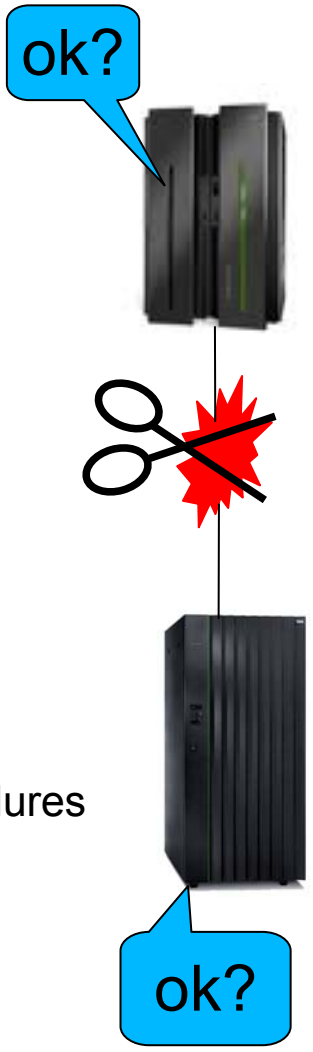


Linux on System z Quality: Kernel Hardening

- Detect and isolate HW problems
- HW Problems –> machine checks
  - inappropriate usage
  - unexpected external events
  - HW failure
- Linux on System z kernel
  - handles all machine checks
  - tries to isolate effect of HW Problem
  - recover from problem
  - detects and handles HW configuration events
  - emphasis on data integrity
- Device drivers
  - recovery from I/O errors
  - fast fail support for PAV (parallel access volume)

Linux on System z Quality: Extensive Testing

- Distribution test for Linux on System z distributions (SLES and RHEL)
  - **IBM**: test on internal driver for upstream code
  - Test by open source community
  - **IBM**: automated test suits for system & performance test
    - daily system regression tests
    - daily/weekly performance regression tests
  - System test by distribution partners
  - **IBM**: system & distribution test for Linux on System z
  - **IBM**: performance test for Linux on System
- Device qualification or extensive system test
  - for IBM storage subsystems
    - DS8K, DS6K, SVC, XIV
  - FICON & FCP
  - complex set ups (multi-path, copy services, ...)
  - extensive bad path test covering storage, fabric and connectivity failures
  - resulted in many improvement of generic SCSI stack
  - focus on data integrity
- Special Test Activities
  - large test case collection
  - increase test case base for test automation (regression testing)
  - improvements on code coverage
  - Install Day



Linux on System z Availability: Minimize Disruptions

- Automatic reboot on failure
  - IPL, dump or dump & IPL on panic
  - in z/VM: vmwatchdog
- Little/non-disruptive service
  - Reboot with alternate parameters
  - ability to dynamically vary on/off devices
  - hotplugable CPUs and memory
  - suspend/resume
- Emergency access to Linux in z/VM (without IP connection)
  - via 3270
  - Linux terminal server

Linux on System z Serviceability: Kernel Message Documentation

- For every Linux on System z specific kernel component (module)
- Kernel message documentation
  - –unique message id
  - –explanation
  - –user response
  - –severity
- Available as
  - –book, see http://www.ibm.com/developerworks/linux/linux390/documentation_dev.html
  - –man pages

Chapter 39. zfcp

This section contains messages that are issued by the SCSI-over-Fibre Channel device driver (zfcp device driver) for the QDIO-based System z SCSI-over-Fibre Channel adapter. The zfcp device driver provides support for Fibre Channel-attached SCSI devices on Linux on System z.

zfcp.000866 *<bus ID of the zfcp device>*: **The FCP adapter cannot support more NPIV ports**

Explanation: N_Port ID Virtualization (NPIV) ports consume physical resources on the FCP adapter. The FCP adapter resources are exhausted. The connection is not operational.

User response: Analyze the number of available NPIV ports and which operating system instances use them. If necessary, reconfigure your setup to move some NPIV ports to an FCP adapter with free resources.

Severity: Warning

zfcp.00beaa *<bus ID of the zfcp device>*: **ERP failed for unit 0x***<LUN>* **on port 0x***<WWPN>*

Explanation: An error occurred on the SCSI device at the specified LUN. The error recovery procedure (ERP) could not resolve the error. The SCSI device is not available.

User response: Verify that the LUN is correct. Check the fibre channel fabric for errors related to the specified WWPN and LUN, the storage server, and Linux.

Severity: Error

zfcp.01a8f2 *<bus ID of the zfcp device>*: **All NPIV ports on the FCP adapter have been assigned**

Explanation: The number of N_Port ID Virtualization (NPIV) ports that can be assigned on an FCP adapter is limited. Once assigned, NPIV ports are not released automatically but have to be released explicitly through the support element (SE).

User response: Identify NPIV ports that have been assigned but are no longer in use and release them from the SE.

Severity: Warning

zfcp.020115 *<bus ID of the zfcp device>*: **Registering port 0x***<WWPN>* **failed**

Explanation: The Linux kernel could not allocate enough memory to register the remote port with the indicated WWPN with the SCSI stack. The remote port is not available.

User response: Free some memory and trigger the rescan for ports.

Severity: Error

zfcp.058803 *<bus ID of the zfcp device>*: **FCP adapter maximum QTCB size (***<maximum supported size>* **bytes) is too small**

Explanation: The queue transfer control block (QTCB) size requested by the zfcp device driver is not supported by the FCP adapter hardware.

User response: Update the firmware on your FCP adapter hardware to the latest available level and update the Linux kernel to the latest supported level. If the problem persists, contact your support organization.

Severity: Error

zfcp.0cf3fa *<bus ID of the zfcp device>*: **Creating an ERP thread for the FCP device failed.**

Explanation: The zfcp device driver could not set up error recovery procedure (ERP) processing for the FCP device. The FCP device is not available for use in Linux.

User response: Free some memory and try again to load the zfcp device driver. If the zfcp device driver has been compiled into the kernel, reboot Linux. Consider assigning more memory to your LPAR or z/VM guest virtual machine. If the problem persists, contact your support organization.

Severity: Error

zfcp.10efb5 *<bus ID of the zfcp device>*: **Opening WKA port 0x***<destination ID of the WKA port>* **failed**

Explanation: The FCP adapter rejected a request to open the specified well-known address (WKA) port. No retry is possible.

User response: Verify the setup and check if the maximum number of remote ports used through this adapter is below the maximum allowed. If the problem persists, gather Linux debug data, collect the FCP adapter hardware logs, and report the problem to your support organization.

