Bob Rogers
IBM Corporation
rrogers@us.ibm.com

50 YEARS OF GIVE AND TAKE

THINK GLOBAL – ACT LOCAL

S83 Getting the Most Out of Your z10 with z/OS

Trademarks

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

AIX*	FlashCopy*	MVS	System z
AnyNet*	GDPS*	Notes*	System z9
CICS*	HiperSockets	OMEGAMON*	System z10
DB2*	Hiperspace	Open Class*	SystemPac*
developerWorks*	HyperSwap	Parallel Sysplex*	Tivoli*
DFSMSdfp	IBM*	PR/SM	VTAM*
DFSMSdss	IBM eServer	Processor Resource/ Systems Manager	WebSphere*
DFSMShsm	IBM logo*	pSeries*	z/Architecture*
DFSMSrmm	IMS	RACF*	z/OS*
DFSORT*	Infoprint*	Redbook	z/VM*
DRDA*	IP PrintWay	RMF	zSeries*
DS8000	iSeries	System i	
ESCON*	Language Environment*	System Storage	
FICON*			

* Registered trademarks of IBM Corporation

The following are trademarks or registered trademarks of other companies.

InfiniBand® is a registered trademark of the InfiniBand Trade Association (IBTA).
Intel is a trademark of the Intel Corporation in the United States and other countries.
Linux is a trademark of Linux Torvalds in the United States, other countries, or both.
Java and all Java-related trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc., in the United States and other countries.
Microsoft, Windows and Windows NT are registered trademarks of Microsoft Corporation.
UNIX is a registered trademark of The Open Group in the United States and other countries.
All other products may be trademarks or registered trademarks of their respective companies.
The Open Group is a registered trademark of The Open Group in the US and other countries.

Notes:

Performance is in Internal Throughput Rate (ITR) ratio based on measurements and projections using standard IBM benchmarks in a controlled environment. 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 processed. Therefore, no assurance can be given that an individual user will achieve throughput improvements equivalent to the performance ratios stated here.

IBM hardware products are manufactured from new parts, or new and serviceable used parts. Regardless, our warranty terms apply.

All customer examples cited or described in this presentation are presented as illustrations of the manner in which some customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics will vary depending on individual customer configurations and conditions.

This publication was produced in the United States. IBM may not offer the products, services or features discussed in this document in other countries, and the 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.

All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.

Information about non-IBM products is obtained from the manufacturers of those products or their published announcements. IBM has not tested those products 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.

Prices subject to change without notice. Contact your IBM representative or Business Partner for the most current pricing in your geography.

This presentation and the claims outlined in it were reviewed for compliance with US law. Adaptations of these claims for use in other geographies must be reviewed by the local country counsel for compliance with local laws.

z10 and z/OS Innovation

- The IBM System z10 make major leaps forward in processor cycle time. This demands specific reactions in z/OS software.
  - Fast cycle time makes accesses to caches and main memory take more cycles
  - The software needs to respond my using processor cache more efficiently
- ***Hiperdispatch*** improves cache efficiency by keeping work running on the same set of processors with the same shared caches.
- The ***data prefetch instruction*** allows software to reduce the latency to access "cold" data which is not in CPU-related cache.
- The ***large page support reduces*** the overhead of dynamic address translation by reducing cache miss latency to access translation data
- Enabling the gathering of hardware performance data on customer systems with ***Hardware Instruction Services (HIS)*** facilitates the improvement of z/OS-based software.

z10 and z/OS Innovation

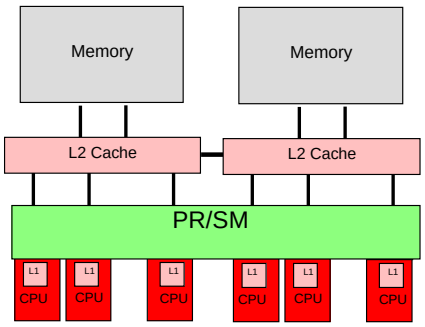
- Higher CPU capacity requires greater I/O bandwidth and efficiency. The z10 introduces a more efficient I/O protocol called ***High Performance Ficon (zHPF).*** The simplified protocol reduces the back and forth chatter between devices and channels.
- The z10 provides and innovative capability to reduce the "mean time to recovery". By allowing the gathering of diagnostic data with stand-alone dump and the re-IPL of the system to be initiated programmatically, the time till the system is again doing productive work is reduced. The capability is called ***Single-System Automatic Stand-alone Dump and Re-IPL.***
- These are not the only value items that z10 and z/OS deliver together, but they are all that fit in this session.

