SDC
STORAGE DEVELOPER CONFERENCE
SNIA SANTA CLARA, 2014

Making Storage Smarter

Jim Williams
Martin K. Petersen

ORACLE®

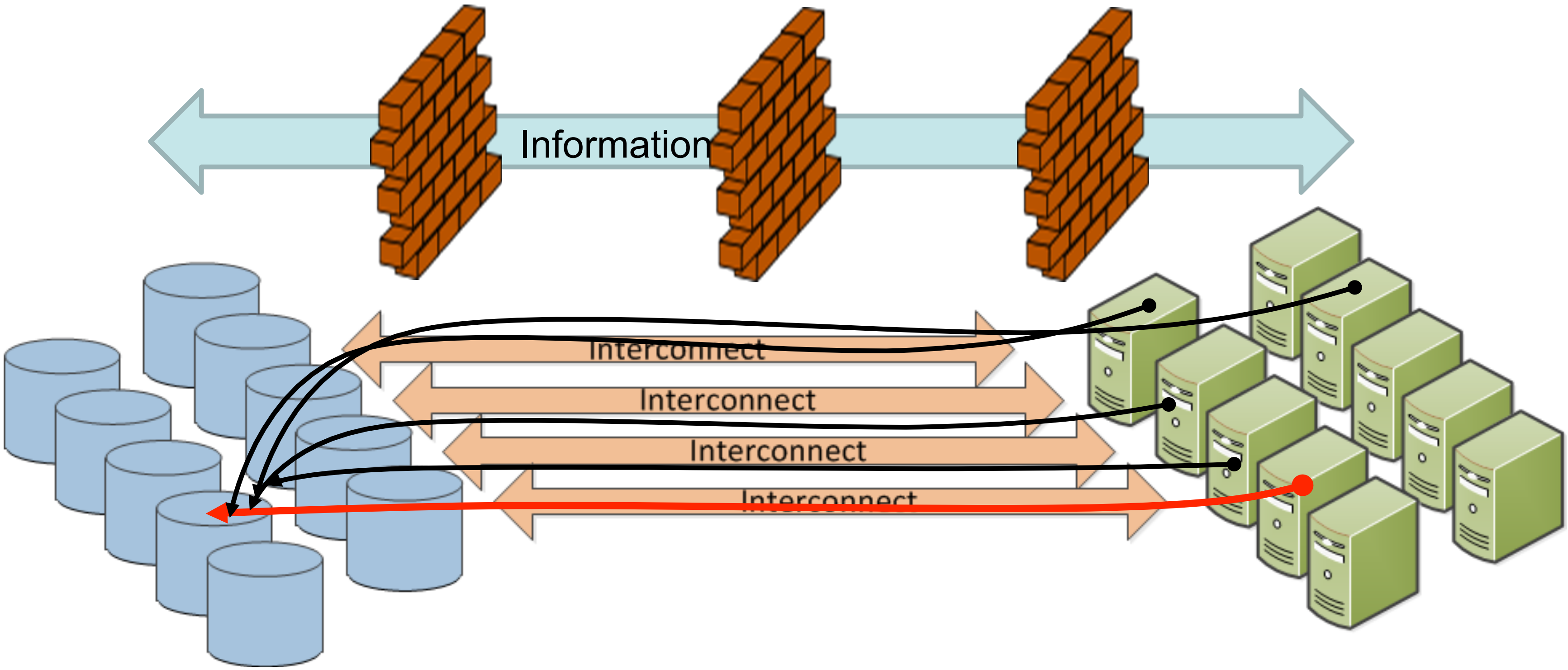
Agenda

- Background
- Examples
- Current Work
- Future

Definition

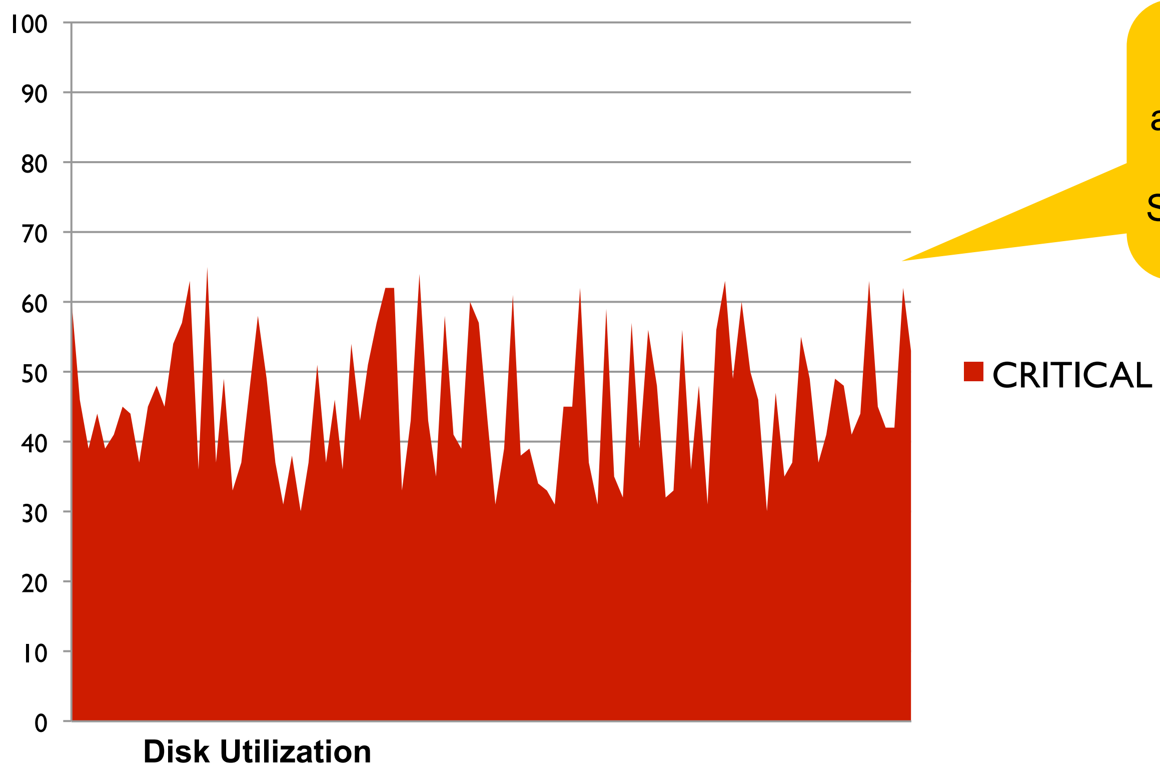
- Storage is made smarter by exchanging information between the application and the physical storage layer thereby enabling storage to optimize its utilization of resources in meaningful ways.

Making Storage Intelligent



Unfortunately storage abstraction and layering makes exchanging performance hints, not defined by existing standards, difficult, if not impossible.

