IAM V8.1 Enhancements Further Improve VSAM Application Performance

Richard Morse
Innovation Data Processing

August 14, 2006
Session 3047

IAM V8.1 Overview

- What is IAM?
- Unique Features of IAM
- IAM V8.1 Enhancements
- IAM V8.1 Performance
- Future Plans
- Summary

What is IAM?

- High Performance Access Method
  - Well established for over 30 years
  - Continuously evolving to be responsive to customer requirements
- An alternative to VSAM
  - Plug compatible VSAM Interface
  - Supports KSDS, ESDS, RRDS and AIX type datasets
  - Provides CPU time and I/O savings
  - Hardware or Software data compression techniques
  - Minimizes manual tuning
  - Does not replace VSAM
  - Selected for use at dataset level

Unique Features of IAM

- ESDS Over 4 Gigabytes
  - PSEUDORBA Feature
  - Allows use by programs limited to 4-byte RBA
  - Can be used for ESDS datasets under CICS
  - RBA value is relative record number
  - Does not require SMS Extended Format data sets
  - XESDS for 8-byte RBA support
- No CI Lockout / Exclusive Control
  - IAM internally locks at record level
  - No CI or block level lockouts
  - Improved performance for online updates
  - Allows for larger block sizes than VSAM CI Size

Unique Features of IAM

- Data Compression
  - Software or z/OS Hardware compression
  - Supported on all IAM datasets: KSDS, ESDS, RRDS, AIX
- Real Time Tuning
  - Dynamic Buffer Management
  - Minimizes need for manual tuning
- IAM Record Level Sharing
  - Single system
  - Sysplex in development
- Index in Virtual Storage
  - No Index Component I/O or Buffers

IAM V8.1 Enhancements

- Dynamic Data Space
- Hardware Compression Enhancements
- Turbo Buffering Mode
- Prime Related Overflow
- Additional Enhancements
  - Large Format Sequential Data Sets
  - Multiple IAMRLS Address Spaces
  - Prime Related Overflow
  - Plus more….

IAM V8.1 Enhancement
Dynamic Data Space

- Cache for randomly accessed records
- Enhancement based on DYNCORE feature
- Up to 2 gigabyte size per open IAM file
- Least Recently Used managed removal of old records
- Improved memory management
- Improved search algorithm
- Specified by IAM ACCESS Override of DYNDS=nnnn (meg)
- Benefit is I/O reductions

IAM V8.1 Enhancement
Hardware Compression

- Automatic Dictionary Build during File Load
- Objectives:
  - Build dictionary with minimal CPU time
  - Amount of compression generally comparable with SMS-VSAM hardware and IAM software compression
  - Can be set as default compression algorithm
- Uses up to first 8 megabytes of data written
- Dictionary is stored within IAM dataset
- Alternatively use dictionaries created by CSRBDICT

IAM V8.1 Enhancement
Turbo Mode Buffering

- Turbo Mode Enhances IAM's Real Time Tuning
  - Is the default as IAM V8.1 is shipped, but can be turned off
- Basic Real Time Tuning
  - Dynamic adjustment of number of buffers within a range
  - Dynamic management of buffers based on access patterns
- Turbo Mode Objectives:
  - Improve performance with better I/O and buffer management
  - Provide additional I/O reductions
  - Improve sequential update processing
  - Reduce or eliminate manual tuning
  - Greater responsiveness to application I/O demands
  - Allow larger number of buffers to be used

IAM V8.1 Enhancement
Turbo Mode Buffering

- Aggressively increase number of buffers
  - For first I/Os after open or during heavy I/O periods
  - Compensates for VSAM with large numbers of buffers
  - Increased frequency of buffer evaluation until at 80% of MAXBUFNO
  - Eliminate wait for buffer availability if not at maximum buffers
  - Obtain multiple buffers at a time
    - Reduces CPU time for MVS storage management
- Increase default maximum buffer space
  - BUFSP=65536 (K) up from 875K
  - CICSBUFSP=1024 (K) up from 256K
- Increased maximum number of buffers from 2048 to 8192

IAM V8.1 Enhancement
Turbo Mode Buffering

- Revised initial buffer quantity obtained when opened
  - Batch: Enough buffers for up to 16 tracks
  - Online: Enough buffers for up to 1 cylinder (15 tracks)
  - Improved Open performance for large datasets
- Write multiple non-contiguous blocks per EXCP
- Write multiple blocks per EXCP to Extended Areas
- New Override of BUFSPACE
  - Specifies maximum amount of storage to use for buffers
  - Specified on a dataset basis
  - Specification is not sensitive to block size being used

IAM V8.1 Enhancement
Prime Related Overflow

- New Optional Overflow Structure
- Records in an Overflow Block will be Related to Same Prime Block
- Reduces Overflow Index Size
  - Indexed at block level rather than record level
- Use with IAMRLS or Share Option 1 ONLY!
- Will Use More DASD Space

Additional IAM V8.1 Enhancements

- Support for Large Format Sequential Datasets
  - Requires z/OS 1.7 or above
  - Supports > 64k tracks per volume
  - Does not need to be on an SMS managed volume
- Support for Multiple IAMRLS address spaces per LPAR
  - Identified and selected by RLSID parameter
- Initial use of 64-bit Virtual Storage
  - Used as cache for extended overflow blocks
  - May improve performance of sequential processing
  - Be aware of potential paging impact

