What’s New with VSTs

Richard Banville
Fellow, OpenEdge Development
November 16, 2017
VSTs: What and Why

- Virtual System Tables
  - Meta-schema definition but no stored record data
- Useful for DBA
  - Can monitor activity
    - Detect, analyze and fix issues in running system
    - Monitor for maintenance needs
    - Local or remote monitoring
  - Can change parameters
- Useful for applications
  - Embed within the application
    - Control private buffers for example
    - Monitor activity for query performance
And More

- 30 other VSTs
- Additional perspectives
  - Database
  - Resources
VSI Common Questions
Fun with VSIs and VST Rowids

▪ A VSI is a Virtual “System-table” Index

▪ Like a regular/real index
  • Uses cursors and brackets for
    – Positioning
    – Filtering/choosing data
    – Retrieving data
  • Provides a rowid for record retrieval

▪ Unlike a regular/real index
  • Does not have a physically stored b-tree
Fun with VSIs and VST Rowids

- All VSTs have exactly one VSI on its one ID field
  - The VST rowids always start at 1
  - The VST ID field almost always start as 1
    - Except for _TableStat, _IndexStat

- VST’s ID vs numerical association
  - ID: _Connect-id = 1
  - Numerical association can start at 1 or 0
    - _Connect-Usr = 0
Quickest Access to VST Data

Indexes retrieve rowid’s for a record that satisfies your request

- **FAST**: A “find” using a non-index field is slower than a find by an indexed field

  ```sql
  find _connect where _connect-Usr = 10 // Find user id 10
  ```

  - 11 index requests return 11 rowids to the AVM, one at a time
  - The AVM requests 11 records one at a time and filters out 10 of them returning 1 record to the user.

- **FASTER**: Lookup using VSI

  ```sql
  find _connect where _connect-id = 11 // Find user id 10
  ```

  - VSI simulates index “search” returning rowid 11, the 11th entry in the user control structure
  - Index returns 1 rowid and 1 record is requested by the AVM

- **FASTEST**: For VSIs, the indexed field value IS the rowid

  ```sql
  find _connect where recid(_connect) = 11 // Find user id 10
  ```

  - AVM requests 1 record directly – no VSI involvement
Special Indexing

- `TableStat-id, IndexStat-id`
  - Id is object number starting at base value
  - Object number is therefore indexed
  - Rowid != Stat-Id
  - Rowid = (Table# - base) + 1

OE 11.7.0 fixed VSI for shifting of base != 1

```
Find _DbParams where _DbParams-Name = "-basetable".
theEntry = (myTable-Num - _DbParams-Value) + 1.
Find _TableStat where _TableStat-Id = theEntry.
// OR, using Recid rather than VSI
Find _TableStat where recid(_TableStat) = theEntry.
```
Special Indexing

- Find MY stats: _UserTableStat-Id, _UserIndexStat-Id
  - Object stat data by user by table
  - Stat-Id = Rowid
  - Rowid = (User# * range) + (table# - base) + 1

*NOTE: -baseusertable in OE 11.7.1

OE 11.7.0 fixed VSI for base != 1
VSTs: Getting updates / reverting updates

- Applying new vst updates
- OE 11.4 backward compatibility - after OE 11.5 updatevst
  - New VST tables must be removed for backward compatibility

Database Features

<table>
<thead>
<tr>
<th>ID</th>
<th>Feature</th>
<th>Active</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td>64 Bit DBKEYS</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>New VST Tables</td>
<td>Yes</td>
<td></td>
</tr>
</tbody>
</table>

An invalid feature 23 has been encountered in the database’s Enabled feature list. (11727)
The list of enabled features in database x contains features that are not recognized by this codebase (11810)

proutil <db> -C disablenewvsttables
So What’s New?
Lock Improvements

- Lock table (OE 11.4) and _UserLock table (OE 11.6) query performance
  - Unreliable sequential hash table scan -> now reliable sequential scan
  - Only returns active lock entries (OE 11.5)

- L 100 000: For each _Lock:
  - 179 (sec) now 1 (sec) improvement ??? %
  - Eliminated latch contention 7.4 mill/sec ~1 billion latches now ~190 latches

- L 250 000: For each _Lock:
  - Gave up on the query after 58 minutes!