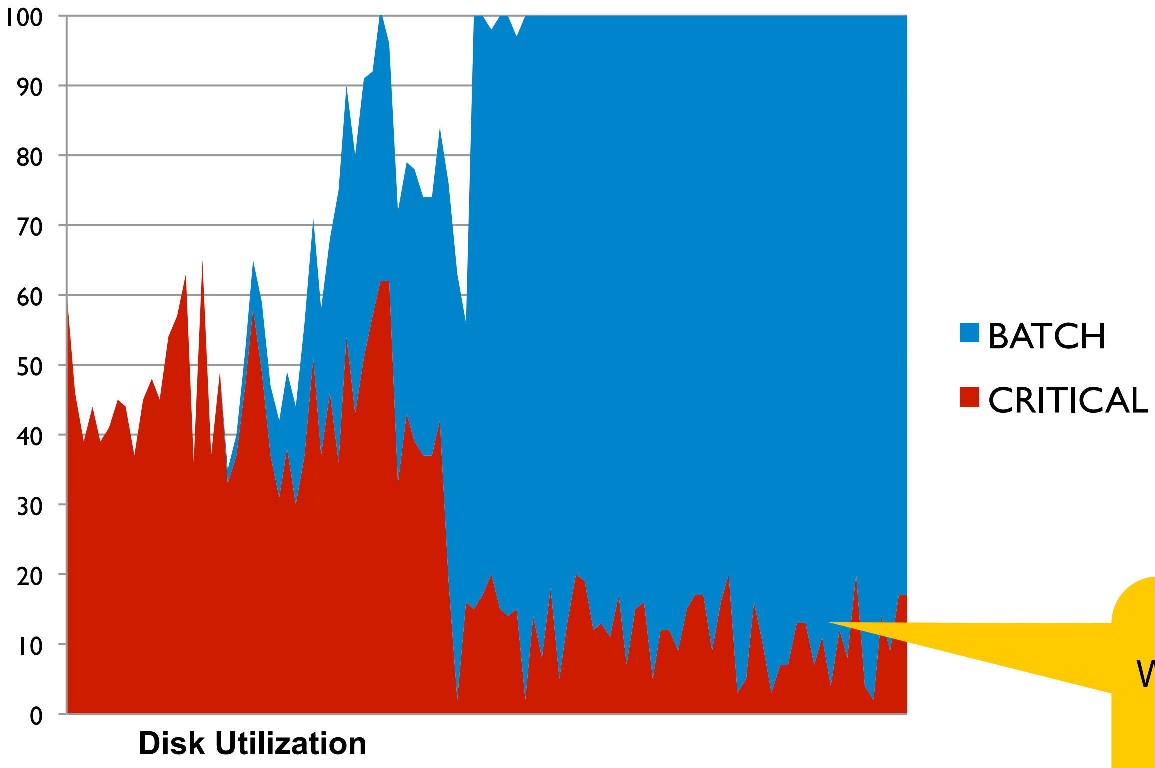
A Use Case for Intelligent Storage



Your disk utilization is around 50% and