Exadata Resource Management: Achieve Mind-blowing Performance, Consistently
UTOUG Fall Symposium 2015

Kasey Parker
Enterprise Architect
Kasey.Parker@centroid.com
Centroid Overview

Leader in Technology, Consulting and Managed Services since 1997

- Part of Oracle’s Top 25 Strategic Partner Program
- Focused on Consulting, Managed Services, Cloud Services and Resell
- Capabilities From Oracle Applications to Technology to Infrastructure

Specializations
Oracle Applications
Oracle Database & Core Technologies
Oracle Engineered Systems
Oracle Server & Storage Solutions
Oracle Performance Tuning
Oracle Data Warehousing
Oracle Business Intelligence
About Kasey Parker

QUICK FACTS

- Reside in Utah with wife and 5 kids
- Oracle DBA / Architect
  - 15 Years of Oracle Experience
  - Oracle Certified Exadata specialist
  - Performance Tuning specialist
  - Data Warehousing specialist
  - OCP DBA
  - Last 5 years focused on Oracle Engineered Systems
- Many industries
  - Financial Services, Manufacturing, Health/Nutrition, City/County Government, Retail
- Academic
  - Brigham Young University Alumni
    - Management Information Systems
Agenda

- Exadata Overview
- Why Exadata?
- Exadata Resource Management
  - IORM / DBRM
  - Architecture
  - Implementation
  - Monitoring
  - New in Exadata X5
  - Benefits / Case Study
EXADATA OVERVIEW
**Exadata Architecture X5-2**

**Complete | Optimized | Fully Redundant | Scale-Out**

**Scale-Out Database Servers**
- 8x 2-socket, or 2x 8-socket database servers
- Fastest Xeon 18-core chips, 256 to 768 GB DRAM
- Oracle Database, ASM, RAC, Oracle Linux

**Scale-Out, 2-socket Intelligent Storage Servers**
- 16 Xeon cores/server enables DB offload to storage
- **Extreme Flash Storage** ➔ 8x 1.6TB PCI Flash Drives
  or
- **High Capacity Storage** ➔ 4x 1.6 TB PCI Flash Cards + 12 x 4 TB SAS disks

**High-Speed InfiniBand Network**
- Unified internal connectivity (40 Gb/sec)
- 10 Gb or 1 Gb Ethernet data center connectivity
Elastically Scale-Out
from Eighth-Rack to Multi-Rack

- Start with 2 Database Servers and 3 Storage Servers
  - Add Database or Storage Servers as needed
- Expand older machines with new generation servers
Workload Optimized Configurations

**DB In-Memory Machine**
Wants many DB Servers
few Storage Servers

**Extreme Flash OLTP Machine**
All-flash IOPs enables
capacity based OLTP sizing

**Data Warehouse Machine**
More High Capacity Storage for
longer data retention

- **DB In-Memory Machine**
  - 16 Database Servers
  - +
  - 5 High Capacity Storage Servers

- **Extreme Flash OLTP Machine**
  - 8 Database Servers
  - +
  - 8 Extreme Flash Storage Servers

- **Data Warehouse Machine**
  - 8 Database Servers
  - +
  - 14 High Capacity Storage Servers
## Exadata Hardware Summary

