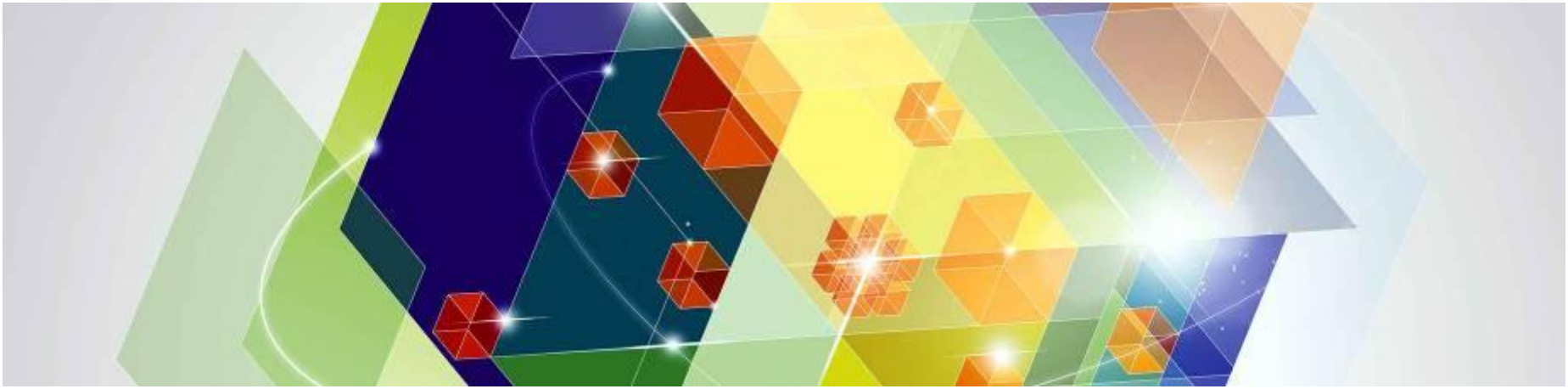
z/OS Large Memory: Size Does Matter

SHARE in Anaheim
March 13th, 2014
Session 15142

Elpida Tzortzatos:
elpida@us.ibm.com



© 2013 IBM Corporation

Agenda

- *Large Memory Client Value*
- *Large Pages 1MB & 2GB*
- *Large Page Performance Results*

© 2013 IBM Corporation

Large Memory Client Value

- ***An increasing number of applications today rely on in-memory caches, heaps, indexes, bufferpools and hashtables for scaling for the performance that is required by large workloads and big data.***
  - **System z Large Memory (up to 1TB per z/OS partition) and N-way Scaling allow applications on z to transparently extend their memory footprints up to terabytes in order to achieve high in-memory hit rates that are vital to high performance.**
- ***z Memory is managed efficiently across all the layers of the System z Stack to provide for reduced data movement, better memory management for JAVA heaps and database bufferpools.***
  - **Studies have shown the performance suffers with the amount of data transferred. z/OS provides APIs and services to middleware and system components that minimize data transfers reducing response latencies and improving application execution time.**

© 2013 IBM Corporation

Large Memory Client Value ...

- ***Superb Large Memory and N-Way scaling on z enable the wide-scale deployment that allows system z to host Business Analytics, IT Analytics, Cloud, and Big Data applications as well as serve as the back end for Mobile applications.***
- ***Continued exploitation of system z Large Page Support in system components, middleware, and customer applications, provides* improved transactional response times and greater throughput. This can translate to better performance and overall CPU savings for DB2, JAVA, and analytic workloads.**

© 2013 IBM Corporation

Large Memory Exploitation

- ***GOALS***
  - **Substantial Latency Reduction for interactive workloads**
    - Google, Amazon studies show substantial business value for faster response times, even sub-second
  - **Batch Window Reduction**
    - More Concurrent Workload
    - Shorter Elapsed times for Jobs
  - **Measurable 'tech-dividend' CPU benefit**
    - Ability to run more work at the same HW and SW MSU rating -or-
    - Ability to run the same workload with lower HW and SW MSU rating

Rollout

- Dump Transfer Tool for large dumps
- SVC and SAD Dump performance
- 1MB pages
- 2GB pages
- IO adapters with >1 TB memory addressability
- Flash Memory
- DB2, Java & IMS Large Memory Scale
- z/OS Large Memory Scale
- CFCC Large Memory Scale

© 2013 IBM Corporation

DB2 Large Memory Benefits

- ***Conversion to DB2 Page Fix Buffers***
  - **Exploit 1MB and 2GB pages**
  - **Paging Spike / Storm Concerns mitigation with Flash and Large Memory**
  - **Pagefix CPU 'tech-dividend' 0-6%**
  - **Large Page CPU 'tech-dividend' 1-4%**
- ***Increase Buffer Pool size in z/OS***
  - **Response times up to 2X**
  - **CPU 'tech-dividend' up to 5%**
- ***Increased Global Buffer Pool Size in CF***
  - **4K page hit in CF is 10X faster transfer than a 4K page hit in disk control unit cache.**

- November 2011 Information on Demand white paper, titled “Save CPU Using Memory,” Akiko Hoshikawa showed that the IBM Relational Warehouse Workload (IRWW) had a 40-percent response-time reduction while gaining a 5-percent CPU performance improvement by exploiting increased buffer-pool sizes.

© 2013 IBM Corporation

Large Memory Study Preliminary Results

- ***A customer representative financial services workload***
  - **Memory intensive, large number of tables, and random I/O behavior**
  - **An example of a financial services DB2 workload with variable memory sizes**
    - Test Scenarios:
      - DB2 11 Single System
        - ➢ 256 GB / 512 GB / 1024 GB real storage
      - DB2 11 Single System – minimal #of BPs
        - ➢ Start with 256 GB real storage

© 2013 IBM Corporation

Test Environment – Single System

System z Database Server
IBM zEC12 2827-HA1

- **12 CPs**
- **Up to 1024 GB of real storage**
- **DB2 11 for z/OS**
- **z/OS 1.13**
- **Up to 675 GB LFAREA**
- **1M large frames**
- **PAGESCM=NONE**

IBM System Storage
- Dual Frame DS8870 w/ 8 GB adapters and LMC 6.2 for 60M Account Banking Database
- DS8700 for DB2 logs

FICON Express8S

10 GbE Network

Application Servers
- 24 IBM PS701 8406-71Y Blade Servers
- Each with 8 3.0 GHz processors and 128 GB memory
- AIX 7.1.0
- DB2 Connect 10.1 FP2

Presentation Server
- IBM 9133-55A with 4 2.1 GHz processors and 32 GB memory

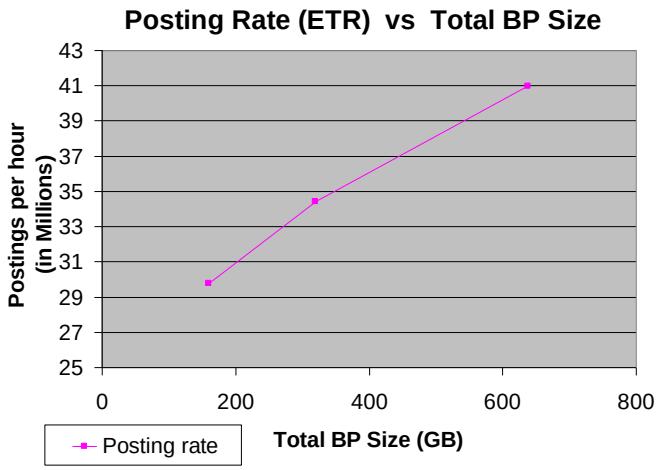
© 2013 IBM Corporation

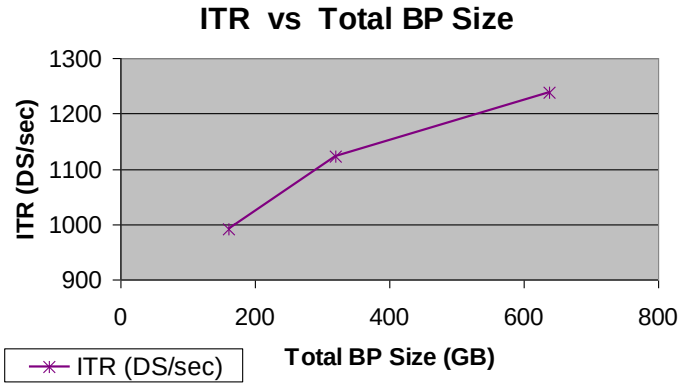
Results: Single System

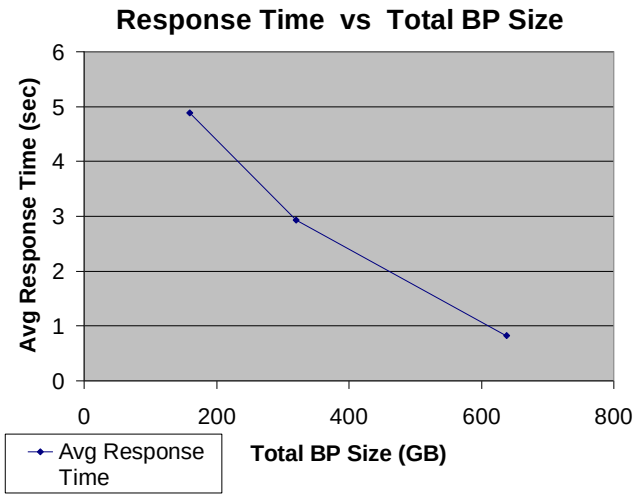
- *With zEC12-12w; CPU 70 – 80%*
- *256 GB -> 512 GB real storage*
  - 13% improvement in ITR
  - 16% improvement in ETR
  - 40% improvement in response time
- *256 GB -> 1024 GB real storage*
  - 25% improvement in ITR
  - 38% improvement in ETR
  - 83% improvement in response time
- *Largest bufferpool: 144 GB*