HiperDispatch

HiperDispatch
Hardware and Hypervisor View

- Hypervisor (PR/SM)
  - Virtualization layer at Operating System image level
  - Distributes physical resources
    - •Memory
    - •Channels
      - –EMIF
    - •Processors
      - –Logical processors dispatched on physical processors
      - –Dedicated / Shared
      - –Affinities
      - –Share distribution based on weights

Logical View of 2 Book System

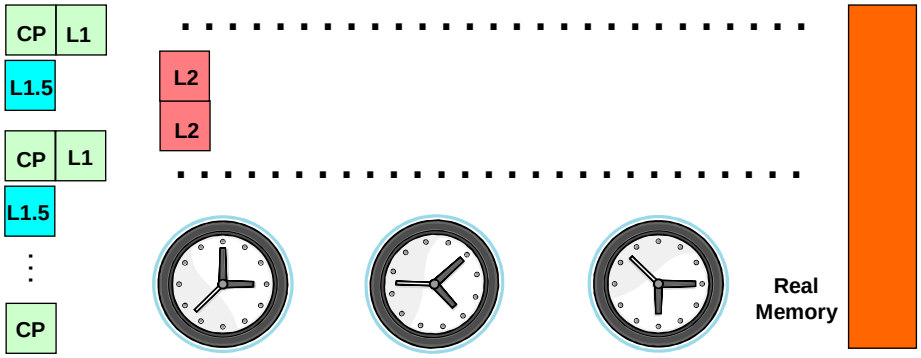


The motivation for HiperDispatch

- In the past, System z hardware, firmware, and software design/development have remained relatively independent of each other
- Modern processor and memory designs make a closer cooperation appropriate. Topology is important:
  - Different CPUs in the complex have different distances to the various sections of memory and cache (here, “distance” is measured in CPU cycles.)
  - Memory access times can vary from less than 10 cycles to several hundred cycles depending upon cache level and whether the access is local or remote.
- Hardware caches are most efficient when each unit of work is consistently redispatched on the same physical CPU (or small set of CPUs)

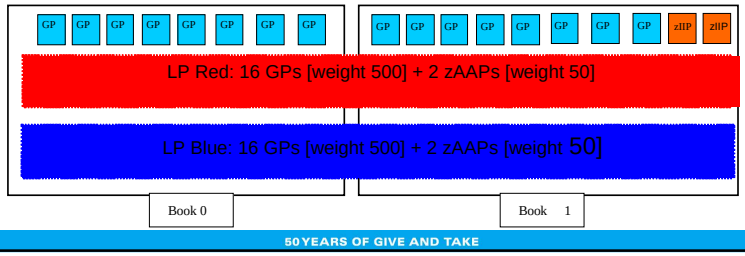
Memory and Cache Latencies

- **Cache and memory latency on a hypothetical modern server**
  - **L1 Cache – 1 machine cycle**
  - **L1.5 Cache – 4 machine cycles**
  - **L2 Cache – variable, 10’s of machine cycles**
  - **Real memory – ~ 600 machine cycles**



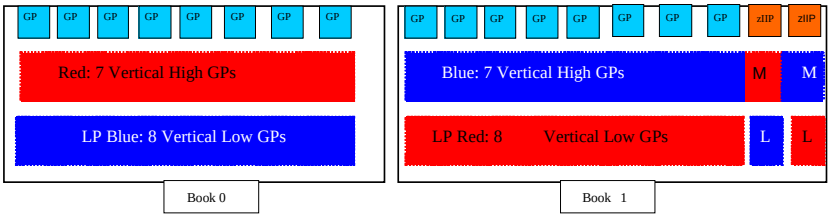
Horizontal CPU management

- PR/SM guarantees an amount of CPU service to a partition based on weights and system capacity
- PR/SM distributes a partition's share evenly across the logical processors
- Any extra service left by other partitions is also distributed evenly across the logical processors.
- Additional logical processors may be required to receive extra service which is left by other partitions.
- The OS must run on all logicals to gather all its share
  - z/OS Alternate Wait Management