Additional IAM V8.1 Enhancements

- Performance Improvement for AIX under IAMRLS
  - I/O at sphere level passed to IAM RLS, rather than dataset level
- Reduce Dynamic Allocations with AIX and IAMRLS
  - Only allocate a dataset once, not for each concurrent use
- Support for SuperC functions under ISPF on IAM files
- Identification of IAM files by DFSMS ACS Routines
  - Look for DDNAME=@#$IAM

IAM V8.1 Performance

- Tested against VSAM with System Managed Buffering
- Objectives:
  - Demonstrate IAM V8.1 Turbo Buffering Performance
  - Answer question: Is IAM better performer than VSAM SMB?
  - Approximate user experience converting from VSAM
  - Minimize manual tuning performed
  - Set up to allow maximum automatic buffers
    - REGION=0M for VSAM SMB resulted in 100+ megabytes for buffers
    - IAM Global Option BUFSP set to 128 Megabytes maximum
  - Tested with and without Hardware Data Compression
- Benchmark based on user specifications

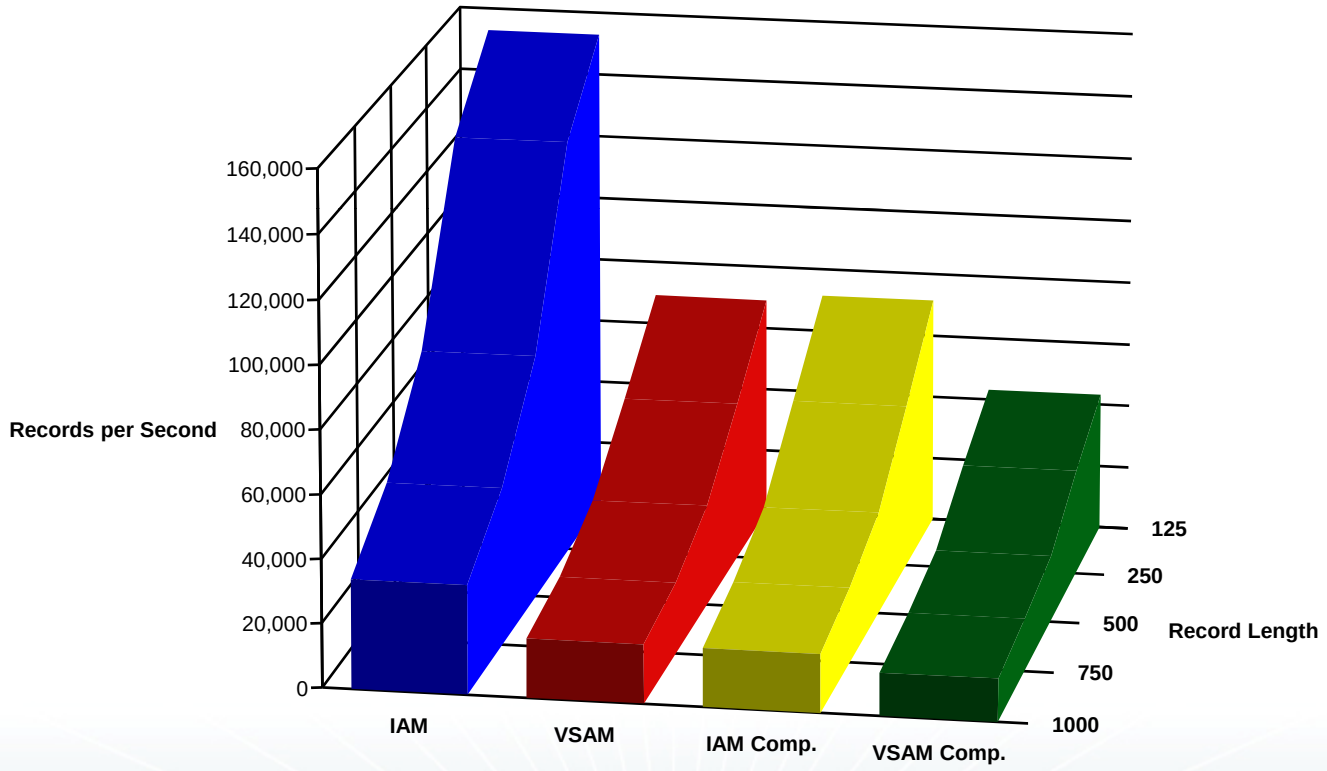
IAM V8.1 Performance Benchmark Description

- Files contained 2,000,000 records each
- CI Size of 8192
- Key Length of 8
- Five Tests with different fixed record lengths:
  - 125 bytes
  - 250 bytes
  - 500 bytes
  - 750 bytes
  - 1,000 bytes

IAM V8.1 vs. VSAM
File Load Performance

- Test Description:
  - Load file with 2,000,000 Records
- Results without Compression
  - IAM used 39% to 41% less CPU time
  - IAM ran in 46% to 62% less elapsed time
  - IAM used 85% less EXCP's
  - IAM used 8% to 17% less DASD space
- Results with Compression
  - IAM used 1% to 35% less CPU time
  - IAM ran in 27% to 40% less elapsed time
  - IAM used 87% to 91% less EXCP's
  - IAM used 25% to 45% less DASD space

