IBM z Systems

TPFUG – Various Enhancements

Michael Shershin, TPF Development Lab

Date of Presentation

IBM

PJ42031 – z/Architecture Mode

PJ42031 – z/Architecture Mode

- Before PJ42031
  - z/TPF IPL started in ESA/390 architecture.
  - Switch is made to z/Architecture in IPLA.
- With PJ42031 – z/TPF no longer uses ESA/390 architecture
  - IPL changes include updates to IPL2 and IPLA
  - Software IPL changes
  - Standalone dump changes (sadump)
- Need PJ42534
  - ZIMAG PRIMARY updates to handle both old and new IPL2.
- Implementation of IPL2 requires IPL of a loader general file (LGF).
- IBM z13 announcement letter – dated January 14, 2015
  - Statements of General Direction (p.23) says:
    - **The IBM z13 will be the last z Systems server to support running an operating system in ESA/390 architecture mode**; all future systems will only support operating systems running in z/Architecture mode.

© 2015 IBM Corporation

PJ42299 – FARF6 short term pools

PJ42299 – FARF6 short term pools

- Problem: Number of allocated FARF5 (4-byte) file address are approaching capacity
  - FARF6 (8-byte) file addresses exist for long term pools
    - 4D6 – FARF6 4 K duplicated long term pools
  - Next largest use of file address are short term pools
- With PJ42299
  - Ability to allocate new short term pool type
    - 4S6 – FARF6 4 K short term pools
  - z/TPF now supports 11 pool types
    - FARF3/4/5 pool types: SST, SLT, SDP, LST, LLT, LDP, 4ST, 4LT, 4DP
    - FARF6 pool types: 4D6, 4S6
  - Controls for 4S6 pools are the same as other short term pool types
    - Set size (ZGFSP SET)
    - Ratio dispensing (ZGFSP RTO)
    - Recycle short term directory (ZGFSP RCY)
    - Recycle time (ZGFSP 4SR)
    - Fallback pool schedule (ZGFSP FLB)
    - Force reorder (ZPOOL FORCE REORDER)

© 2015 IBM Corporation

PJ42299 – FARF6 short term pools

- Secondary fallback for 4S6 pools is 4D6 (FARF6 4 K duplicated long term) pools
- Allocation of 4S6 pools can be done with Norm state pool reallocation
- Internal control records for short terms need to be transformed
  - Short term common control record (STCCR)
  - Short term processor control record (STPUR)
  - Transformation happens the first time that a pool reallocation is done using PUT 11 software
- Because 4S6 pools are 8-bytes, must use DECBs to retrieve a 4S6 pool.
- PI19018 provides TPFDF support for 4S6 pools
- PJ42694 is a follow-on APAR.
  - Updates to 1052 state pool reallocation to allocate 4S6 pools
  - Updates to ZGFSP DSP and ZGFSP RIV

© 2015 IBM Corporation

PJ42299 – FARF6 short term pools

```
==> ZDFPC
CSMP0097I 15.28.33 CPU-B SS-BSS  SSU-HPN  IS-01
DFPC0011I 15.28.33 07MAR DFPC  AVAILABLE FILE POOL COUNTS
                               FILE      CORE      ORD
SST DEVA                    109 450       394 00000008
    DEVB                    168 000       501 00000503 _
    TOT                     277 450       895
SDP DEVA                    229 848      2012 00000085
    DEVB                    249 846      1393 00000556
    TOT                     479 694      3405
LST DEVA                    129 030      1314 000000E6
    DEVB                     66 528      1056 000005FC _
    TOT                     195 558      2370
LDP DEVA                    425 552     13156 00000116
    DEVB                    352 201     10930 0000061B
    TOT                     777 753     24086
4ST DEVA                    157 800         0 00000290
    DEVB                     41 472         0 00000698 _
    TOT                     199 272         0
4S6 DEVA                     25 320       119 000009B2
    DEVB                     24 192         0 00000A01
    TOT                      49 512       119
4DP DEVA                  1 007 516      9724 0000043E
    DEVB                  2 523 981     11447 00000701 _
    TOT                   3 531 497     21171
4D6 DEVA                    198 663      2317 000008B5
    DEVB                    699 375      7129 00000901
    TOT                     898 038      9446
END OF DISPLAY+
```

© 2015 IBM Corporation

PJ42341 – Improve CORUC collision handling

PJ42341 – Improve CORUC collision handling

- CORHC (core hold) and CORUC (core unhold) are used to serial access to a resource
- Collisions happen when:
  - ECB 1 on IS-1 does a CORHC and becomes the holder of the resource
  - ECB 2 on IS-2 does a CORHC and waits. This is a collision.
  - ECB 1 on IS-1 does a CORUC to free the resource.
  - ECB 2 on IS-2 obtains the resource. The collision is resolved.
- Before PJ42341
  - On a collision time to dispatch an ECB that is now the owner can be significant.
  - Limits capacity for an individual resource.
  - When ECB 1 does the CORUC, dispatching of ECB 2 is scheduled using:
    - ECB 2 added to cross list for IS-2
    - When dispatched from the cross list, ECB 2 is added to the ready list for IS-2.
    - When dispatched from the ready list, ECB 2 starts processing