you are running Critical reports.

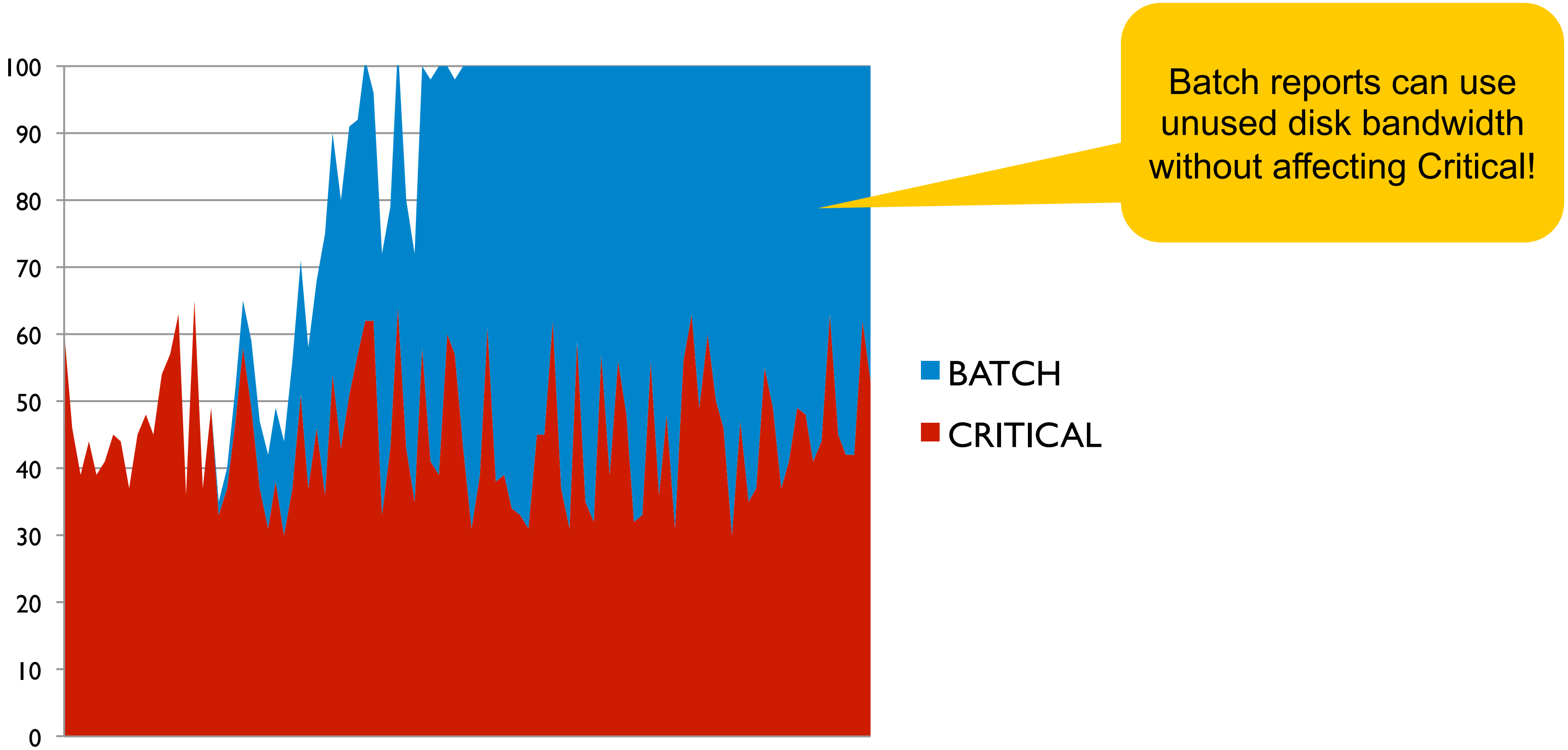
Should you run a non-critical report?

