Oracle Exadata Deep Dive

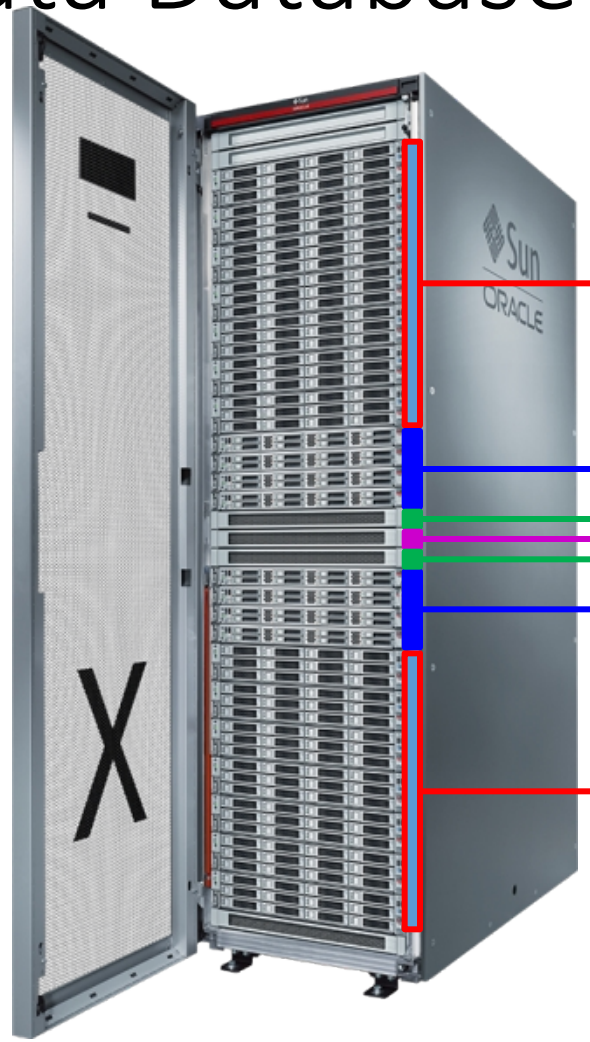
Karan Dodwal

Senior Oracle DBA

Exadata Database Machine

- Fully integrated platform for Oracle Database
- Based on Exadata Storage Server storage technology
- High-performance and high-availability for all Oracle Database workloads
- Balanced hardware configurations
- Scale-out architecture
- Well suited for cloud and database consolidation platform
- Simple and fast to implement

