z/TPFDF Update

Chris Filachek
z/TPF and z/TPFDF Architecture & Development

z/TPFDF Delivered Enhancements

Optimized B+Tree Add Operations

A z/TPF programmer can efficiently add LRECs to a B+TREE data file without knowing the current position in the data file.

Previously: Programmers had to know the current LREC position and decide how to efficiently add an LREC

a. DBADD without USEBTREE: Check current block and then sequentially search data blocks for add location

b. DBADD with USEBTREE: Ignore current block and use B+TREE to find add location

Now: Programmers can skip the USEBTREE option and always add LRECs efficiently using the B+TREE

a. DBADD without USEBTREE: Check current block and then use B+TREE to find add location

b. DBADD with USEBTREE: Ignore current block and use B+TREE to find add location

APAR PI76015

Encryption and Cache Updates

A database administrator can make room for new files in the z/TPFDF control global by removing unused entries and without resizing the global.

- Encryption and cache controls for z/TPFDF files are stored in the IDFCNTRL format-2 global
- z/TPFDF files are added to the control global by the ZUDFM ENCRYPT / CACHE commands
- Use new REMOVE parameter on the ZUDFM ENCRYPT / CACHE commands to remove a z/TPFDF file from the control global
  - File is removed only if encryption and cache are disabled for the file

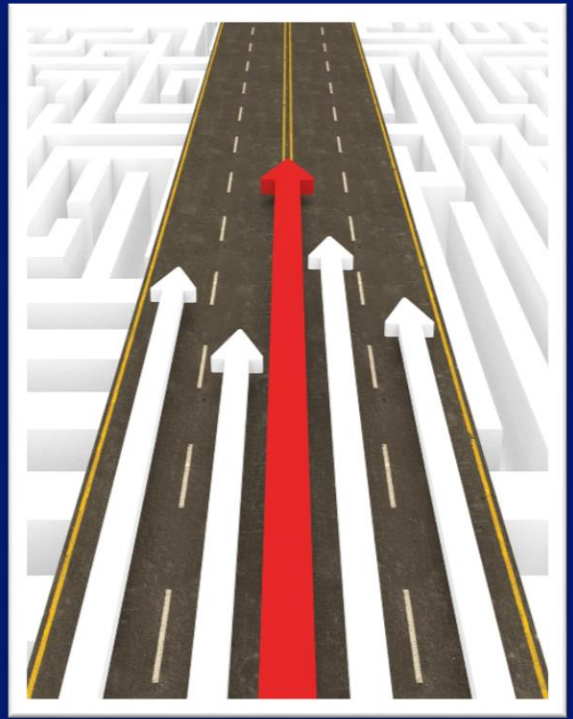
APAR PI85156

Support for FARF6 Fixed Records

A Database administrator can allocate large z/TPFDF fixed record databases and create extremely wide indexes without impacting 4-byte file addressing.

Previously: Fixed records could be allocated using only 4-byte file addresses (FARF 3/4/5)

- Limited to 4 billion addresses shared across fixed and pools
- Size of z/TPFDF indexes and hashes may be limited based on availability of file addresses



Now: Define z/TPFDF files using FARF6 fixed records

- Allocate a single fixed record with up to 4 BILLION ordinals
- Minimize overflow records across a large hash space
- Does not impact to availability of 4-byte file addresses
- Use FARF6 fixed records in new databases only

APARs PJ44596 & PI76032

CRUISE Enhancements

A database administrator can restore fixed record databases faster and with fewer z/TPF system resources.

- **Previously:** CRUISE REBUILD restores all records to pools and then copies primes to fixed records
  - Consistent database view during restore
  - Extra copy step increases restore time
  - Impractical on systems with few available pools

- **Now:** Use CRUISE REBUILD with FIXDIRECT option to restore primes directly to fixed records
  - Avoids extra I/O – Restore databases faster
  - Avoids extra pool use – Restore databases to systems with limited pools
  - Intended for test systems - Database is not consistent during restore

APAR PI83551

For updates on Data Events for z/TPFDF, see “Business Events Update”

z/TPFDF Future Enhancements

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.

