Tips and Tricks for Optimizing Performance with SAP Sybase ASE

Chris Baker
Principal System Consultant
SAP
c.baker@sap.com

Disclaimer

In this session we will be giving tips and guidelines for ASE to provide novices and experienced DBAs with information they may not know

This is by no means exhaustive. We strongly suggest taking advantage of the education course offerings available to enhance your in-depth knowledge of ASE

Agenda

- Planning
- Configuration
- Running/Monitoring
- Troubleshooting

Planning

Planning

1. **Review latest platform Release Bulletins, Installation and Configuration Guides**
   - ✓ http://infocenter.sybase.com/help/topic/com.sybase.infocenter.help.ase.15.7.2/title.htm
2. **Ensure for production servers that memory and CPU map 1:1 to physical hardware.**
   - ✓ Avoid swapping at all costs
   - ✓ Do not overallocate CPU's
     - 1 physical core = 1 engine (pre 15.7)
     - 1 physical core = 1 thread (syb_default_pool)
   - ✓ Be aware of SAN and storage configuration

Planning

3. Make sure O/S configuration settings are sufficient

- ✓ RH Linux
  - ulimits - /etc/security/limits.conf
  - shared memory - /etc/sysctl.conf
  - /proc/sys/fs/aio_max_nr – might need increasing
  - noop scheduler
- ✓ etc.

4. O/S patches and drivers are up-to-date

- ✓ Might require firmware updates

Planning

5. **Separate Log, Tempdb and Data devices**
   - ✓ Spreads the I/O load
   - ✓ Separate Log ensures proper database recovery from database and transaction log dumps
6. **Target R/W speeds should be 5ms per I/O or better**
7. **Use Concurrent/Direct I/O for file systems**
   - ✓ Turn off journaling
   - ✓ Buffered I/O OK for tempdbs

Configuration

Configuration

1. **Use the Threaded Kernel**
   - ✓ sp_configure ‘kernel mode’, ‘threaded’
2. **Use 4K server page sizes or better**
   - ✓ Performance with modern storage systems
3. **Use UTF-8 character set**
4. **Separate Data, Log and Tempdb Caches**
   - ✓ Log caches should always have cache partitions = 1
5. **Create Multiple Tempdbs**
   - ✓ Dedicate a tempdb to ‘sa’
6. **Enable Statement Cache and literal autoparam**
   - ✓ sp_configure ‘statement cache size’
   - ✓ sp_configure ‘number of open objects’

Configuration

7. **Number of worker processes**
   - ✓ Provide a pool for parallel index operations, partitions and ‘dbcc checkstorage’
8. **Configure Job Scheduler for maintenance jobs**
   - ✓ Update Index Statistics
9. **Create Index Performance**
   - ✓ number of sort buffers = 5000
10. **Set max possible CPU utilization**
    - ✓ max online engines = <number of cores>

Configuration

11. Network

- ✓ max network packet size = 16384
- ✓ additional network memory = 10485760

12. Reduce Tempdb I/O

- ✓ direct i/o off, dsync off (filesystem)
- ✓ max buffers per lava operator = 20480
- ✓ session tempdb log cache size = 32768

13. Add an ‘sa’-only listener

- ✓ Additional ‘master’ entry in server interfaces file

Running/Monitoring

Running/Monitoring

1. **Use Optimization Goals and Login Triggers**
   - ✓ set plan optgoal allrows_mix | allrows_oltp | allrows_dss
2. **Set optimizer level = ase_current**
   - ✓ Otherwise no benefit from patches
3. **Run 'Update Index Statistics' regularly**
   - ✓ Leave steps at 20-40
   - ✓ Histogram tuning factor = 20
   - ✓ Job Scheduler (datachange)
4. **Change 'enable housekeeper GC' from default**
   - ✓ 4 - for batches
   - ✓ 5 - for OLTP

Running/Monitoring

5. **Check statistics regularly**
   - ✓ systabstats/sysstatistics
   - ✓ sp_statistics
   - ✓ capture missing statistics
6. **Manage ASE lock granularity**
   - ✓ Don’t overuse DRL
7. **Monitoring**
   - ✓ Turn on MDA monitoring
   - ✓ Configure and use Sybase Control Center