Vertical CPU Management

- Logical processors are classified as *vertical high*, *medium* or *low*
- PR/SM quasi-dedicates *vertical high* logicals to physical processors
- The remainder of the share to distributed to the *vertical medium* processors
- *Vertical low* processors are only given service when other partitions do not use their entire share.
- *Vertical low* processors are parked by z/OS when insufficient extra service is available



LP Red: 16 GPs [weight 500] + 2 zAAPs [weight 50]
LP Blue: 16 GPs [weight 500] + 2 zAAPs [weight 50]

HiperDispatch mode

- PR/SM
  - Supplies topology information/updates to the z/OS guest
  - Ties *high priority* logicals to physicals (gives 100% share)
  - Distributes remaining share to *medium priority* logicals
  - Distributes any additional service to unparked *low priority* logicals
- z/OS
  - Associates tasks with a small subsets of logical processors
  - Dispatches work to associated subset of logicals when possible
  - Dispatches work to some other CPU when necessary
  - Parks *low priority* processors that are not needed or will not get service

The combination provides the processor affinity that maximized the efficiency of the hardware caches

Addressing Workload Variability

- **SRM Balancer** spreads workload across Affinity Nodes by priority in an attempt to keep the work evenly distributed
  - Historical address space utilization statistics are collected every 2 seconds in an effort to "predict" future requirements
- **Supervisor** implements "*needs-help*" algorithm to address transient spikes in utilization
  - Maintains priority-based Affinity Node utilization statistics
  - Responsively acts on statistics by asking other LPs for "Help"
- **SRM** tracks PR/SM white space attributes to dynamically address longer term workload requirements
  - Adds / removes Logical Processors to / from existing Affinity Nodes when both the zOS workload warrants it, and the partner LPARs allow it
  - Parks / unparks low priority LPs based on available extra capacity

Special processing for SYSSTC

- **Work classified to SYSSTC typically contains lots of short-running local SRBs required for transaction flow.**
  - Examples of address spaces recommended to be classified into SYSSTC are VTAM, TCP/IP and IRLM.
- **SRBs classified into SYSSTC can execute on <u>any</u> available logical processor even HiperDispatch mode.**
- **WLM service policies should be reviewed with this in mind.**

Controlling HiperDispatch

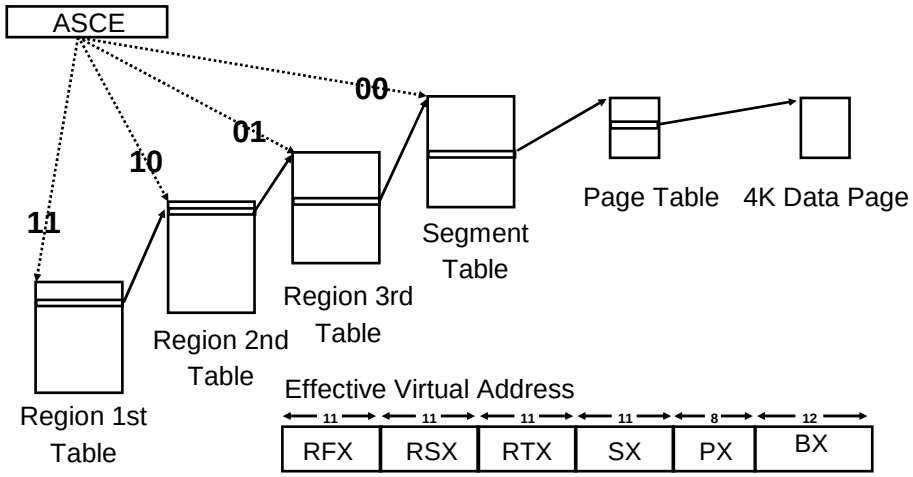
- **HiperDispatch mode is enabled by specifying HIPERDISPATCH=YES in IEAOPTxx**
  - The default is HIPERDISPATCH=NO for compatibility
  - HIPERDISPATCH=YES is recommended
  - There is a HealthChecker routine to remind if HIPERDISPATCH=NO
- **Control authority for global performance data must be enabled for proper operation in HiperDispatch mode**
  - This option is selected in the logical partition security controls on the Hardware Management Console
  - This is the default selection.