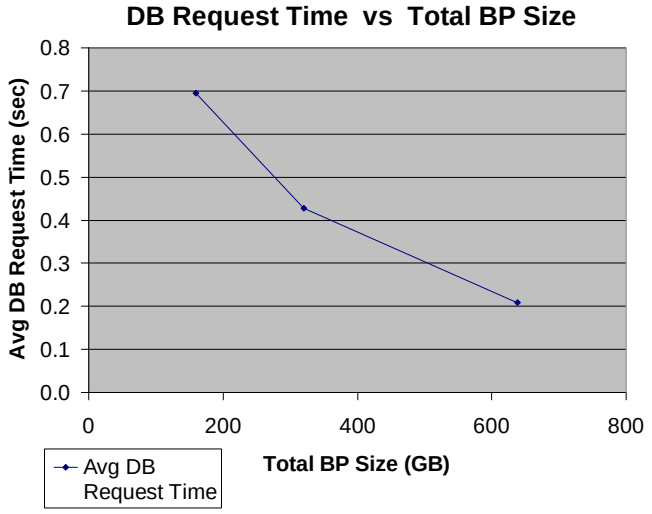
© 2013 IBM Corporation

Execution time plots illustrating the performance improvements in throughput and transaction response times for various BP sizes









© 2013 IBM Corporation

Large Memory Exploitation Roadmap

- ***Large Memory is <u>architecturally</u> transparent***
  - **No application changes needed for functional correctness**
  - **Scalability and best performance achieved by taking advantage of new memory technologies such as large pages**
    - Minimal code changes, limited to the memory allocation part of the application code
    - Optionally provide OS with hints on memory reference patterns
- ***Memory Management coordinated across hardware, hypervisors, software***
  - **Memory Affinity**
  - **Cross-Stack Cooperative Model of Memory Management**
  - **Provide APIs for applications to provide information on memory usage patterns**
- ***Large Page Support (1MB, 2GB)***
  - **Exploitation enhancements in plan across z/OS stack, aligned with large memory rollout**
- ***Large Memory Instrumentation and Tooling***
  - **New SMF data, monitoring statistics, etc**
  - **Tooling to help determine the memory capacity that is optimal for the target environment**

© 2013 IBM Corporation

Large Page Support

© 2013 IBM Corporation

1MB Pageable Large Pages

© 2013 IBM Corporation

Why Implement Large Pages

- ***Problem: Translation Lookaside Buffer (TLB) Coverage shrinking as % of memory size***
  - Over the past few years application memory sizes have dramatically increased due to support for 64-bit addressing in both physical and virtual memory
  - TLB sizes have remained relatively small due to low access time requirements and hardware space limitations
  - Therefore TLB coverage today represents a much smaller fraction of an applications working set size leading to a larger number of TLB misses
  - Applications can suffer a significant performance penalty resulting from an increased number of TLB misses as well as the increased cost of each TLB miss
- ***Solution: Increase TLB coverage without proportionally enlarging the TLB size by using large pages***
  - Large Pages allow for a single TLB entry to fulfill many more address translations
  - Large Pages will provide exploiters **with better TLB coverage**
- ***Benefit:***
  - Better performance by decreasing the number of TLB misses that an application incurs

© 2013 IBM Corporation

Overview

- ***Problem Statement / Need Addressed:***
  - **Performance degradation due to increased TLB (Translation Lookaside Buffer) misses was addressed with the introduction of large pages back in z/OS V1R9 (See Appendix for z/OS V1R9 Overview)**
  - **One limitation with these large pages is that they are implicitly fixed (which means they cannot be stolen/paged out if a more recent or more active use of large pages occurs)**
    - And if even we did attempt to page them out, there is currently no way to retain their 'large page' attribute once on AUX

© 2013 IBM Corporation

Overview

- ***Solution:***
  - **With the introduction of FLASH Support there is now a way to page out large pages and retain their 'large page' attribute**
  - **This item, will introduce large pages that are pageable**
    - **64-bit Pageable Large Pages**
      - IARV64 GETSTOR and GETCOMMON Support
    - **31-bit Pageable Large Pages**
      - STORAGE Support (31-Bit virtual user region subpools, low private subpools 0-127, 129-132, 240, 244, 250-252)
      - CPOOL Support
    - **Dataspace Pageable Large Pages**
      - DSPSERV (Dataspace) Support

© 2013 IBM Corporation

Overview

- ***Benefit / Value***
  - **Same performance benefits as fixed large pages with the flexibility of pages that can be stolen/paged out**
  - **Pageable large pages will be allocated from a new z/OS managed memory pool called the PLAREA (separate from the LFAREA)**
  - **This pool will be automatically configured by the system**
  - **Pageable Large Pages can reside in either reconfigurable or non-reconfigurable storage**

© 2013 IBM Corporation

Installation

- *z/OS 1.13 plus installation of the Web Deliverable to obtain the "RSM Enablement Offering"*

IBM Confidential

© 2013 IBM Corporation

Interactions & Dependencies

- ***Software Dependencies***
  - **None**
- ***Hardware Dependencies***
  - **zFlash Express (zEC12 machines)**
- ***Exploiters***
  - **None**

© 2013 IBM Corporation

Externals Overview

- ***This line item focuses on 64-bit Pageable Large Page Support***
  - **Via IARV64 GETSTOR/GETCOMMON macro**
- ***When a large page memory object is backed will depend on the PAGEFRAMESIZE attribute of the memory object***
  - **Fixed Large Page object pages will be backed at allocation time**
  - **Pageable/DREF Large Page object pages will be backed when referenced**
- ***When any virtual address within the megabyte is referenced, the whole megabyte will be backed by a pageable large page***

© 2013 IBM Corporation

Externals Overview

- ***For pageable/DREF large memory objects, we will support mixed memory objects***
  - **Parts of the memory object can be backed by 1MB page frames while other parts can be backed with 4KB page frames**
  - **Page frame size will be decided when the megabyte is backed (either at fault-time or page-in)**
- ***Pageable large pages can be paged out***
  - **Without FLASH, large pages will be demoted to 256 4KB pages, a page table will be built and the pages will be paged out to DASD**
  - **With FLASH, large pages will be paged out to FLASH and retain their 'large page' attribute when paged back in**

© 2013 IBM Corporation

Externals Overview

- ***Certain behavior may cause 1M pageable large page demotion*, which in turn, eliminates any performance gains through the use of large pages***
- ***The following IARV64 services may cause demotion if not 1MB aligned and not in megabyte multiples:***
  - **PAGEFIX**
  - **DISCARDDATA**
  - **PAGEOUT**
  - **PROTECT**
  - **UNPROTECT**

**** Demotion = The breaking up of a 1M pageable large page into 256 4K pages. Once demoted, it will remain demoted until all 256 4K pages are returned.***

© 2013 IBM Corporation

Externals: Executable Macros

- ***IARV64 GETSTOR and GETCOMMON services enhanced to support Pageable Large Pages:***
  - **PAGEFRAMESIZE=PAGEABLE1MEG|DREF1MEG**
  - **Existing PAGEFRAMESIZE parameter supports the above keyword values**
  - **PAGEFRAMESIZE=PAGEABLE1MEG**
    - The memory object should be backed by pageable 1 megabyte pages if possible. If none are available, the request will be backed by 4K pages.
  - **PAGEFRAMESIZE=DREF1MEG**
    - The memory object should be backed by pageable 1 megabyte pages that can be referenced while running disabled if possible. If none are available, the request will be backed by 4K pages.

© 2013 IBM Corporation

Externals: Executable Macros

- ***TYPE=PAGEABLE|DREF Behavior with Pageable Large Pages***
  - **PAGEFRAMESIZE=PAGEABLE1MEG|DREF1MEG with TYPE=**
    - The default is TYPE=PAGEABLE when:
      - PAGEFRAMESIZE=PAGEABLE1MEG is specified or
      - PAGEFRAMESIZE=4K is specified or
      - PAGEFRAMESIZE=MAX is specified and the memory object is backed by 4K pages
    - The default is TYPE=DREF when:
      - PAGEFRAMESIZE=DREF1MEG is specified
    - Implicitly fixed when:
      - PAGEFRAMESIZE=1MEG is specified or
      - PAGEFRAMESIZE=MAX is specified and the memory object is backed by fixed 1M pages
  - **PAGEFRAMESIZE takes precedence over TYPE**
    - If PAGEFRAMESIZE=PAGEABLE1MEG is specified with TYPE=DREF, the request is for pageable (not DREF) 1M pages

© 2013 IBM Corporation

Externals: Executable Macros

- ***IARV64 PROTECT/UNPROTECT service enhanced to improve usability***
  - **Prior to this support, the value specified in the AMOUNT parameter was determined by the contents of the memory object**
    - If the memory object was backed by fixed large pages, the value was in units of 1M pages
    - If the memory object was backed by 4K pages, the value was in units of 4K pages
  - **New optional parameter: AMOUNTSIZE=4K|1MEG**
    - AMOUNTSIZE=4K requests that the AMOUNT value specified on the RANGLIST parameter is treated as the number of 4K pages to be acted on
    - AMOUNTSIZE=1MEG requests that the AMOUNT value specified on the RANGLIST parameter is treated as the number of 1M pages to be acted on

© 2013 IBM Corporation

Externals: Mapping Macros

- ***The following fields have been added to the RSM Control and Enumeration Area (IARRCE):***

Offsets	Type	Len	Name/Description
334 (14E)	X'10'		RCEPAGEABLELARGE Pageable Large supported
952 (3B8)	CHARACTER	8	RCEPLSZ Initial Pageable Large Area size (in bytes)
960 (3C0)	DBL WORD	8	RCEPMMSS Number of failed attempts to back storage with preferred pageable large frames
968 (3C8)	DBL WORD	8	RCEPLSID Number of system-initiated demotions from pageable large frame groups to 4K page frames
976 (3D0)	DBL WORD	8	RCEPLRID Number of request-initiated demotions from pageable large frame groups to 4K page frames

© 2013 IBM Corporation

Externals: Mapping Macros

- ***The following fields have been added to the RSM Control and Enumeration Area (IARRCE):***

Offsets	Type	Len	Name/Description
984 (3D8)	SIGNED	4	RCEPMAFC Number of available pageable large frame groups (pref)
988 (3DC)	SIGNED	4	RCEPLHWM High water mark for the number of pageable large frame groups used by the system
992 (3E0)	SIGNED	4	RCEPSAFC Number of available pageable large single frames (pref)
996 (3E4)	SIGNED	4	RCEPLTHRESMINCNT Minimum threshold count for the number of available pageable large frame groups

