

Streamlining Operations

OpenEdge 12.8 for DBAs

Marie Candela

Principal Technical Product Manager

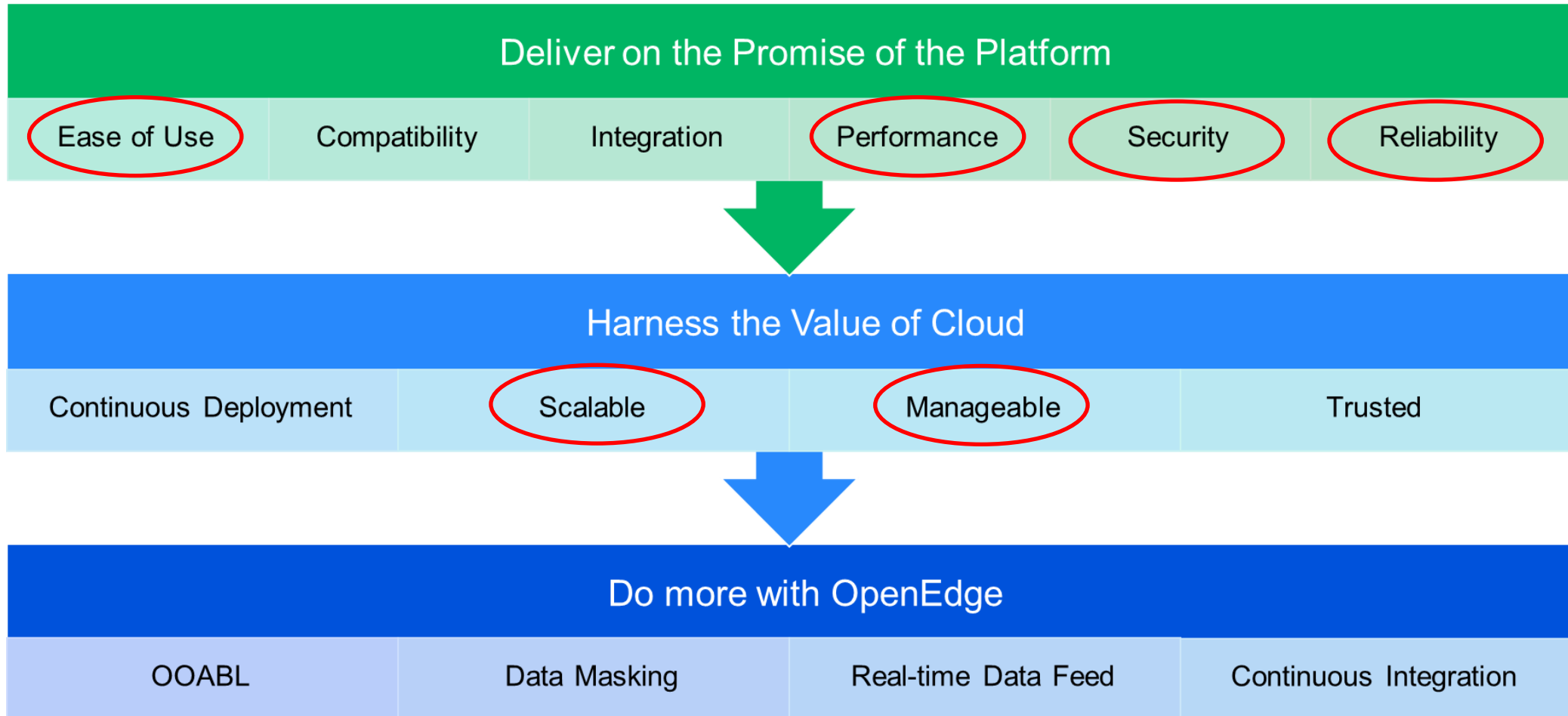
September 2024



Agenda

- Themes
- Enhancements
- Wrap Up
- Q & A

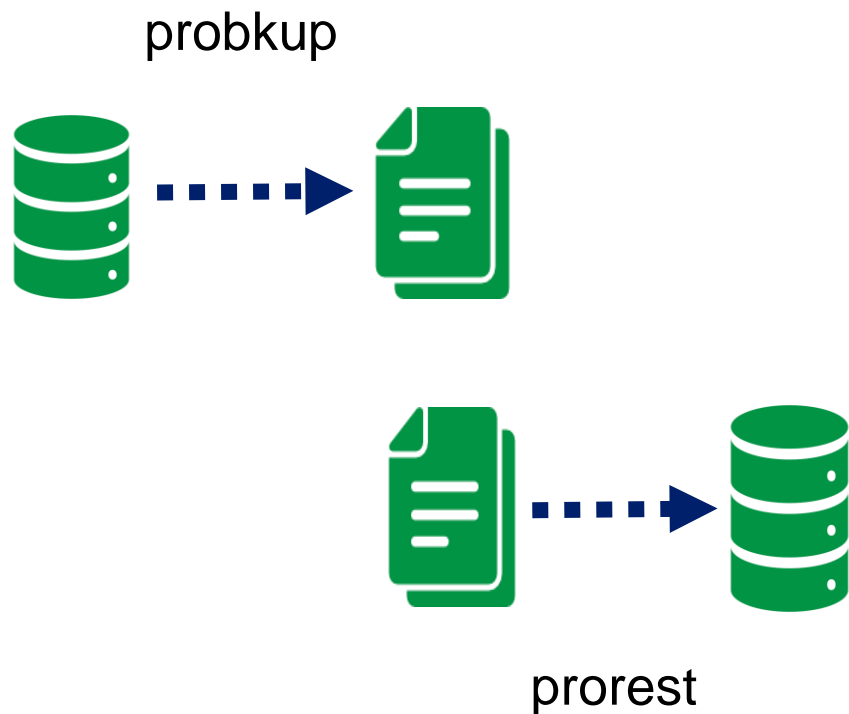
Themes for OpenEdge 12.8



Performance Enhancements

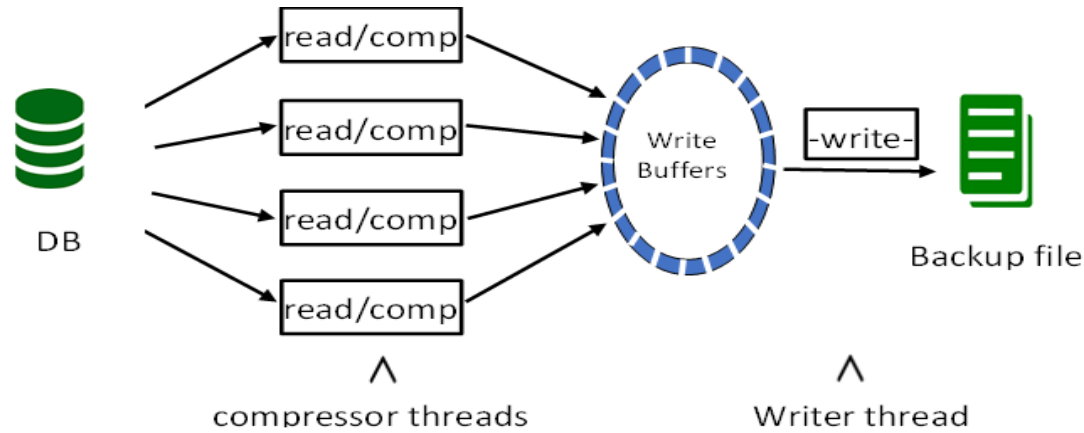
Backup and Restore

Backup and Restore

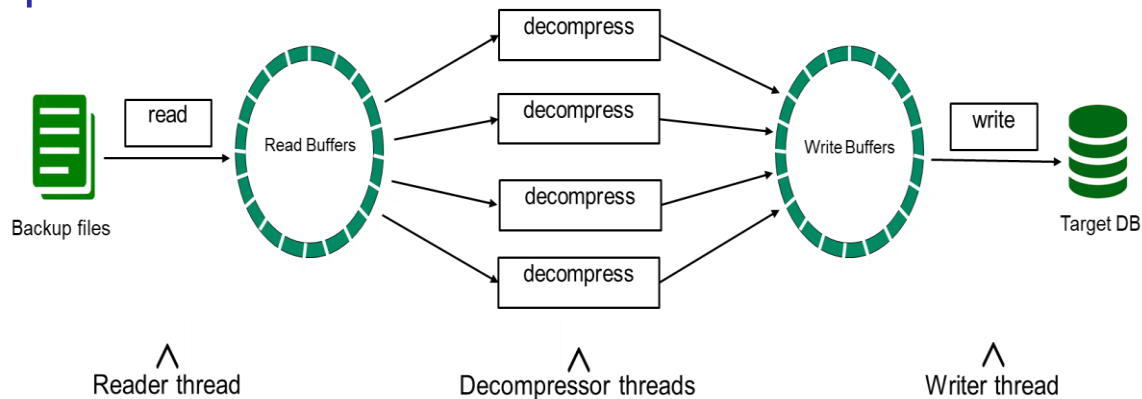


- probkup and prorest are CPU intensive operations
- They spend significant time in:
 - Disk I/O
 - Compressing/decompressing backup blocks
 - CRC check to verify there are no corruptions in the database backup blocks
- Analysis: the operations can spend 90% of their time in compression and checksum tasks

probkup



prorest



Multi-threaded Backup and Restore

- Cached read and write buffers
 - minimize disk I/O contention
 - improve concurrency
- Parallel buffer transfer operations
- Parallel compression / decompression
- Restore: Parallel CRC check
- Asynchronous disk I/O

Threading Availability

```
probkup online testdb testdb.bk -com -thread 1 -threadnum 4 -wbf 4  
prorest testdb testdb.bk -thread 1 -threadnum 4 -rbf 4 -wbf 4
```

- probkup **online**
 - Database stays online
 - Users stay connected
- prorest
 - Restores full and incremental backups
 - Works with `-noverify`
- Options
 - `-thread`: enable threading [off]
 - `-threadnum`: number of threads [lesser of (6 or the number of CPUs)]
 - `-wbf`: number of cached database block write buffers
 - `-rbf`: number of cached backup block read buffers [prorest only]