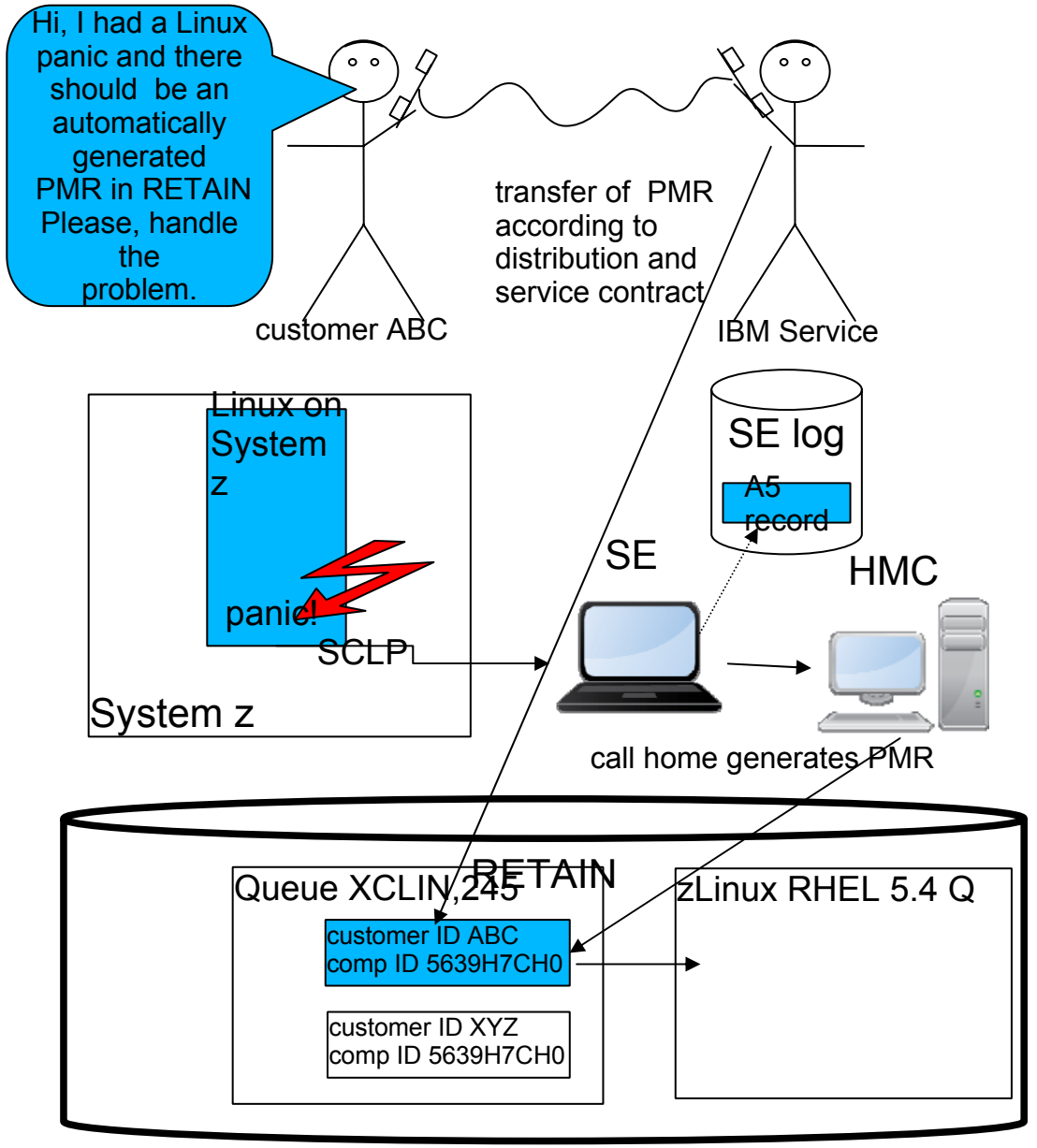
Severity: Warning

zfcp.128ff1 *<bus ID of the zfcp device>*: **The WWPN assignment file on the FCP adapter has been damaged**

Explanation: This is an FCP adapter hardware problem.

Linux on System z Serviceability: Call Home on Panic

- HW Call Home function can be configured to report kernel panics for Linux images running in LPARs
- early alert to service and development teams
- customer can reference to automatically generated PMR in service call



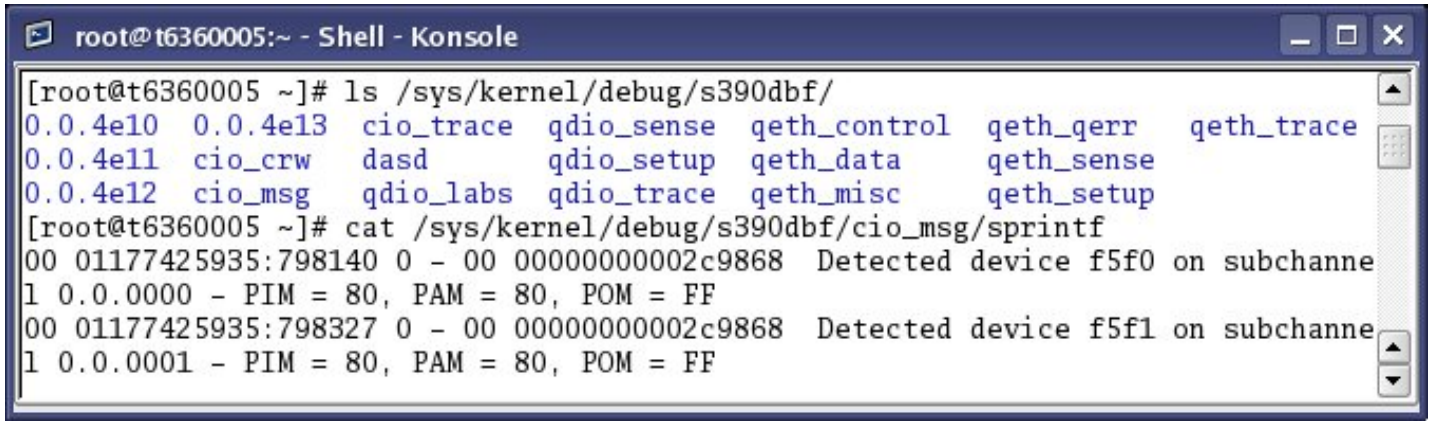
animated slide

Linux on System z Serviceability: Efficient Error Isolation

- Dbginfo.sh
  - Tool to capture most important data for problem analysis
- Debug feature (s390dbf)
  - Every System z component records key events to s390dbf
  - Read via debugfs or with lcrash/crash
- Sense/trace information logging
  - For DASD, zFCP
  - Can be enabled dynamically
- Performance information
  - For FCP, NW (qdio/qeth)
  - Can be enabled dynamically
- Analysis of virtual LANs
  - Guest LAN sniffer

Linux on System z Serviceability: S390 Debug Feature (s390dbf)