© 2013 IBM Corporation

Externals: Mapping Macros

- ***The following fields have been added to the RSM Control and Enumeration Area (IARRCE):***

Offsets	Type	Len	Name/Description
1000 (3E8)	SIGNED	4	RCEPLTHRESMAXCNT Maximum threshold count for the number of available pageable large frame groups
1004 (3EC)	SIGNED	1	RCEPLTHRESMINPCT Minimum threshold percentage for the number of available pageable large frame groups
1005 (3ED)	SIGNED	1	RCEPLTHRESMAXPCT Maximum threshold percentage for the number of available pageable large frame groups
1008 (3F0)	SIGNED	4	RCEPLFRM Number of pageable large frame groups in-use by the system

© 2013 IBM Corporation

Externals: Mapping Macros

- **The following fields have been added to the RSM Control and Enumeration Area (IARRCE):**

Offsets	Type	Len	Name/Description
1012 (3F4)	SIGNED	4	RCENSAFC Number of available pageable large single frames (non-pref)
1052 (41C)	SIGNED	4	RCELARGEUSEDPL Number of fixed large frame used to satisfy pageable large frame requests
1056 (420)	SIGNED	4	RCEPLXRM Number of pageable large frame groups that are fixed
1060 (424)	SIGNED	4	RCENMAFC Number of available pageable large frame groups (non-pref)

© 2013 IBM Corporation

Externals: Mapping Macros

- **The following fields have been added to the RSM Control and Enumeration Area (IARRCE):**

Offsets	Type	Len	Name/Description
1064 (428)	DBL WORD	8	RCENMMSS Number of failed attempts to back storage with pageable large frames (non-pref)
1244 (4DC)	SIGNED	4	RCELARGEALLOCATEDPL Number of Fixed Large Pages allocated as Pageable Large Pages - different from RCELargeUsedPL which is the number of fixed large pages currently used as pageable large pages. If they were to be demoted, they would not be counted in RCELargeUsedPL. Demoted pageable large pages are included in this field.
1248 (4E0)	SIGNED	4	RCEPLTOTAL Total Number of Pageable Large Pages

© 2013 IBM Corporation

Externals: Mapping Macros

- ***The following fields have been added to the RSM Control and Enumeration Area (IARRCE):***

Offsets	Type	Len	Name/Description
1252 (4E4)	SIGNED	4	RCELARGEUSEDPLHWM High-Water mark of the number of large pages allocated on behalf of pageable large requests
1256 (4E8)	SIGNED	4	RCELARGEUSED1MHWM High-Water mark of the number of large pages allocated on behalf of fixed large requests
1262 (4EC)	SIGNED	4	RCELARGEUSED4KHWM High-Water mark of the number of large pages allocated on behalf of 4K page requests

© 2013 IBM Corporation

Externals: Mapping Macros

- ***The following fields have been added to the RSM Address Space Block Extension (IARRAX):***

Offsets	Type	Len	Name/Description
336 (150)	SIGNED	4	RAXPLFRM Number of pageable large frame groups currently used by this address space
340 (154)	SIGNED	4	RAXPLHWM High-Water mark for the number of pageable large frame groups used by this address space
344 (158)	DBL WORD	8	RAXPMMSS Number of failed attempts to back storage with pageable large frames by this address space (pref)
352 (160)	DBL WORD	8	RAXPLSID Number of system-initiated demotions from pageable large frames groups to 4k page frames for this address space

© 2013 IBM Corporation

Externals: Mapping Macros

- ***The following fields have been added to the RSM Address Space Block Extension (IARRAX):***

Offsets	Type	Len	Name/Description
360 (168)	DBL WORD	8	RAXPLSID Number of request-initiated demotions from pageable large frames groups to 4k page frames for this address space
368 (170)	DBL WORD	8	RAXNMMSS Number of failed attempts to back storage with pageable large frames by this address space (non-pref)
370 (178)	SIGNED	4	RAXPLXRM Number of pageable large frame groups currently fixed by this address space

© 2013 IBM Corporation

Externals: System Commands

- ***DISPLAY VIRTSTOR,{HVSHARE|HVCOMMON|LFAREA}***
  - **Existing system command was enhanced to report (via message IAR019I) the status of the LFAREA (how the fixed large pages are currently being used by the system)**
- ***See Externals: Messages section for complete details***
- ***DISPLAY VIRTSTOR,LFAREA was primarily added to help determine if the right LFAREA size was specified for your system. For example,***
  - **If the high water mark for the number of fixed large pages used on behalf of 4k page requests is high, you may want to decrease your LFAREA size or add more memory to your configuration.**

© 2013 IBM Corporation

Externals: Messages

- *D VIRTSTOR,LFAREA*

```
IAR019I  14.37.22 DISPLAY VIRTSTOR 735
 SOURCE =  WE
 TOTAL LFAREA = 64M
 LFAREA AVAILABLE = 32M
 LFAREA ALLOCATED (1M) = 16M
 LFAREA ALLOCATED (4K) = 4M
 LFAREA ALLOCATED (PAGEABLE1M) = 12M
 MAX LFAREA ALLOCATED (1M) = 62M
 MAX LFAREA ALLOCATED (4K) = 5M
 MAX LFAREA ALLOCATED (PAGEABLE1M) = 14M
```

© 2013 IBM Corporation

Externals: Abend and Abend reason codes

- ***New reason code for ABEND DC2 are as follows:***
  - **0065**
    - There are no frames available to allow demotion of large page that the discard was attempted against.
  - **006B**
    - Protect failed, requester specified a protect range that included pages backed by fixed large frames with AMOUNTSIZE=4K
  - **0070**
    - Pagefix failed, range must start on a segment boundary and number of pages to process must be in multiples of 256
    - This is only the case for large pages that have already been fixed since we do not want to demote them once the large pages are fixed

© 2013 IBM Corporation

Internals Details

- ***Real storage consists of increments that are online or offline***
- ***Increment size depends on installed real storage amount***
- ***Example: machine z196 or z10***

Amount of Real	Increment Size
0G - 128GB	256MB
>28GB to 256GB	512MB
>256GB to 512GB	1GB
>512GB to 1TB	2GB

- The increments are represented in RSM by AIME control blocks
- 4K frames are represented in RSM by PFTE control blocks

© 2013 IBM Corporation

Internals Details

- ***Initialization – Old Storage Map***

- RSU
  - 0% to 99% total storage, or offline only
  - Range: top of storage down to top of V=R area
- Quad
  - 12.5% total online storage
  - Can take from RSU
  - Range: top of storage down to 16M line
- 1M LFArea (Fixed 1M Large Pages)
  - (0% to 80% online storage) – 2G
  - Can take from RSU
  - Range: top of storage down to 2G Bar

© 2013 IBM Corporation

Internals Overview

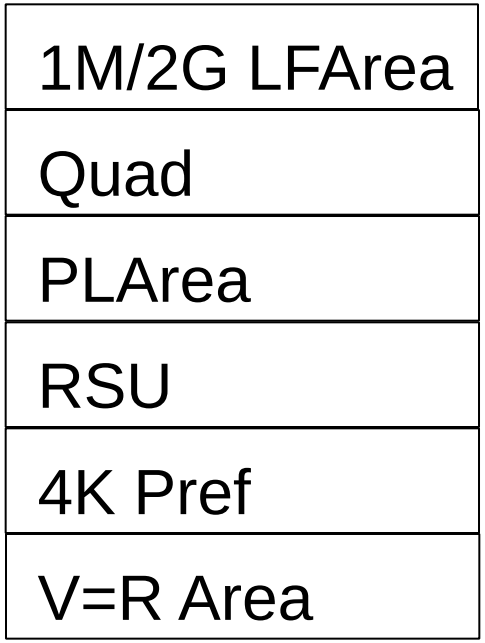
- ***Initialization – Old Storage Map***

- 4K Pref
  - 0 to xx% online storage
  - Range: top of storage to top of V=R
- V=R Area (sysparm REAL)
  - 0 to xx% online storage
  - Range: 24M (top of system region) up to bottom of Nuc
  - Created earlier in IAXMU

© 2013 IBM Corporation

Internals Overview

Initialization – New Storage Map

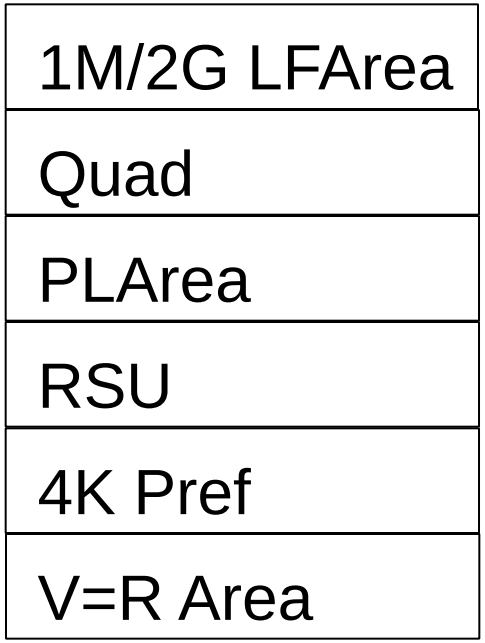


- 1M/2G LFArea (Fixed 1M/2G Pages)
  - (0% to 80% online storage) – 2G or 0% to 80% (online storage - 4G)
  - Range: top of storage down to 2G Bar
- Quad
  - 12.5% total online storage
  - Range: top of storage down to 16M line
- PLArea (Pageable 1M Large Pages)
  - 12.5% of (online storage – quad size)
  - Range: top of storage down to 2G Bar