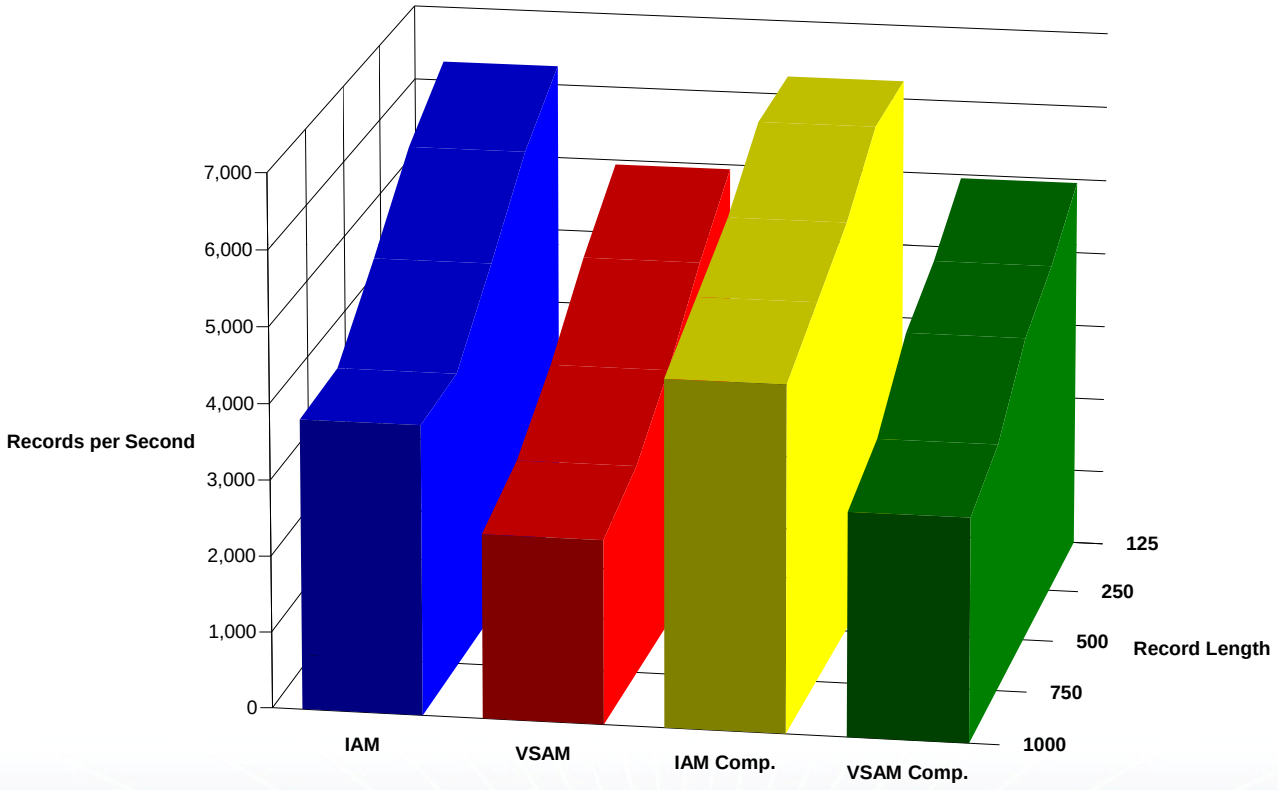
IAM V8.1 vs. VSAM
File Load Performance



IAM V8.1 vs. VSAM
Random Read Performance

- Test Description
  - 200,000 Random Reads
  - Spread throughout 10% of file
  - Both IAM and VSAM required additional buffers for large record sizes
- IAM used 20% to 27% less CPU time
- IAM used 21% to 36% less elapsed time
- IAM used 39% to 61% less EXCP's