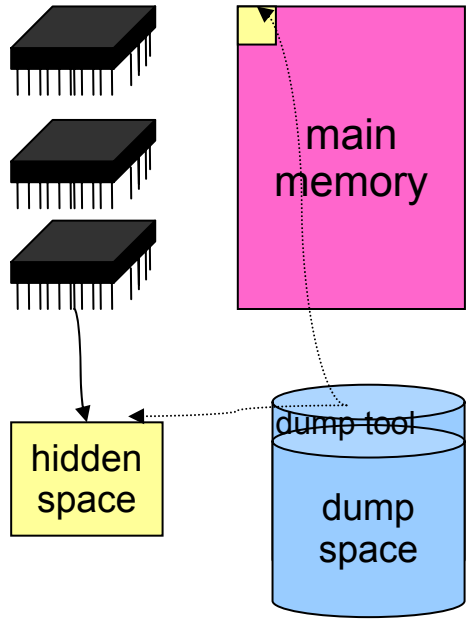
- z/Linux specific driver tracing environment
- Uses component specific wraparound memory buffers
- Available in live system and in system dumps
- Live System:



- Views: hex_ascii and sprintf
- Set Level: `# echo 6 > level`
- Flush dbf: `# echo - > flush`
- Increase buffer size: `# echo 10 > pages`

Linux on System z Serviceability: Out-of-band Dump

- **Goal**
  - store all CPU states and all of main memory
- **Procedure**
  - preparation
    1) write dump tool as IPL program to dump device (using zipl)
  - dumping
    1) stop all CPUs and store CPU state (into some hidden space)
    2) IPL dump tool (possibly with special dump option)
    3) dump tool saves (while running in main memory) saves stored CPU states and original contents of main memory to dump space
    4) a Linux is IPLed and used to read dump from dump space (zgetdump)



Linux on System z Serviceability: Out-of-band Dump Tools

- Triggered from HMC/SE or guest console
- work always, independent of status of crashed system
- supported dump media
  - –DASD (multi volume)
  - –SCSI disks
  - –FICON tapes
- do not require space in image of Linux
- in comparison: kdump
  - –is standard Linux technology (+)
  - –triggered, by panic, requires minimal functionality of kernel (-)
  - –reserves space for dump kernel and ram disk (-)
  - –requires devices to be in recoverable state (-)

Tool	Stand alone tools			VMDUMP
	DASD	Tape	SCSI	
Environment	VM&LPAR		VM&LPAR	VM
Preparation	zipl -d /dev/<dump_dev>		mkdir /dumps/mydumps; zipl -D /dev/sda1 ...	---
Creation	Stop CPU & Store status ipl <dump_dev_CUU>			VMDUMP
Dump medium	ECKD or FBA	Tape cartridges	LINUX file system on a SCSI disk	VM reader
Copy to FS	zgetdump /dev/<dump_dev> > dump_file		---	Dumpload ftp ... vmconvert ..
Compression	no	yes	yes	no
Viewing	lcrash or crash			

- See “Using the dump tools” book on http://www-128.ibm.com/developerworks/linux/linux390/index.html

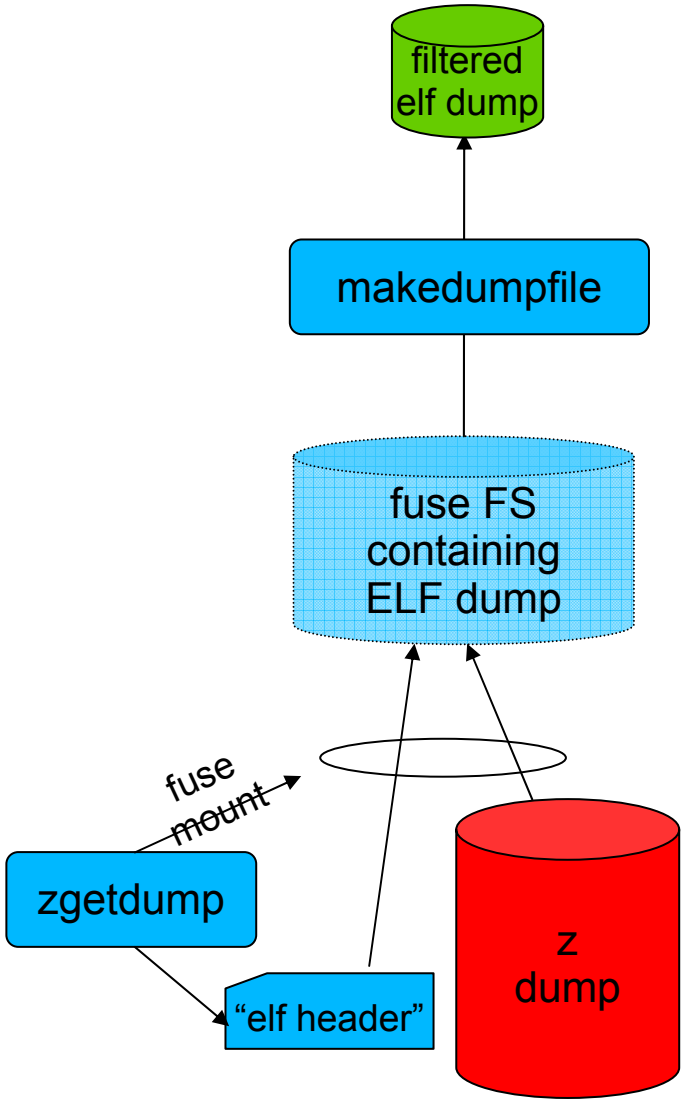
Dealing with Large Dumps

- More and more customers run large images >100GB
- Problem:

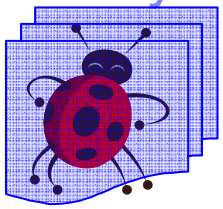
–How do you send half a TB of data?

- Solution:

1) update of dump tool documentation: how to split dumps
2) to analyze kernel problems only kernel data is needed:
•filter out non-kernel pages
•makedumpfile tool ported to System z can filter out
- file caches
- user space pages
- free pages
• zgetdump creates virtual elf dump



Beyond RAS: Linux on System z Support for HA & DR



Support for HA

- STONITH support
  - snIPL for z/VM guests & LPARs
- watch dog timer support in z/VM
  - vmwatchdog
- virtual IP address support for layer 3 NW adapters
  - IP address take over

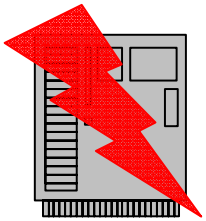
e VIPA

- protecting cluster image control from uncoordinated manual control
  - control program identification
- supported HA solutions
  - IBM Tivoli System Automation Multiplatform
  - Linux HA (SLES)

Support for Disaster Recovery

- GDPS/PPRC Multiplatform
  - protect from
    - planned outages
    - unplanned outages
  - Hyperswap & site failover
  - for Linux on System z images
    - on z/VM guests
    - on LPARs

- Disk Mirroring with Real Time Enhancements
  - special offering



- XRC support
  - time stamps

Summary

- System z HW is architected for RAS
  - for all components and
  - on all levels
- Linux on System z profits from the RAS features of System z HW and z/VM
- Linux on System z invests a lot in reliable device drivers
  - Emphasis on data consistency
  - extensive device testing
- Linux on System z features a lot of tooling for quick problem determination

Thank You!

