Failing in Place for Low-Serviceability Infrastructure Using High-Parity GPU-Based RAID

Matthew L. Curry, University of Alabama at Birmingham and Sandia National Laboratories, mlcurry@sandia.gov

H. Lee Ward, Sandia National Laboratories, lee@sandia.gov

Anthony Skjellum, University of Alabama at Birmingham, tony@cis.uab.edu

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

Introduction

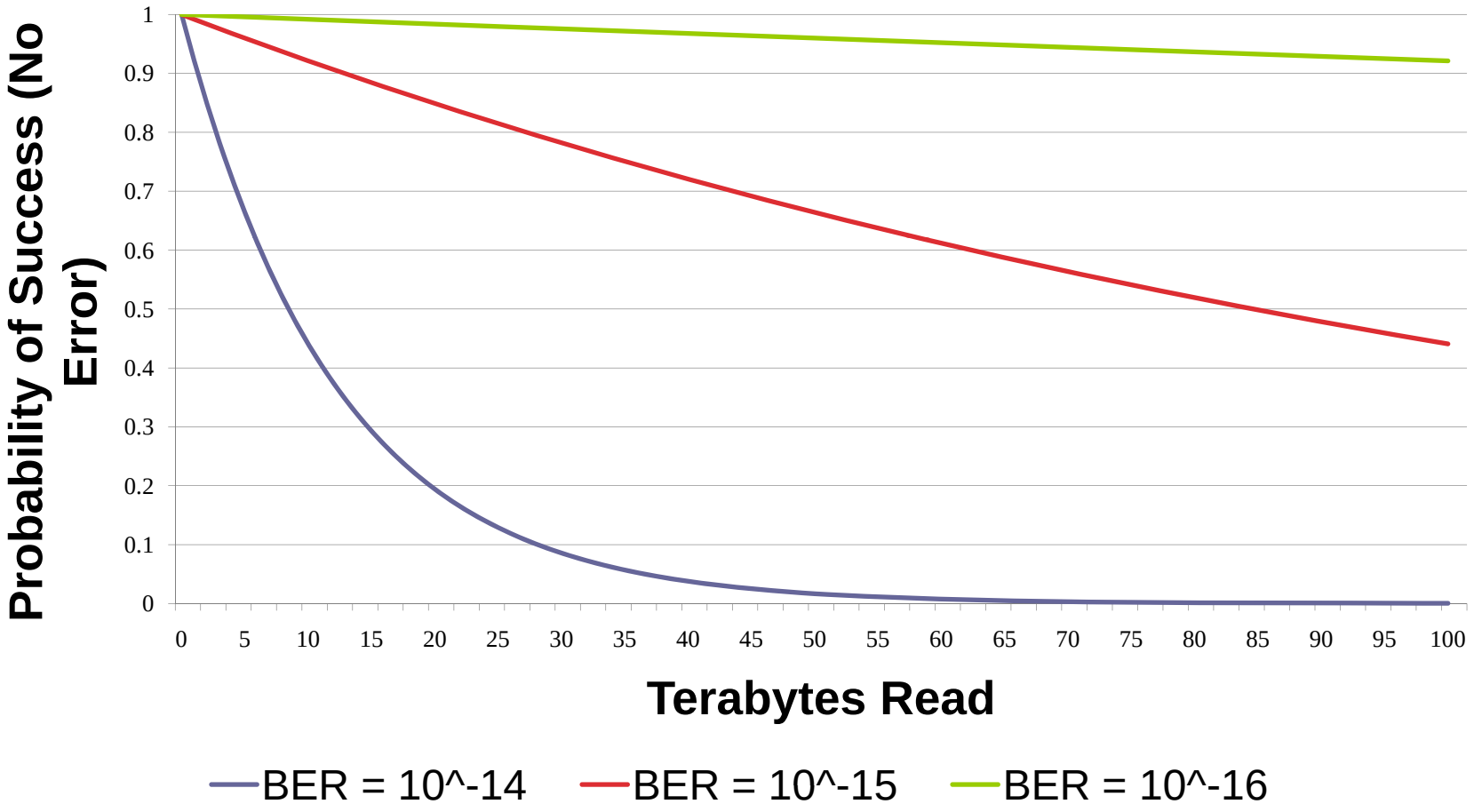
- RAID increases speed and reliability of disk arrays
- Mainstream technology limits fault tolerance to two arbitrary failed disks in volume at a time (RAID 6)
- Always-on and busy systems that do not have service periods are at increased risk of failure
  - Hot spares can decrease rebuild time, but leave installations vulnerable for days at a time
- High-parity GPU-based RAID can decrease long-term risk of data loss