Exadata Database Machine X6-2 Full Rack



14 Exadata Storage Servers
(all High Capacity or all Extreme Flash)

8 Database Servers

2 QDR (40Gb/s) InfiniBand Switches (36 port)

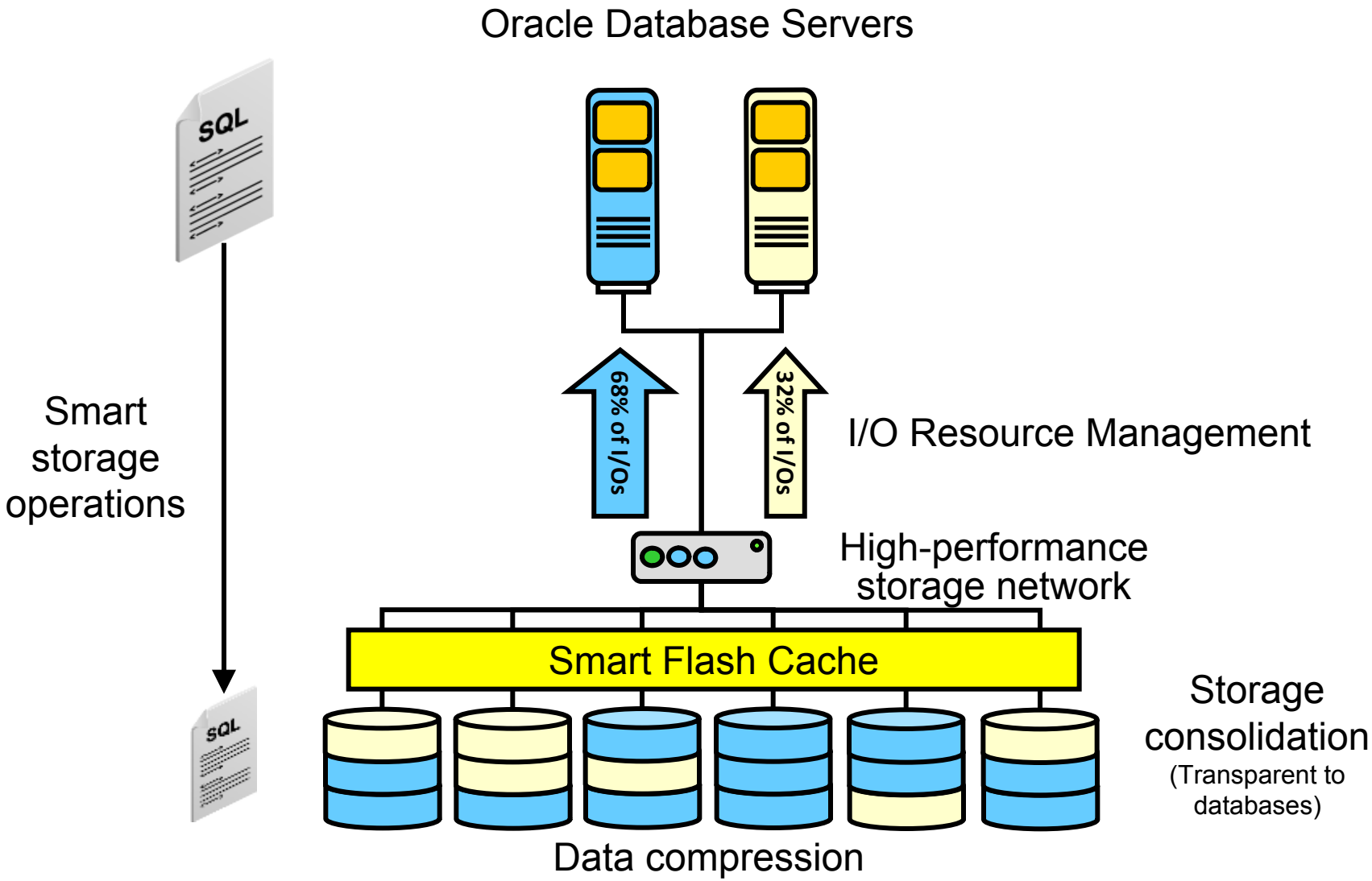
1 Cisco Ethernet Switch (48 port)

