What's the Story in EBS Glory: Evolutions and Lessons in Building Cloud Block Store

Weidong Zhang, Erci Xu, Qiuping Wang, Xiaolu Zhang, Yuesheng Gu, Zhenwei Lu, Tao Quyang, Guanqun Dai, Wenwen Peng, Zhe Xu, Shuo Zhang, Dong Wu, Yilei Peng, Tianyun Wang, Haoran Zhang, Jiasheng Wang, Wenyuan Yan, Yuanyuan Dong, Wenhui Yao, Zhongjie Wu, Lingjun Zhu, Chao Shi, Yinhu Wang, Rong Liu, Junping Wu, Jiaji Zhu, Jiesheng Wu

USENIX FAST'24.

2024. 10. 16

Presented by Jeyeon Lee

jeyeonlee@dankook.ac.kr

Elastic Block Storage

- A Storage Service of ALIBABA CLOUD
  - Services in the form of virtual block devices with
    - High performance
    - High availability
    - High elasticity

  The Goals of EBS

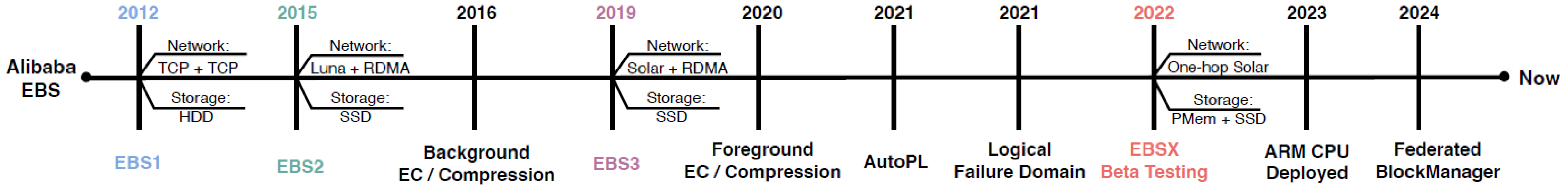
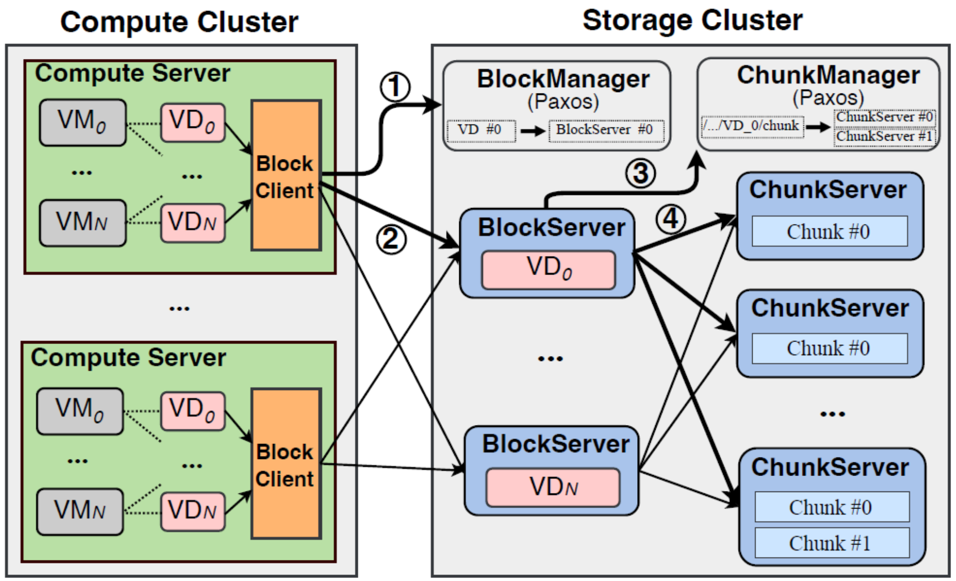