<table>
<thead>
<tr>
<th></th>
<th>X5-2 Full</th>
<th>X5-2 Half</th>
<th>X5-2 Quarter</th>
<th>X5-2 Eighth</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database Servers</td>
<td>8</td>
<td>4</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>Database Grid Cores</td>
<td>288 (min 112)</td>
<td>144 (min 56)</td>
<td>72 (min 28)</td>
<td>36 (min 16)</td>
</tr>
<tr>
<td>Database Grid Memory (GB)</td>
<td>2048 (max 6144)</td>
<td>1024 (max 3072)</td>
<td>512 (max 1536)</td>
<td>512 (max 1536)</td>
</tr>
<tr>
<td>InfiniBand switches</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>Ethernet switch</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Exadata Storage Servers</td>
<td>14</td>
<td>7</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>Storage Grid CPU Cores</td>
<td>224</td>
<td>112</td>
<td>48</td>
<td>24</td>
</tr>
<tr>
<td>Raw Flash Capacity</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EF</td>
<td>179.2 TB</td>
<td>89.6 TB</td>
<td>38.4 TB</td>
<td>19.2 TB</td>
</tr>
<tr>
<td>HC</td>
<td>89.6 TB</td>
<td>44.8 TB</td>
<td>19.2 TB</td>
<td>9.6 TB</td>
</tr>
<tr>
<td>Raw Storage Capacity</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EF</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>HC</td>
<td>672 TB</td>
<td>336 TB</td>
<td>144 TB</td>
<td>72 TB</td>
</tr>
<tr>
<td>Usable mirrored capacity</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EF</td>
<td>80 TB</td>
<td>40 TB</td>
<td>16.8 TB</td>
<td>8 TB</td>
</tr>
<tr>
<td>HC</td>
<td>300 TB</td>
<td>150 TB</td>
<td>63 TB</td>
<td>30 TB</td>
</tr>
<tr>
<td>Usable Triple mirrored capacity</td>
<td>53.3 TB</td>
<td>26.8 TB</td>
<td>11.5 TB</td>
<td>5.7 TB</td>
</tr>
<tr>
<td>EF</td>
<td>200 TB</td>
<td>100 TB</td>
<td>43 TB</td>
<td>21.5 TB</td>
</tr>
<tr>
<td>HC</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
# Exadata Hardware Summary Cont’d

## Exadata X5-2 SQL IO Performance

<table>
<thead>
<tr>
<th>Flash Cache SQL Bandwidth¹,³</th>
<th>Extreme Flash</th>
<th>X5-2 Full Rack</th>
<th>X5-2 Half Rack</th>
<th>X5-2 Quarter</th>
<th>X5-2 Eighth</th>
</tr>
</thead>
<tbody>
<tr>
<td>High Capacity</td>
<td>140 GB/s</td>
<td>100 GB/s</td>
<td>43 GB/s</td>
<td>21.5 GB/s</td>
<td></td>
</tr>
</tbody>
</table>

| Flash SQL IOPS²,³           | 8K Reads      | 4,140,000      | 2,070,000     | 887,143      | 443,571     |
|                            | 8K Writes     | 4.14M / 2.69M  | 2.07M / 1.35M | 887K / 579K  | 444K / 289K |

| Disk SQL Bandwidth¹,³       | Extreme Flash | N/A            | N/A           | N/A          | N/A         |
| High Capacity               | 20 GB/s       | 10 GB/s        | 4.5 GB/s      | 2.25 GB/s    |             |

| Disk SQL IOPS²,³            | Extreme Flash | N/A            | N/A           | N/A          | N/A         |
| High Capacity               | 32,000        | 16,000         | 7,000         | 3,500        |

---

1 - Bandwidth is peak physical scan bandwidth achieved running SQL, assuming no compression. Effective data bandwidth will be much higher when compression is factored in.

2 - IOPS – Based on read IO requests of size 8K running SQL, typically with sub-millisecond latencies. Note that the IO size greatly effects flash IOPS. Others quote IOPS based on 2K, 4K or smaller IOs that are not relevant for databases and measure IOs using low level tools instead of SQL.

3 - Actual Performance varies by application.
WHY EXADATA?
Why Exadata?

Exadata designed to eliminate the most common bottleneck for large databases...

IO performance from storage to database
Why Exadata?

Solving IO Bottleneck

Solution 1: Enlarge Pipe

- Physical disks, on all cells, work in parallel to serve IO requests
- Large Infiniband pipe (40GB/Sec)
Why Exadata?

Can’t we do that with other high-performance storage solutions?

YES...

Nothing Magical about Exadata hardware, and it’s still the same Oracle Database
Why Exadata?