Current Storage Architectures



Will the performance of your Critical reports be acceptable if their I/O throughput drops?

Preferred Behavior



Disk Utilization by Consumer Group

Motivation for Intelligent Storage

- Storage vendor:
  - Competitive advantage
  - Improves margin by adding product value
- Customer:
  - Reduces need for over provisioning and enables greater resource sharing
  - Ensure critical applications getting sufficient resources
  - Improved storage performance

What kinds of information should be exchanged?

- Meta-data sent to the storage device that:
  - Improves the storage device's ability to optimize its internal resources
    - Expected cache usage
    - Latency vs bandwidth tradeoffs
  - Metadata that facilitates data placement policy
  - Metadata communicating workload priority
  - Metadata communicating quality of service

Nature of Metadata

- Static vs dynamic
  - I/O operation or
  - Data location oriented
- Application specific or application generic
- Does metadata span multiple applications
- Is the storage device provided an optimization policy from the application layer
- What about non-performance related metadata

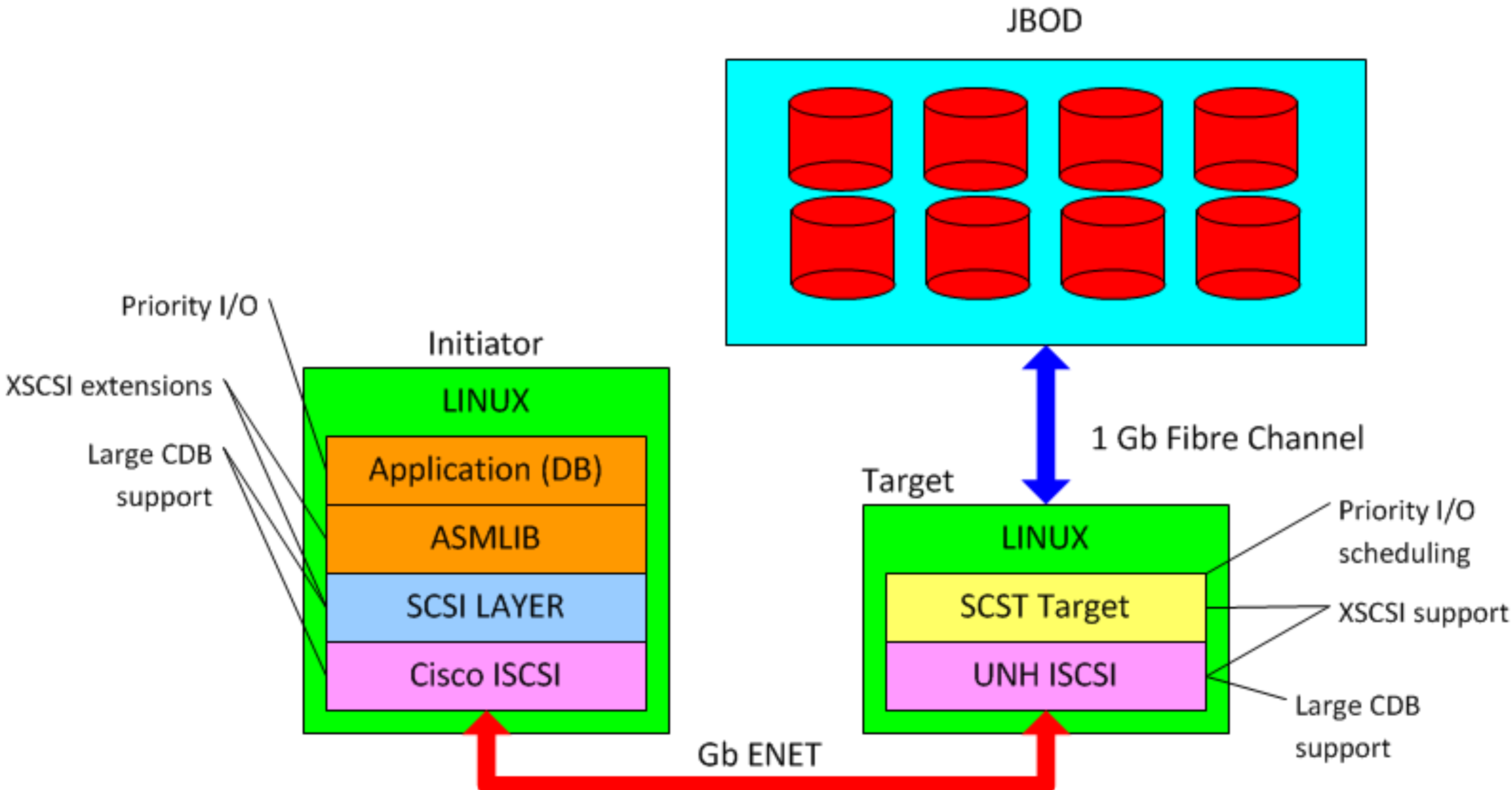
Examples

- Internal Oracle prototype
- Oracle Intelligent Storage Protocol
- Exadata I/O Resource Manager