Large Page Support

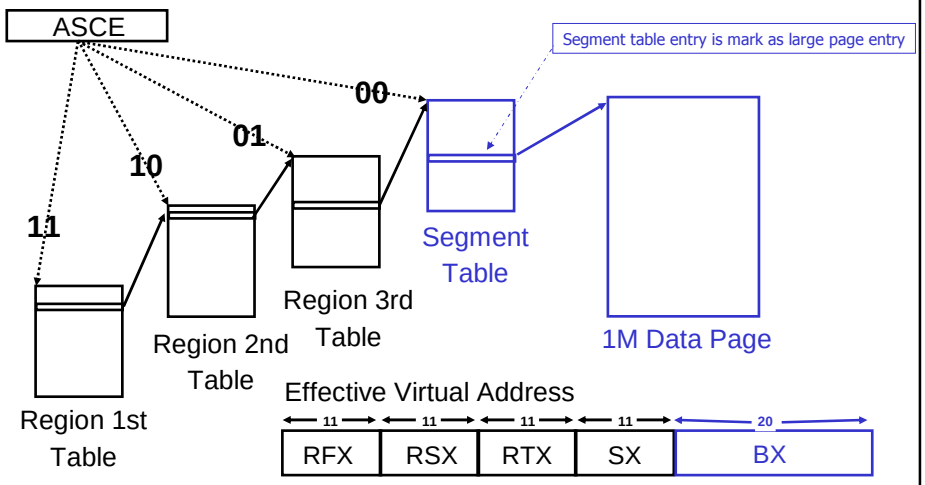
z/OS Large Page Support

- The Large Page size on z10 is 1 Megabyte
- z/OS V1.10 supports both 4K and 1MB page sizes
- The Real Storage Manager (RSM) maintains an area from which large pages are allocated
- If the system is constrained for 4K pages
  - Available 1MB pages are “broken up” and used to satisfy 4K page requests
  - Pages can later be coalesced into 1MB pages
- Large Pages are NOT be pageable
- The Large Pages area is NOT reconfigurable

Address Translation for 4K Page



Address Translation for 1M Page



z/OS Large Page Support

- The installation specifies the maximum amount of real storage to be used for Large Pages in IEASYSxx
  - LFAREA=(xx%| xxxxxxM | xxxxxxG| xxxxxxT)
- Authorization to request backing with large pages
  - Authorized callers are allowed to request the use of large pages
    - Supervisor state or key 0-7 or APF authorized
  - Unauthorized programs must run under a USERID that has Read access to facility class resource IARRSM.LRGPAGES
- Exploiters
  - Java 6 SR1
    - requests large pages to back the Java Heap
  - DB2 Version X
    - intends to exploit large pages for buffer pools

Data Prefetch Capabilities

Extract Cache Attributes (ECAG)

ECAG	$R_1,R_3,D_2(B_2)$					[RSY]
EB	R_1	R_3	B_2	DL_2	DH_2	4C

R_1 — Result placed in R_1.0-63

Selected bits of 2nd-operand address form a code:

Bits 56-59: Attribute Indication:

- 0 – Extract topology summary (for up to 8 levels)
- 1 – Extract line size of cache in bytes.
- 2 – Extract total cache size in bytes.
- 3 – Extract set-associativity level.

4:15 – Reserved

Bits 60-62: Level indication of cache.

Bit 63: Type indication (0=data, 1=instruction)

Topology Summary in R_1 (one byte per cache level):

Bit 0-3: Reserved, stored as zeros
Bits 4-5: Cache scope, as follows:
- 00 – Cache level does not exist
- 01 – Cache is private to the CPU
- 10 – Cache may be shared
- 11 – Reserved

Bits 6-7: Cache type, as follows:
- 00 – Separate instr. & data caches
- 01 – Only instruction cache
- 10 – Only data cache
- 11 – Unified instruction & data cache

Condition Code is Unchanged

PREFETCH DATA (PFD)

PFD	$M_1,D_2(X_2,B_2)$					[RXY]
E3	M_1	X_2	B_2	DL_2	DH_2	36

Code:

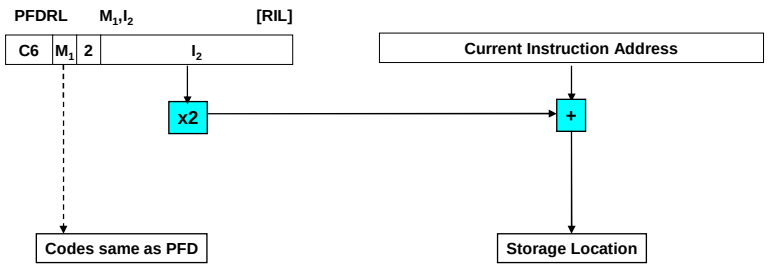
1 - Prefetch for fetch
2 - Prefetch for store
6 - Release cache line from store, retain for fetch.
7 - Release cache line

All other codes reserved; reserved codes act as no-op

Storage Location

Condition Code is Unchanged

Prefetch Data Relative Long



Condition Code is Unchanged

Examples of Prefetch Data usage

- **A number of z/OS elements have taken advantage of the data prefetch capability of z10 to improve CPU performance**

JVM	J9ZeroMemory Routine clearing memory in Java 6 SR 2
VLF	Prefetches the MVC loop to move an object.
CommServer	Prefetches around the move and copy from user storage.
DFSORT	Prefetches around the copy to a contiguous buffer

Hardware Instrumentation Services

Motivation

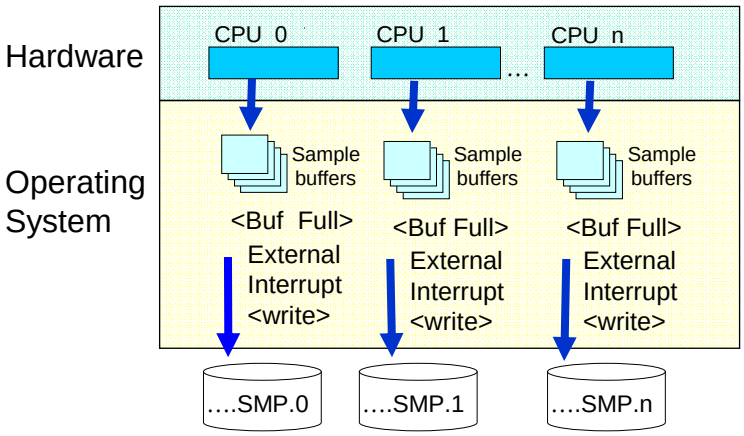
- Provides better field diagnosis tools for z/OS
  - Hardware perspective
    - How the customer/vendor application interacts with the hardware
  - Software perspective
    - Where CPU time is being spent and why
- Makes additional performance information available to z/OS vendors and customers
  - Provides raw materials for more complete analysis
  - Facilitates more efficient applications on z/OS

High level description

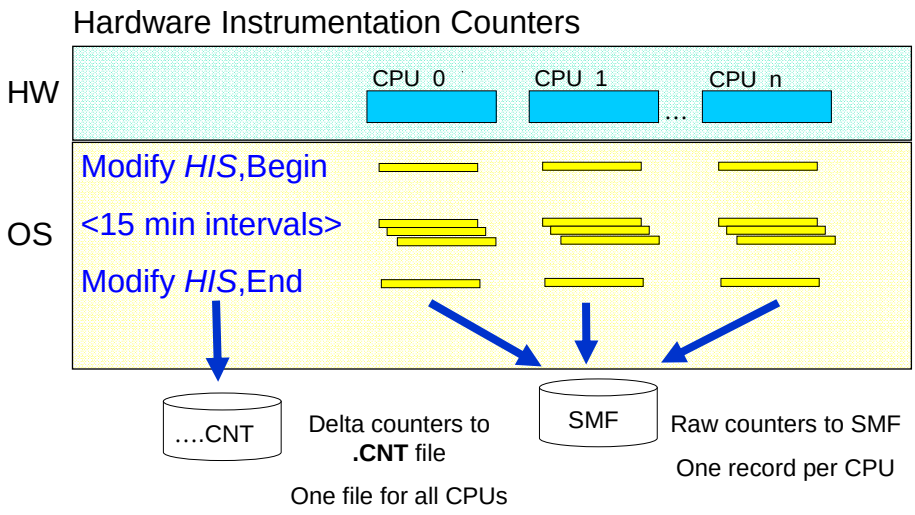
- Data collection done by System z hardware
  - Low overhead
  - Little/No skew in sampling
  - Access to information which is not available from software
    - Information about how software and hardware interact
- 2 Basic Modes
  - COUNTERS
  - SAMPLING
    - *SAMPFREQ=800000* is the default (samples per minute), = 13,333 / sec
      8M samples in 10 minutes is the default (*DURATION=10* is the default, 10 minutes)
    - **Recommendation – Start with a small frequency**, e.g. *SAMPFREQ=320*, and increase after early experiences – e.g. ensure enough disk space for output
      Smaller z10 BCs should increase only up to *SAMPFREQ=130000* (for DURATION=60)
- IBM Research article
  - ***"IBM System z10 performance improvements with software and hardware synergy"***
  - http://www.research.ibm.com/journal/rd/531/jackson.pdf

