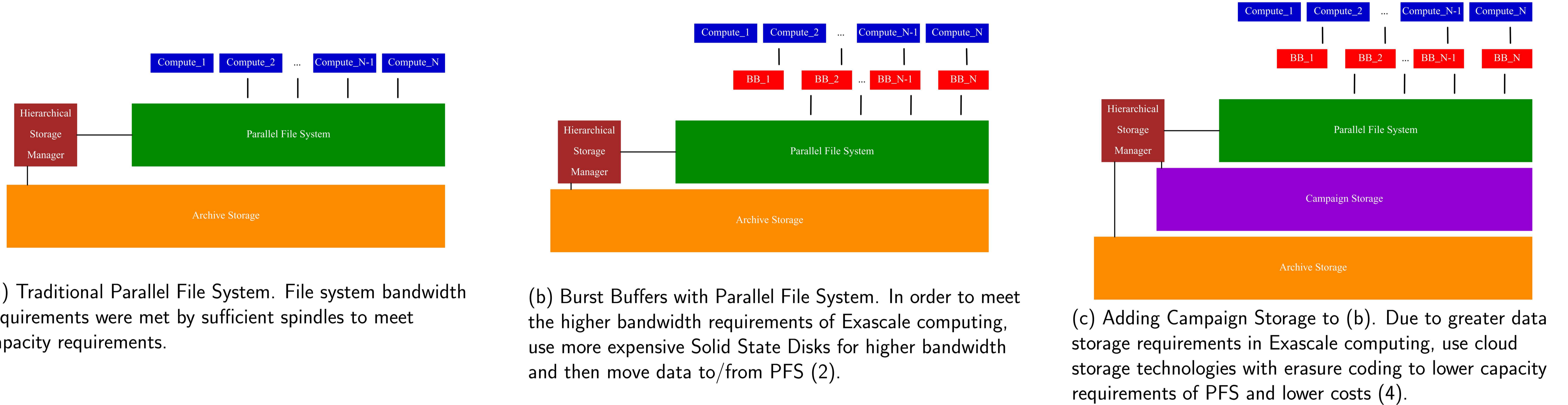


Campaign Storage: Erasure Coding With GPUs

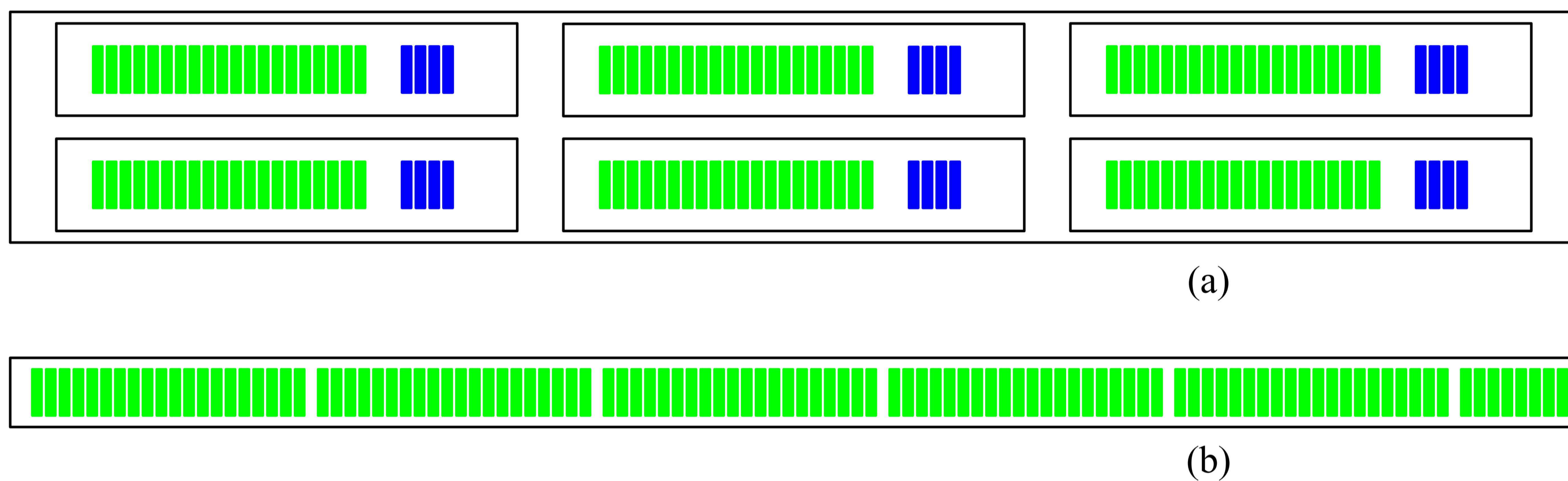
Walker Haddock¹, Matthew L. Curry², Purushotham V. Bangalore¹, Anthony Skjellum³