Do You Have Questions?

Dr. Reinhard Buendgen

Linux on System z
Architect for Crypto & RAS

IBM Deutschland Research & Development GmbH
Schoenaicher Strasse 220
71032 Boeblingen, Germany

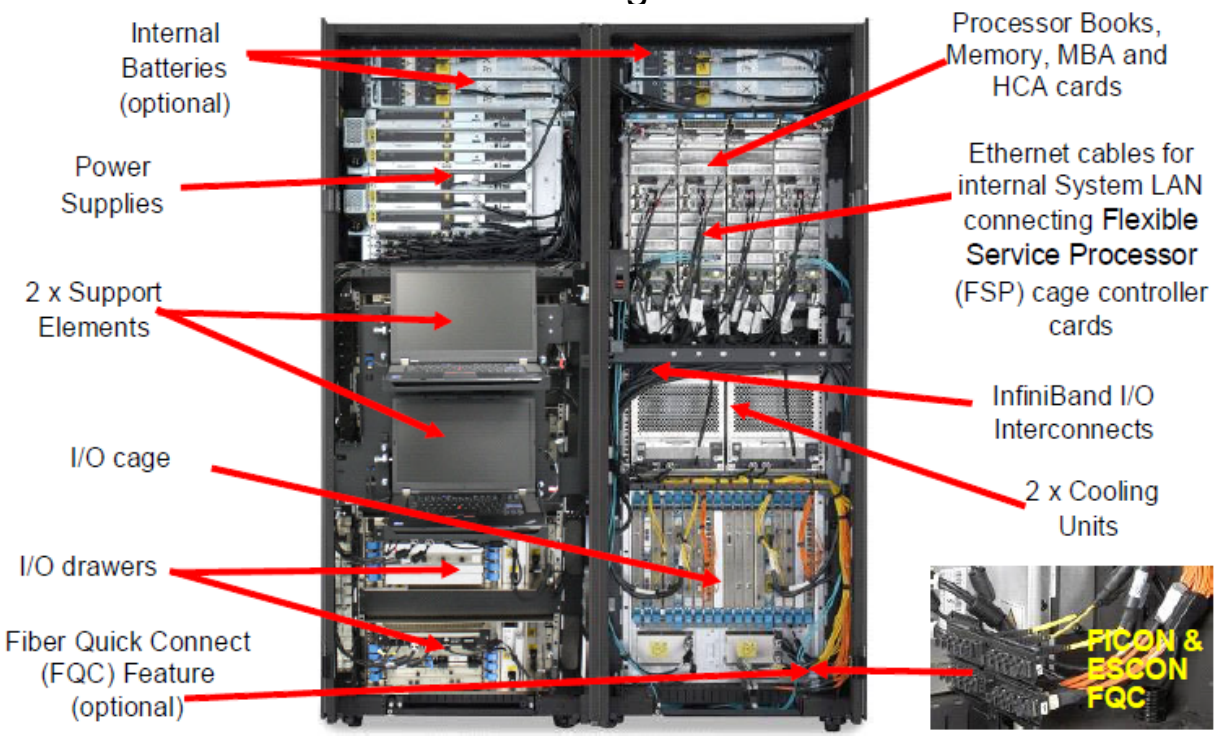
Phone +49 7031 16-1130
buendgen@de.ibm.com

Backup

System z HW Redundancy: Power / Cooling / Infrastructure

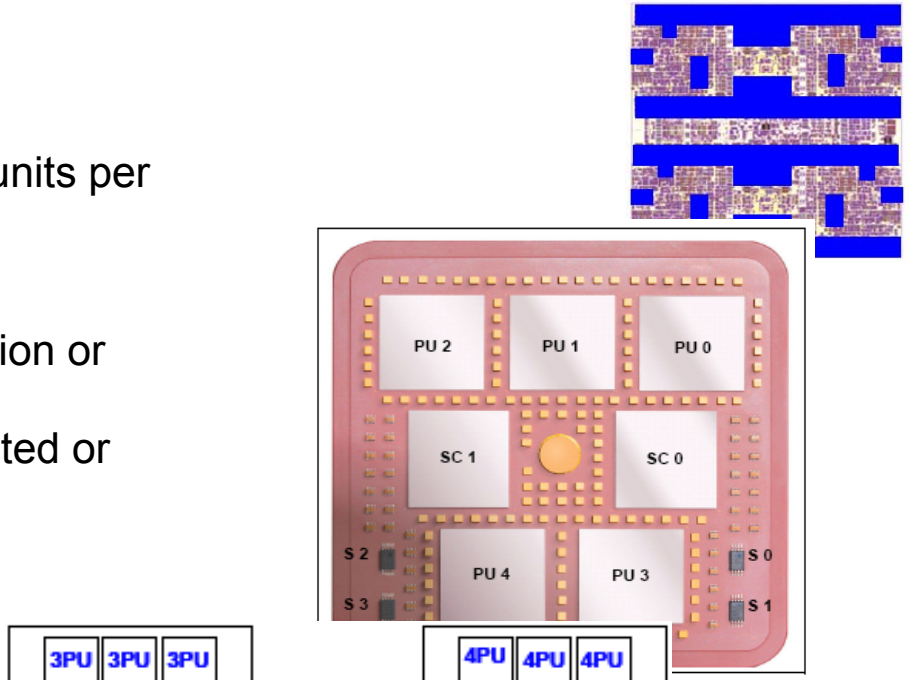
- 2 oscillator cards, 2 SEs, 2 FSPs per book, 2 FSPs per I/O cage
- **Optional internal battery facility (IBF)**
- Hybrid cooling system:

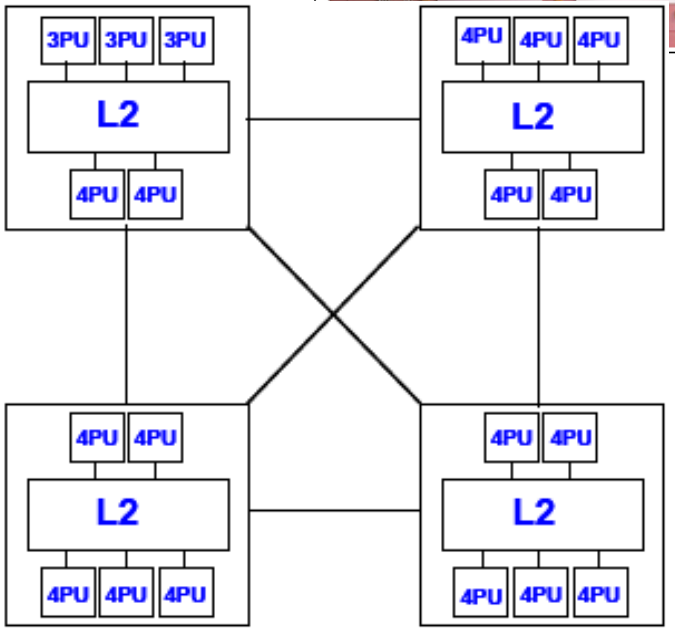
–modular refrigeration unit (MRU)
–Standby redundant air moving devices (AMD)
–cycle steering: frequency and voltage to reduce temperature