Compression

- RLE compression algorithm is rather basic and insufficient
- Use the third party ZSTD compression algorithm
- Probkup `-com -comlevel`
- Ability to specify `-comlevel` to choose speed vs space
- Starting with 12.5, default became ZSTD instead of RLE
- Online and offline backups
- Improves performance for Backup *and* Restore

Syntax:

`probkup [online] db-name db-name.bk [-com -comlevel n]`

`prorest new-db-name db-name.bk`

<code>-com</code>	<code>-comlevel</code>	Compression Algorithm Used
not specified	not specified	No compression
not specified	specified	No compression
specified	not specified	ZSTD - uses default <code>-comlevel: 3</code>
specified	1	RLE – decent speed, very low compression
specified	2	ZSTD – Fastest: least compression
specified	3	ZSTD – Balanced: speed vs compression (Default)
specified	4	ZSTD – Slowest: highest compression

CRC in the Restore process

- prorest spends a significant percentage of its time verifying database integrity. This is done as part of the restore using cyclic redundancy check (CRC)
- prorest **-noverify** option skips performing the CRC
- To ensure no corruption in the backup: partial verify option (-vp) on prorest
 - prorest -vp **only** reads the backup volumes and computes and compares the backup block CRCs with those in the block headers
 - Does not actually restore the database
 - May be run separately and standalone
 - Use this on each backup file after created
 - If the partial verify detects a CRC mismatch, do NOT restore the database with -noverify