2 Power Distribution Units (PDUs) at rear

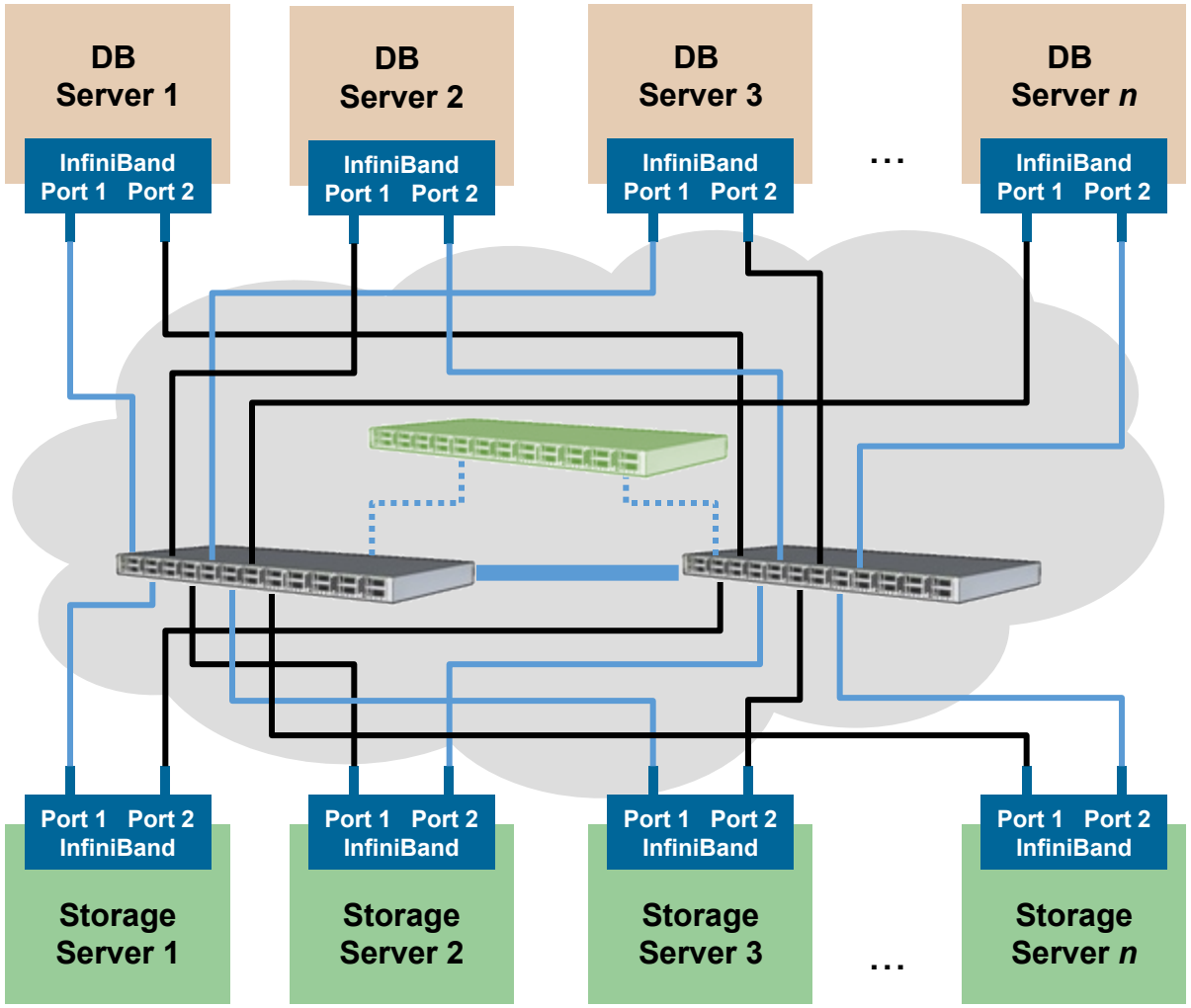
Spare Parts Kit:

- High Capacity: 1 disk drive and 1 flash card
- Extreme Flash: 1 flash drive

Exadata Storage Server Features



InfiniBand Network



Exadata X6-2 High Capacity Storage Server

ORACLE EXADATA

Processors	20 Intel CPU Cores 2 x Ten-Core Intel Xeon E5-2630 v4 (2.2 GHz)
System Memory	128 GB DDR4 Memory (8 x 16 GB)
Disk Drives	96 TB 12 x 8 TB 7,200 RPM High Capacity SAS Disk Drives
Flash	12.8 TB 4 x 3.2 TB Sun Flash Accelerator F320 NVMe PCIe Cards
Disk Controller	Disk Controller Host Bus Adapter with 1GB Write Cache
InfiniBand Network	Dual-Port QDR (40Gb/s) InfiniBand Host Channel Adapter
Remote Management	Integrated Lights Out Manager (ILOM) Ethernet Port
Power Supplies	2 x Redundant Hot-Swappable Power Supplies

