

IAM: Improving Performance for Transaction and Batch VSAM Applications

Richard Morse
Innovation Data Processing

February 6, 2013
Session Number 12675

www.Linkedin.com/pub/richard-morse/13/597/775/
EMAIL: rmorse@fdinnovation.com

IAM: Improving Performance for Transaction & Batch VSAM Applications



- What is IAM
- IAM Concepts
- IAM Performance Strategies
- IAM Performance Summary
- IAM/PLEX
- IAM Version 9.1 Enhancements

What is IAM?

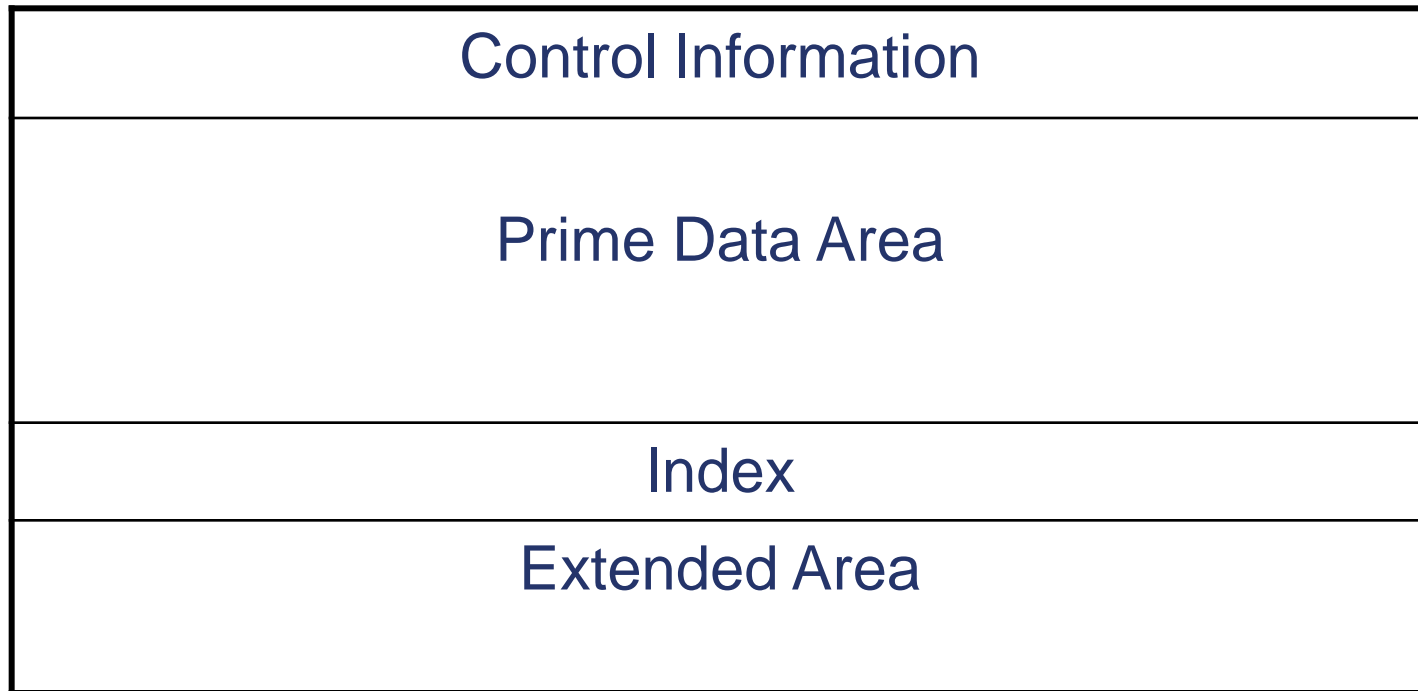
- Performance oriented access method
 - Well established for over 40 years
 - Continuously evolving to be responsive to customer
- An alternative to VSAM
 - Plug compatible VSAM API (Application Programming Interface)
 - Supports KSDS, ESDS, RRDS and AIX type datasets
 - Provides CPU time and I/O savings
 - Hardware or Software data compression techniques
 - Minimizing manual tuning
 - Does not replace VSAM
 - Selected for use at the dataset level

IAM Concepts

- Data stored in fixed length blocks
 - Not restricted to certain sizes
 - Maximizes space utilization of DASD architecture
- Resides on DASD in 1 of 3 formats:
 1. Non-VSAM dataset (DSORG=PS)
 2. DSNTYPE=LARGE Format
For >64K tracks per volume
 3. DFSMS Extended Format Sequential Dataset
For >64K tracks per volume
- Can Reside in the Extended Addressing Space (EAS) on Extended Address Volumes (EAV)

IAM Concepts

- IAM File Structure



IAM Concepts

- Transparent VSAM Interface
 - KSDS
 - ESDS
 - AIX – Alternate Index (Optional feature)
 - RRDS and Variable RRDS (Included with AIX feature)
 - IAM does not support Linear Data Sets (LDS)
- Data Compression
 - Software or System z Hardware compression
 - Can be used on all IAM supported dataset types: KSDS, ESDS, RRDS, Variable RRDS