Solving IO Bottleneck

Solution 2: Reduce IO operations

- Exadata’s Secret Sauce: Storage Offloading, Smart Flash Cache and Hybrid Columnar Compression (HCC)
- 10X reduction in data sent to database servers common
Why Exadata?

Exadata’s Secret Sauce

**Intelligent storage**
- Scale-out InfiniBand storage
- Smart Scan query offload

![Intelligent storage diagram](image)

**Hybrid Columnar Compression**
- 10x compression for warehouses
- 15x compression for archives

![Hybrid Columnar Compression diagram](image)

**Smart PCI Flash Cache**
- Accelerates random I/O up to 30x
- Triples data scan rate

![Smart PCI Flash Cache diagram](image)

Benefits Cascade to Copies

- Standby
- Test
- Dev
- Backup

Data remains compressed for scans and in Flash

![Benefits Cascade diagram](image)
EXADATA
RESOURCE MANAGEMENT
IORM & DBRM
Exadata IO Resource Management

• IO Resource Manager (IORM)
  – Governs / Meters IO from different workloads on Exadata

• Competing IO workloads
  – Common challenge with shared storage
  – Batch vs. OLTP
  – Warehouse vs. OLTP
  – Production vs. Test and Development

• Typical Mitigation Strategy
  – Over-provisioning Resources – Becomes Expensive

• Exadata addresses with IORM
IORM and DBRM

- Oracle Database Resource Manager (DBRM)
  - Used for competing workloads in a single database
  - Manages CPU and other DB resources, e.g. parallelism
  - Not Exadata Specific

- Exadata IORM / DBRM Integration
  - IO resources also controlled by DBRM
  - Consumer Group Categories managed across DBs

- DBRM resource plan = “intra-database” plan
IORM Plans

IO Resource Management

Inside One Database

Intradatabase Resource Plan (DBRM)

Across Multiple Databases

Interdatabase Resource Plan

Category Resource Plan

IORM Plan
IORM Architecture

- Managed Services
- Cloud Services
- Consulting Services
- Licensing
iDB message contains DB name, consumer group, category, IO type

PROD Cell Disk 1
FIN_OLTP
FIN_REPORTS
SHIPPING_OLTP
OTHER_GROUPS
HIGH
MEDIUM
LOW

PROD Cell Disk 2

PROD Cell Disk 3

PROD Cell Disk N

DWPRD Cell Disk 7
DW_CRITICAL
DW_NORMAL
OTHER_GROUPS
HIGH
MEDIUM
LOW

DWPRD Cell Disk 2

DWPRD Cell Disk 3

DWPRD Cell Disk N

DEV Cell Disk 11
DEV_ALL
OTHER_GROUPS
HIGH
MEDIUM
LOW

DEV Cell Disk 2

DEV Cell Disk 3

DEV Cell Disk N

Disk 1
Disk 3
Disk 5
Disk 7
Disk 9
Disk 11
Disk 2
Disk 4
Disk 6
Disk 8
Disk 10
Disk 12

Exadata Cell Disks

CellSRV
IORM Architecture Cont’d

PROD Cell disk 1
FIN OLTP
FIN REPORTS
SHIPPING OLTP
OTHER_GROUPS
HIGH
MEDIUM
LOW

DWPRD Cell Disk 7
DW_CRITICAL
DW_NORMAL
OTHER_GROUPS
HIGH
MEDIUM
LOW

DEV Cell Disk 11
DEV_ALL
OTHER_GROUPS
HIGH
MEDIUM
LOW

PROD Cell disk N

DWPRD Cell Disk N

DEV Cell Disk N

CellsRV

Disk 1
Disk 3
Disk 5
Disk 7
Disk 9
Disk 11

Disk 2
Disk 4
Disk 6
Disk 8
Disk 10
Disk 12

Exadata Cell Disks
IORM Example

DBRM / Intra-database Plan Example

**Database DBM**
- OM OLTP Consumer group
- Other OLTP Consumer group
- Reporting Consumer group

**Database XBM**
- Online query Consumer group
- Batch query Consumer group
IORM Example Cont’d

Category Plan Example

Database DBM
- OM OLTP Consumer group
- Other OLTP Consumer group
- Reporting Consumer group