© 2013 IBM Corporation

Internals Overview

- ***Initialization – New Storage Map***

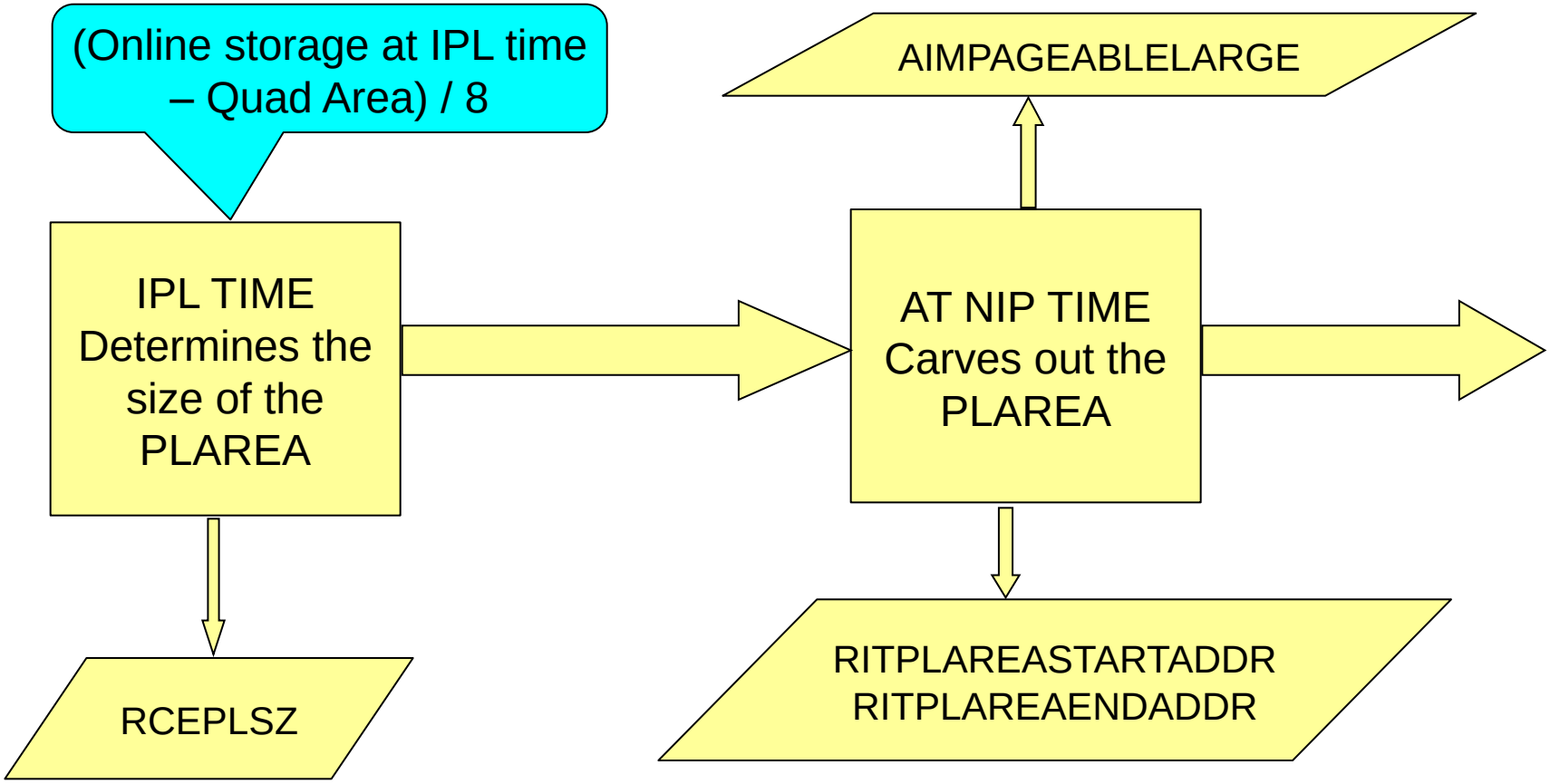


- RSU
  - 0% to 99% total storage, or offline only
  - Range: top of storage down to top of V=R area (can shrink via convert)
- 4K Pref
  - 0 to xx% online storage
  - Range: top of storage to top of V=R
- V=R Area (sysparm REAL)
  - 0 to xx% total online storage
  - Range: 24M (top of system region) up to bottom of Nuc
  - Created earlier in IAXMU

© 2013 IBM Corporation

Internals Overview

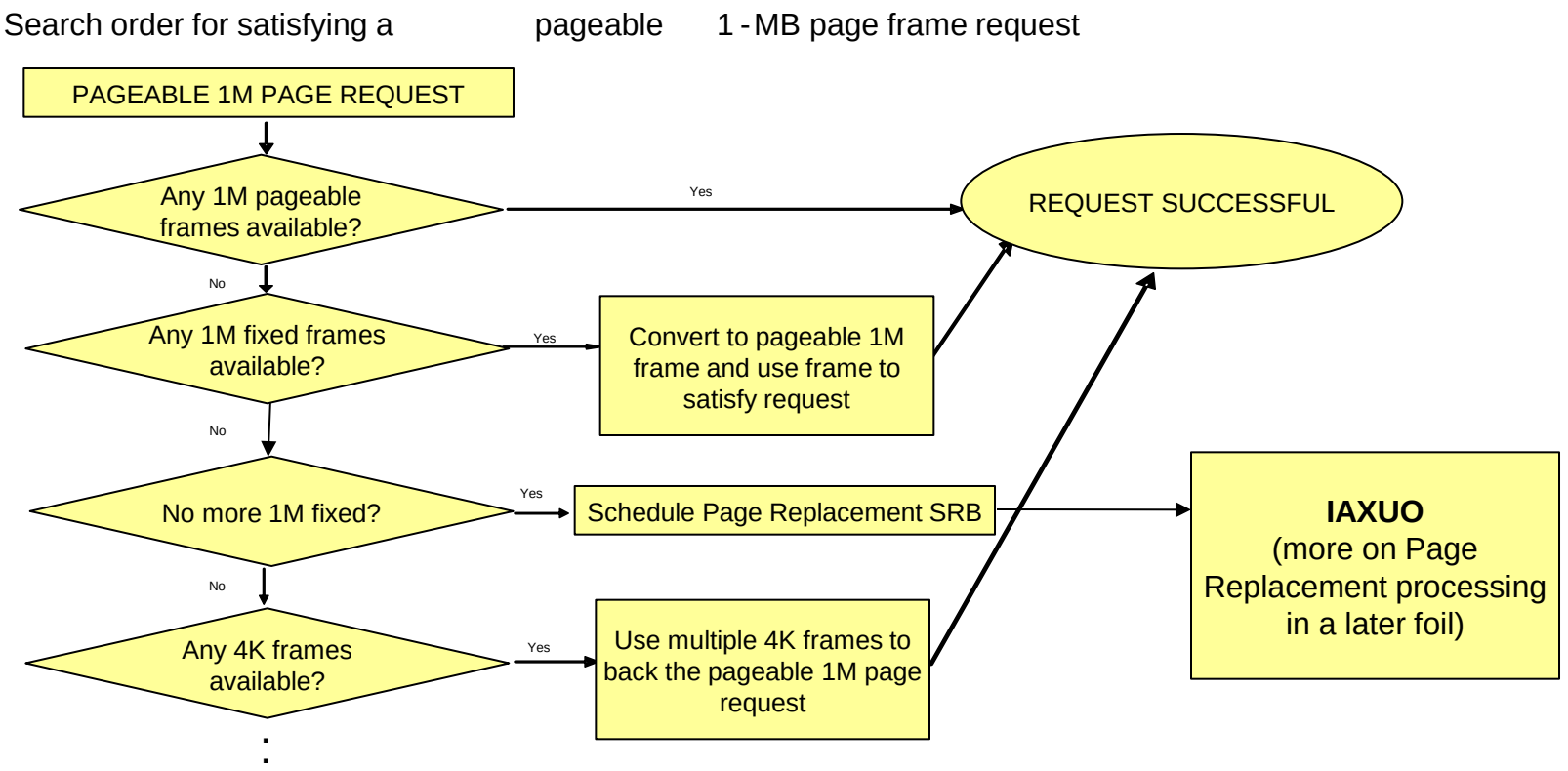
- ***Creating the PLAREA (Pageable Large Area)***