IAM Performance Strategies

- Index in Virtual Storage
- Real Time Tuning Buffer Management
- Insert Strategy
 - Record Based Overflow
 - Prime Related Overflow
 - Prime Extension
- Data Compression
- Dynamic Tabling
- Overflow Caching

IAM Performance Strategies

- Index in Virtual
 - Read into storage during open
 - Eliminates index buffers and I/O
 - Compressed format to Reduce Size
 - Prime Data Index Compressed
 - Overflow index compressed
 - Use 64-bit virtual or z/OS Data Space storage
 - Prime Data index never changed after file load

IAM Performance Strategies

- Real Time Tuning
 - Dynamic buffer management based on application processing
 - LRU management of randomly processed blocks
 - Automatic deferred writes for batch updates
 - Immediate reuse of buffers with sequentially processed blocks
 - Sequential read ahead
 - Sequential multiple blocks read / written per physical I/O
 - In mixed random / sequential environments, dynamically balances buffer usage based on application I/O demands

IAM Performance Strategies

- Real Time Tuning (continued)
 - Dynamically adjusts number of buffers
 - Works within a range of minimum / maximum number of buffers
 - Periodically evaluates buffer usage, and adjusts as necessary
 - Provides indication if larger maximum would reduce I/O
 - Maximum buffer defaults (installation modifiable):
 - *32,768K buffer space for Batch / TSO*
 - *1,024K buffer space for CICS*
 - *Defaults should yield excellent performance for most datasets*
 - Increase maximum by using BUFND or BUFSP
 - Can use IAM Override facility to override buffering value
 - *MINBUFNO, MAXBUNO, BUFSP*
 - Turbo mode increases responsiveness and buffer limits

IAM Performance Strategies

- Real Time Tuning (continued)
 - Uses 31-bit virtual storage for all buffers
 - If a buffer is acquired in 24-bit storage, it will be released
 - Does not connect buffers to place holders (strings)
 - Eliminates CI lockout / exclusive control problems
 - May require less buffers than VSAM
 - Does not use VSAM LSR buffers (although IAM can be used with applications that have specified LSR buffering)
- Bottom line:
 - Simplified Tuning and Improved Performance
 - Typical results are a 30% to 80% reduction in elapsed time

IAM Performance Strategies

- Real Time Tuning: File Load Buffering
 - Essentially sequential output process
 - Defaults to obtaining enough buffers for 2 cylinders of data
 - Controlled by CRBUFOPT Override or Global Option
 - When 1/2 buffers are filled, issues EXCP to write that set of buffers
 - Application can concurrently fill up rest of buffers
 - Uses Data Space to hold index while writing data
 - Note: For SMS Extended Format datasets BSAM is used, so IAM does not have direct control on number blocks written per physical I/O

IAM Performance Strategies

- Insert Strategy: Record Based Overflow
 - Record placement based on space, not on key
 - Indexed by record in virtual storage
- Benefits:
 - Less I/O overhead than VSAM CI/CA splits
 - More efficient use of DASD space
 - Unused Overflow space has no restrictions on use
 - Works very well for most files
- Potential Problems
 - Storage required for index
 - Possible long time to open and/or read sequentially

IAM Performance Strategies

- Insert Strategy: Prime Related Overflow (PRO)
 - Record placement based on key by a block split technique
 - All records in block related to same Prime Data Block
 - Indexed by block, and is written to disk
- Benefits:
 - Reduces Overflow Index Size and Reorg Frequency
 - Improved Sequential Processing over Record based overflow
 - Works well on files with hundreds of thousands of inserts
 - No restrictions on reuse of empty overflow blocks
- Potential Problems
 - Will Use More DASD Space
 - Not as sharable without IAM/RLS or IAM/PLEX

IAM Performance Strategies

- Insert Strategy: Prime Extension
 - Records with Keys Beyond Current End of File
 - Records in Ascending Key Sequence
 - Indexed by block, written out to disk
- Benefits:
 - Less index storage required than overflow
 - Good sequential performance
 - Reduced need for Reorg
- Potential Problems
 - Restricted reuse of empty space to records within key range
 - Index is not compressed

IAM Performance Strategies

- Data Compression
 - Increases effective amount of data transfer per I/O
 - Reduces EXCP counts
 - Reduces data set size
 - IAM Software Compression
 - High performance, proprietary run length encoding algorithm
 - No dictionary required
 - Typical results are 20% to 50% compression
 - IAM use of System z Hardware Compression
 - Dictionary dynamically built during file load
 - Optional user provided customized dictionaries
 - Customized dictionaries may achieve > 90% compression

IAM Performance Strategies

- IAM's Dynamic Data Space
 - Record based cache in virtual storage
 - Used for randomly read records
 - May significantly reduce I/O and buffer needs
 - Records stored in segments, less unused storage for variable length records
 - Dynamic LRU management of records in table
 - Statistics provided in IAMINFO reports