Bottom line
- Vast query performance improvement for _Lock and _UserLock
- Can now reliably be used in production w/no impact
_Lock Improvements (OE 11.4)

- **User Identification**
  - _Lock-DomainId
  - _Lock-Tty /dev/pts/6 myHostName 6

- **Transaction information**
  - _Lock-TransId – Join to _Trans table for deep txn information
  - _Lock-Trans-State
    - NONE (dead!)*
    - BEGIN (allocated)*
    - ACTIVE (doing stuff)
    - PREPARING (Prep)
    - COMMITTING (Phase 2)²DB
    - PREPARED (Phase 1)²DB
  - _Lock-Trans-Flags
    - FWD
    - UNDO
getDbParams (OE 11.5 – new table)

- Consistent reporting of database startup parameters
  - `getDbParams` – New VST table with startup parameters
    - _Startup VST is obsolete
  - `getDbParams`, prolog and broker to .lg file, promon with translated messages
  - Contains `{Name, value, msg #, description, default info, modifiable flag}`

- Update `getDbParams`-value to change startup parameter online

```
Find `getDbParams` where `getDbParams`-Name = "-spin".
Assign `getDbParams`-Value = "10000".
```

- Values are all character datatype
- Failure to assign is silent so double check!
**Dbparams Example**

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
<th>Description</th>
<th>Msg-Num</th>
<th>Default</th>
<th>Modifiable</th>
</tr>
</thead>
<tbody>
<tr>
<td>-aiarcdir</td>
<td>0</td>
<td>After-image Management Archival Directory List (-aiarcdir): Not Enabled</td>
<td>13873</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>-aiarcdircreate</td>
<td>0</td>
<td>Create After-image Management Archival Directory(s) (-aiarcdircr)</td>
<td>13874</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>-aiarcinterval</td>
<td>-1</td>
<td>After-image Management Archival Interval (-aiarcinterval): -1</td>
<td>13872</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>-aibufs</td>
<td>20</td>
<td>Number of After-Image Buffers (-aibufs): 250</td>
<td>4256</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>-aistall</td>
<td>0</td>
<td>After-Image Stall (-aistall): Not Enabled</td>
<td>4254</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>-baseindex</td>
<td>1</td>
<td>Starting index number for statistics range (-baseindex): 1</td>
<td>17555</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>-basetable</td>
<td>1</td>
<td>Starting table number for statistics range (-basetable): 1</td>
<td>17554</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>-baseuserindex</td>
<td>1</td>
<td>Starting index number per user for statistics range (-baseuserindex): 1</td>
<td>18406</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>-baseusertable</td>
<td>1</td>
<td>Starting table number per user for statistics range (-baseusertable): 1</td>
<td>18404</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>-bibufs</td>
<td>20</td>
<td>Number of Before-Image Buffers (-bibufs): 250</td>
<td>4252</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>-bistall</td>
<td>0</td>
<td>BI File Threshold Stall (-bistall): Disabled.</td>
<td>6551</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>-bithold</td>
<td>0</td>
<td>BI File Threshold size (-bithold): 0.0 bytes</td>
<td>9238</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>-blocksize</td>
<td>8192</td>
<td>Database Blocksize (-blocksize): 8192</td>
<td>6573</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>-bwdelay</td>
<td>0</td>
<td>BIW writer delay (-bwdelay): 0</td>
<td>12812</td>
<td>yes</td>
<td>yes</td>
</tr>
</tbody>
</table>
Servers - Server parameters (OE 11.5)

- Primary / manual broker or server
- Auto DB and OE SQL Server information
  - _Server-Broker-Pid
  - _SrvParam array fields {Name, Value, Type, Desc, Msg-Num, Default, Modifiable}
  - One array entry per parameter
    - -maxport
    - -minport
    - -Ma
    - -Mi
    - -Mm
    - -Mp
    - -Mpb
    - -N
    - -PendConnTimeout
    - -S
    - -ServerType
    - -SQLCursors
    - -SQLStack
    - -SQLStmtCache
    - -SQLTempStoreBuff
    - -SQLTempStoreDisk
    - -SQLTempStorePageSize
    - -SQLTruncateTooLarge
    - -maxport
    - -minport
    - -Ma
    - -Mi
    - -Mm
    - -Mp
    - -Mpb
    - -N
    - -PendConnTimeout
    - -S
    - -ServerType
    - -SQLCursors
    - -SQLStack
    - -SQLStmtCache
    - -SQLTempStoreBuff
    - -SQLTempStoreDisk
    - -SQLTempStorePageSize
    - -SQLTruncateTooLarge
_UserLock

- Contains multiple field arrays with info of first 512 current locks
- Added **current total locks** and **current totals by lock type** (OE 11.6)
  - _UserLock-Total-
    - Record
    - Table
    - Purge
    - RecGet
    - Partition
    - SHR
    - EXCL
    - IS
    - IX
    - SIX