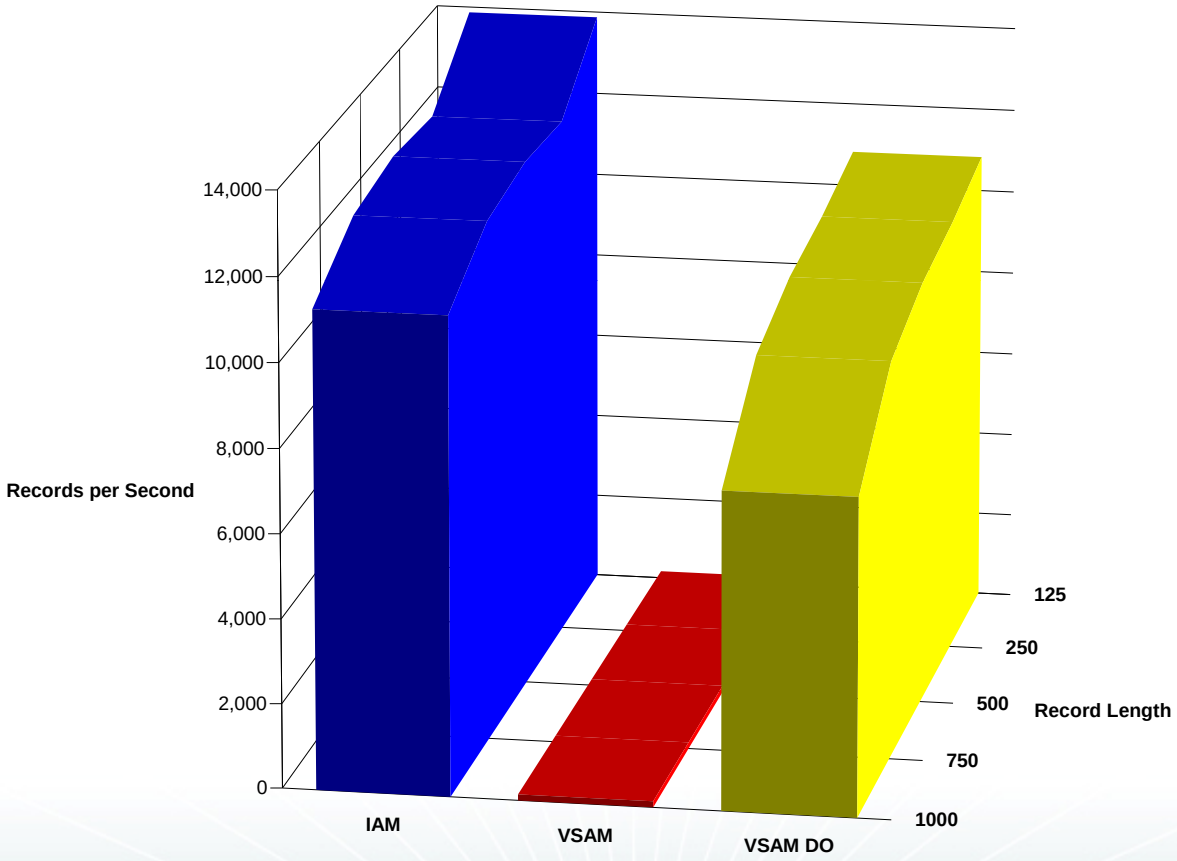
IAM V8.1 vs. VSAM
Random Read Performance



IAM V8.1 vs. VSAM
Access is Dynamic Performance

- Test Description
  - 100,000 Random Reads
  - Covered 1% of the file (20,000 records)
- IAM vs. VSAM
  - IAM used 92% to 93% LESS CPU Time
  - IAM ran in 99% LESS Elapsed Time
  - IAM required 99.9% LESS EXCP's
- IAM vs. VSAM w/ACCBIAS=DO
  - IAM used 22% to 25% less CPU time
  - IAM used 40% to 53% less EXCP's
  - IAM ran in 16% to 33% less elapsed time

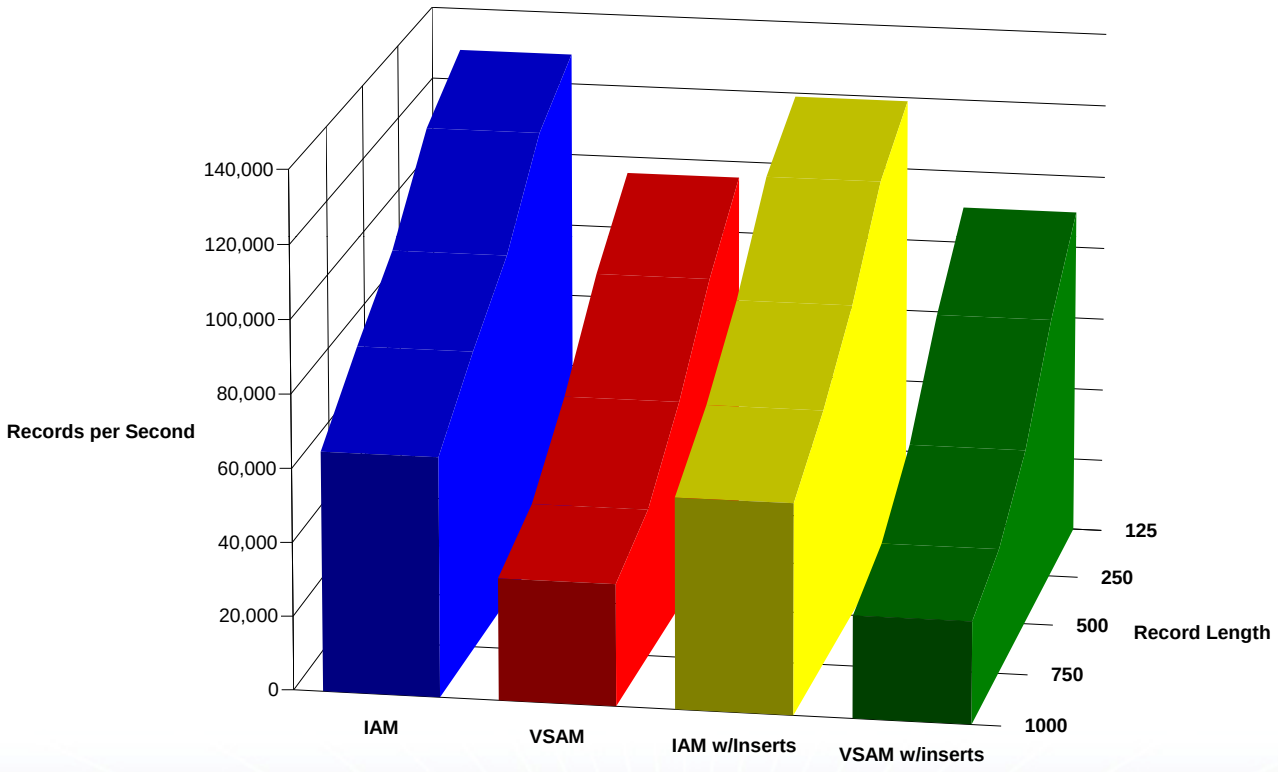
IAM V8.1 vs. VSAM
Access is Dynamic Performance



