Making Databases Smarter And Faster

Innovations Enabled By Engineering Hardware And Software Together

Cristóbal Pedregal Martin
Systems Technologies
Oracle

CERN openlab Technical Workshop
23 January 2019

Safe Harbor Statement

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

innovations:
easy
to recognize…

… but
not so easy
to produce

Detour How Information Technology Evolves

- Each hardware component has a **given** capacity, speed, and price, that **change at different rates over short times** (single-digit years)
  - For example, consider memory versus network speed, disk versus memory capacity
- A good system must embody in its design and configuration the relative speeds (capacity, etc.) of its components, not just the expected workloads

  people.eecs.berkeley.edu/~rcs/research/interactive_latency.html
- Therefore, you cannot expect all your design decisions and tradeoffs to apply unchanged to a new generation of components
  - Not to mention introducing "new" components (e.g., FPGA, nonvolatile RAM)

Agenda

1. What Are Innovations
2. Dramatically Reducing Database I/O
3. Fast Response Database Transactions
4. Summary and References

Dramatically Reducing Database I/O via Database Awareness in Storage System

Challenge: Avoid Reading Data that Is Irrelevant to a Query

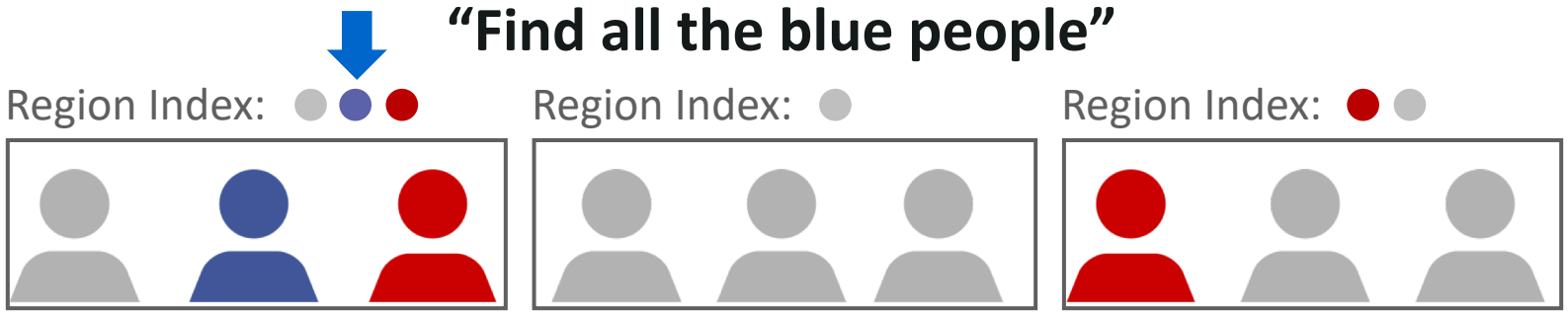
- Databases frequently need to find a small amount of relevant rows in a large table
  - e.g., matches to a value, range lookup, finding max value
- Want to avoid reading blocks from storage if the block does not contain relevant rows

“Find all the blue people”

Insight: Can Learn from Past Queries to Eliminate Future I/O

Exadata Storage Indexes

- By keeping key statistics about which data is in each region of storage we can avoid reading regions that don't contain relevant data
  - Example key statistics are min and max values for various columns
- Can skip regions when the key statistics tell us the value cannot be there
- Key statistics are *automatically chosen* based on previous queries



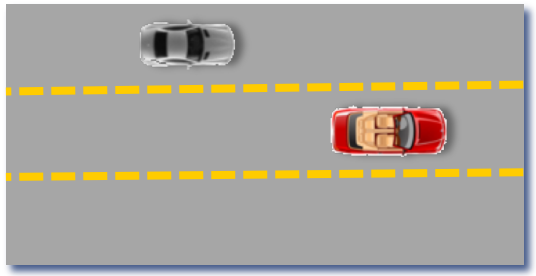
Benefits of Exadata Storage Indexes

- Up to orders of magnitude faster performance
  - Skipping a region eliminates the I/O and CPU needed to search the rows in the region
  - Conceptually similar to fine-grained automatic partitioning, but on many columns
- Completely automatic
  - Statistics are automatically collected for frequently searched columns
- No overhead
  - CPU and memory cost of maintaining a storage index is near-zero and it is offloaded to the storage servers anyway

Fast Response Database Transactions Even When Running Mixed Database Workloads

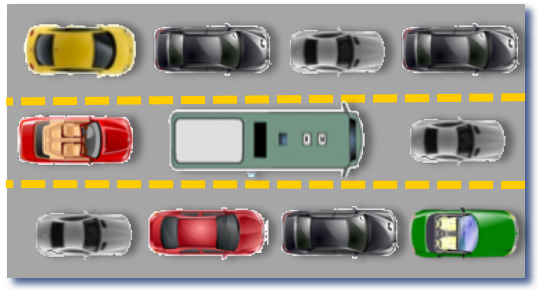
Challenge: Mixing Workloads Degrades Response Times