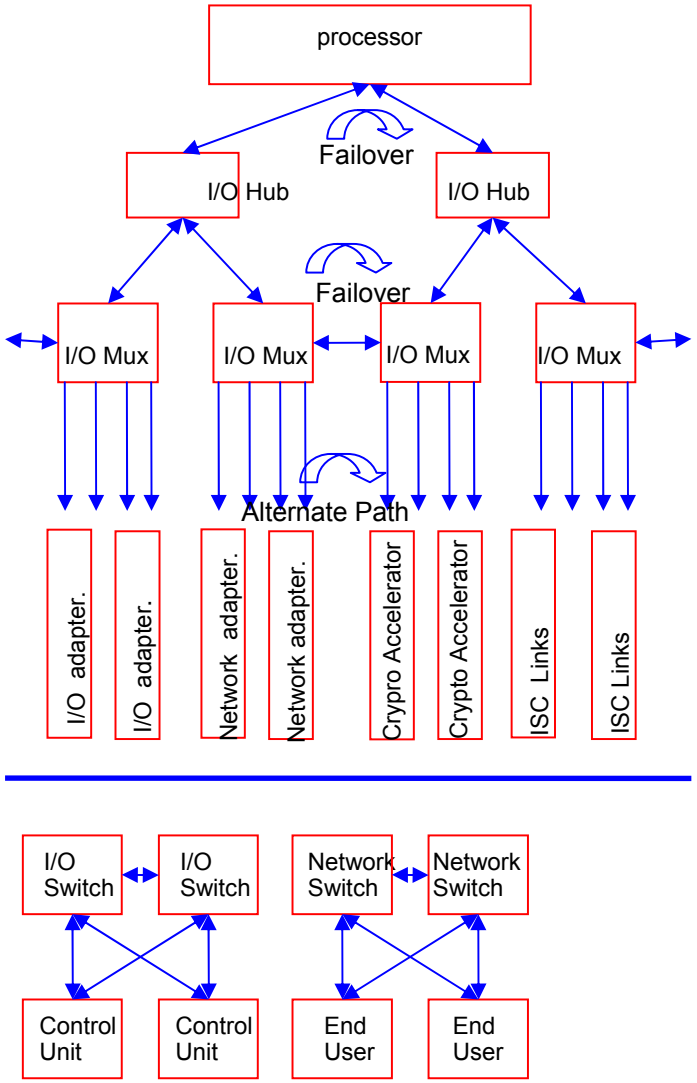
System z HW Redundancy: CPUs

- pre z10:
  - redundant instruction and execution units per CPU
- z10 and later:
  - all execution units have parity prediction or residue-checking
  - all control state machines are duplicated or feature consistency checking
- transparent retry on miscompares
- well proven machine check architecture
- Spare CPUs
  - consistent failure states
  - dynamic sparing (= workload of a CPU will be transparently migrated to spare CPU)
- CPU/Memory interconnects
  - z10: fully connected books
  - z9: dual ring
  - concurrent book repair





System z HW Redundancy: Fully Redundant I/O Subsystem



Fully Redundant I/O Design

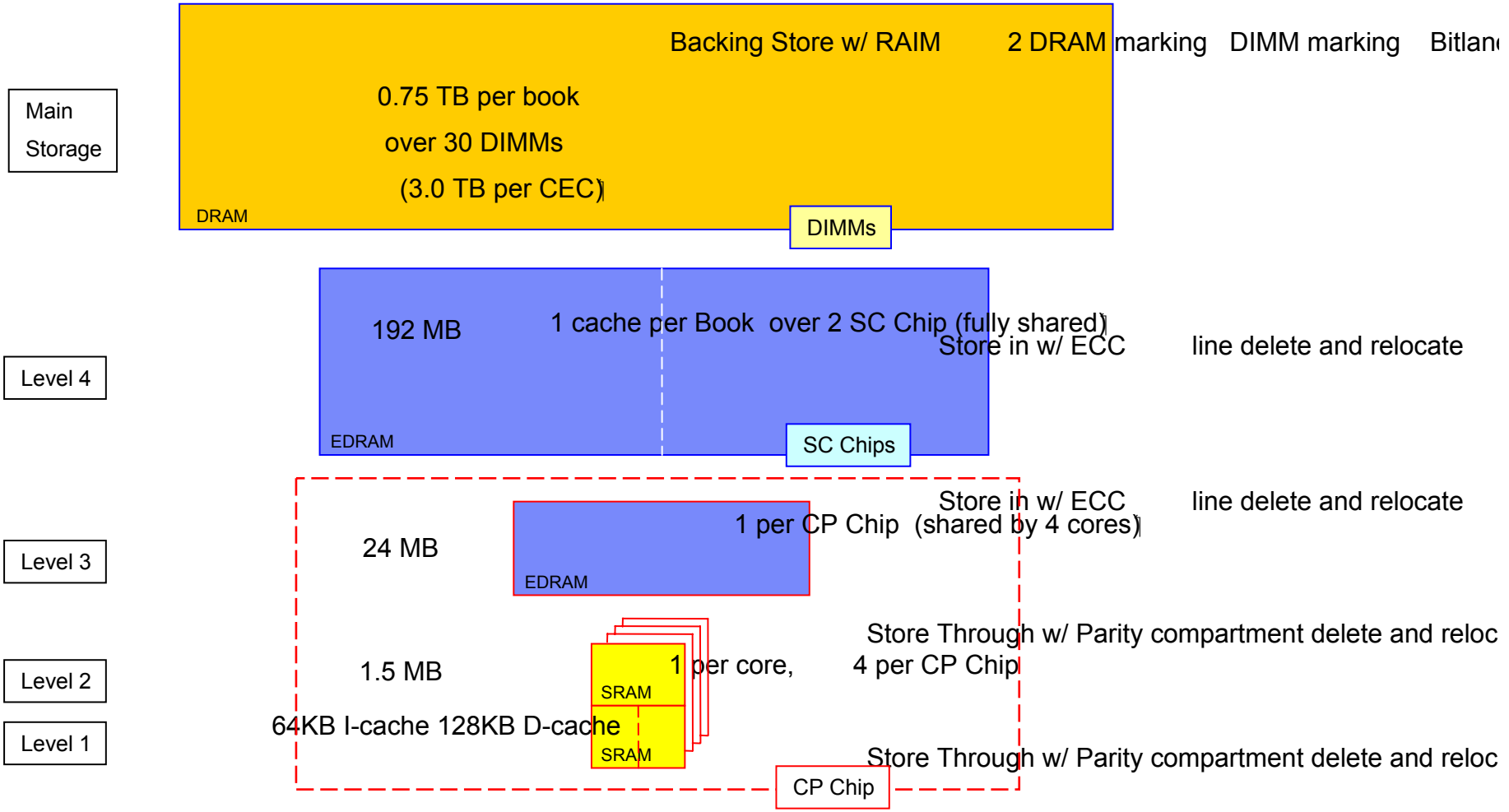
- SAP / CP sparing
- SAP Reassignment
- I/O Reset & Failover
- I/O Mux Reset / Failover
- Redundant I/O Adapter
- Redundant Network Adapters
- Redundant ISC links
- Redundant Crypto Accelerators
- I/O Switched Fabric
- Network Switched/Router Fabric
- High Availability Plugging Rules
- I/O rebalancing on CBA
- Channel Initiated Retry
- High Data Integrity Infrastructure
- Type 2 Recovery
- I/O Alternate Path
- Network Alternate Path
- Virtualization Technology