Motivation: Inaccurate Disk Failure Statistics

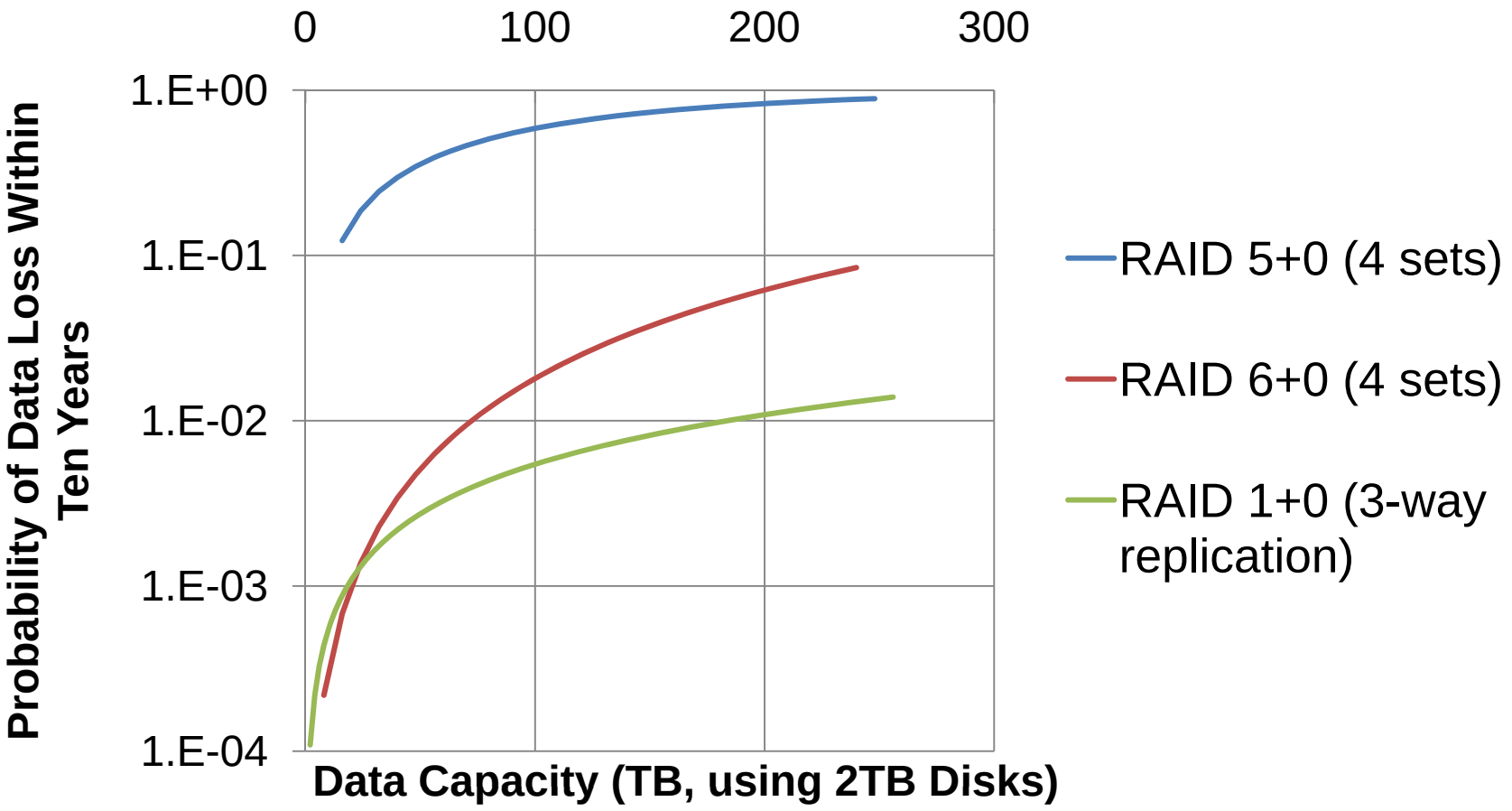
- Mean Time To Data Loss of RAID often assumes manufacturer statistics are accurate
  - Disks are 2-10x more likely to fail than manufacturers estimate*
  - Estimates assume low replacement latency and fast rebuilds… Without load

*Bianca Schroeder and Garth A. Gibson. “Disk failures in the real world: What does an MTTF of 1,000,000 hours mean to you?” In *Proceedings of the 5th USENIX Conference on File and Storage Technologies*, Berkeley, CA, USA, 2007. USENIX Association.

