IMPROVING PERFORMANCE & REDUCING CLOUD COSTS WITH 'FREE' BASE-LEVEL IN-MEMORY

David Kurtz

UKOUG 2023

/*+Go-Faster*/ Consultancy

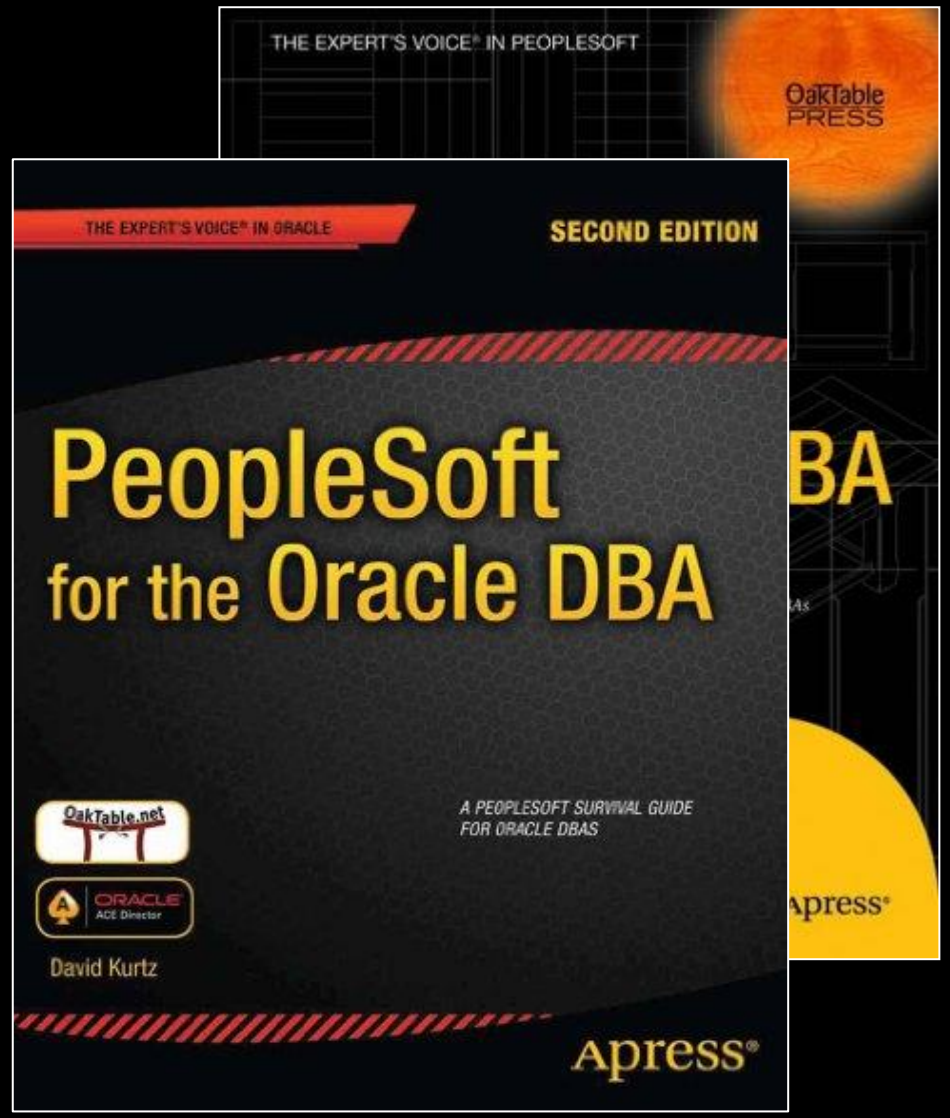
WHO AM I

Go-Faster Consultancy

- Performance tuning
  - Oracle RDBMS
  - PeopleSoft ERP
- www.go-faster.co.uk
- blog.go-faster.co.uk
- blog.psftdba.com

Oak Table

ORACLE ACE Director



PEOPLESOFT FINANCIALS GENERAL LEDGER

Data Warehouse of Financial Information

- **Every transaction produces two entries per ledger**
- **You may have multiple ledger, and there may be related adjustment ledger**
- **Each entry has a posted amount, base currency amount**
- **Dimensions**
  - **Fiscal Year, Accounting Period**
  - **Business Unit, Ledger, Account, Dept, Operating Unit, Product, Fund, Class, Program, Affiliate, Project, Book Code, Currency CD, Statistics Code, Chartfield 1-3**
- **Daily GL Reports (nVision, extracts to other DW)**
- **At month end, adjustment transactions, real-time reporting to see those adjustments**