Exadata X6-2 Extreme Flash Storage Server

ORACLE
EXADATA

Processors	20 Intel CPU Cores 2 x Ten-Core Intel Xeon E5-2630 v4 (2.2 GHz)
System Memory	128 GB DDR4 Memory (8 x 16 GB)
Flash Drives	25.6 TB 8 x 3.2 TB Sun Flash Accelerator F320 NVMe PCIe Drives
InfiniBand Network	Dual-Port QDR (40Gb/s) InfiniBand Host Channel Adapter
Remote Management	Integrated Lights Out Manager (ILOM) Ethernet Port
Power Supplies	2 x Redundant Hot-Swappable Power Supplies

Exadata Database Machine X6-2 Database Server

Processors	44 Intel CPU Cores 2 x 22-Core Intel Xeon E5-2699 v4 Processors (2.2GHz)
System Memory	256 GB (Expandable to 768 GB)
Disk Drives	4 x 600 GB 10K RPM SAS Disk Drives (Expandable to 8 Disks)
Disk Controller	Disk Controller Host Bus Adapter with 1GB Write Cache
Network Interfaces	• Dual-Port QDR (40Gb/s) InfiniBand Host Channel Adapter • Four 1/10 Gb Ethernet Ports (copper) • Two 10Gb Ethernet Ports (optical)
Remote Management	Integrated Lights Out Manager (ILOM) Ethernet Port
Power Supplies	2 x Redundant Hot-Swappable Power Supplies

Exadata Smart Scan: Overview

- Smart Scan includes:

  - Full Table and Fast Full Index Scans: Scans are performed inside Exadata Storage Server, rather than transporting all the data to the database server.
  - Predicate filtering: Only the requested rows are returned to the database server, rather than all the rows in a table.
  - Column filtering: Only the requested columns are returned to the database server, rather than all the table columns.
  - Join filtering: Join processing using Bloom filters are offloaded to Exadata Storage Server.

Smart Scan Requirements

- Smart Scan is not governed by the optimizer, but it is influenced by the results of query optimization.
  - Query-specific requirements:
    - Smart Scan is possible only for full segment scans.
    - Smart Scan can only be used for direct-path reads:
      - Direct-path reads are automatically used for parallel queries.
      - Direct-path reads may be used for serial queries:
        - They are not used by default for small serial scans.
        - Use `_serial_direct_read=TRUE` to force direct-path reads.
  - Additional general requirements:
    - Smart Scan must be enabled within the database.
    - Segments must be stored in appropriately configured disk groups.

Situations Preventing Smart Scan

- Smart Scan cannot be used in these circumstances:
  - Scan on a clustered table
  - Scan on an index-organized table
  - Fast full scan on a compressed index
  - Fast full scan on a reverse key indexes
  - Table has row-level dependency tracking enabled
  - `ORA_ROWSCN` pseudocolumn is being fetched
  - Optimizer wants the scan to return rows in `ROWID` order
  - Command is `CREATE INDEX` using `NOSORT`
  - `LOB` or `LONG` column is being selected or queried
  - `SELECT ... VERSIONS` flashback query is being executed
  - More than 255 columns are referenced in the query
  - Data is encrypted and cell-based decryption is disabled
  - To evaluate a predicate based on a virtual column

Smart Scan Execution Plan: Example

```
SQL> explain plan for select count(*) from customers where cust_valid = 'A';

Explained.

SQL> select * from table(dbms_xplan.display);

-------------------------------------------------------------------------------
| Id  | Operation                  | Name      | Rows  | Bytes | Cost (%CPU)|
-------------------------------------------------------------------------------
|   0 | SELECT STATEMENT           |           |     1 |     2 |   627K  (1)|
|   1 |  SORT AGGREGATE            |           |     1 |     2 |            |
|*  2 |   TABLE ACCESS STORAGE FULL| CUSTOMERS |    38M|    73M|   627K  (1)|
-------------------------------------------------------------------------------

Predicate Information (identified by operation id):
---------------------------------------------------

   2 - storage("CUST_VALID"='A')
       filter("CUST_VALID"='A')
```

