SAS - When Big Memory Applications Meet Big Memory Machines

Maureen Chew, Oracle
Gary Granito, Oracle
Agenda

- Big Memory Utilization Opportunities for SAS
- Oracle Database 12c
- Conventional SAS Usage
  - Transparent Large Memory Utilization
  - Directed Large Memory Utilization
- SAS 9.4 High Performance Analytics
Big Memory Convergence with SAS

- Large product portfolio & diverse platform support
- Setting the Big Picture for Big Memory, Big Data
  - Separation of Data and Compute Platforms
  - Blurring of In-Memory and In-Database
COMPREHENSIVE ORACLE ARCHITECTURES FOR SAS PLATFORMS
FOCUS ON IN-MEMORY

Data Platform
- Exadata*
- SAS In-Database Embedded Processing (EP)
  - Parallel Data Feeders
  - Scoring Accelerator
- SuperCluster* (SSC - SPARC)
- Big Memory Machine (M5/M6)
- ZFS Storage Appliance*
- Big Data Appliance* (BDA)
- Shared Storage for SAS Grid Computing

In-Memory
- 12c In-Memory
- ACCESS

Compute Platform
- Big Data Appliance (BDA)
- Exalogic
- Oracle Virtual Compute Appliance (OVCA)
- SuperCluster (SSC - SPARC)
- Big Memory Machine (M5/M6)
- Oracle Linux
- Engineered Systems
- Oracle Solaris

- SAS/Hadoop
- SAS VA - Visual Analytics
- SAS HPA - High Performance Analytics (Distributed)
- SAS LASR – In Memory Analytic Server
- IMSTAT + LASR
- SAS HPA - High Performance Analytics (SMP)
- SAS LASR – In Memory Analytic Server
- SAS In-Database Embedded Processing (EP)
- Parallel Data Feeders
- Scoring Accelerator
- 12c In-Memory
- ACCESS
<table>
<thead>
<tr>
<th>Exadata Database Machine</th>
<th>Exalogic Elastic Cloud</th>
<th>Oracle Virtualized Compute Appliance (OVCA)</th>
<th>Big Data Appliance</th>
<th>SPARC SuperCluster</th>
</tr>
</thead>
<tbody>
<tr>
<td>RDBMS storage compression and database parallelization via “Exadata Storage Servers”</td>
<td>Extreme -performance I/O connecting large amount of compute power and memory</td>
<td>VM Server virtualization – runs Oracle Linux, Oracle Solaris, Windows. Software Defined Networking</td>
<td>Massive disk storage array with high-bandwidth I/O for loading ‘big’ data</td>
<td>SPARC servers, high-performance I/O and Exadata storage servers in one rack</td>
</tr>
</tbody>
</table>
SAS on Oracle Big Memory Machine

A 32 Socket / 32 TB High-End Datacenter Server

- **Compute**
  - Up to 32 x SPARC M5 6-core 3.6GHz CPUs
  - Up to 1024x DDR3 DIMMs for max memory of up to 32TB

- **I/O and storage**
  - 32 x 2.5” SAS-2 internal drives
  - 64x PCIe Gen3 low profile internal slots

- **Scalability and investment protection**
  - Upgradable with M6 processor
Oracle Database 12c – In-Memory

- Tony Baer, Ovum StraightTalk, Principal Analyst
  
  **How Oracle Database 12c embraces in-memory architecture**

  - Operational and transactional data platforms – historically separated from analytical stores – Natural evolution to blend the two
  - In-memory allows for reduction / elimination of analytical indexes which geometrically increase storage requirements
  - Analytical indexes speed query performance, but multiple indexes create OLTP performance cost
  - Differentiator: pairing with disk and in-memory based row store that will instantly replicate data to columnar tables

- Oracle Big Memory Machine – a sweet spot for Oracle Database 12c In-Memory
Conventional SAS – Transparent Benefit of Large Memory

- On Concurrency & Scalability
  - Multi-User
  - Multi-Threaded
  - Multi-User, Multi-Threaded

- MEMSIZE / SORTSIZE
  - User directed (in general)
  - Choice made w/o due consideration to overall impact
Conventional SAS – Directed Use/Benefit of Large Memory

- **SASFILE – Holding a File in Memory**
  - `sasfile mylibname.census open;`
  - `data test1; set mydata.census; run;`
  - `sasfile mydata.census close;`

- **SASWORK & TMPFS**
  - Performance increase can be dramatic
  - Is this different than FLASH drives?
  - Won’t work with Checkpoint/Restart; requires SASWORK persistence
SAS on Oracle Big Memory Machine

SAS Scale Up, Scale Out – proc LOGISTIC

**SAS Scalability Test**

- Compute Intensive proc LOGISTIC
- Add identical SAS Jobs
  - Observe Job Turnaround
- Scale Up: Single 48 core Physical Domain (PDOM)
- Scale Out: Single 48-job workload, run concurrently on 4 PDOMs – 192 jobs

**Testing Outcome**

- Excellent Scalability Results
- Single PDOM: 48 Jobs run in ~2X the time of 1 job.
  - 24X Work Accomplished – Scale Up
- Scaling to 4 PDOMs – 192 jobs in 2x the time of 1 job
  - 96X Work Accomplished – Scale Out

---

#SASGF14
SASWORK on TMPFS

SAS Scalability Testing with TMPFS

- Single 48 core Physical Domain (PDOM)
  - SASWORK SAN vs. TMPFS (In Memory File System) comparison
  - Scale Up Workload – run up to 64 concurrent jobs
  - System is oversubscribed at >48 jobs

- Testing Outcome
  - SAN performance is fine but throughput hits expected max
  - Avg step time remains nearly flat (red line) for TMPFS; exponential increase for SAN
  - Reducing SASWORK I/O using TMPFS enables excellent scalability
Full Scale Concurrency on Big Memory Machine

- SAS Global Trading Application – Scale Up & Scale Out
- Use TMPFS for SASWORK
- 256 CPU/memory intensive complex sas streams running concurrently. Each uses 750MB RAM + 80+GB I/O to TMPFS
SAS High Performance Analytics

- **SAS LASR Analytic Server**
  - In-Memory Analytic platform that allows for concurrent access to data loaded into memory

- **SAS High Performance Procedures**
  - `hpreg`, `hplogistic`, `hpreduce`, `hpsplit`, etc

- **Supports distributed (MPP) and non-distributed (SMP)**
SAS on Oracle Big Memory Machine

SAS High Performance Analytics (HPA) Scalability Results

- **SAS HPA Testing**
  - hpslit scale up test
  - THREADs/job=4
  - Oversubscribe 48core/2TB PDOM with up to 96 concurrent jobs
    - Observe Avg JobTime

- **Testing Outcome**
  - Excellent scalability results
  - At 48 jobs, avg time is ~2x single job
    - ~24X scalability
  - At 96 jobs, avg time is~3X single job
    - ~32X workload scalability
Why SAS on Big Memory Machine - Large Memory ....

- Supports New & Existing Features
- Enables IT Agility
- Reduces Performance Risk
  - Masks resource consumption chaos
  - Application Memory
  - Kernel Memory
  - File system caching
- Provides the best performance "Insurance"
Thank You

maureen.chew@oracle.com
gary.granito@oracle.com