source: Wiliam Clarke

System z HW Integrity: Error Detection and Recovery

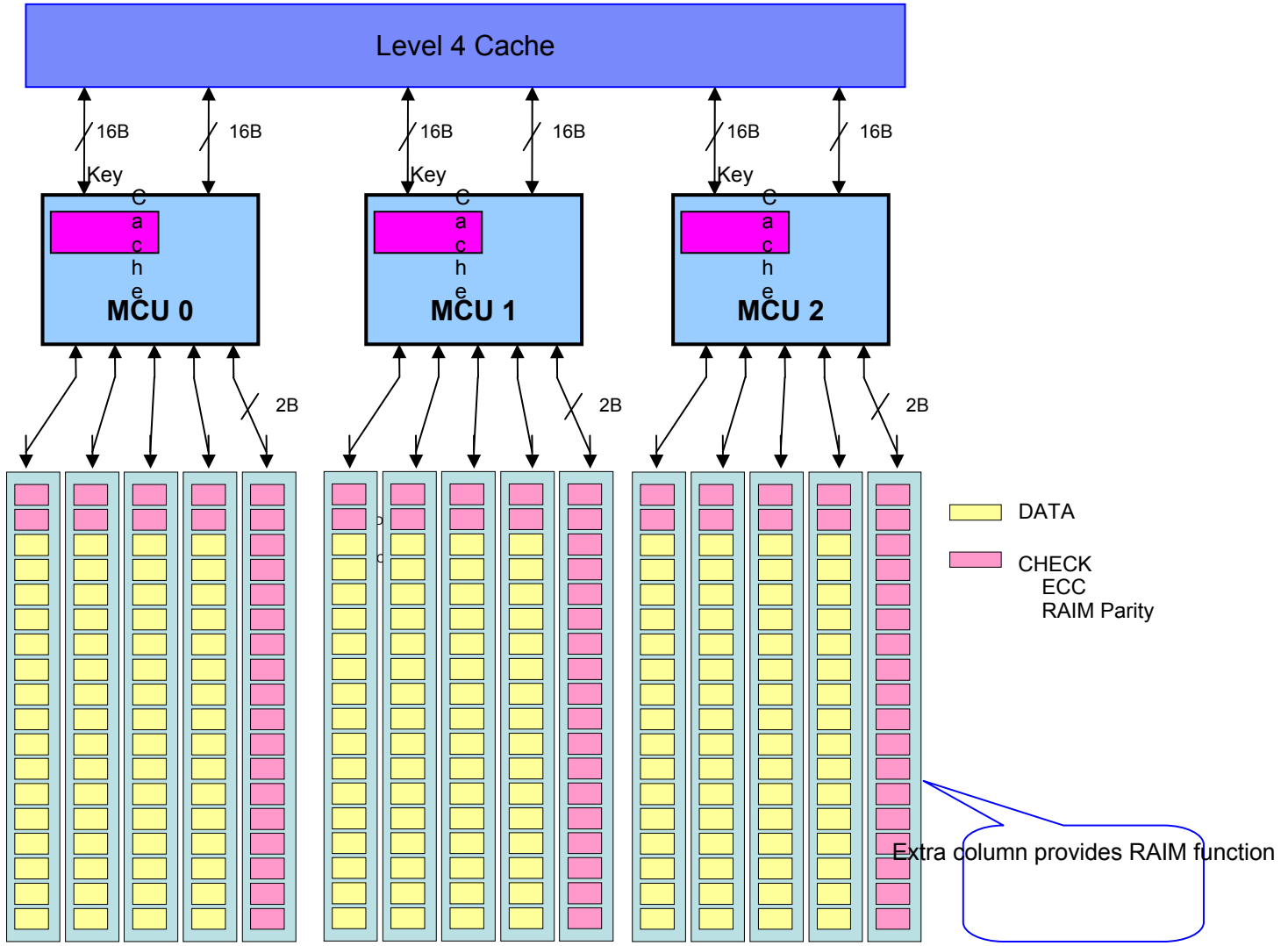
- Permanent error checking & recovery in all components
- Error Recovery
  - Counting recoverable errors & report on exceeding threshold
  - Non-recoverable error: determine & fence failing component
  - Complex logic for determination of failing components able to deal with multiple failures.
- Error Reporting to SE and Call Home
  - Comprehensive field tracking
  - Large Call Home DB used for failure analysis and prevention