Example of a Situation Preventing Smart Scan

```
SQL> explain plan for select count(*) from cust_iot where cust_id > '10000';

Explained.

SQL> select * from table(dbms_xplan.display);


-------------------------------------------------------------------------------
| Id  | Operation          | Name    | Rows  | Bytes | Cost (%CPU)| Time     |
-------------------------------------------------------------------------------
|   0 | SELECT STATEMENT   |         |     1 |    13 | 21232   (1)| 00:04:15 |
|   1 |  SORT AGGREGATE    |         |     1 |    13 |            |          |
|*  2 |   INDEX RANGE SCAN| CUST_PK |    86M|  1071M| 21232   (1)| 00:04:15 |
-------------------------------------------------------------------------------

Predicate Information (identified by operation id):
---------------------------------------------------

   2 - access("CUST_ID">10000)
```

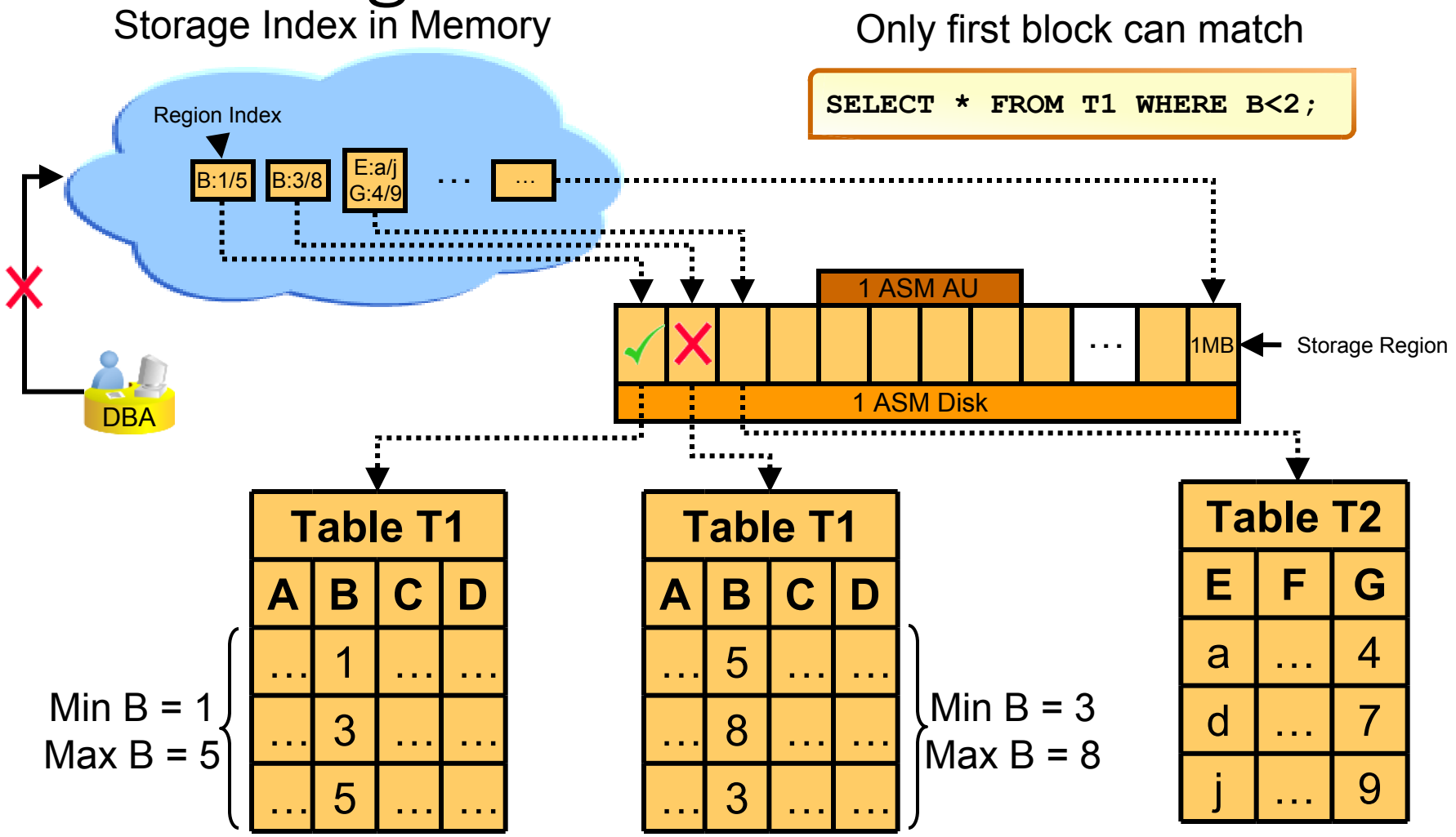
Smart Scan Statistics: Example

```
SQL> select count(*) from customers where cust_valid = 'A';

  COUNT(*)
----------
   8602831

Elapsed: 00:00:11.76

SQL> SELECT s.name, m.value/1024/1024 MB FROM V$SYSSTAT s, V$MYSTAT m
  2  WHERE s.statistic# = m.statistic# AND
  3  (s.name LIKE 'physical%total bytes' OR s.name LIKE 'cell phys%'
  4  OR s.name LIKE 'cell IO%');

NAME                                                                  MB
---------------------------------------------------------------- ----------
physical read total bytes                                      18005.6953
physical write total bytes                                              0
cell physical IO interconnect bytes                            120.670433
cell physical IO bytes sent directly to DB node to balance CPU u        0
cell physical IO bytes saved during optimized file creation             0
cell physical IO bytes saved during optimized RMAN file restore         0
cell physical IO bytes eligible for predicate offload          18005.6953
cell physical IO bytes saved by storage index                           0
cell physical IO interconnect bytes returned by smart scan     120.670433
cell IO uncompressed bytes                                     18005.6953
```