© 2013 IBM Corporation

Internals Overview

- ***A closer look at processing a Pageable 1M page request***

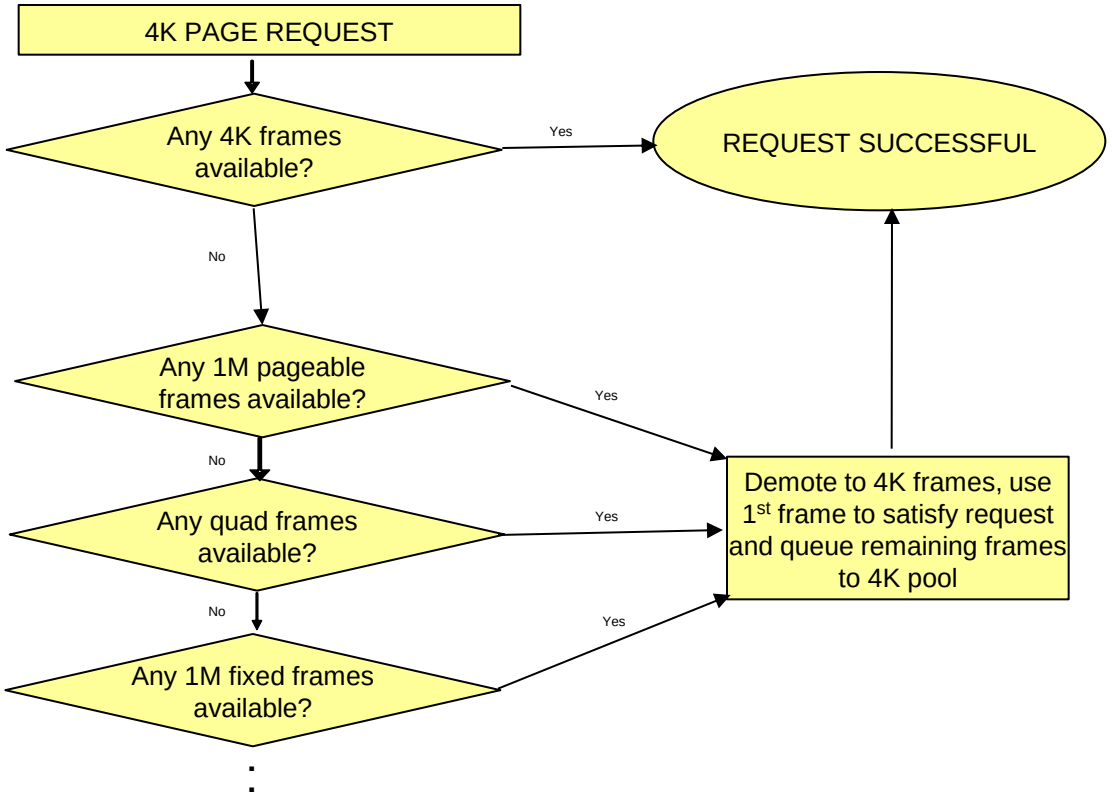


© 2013 IBM Corporation

Internals Overview

- ***A closer look at processing a 4K page request***

Search order for satisfying a 4K page frame request -



© 2013 IBM Corporation

2GB Fixed Large Pages

© 2013 IBM Corporation

Overview

- ***Problem Statement / Needs Addressed: Performance degradation due to increased TLB (Translation Lookaside Buffer) misses***
  - **Over the past few years application memory sizes have dramatically increased due to support for 64-bit addressing in both physical and virtual memory**
  - **TLB sizes have remained relatively small due to low access time requirements and hardware space limitations**
  - **Therefore TLB coverage today represents a much smaller fraction of an applications working set size leading to a larger number of TLB misses**
  - **Applications can suffer a significant performance penalty resulting from an increased number of TLB misses as well as the increased cost of each TLB miss**
  - **Introduction of 1M pages in z/OS v1r9 was a good start (*see Appendix*)**

© 2013 IBM Corporation

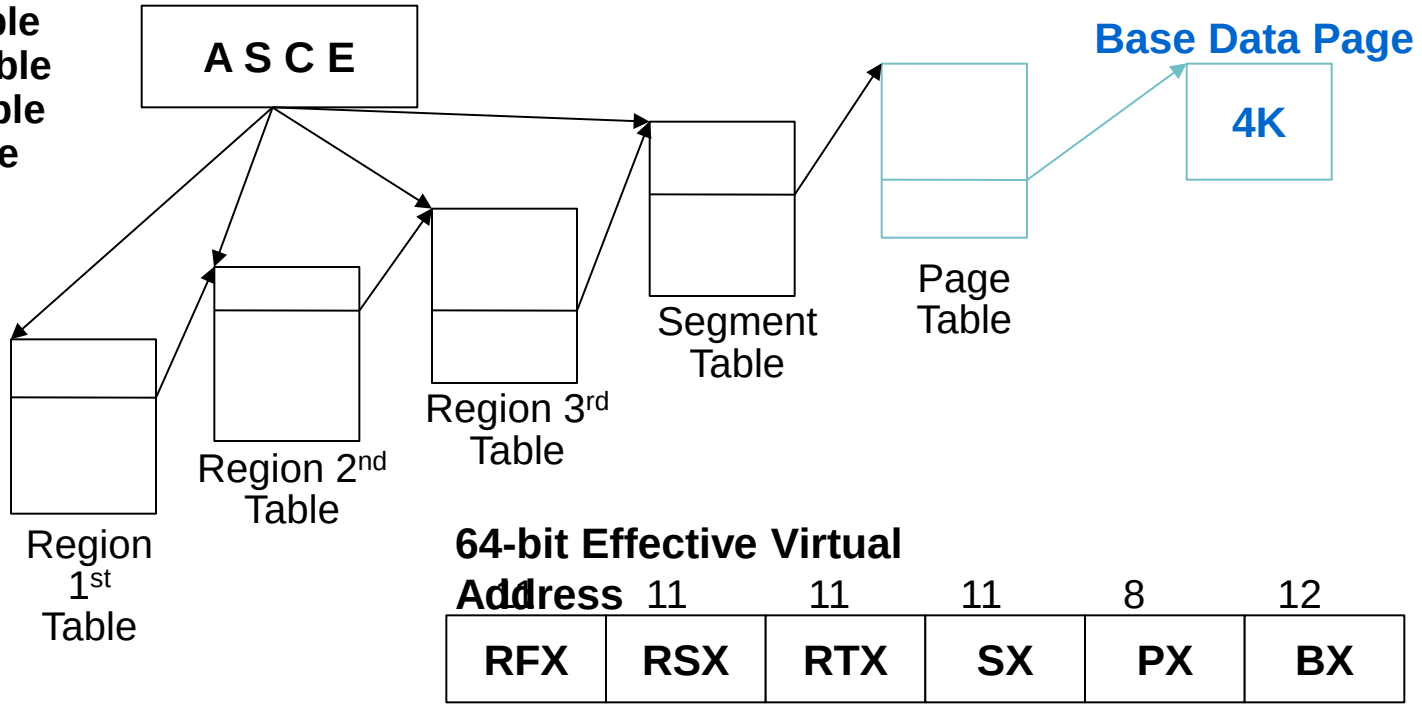
Overview

- ***Solution: Effectively Increase TLB coverage (not size) by using 2G pages***
  - **A 2G page is 524,288 times larger than a 4K base page and 2,048 times larger than a 1M Large page**
  - **2G Page allows for a single TLB entry to fulfill many more address translations than either a 1M Large page or 4K base page**
- ***Benefit / Value***
  - **Better performance by decreasing the number of TLB misses that an exploiter application incurs**
    - Less time spent converting virtual addresses into physical addresses
    - Less real storage used to maintain DAT structures

© 2013 IBM Corporation

Overview: 4K Base Page DAT Architecture

ASCE.60-61
11 – Region 1st table
10 – Region 2nd table
01 – Region 3rd table
00 – Segment table

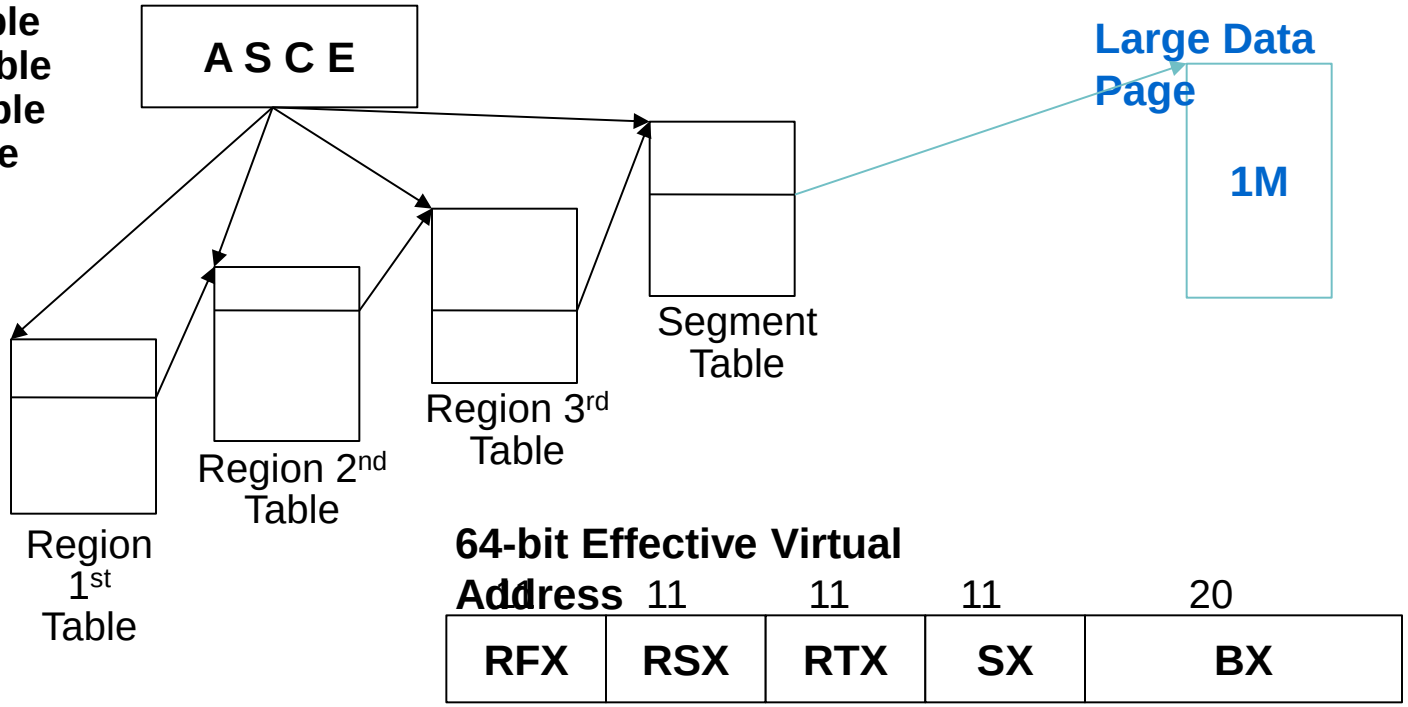


- Translation can start at R1T, R2T, R3T or SGT
- The starting point of the translation is designated in the Address Space Control Element (ASCE.60-61)

© 2013 IBM Corporation

Overview: 1M Large Page DAT Architecture

ASCE.60-61
11 – Region 1st table
10 – Region 2nd table
01 – Region 3rd table
00 – Segment table