TWO PEOPLESOFT FINANCIALS CUSTOMERS GENERAL LEDGER

US Based Insurance Company

- Financials 9.3 / PeopleTools 8.58
- nVision
  - Report mostly off Summary Ledgers
  - 4 fiscal years, >2 billion Ledger rows, 175Gb

- Oracle 19.9 -> 19.19
- Supercluster M7 -> ExaData C@C X9M2
- 2-node RAC
- 192Gb memory / compute node -> 1390Gb

- In production with Base Level In-Memory on SC
- Now in production full In-Memory on EXAC@C

Multi-National Bank

- Financials 9.3 / PeopleTools 8.58
- nVision
  - Reporting of Ledger
  - 7 fiscal years, 1.7 billion rows, 82Gb

- Oracle 19.11
- Exadata X8
- 3-node RAC
- 768Gb memory / compute node

- Testing did not show a sufficiently significant improvement to justify

PEOPLESOFT GENERAL LEDGER & SUMMARY LEDGERS

US Based Insurance Company

- In production with Base Level In-Memory
- Clear user benefit
  - Just IM cuts several hours of overnight reporting suite.
  - Reports available at start of continental European working day.
- Transitioned to Exadata Cloud @ Customer
  - Not BYOL so licence includes In-Memory and lots of other things.
  - Additional benefit from larger In-Memory store containing more segments.

Partition Ledger and Summary Ledger tables

- Partitioning: Fiscal Year & Accounting Period
- Sub-Partitioning: Ledger

HCC

- historical (static) partitions on LEDGER and LEDGER_BUDG
  - Incremental on-line rebuild
- MVs on Summary Ledgers, also partitioned
- Fully Fast Refreshed after SL refresh

In-Memory

- Even selected partitions of LEDGER and LEDGER_BUDG are too big
- Summary Ledger Partitions for current and/or previous fiscal year

HOW MUCH IN-MEMORY STORE?

- Base-level can only have 16Gb / node
  - All of it

Insurance Company
- Supercluster
  - Only 192Gb physical memory
  - Increased SGA by 8Gb
  - Decreased minimum buffer cache by 8Gb
  - Increased minimum shared pool by 2Gb
- ExaC@C
  - 1Tb allocated - not problem to grow SGA

Normally you would not steal memory from existing SGA

Bank
- 768Gb Physical memory
- No problem to grow SGA by 16Gb

IN-MEMORY PARAMETERS

```
inmemory_force='BASE_LEVEL'
```

Enables In-Memory without triggering licence tracking

Restrictions

- **16Gb In-Memory store per instance**
- **Compression levels other than QUERY LOW**
- **Excluded columns**
- **CELLMEMORY feature**
- **Automatic in-Memory**

IN-MEMORY PARAMETERS

```
inmemory_size=16G
sga_max_size / sga_target +16Gb
```

```
inmemory_optimized_arithmetic='ENABLE'
```

- Increased in-memory space overhead

```
inmemory_trickle_repopulate_servers_percent = 5
```

- Increased from default of 1.

```
_inmemory_64k_percent=10
```

- THIS IS AN UNDOCUMENTED PARAMETER

In-Memory Store is divided into

- 1Mb pool
  - data stored here
- 64Kb pool
  - metadata about DML changes
  - Observed 30% by default
  - Reduced to 10%

64Kb pool is very lightly used

- Releases 3.2Gb / node

WHICH PARTITIONS INTO INMEMORY

Insurance Company

- Most of the GL reporting runs off the summary ledgers
- Most of the queries rewrite to use the MVs

In-Memory Advisor

- We were not successful with it.
- It ran for ever! Probably a PeopleSoft effect.

ASH - segments with the highest I/O per Gb

- Identify whole fiscal years for certain SLs
- Work down the list until you fill In-Memory

Bank

- We can only consider the various ledger tables

- ASH - segments with the highest I/O per Gb
  - Current and previous 3 years of main ledger table
    - Main ledger subpartitions only (not adjustment ledgers).

- PL/SQL script to set certain partitions `INMEMORY/NO INMEMORY`
  - Each new fiscal year this moves on.