Evolution of HPC Storage Implementations

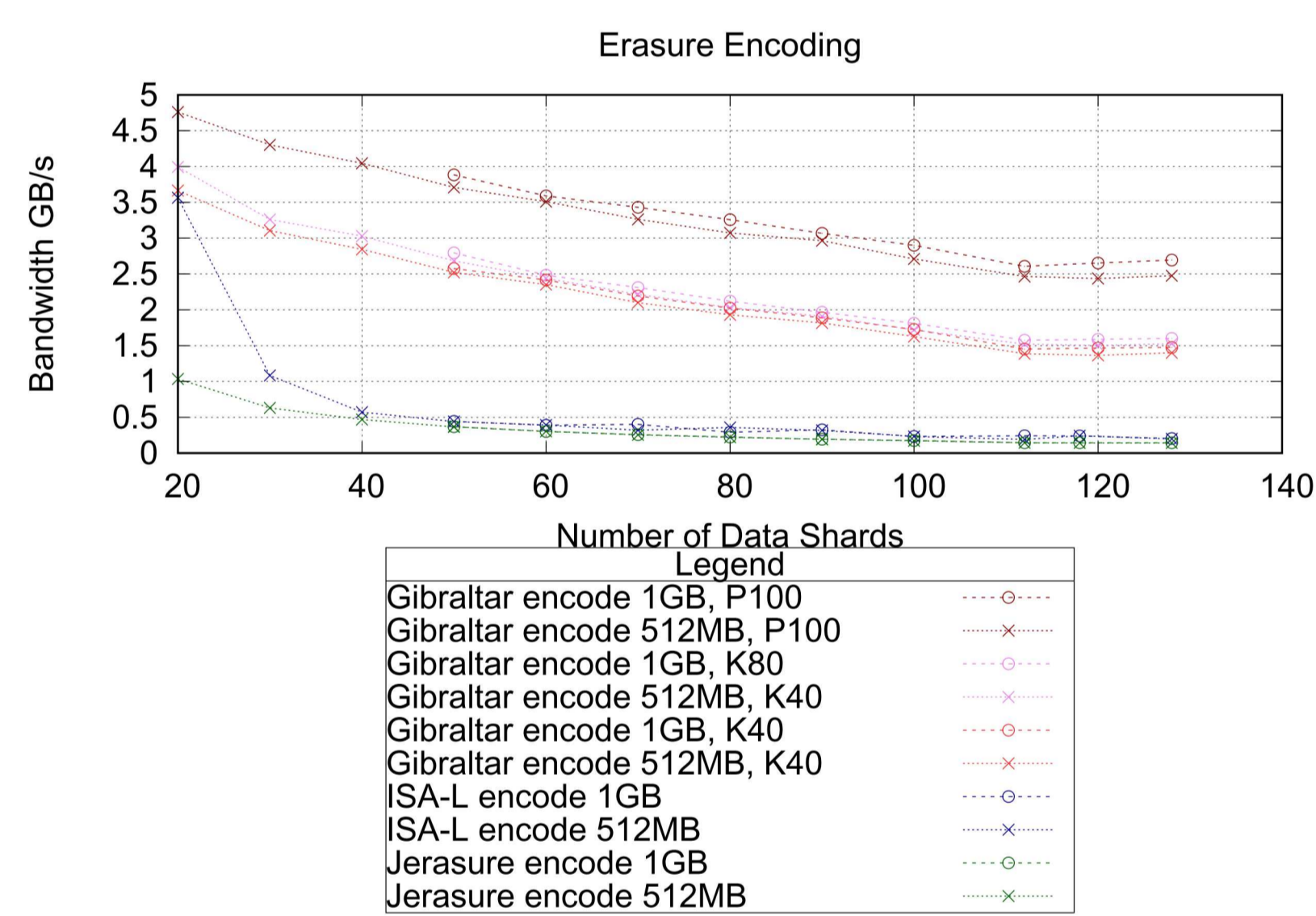


Lazy Erasure Repair

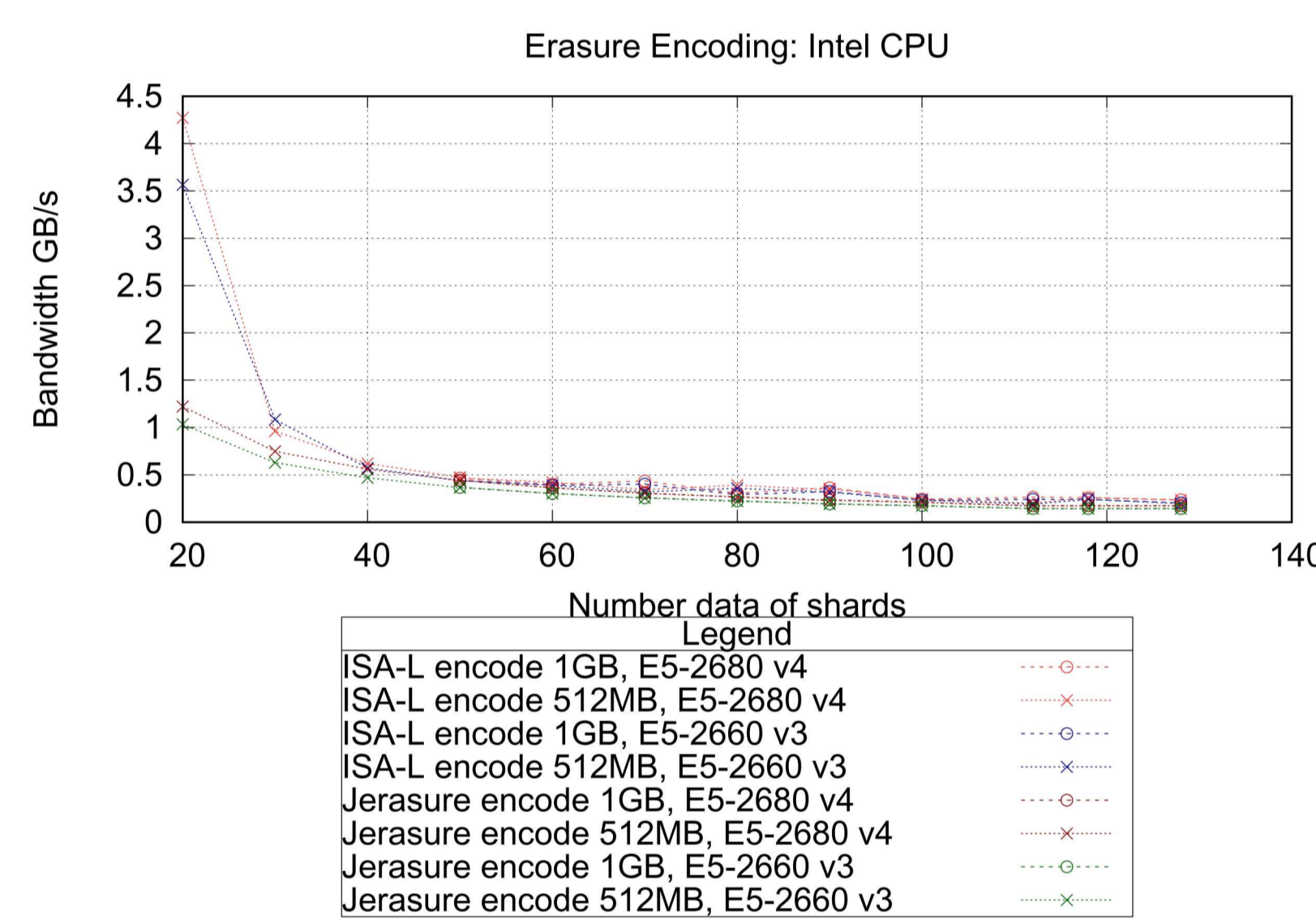
- Damaged objects can be read with acceptable performance – does not harm user experience!
- Changing the stripe from 20+4 to 120+24 decreases the urgency to rebuild while storage overhead is constant.