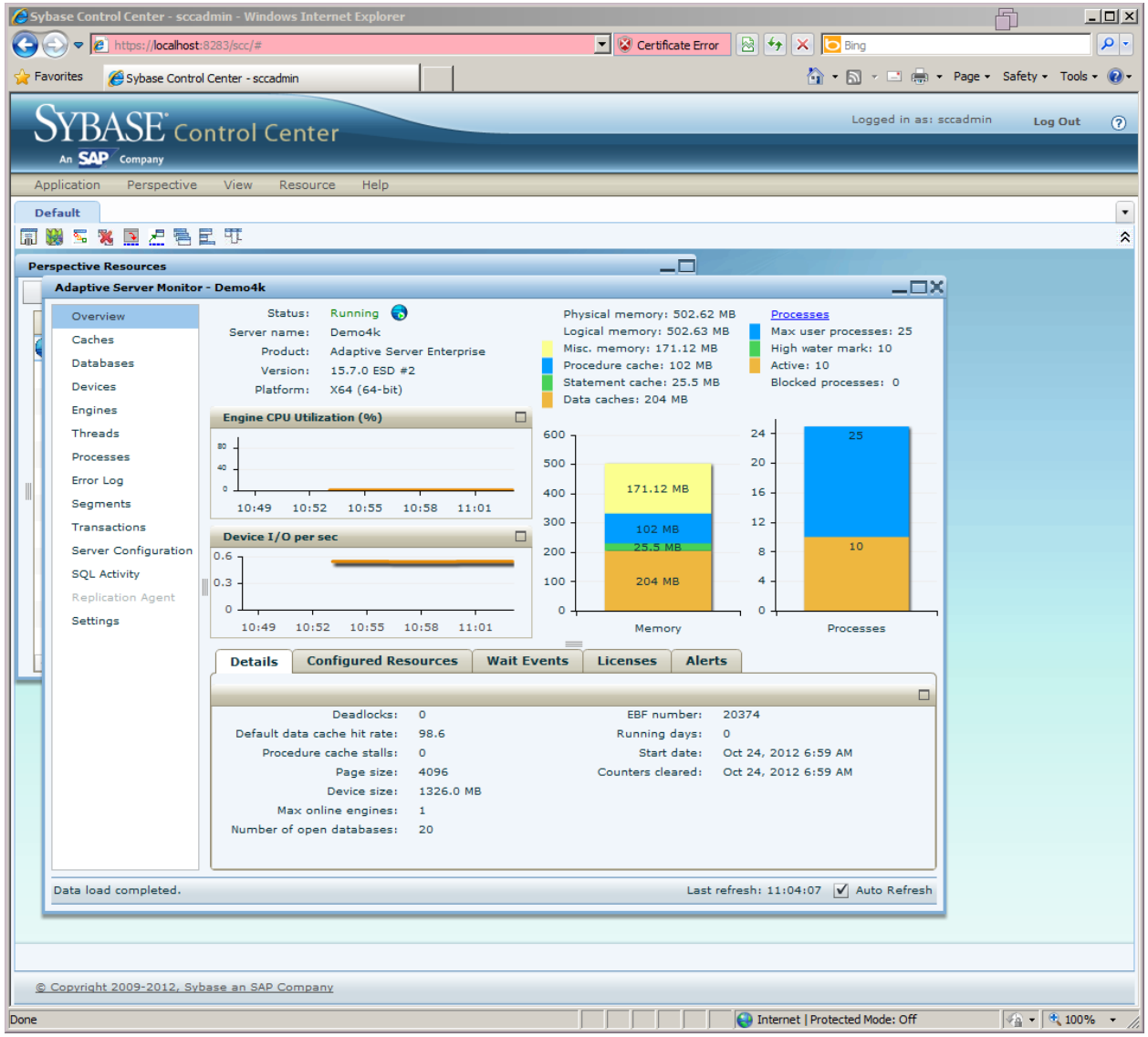
Optimization Goals

- **fastfirstrow**
  - ✓ optimizes queries so that Adaptive Server returns the first few rows as quickly as possible.
- **allrows_oltp**
  - ✓ optimizes queries so that Adaptive Server uses a limited number of optimization criteria (described in “Optimization criteria”) to find a good query plan. allrows_oltp is most useful for purely OLTP queries.
- **allrows_mixed**
  - ✓ optimizes queries so that Adaptive Server uses most available optimization techniques, including merge_join and parallel, to find the best query plan. allrows_mixed, which is the default strategy, is most useful in a mixed-query environment.
- **allrows_dss**
  - ✓ optimizes queries so that Adaptive Server uses all available optimization techniques to find the best query plan, including hash join, advanced aggregates processing, and bushy tree plan. allrows_dss is most useful in a DSS environment.