Motivation: Unrecoverable Read Errors



Result: Hot Spares Are Inadequate Under Load



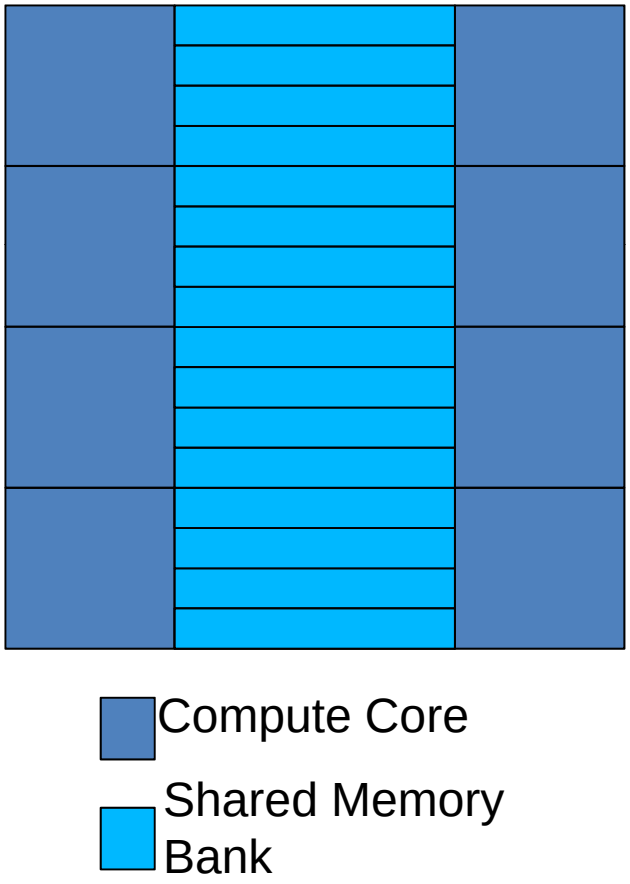
MTTR = One Week, MTTF = 100,000 Hours, Lifetime = 10 years

Solution: High-Parity RAID

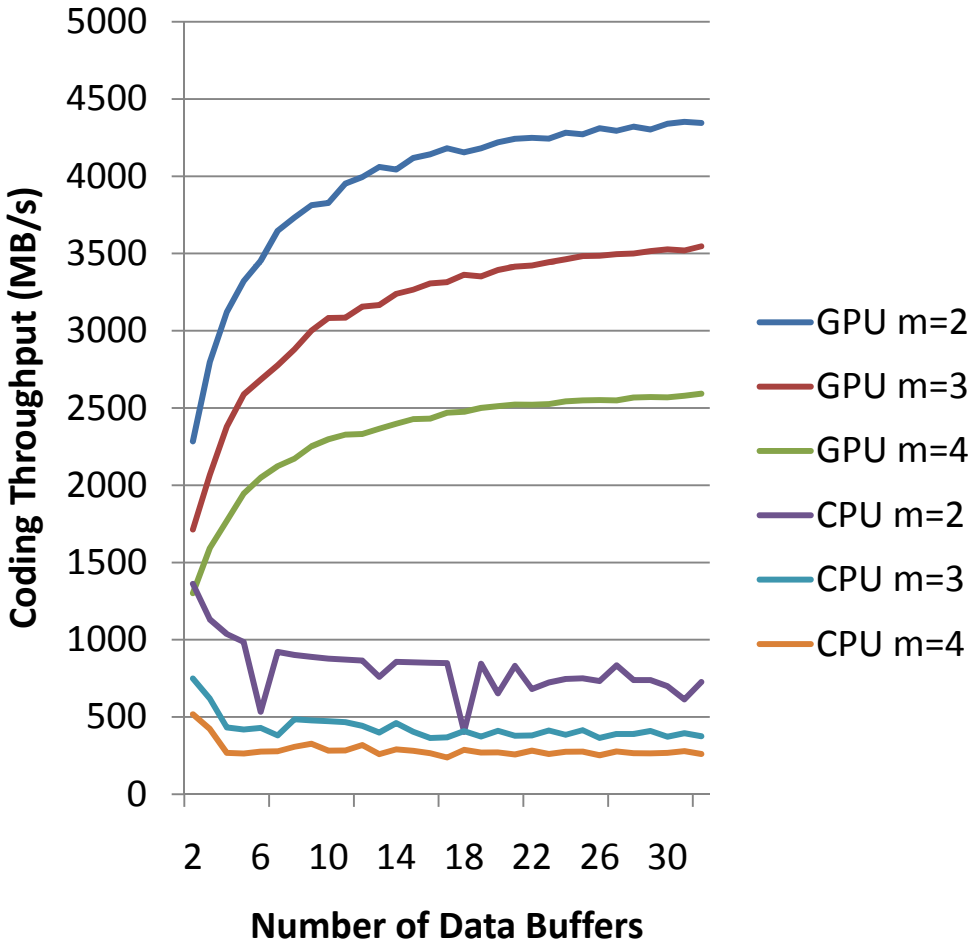
- Use Reed-Solomon Coding to provide much higher fault tolerance
  - Configure array to tolerate failure of any *m* disks, where *m* can vary widely
  - Analogue: RAID 5 has *m=1*, RAID 6 has *m=2,* RAID 6+0 also has *m=2*
  - Spare disks, instead of remaining idle, participate actively in array, eliminating window of vulnerability
- This is very computationally expensive
  - *k+m* RAID, where *k* is number of data blocks per stripe and *m* is number of parity blocks per stripe, requires *O(m)* operations per byte written
  - Table lookups, so no vector-based parallelism with x86/x86-64
  - Failed disks in array operate in degraded mode, which requires the same computational load for reads as writes