How the facility works

Hardware Instrumentation Sampling



How the facility works



Operator Control for HIS

- **START HISPROC** to start Hardware Instrumentation Services
- **MODIFY HISPROC** to start and stop data collection
  - Specify counters/samples, # buffers, path to output directory, sampling freq., etc.
  - A map is optionally collected at stop.
    - Can map entire system, or selected Jobnames/ASIDs.
  - Results are written to a directory in the HFS
  - Raw counter data are written to SMF
- **DISPLAY HIS** to display status
- **STOP HISPROC** to stop HIS

System Requirements

- Hardware Instrumentation is available for customer and vendor use on System z10 processors with the latest service.
  - Restricted on sub-capacity z10 BC machines
- z/OS support is provided in the base in R11
- z/OS support has been provided as service for prior releases:
  - z/OS V1R8 (HBB7730) - UA43988
  - z/OS V1R9 (HBB7740) - UA43989
  - z/OS V1R10 (HBB7750) - UA43990

Counter Data

- Basic Counter Set
  - Cycle count
  - Instruction count
  - Level-1 I-cache directory write count
  - Level-1 I-cache penalty cycle count
  - Level-1 D-cache directory write count
  - Level-1 D-cache penalty cycle count
- Problem State Counter Set
  - Problem state cycle count
  - Problem state instruction count
  - Problem state level-1 I-cache directory write count
  - Problem state level-1 I-cache penalty cycle count
  - Problem state level-1 D-cache directory write count
  - Problem state level-1 D-cache penalty cycle count
- Extended Counter Set
  - Number and meaning of counters are model dependant

SMF Record type 113, subtype 2

Layout: (SMF manual, HISYSMFR macro)

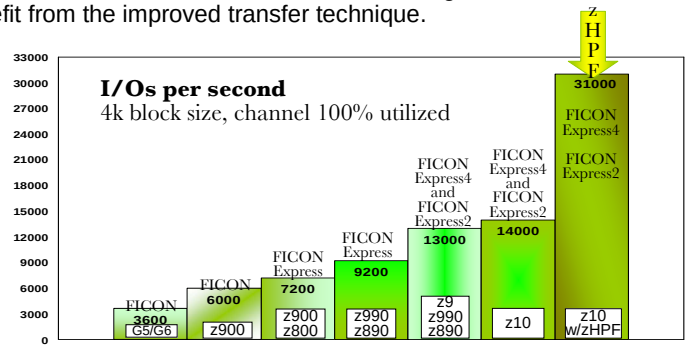
- Standard SMF record header (‘1C’x bytes)
- SMF record control information
  - – TOD when SMF record is written, etc.
  - –Offset, length, and number of data sections
- Data section
  - –TOD when counter data was captured
  - – CPU number
  - –Offset, length, and number of Counter Set Sections
  - –Offset, length, and number of Counter Sections
  - –Counter Set Sections
    - •Counter Set type (1=BASIC, 2=PROB, 3=CRYPTO, 4=EXT)
    - •Bit mask identifying the counters being recorded in array
      - • e.g. ‘FC00000000000000’x => counters 0-5 are valid
  - – Counter Sections – 8-byte counter values (contiguous)

zHPF - High Performance FICON

What is High Performance FICON?