IBM
© 2015 IBM Corporation

PJ42341 – Improve CORUC collision handling

- With PJ42341
  - On a collision time to dispatch an ECB that is now the owner is reduced.
  - Increases capacity for an individual resource.
  - When ECB 1 does the CORUC, dispatching of ECB 2 is scheduled using:
    - ECB 2 added to cross list for IS-2
    - When dispatched from the cross list, ECB 2 starts processing.

© 2015 IBM Corporation

PJ42416 – Unplanned mod down

PJ42416 – Unplanned mod down

- Only affects unplanned mod down in loosely coupled complexes
- Addresses situation where a significant number of unplanned mod downs are happening simultaneously.
  - For example, an LSS is lost.
- Before PJ42416, serialization to update the keypoint 6 extension record can increase time required to process a large number of unplanned mod downs.
- With PJ42416, the ability to bundle unplanned mod downs is provided.
  - Bundling reduces time required to take a group of modules offline
  - Up to 64 mod downs can be bundled into one request.
  - Bundling is enabled with command: ➔ ZRSMD BUNDLE
    - One command enables bundling for all subsystems.
  - PJ42416 must be installed on all processors in the loosely coupled complex before enabling bundling.

© 2015 IBM Corporation

PJ42416 – Unplanned mod down

- High level bundling logic is:
  - Unplanned mod down happens – ECB is created to take down the mod
  - The keypoint 6 extension record (CK6E) is retrieved with hold (FIWHC equivalent)
    - Intent is to get ownership of unplanned mod down process
  - When CK6E is held, the MFST is searched to determine if more than one unplanned mod down is in progress.
    - Ownership of up to 64 unplanned mod downs will be assumed by this ECB
  - The first sync point is performed for the bundled group
    - First sync point sets write-only for the mods in the bundle on all processors in the complex
  - After the first sync point is complete, the following is done.
    - Module queues are purged for all mods in the bundle.
    - One IPC broadcast is sent to all processors in the complex to start mod down for all mods in the bundle.
  - Individual ECBs for each mod in the bundle are created and traditional unplanned mod down logic is executed.
    - Update MFST in memory to show module offline.
    - Lock movement
    - Update keypoint 6 to show module offline.
    - Update keypoint 6 extension record to show module as no longer in write-only mode.

© 2015 IBM Corporation

Disclaimer

- Any reference to future plans are for planning purposes only. IBM reserves the right to change those plans at its discretion. Any reliance on such a disclosure is solely at your own risk. IBM makes no commitment to provide additional information in the future.

© 2015 IBM Corporation

Proposed: 2 GB page support

2 GB page support

- Leverage hardware capability
  - Dynamic addressing translation - one region table entry covers 2 gig of memory
  - 2 GB page support provides hardware capability to not require page and segment table entries for 2 GB of real memory
- Performance improvement
  - One TLB entry for 2 gig of memory
- Automatically used on machines that support 2 GB pages
  - zEC13
  - zEC12
- Memory that will use 2 GB pages
  - VFA
  - ECB trace buffers