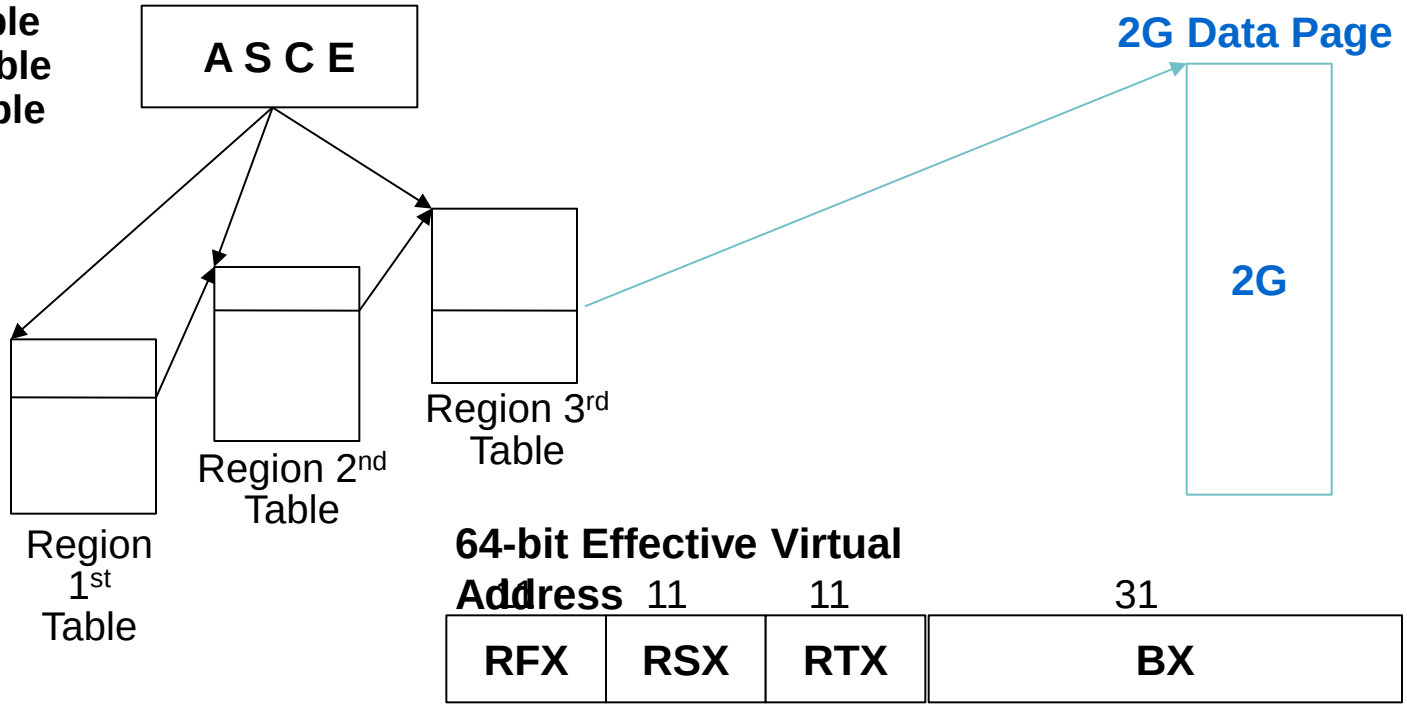


- Translation can start at R1T, R2T, R3T or SGT
- The starting point of the translation is designated in the Address Space Control Element (ASCE.60-61)

© 2013 IBM Corporation

Overview: 2G Page DAT Architecture

ASCE.60-61
11 – Region 1st table
10 – Region 2nd table
01 – Region 3rd table



- Translation can start at R1T, R2T, or R3T
- The starting point of the translation is designated in the Address Space Control Element (ASCE.60-61)

© 2013 IBM Corporation

Overview

- ***The 2G Page size in z/OS will be 2 Gigabytes***
  - **2GB page size was chosen because it is the natural next size when considering z/Architecture DAT structures**
- ***4K, 1M and 2G page sizes will be supported***
- ***Customers specify the amount of real storage to be used for both 1M Large and 2G Pages via the LFAREA keyword on IEASYSxx***
- ***If the system is constrained for 4K pages***
  - **2GB pages will NOT be used to back 4K page requests**
  - **1MB pages will continue to be used to back 4K page**
- ***1M Fixed Large and 2G Pages will NOT be reconfigurable***

© 2013 IBM Corporation

Overview

- ***2G Pages are implicitly fixed pages and will not be paged out to AUX***
  - **primary reason: lack of a Page Table/External Page Table for 2G pages**
    - External Page Table is a logical extension of the Page Table that contains information such as where a page was placed on Auxiliary storage
  - **Other reasons include:**
    - The great delay in paging in a 2GB page
    - The expense of constructing 2GB of contiguous real storage from 524,288 4K frames when required
- ***2G Pages will be backed at allocation time by 524,288 contiguous 4K real storage frames***

© 2013 IBM Corporation

Externals: Usage & Invocation

- ***The following IARV64 requests are enhanced for 2G Pages:***
  - **GETSTOR – obtains above the bar memory objects backed by 2G Pages**
  - **DETACH – Frees a memory object backed by 2G Pages**
  - **PROTECT/UNPROTECT – Protects/unprotects memory object backed by 2G Pages**
  - **LIST – Provides information about memory objects, including whether they are backed by 2G Pages**

© 2013 IBM Corporation

Externals: Usage & Invocation

- ***The following IARV64 requests cannot be used on 2G Pages:***
  - **DISCARDDATA – since memory objects that are backed by 2G Pages are implicitly 'fixed'**
  - **PAGEFIX/PAGEUNFIX - since memory objects that are backed by 2G Pages are implicitly 'fixed'**
  - **PAGEOUT/PAGEIN - since memory objects that are backed by 2G Pages are implicitly 'fixed'**
  - **CHANGEGUARD – at this time, IBM does not see the need to support this function for 2G pages**
  - **GETSHARED/SHAREMEMOBJ/CHANGEACCESS - IBM may support this function at a later date**
  - **GETCOMMON - IBM may support this function at a later date**

© 2013 IBM Corporation

Externals: Usage & Invocation Examples

- *Prior to 2G Page support (and still supported), 2G of above the bar storage backed by 1M Large Pages can be requested using SEGMENTS:*
  - IARV64 REQUEST=GETSTOR, SEGMENTS=2048, PAGEFRAMESIZE=1MEG, ORIGIN=output@
- *With 2G Page support, 2G of above the bar storage backed by 1M Large Pages can be requested using new keywords UNITS and UNITSIZE, and a new argument for PAGEFRAMESIZE (1M instead of 1MEG):*
  - IARV64 REQUEST=GETSTOR, UNITS=1, UNITSIZE=2G, PAGEFRAMESIZE=1M, TYPE=FIXED, ORIGIN=output@
- *With 2G Page support, 2G of above the bar storage backed by 2G Pages can be requested using new keywords UNITS and UNITSIZE, and a new argument for PAGEFRAMESIZE:*
  - IARV64 REQUEST=GETSTOR, UNITS=1, UNITSIZE=2G, PAGEFRAMESIZE=2G, TYPE=FIXED, ORIGIN=output@

© 2013 IBM Corporation

Externals: Usage & Invocation

IARV64 REQUEST=GETSTOR
{SEGMENTS=xsegments,
PAGEFRAMESIZE=4K,
PAGEFRAMESIZE=1MEG,
PAGEFRAMESIZE=MAX,
TYPE=PAGEABLE,
TYPE=DREF,
UNITS=xunits,
UNITSIZE=1M,
PAGEFRAMESIZE=4K,
PAGEFRAMESIZE=1M,
UNITSIZE=2G,
PAGEFRAMESIZE=4K,
PAGEFRAMESIZE=1M,
PAGEFRAMESIZE=2G,
TYPE=PAGEABLE,
TYPE=DREF,
TYPE=FIXED}

NOTES:

SEGMENT and UNITS are mutually exclusive parameters.

When SEGMENTS is used, there are default values for PAGEFRAMESIZE and TYPE.

When UNITS is used, UNITSIZE, PAGEFRAMESIZE and TYPE must be specified - there are no defaults.

The way you code SEGMENTS is unchanged (as well as its behavior).

© 2013 IBM Corporation

Externals: Usage & Invocation

New GETSTOR Syntax

- *UNITS specifies the number of units.*
- *UNITSIZE indicates the size to be applied to the UNITS specification.*
  - **When UNITSIZE=1M is specified, the memory object size is in one megabyte units. For example, a request for UNITS=3 with UNITSIZE=1M is a request for 3 megabytes of virtual storage starting on a 1 megabyte boundary.**
  - **When UNITSIZE=2G, the memory object size is in two gigabyte units. For example, a request for UNITS=3 with UNITSIZE=2G is a request for 6 gigabytes of virtual storage starting on a 2 gigabyte boundary.**

© 2013 IBM Corporation

Externals: Usage & Invocation

New GETSTOR Syntax

- ***PAGEFRAMESIZE used with UNITS is similar in behavior as with SEGMENTS except for the following exceptions:***
  - **PAGEFRAMESIZE must be specified (no defaults)**
  - **PAGEFRAMESIZE=4K is not supported with TYPE=FIXED (error at assembly time)**
  - **PAGEFRAMESIZE=1M is supported (SEGMENTS supported PAGEFRAMESIZE=1MEG)**
  - **PAGEFRAMESIZE=MAX is not supported (either you get the requested page frame size or the request is failed)**
  - **PAGEFRAMESIZE=2G is supported to request memory objects backed by 2G frames**
  - **PAGEFRAMESIZE=2G only valid with TYPE=FIXED**

© 2013 IBM Corporation

Externals: Usage & Invocation

IARV64 GETSTOR Examples

UNITS syntax	Equivalent (if any) SEGMENTS syntax	Resulting page frame size and type
Units=x, Unitsize=1M, Pageframesize=1M, Type=PAGEABLE	Segments=x, Pageframesize=PAGEABLE1MEG, Type=PAGEABLE	1M pageable
Units=x, Unitsize=1M, Pageframesize=1M, Type=DREF	Segments=x, Pageframesize=DREF1MEG, Type=DREF	1M DREF (from PLAREA)
Units=x, Unitsize=1M, Pageframesize=1M, Type=FIXED	Segments=x, Pageframesize=1MEG	1M fixed (from LFAREA)

Note: There is no PAGEFRAMESIZE=MAX equivalent when specifying UNITS

© 2013 IBM Corporation

Externals: Usage & Invocation

IARV64 GETSTOR Examples

UNITS syntax	Equivalent (if any) SEGMENTS syntax	Resulting page frame size and type
Units=x, Unitsize=2G, Pageframesize=2G, Type=PAGEABLE		Not supported - Syntax error at assembly time
Units=x, Unitsize=2G, Pageframesize=2G, Type=DREF		Not supported - Syntax error at assembly time
Units=x, Unitsize=2G, Pageframesize=2G, Type=FIXED		2G fixed (from LFAREA)

© 2013 IBM Corporation

Externals: Mapping Macros

- **The following fields have been added to the RSM Control and Enumeration Area (IARRCE):**