Oracle 2004 Smart Disk Research

- Worked with major storage partner investigating ways of exchanging database related hints
  - I/O priority
  - Cache hints
  - Protection zones
  - Disk Keys for optimistic locking
- Build a target disk supporting I/O priority
- Made several performance measurements
  - Transaction vs backup workload

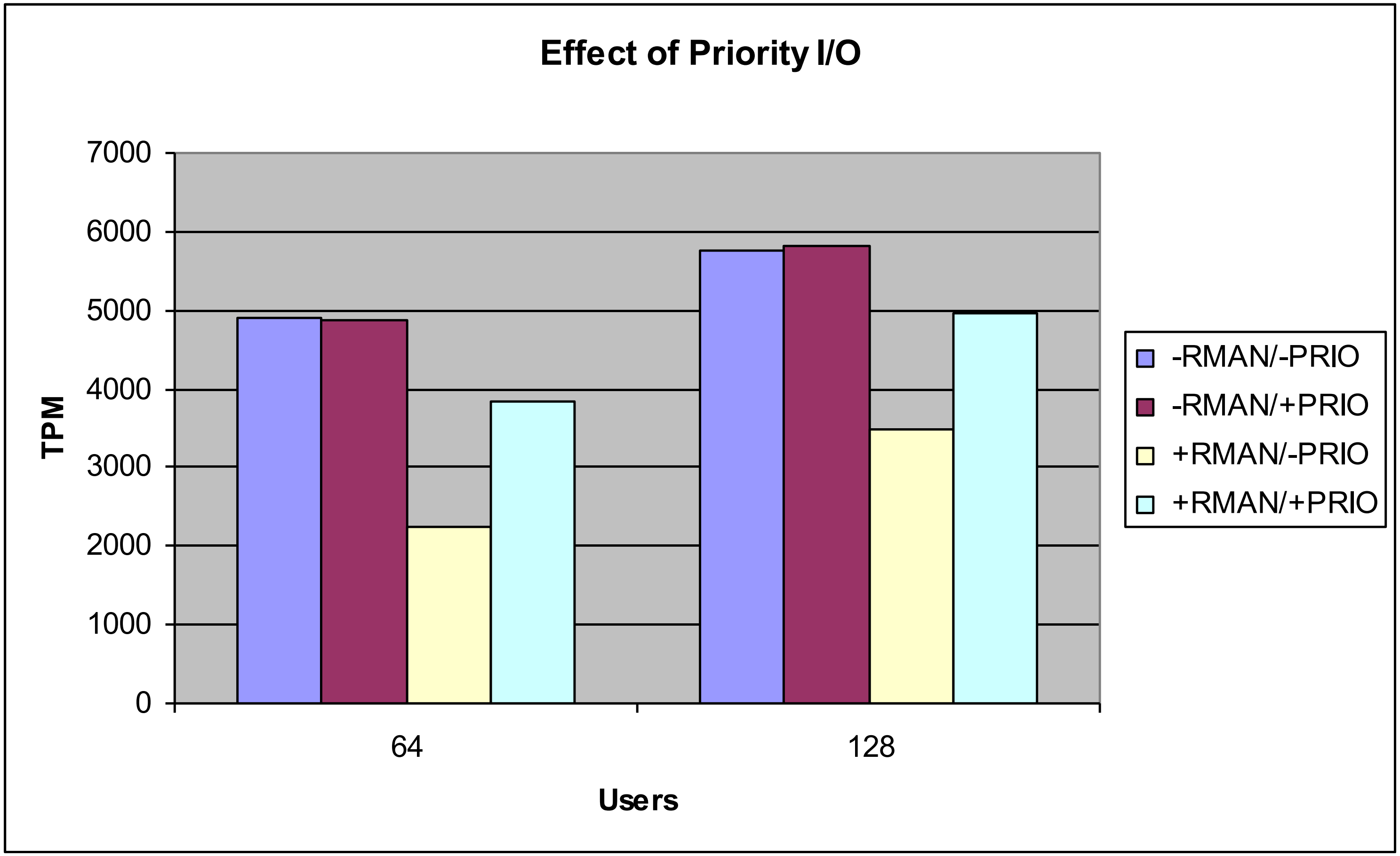
R&D Research 2004



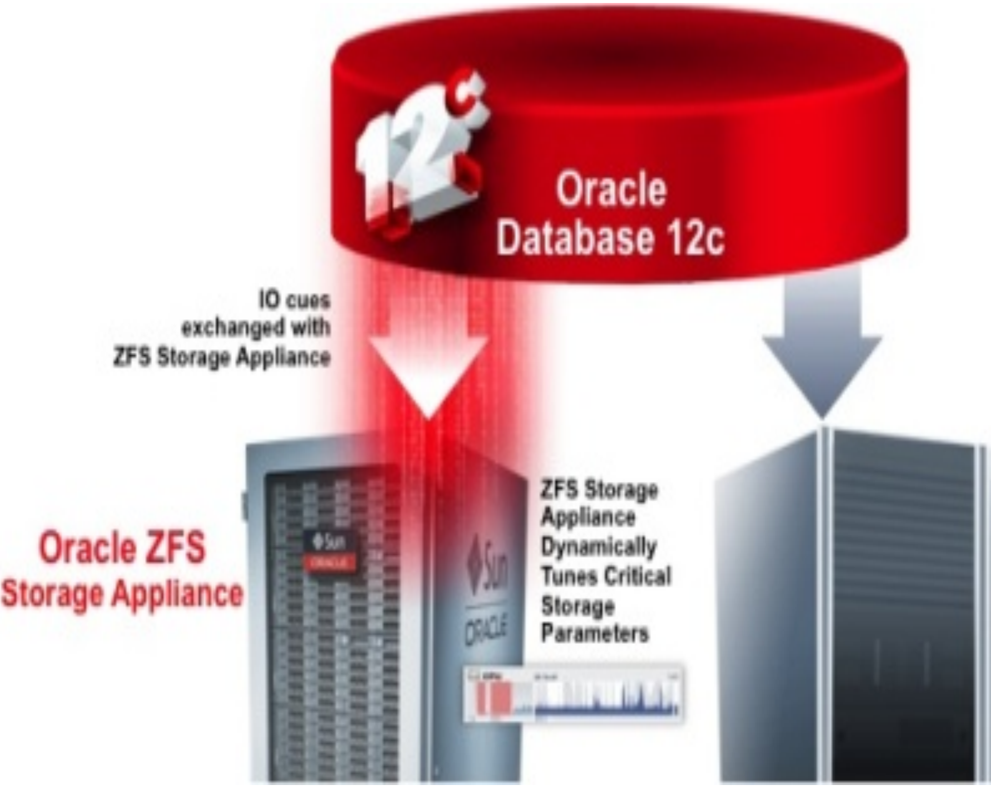
Continued

- Passed I/O priority field from database to disk
- Mixed workload of TPC-C and backup
- Measure effect of prioritization and benefit to transaction workload
- Larger loads benefited more
- Small storage bandwidth impact of extended CDB