© 2015 IBM Corporation

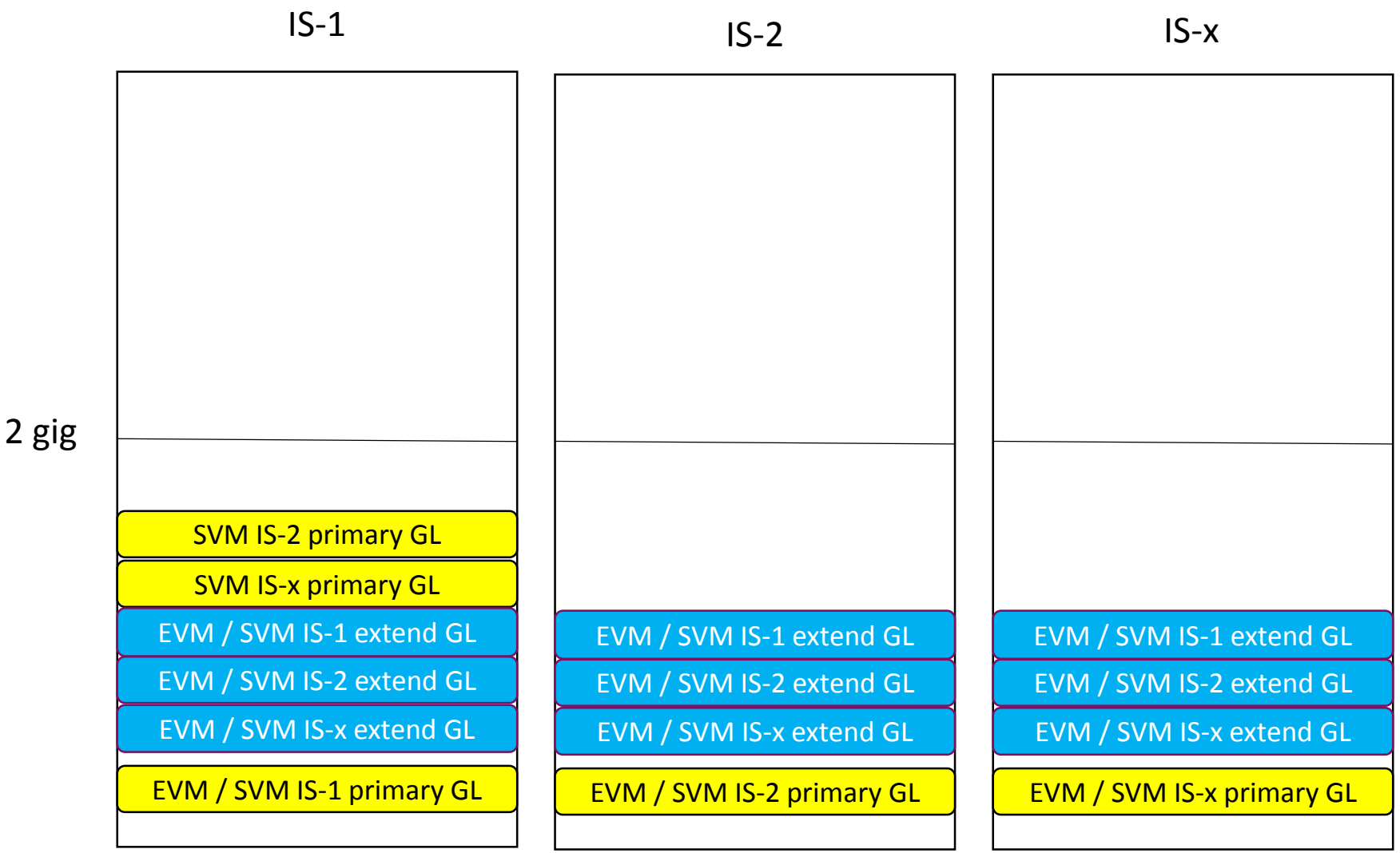
Proposed: Format 1 Globals Enhancements

Format 1 Globals

- Purpose is to allow large number of I-streams
  - Reduce memory usage below 2 gig for Format 1 I-stream unique globals
- Format 1 globals consists of:
  - Primary globals
    - Original global support.
    - In TPF 4.1 primary globals resided below the 16 meg line.
    - In z/TPF primary globals reside above 16 mg line but below 2 gig.
    - I-stream unique primary globals have same EVM addresses in all I-streams
  - Extended globals
    - Enhancement in TPF 4.1 that allowed globals to reside above the 16 meg line.
    - In z/TPF extended globals reside above 16 meg line but below 2 gig.
    - I-stream unique extended globals have different EVM address on each I-steam
- Proposed change:
  - I-stream unique extended globals to have same EVM address in all I-streams.
  - I-stream unique primary and extended globals on main I-stream
    - EVM and EVM addresses below 2 gig
  - I-stream unique primary and extended globals on application I-streams
    - EVM addresses below 2 gig (same virtual address as main I-stream)