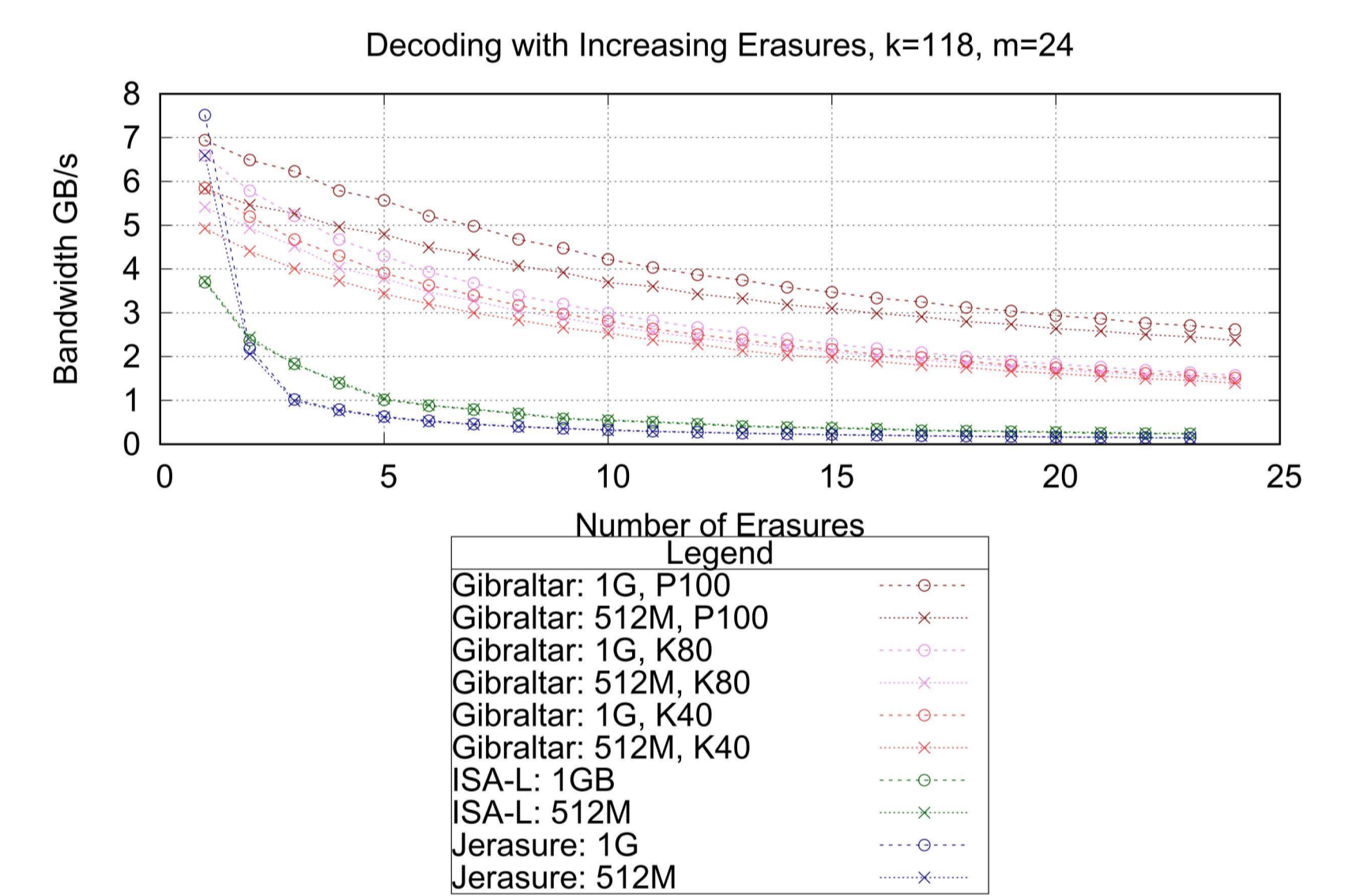
Results for GPU Assisted Erasure Coding



Comparison between Gibraltar (0; 1) GPU erasure coding on the NVIDIA K40 GPU with the Intel ISA-L (10) and Jerasure (7; 6; 8; 9) erasure encoding libraries. Gibraltar produces 1.5 GB/s at 128 data shards with 24 parity shards.