Index Check

Multi-threaded Index Check

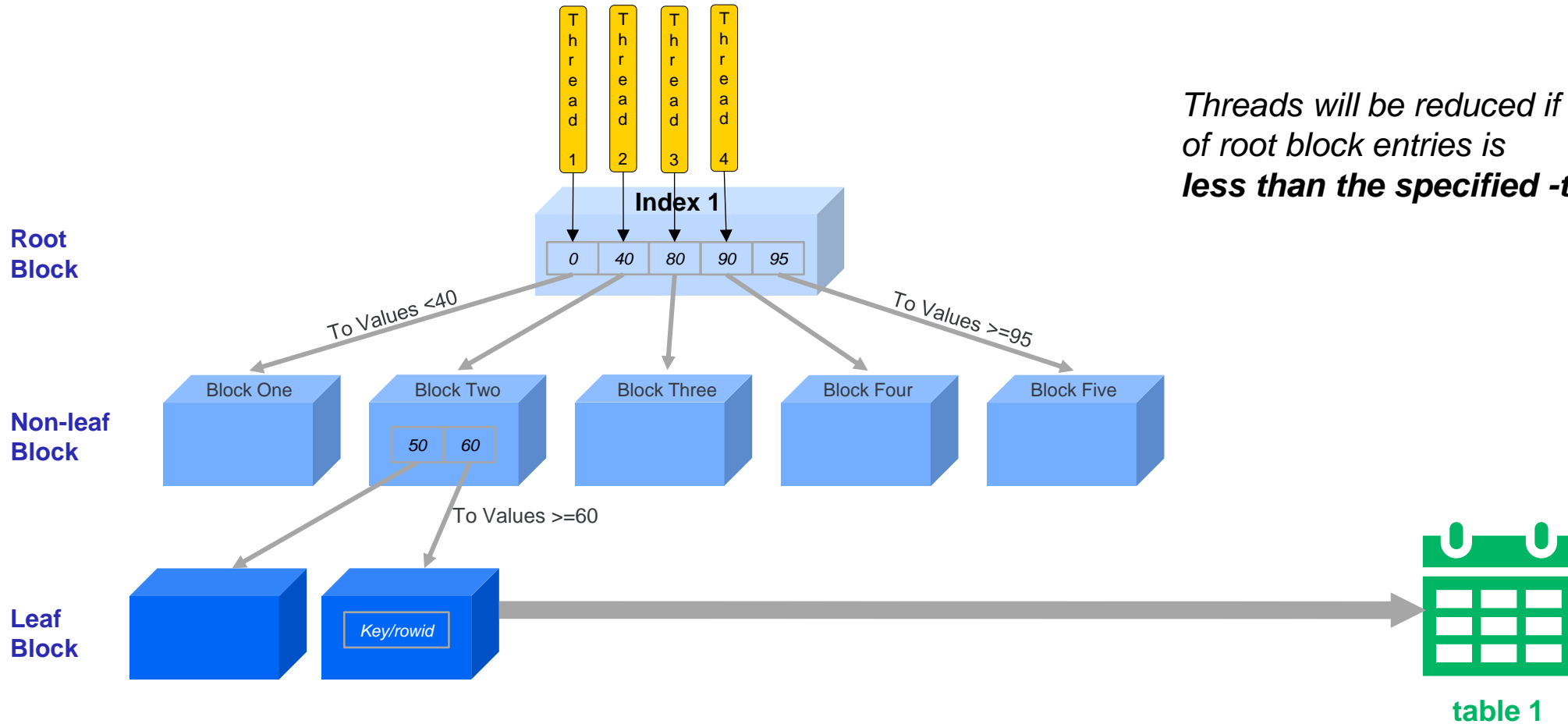
Challenge

- Index check is one of the two most commonly used Database health check tools. It is used to identify problems and to ensure database integrity. It is particularly important in a DB down situation.
- Due to the volume of structures, it can take a very long time to execute
- On a terabyte database, index check can run for more than a month.

Enhancement

- Implement threading techniques to improve parallel processing in the index check operation
- Available for Online IDXCHECK for options 2, 3, and 4
- The threading model varies with the processing algorithm of each option

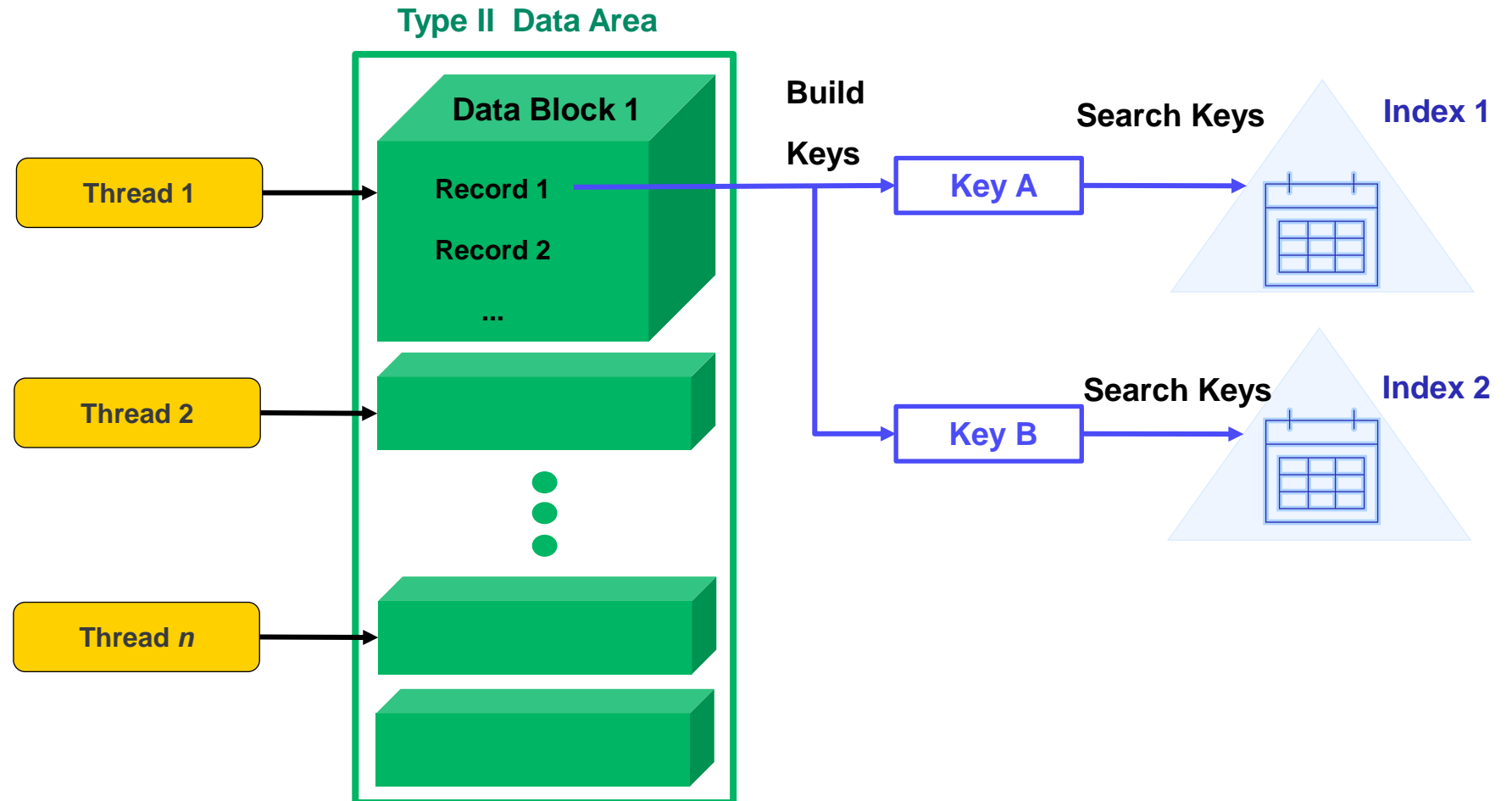
Key Entry-centric Thread Model



Threads will be reduced if the number of root block entries is less than the specified -threadnum

Record-centric Thread Model

1. Reads next data block in Area
2. Extract next record from data block and build index keys
3. Search index trees with keys



Threading Availability

```
proutil testdb -C idxcheck -thread 1 -threadnum 4
```

- **Online IDXCHECK ONLY**
 - Option 2: Validate keys for each record
 - Option 3: Validate record for each key
 - Option 4: Validate key order
- Threads are not spawned:
 - If a word index is selected
 - For small table objects (option 2)
 - For small index trees (options 3 & 4)
- **Options**
 - -thread: enable threading [on]
 - -threadnum: [lesser of (6 or the number of CPUs)]
 - Cannot exceed the number of CPUs

Index Fix

Index Fix Record Scanning: Options 1 & 8

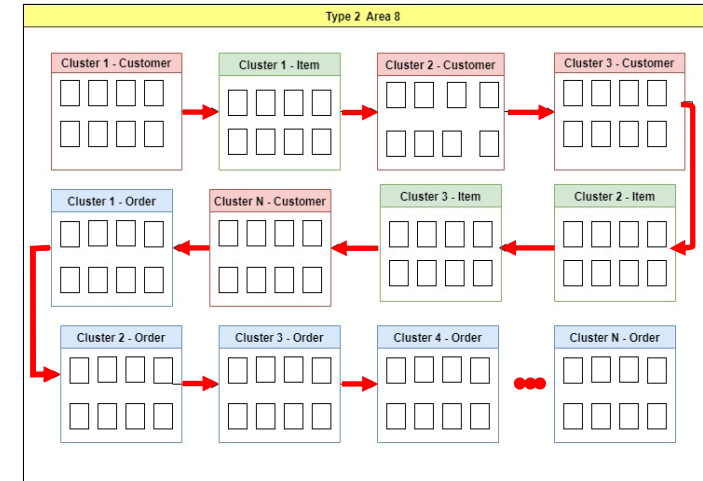
- Options 1 & 8 Scan the **records** for missing or incorrect index entries with index block validation

Challenge

- Scans the whole area, the blocks of EACH cluster, and follows the dbkey of the next block or cluster

Enhancement

- Take advantage of Type II clusters
- Use the object list block to scan ONLY the clusters belonging to the table items, following the cluster link
- Applies ONLY to the scenario of a single table in an Area that contains multiple tables



Index Fix Index Scanning - Option 2

- Option 2 scans the **indexes** for invalid index entries. If a problem is found in a key, the key is fixed.