Offsets	Type	Len	Name/Description
1256(4E8)	DBL WORD	8	RCE2GMEMORYOBJECTS Number of 2G Memory Objects allocated in the system
1264(4F0)	DBL WORD	8	RCE2GPAGESBACKEDINREAL Number of 2G Pages backed in real storage
1272(4F8)	DBL WORD	8	RCE2GNONRECONLFASIZE Size of Non-Reconfigurable 2G Frame Area in 2G units
1280(500)	DBL WORD	8	RCE2GNONRECONLFAUSED Number of 2G frames in the Non-Reconfigurable 2G Frame Area that are allocated
1288(508)	SIGNED	4	RCE2GHWM High Water Mark of the number of 2G Pages allocated in the system

© 2013 IBM Corporation

Externals: Mapping Macros

- ***The following fields have been added to the RSM Address Space Block Extension (IARRAX):***

Offsets	Type	Len	Name/Description
552 (228)	DBL WORD	8	RAX2GMEMORYOBJECTS Number of 2G Memory Objects allocated by this address space
560 (230)	DBL WORD	8	RAX2GPAGESBACKEDINREAL Number of 2G Pages backed in real storage owned by this address space

© 2013 IBM Corporation

Externals: Configuration & Setup

- ***Old form sysparm LFAREA (still supported)***
  - **LFAREA = (xx% | mmmmmmM | gggggG | ttttttT)**
  - **Pct formula: (xx% of online storage at IPL) minus 2G**
  - **Max LFAREA is 80%**
    - Example for 16G online real: (.8 * 16G) – 2G = 12.8G – 2G = 10.8G
- ***New form sysparm LFAREA (supports both 1M and 2G)***
  - **LFAREA = ([1M=req] [,2G=req] [,prompt | noprompt])**
  - **Pct formula: xx% of (online storage at IPL minus 4G)**
  - **Max LFAREA is 80%**
    - Example for 16G online real: .8 * (16G – 4G) = .8 * 12G = 9.6G
- ***Default LFAREA: zero 1MB pages and zero 2GB pages***
- ***Located in the non-reconfigurable area and takes precedence over RSU***

© 2013 IBM Corporation

Externals: Configuration & Setup

- ***LFAREA = ([1M=req] [,2G=req] [,prompt | noprompt])***
  - **req is: (target[%] [,minimum[%]])**
    - **target** specifies the desired number of pages (1M or 2G) to be reserved.
    - **target%** specifies the percentage of (online real storage at IPL – 4G) to be reserved for 1M or 2G pages.
    - **minimum** specifies the minimum number of pages that will be acceptable in satisfying the request if the target number can not be reserved.
    - **minimum%** specifies the minimum percentage of (online real storage at IPL – 4G) to be reserved for pages that will be acceptable in satisfying the request if the target percentage can not be reserved.

© 2013 IBM Corporation

Externals: Configuration & Setup

- ***LFAREA = ([1M=req] [,2G=req] [,prompt | noprompt])***
  - **prompt indicates that the operator should be prompted for a new LFAREA= specification if the request can not be satisfied by at least the minimum (which defaults to target if not specified), or the sum of minimums. The default is prompt.**
    - Example: 16G system, LFAREA=(1M=(10%),2G=(12,10)) will result in a series of messages and a prompt for a new specification.
  - **noprompt indicates that the operator is not to be prompted if the request can not be satisfied, and that instead, zero pages are to be reserved for both 1M and 2G pages. Note that syntax errors, however, will always result in a prompt for re-specification.**
  - **Rework !!!**

© 2013 IBM Corporation

Externals: Messages

- **Deleted Messages (replaced by new messages):**
  - IAR020I THE LFAREA WAS SPECIFIED BUT LARGE PAGE SUPPORT IS NOT AVAILABLE
  - IAR021I THE LFAREA WAS SPECIFIED BUT SUFFICIENT STORAGE IS NOT AVAILABLE
  - IAR022I MAXIMUM AVAILABLE LFAREA SIZE IS YYYYYYM
  - IAR023A INVALID LFAREA PARM- RESPECIFY OR PRESS ENTER FOR LFAREA=0M
  - IAR025I THE LFAREA WAS NOT COMPLETELY SATISFIED - SIZE IS SET TO YYYYYYM
  - IAR028I THE LFAREA SPECIFICATION RESULTED IN xxxxxxxx 1MB PAGES BEING RESERVED IN THE LARGE FRAME AREA

© 2013 IBM Corporation

Externals: Messages

- **Changed Message IAR006A:**
  - Old: IAR006A INVALID {VRREGN | REAL | RSU} PARM - RESPECIFY OR PRESS ENTER FOR THE DEFAULT
  - New: IAR006A INVALID {VRREGN | REAL} PARM - RESPECIFY OR PRESS ENTER FOR THE DEFAULT
- **Changed Message IAR013I**
  - Old: IAR013I xxxxxxM STORAGE IS RECONFIGURABLE
  - New: IAR013I xxxxxx{M|G|T} STORAGE IS RECONFIGURABLE
- **Changed Message IAR026I**
  - Old: IAR026I THE RSU VALUE SPECIFIED EXCEEDS THE TOTAL AMOUNT OF REAL STORAGE AVAILABLE ON THIS SYSTEM: yyyyyyyyM
  - New: IAR026I THE RSU VALUE SPECIFIED EXCEEDS THE TOTAL AMOUNT OF REAL STORAGE AVAILABLE ON THIS SYSTEM: yyyyyyyy{M|G|T}

© 2013 IBM Corporation

Externals: Messages

- **New Messages:**
  - IAR029I RSU VALUE IS NOT VALID.
  - IAR030I PAGESCM VALUE IS NOT VALID - AT THE FOLLOWING PROMPT RESPECIFY OR PRESS ENTER FOR PAGESCM=ALL.
  - IAR040I REAL STORAGE AMOUNTS:

    TOTAL AVAILABLE ONLINE: xxx[M | G | T]

    LFAREA LIMIT FOR xM, xG, OR xT : xxx[M | G | T] | [0 (not supported)]

    LFAREA LIMIT FOR SUM OF 1M= AND 2G= : xxx[M | G | T] | [0 (not supported)]

    LFAREA LIMIT FOR 2GB PAGES FOR 2G= : [xxxxxxx] | [0 (not supported)]
  - IAR041I LFAREA=(specification) WAS SPECIFIED BUT [2GB PAGE SUPPORT IS | 1MB PAGE SUPPORT AND 2GB PAGE SUPPORT ARE] NOT AVAILABLE.
  - IAR042I LFAREA=(specification) WAS SPECIFIED WHICH RESULTS IN A REQUEST FOR [ZERO 1MB] [AND] [ZERO 2GB] PAGES.
  - IAR043I LFAREA=(specification) WAS SPECIFIED WHICH TOTALS [xxxxxxxM | xxxxG | xT] AND EXCEEDS THE SYSTEM LIMIT OF [yyyyyyyM | yyyyG | yT].
  - IAR044I LFAREA=(specification) WAS SPECIFIED WHICH EXCEEDS THE AMOUNT AVAILABLE IN THE LARGE FRAME AREA.

© 2013 IBM Corporation

Externals: Messages

- ***New Messages:***
  - **IAR045I VALID RANGE FOR LFAREA [xM, xG, xT | 1M= | 2G=] IS 0[M] TO xxxxxxx[M], OR 0% TO xx%. A MINIMUM OF yy% MUST BE SPECIFIED TO RESERVE AT LEAST ONE [1MB | 2GB] PAGE.**
  - **IAR045I VALID VALUE FOR LFAREA [xM | 1M= | 2G=] IS 0[M] OR 0%. [HARDWARE SUPPORT UNAVAILABLE | NO STORAGE AVAILABLE] FOR [1MB | 2GB] PAGES.**
  - **IAR046I LFAREA=(specification) WAS SPECIFIED WHICH IS NOT VALID. REASON:**

    [VALUE EXCEEDS MAXIMUM NUMBER OF DIGITS.]

    [MINIMUM VALUE GREATER THAN TARGET VALUE.]

    [MISSING EXPECTED VALUE.]

    [EXTRANEOUS CHARACTERS DETECTED.]

    [DUPLICATE KEYWORDS OR SPECIFICATIONS DETECTED.]

    [MUTUALLY EXCLUSIVE KEYWORDS OR VALUES DETECTED.]

    [MISSING MULTIPLIER M, G, OR T.]

    [UNRECOGNIZED SPECIFICATION.]

    [NON-NUMERIC DETECTED WHERE NUMERIC EXPECTED.]

    [PERCENTAGE WITH NON-PERCENTAGE IN SAME KEYWORD.]

    [REQUEST WITH PERCENTAGE EXCEEDS SYSTEM LIMIT.]

© 2013 IBM Corporation

Externals: Messages

- ***New Messages:***
  - **IAR047I AT THE FOLLOWING PROMPT, SPECIFY THE COMPLETE LFAREA PARAMETER OR PRESS ENTER FOR ZERO 1MB AND ZERO 2GB PAGES**
    - Note: IAR047I will be issued before the system issues

      IEA341A RESPECIFY LFAREA= PARM OR PRESS ENTER.
  - **IAR048I [LFAREA=(specification) WAS PROCESSED | LFAREA WAS NOT SPECIFIED | LFAREA WAS CANCELED] WHICH RESULTED IN xxxxxxx 1MB PAGE[S] AND yyyyyyy 2GB PAGE[S] [. REQUEST REDUCED DUE TO: | DUE TO: NOPROMPT SPECIFIED AND]**

    [INSUFFICIENT STORAGE.]

    [OVER SYSTEM LIMIT.]

    [NO HARDWARE SUPPORT.]

    [INSUFFICIENT STORAGE AND OVER SYSTEM LIMIT.]

    [INSUFFICIENT STORAGE AND NO HARDWARE SUPPORT.]

    [OVER SYSTEM LIMIT AND NO HARDWARE SUPPORT.]