High Performance FICON (zHPF) enhances the z/Architecture and the FICON interface architecture in order to provide optimizations for OLTP workloads..

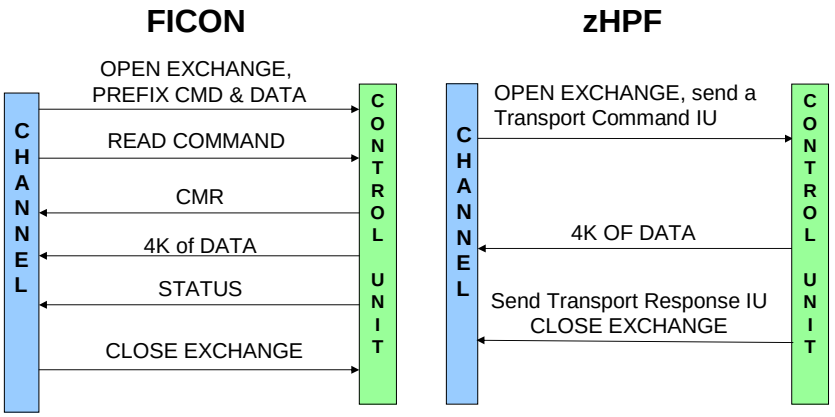
zHPF is a new data transfer protocol that is optionally employed for accessing data from an IBM DS8000 storage subsystem. Data accessed by DB2, IMS Fast Path, PDSE, VSAM, zFS, HFS, CVAF, Catalog, Extended Format SAM can benefit from the improved transfer technique.



Transport Control Words (TCWs)

- **Transport Control Words (TCWs) are a replacement for Channel Command Words (CCWs)**
- **Multiple commands can be packaged in a single TCW**
- **The channel does not keep track of each command while the control unit processes them**

Link Protocol Comparison (4KB READ)



zHPF provides a much simpler link protocol than FICON

zHPF eligibility (current implementation)

- **Reads and update writes that don't cross track boundaries and don't cross Locate Record domains**
- **I/Os that are eligible**
  - **DB2 Synch I/Os, some DB2 log I/Os and some deferred writes**
  - **ISPF Browse/Edit**
- **I/Os that are not eligible**
  - **Format writes, DB2 prefetch**
  - **BSAM I/Os used by high level languages and utilities**

Media Manager

- **Media Manager uses zHPF protocol when appropriate**
  - **if zHPF is activated in the z/OS system**
  - **if the hardware supports z/HPF for the type of I/O being done**
  - **construct a zHPF channel program using TCWs**
- **If not eligible, Media Manager constructs a FICON channel program using CCWs**

zHPF Options and Commands

- New System Parameter
  - **ZHPF=YES|NO** parameter in the **IECIOSxx** parmlib member
  - The default is **ZHPF=NO**
- Enhanced Commands
  - **SETIOS ZHPF=YES|NO** Turn zHPF function on and off
  - **D IOS,ZHPF** Displays zHPF status for the MVS image
  - **D M=DEV** Displays zHPF support status by device
  - **D M=CHP** Displays whether zHPF is supported by the channel subsystem

Observing zHPF activity

- **New fields in RMF Channel activity report**
  - **RATE: The number of native FICON or zHPF operations per second at the physical channel level.**
  - **ACTIVE: The average number of native FICON or zHPF operations that are concurrently active. Often referred to as the number of "open exchanges".**
  - **DEFER: Number of deferred native FICON or zHPF operations per second. This is the number of operations that could not be initiated by the channel due to lack of available resources.**
- **GTF and IPCS**
  - Use the CCW option to trace both FICON and zHPF channel programs
  - IPCS will format both types of channel programs

zHPF – System Requirements

- Hardware
  - IBM System z10 Server (Driver 76D)
  - FICON Express2 or Express4 features
  - DS8000 version 4.1 or Higher
    - Code bundle R12q.9b081017b (Bundle 64.1.16.0) for the R4.1 GA
    - High Performance FICON indicator (feature code 0709 - one time charge)
- Software
  - z/OS V1.8, V1.9, or V1.10 with PTFs.
  - z/OS V1.7, with the IBM Lifecycle Extension for z/OS V1.7 (5637-A01) with PTFs.
  - RMF – Requires APAR OA21140 for new function support for High Performance FICON (zHPF)