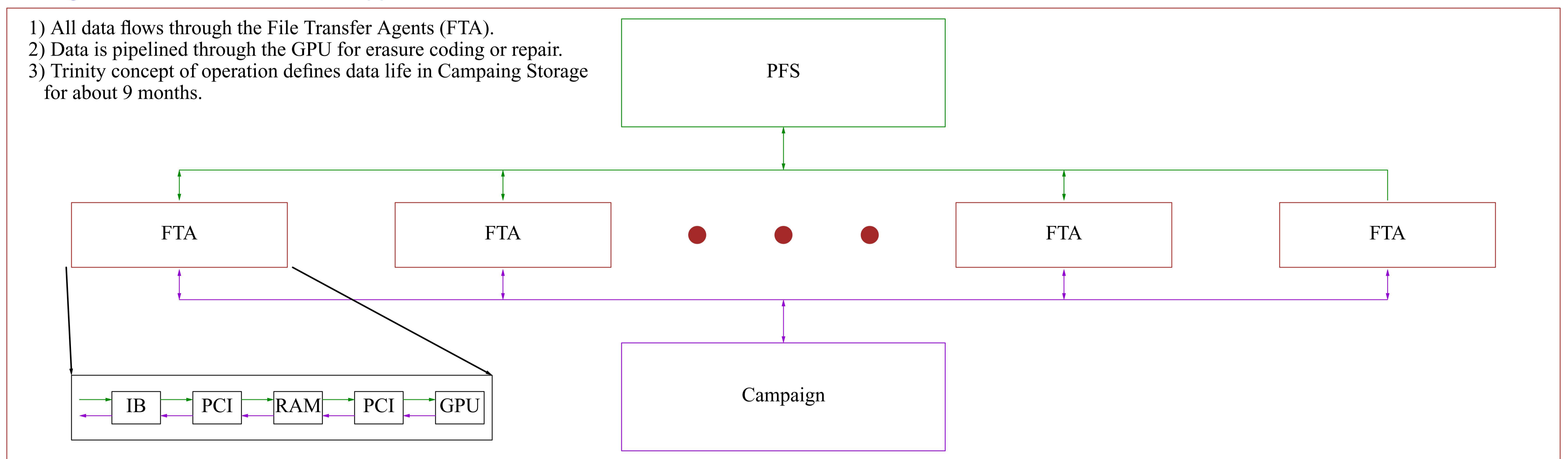
Benchmarks comparing two different generations of Intel CPUs erasure encoding with the Jerasure and ISA-L libraries. (These are all a single core of the CPU.)



Recovery rates while increasing erasures to 24. Gibraltar produces over 1.5 GB/s at 20 erasures while ISA-L has dropped to 267 MB/s at 20 erasures.

Erasure Coding Services on File Transfer Appliances

- 1) All data flows through the File Transfer Agents (FTA).
- 2) Data is pipelined through the GPU for erasure coding or repair.
- 3) Trinity concept of operation defines data life in Campaign Storage for about 9 months.



Bibliography

[1] Curry, M. L., Skjellum, A., Lee, W., et al. (2016). Accelerating Read-Scalable Coding in RAID systems with GPUs. In Proceedings of the 2016 IEEE International Parallel & Distributed Processing Symposium, pages 1-6. Miami, FL, USA: IEEE.

[2] Curry, M. L., Skjellum, A., Lee, W., et al. (2016). Gibraltar: A Read-Scalable Coding Library for Storage Applications on Programmable Graphics Processors. *Concurrency and Computation: Practice and Experience*, 30(10):2477-2496.

[3] Haddock, Walker, Curry, Matthew L., Bangalore, Purushotham V., and Skjellum, Anthony. (2017). GPU Erasure Coding for Campaign Storage HPC-DCDC RC'17.

Acknowledgment

This material is based upon work supported by the National Science Foundation under Grants No. OAC-1641310, CNS-0801487 and CNS-1229200. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.