Bringing Multi-.Threading to the I/O Level

Rodrigo Medeiros
Solutions Architect, Data Platforms

@rationalrodrigo   @DataCore

Empowering real-time, always-on data
Objectives

- Discuss I/O parallelism and how it affects database performance
- Discuss compute parallelism vs I/O parallelism
- Demo: software solution to improve I/O
So Much Data, So little Time

- Large data sets even in small dataset
- Global business, activity in multiple time zones = 24/7 activity
- Smaller windows for Batch Processing
- Fresh, accurate, and readily available, regardless of volume.

“ I never guess. It is a capital mistake to theorize before one has data.”  

Sir Arthur Conan Doyle: author of Sherlock Holmes
Graph these comma-separated phrases: data, reasonable

between 1880 and 2008 from the corpus English with smoothing of 3.

Search lots of books
Transaction Processing

• A two-second response is too much
• Online experience and the expectation of near-instant responses
• Larger volumes of data
• Sub-second search and update requirements
• Data-intensive UX pushing the volume of reads higher

“You can have it all. Just not all at once.”

Oprah Winfrey
“Throwing hardware” at the problem

- Processing power / cores
- Memory
- Storage
- Network
- More nodes

Downside
- Have to ask (plead)
- Hard to guess which one to start with?
- Get it wrong and it gets harder to ask again
Fixing the Code

- Collect Data
- Identify bottlenecks
- Change code, indexes, settings
- Test and deploy
The unusual suspect

- Layers of I/O developed for a single CPU
- Assumption: it will be slow, so don’t wait for it
- Global queues
- Some serialization
- Legacy code in the I/O stack
About DataCore

• Parallel processing background
• Pioneers in Software Defined Storage (SDS)
  • Patent holders of Thin Provisioning
• World record for Storage Performance Council
  • SPC-1 2016, 5M IOPS, 1/5th of the cost of 2nd place
• MaxParallel is part of a successful SDS stack
  • Production installs for over 2 decades
No parallelism in I/O and Database Performance

4 concurrent tasks on 8-core server
Amdahl’s Law

Parallel Portion
- 50%
- 75%
- 90%
- 95%

Speedup vs. Number of Processors
Adding complexity to the picture...

Increased I/O complexity in virtualized environments

Latency variability at the storage layer
Asynchronous I/O and context switching

1. I/O REQUEST
2. DO OTHER WORK
3. ISR
4. DPC
5. APC
Asynchronous I/O with very fast drives

1. I/O REQUEST
2. DO OTHER WORK
3. ISR
4. DPC
5. APC

APPLICATION

FILE SYSTEM (NTFS)
VOLUME MANAGEMENT
DISC CLASS
...

HARDWARE INTERFACE

SSD TECHNOLOGY

μsec μsec μsec
Synchronous I/O
Potential Solutions

- Minimize suspension and context switching caused by I/O delays
- Smarter I/O cache
- Smarter read-ahead
- Speed matching buffer (for bursts of writes)
- Implement multiple I/O queues, multi-threaded I/O processing
- Smarter I/O scheduling
- Helper threads
Faster data access + better CPU utilization

Delays & waste caused by serial scheduling

Parallel I/O with DataCore
Demo: MaxParallel on Windows
Where it resides

Windows: filter driver (kernel)

Linux: block device
Summary

• Serialization in I/O can hinder performance
  ...and may not be immediately visible

• Additional Tool
  software can be an additional tool in performance optimization

• Good fit in some environments
  higher core count, heavy and diverse workloads, slow drives

• Not relevant in some environments
  over-provisioned hardware, modest workloads, fast drives
Questions?

info@datacore.com
www.datacore.com
@DataCore
@rationalrodrigo