IAM V8.1 vs. VSAM
Sequential Read Performance

- Test Description
  - Sequentially Read Entire File (2,000,000 Records)
  - Tests With and Without Inserted Records
- IAM used 14% to 20% less CPU time
- IAM ran in 25% to 51% less elapsed time
- IAM used 67% to 79% less EXCP's

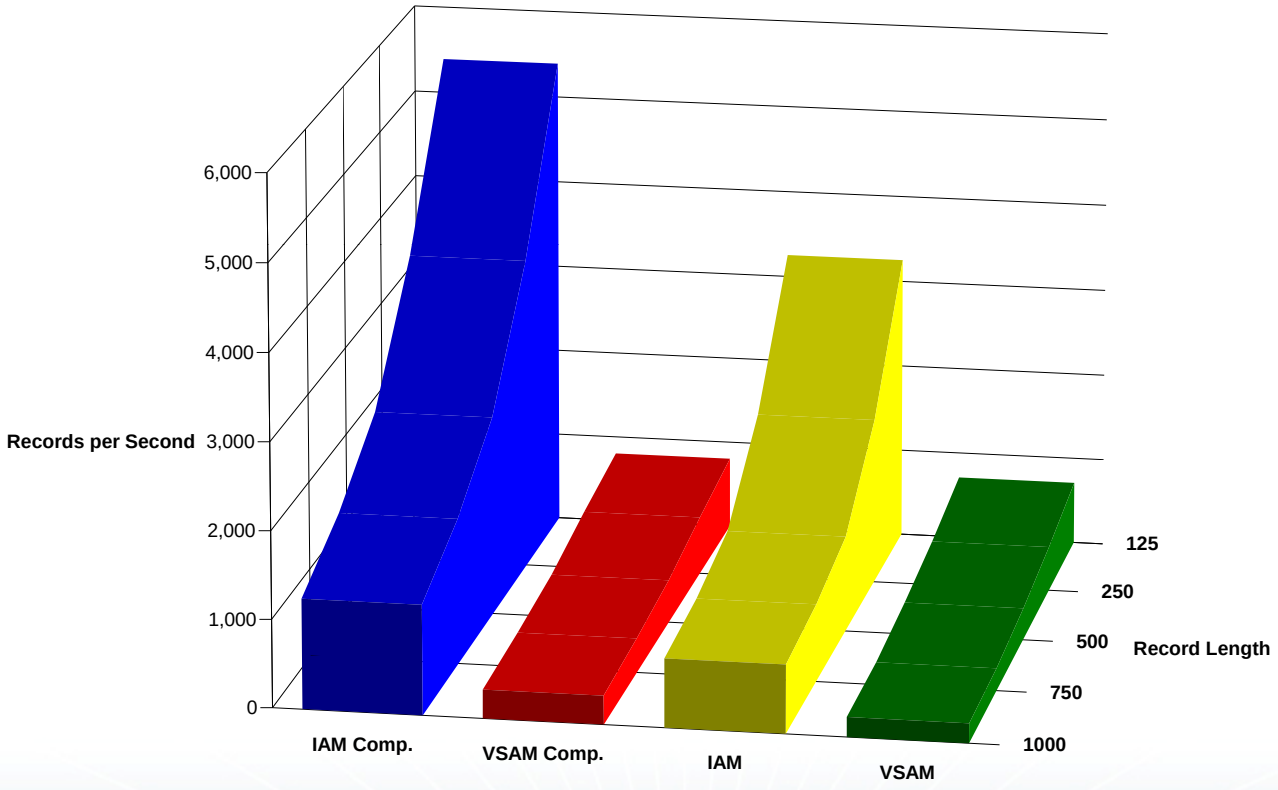
IAM V8.1 vs. VSAM
Sequential Read Performance



IAM V8.1 vs. VSAM
Random Insert Performance

- Test Description:
  - Randomly Inserted 38,003 Records
  - Within a key range of 400,000 Records
  - 9.5 % Inserted Records Within Above Range
  - Files defined with 10% CI Free Space
- IAM used 66% to 72% less CPU time
- IAM ran in 67% to 85% less elapsed time
- IAM used 72% to 92% less EXCP's
- For uncompressed data, IAM used 26% to 32% less space