Challenge

- The process locks ALL the tables for each index key.
- When we commit a transaction it is expensive as it generates a lot of latch activity.

Enhancement

- Reduce the number of transactions required to do the index scan
=> reduces the number of commits and number of latches acquired and released
- Applies when scanning ALL tables, with ALL or many indexes

Index Compact

Index Delete Chain

Challenge

- Index entry deletion for unique indexes is a special case
 - Index entries are not removed, they are modified to become “delete holders” that remain in the trees
 - When a block is full of delete holders, it is put on the delete chain of the index
 - When a table is deleted, the whole index might be put on the index delete chain
- Mass purge operations can result in more blocks on the delete chain than remaining blocks in use
=> Queries scanning over unused index blocks adversely impact OLTP performance
- Currently the cleanup of unused blocks on the index delete chain is embedded in some operations, e.g. idxbuild and record create/update

Enhancement

- Customers have requested the ability to quickly remove unused index blocks that they know they have added as a result of a mass delete activity
- The IDXCOMPACT utility operates in three phases:
 - **Phase 1** — If the index is a unique index, the delete chain is scanned and the index blocks are cleaned up
 - **idxcompact unusedblocks** cleans up the unused index blocks by executing only Phase 1 of idxcompact

Index Activate

Index Activate Performance

Challenge

1. _File table reads for every single record
idxactivate touches

2. Idxactivate is wasteful when inserting
index entries.

For every record, it verifies the entry does not exist in the target index before the insertion.

Enhancement

1. _File rarely changes

Implement a cache to read once during the building phase of idxactivate

2. Stop doing the verification

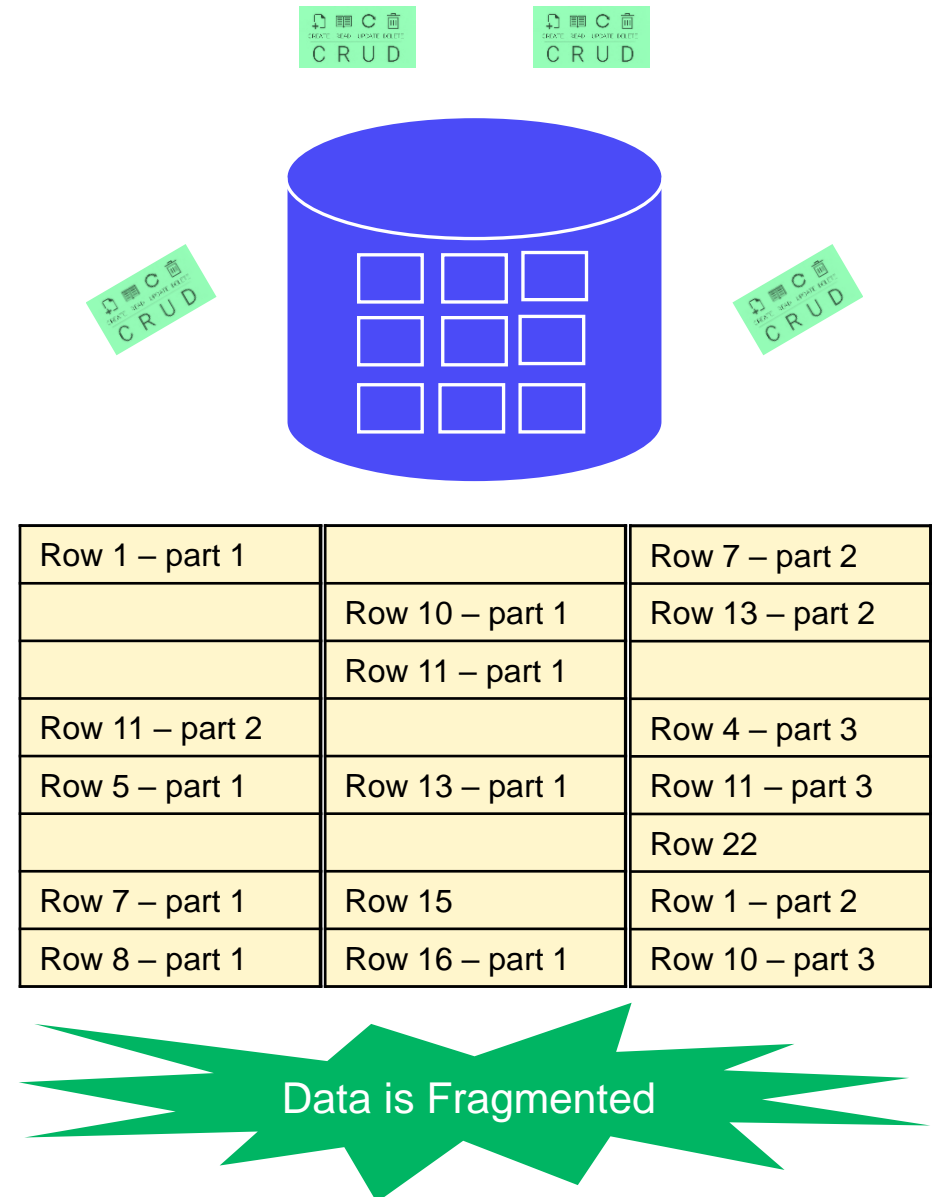
The index manager already handles duplicate key entries

Space Management

Tables

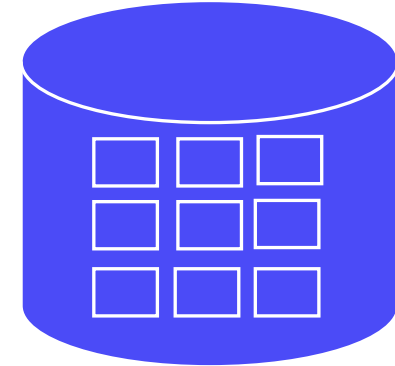
Record Fragmentation

- Context
 - Tables are created pristine and orderly, records contained within their blocks
 - Over time usage can create record fragmentation
 - Records split over multiple blocks (Row 1 – part 1, Row 1 – part 2)
 - Records scattered and out of order – impacts sequential access
- Existing options
 - Restrict access to the table and indexes and perform a dump & load operation offline
 - tablemove utility – doubles disk space



Online Table Reorganization

- Solution
 - **tablereorg** utility
 - Type II storage areas
 - Specify index to sort on; defaults to primary
 - During the **tablereorg** space is reused as cluster becomes empty so does not double the size of table reserved space
- Outcome
 - Scatter factor is reduced - no longer indicates need for a dump and re-load
 - Improve application performance when doing sequential scan operations for queries and table scans
 - Avoids database downtime when trying to improve performance