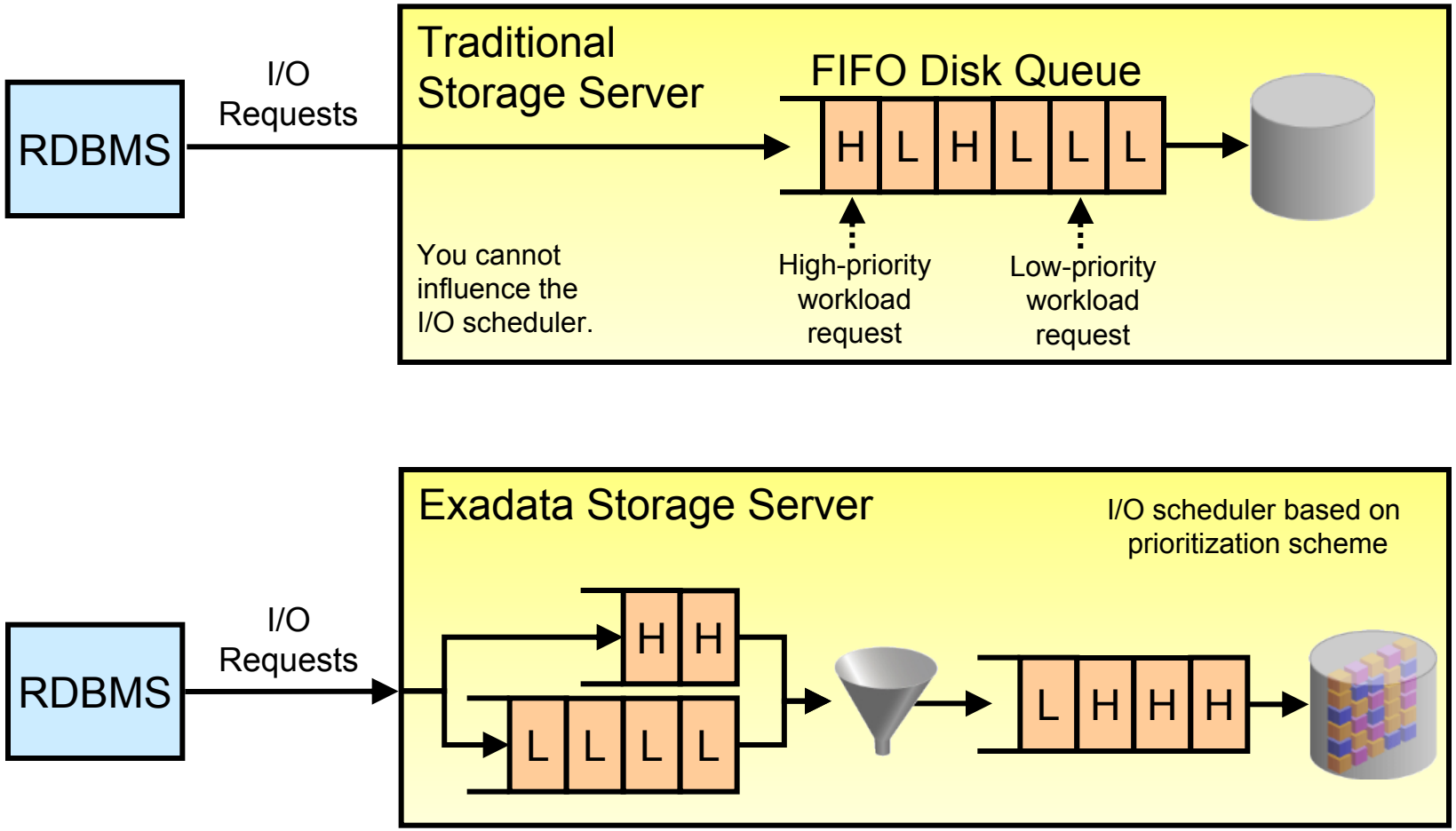
I/O Sent Directly to Database Server to Balance CPU Usage: Example

```
SQL> select count(*) from customers where cust_valid = 'A';

  COUNT(*)
----------
   8602831

Elapsed: 00:01:42.59

NAME                                                                  MB
---------------------------------------------------------------- ----------
physical read total bytes                                      18005.6953
physical write total bytes                                             0
cell physical IO interconnect bytes                            2475.24233
cell physical IO bytes sent directly to DB node to balance CPU u 2394.57133
cell physical IO bytes saved during optimized file creation            0
cell physical IO bytes saved during optimized RMAN file restore        0
cell physical IO bytes eligible for predicate offload          18005.6953
cell physical IO bytes saved by storage index                          0
cell physical IO interconnect bytes returned by smart scan     2475.24233
cell IO uncompressed bytes                                     18005.6953

EVENT                                    TOTAL_WAITS  WAIT_SECS AVG_WAIT_SECS
---------------------------------------- ----------- ---------- --------------
cell smart table scan                           9128      98.19          .0108
```

