Arbitrary Dimension Reed-Solomon Coding and Decoding for Extended RAID on GPUs

Matthew Curry, H. Lee Ward, Anthony Skjellum, and Ron Brightwell

University of Alabama at Birmingham

Sandia National Laboratories

November 17th, 2008
The Need for More Reliable RAID

- Lack of Failure Prediction
  - SMART
  - MTTF
- Larger Disks
  - Stagnating Speeds
  - Bit-Error Rates
- Correlated Failures
  - Batch-Correlated Failures
  - Environment-Related Failures
Current Method: Nested RAID

- Stripe data over several RAID arrays
  - RAID 1 + 0: Stripe over multiple RAID 1 sets
  - RAID 5 + 0: Stripe over multiple RAID 5 sets
  - RAID 6 + 0: Stripe over multiple RAID 6 sets

- Reliability is marginally improved over non-“+0” variants, while requiring significantly more hardware.
Enabling RAID N+3 and Beyond

- Need a fast method of creating arbitrary amounts of parity
- Reed-Solomon Coding is an obvious solution, but performance is lacking
- On an x86-based CPU, performance is limited to approximately 90 MB/s per core to do $n + 3$ parity
- Main limitation: A lack of the ability to do parallel table lookups, a crucial optimization for Reed-Solomon coding
GPU Architecture

Figure: G80 Architecture
Framing the Experiment

Operating System Kernel

Network Buffer

iScsi Request

Block Buffer Cache

Retire write request and complete asynchronously

Driver
(Finish Request)

Network Packet

Disk Writeout Buffer

Network Packet

Network Buffer

iScsi Reply

GPU Accumulate Buffer

GPU (Parity Calculation Performed)

Driver copies write request data to accumulation buffers

Buffers rotate as GPU finishes operating on Operate Buffer

Contents of Operate Buffer Transferred to GPU Memory

GPU RAID

Matthew Curry, et. al (UAB/SNL)

November 17th, 2008
Throughput of Generation of $m$ Blocks with 2MB Maximum Buffer
Recovery Performance

Data Recovery Performance

Throughput (MB/s) vs. Size of Input (KB)

- NVIDIA 260: 13+3
- NVIDIA 260: 29+3
- Core 2 Duo: 13+3
- Core 2 Duo: 29+3

Matthew Curry, et. al (UAB/SNL)
Percentage of Time in PCI Transfer

Figure: Matthew Curry, et. al (UAB/SNL) GPU RAID November 17th, 2008
Conclusions

- A $300 GPU can support the workload of a sizable RAID array that can support any three disks failing.
  - 16-disk array at 100 MB/s per disk (vs. 7 for CPU)
  - 32-disk array at 51 MB/s per disk (vs. 4 for CPU)
- PCI-Express transfers can be fully hidden by the computation when done in parallel
- Future work includes building a working RAID system which includes this component (which will be available soon).
Thank you.

Matthew Curry
curryml@cis.uab.edu