Use GPUs to Provide Fast RAID Computations

- Reed-Solomon coding accesses a look-up table
- NVIDIA CUDA architecture supplies several banks of memory, accessible in parallel
- 1.82 look-ups per cycle per SM on average
- GeForce GTX 285 has 30 SMs, so can satisfy 55 look-ups per cycle per device
  - Compare to one per core in x86 processors

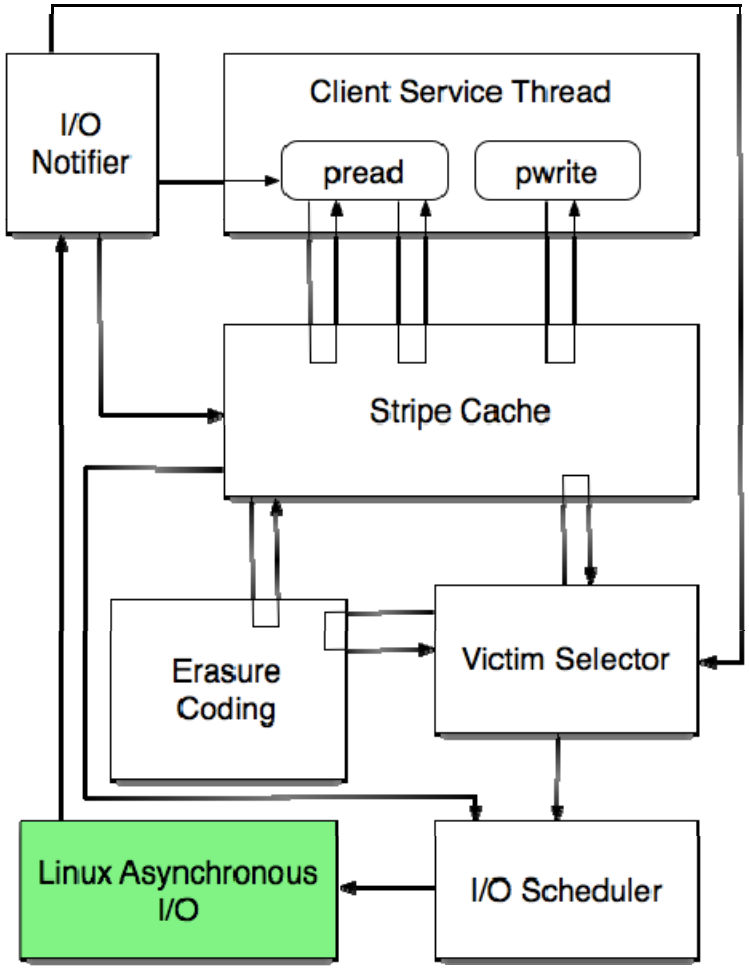


Performance: GeForce GTX 285 vs. Intel Extreme Edition 975



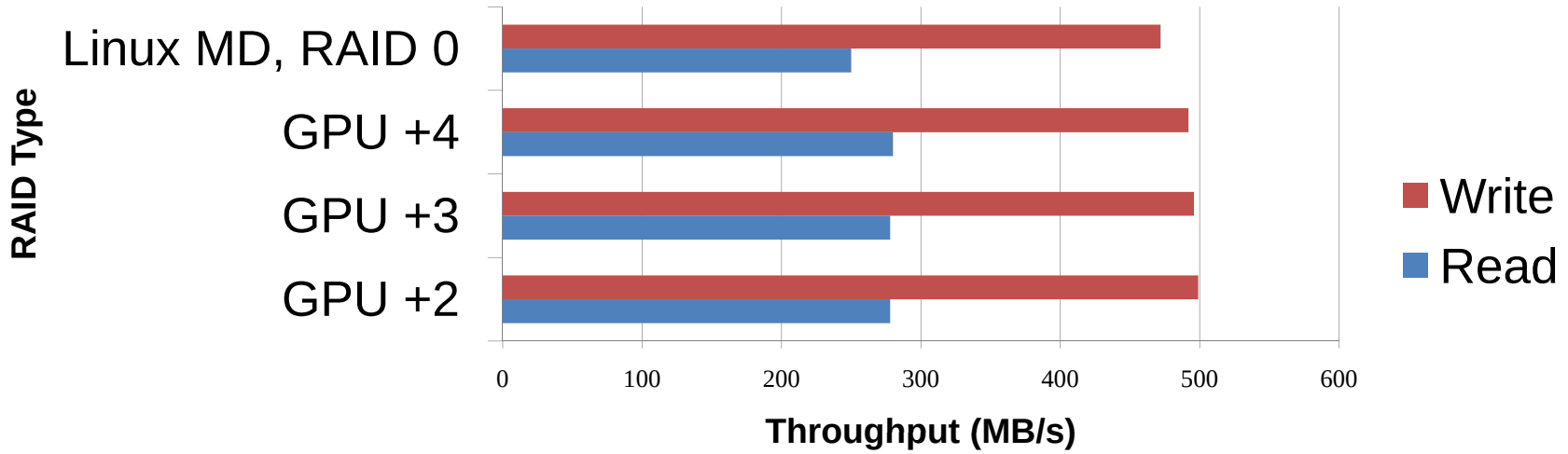
- GPU capable of providing high bandwidth at 6x-10x CPU speeds
- New memory layout and matrix generation algorithm for Reed-Solomon yields equivalent write and degraded read performance

A User Space RAID Architecture and Data Flow



- GPU computing facilities are inaccessible from kernel space
- Provide I/O stack components in user space, accessible via iSCSI
  - Accessible via loopback interface for direct-attached storage, as in this study
- Alternative: Micro-driver

Performance Results



- 10 GB streaming read/write operations to a 16-disk array
- Volumes mounted over loopback interface
  - stgt version used is bottleneck, later versions have higher performance

Advance: GPU-Based RAID

- Prototype for general-purpose RAID
  - Arbitrary parity
  - High-speed read verification
  - High-parity configuration for initial hardware deployments
    - Guard against batch-correlated failures
- Flexibility not available with hardware RAID
  - Lower-bandwidth higher-reliability distributed data stores

Future Work

- UAB
  - Data intensive “active storage” computation apps
  - Computation, Compression within the storage stack
  - NSF-funded AS/RAID testbed (.5 Petabyte)
  - Multi-level RAID architecture, failover, trade studies
  - Actually use the storage (e.g., ~200TB of 500TB)
- Sandia
  - Active/active failover configurations
  - Multi-level RAID architectures
  - Encryption within the storage stack
- Commercialization

Conclusions

- RAID reliability is increasingly related to BER
- Window of vulnerability during rebuild is dangerous, needs to be eliminated if system stays busy 24x7
  - Hot spares are inadequate because of elongated rebuild times under load
- High-parity GPU RAID can eliminate window of vulnerability while not harming performance for streaming workload
- Fast writes and degraded reads enabled by GPU make this feasible