*Lock table overflow, increase -L on server (915)*

- Who’s the guilty party?
  - _UserLock-HWM (OE 11.7.0)

*For each _UserLock by _UserLock-HWM descending:*
User Activity Statistics (OE 11.7.0)

- Separated database and user level parameters for table and index activity statistics

Example:

- 2000 tables 8000 indexes & 1000 users
- \((2000 \times \text{tstatsize}(80)) + (8000 \times \text{istatsize}(48))\) * 1000 = \(~380\text{Mb}\)
- ~Equivalent to ~B 50 000

New Parameters

- `-baseuserindex`
- `-baseusertable`
- `-userindexrangesize`
- `-usertablerangesize`

- Defaults to database level –basetable/index params
- `_DbParams` and `_StatBase` can update base stats value on the fly.

Find first _Statbase.

Assign _BaseUserTable = 100 _BaseUserIndex = 10.
User Activity Statistics (OE 11.7.0)

- Separated database and user level parameters for table and index activity statistics
- Example:
  - 2000 tables 8000 indexes & 1000 users
    \[(2000 \times \text{tstatsize}(80)) + (8000 \times \text{istatsize}(48))\] \times 1000 = \(~380\text{Mb}\)
  - ~Equivalent to ~B 50 000
- New Parameters
  - -\text{baseuserindex}
  - -\text{userindexrangesize}
  - -\text{baseusertable}
  - -\text{usertablerangesize}
  - Defaults to database level –\text{basetable/index params}
- \_DbParams and \_StatBase can update base stats value on the fly.

Find first \_DbParams where \_DbParams-Value = “-\text{baseusertable}”.
Assign \_DbParams-Value = 100.
Database Buffer Pool activity (OE 11.7.0)

- **_ActBuffer**: {-B}, {-B2} and {-B, -B2}
  - Provides read/write, and checkpoint activity in the buffer pools
  - Stats on # of each block type (RM, index, master, …) in the cache
  - # active buffers – tells you if your buffer pool if approaching full or not
    - _Buffer-Active
    - _Buffer-Master
    - _Buffer-Index
    - _Buffer-RM
    - _Buffer-Free
    - _Buffer-Seq
    - _Buffer-Area
    - _Buffer-Object
    - _Buffer-ObjList
    - _Buffer-Control
    - _Buffer-ClusterMap
    - _Buffer-ObjClist
    - _Buffer-BlockMap
Checkpoint - Synchronize memory and disk

1. All database changes are halted
2. BI buffer pool is flushed (-bibufs)
3. AI buffer pool is flushed (-aibufs)
4. Db buffer pool is scanned (-B)
   1. Buffers previously marked for checkpoint are written out (Buffer Flushes / DB Writes)
   2. “Dirty” buffers are marked for the next checkpoint & added to checkpoint queue
5. File system cache is synchronized to disk
6. Db changes are allowed once again

- It is important to know the activities impacting the checkpoint
  - Avoid “Buffers flushed at checkpoint”
  - Increase “Time between checkpoints”
More Checkpoints

- History of 8 checkpoints is not a history
- Additional checkpoints (OE 11.7.0)
  - `numCheckpointStats`
    - New startup parameter
    - Default 32
    - Max 1024
- Also included
  - `Checkpoint-Buffers`: written so far (OE 11.6.0)
  - `Checkpoint-Cluster`: bi cluster ring information (OE 11.7.0)
  - `Checkpoint-Number` (OE 11.7.0)
Checkpoint – Why is my checkpoint taking so long?

<table>
<thead>
<tr>
<th>CPT Q</th>
<th>Scan</th>
<th>APW Q</th>
<th>Database Writes</th>
<th>BI Writes</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>99</td>
<td>9</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>188</td>
<td>19</td>
</tr>
</tbody>
</table>

- “Flushes” changed to “Writes”
- Added BI writes (OE 11.6)
- Still not enough information
  - What else happens in a checkpoint?
  - How long does each activity take?
Checkpoint – Why is my checkpoint taking so long?