System z HW Integrity: End-to-End Memory Integrity



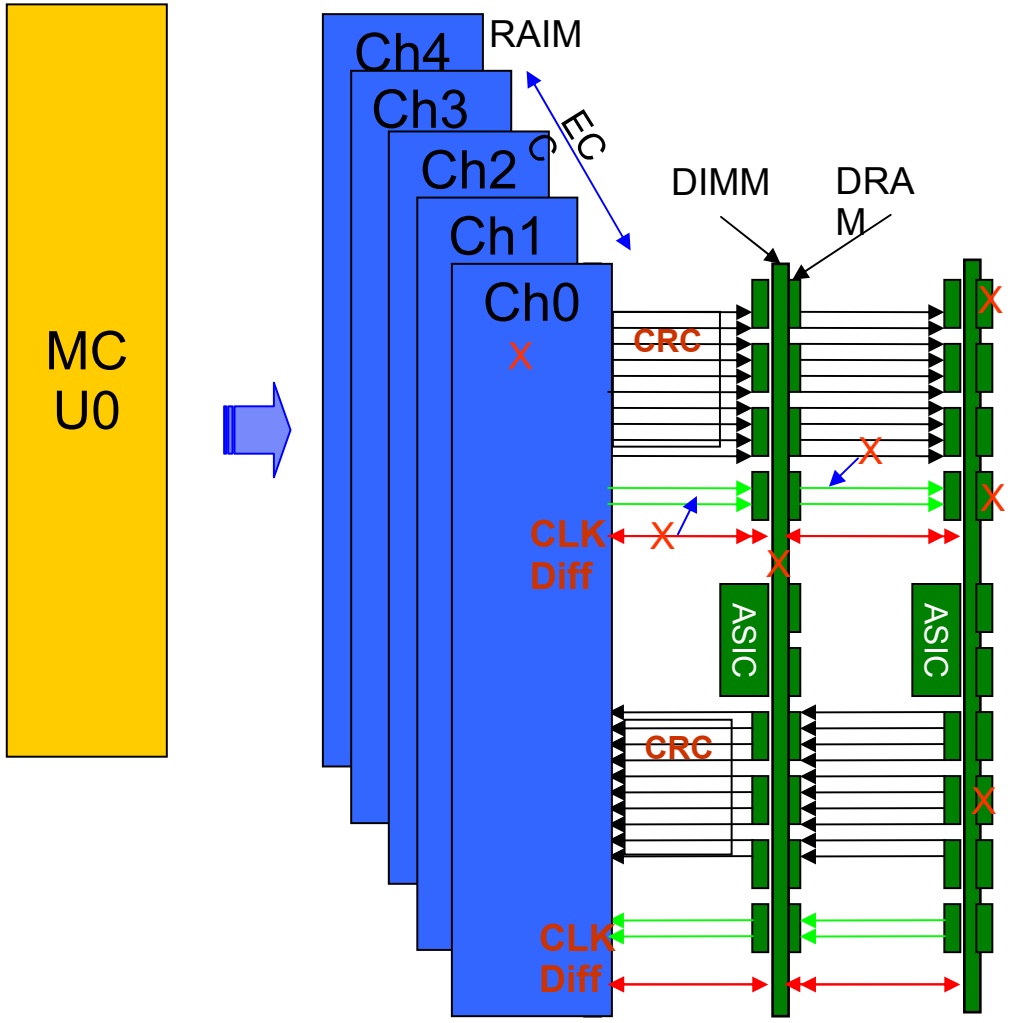
source: Wiliam Clarke

zEnterprise RAIM Technology (Redundant Array of Independent Memory)



source: Wiliam Clarke

zEnterprise – Layers of Memory Recovery



2- Deep Cascade
Using Quad High DIMMs

Layers of Memory Recovery

ECC
- Powerful 90B / 64B Reed Soloman code

DRAM Failure
- Marking technology; no half sparing needed
- 2 DRAM can be marked
- Call for replacement on third DRAM

Lane Failure
- CRC with Retry
- Data – lane sparing
- CLK – RAIM with lane sparing

DIMM Failure (discrete components, VTT Reg.)
- CRC with Retry
- Data – lane sparing
- CLK – RAIM with lane sparing

DIMM Controller ASIC Failure
- RAIM Recovery

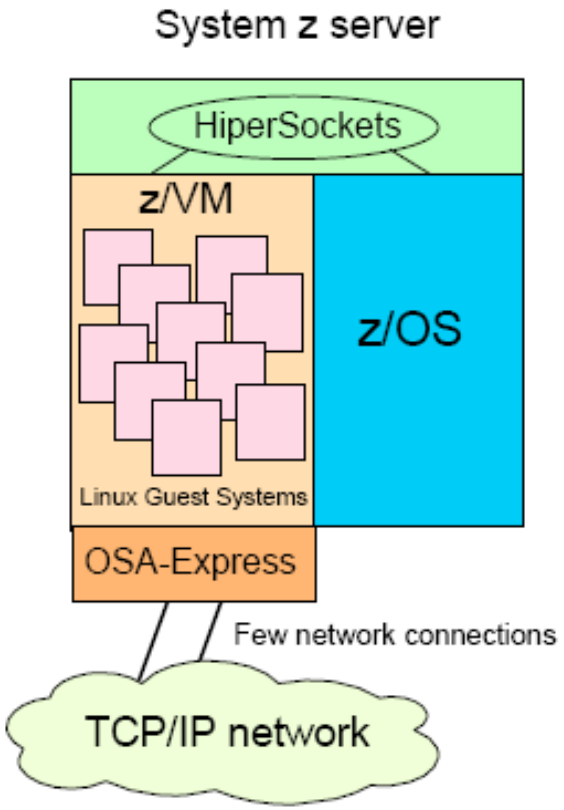
Channel Failure
- RAIM Recovery

source: Wiliam Clarke

System z HW Integrity: Reliable Networking

- HiperSockets
  - network for LPAR to LPAR communication
  - in memory
  - no routing

    - reliable,
    - physically secure



System z HW Integrity: Storage Integrity

- ESCON/FICON/HPF Protocol
- End-to-end data integrity
  - ensures that
    - sent data = received data