Row 1 – part 1		Row 7 – part 2
	Row 10 – part 1	Row 13 – part 2
	Row 11 – part 1	
Row 11 – part 2		Row 4 – part 3
Row 5 – part 1	Row 13 – part 1	Row 11 – part 3
		Row 22
Row 7 – part 1	Row 15	Row 1 – part 2
Row 8 – part 1	Row 16 – part 1	Row 10 – part 3



Optimizing tablereorg

Cancel and Resume

- Cancel an in progress tablereorg operation
- Restart with **resume** on the command line to request that the cancelled tablereorg operation resume where it left off
- Resume will be from the first block of the current cluster

Restrict

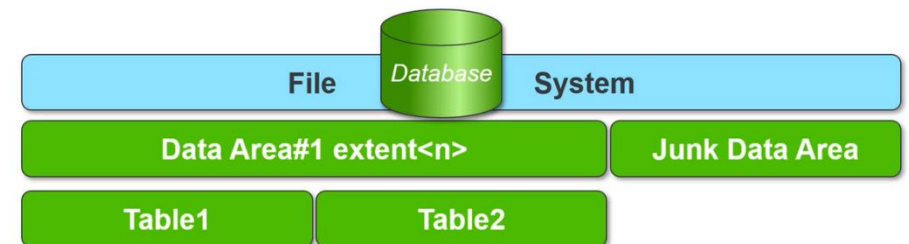
- Reorganizes **sections** of a table instead of the **entire** table
- Parameters indicate starting and ending positions within the table – manages how much of the table is scanned
- Omitting the restrict option reorganizes the entire table

Smarter

- Smart scanning is the ability to skip reorganization of records that are already in the order of the specified index
- Scans until an out of order record is encountered
- Starts the reorganization process at the first block in the cluster where the violation occurred

Table Truncation

- Context
 - As time goes by, customers find they no longer have use for some data, perhaps because it is historical, CDC or _KeyEvent data, the existing application no longer uses a table, or they want to move the data out of a storage area.
 - Leaving unused data contributes to storage costs and processing overhead.
- Existing options
 - Table drop: uses EXCL/DB access and invalidates the rcode
 - Record delete by the client: scales linearly, impacts application performance (concurrency), and does not return the storage to the area

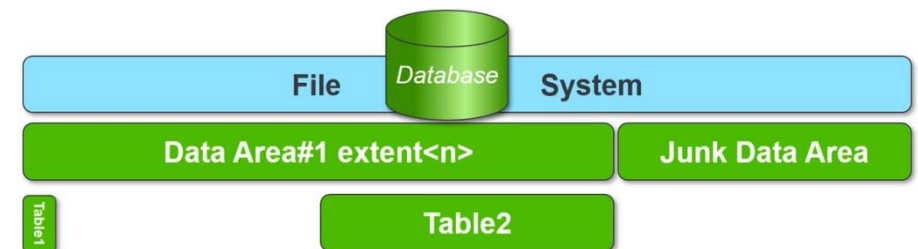


Online tablemove [truncate]

- tablemove [truncate] allows you to truncate in place (leave in same storage area) or truncate and move to a different storage area
- Makes its storage space previously reserved to the area available for reuse in the area
- Online with active applications currently running
- Tables and LOBs must be in Type II storage areas
- Does not excessively grow the BI file
- Very quick – done at cluster chain level and scales flat
- Recoverable
- Changes are propagated to Replication targets

- Outcome

- Closest thing to being able to do a drop table online without affecting the rcode and CRCs
- DBAs will be able to remove database object (Tables, Indexes, and LOBs) content quickly and return the storage to the storage area for reuse by other objects.
- Avoids database downtime when trying to improve performance



Areas

Online Area Truncation

- Context

- The area truncate operation is an offline operation
- It requires the AI to be disabled and truncates the BI
- It forces Replication targets to be rebased

- Outcome

- Avoids costly database downtime for storage area management

- Solution

- Enhance existing **proutil truncate area** command to work online
- Works with After Imaging enabled
- Works with Replication
- Works with Type I and Type II areas
- Works with PUB and Non-PUB tables
- Schema is preserved
- Area high water mark is reduced
- All object chains are reset: no rm chain or index delete chain affecting a truncated object
- Prerequisite: Area needs to be empty of objects

Online Area Removal

- Context

- The **prostrct remove** utility removes extents from storage areas. When all extents are removed from the storage area, prostrct remove removes the storage area.
- prostrct remove is available only when the database is offline, therefore requiring a maintenance outage to use it.

- Solution

- Enhance existing prostrct remove command to work online – **prostrct removeonline**
- Works with Replication
- Works with Type I and Type II areas
- Data storage areas only
- Fixed and variable extent types
- Prerequisite: Area must be empty of storage objects and truncated

- Outcome

- Avoids costly database downtime for storage area management
- Enables customers to reduce their space footprint dynamically, without waiting for a scheduled maintenance window

Putting it all together

Returning Storage Space to the OS Online

Online Truncate Table

Use **tablemove [truncate]** to truncate table data online to give storage space back to the storage area.



Online Truncate Data Storage Area

proutil truncate area truncates the data storage area to remove tables and associated objects.

This reduces the high-water mark to reflect object removal.



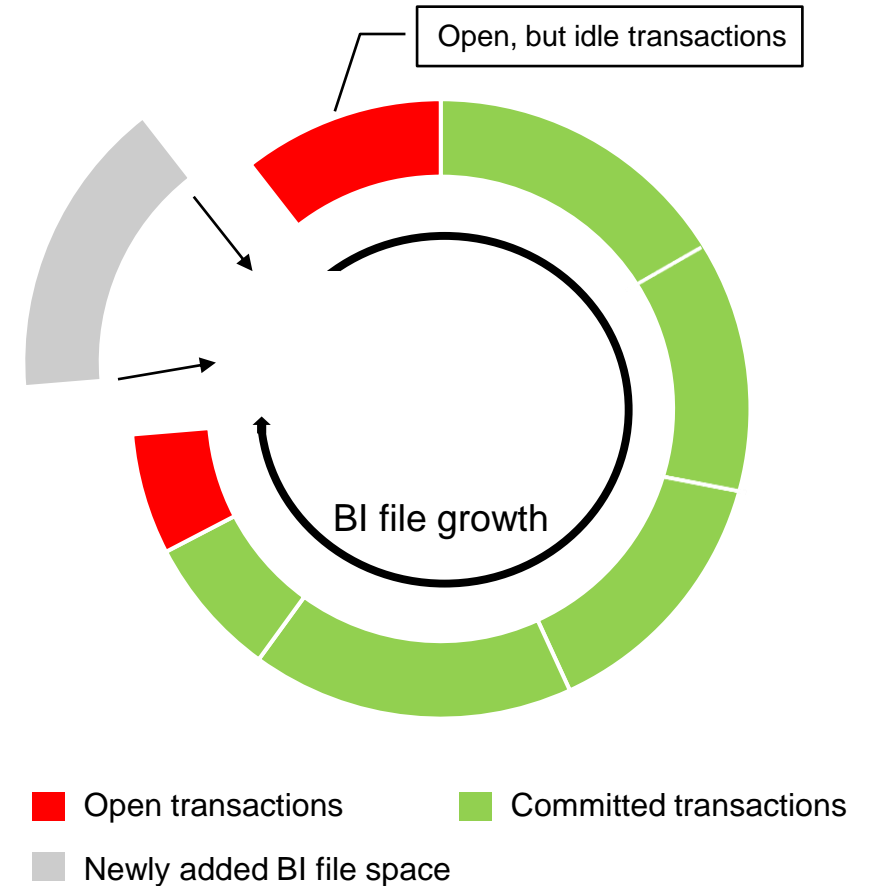
Online Remove Data Storage Area

Return space to the operating system using a **prostrct remove online** operation.