Housekeeper

- **The housekeeper performs 3 main functions**
  - ✓ Wash
    - Runs only during idle times
    - Can help reduce buffer washing during transaction processing
    - Managed using 'housekeeper free write percent' (default = 1) i.e. only increase disk i/o by 1% above current
  - ✓ Chores
    - Only runs at idle times. Performs tasks such as:
      - Flushing table statistics
      - Flushing account statistics
      - Handling timeout of detached transactions
      - Checking licence usage
  - ✓ Garbage Collection
    - Controlled by 'enable housekeeper GC'
    - defaults to lazy collection of forwarded and deleted rows (1)
      - tables will require more maintenance (rebuild/reorg) to recover space for reuse
    - Set to 4 for batch processing
      - aggressive collection of forwarded rows and deleted space
    - Set to 5 for OLTP processing
      - aggressive collection of forwared rows and lazy collection of deleted space

Sybase Control Center



Troubleshooting

Troubleshooting

1. **Run sp_sysmon to get a general look at server performance**
   - ✓ sp_sysmon “00:10:00”
2. **MDA is your friend**
   - ✓ Narrow down to the query in question
   - ✓ Examine waits, device io performance, cache usage, etc.
3. **Isolate and examine the query using set options**
4. **Use QP Metrics/QP Tune**
   - ✓ enable metrics capture
5. **Application Tracing**
   - ✓ connect to a spid and capture show output

A MDA Starter Config

Always on:

- Must have's:
  - enable monitoring = 1
  - wait event timing = 1
  - process wait events = 1
  - object lockwait timing = 1
  - SQL batch capture = 1
  - statement statistics active = 1
  - per object statistics active = 1
  - max SQL text monitored = 2048
- Nice to have's:
  - errorlog pipe active = 1
  - errorlog pipe max messages = 1000
  - deadlock pipe active = 1
  - deadlock pipe max messages = 100

As needed (dynamically):

- enable stmt cache monitoring = 1
- statement pipe active = 1
- statement pipe max messages = 25000

Only when truly desperate:

- Use sysquerymetrics?
- …Or enable the text pipe
  - Gets everything - not just bad queries
    - sql text pipe active = 1
    - sql text pipe max messages = 1000
- Use app tracing/sp_showplan instead? (or plan pipe)
  - plan text pipe active = 1
  - plan text pipe max messages = 1000

Some MDA Tables

Problem	MDA Tables
Currently Running Queries	monProcessStatement, monProcessObject
Finding Tempdb Hogs	monProcessObject
Previously Executed Queries/Proc Issues	monSysStatement
Proc Cache Optimization Hogs	monSysStatement, monProcessStatement
What is ASE waiting for?	monSysWaits
Find currently open objects	monOpenObjectActivity
SQL and Plan Text	monSysSQLText, monSysPlanText

Set option show….

- **set option show <normal/brief/long/on/off>**
  - ✓ Basic syntax common to all
  - ✓ Execute this one first prior to any of the below when used together (other wise it will reset the below to the level of detail for show command itself)

… show_lop <…>	**Shows logical operators used.**
… show_managers <…>	**Shows data structure managers used.**
… show_log_props <…>	**Shows the logical managers used.**
… show_parallel <…>	**Shows parallel query optimization.**
… show_histograms <…>	**Shows the histograms processed.**
… show_abstract_plan <…>	**Shows the details of an abstract plan.**
… show_search_engine <…>	**Shows the details of a search engine.**
… show_counters <…>	**Shows the optimization counters.**
… show_best_plan <…>	**Shows the details of the best QP plan.**
… show_code_gen <…>	**Shows the details of code generation.**
… show_pio_costing <…>	**Shows estimates of physical I/O**
… show_lio_costing <…>	**Shows estimates of logical input/output.**
… show_elimination <…>	**Shows partition elimination.**
… show_missing_stats <…>	**Shows columns with missing stats.**


Q&A

Useful links

- **SAP ASE Developer Center**
  - ✓ http://scn.sap.com/community/developer-center/oltp-db
- **SAP Sybase Education Curriculum**
  - ✓ https://training.sap.com/us/en/courses-and-curricula/sybase
- **Sybooks ASE 15.7 Documentation**
  - ✓ http://infocenter.sybase.com/help/topic/com.sybase.infocenter.help.ase.15.7.2/title.htm
- **SAP Notes specific to running SAP on ASE**
  - ✓ https://service.sap.com/sap/support/notes/1680803
  - ✓ https://service.sap.com/sap/support/notes/1722359
  - ✓ https://service.sap.com/sap/support/notes/1539125