Database XBM
- Online query Consumer group
- Batch query Consumer group

Interactive category
Batch category
IORM Example Cont’d

• **DBM Intra-database Resource Plan**
  – 50% of resources allocated to “OM OLTP”
  – 30% of resources allocated to “OTHER OLTP”
  – 20% of resources allocated to “REPORTING”

• **XBM Intra-database Resource Plan**
  – 70% of resources allocated to “ONLINE QUERY”
  – 30% of resources allocated to “BATCH QUERY”

• **Category Plan**
  – 70% of resources allocated to INTERACTIVE category
  – 30% of resources allocated to BATCH category

• **Inter-database Plan**
  – 60% of resources allocated to database DBM
  – 40% of resources allocated to database XBM
IORM Example Cont’d

![Diagram showing category plan, interdatabase plan, intradatabase plan, and IORM allocation with percentages for different categories such as DBM: OM OLTP, OTHER OLTP, XBM: ONLINE QUERY, DBM: REPORTING, and XBM: BATCH QUERY. All User IO = 100%.]
IORM – Understanding the Math

Methods are evaluated in a fixed order:
1. Category plans first
2. Inter-Database plans next
3. Intra-Database plans last

\[ \text{CG\%} = \left( \frac{\text{Intra CG\%}}{\text{sum}(X)} \right) \times \text{DB\%} \times \text{Cat\%} \]

- **“CG\%”** = IORM Determined resource allocation for consumer group sessions
- **“Intra CG\%”** = Resource allocation for consumer group in an Intra-database plan
- **“X”** = sum of Intra-database consumer group allocations for all consumer groups in the same category and database
- **“DB\%”** = percent of database allocation in the Inter-database plan
- **“Cat\%”** = percent of resource allocation for the category in which the consumer group belongs
IORM Rules

• IORM only “engaged” when needed
  – IO to cell disks is saturated

• Leftover disk allocation
  – Dispersed across other configured resource plans
  – Max limits available

• Background IO prioritized relative to user IO
  – Redo and control file writes always take precedence
  – DBWR writes scheduled at same priority as user IO

• OTHER_GROUPS consumer group
  – Default if no intra-database plan set

• Flash Disk IO
  – Prioritization and Min / Max Limits Governed (New)
IORM Implementation

1. Set IORM Objective
   • To something other than basic

2. Create IORM Plan
   • DB Plan (inter-database) and/or Category Plan

3. Create Intra-database (DBRM) Resource Plan(s)
   • Create Consumer Groups
     • Consumer Group Categories = IORM Categories
   • Map Sessions to Consumer Groups

4. Monitor IORM Metrics
IORM Objective

- **Basic (Default)**
  - Minimal management – prevents extreme I/O latencies
  - Resource Plans not enforced

- **Low_latency**
  - Minimize latency by limiting concurrent I/O requests
  - Useful for critical OLTP workloads
  - Performance of high-throughput workloads may suffer

- **High_throughput**
  - Maximizes throughput by not limiting concurrent I/O requests
  - Useful for batch and data warehouse workloads
  - Performance of latency-critical workloads may suffer

- **Balanced**
  - Balance low disk latency and high throughput
  - Useful for mixed workloads

- **Auto**
  - Objective set dynamically based on active plans and workloads
IORM Plan Syntax

Create Using CELLCLI / DCLI / EM12c

CellCLI> alter iormplan
>     dbplan= ((name=sales_prod, level=1, allocation=80),
>           (name=finance_prod, level=1, allocation=20),
>           (name=sales_dev, level=2, allocation=100),
>           (name=sales_test, level=3, allocation=50),
>           (name=other, level=3, allocation=50)),
>     catplan=''