IAM Performance Strategies

- IAM Overflow Area Cache
 - Block based cache area for record based overflow blocks
 - Entire overflow area at open time is cached
 - Cache is in 64-bit virtual storage
 - Usage is triggered by CACHE64 Override
- Benefits:
 - Intended usage is for files with large record based overflow
 - Inserts have been done in a very random manner
 - Improve sequential processing

IAM Performance Strategies

- IAM Run Time Reports: IAMINFO
 - One page statistical report on IAM file activity
 - Produced whenever an IAM file is closed
 - Requires IAMINFO DD card
 - Optionally can be written as SMF record
 - IAMINFO Report from provided IAMSMF program
 - Can be produced in CSV format for spread sheet use
 - Provides detailed information to assist with tuning
 - IAM368 Message if more buffers would have reduced I/O
 - IAM373 Message will tell you if file should be reorganized

IAM Performance Strategies

- Installation Selectable Defaults
 - Buffering
 - Data Compression
 - SMF Records
 - Data Space Size
 - Use of 64 bit Virtual and Index Space
 - Can be easily changed with provided program: IAMZAPOP

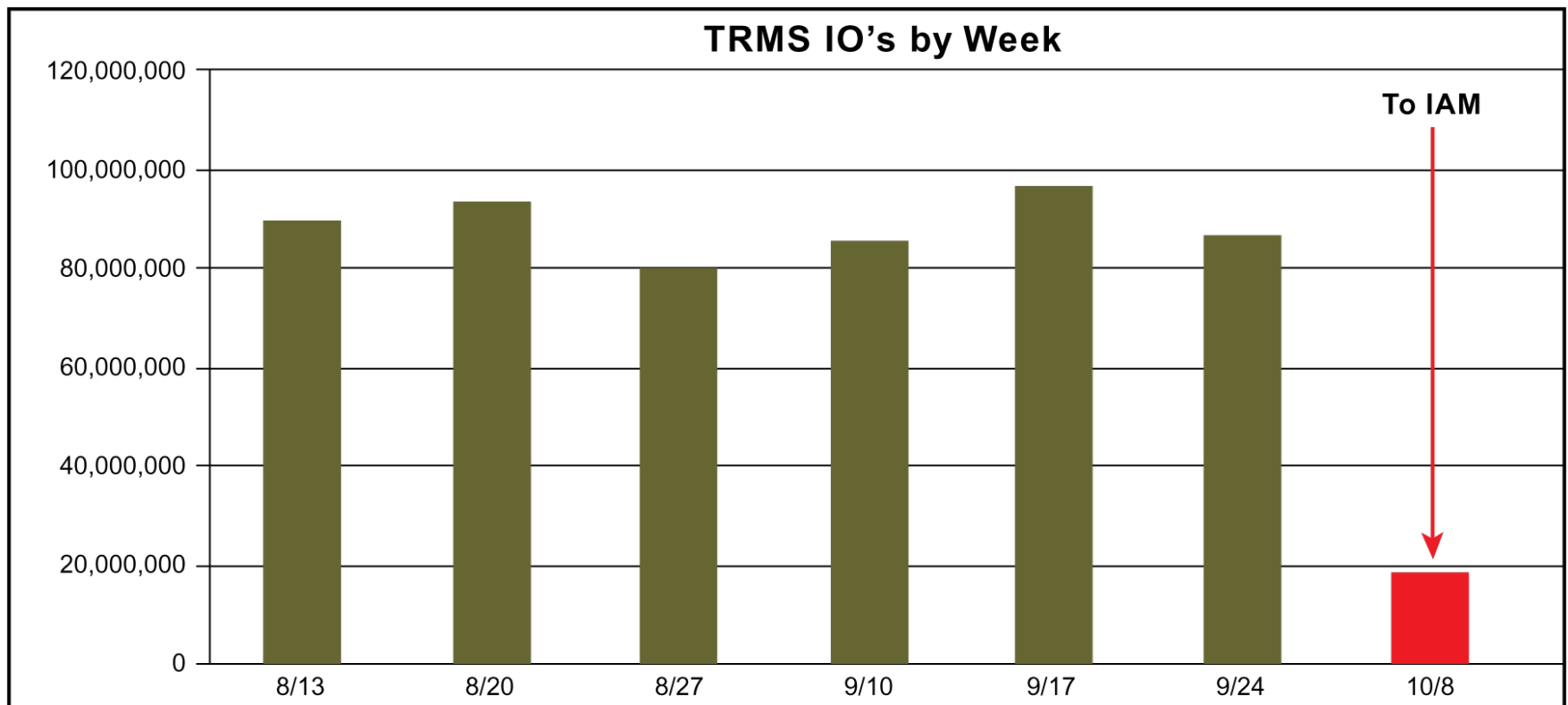
Performance Summary

- IAM Improves VSAM Application Performance
 - Index in virtual storage
 - Eliminates index component I/O and buffers
 - Dynamic Real Time Tuning
 - IAM dynamically selects best buffer management technique
 - IAM dynamically decides on number of buffers
 - Record based overflow
 - Eliminates I/O overhead of CI and CA splits
 - Data Compression
 - Increases effective data transfer per I/O
 - Reduces EXCP counts