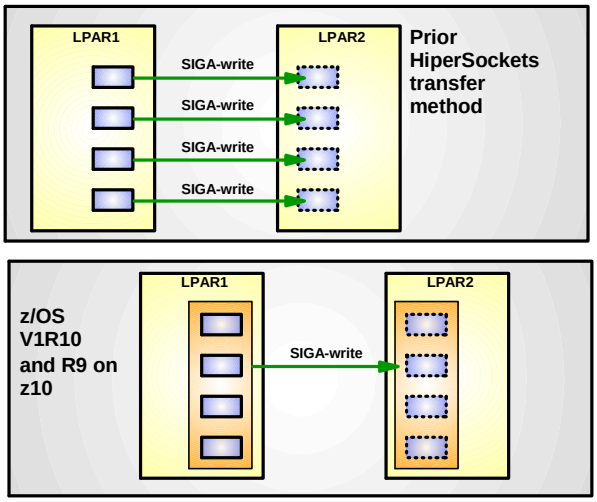
Hipersockets Multiple Write

HiperSockets Multiple Write

- The IBM System z10 provides an enhancement for HiperSockets designed to improve the efficiency of the processing for internal LPAR-to-LPAR communications associated with large messages.
- The new HiperSockets Multiple Write Facility allows messages which span multiple output buffers to be transferred with a single write operation from the source LPAR.
- This enhancement will improve the HiperSockets throughput while also lowering the CPU cost related to the processing of transferring large messages from the source to the target LPAR.
- Also available on z/OS V1.9 with a PTF

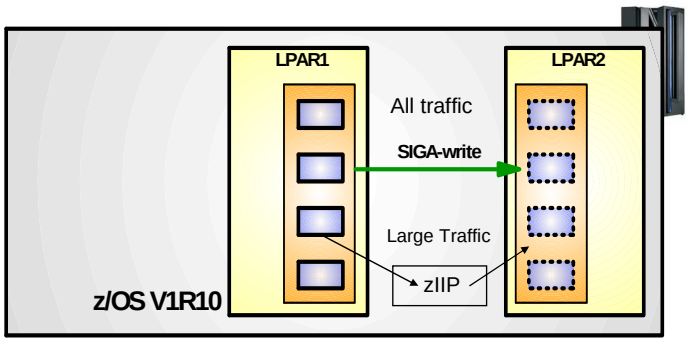
Improved Performance with HiperSockets Write Multiple

- Write multiple improves the efficiency of HiperSockets for large messages
- Support in z/OS R10 (or PTF on R9)



zIIP Assisted HiperSockets

- Using zIIPs lowers the cost of HiperSockets for large messages



Single System AutoIPL

Single-System Automatic SADump and re-IPL

- Auto-IPL quickly and automatically initiates either a Standalone Dump (SADMP) or re-IPL of z/OS, or both when a z/OS system enters a disabled wait state.
  - **AutoIPL** Policy is specified via DIAGxx parmlib member
  - A hard-coded **Wait State Action Table** designates disabled wait state codes/reason codes for which **AutoIPL** processing should or should not be performed, and which actions to be taken
    - Most non-restartable disabled wait states/reasons will be eligible for **AutoIPL** processing
  - **VARY XCF** commands support new options to IPL Stand Alone Dump, re-IPL z/OS or both after the target image has been partitioned out of the sysplex

AutoIPL and SFM

- The PTF for APARs OA26993/OA26995, along with underlying LPAR firmware support, now enables AutoIPL to be used in configurations where an SFM policy is active.
  - With this support, requested AutoIPL actions will be performed in accordance with the DIAGxx parmlib member, even when an SFM policy is active in the sysplex.
- Use of SFM and system isolation to quickly and automatically remove a failed system from the sysplex remains a highly-recommended "best practice" for sysplex availability.
- In multisystem sysplexes the prompt removal of failed systems from the sysplex to permit cleanup of shared resources is of primary importance, enabling the remaining systems in the sysplex to continue normal operation.