BI File Management

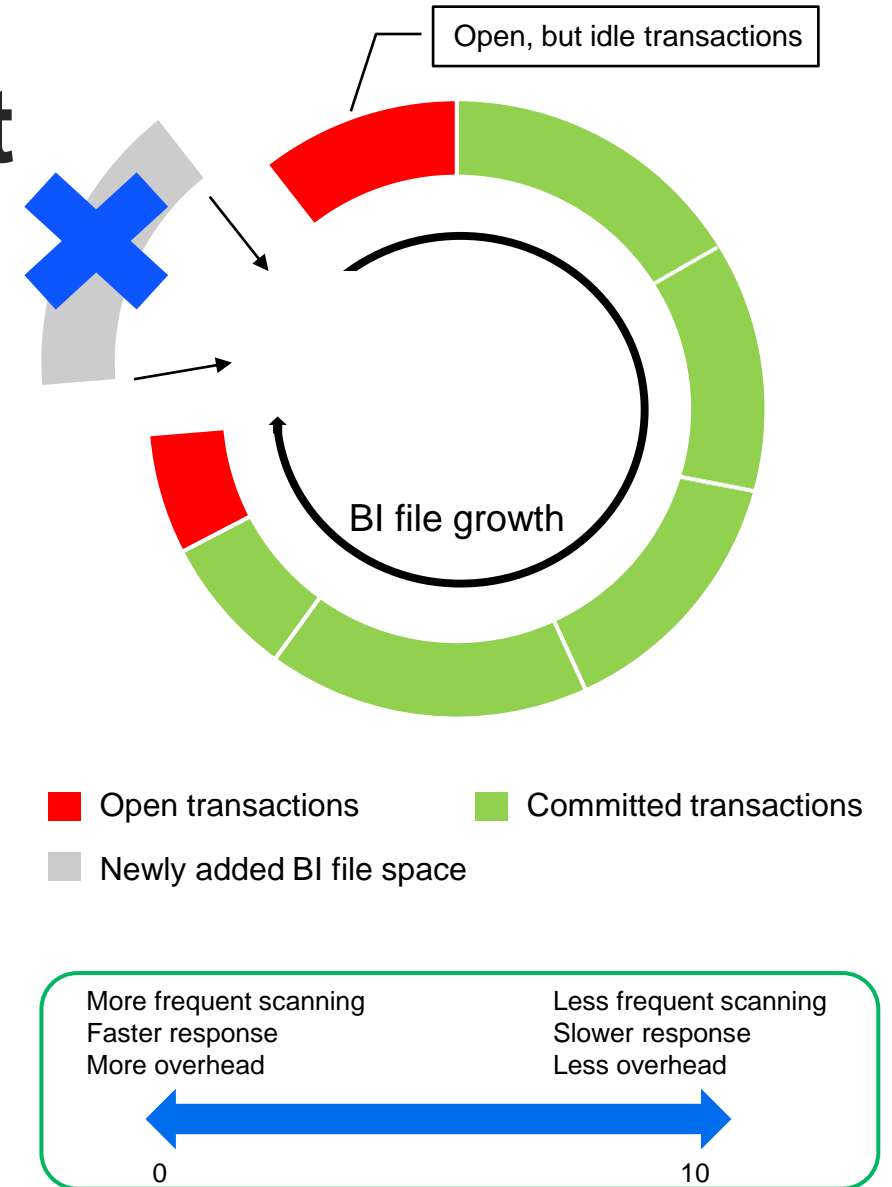
Database Before Imaging (BI)

- Context
 - Journal of database activities
 - Critical for basic database functionalities
 - Can become a huge performance bottleneck
- Challenge
 - User/application mistakes can lead to unexpected and expensive BI file growth
 - Long running idle transactions exacerbate
 - No other way on replication target database
- Existing option
 - Shutdown the database and truncate BI



Online BI Space Management

- Solution
 - Scan BI file clusters for committed and recyclable clusters to reuse and reduce or eliminate unexpected BI file growth
 - Before-image manager process is **probim**
 - New database startup parameter for tuning: `-biscantime`
 - Tuned online in `promon` and `_DBParam VST`
- Outcomes
 - Reduce unnecessary database storage usage
 - Avoid database performance slowdown caused by user mistakes



BI Management Analysis Findings

- BI space management can significantly reduce or eliminate abnormal BI growth caused by users/applications
 - Up to ~9% performance gain in the case of long idle transactions
 - No noticeable resource overhead from BI Space Management
 - Smaller `-biscantime` leads to a quicker response to BI growth, with little more overhead compared to larger `-biscantime`
- **Set `-biscantime` default to 10 seconds. You can tune it down online when still seeing BI growth**

Security

Hardware Security Module for your TDE key store

A Hardware Security Module (HSM) is used when your cryptographic keys need the highest level of protection