Only OLTP



- Transactional workloads (OLTP) often run concurrently with high throughput workloads
  - Database consolidation, batch, real-time analytics, reporting, backups

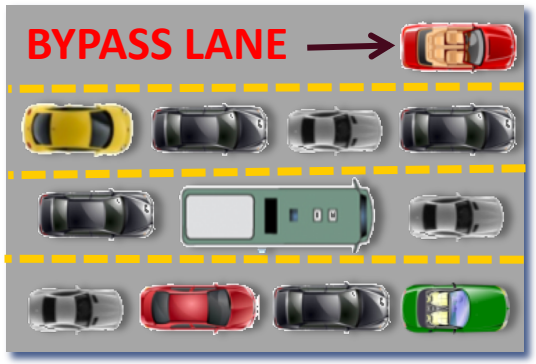
Mixed Workload



- However, high throughput workloads can severely degrade OLTP
  - They create long network queues, delaying critical OLTP messages

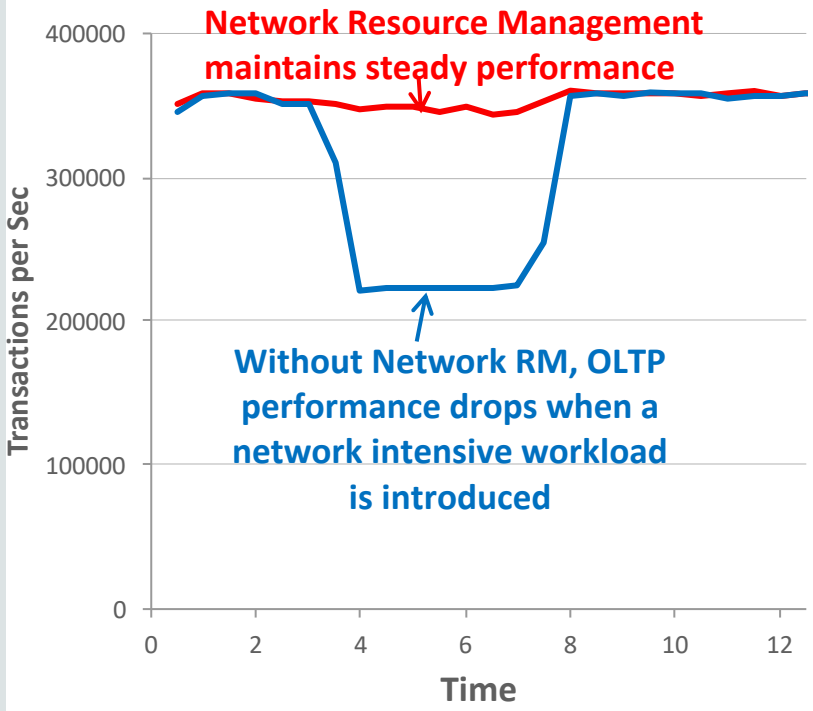
Insight: Identify and Accelerate Critical OLTP Messages Across Full Stack

Network Resource Management



- Database tags messages that require **low-latency**
  - Log writes, locks, cache-fusion messages, etc.
- Low-latency messages bypass all other messages
  - Reporting, backups, batch, etc.
  - Even partially sent messages are bypassed
- Exadata accelerates low-latency messages *in all layers:* database, network cards, switches, and storage
  - Otherwise bottleneck just moves

Benefits of Network Resource Management



- OLTP messages show consistent, low latency despite consolidation and mixed workloads
- Full-stack integration makes it *automatic*
  - No VLANs, no QoS settings, etc.
- Fine grained and database aware
  - Specific messages, not whole sockets

Agenda

1. What Are Innovations
2. Dramatically Reducing Database I/O via Database Awareness in Storage System
3. Fast Response Database Transactions Even When Running Mixed Database Workloads
4. Summary and References

Summary?

- Abstraction and modularity are good. Just don't let them limit you
  - Sometimes it's worth breaking the abstractions and allow semantics to percolate – these and other innovations yield **orders of magnitude gains** (speed, space,… money)
    - "Operational" (transactional) Databases that support Analytics work on your real-time business data
  - But to break abstractions you need to **master the full hardware and software stack**
- Building infrastructure-grade systems is both hard and specialized
  - You can be smart and humble – consider existing solutions first
  - if an existing solution is good enough for CERN, it's probably good enough for you ***Le mieux est l'ennemi du bien*** (Voltaire, and a lot of engineers since)
    - but see Joel Spolsky's comments on how to keep your hackers entertained ;-)

More insights & innovations at **`oracle.com/goto/dbinsights`** and **`https://www.oracle.com/technetwork/database/exadata/overview/index.html`**