Performance Summary

User Experience

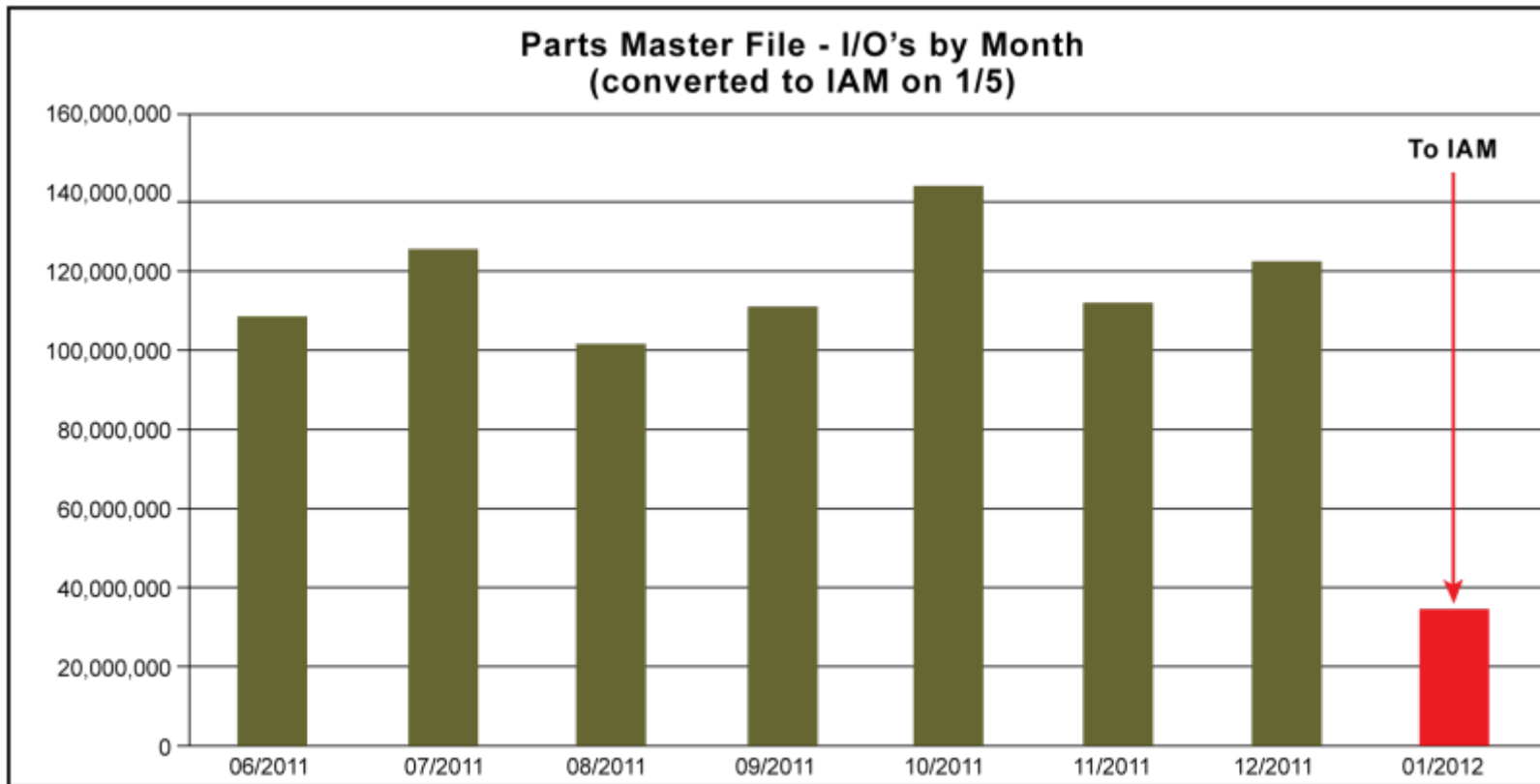
TRMS Database I/O Savings of About 79%



Performance Summary

User Experience

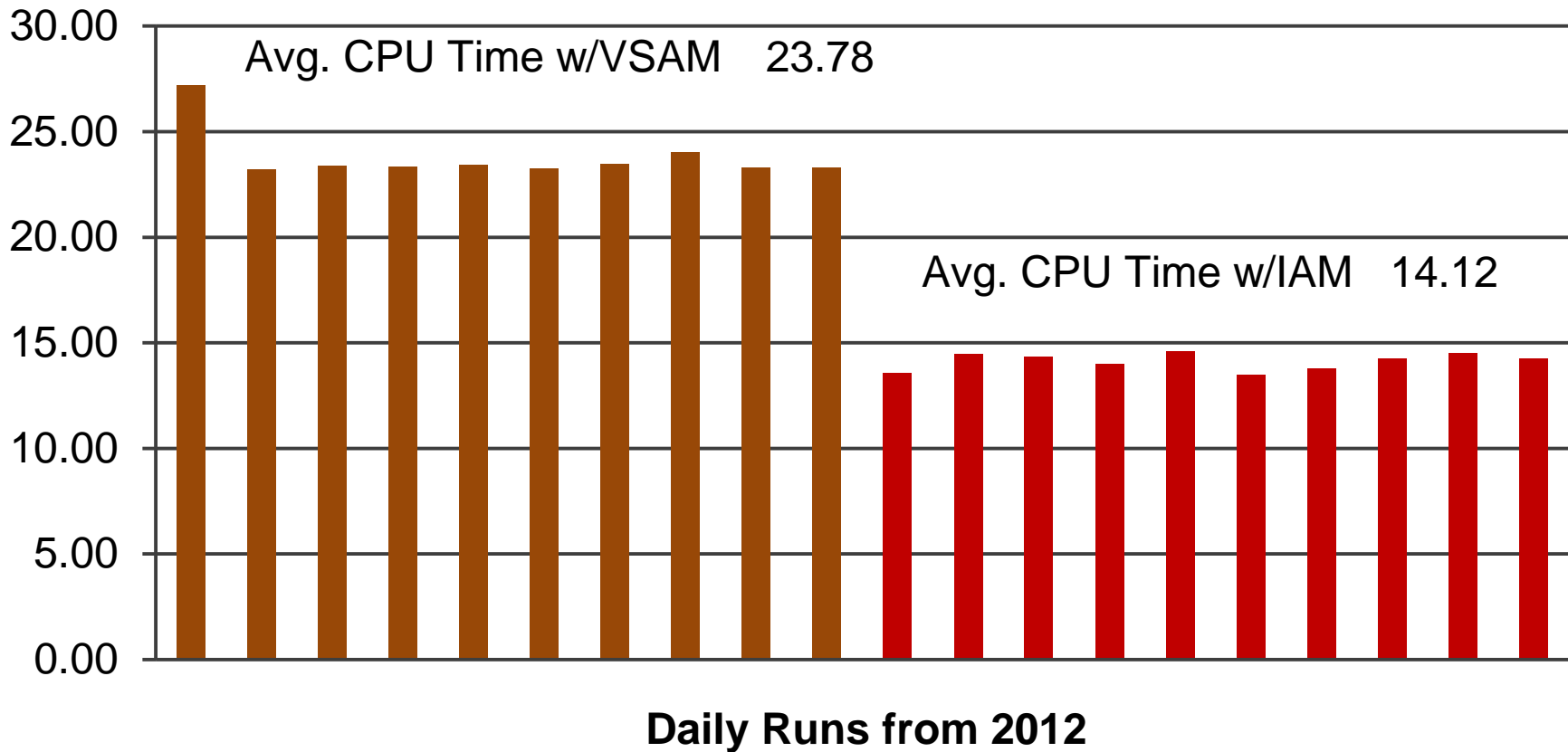
Parts Master File I/O Savings of 68%



Performance Summary

With a Lawson Application from INFOR

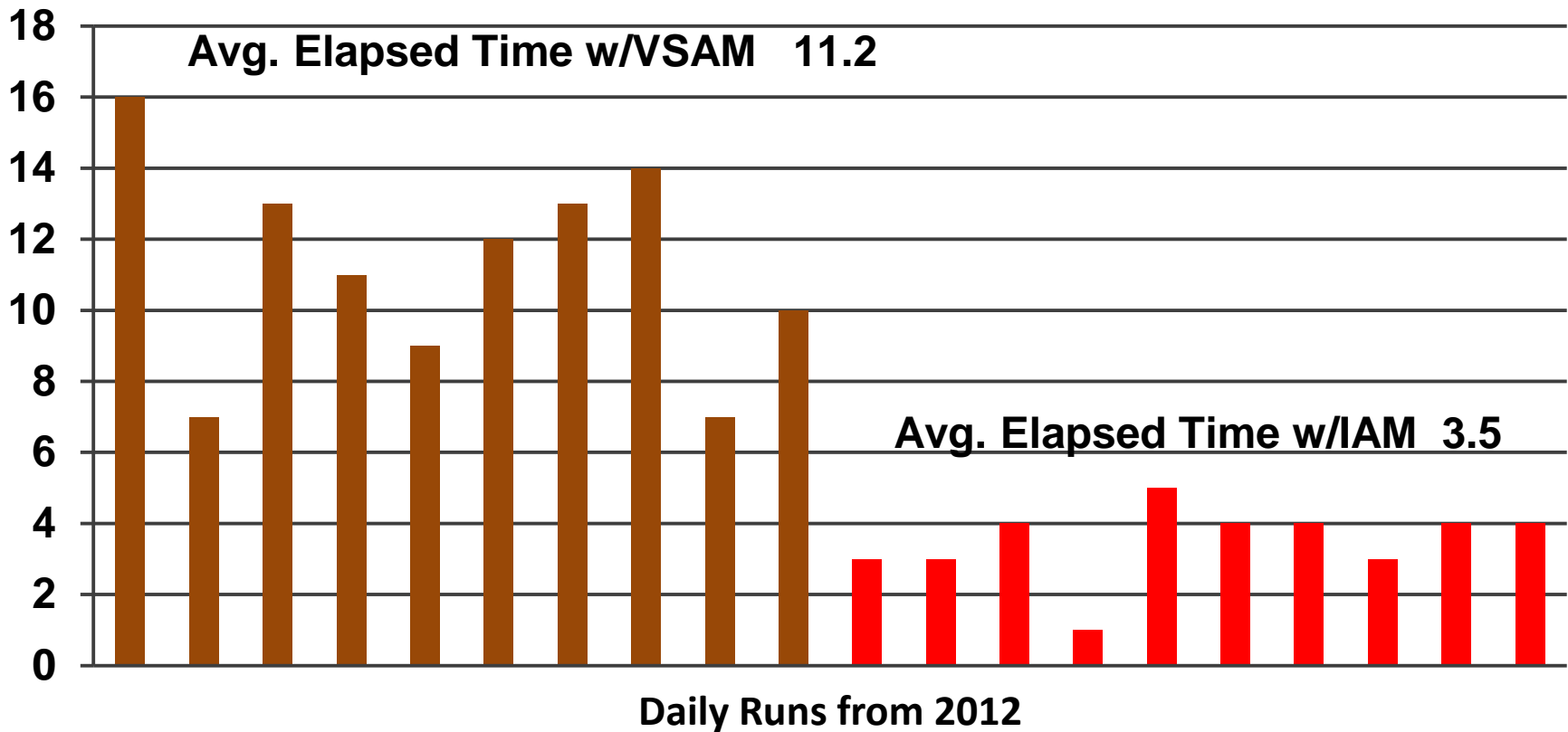
IAM Reduced CPU Time by 40.6%



Performance Summary

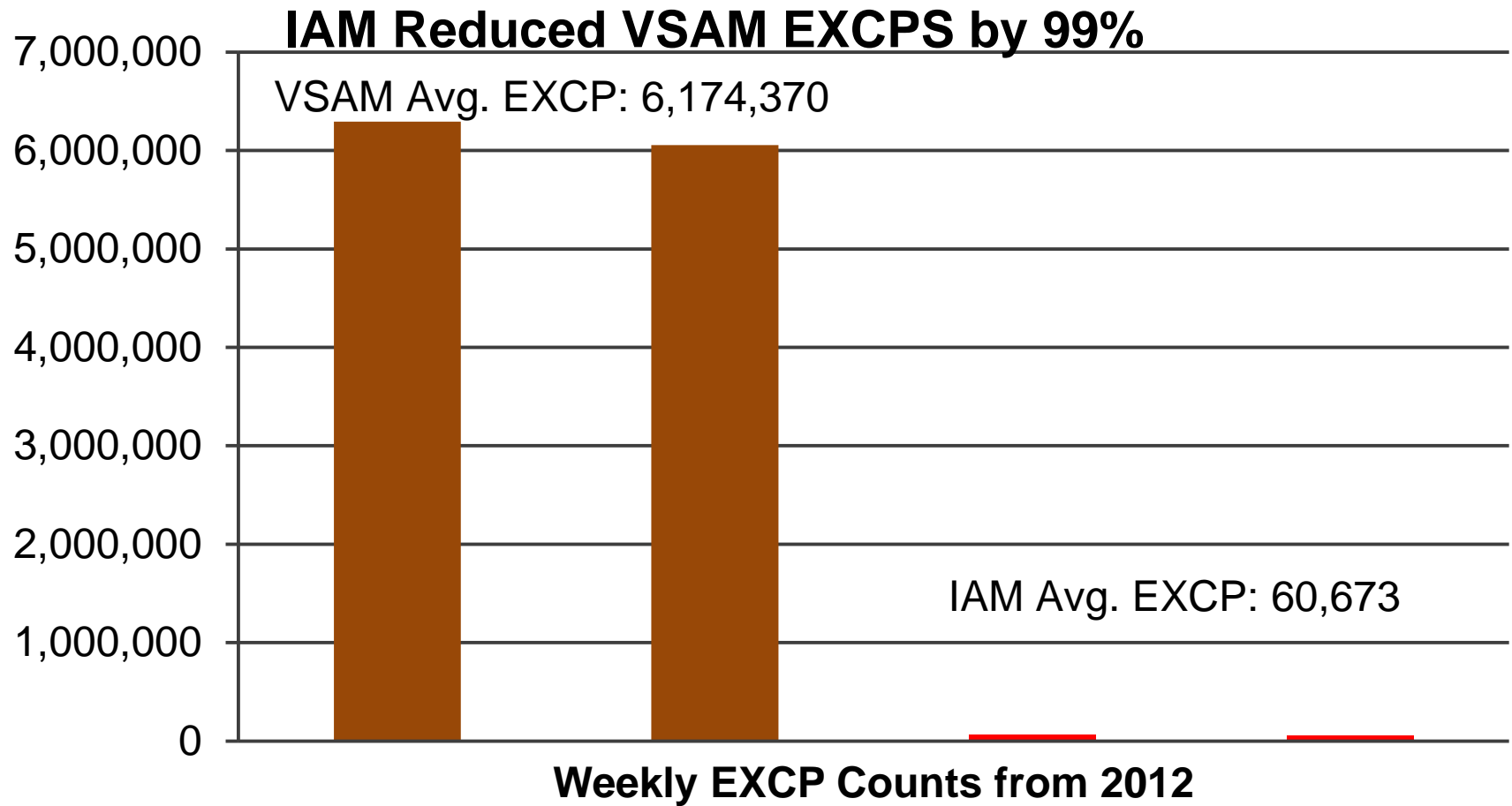
With a Lawson Application from INFOR

IAM Reduced Elapsed Time by 68.8%



Performance Summary

With a Lawson Application from INFOR