      for reads

      for writes
    - from OS device driver to control unit

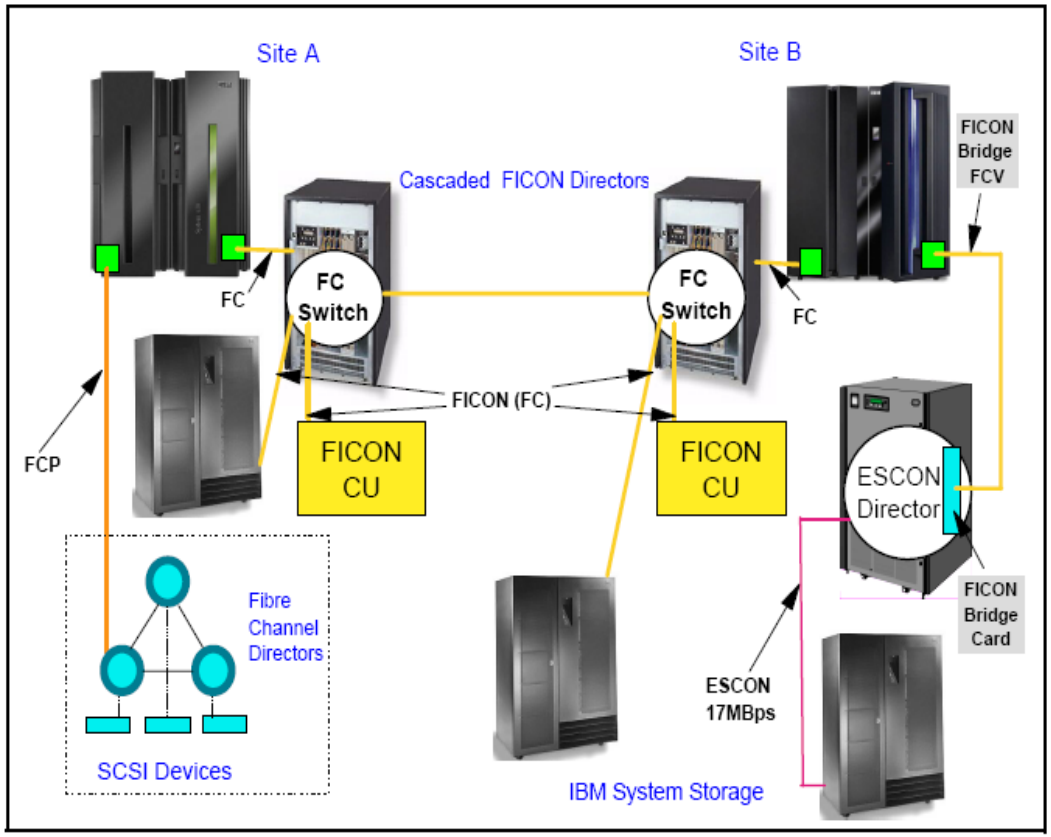


Figure 1-3 FICON connectivity

System z Service: Concurrent Maintenance

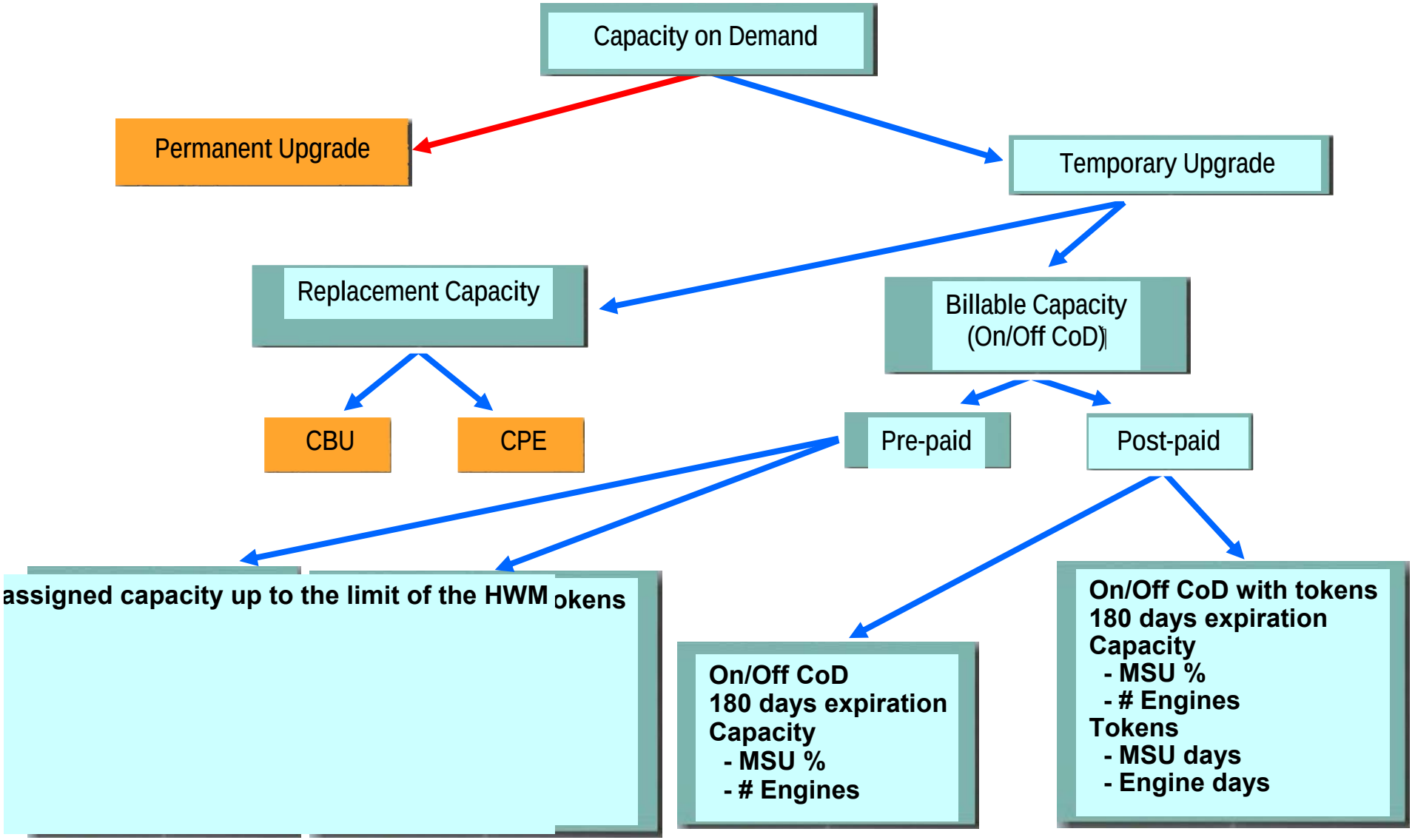
Concurrent maintenance of almost all components:

- Book
- Oscillator,
- Adapters: ESCON/FICON adapters, OSA, Crypto
- Cage Controller, Power supply, Battery, Thermal, Support Element, HMC
- Licensed Internal Code
  - CPU
  - LPAR
  - Channels
  - OSA
  - Power/Thermal
  - Support Element
  - FSP
  - HMC

System z Service: Concurrent Upgrade

- Concurrent upgrade
  - –CPU/Memory/Book
  - –Adapters: ESCON, OSA, Crypto
    - concurrent FW upgrade
    - concurrent install (HW & config)
- Licensed Internal Code
  - –CPU, LPAR, Channels, Power/Thermal, Support Element, FSP, HMC

System z Service: Avoiding Planned Outages with Dynamic Capacity



z/VM RAS Features

- most customers run Linux on System z in a z/VM guest
- in addition to the System z HW, z/VM provides further RAS features

Integrity Inherited from z/VM

- Machine Check Handling
  - Automatic and transparent recovery from
    - machine checks,
    - channel checks, and
    - storage errors.
- Network Availability
  - Error-free, reliable guest-to-guest communication
    - HiperSockets
    - Guest LAN, VSwitch
    - IUCV,
    - virtual CTCA
- Guest isolation

z/VM Availability Features for Linux

- Guest Availability
  - VM time bomb
    - An easy surveillance mechanism to keep your server farm up and running
    - Implemented as Linux watchdog
    - To be used by TSA resource protection

- Network Availability
  - Highly available network access with z/VM VSWITCH
    - Connecting VSWITCH to 2 OSA-E adapters
    - Guests attached to VSWITCH are still reachable if one of the adapters fails

z/VM Serviceability Support

- Service: z/VM & Linux on zSeries Instrumentation and Tools
  - zVM diagnostic facilities
    - TRACE, TRSOURCE, DISPLAY, DUMP, and other advanced guest and host diagnostic facilities.

- Multi-System RAS with z/VM
  - Centralized and consolidated error recording and reporting.
  - Centralized and consolidated performance recording and reporting.