Recoup and CRUISE Statistics: As-Is

- A database administrator can use the ZRECP STA and ZFCRU DISPLAY-STATISTIC commands to display database statistics for a z/TPFDF database.
- Use statistics to understand database size, current state, migration progress, etc.
  - Statistics include block counts, chain lengths, LREC & LLR counts, encryption status, etc.
- Statistics for a single database are limited to a maximum number of referenced file IDs
  - A database is a top-level file ID and file IDs referenced through that top level ID
  - Statistics for referenced file IDs past the limit are not collected
    - For example, recoup would only collect statistics for 16 of 30 referenced file IDs
  - May not have complete view of a database

- Recoup limited to 17 IDs
  (1 top-level ID + 16 referenced IDs)
- CRUISE limited to 22 IDs
  (1 top-level ID + 21 referenced IDs)

Recoup and CRUISE Statistics: To-Be

A database administrator can use the ZRECP STA and ZFCRU DISPLAY-STATISTIC commands to display the complete set of statistics for a z/TPFDF database.

- Provide a full view of the database by increasing the maximum number of referenced file IDs for recoup and CRUISE statistics
- Use the same limit for both recoup and CRUISE

- How many referenced file IDs do you have in a database?
- Does the number of referenced file IDs tend to grow? How quickly?

Variable Length Algorithms: As-Is

- Algorithms are used with indexes to find subfiles using data other than the prime file address
  - For example: Find a customer subfile using the customer ID, phone number, email, etc.

- Algorithms are defined with a fixed length in the DBDEF
  - Variable length data (names, emails, etc.) is padded or truncated to fit the fixed length
  - Index LRECs contain the fixed length algorithm

- Algorithms with a large difference between average length and the fixed length
  - Wasted space in many LRECs
  - More overflow records than desired
  - For example: Index using email addresses
    - Fixed length of 250 characters, but may only average 20 characters

Example Index Record

z/TPFDF Header
Index LREC 1: jane.doe@aol.com **Padding - Wasted Space**
Index LREC 2: john.smith@gmail.com **Padding - Wasted Space**
Index LREC 3: jonathan.edward.winters @My.Really.Big.Important .Company.com
z/TPFDF Trailer

Variable Length Algorithms: To-Be

A database administrator can use variable length algorithms and support large algorithm lengths while efficiently using storage and minimizing overflows.

- Algorithms are defined as variable length in the DBDEF
  - Variable length algorithms consist of a length followed by data
  - No padding required

- Size of index LREC uses the actual algorithm length
  - No “wasted space” in index LRECs
    - Each LREC is sized appropriately
  - Store more LRECs per index record
  - Minimize overflow records
    - Fewer overflow records reduces I/O and CPU overhead

Example Index Record

Variable Length Algorithms: Our Current Thoughts

- Format of variable length algorithm is length field followed by data
  - Length field: 2 bytes
  - Maximum data length: 253 bytes
  - Maximum algorithm size (length + data) is 255 bytes
- Applications use new algorithm string format to access subfiles
  - Only affects files defined with new variable length algorithms
- Do you need to set a lower limit for the maximum algorithm length?
  - For example: Limit the maximum length to 100 bytes?
- Is there a use case for variable length algorithms in multi-level indexes?

Algorithm String: Fixed Length Algorithms

Index data	Algorithm data	Algorithm Padding
john.smi	john.smith@gmail.com	**********

Algorithm String: Variable Length Algorithms

Index data	Alg Len	Algorithm data
john.smi	20	john.smith@gmail.com

Calling all Sponsor Users!

Help us make sure our enhancements solve your needs!

Contact us if you are interested in either of the following z/TPFDF enhancements….

- Recoup and CRUISE statistics
- Variable Length Algorithms

Thank You!

Trademarks

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.