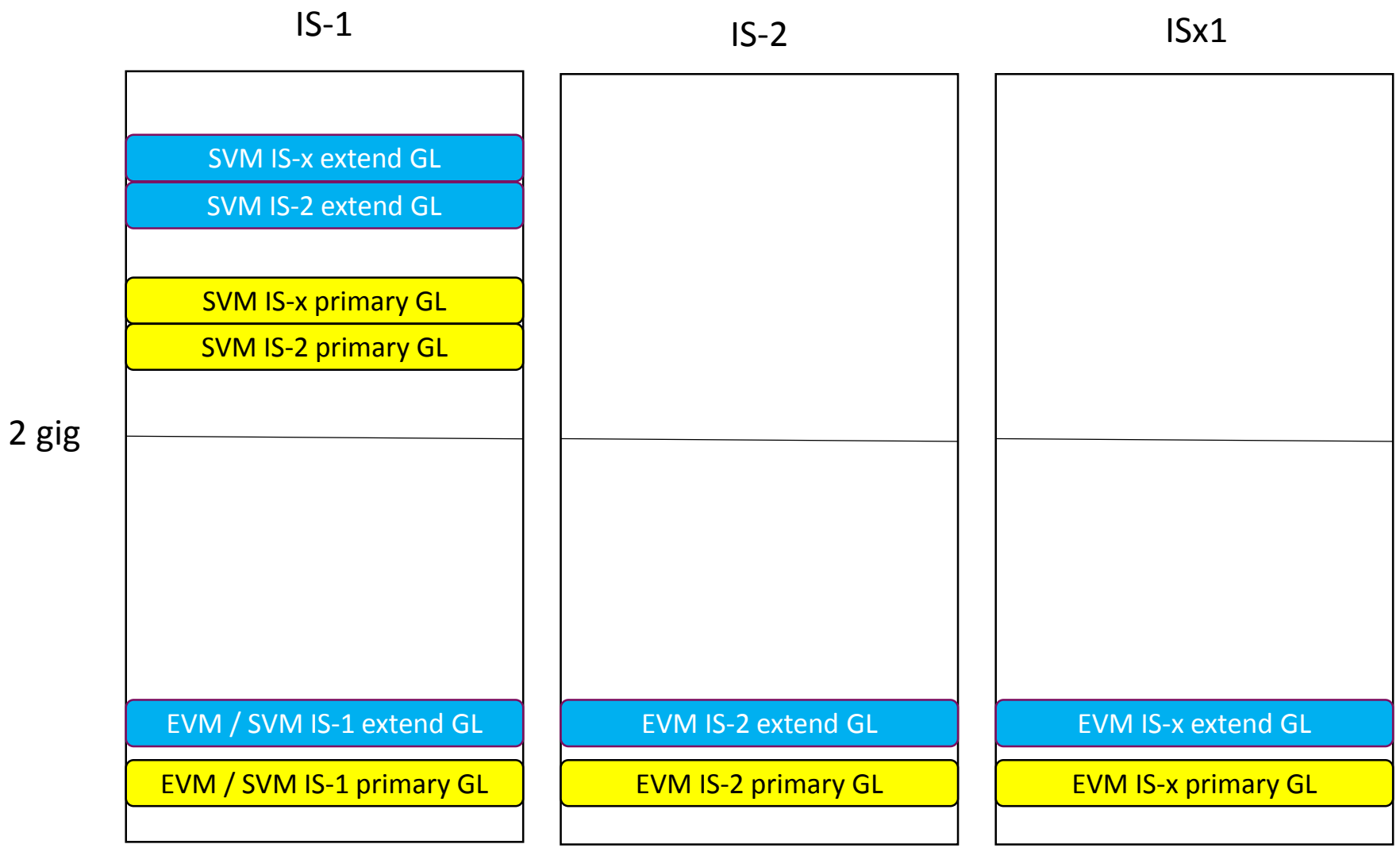
© 2015 IBM Corporation

Format 1 Globals - Current



© 2015 IBM Corporation

Format 1 Globals - Proposed



© 2015 IBM Corporation

Format 1 Globals

- No change to I-stream shared globals
- Intend to support both:
  - Current memory allocation below 2 gig
  - Proposed memory allocation above 4 gig
  - A switch will indicate the memory allocation to use.
    - Default will be the current memory allocation below 2 gig.
    - Switch will be processor unique.

© 2015 IBM Corporation

Proposed: Enhanced logical record cache

Logical Record Cache

- Processor cache
  - Uses memory in system heap
- Record sizes up to 4 K record are supported
- Create a cache in application
  - newCache() or tpf_newCache_ext()
  - Token is returned.
  - Token must be used on subsequent cache requests.
  - Application must save the token somewhere for subsequent reuse
- Multiple caches can be created.
- On IPL, a newcache() call must be made by the application to re-create the cache
- Other APIs to use cache include
  - readCache()
  - updateCacheEntry()
  - flushCache()
- Use of CF (loosely coupled only)
  - One processor updates a specific entry
  - CF can be used to invalidate entries on other processors in the loosely coupled complex.

© 2015 IBM Corporation

Enhanced logical record cache

- Allow objects greater than 4 K into logical record cache
- Allow objects of varying sizes
- Provide ability to manage a cache through commands
  - Create a cache
  - Change the size of the cache
- Provide ability to reference a cache name by name.
  - Do not need to save a token when the cache is created.
- On IPL, automatically create the cache.

© 2015 IBM Corporation

Trademarks

- IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml.
- *(Include any special attribution statements as required – see Trademark guidelines on https://w3-03.ibm.com/chq/legal/lis.nsf/lawdoc/5A84050DEC58FE31852576850074BB32?OpenDocument#Developing%20the%20Special%20Non-IBM%20Tr)*

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.
- 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.

© 2015 IBM Corporation