PARALLEL_FORCE_LOCAL & DUPLICATED IN-MEMORY STORE

In-Memory uses parallel query

`PARALLEL_FORCE_LOCAL=FALSE`

- Read In-Memory store on different RAC nodes

70-90% of 16Gb / node

Only option at Bank

- No summary ledgers
- Need to store LEDGER
- 3 x 90% x 16Gb = 43.2Gb

SuperCluster M8

`PARALLEL_FORCE_LOCAL=TRUE`

- Can't read remote in-memory store

In-Memory Duplicate

- Object in In-Memory store on each RAC node
- Can only do this on engineered system
- 70-90% of 1 x 16Gb

On supercluster - we got better performance

- Better than another 14.4Gb of partitions

PARALLEL_FORCE_LOCAL & DUPLICATED IN-MEMORY STORE

Exadata X9M

`PARALLEL_FORCE_LOCAL=FALSE`

- Better to add another 14.4Gb of in-memory segments

Interconnect

- RoCE network – 200Gbps

SuperCluster M8

`PARALLEL_FORCE_LOCAL=TRUE`

- Duplication and local parallel better than another 14.4Gb of partitions

Interconnect

- InfiniBand – 40Gbps

DISTRIBUTE BY SERVICE

Sometimes, the segments do not distribute evenly across the RAC nodes

- **One in-memory store fills up, while others have room to spare**

Distribute by service

- **Node specific service per node**
- **No failover/load-balancing**
  - **if we lose a node, no in-memory store has free space to fail over into**

Script to mark certain partitions in-memory

- **Specify service**
- **Hash / Round-Robin**

```
l_sql := l_sql
  ||' DISTRIBUTE FOR SERVICE '
  ||l_service_name
  ||dbms_utility.get_hash_value(
     i.subpartition_name,1,l_num_instances);
```

```
ALTER TABLE SYSADM.PS_LEDGER
MODIFY SUBPARTITION LEDGER_2021_05_ACTUALS
INMEMORY
MEMCOMPRESS FOR QUERY LOW
DISTRIBUTE FOR FINPRD1 DEFAULT
```

BANK TEST SYSTEM: LEDGER, BUT NO SUMMARY LEDGERS

Never went into production

Does not use summary ledgers

Only real candidate for In-Memory is PS_LEDGER

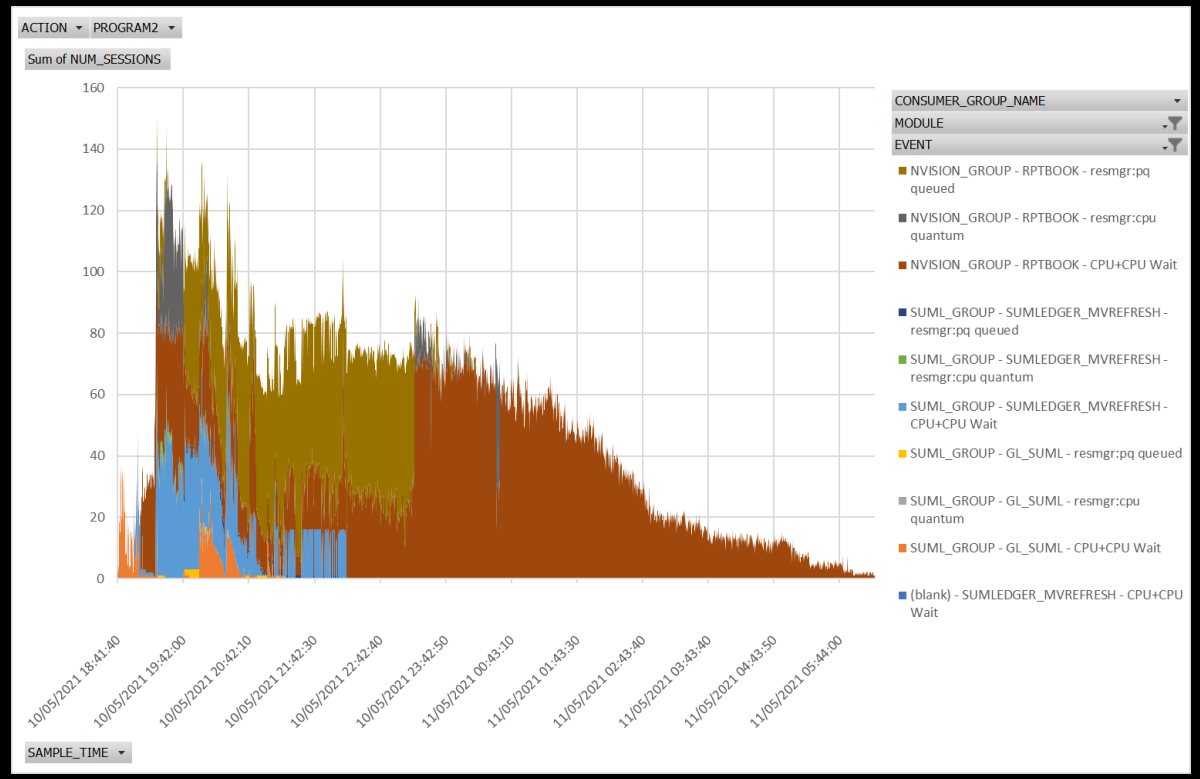
- Can fit 3 or 4 fiscal years

We did not seeing much improvement

- Reporting is less concentrated on ledger than would be on a summary ledger
- Legacy processes creating reporting tables
- I suspect because we are using indexes too often,
  - and not doing enough full scan/smart scan
  - Don’t want to use Exadata System Stats due to mixed workload.
- Bloom filter hash join requires equijoins between Ledger and Tree selector tables