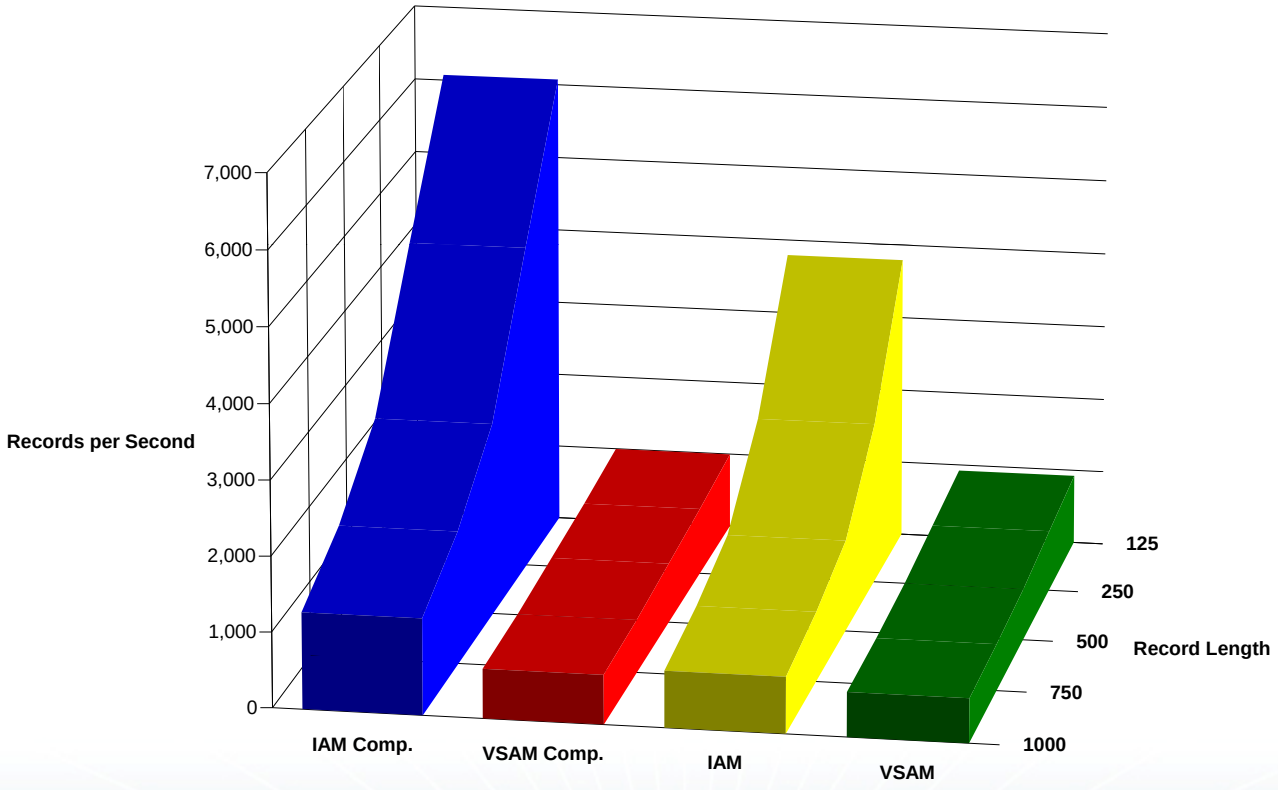
IAM V8.1 vs. VSAM
Random Insert Performance



IAM V8.1 vs. VSAM
Random Delete Performance

- Test Description:
  - Performed 40,000 Random Reads
  - Deleted 38,003 Records
  - Within key range of 400,000 Records
  - Received 1,997 Records Not Found
- IAM used 32% to 65% less CPU time
- IAM ran in 21% to 84% less elapsed time
- IAM used 31% to 92% less EXCP's

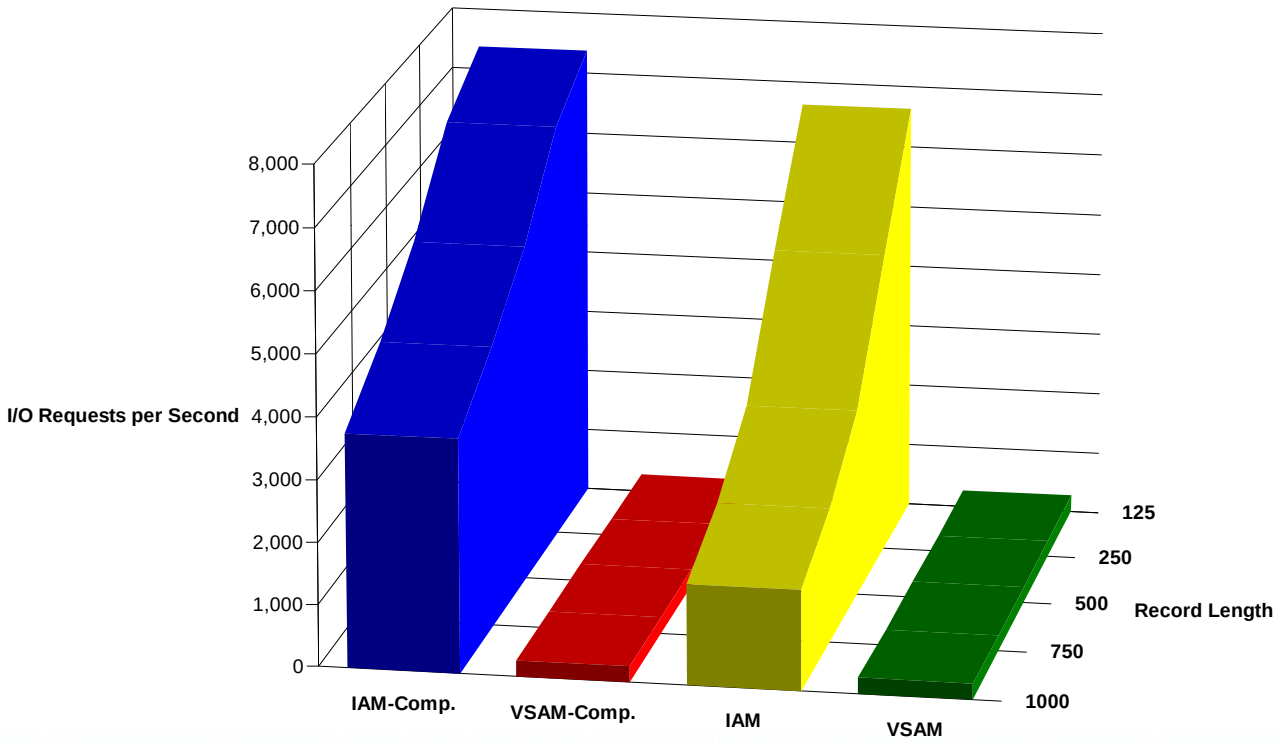
IAM V8.1 vs. VSAM
Random Delete Performance