<table>
<thead>
<tr>
<th>Database</th>
<th>Level 1</th>
<th>Level 2</th>
<th>Level 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>sales_prod</td>
<td>80%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>finance_prod</td>
<td>20%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>sales_dev</td>
<td></td>
<td>100%</td>
<td></td>
</tr>
<tr>
<td>sales_test</td>
<td></td>
<td></td>
<td>50%</td>
</tr>
<tr>
<td>other</td>
<td></td>
<td></td>
<td>50%</td>
</tr>
</tbody>
</table>

Be careful with multiple levels!
IORM Plan Syntax Cont’d

Using DCLI

• DCLI Script

```shell
# cellcli -e alter iormplan inactive
# cellcli -e alter iormplan objective="auto",dbplan="((name=dbm,level=1,allocation=60)),((name=xbm,level=1,allocation=40)),((name=other,level=2,allocation=100))",catplan="((name=INTERACTIVE,level=1,allocation=70)),((name=BATCH,level=1,allocation=30)),((name=other,level=2,allocation=100))"
# cellcli -e alter iormplan active

[oracle@cm01dbm01 iorm]$ dcli -g ..,/cell_group -x ./iorm.scr
cm01cel01: IORMPLAN successfully altered
cm01cel01: IORMPLAN successfully altered
cm01cel01: IORMPLAN successfully altered
cm01cel02: IORMPLAN successfully altered
cm01cel02: IORMPLAN successfully altered
cm01cel02: IORMPLAN successfully altered
cm01cel03: IORMPLAN successfully altered
cm01cel03: IORMPLAN successfully altered
cm01cel03: IORMPLAN successfully altered
```

• Execute on a compute node with DCLI command
IORM Monitoring

IORM Metrics using CELLCLI / DCLI

- Metric Groupings
  - Category IORM
    - objectType='IORMCATEGORY'
    - name like 'CT_.*' 
  - Inter-Database IORM
    - objectType='IORMDATABASE'
    - name like 'DB_.*' 
  - Consumer group IORM (Intra-database)
    - objectType = ‘IORM_CONSUMER_GROUP’
    - name like ‘CG_.*’ 

- Current or Historical
  - LIST METRICCURRENT / LIST METRICCHISTORY

- Separated by Large (> 128KB) and Small (<=128KB)
IORM Monitoring Cont’d

IORM Metrics using CELLCLI / DCLI

- Display Metrics and Definitions
  - `# cellcli -e list metricdefinition where objectType='IORM_DATABASE'`
  - `IORM_CATEGORY, IORM_CONSUMER_GROUP`

- Examples

<table>
<thead>
<tr>
<th>Metric Name</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB_IO_RQ_SM</td>
<td>Total number of IO requests issued by database since any resource plan was set</td>
</tr>
<tr>
<td>DB_IO_RQ_LG</td>
<td>Total number of IO requests issued by database since any resource plan was set</td>
</tr>
<tr>
<td>DB_IO_RQ_SM_SEC</td>
<td>IO requests per second issued by database in last minute</td>
</tr>
<tr>
<td>DB_IO_RQ_LG_SEC</td>
<td>IO requests per second issued by database in last minute</td>
</tr>
<tr>
<td>DB_IO_WT_SM</td>
<td>Total number of seconds IO requests, issued by database, waited to be scheduled</td>
</tr>
<tr>
<td>DB_IO_WT_LG</td>
<td>Total number of seconds IO requests, issued by database, waited to be scheduled</td>
</tr>
</tbody>
</table>
Which database has heaviest load?

```
[oracle@cm01dbm01 ws8]$ sh ./lab63_1.sh
Show Small (< 128k) IO Requests per database
dcli -g ./cell_group cellcli -e list metric current where name=DB_IO_RQ_SM
  cm01cel02: DB_IO_RQ_SM  DBM  12,092 IO requests
  cm01cel02: DB_IO_RQ_SM  DBUA0  7,315 IO requests
  cm01cel02: DB_IO_RQ_SM  XBM  2,381,611 IO requests
  cm01cel03: DB_IO_RQ_SM  DBM  6,438 IO requests
  cm01cel03: DB_IO_RQ_SM  DBUA0  7,526 IO requests
  cm01cel03: DB_IO_RQ_SM  XBM  2,350,380 IO requests
Show Large (> 128k) IO Requests per database
  dcli -g ./cell_group cellcli -e list metric current where name=DB_IO_RQ_LG
  cm01cel02: DB_IO_RQ_LG  DBM  6 IO requests
  cm01cel02: DB_IO_RQ_LG  DBUA0  1 IO requests
  cm01cel02: DB_IO_RQ_LG  XBM  148,833 IO requests
  cm01cel03: DB_IO_RQ_LG  DBM  24 IO requests
  cm01cel03: DB_IO_RQ_LG  DBUA0  1 IO requests
  cm01cel03: DB_IO_RQ_LG  XBM  147,652 IO requests
```