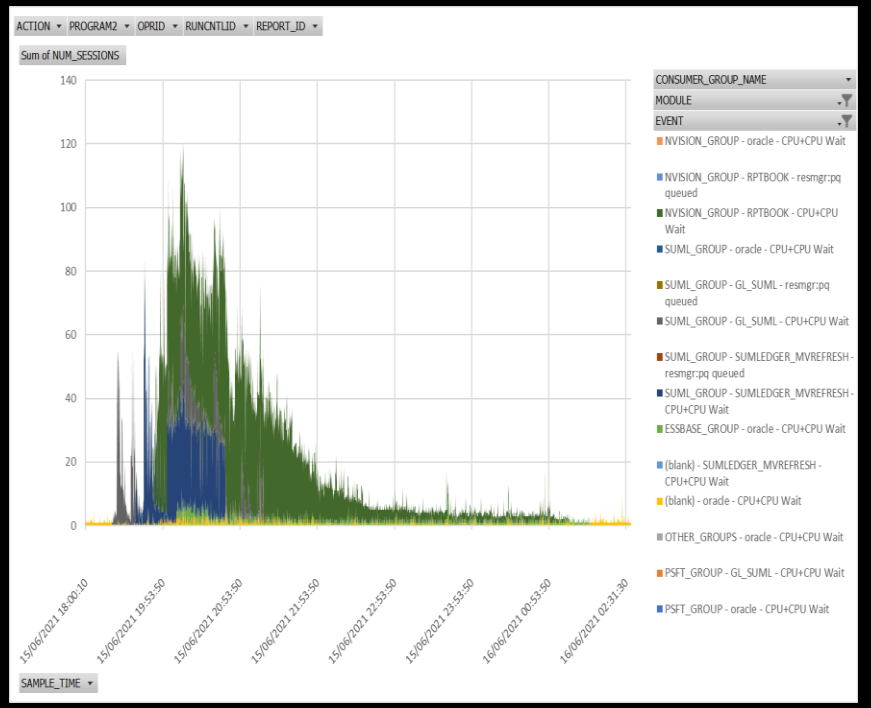
INSURANCE CO. SUPERCLUSTER TEST: COMPARISON OF DB TIME

Without in-memory – 18:41 - 05:44 = 11h3m



INMEMORY DUPLICATE – 18:00 - 1.08 = 7h8m

```
parallel_force_local=TRUE
inmemory_optimized_arithmetic='ENABLE'
```