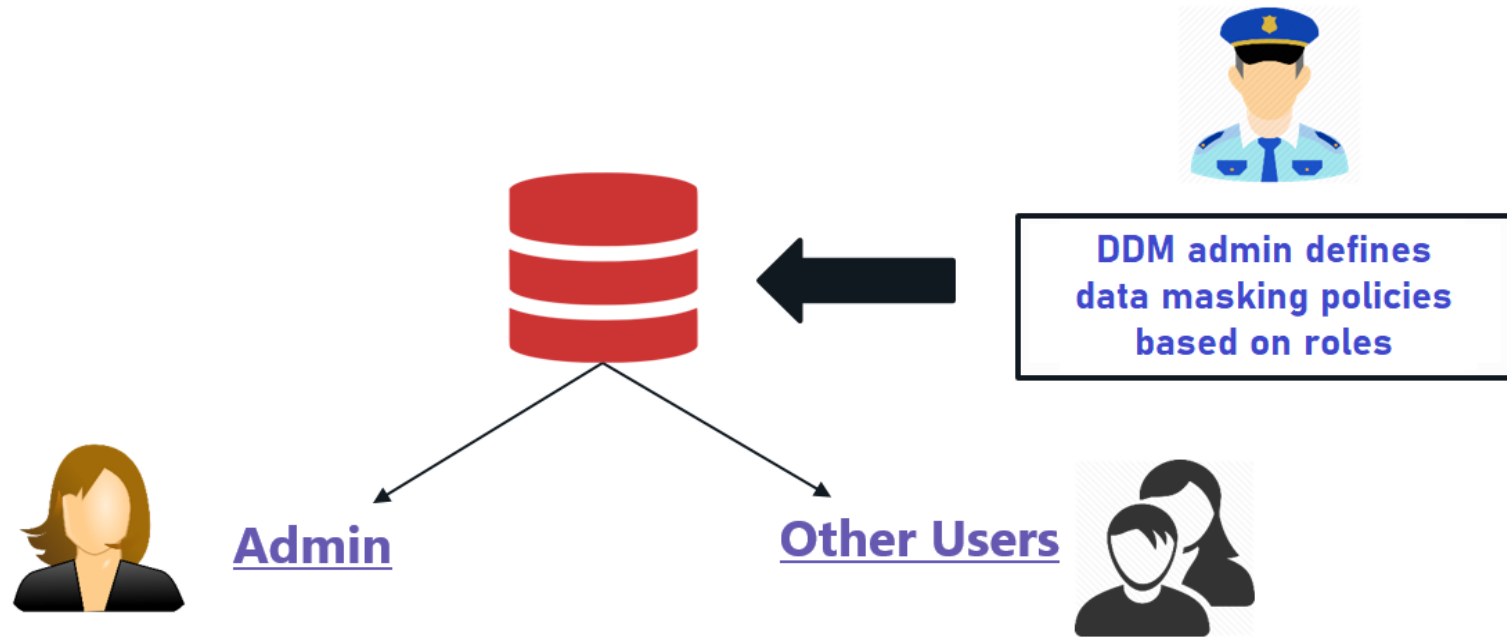
HSMs are specialized devices that securely store/use cryptographic keys

- Tamper-resistant hardware
- Stores and protects keys, and makes available to authorized users
- Keys do not need to be loaded into the web/application server memory

Can be used to augment OpenEdge TDE encryption key storage security



Protect Sensitive Data with Dynamic Data Masking (DDM)



First Name	Last Name	Age	email	SSN
Bob	Jones	48	bob@xms.com	123456789
Alis	Wonder	46	alis@xms.com	987654321

First Name	Last Name	Age	email	SSN
Bob	Jones	?	xxx@xxx.com	XXXXX6789
Alis	Wonder	?	xxx@xxx.com	XXXXX4321

Ease of Use

Database Edition Skew

Production Databases

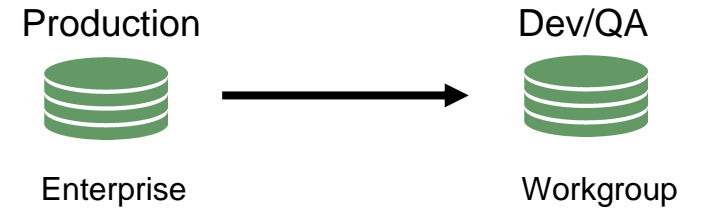
Dev/QA Databases



Enterprise License

Workgroup License

Large File Support



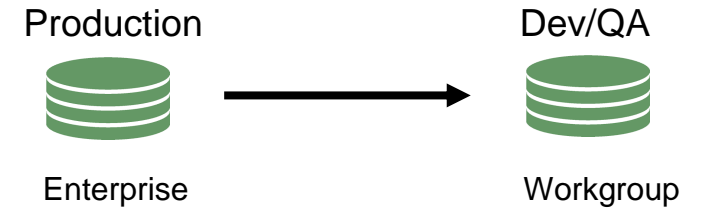
Challenge

- *“Database having existing file extents greater than 2GB large file size limit cannot be opened under a sub-Enterprise license”*

Enhancement

- No longer constrain Large File support to Enterprise edition
- Databases will automatically have support enabled if:
 - Newly created
 - Opened
 - Converted with conv1112
 - Restored from an earlier backup

CDC Enabled Database



Challenge

- The restored backup cannot be brought online in those systems without a CDC license.
- CDC cannot be disabled on the backup because CDC policies exist.
- The policies cannot be removed because there is no CDC license.

Enhancement

- Proutil db-name -C **deactivateCDC**
 - No license, data purging, or policy removal required
 - All writes to CDC tables stopped, clients can access
- Proutil db-name -C **activateCDC**
 - Requires license. All writes resumed.
- Both can be run online or offline

AI Archiver Enablement

Challenge

Can be enabled by either:

- **Offline** through rfutil

```
rfutil <db-name> -C aiarchiver enable
```

- **Online** through probkup

```
probkup online mydb mydb.bck enableaiarchiver
```

Enhancement

- Enhance the rfutil aiarchiver enable command to be able to operate online
- Have all of the arguments supported by probkup online:
 - -aiarcdir <dir-list>
 - -aiarcdircreate
 - -aiarcinterval <n>

Backup File for What Version?

Challenge

- *We need “anything that gives someone the ability to see what the version is that backed the file up, especially if prorest cannot restore it. We had this issue attempting to restore a V9 backup and had to poke around with installs trying to figure out which one might do it.”*

Enhancement

- **prorest -describe** lists basic information about a database backup file to help you identify which backup you want
- Full and incremental backups
- Compatible from version 9 onwards