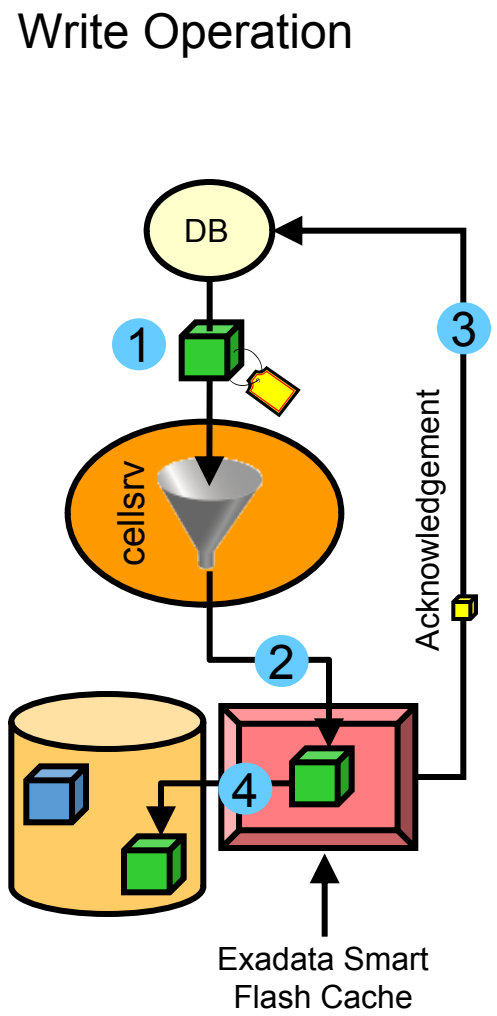
Exadata Storage Index: Overview



I/O Resource Management



Using Exadata Smart Flash Cache: Write-Back Cache



- How it works:
  - Suitable writes go to flash only.
  - Data is automatically written to disk as it ages out of the cache.
  - Active data blocks can reside in flash indefinitely.
  - Reads are handled the same way as in write-through mode.
- Characteristics:
  - Ideal for write-intensive applications.
  - For many applications, most I/O is serviced by flash.
  - If a problem is detected, I/O operations transparently fail over to mirrored copies of data also on flash.

Setting the Flash Cache Mode

- Providing write-back mode:

```
CellCLI> DROP FLASHCACHE
CellCLI> ALTER CELL SHUTDOWN SERVICES CELLSRV
CellCLI> ALTER CELL flashCacheMode = WriteBack
CellCLI> ALTER CELL STARTUP SERVICES CELLSRV
CellCLI> CREATE FLASHCACHE ALL
```

- Causing write-through mode:

```
CellCLI> ALTER FLASHCACHE ALL FLUSH
CellCLI> DROP FLASHCACHE
CellCLI> ALTER CELL SHUTDOWN SERVICES CELLSRV
CellCLI> ALTER CELL flashCacheMode = WriteThrough
CellCLI> ALTER CELL STARTUP SERVICES CELLSRV
CellCLI> CREATE FLASHCACHE ALL
```

Exadata Best Practices

ASM Allocation Unit Size

- By default, ASM uses an allocation unit (AU) size of 1 MB.
- For Exadata storage, the recommended AU size is 4 MB.
  - AU size must be set when a disk group is created.
  - AU size cannot be altered after a disk group is created.
  - AU size is set using the `AU_SIZE` disk group attribute.

```
SQL> CREATE DISKGROUP data NORMAL REDUNDANCY
     DISK 'o/*/data_CD*'
     ATTRIBUTE 'compatible.rdbms' = '11.2.0.0.0',
               'compatible.asm' = '11.2.0.0.0',
               'cell.smart_scan_capable' = 'TRUE',
               'au_size' = '4M';
```

Index Usage

- Queries that require indexes on a previous system might perform better using Exadata and Smart Scan.
- Consider removing indexes where Smart Scan delivers acceptable performance.
- Removing unnecessary indexes will:
  - Improve DML performance
  - Save storage space
- Test the effect of removing indexes by making them invisible:

```
SQL> ALTER INDEX <index_name> INVISIBLE;
```

Extent Size

- Segments should have extents that are a multiple of the ASM AU size:
  - Stops needless proliferation of small extents in the database
  - Optimizes I/O by aligning extent and ASM AU boundaries

```
SQL> CREATE TABLE t1
     (col1 NUMBER(6), col2 VARCHAR2(10))
     STORAGE ( INITIAL 8M MAXSIZE 1G );
```

```
SQL> CREATE BIGFILE TABLESPACE ts1
     DATAFILE '+DATA' SIZE 100G
     DEFAULT STORAGE ( INITIAL 8M NEXT 8M );
```

  - For very large segments, it is optimal to stripe each extent across all of the available disks

Exadata Specific System Statistics

- Gather Exadata specific system statistics:

```
SQL> exec dbms_stats.gather_system_stats('EXADATA');
```

  - Enables the optimizer to more accurately cost operations using actual performance information:
    - —CPU speed
    - —I/O Performance
  - Sets multi block read count (MBRC) correctly for Exadata
  - Requires at least Oracle Database version 11.2.0.2 BP 18 or 11.2.0.3 BP 8
  - Recommended for all new databases
    - —Test thoroughly before changing existing databases.
    - —Databases with stable good plans do not require a change.

THANK YOU