- Workload prioritization made difficult because disk I/O is not preemptable and not an insignificant performance impact of queuing and scheduling

Continued


Effect of Priority I/O
7000
6000
5000
4000
3000
2000
1000
0
TPM
64
128
Users
-RMAN/-PRIO
-RMAN/+PRIO
+RMAN/-PRIO
+RMAN/+PRIO

Oracle Intelligent Storage Protocol



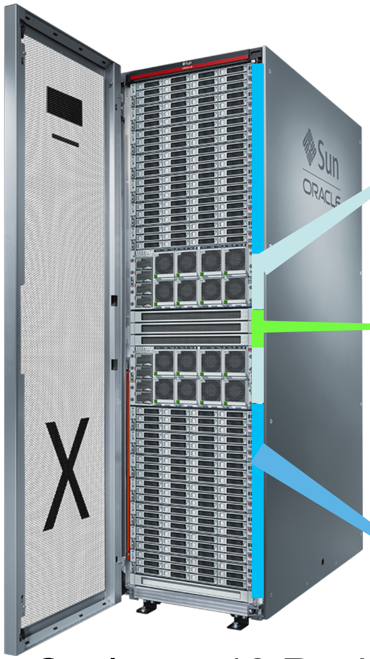
- Storage Awareness of Database Workload
  - Database sends cues about each IO to the storage
  - Over 70 database IO hints are recognized
  - Hints grouped into 5 distinct categories by storage
  - Information used by ZFS Storage appliance to adaptively and automatically adjust for optimal efficiency