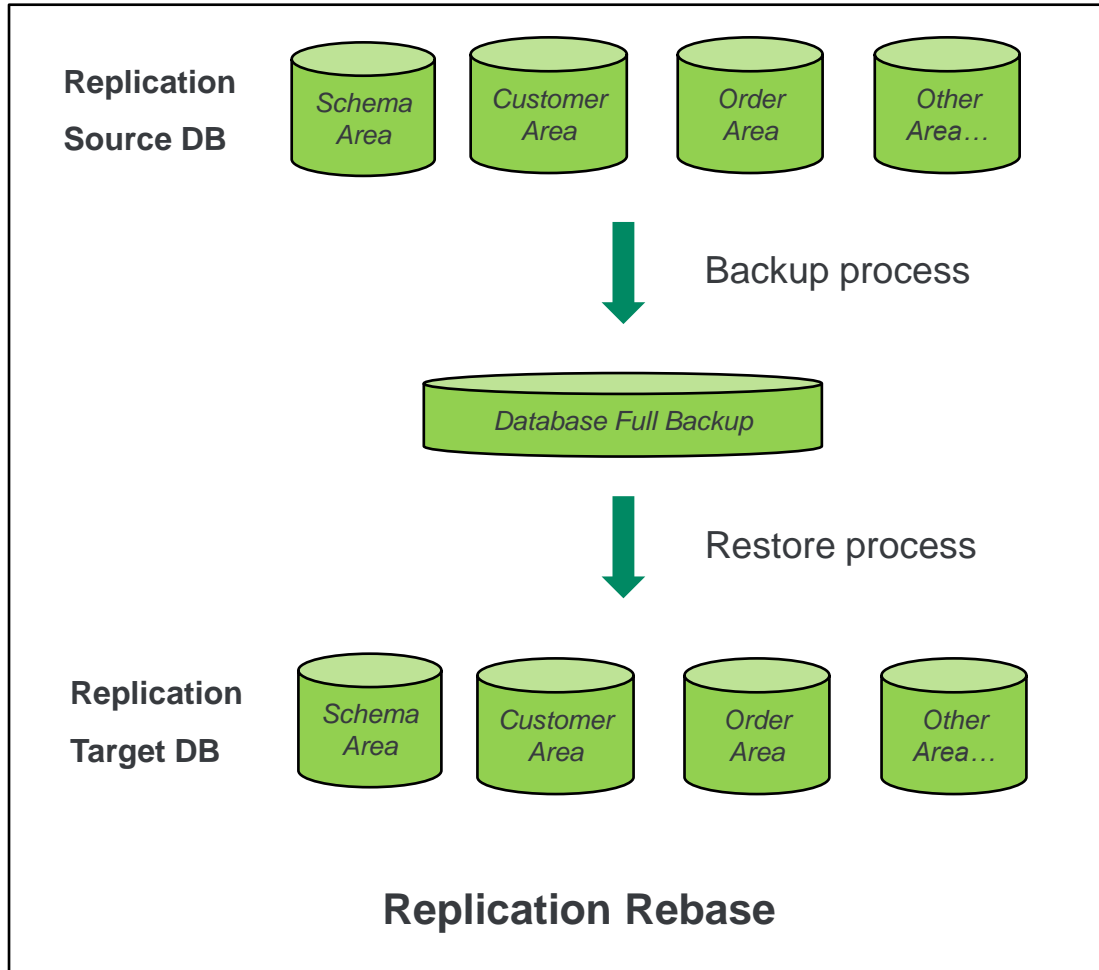
PROREST -describe

Volume 1 of a full backup for database: /largedisk/prorest-describe-demo/demodb.db
The backup was taken at: Tue Oct 12 17:46:11 2021
The OpenEdge version where the backup was taken: OpenEdge Release 12.5
The minimum OpenEdge version to restore the backup: OpenEdge Release 12.5
The backup version: 0xd1 (209)
The platform for the backup: Linux
The endianness for the backup: Little Endian
Online backup (online): TRUE
Incremental backup (-incremental): FALSE
Enable AI online (enableai): FALSE
Backup BI clusters (bibackup): TRUE
Volume size (-vs): 0
Blocking factor (-bf): 34
Backup without recovery (-norecover): FALSE
Compression (-com): TRUE
Compression level (-comlevel): 3
Incremental overlap factor (-io): 0
Redundancy factor (-red): 0

OpenEdge 11 -> 12 Upgrade with Replication Targets

Challenge

- Upgrading from an earlier version to OpenEdge 12 requires a Replication rebase
 - After the conv1112 has been performed on the source database
 - Rebase process is to take a full backup of the source and then restore to create the target(s)
- Expensive, especially in the cutover activity

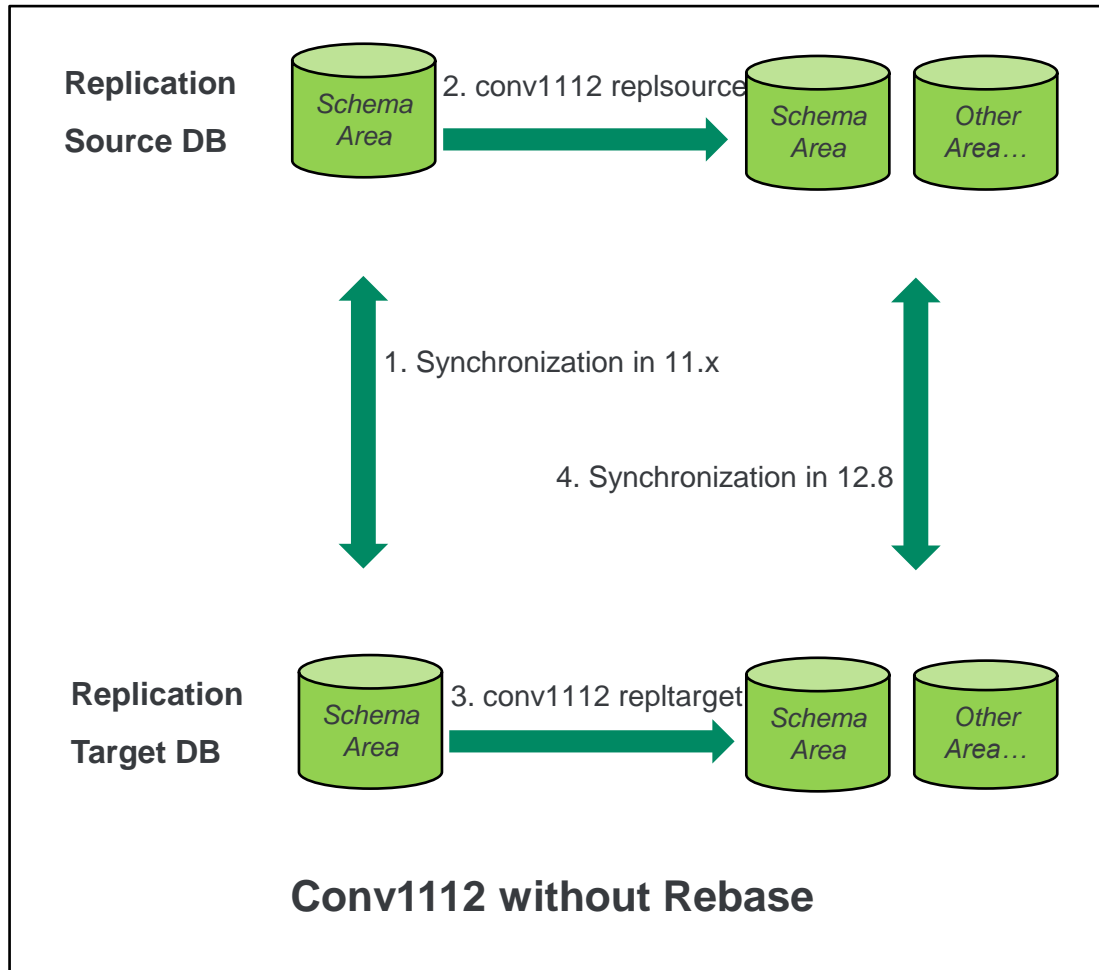


Conv1112 Avoids Replication Target Rebase

Enhancement

- Conv112 is enhanced to provide additional support for Replication enabled databases
- **Conv112 replsource** converts the source database from a version 11 to version 12, and produces a schema area backup file
- **Conv112 repltarget** consumes the schema area backup from the replsource operation to convert the database from a version 11 to version 12 target, and prepares master block data for version 12 synchronization

Note: user tables in the schema area reduce/remove the benefit of this enhancement



Wrap Up

Next Steps

- What's New in OpenEdge 12.8 document

<https://docs.progress.com/bundle/openedge-whats-new/page/Whats-New-in-OpenEdge-12.8.html>

- Explore OpenEdge 12.8

<https://www.progress.com/openedge/whats-new>

- Consult the Platform Compatibility Guide

<https://docs.progress.com/bundle/OpenEdge-12-Platform-Compatibility-Guide/resource/OpenEdge-12-Platform-Compatibility-Guide.pdf>

- **Be mindful of the 11.7 LTS retirement: April 1, 2025**

News You Can Use



Questions?