- Total Duration & Sync Time (OE 10.something)
- DB Write Time, BI Write Time (OE 11.6)
  - Helps identify if APW is keeping up and auto tuning properly
    - Auto tuning enhanced to consider previous checkpoint statistics

- Increase –bibufs to avoid “empty bi buffer waits” can have side effects
  - Monitor APW / BIW activity
  - Monitor Checkpoint cost and find a balance
  - Validate Bi file’s disk speed

<table>
<thead>
<tr>
<th>Database Writes</th>
<th>Performance Timings</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB Writes</td>
<td>BI Writes</td>
</tr>
<tr>
<td>99</td>
<td>9</td>
</tr>
<tr>
<td>188</td>
<td>19</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DB Write</th>
<th>BI Write</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.01</td>
<td>0.05</td>
</tr>
<tr>
<td>0.01</td>
<td>0.07</td>
</tr>
</tbody>
</table>
_AreaStatus (OE 11.7.0)

- _AreaStatus-Type
  - Control, Schema, Data, BI, AI, TL

- AI improvements including
  - **Full**
  - **Busy**
  - **Archived**
  - **Empty**
  - **Locked**
  - **Arch/Locked**

- _AreaStatus-LastOp
  - Timestamp of last state change

- _AreaStatus-AIActivated
  - Timestamp when “New”ly activated

- _AreaStatus-AI-Seq
_Connect (OE 11.7.0)

- Database wide connection information
- `_Connect-NumTrans`
  - # transaction begin requests
  - Includes commit, rollback, and txns with no changes
**_Connect (OE 11.7.0)**

- **Private buffers**
  - Avoid fouling DB buffer pool by table scan type queries
  - `_Connect-NumSeqBuffers, _Connect-UsedSeqBuffers`
    - Already in `_MyConnection`
  - Application can only update `_MyConn-NumSeqBuffers` at runtime

```
Find _MyConnection.
Assign _MyConn-NumSeqBuffers = 20.
  For each History  NO-LOCK: end.  // Table scan to foul buffer pool replacement
Assign _MyConn-NumSeqBuffers = 0.
```
Db user notify (OE 11.7.0)

- Provides active checking for schema changes, no logout/login required
  - Index activate
  - Binary load w/build indexes
  - MT idxbuild
  - TP idxbuild

- Separate client thread “polls” for schema changes

  ```
  proserve <db> -usernotifytime <poll-freq>
  ```

  - Polling is off by default but can be changed at runtime*
    - promon or _DbParams VST

- Suggestion: Set to balance performance vs maintenance requests
  - Set relatively high at db startup: 600 (10 minutes)
  - Anticipate a change? Set it lower prior to maintenance operation
_Connect: DbUserNotify Support (OE 11.7.0)

- _Connect-TimeStamp
  - Schema cache timestamp (OE 11.6)

- _Connect-NotifyTime
  - Currently set poll time

- _Connect-LastNotifyCheck
  - Time of last poll for notifications

- _Connect-Notifications
  - Notifications outstanding or in process
  - Currently only supports “R” for schema re-cache
User Misc Field (OE 11.7.0)

- **What is it?**
  - 16 bytes of “user” defined data
  - `_Connect` and `_MyConnection` share currently set value
  - Could set as part of login process
  - Could set for application monitoring
    - Does not provide history like Application Auditing

- **Security concerns…**
  - Restrict update for non-DBA’s to `_MyConnection` only

```
Find _MyConnection.
Assign _MyConn-UserMisc = “Now I’m here.”.
```

```
For each _Connect:
Assign _Conn-UserMisc = “I’m everywhere!”.
```
Transaction Activity (OE 11.7.0)

- _UserIO and _Trans – per user/trans BI note read/write activity
  - _UserIO-BIRecReads
  - _UserIO-BIRecWrite
  - _Trans-BIRecReads
  - _Trans-BIRecWrites

- User I/O statistics
  - Includes forward and undo processing activity
  - “Is that user doing anything?”

- _Trans statistics
  - Is a transaction doing anything or just sitting there
  - Is the Bi growth really needed or is it from incorrect transaction scoping?
Replication VSTs

- New fields, new tables
- Now consistent with replication utility data

- _Repl-Server
- _Repl-AgentControl
- _Repl-Agent
- _Repl-AgentControlActivity

- _Repl-AgentActivity
- _DbServiceManager
- _DbServiceManagerObjects
- _Repl-InterAgentActivity
Resetting Database Statistics

```
proutil <db> -C zerostats
```

(Use cautiously – all database statistics zeroed)

- VST dates are still character datatype “Thu Apr 27 16:18:21 2017”

- Affect on promon – actions per second vs database up time
  - Value of per second activity degrades over time

```
R&D ➔ 1. Status Displays … ➔ 1. Database

Database was started at: 04/24/17 14:15
It has been up for: 24:09:48
Time of last zerostats operation: 04/25/17 10:15
```