Oracle Intelligent Storage Protocol

- I/O cache hint embedded in NFSv4 I/O requests
- Available ONLY with Oracle Database 12c and Oracle ZFS Storage Appliance OS 8 and above
- Intelligence is in the code within both the database and storage
- Integrated NFS client in database (dNFS)

I/O Resource Management in Exadata

Complete | Optimized | Standardized | Hardened Database Platform



Scales to 18 Racks

- **Powerful Database Servers**
  - 2x 8-socket servers ➔ **240 cores**, up to **12TB DRAM**
- **Unified Ultra-Fast Network**
  - 40 Gb InfiniBand internal connectivity ➔ **all ports active**
  - 10 Gb or 1 Gb Ethernet for data center connectivity
- **Scale-out Intelligent Storage Servers**
  - 14x 2-socket servers ➔ **168** cores in storage
  - 168 SAS disk drives ➔ **672 TB** HC or **200 TB** HP
  - 56 PCI Flash cards ➔ **44 TB** Flash + **compression**

SDC 14

Exadata I/O Resource Manager

- Exadata & Oracle Database Appliances
  - “Complete Solution” batteries included
  - Storage, database, and servers in a single box
  - Oracle engineers and optimizes for a particular objective
- IORM is a component of a complete Resource Management capability
- IORM uses resource plans and tags I/O operations

IORM for Workloads within and across databases

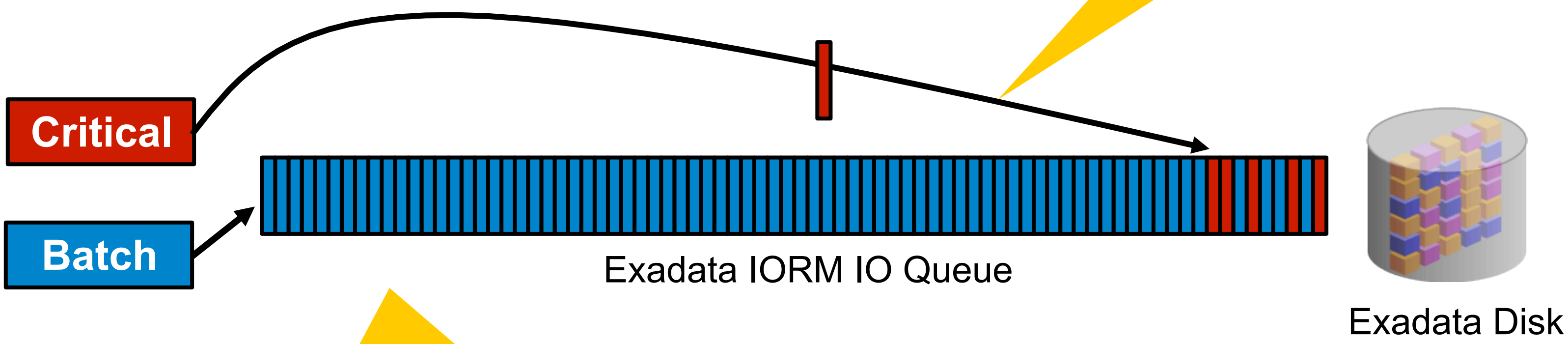
The Database Resource Plan is used to manage both CPU and Disk I/O

Database Resource Plan		
Consumer Group	**Resource Allocation**	**Utilization Limit**
Critical	50	100%
Batch	20	100%
AdHoc	10	50%
ETL	10	100%
Other	10	100%

- Resource plans determine how workloads share available I/O capacity
- Workloads are tagged with a consumer group ID
- IORM works within a database and across databases

A Use Case for IORM: OLTP vs Reports

Exadata IORM can put a new IO anywhere in the queue!



IORM tags every IO:
- Consumer Group id
- Database or PDB id
- Background or foreground?
- Buffer cache or smart scan?

Exadata IORM controls the order of the queue, based on the Resource Plan.

Generic approaches to I/O hints

- POSIX, SUS
- Linux
- NFS v4.2
- T10/T13
- I/O Classification Survey

POSIX

- posix_fadvise()
  - POSIX_FADV_NORMAL
  - POSIX_FADV_SEQUENTIAL
  - POSIX_FADV_RANDOM
  - POSIX_FADV_WILLNEED
  - POSIX_FADV_DONTNEED
  - POSIX_FADV_NOREUSE

Linux-specific Interfaces

- ionice (CFQ I/O scheduler)
  - Idle
  - Best effort, priority *n*
  - Real Time
- REQ_META
  - set by filesystems on metadata