IAM V8.1 vs. VSAM
AIX Processing Performance

- Test Description:
  - 256,000, Random Read
  - 160,000 Updates, 1/2 of which required an update to the other alternate index.
  - 80,000 Inserts
  - 16,000 Deletes.
  - 16,001 Points (Start browses) with 80,000 sequential reads. (5 records read per start browse).
- IAM used 75% to 85% less CPU time
- IAM ran in 84% to 96% less elapsed time
- IAM used 92% to 99% less EXCP's

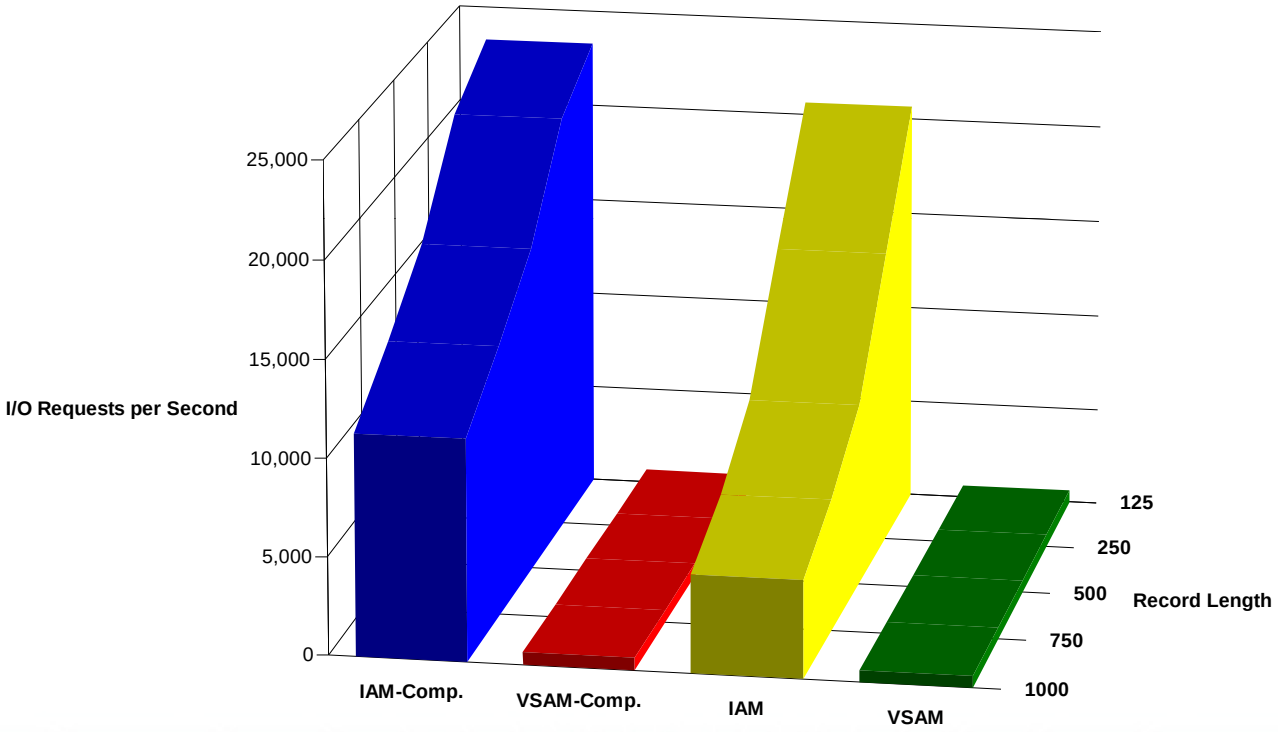
IAM V8.1 vs. VSAM
AIX Processing Performance