Performance Summary

Typical Results



- Reduces Batch Processing Time by 20% to 60%
- Reduces Physical I/O (EXCP's) by 40% to 80%
- Reduces CPU time by 20% to 40%
 - CPU savings may be reduced by Data Compression
- Data Compression Reduces DASD Space by 20% to 50%

Performance Summary

Can IAM Help?



- SMF Analysis Program
 - Determine how much VSAM I/O activity a system has
 - Do we have high enough VSAM I/O activity to justify IAM?
 - Identify Datasets that are Candidates for Conversion to IAM
 - Report on VSAM datasets with most I/O activity
 - Report on largest VSAM datasets
 - What datasets will yield the most savings from IAM?
- Available for Free
 - Includes Review and Analysis of Results at your installation

IAM/PLEX

Record Level Sharing

- Record Level Sharing for IAM Datasets
 - Multiple LPAR's in a SYSPLEX
 - Multiple concurrent CICS regions and batch jobs
 - Accessed by standard VSAM I/O requests
- Journal and Recovery Facilities Utilizing System Logger
- CICS Support
 - Must install IAM provided CICS exits
 - No CICS application program changes required
 - Looks like non-RLS VSAM file to CICS
- Batch applications
 - May need syncpoints for jobs that do heavy update activity

IAM/PLEX

Record Level Sharing

- Basic Record Level Sharing Truth:
 - It requires significantly more computing system resources to facilitate sharing than the resources actually needed to access and update the data.
- IAM/PLEX Design Objective:
 - Provide Record Level Sharing for IAM datasets with the best possible performance.
 - Concerned that additional utilization of Coupling Facility required to support the sharing of the IAM overflow structure would be detrimental to performance