NFS 4.2 ior_hints

- IO_ADVISE4_NORMAL
- IO_ADVISE4_SEQUENTIAL (_BACKWARDS)
- IO_ADVISE4_RANDOM
- IO_ADVISE4_WILLNEED (_OPPORTUNISTIC)
- IO_ADVISE4_DONTNEED
- IO_ADVISE4_NOREUSE
- IO_ADVISE4_{READ,WRITE}
- IO_ADVISE4_INIT_PROXIMITY

T10 / T13 Logical Block Markup Descriptors

Table 59 — READ (10) command

Byte \ Bit	7	6	5	4	3	2	1	0
0	OPERATION CODE (28h)							
1	RDPROTECT			DPO	FUA	RARC	Obsolete	Obsolete
2	(MSB)	LOGICAL BLOCK ADDRESS						
•••								
5								(LSB)
6	Reserved			GROUP NUMBER				
7	(MSB)	TRANSFER LENGTH						
8								(LSB)
9	CONTROL							

T10 / T13 Logical Block Markup Descriptors

M.3 Access Patterns LBMDs

M.3.1 Access Patterns LBMD format for SCSI

The Access Patterns LBMD format processed by SCSI device servers is shown in table M.4.

Table M.4 — Access Patterns LBMD format for SCSI

Bit / Byte	7	6	5	4	3	2	1	0
0	ACDLU	Reserved			LBMD TYPE (0h)			
1	OVERALL FREQUENCY		READ/WRITE FREQUENCY		WRITE SEQUENTIALITY		READ SEQUENTIALITY	
2	Reserved				SUBSEQUENT I/O		OSI PROXIMITY	
3	Reserved							

T10 / T13 Logical Block Markup Descriptors

- READ SEQUENTIALITY (normal, random, seq.)
- WRITE SEQUENTIALITY (normal, random, seq.)
- READ/WRITE FREQ. (normal, read, write)
- OVERALL ACCESS FREQ. (normal, less, more)
- SUBSEQUENT I/O (normal, unlikely, likely)
- OSI PROXIMITY (normal, unlikely, likely)

SDC 14

Combining POSIX, NFS, T10, T13, …

Storage Tier

Access Frequency

Cache

No Reuse

?

Priority

Random

Bootiness

Don’t Need

Sequential

Will Need

I/O Classification Property Matrix

I/O Class	Examples
Transaction	Filesystem or database journals, checkpoints
Metadata	Filesystem metadata
Paging	Swap
Real Time	Audio/video streaming
Data	Normal application I/O
Background	Backup, data migration, RAID resync, scrubbing

About 50 different metrics from a variety of applications consolidated into 6 distinct I/O classes.

I/O Classification Property Matrix

I/O Class	Completion Urgency	Desired Future Access Latency	Predicted Future Access Freq.
Transaction	High	Low	High
Metadata	High	Low	Normal
Paging	High	Normal	Normal
Real Time	High	Normal	Low
Data	Normal	Normal	Normal
Background	Low	Normal/High*	Low

I/O Classification to POSIX/NFS

I/O Class	posix_fadvise()	NFS 4.2
Transaction	Sequential	Sequential, Write
Metadata	Random	Random, Read, Will Need Opportunistic
Paging	Random	Random, Will Need Opportunistic
Real Time	Sequential, Don't Need	Sequential, Don't Need
Data	Normal	Normal
Background	Sequential, No Reuse	Sequential, No Reuse

I/O Classification to T10/T13 Logical Block Markup Descriptor

I/O Class	Read Seq.	Write Seq.	R/W Freq.	Overall Freq.	Subsequent I/O Depend.
Transaction	Sequential	Sequential	Write	High	High
Metadata	Random	Random	Normal	Normal	High
Paging	Random	Random	Normal	Normal	High
Real Time	Sequential	Sequential	Read	Normal	Normal
Data	Normal	Normal	Normal	Normal	Normal
Background	Sequential	Sequential	Normal	Low	Low

Future

- Prototype implementation in Linux being reworked to match current T10/T13 proposal
- Aiming to bring posix_fadvise(), NFS and T10/T13 hints together
  - Linux/POSIX/NFS are discrete flags
  - T10/T13 hints are all-encompassing descriptors
- I/O classes may actually be easier for the storage device to make use of

Conclusion

- Tradeoff between *standards-based* and *proprietary* “smart storage”
  - Various proprietary approaches already shipping with several performance benefits
  - A standards-based approach is more challenging to implement

Conclusion

- Applications and operating systems need to make better use of—and possibly extend—existing interfaces
- Need better standardization across different types of storage
- Some of the standardization work, in its current form, is difficult to make use of

Q & A