© 2013 IBM Corporation

Externals: Abend and Abend reason codes

- ***New reason code for ABEND DC2 are as follows:***
  - **006A – Issued by IAXV8**
    - Bad range detected. Starting address of the range must begin on a region 3rd boundary

- ***New reasons to issue old reason codes for ABEND DC2 are as follows:***
  - **005D – Issued by IAXV3**
    - Not enough frames available for backing 1M **and/or 2G** pages

© 2013 IBM Corporation

Large Page Performance Results

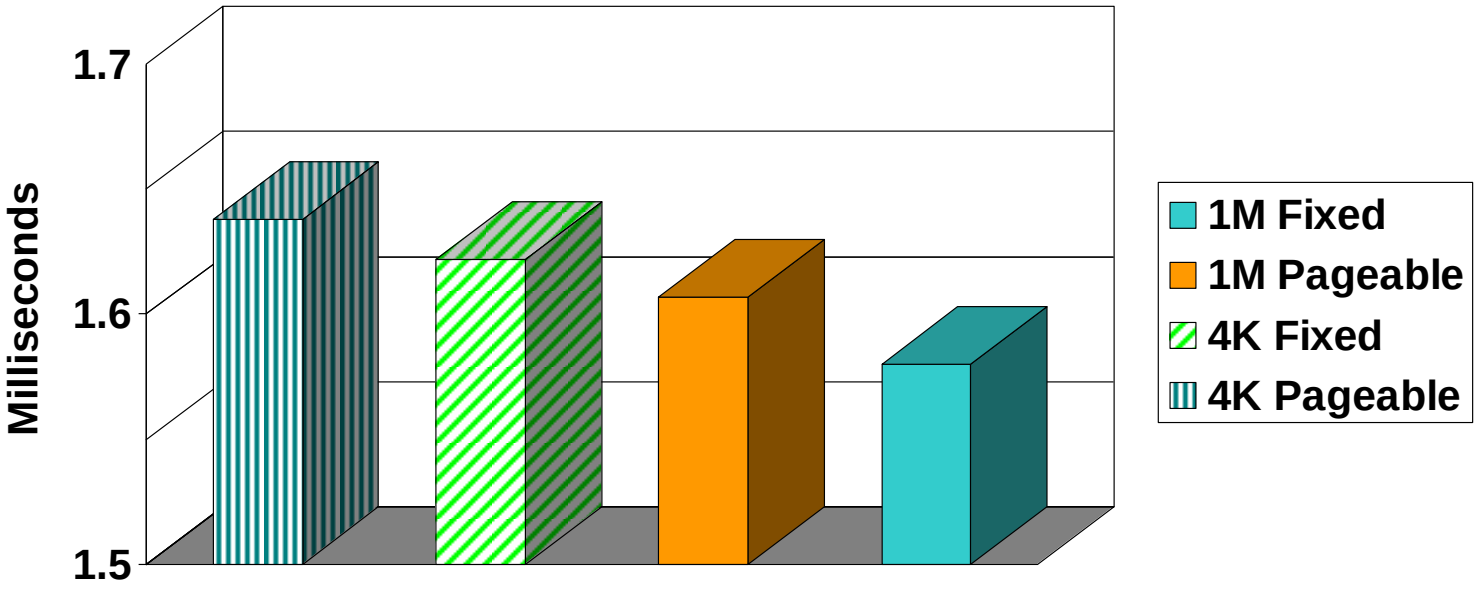
© 2013 IBM Corporation

Pageable 1MB Frames – Example from IBM Brokerage Workload

All of buffer pools are backed by real storage – DB2 10

- **zEC12 16 CPs, 5000-6000 tps (simple to complex transactions)**
  - 120GB real storage with 70GB LFAREA configured for 1MB measurements
- **1MB Pageable frames are 2% better than 4KB pageable frames for this workload**
  - 70GB buffer pools are used, 8-10 sync I/O per transaction
- **1MB frames with PageFixed is the best performer in general**

Total DB2 CPU Time per Transaction



© 2013 IBM Corporation

WAS benchmark: z/OS Performance for Pageable Large Pages

❖ **The WAS Day Trader benchmarks showed up to an 8% performance improvement using Flash Express.**

Java 7 SR3	JIT	Java Heap	Multi Threaded	WAS Day Trader 2.0
31 bit	yes	yes	4%	
64 bit	yes		1%	3%
64 bit		yes	4%	5%

* WAS Day Trader 64-bit Java 7 SR3 with JIT code cache & Java Heap

DETAILS

- *64-bit Java heap (1M fixed large pages (FLPs) or 1M Pageable (PLPs)) versus 4k pages*
  Java heap 1M PLPs improve performance by about
  - 4% for Multi-Threaded workload
  - 5 % for WAS Day Trader 2.0
- *64-bit Java 7 SR3 with JIT code cache 1M PLPs vs without Flash*
  - 3 % improvement for traditional WAS Day Trader 2.0*
  - 1 % improvement for Java Multi-Threaded workload
- *31-bit Java 7 SR3 with JIT code cache and Java heap 1M PLPs vs without Flash*
  - 4 % improvement for Java Multi-Threaded workload

** Note: This test used 64-bit Java 7 SR3 with JIT code cache & Java Heap leveraging Flash and pageable large pages.*
Also , tests used WAS Day Trader app that supports PLP; earlier version of 31-bit Java did not allocate 1M large pages

© 2013 IBM Corporation

z/OS Java SDK 7:16-Way Performance Shows up to 60% Improvement 64-bit Java Multi-threaded Benchmark on 16-Way

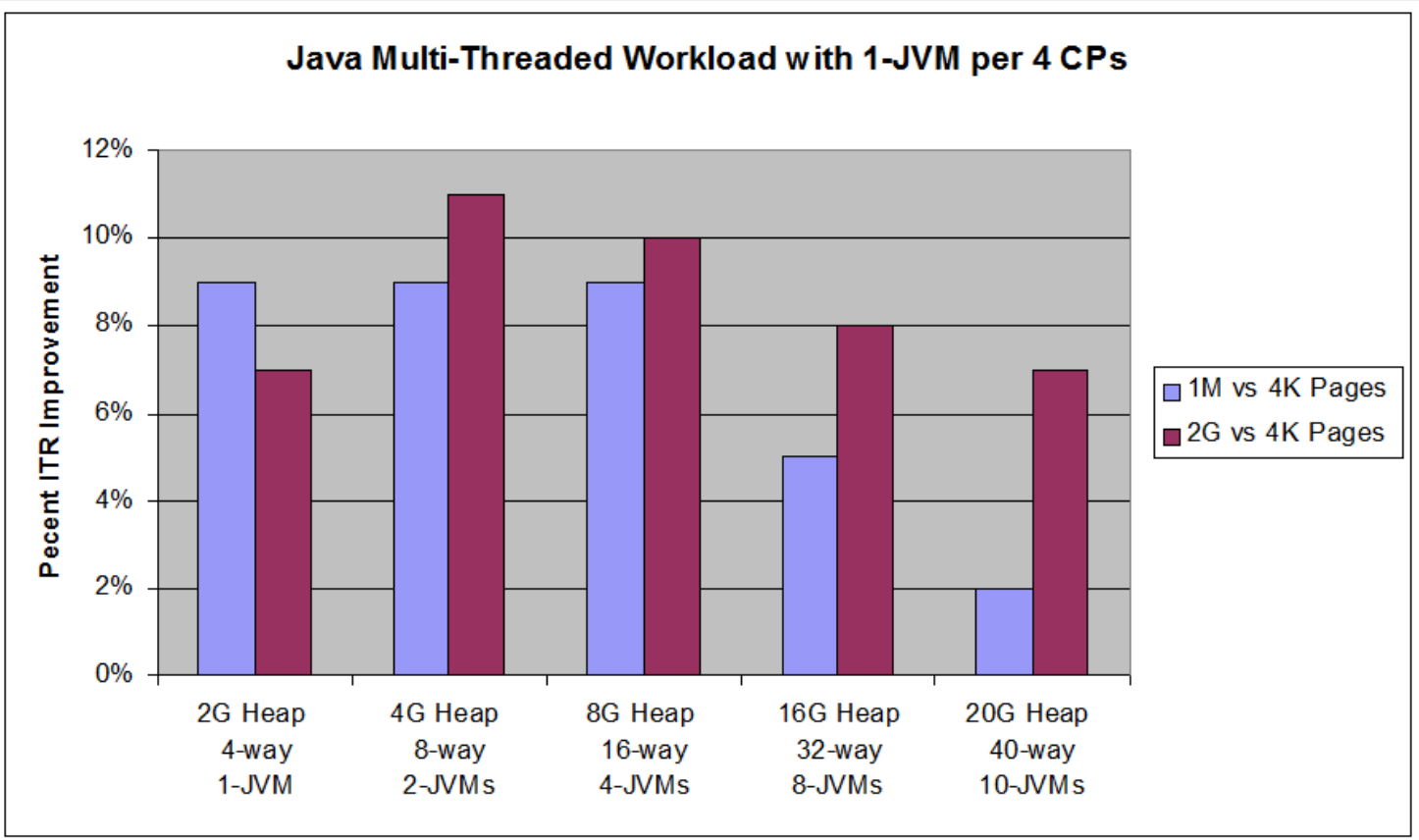
Aggregate 60% improvement from zEC12 and Java7SR3

- **zEC12 offers a ~45% improvement over z196 running the Java Multi-Threaded Benchmark**
- **Java7SR3 offers an additional ~13% improvement** (-Xaggressive + Flash Express pageable 1Meg large pages)

© 2013 IBM Corporation

z/OS Java SDK 7: 2G Page Performance

Multiple 4-way JVM Environment



2G large pages improve performance of multi-JVM environments with large aggregate footprint

(Controlled measurement environment, results may vary)

© 2013 IBM Corporation

Summary

- ***Focus on the design of high performance, scalable , reliable Large Memory***
- ***Large Memory has a large number of benefits including:***
  - **Improving user transaction response times and increasing overall throughput for OLTP workloads**
  - **Enabling faster real time data analysis for Analytic workloads by reducing the time it takes to get from raw data to business insight**
  - **Processing Big Data more efficiently by increasing the speed at which large amounts of data can be scanned**
  - **Simplifying the deployment of scalable applications within cloud infrastructures**

© 2013 IBM Corporation

Questions?

© 2013 IBM Corporation