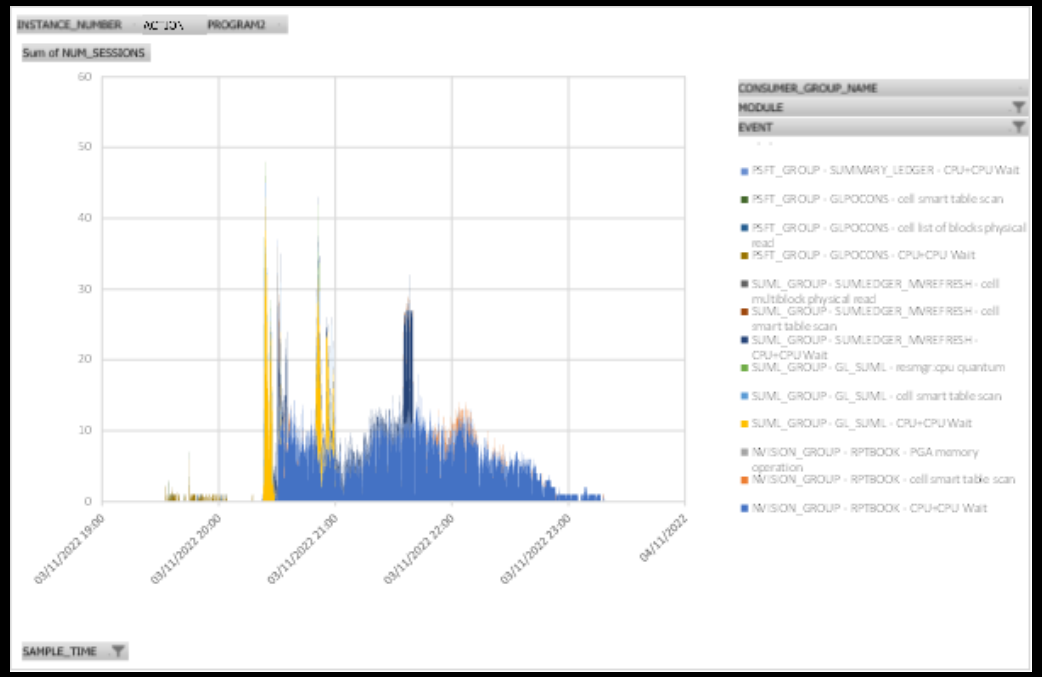
INSURANCE CO. X9M2 C@C TEST: COMPARISON OF DB TIME

Without in-memory – 20:39 – 0:36 = 3h57m

- CPU DB Time: 65900s

6 OCPU/node: concurrent reports 42 →20

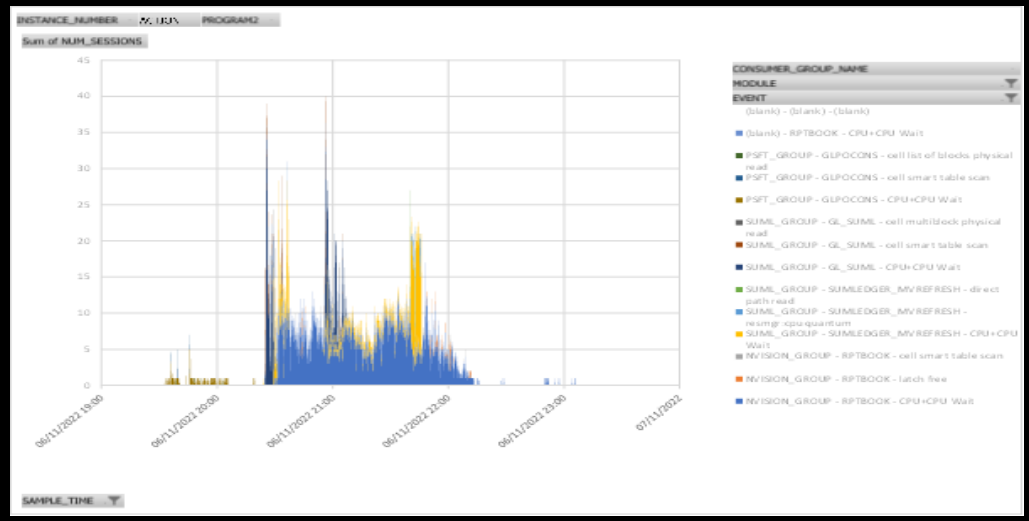
-64% w.r.t Supercluster



In-Memory – 20:30 – 23:06 = 2h36m

- CPU DB Time: 46030s
- -30% CPU reduction w.r.t. no in-memory

```
parallel_force_local=FALSE
Non-duplicated in-memory cache
2x 14.4Gb
```



INSURANCE CO. TEST SYSTEM: LEDGER IN (FULL) IN-MEMORY

Insurance Co. licenced for IM on ExaC@C

We tried putting PS_LEDGER into In-Memory

- We go latching problem during processes that made extensive changes to this table
- Could work around by setting INMEMORY_QUERY=FALSE at session level for specific processes that update table.
- We didn’t see much benefit to continue to examine this.

THE CLOUD INSTRUMENTS PERFORMANCE AS COST

Supercluster

- **Capital Expenditure**
- **Use all of it to obtain the best possible performance.**
- **80CPUs, 192Gb/instance**

Exadata C@C

- **Operational Expenditure**
- **Use what you need to achieve the performance you need to meet you business needs.**
  - **“You can have as much performance as you are willing to pay for”**
- **What are your non-functional requirement?**
  - **“Reports available in EU at 8am CET”**
- **With full In-Memory 6 OCPUs/node**
  - **Without In-Memory it would be 10 OCPUs/node**
  - **With base-level it might have been 7 OCPUs/node**

ACHIEVEMENTS

Reduced

- Elapsed Time
- DB Time – mostly CPU time
- Peak CPU
- Cloud Costs

Your mileage will vary

- Even the same packaged application was very different at different customers

Test one thing at a time

While the system is on-premises

- We are happy to use all the CPU available

When it is in the cloud

- limit Peak CPU further
  - by restricting concurrency
- accept some reports running/finishing later
  - that might have a bearing on a shared/cloud configuration,
  - and that has a bearing on costs