IORM Monitoring Cont’d

Metric IORM script

• Download from MOS Note:
  – Tool for Gathering I/O Resource Manager Metrics: metric_iorm.pl [ID 1337265.1]

• Displays current metrics or metrics from past time period

• Provides metrics showing
  – Resources consuming most IO
  – Disk latencies
  – Flash cache hit metrics per database
  – Overall IORM plan impact on performance

• Works regardless of whether IORM plan is active
ORM Monitoring Cont’d

Monitoring with EM12c Exadata Plugin
New Flash Cache Management

- Flash Cache Resource Management
  - Prior to 12.1.2.1 (X5): enable/disable flash cache by DB
  - Now can set Min and Max Flash Cache by DB
    
    ```sql
    ALTER IORMPLAN dbplan=(
        (name=sales, flashCacheMin = 100M,
        (name=finance, flashCacheLimit = 256M))
    ```

- Container DB (CDB) limits specified directly
  - Pluggable DB limits specified as percent of CDB

- IORM Flash Prioritization Enhancement
  - Flash OLTP IO auto-prioritized over flash reporting IO
IORM Benefits – Case Study

EDW for Large Organization in Utah

• Production and Pre-prod DB Share Exadata ¼ Rack
• IOPS Capacity being Reached
• ETL Workload Competing with Report Workload
• Inconsistent ETL and Report times
  – One nightly ETL job ranging from 30 minutes to 3 hours
• Intra-database and Category Plans Created
  – Segregate Priorities for ETL, Reporting and Critical Query workloads
• Inter-database Plan Created
  – Segregate Production and Pre-production Database Workload
IORM Benefits – Case Study Cont’d

3.5 Days Before and After Enabling IORM/DBRM Plans

Top Timed Events

• Events with a “-” did not make the Top 1st in this set of snapshots, but are displayed for comparison purposes
• Summary for all instances

<table>
<thead>
<tr>
<th>Rank</th>
<th>Event Type</th>
<th>Event Description</th>
<th>Wait Class</th>
<th>Waits</th>
<th>Time(s)</th>
<th>Avg Time(ms)</th>
<th>%DB time</th>
<th>%CPU time</th>
</tr>
</thead>
<tbody>
<tr>
<td>1st</td>
<td>CPU time</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2nd</td>
<td>CPU time</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Average Active Sessions - 7 Day View

[Graph showing average active sessions over a 7-day period with CPU usage highlighted]
IORM on Exadata

Unless you have only one database with a single type of workload – then you should use IORM...

Nearly Everyone using Exadata should be using IORM!
8 Rules to be Successful with IORM

1. Research - Understand how IORM and DBRM work
2. Plan
   – Understand Business performance requirements
   – Map out IORM and DBRM plans
3. Know impact of resource plans – Understand the math
4. Automate Implementation
   – Create scripts to build DBRM and IORM plans
   – Helps understand what you’re doing, facilitates flexibility and control
5. Monitor IORM - Understand how and what key metrics represent
6. Be prepared to change as business conditions change
7. Explain recommendations to Business and IT in layman’s terms
8. Don’t be afraid – it’s easy, logical, and easy to undo
Questions?