IAM/PLEX

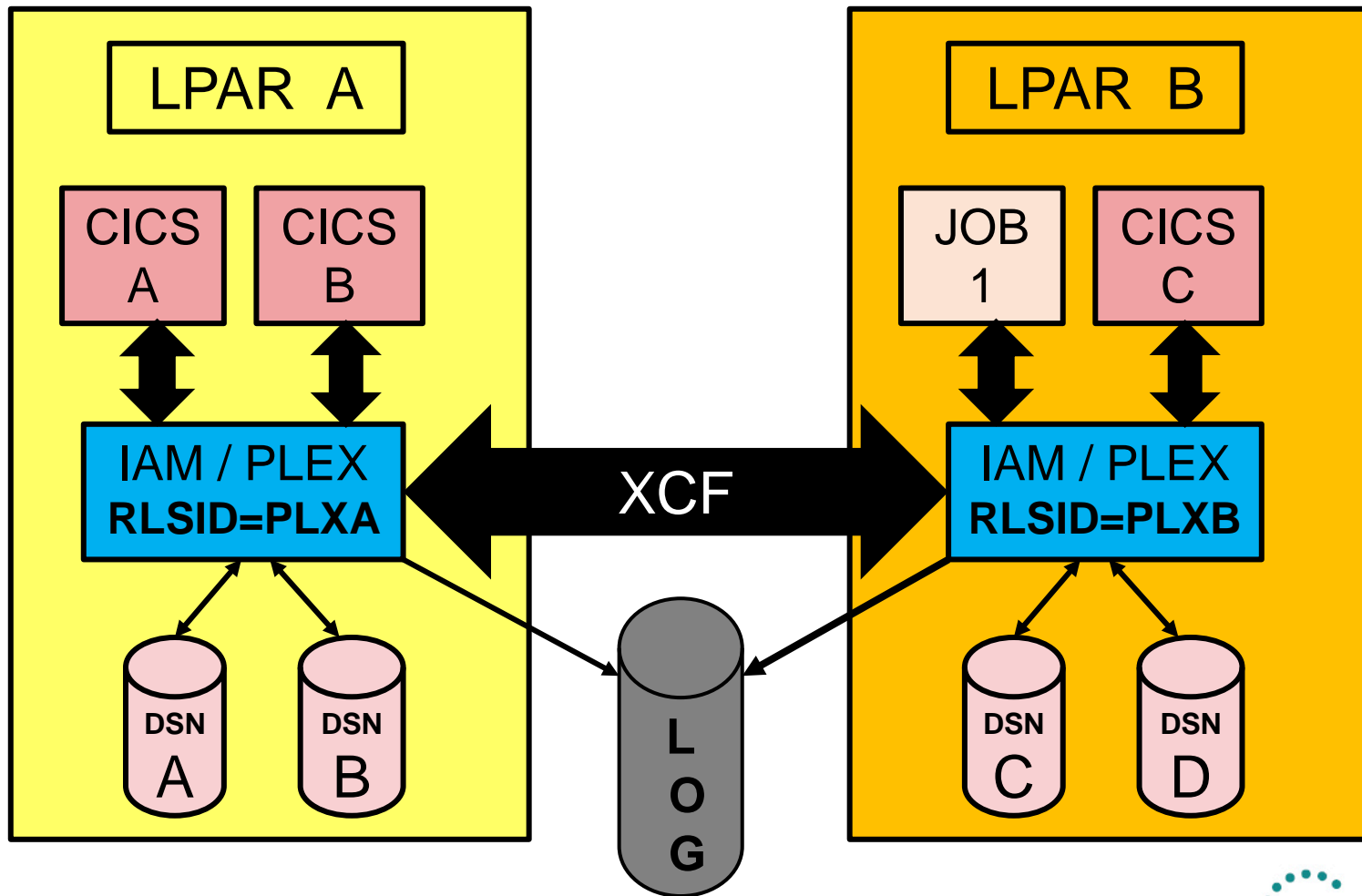
Record Level Sharing

- IAM/PLEX Design Conclusion:
 - IAM performance using the complete Coupling Facility Protocols would not perform at a satisfactory level.
 - IAM Sysplex sharing performance would be better using a single address space to do the I/O, record locking, and journaling rather than multiple calls to Coupling Facility
 - Utilize reliability of existing IAM/RLS structure to build Sysplex level sharing of IAM datasets rather than developing a lot of new processing to the IAM I/O process for necessary Coupling Facility Protocols

IAM/PLEX Concepts

- **RLSID:** A unique name for each IAM/PLEX address space running in a SYSPLEX.
- **RLSGROUP:** Refers to a group of related IAM/PLEX address spaces that form an XCF group that are able to directly communicate with each other.
 - Any particular application address space (CICS region, batch job) can be connected to only one RLSGROUP.
 - Can only access those IAM datasets that are being handled by an IAM/PLEX address space within that RLSGROUP.
- XCF services are used to communicate between IAM/PLEX address spaces within the RLSGROUP.

IAM/PLEX RLSGROUP=IAMPLEX1



IAM/PLEX and IAM/RLS Enhancements

- CICS Automated Recovery Processing
 - If an IAM/PLEX or IAM/RLS region becomes unavailable:
 - Close and disable affected files
 - Abend in-flight transactions causing them to be set up for backout when region is available
 - When IAM/PLEX or IAM/RLS region is available:
 - Open all of the IAM files that had been in use
 - Restart shunted transactions to do the recovery
- Persistent Record Locks
 - An IAM/PLEX address space will re-establish records locks for recoverable files that were held at time it became unavailable

IAM/PLEX and IAM/RLS Enhancements



- HiperDispatch performance enhancement
 - Utilize Pause/Release for faster synchronization
 - Enhanced internal work dispatching
 - Reduce number of I/O subtasks being used
 - Reduce utilization of z/OS local lock
 - Increase use of PLO instruction for serialization
 - Reduce frequency of various z/OS SVC calls
- Automated detection of recoverable files under CICS
 - Eliminates need for override for non-recoverable datasets
- CICS TS 5.1 Support

Additional IAM Version 9.1 Enhancements



- Uncaptured UCB Support
 - IAM files can be allocated and opened with 31-bit UCB
- Automatic support of VSAM editors under ISPF
 - File-Aid, File Manager, Ditto.....
- Automatic recatalog of IAM/AIX after Copy / Renames
 - Eliminate need for DEFINE RECATALOG job steps when name change is consistent amongst components
- GA: March 2013

IAM in Summary



- Transparently improves VSAM application performance
- Uses a simpler file structure, dynamic buffer management and caching
- Reduces physical I/O (EXCP's) by 40% to 80%
- Cuts CPU time by 20% to 40%
- Reduces elapsed processing times 20% to 60%
- Data Compression can save DASD space by 20% to 50%
- Provide Record Level Sharing across a SYSPLEX

QUESTIONS



Closing IAM: Improving Performance for Transaction and Batch VSAM Applications

Richard Morse
Innovation Data Processing

February 6, 2013
Session Number 12675

www.Linkedin.com/pub/richard-morse/13/597/775/
EMAIL: rmorse@fdriinnovation.com