IAM V8.1 vs. VSAM
AIX Non-Unique Key Sequential Performance

- Test Description:
  - Sequentially read entire file through non-unique key AIX
  - Ideal VSAM buffering would be NSR for AIX and LSR for Base
  - VSAM must use same buffering technique for both components
  - SMB always uses NSR buffering for AIX Spheres
  - IAM's Real Time Tuning automatically handles this
- IAM used 80% to 86% less CPU time
- IAM ran in 88% to 97% less elapsed time
- IAM used 93% to 99% less EXCP's

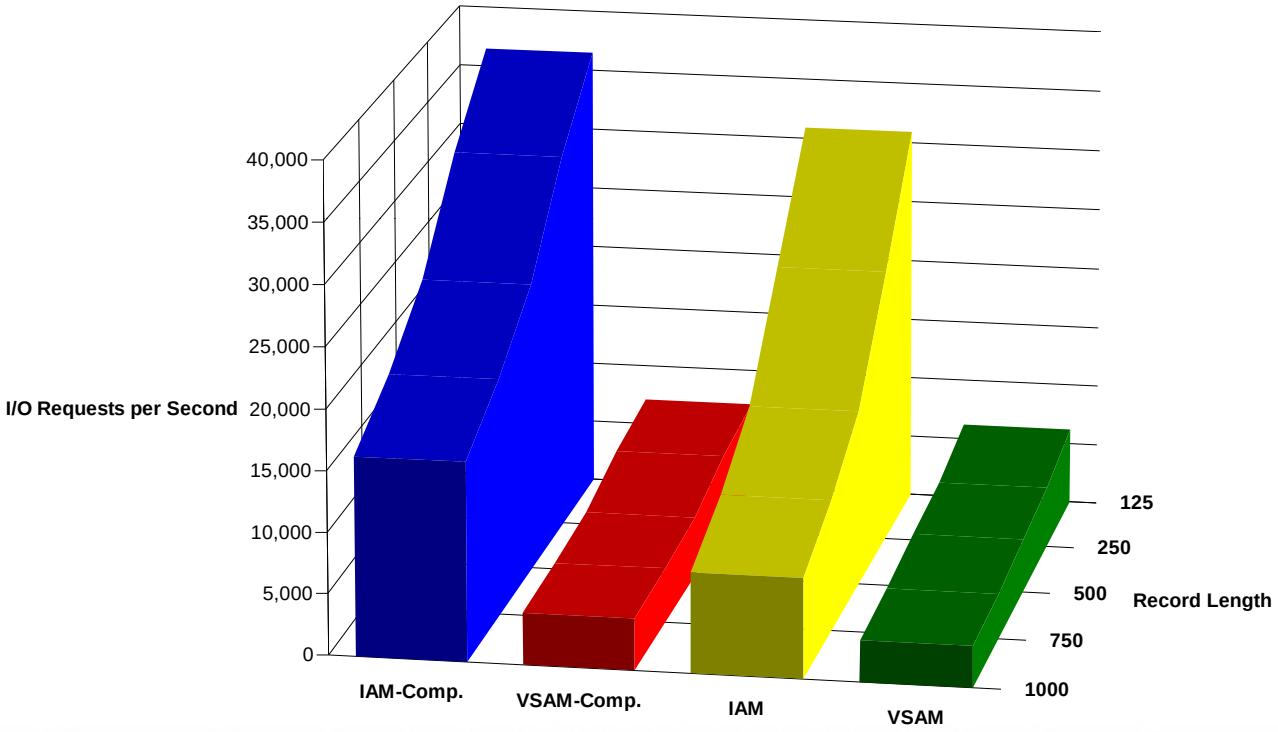
IAM V8.1 vs. VSAM
AIX Non-Unique Key Sequential Performance



IAM V8.1 vs. VSAM
AIX Unique Key Sequential Performance

- Test Description:
  - Sequentially read entire file through unique key AIX
  - VSAM improved over prior test due to sequence for AIX and Base generally being identical
  - IAM's Real Time Tuning still clear winner
- IAM used 48% to 61% less CPU time
- IAM ran in 59% to 81% less elapsed time
- IAM used 72% to 96% less EXCP's

IAM V8.1 vs. VSAM
AIX Unique Key Sequential Performance



IAM V8.1 vs. VSAM Summary

- IAM V8.1 consistently outperforms VSAM SMB
- IAM V8.1 significantly reduces manual tuning
  - Exceptional file load performance
  - IAM easily handles COBOL ACCESS IS DYNAMIC
  - Outstanding performance with AIX
  - Significantly reduces CI lock outs in normal update processing
- IAM reduces application computing costs
  - CPU time
  - EXCP's
  - Elapsed Time
  - DASD space

IAM z/OS 1.8 Support

- Available Now
- V8.0/33 or higher
- V8.1/04 or higher
  - Required for Large Format Sequential Data Set Support

IAM in the Future

- IAM SYSPLEX Record Level Sharing
  - Initially use XCF services
  - Ease of use
  - Less development time
  - Reduced complexity brings greater reliability
- Increased use of 64-bit virtual
  - Index to files
  - Buffers

IAM V8.1 Summary

- Provides substantial value
  - Proven technology
  - Great Performance
  - Easy to use
- IAM Improves VSAM application performance
  - Reduces CPU time, I/O, Elapsed time
  - Little or no tuning required
- IAM ongoing development provides future value
  - SYSPLEX Record Level Sharing
  - Increased use of 64-bit virtual storage