Figure 1: Alibaba EBS Timeline

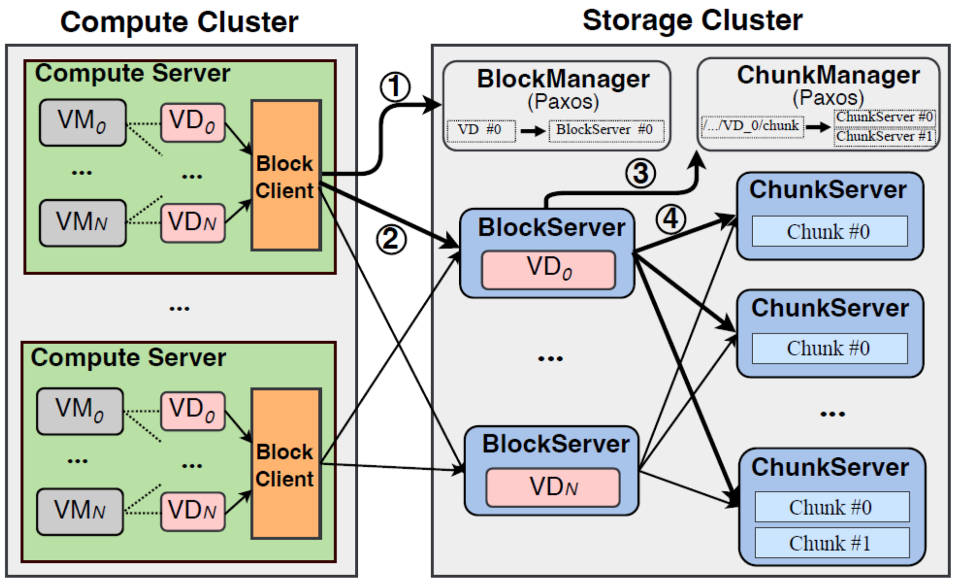
EBS1: An Initial Foray

- Based on a simple disaggregated architecture
  - Compute cluster
  - Storage Cluster
    - Three-way replicates each chunk and stores it as a 64MiB Ext4 file
    - Divides a VD into 64MiB chunks
    - Performs in-place update to the chunk
  - Network
    - Frontend: Compute Cluster-Storage Cluster
    - Backend: BlockServers-ChunkServers
    - Both rely on 10 Gbps TCP/IP network



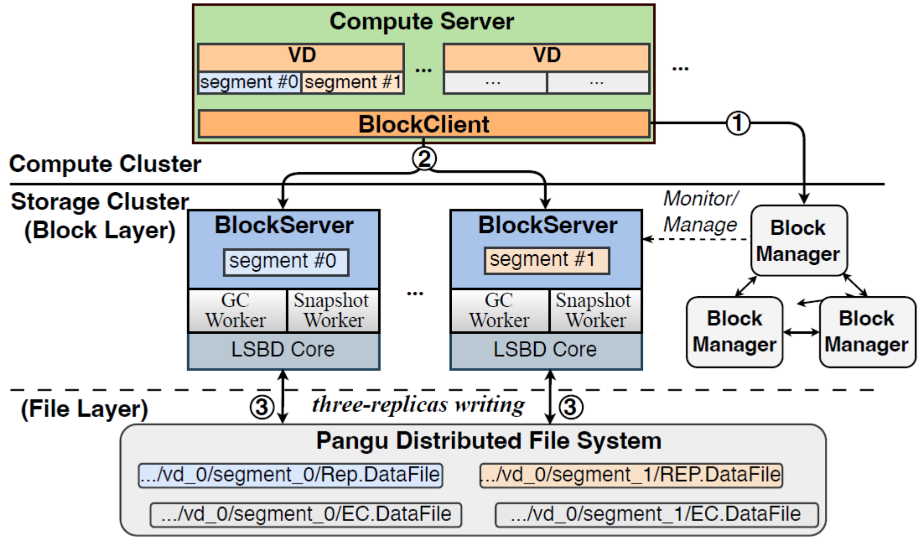
EBS1: An Initial Foray

- Limitations in the aspect of
  - Efficiency due to in-place update
    - Unable to use data compression and EC(Erasure Coding)
    - Compression non-deterministically alters the size of data
    - EC has a minimum size requirement
  - Performance due to the N-to-1 mapping
    - BlockServer can suffer from hotspot issues



EBS2: Speedup with Space Efficiency

- High performance and space efficiency
  - Builds on top of the Pangu(a distributed storage system)
  - BlockServer employs log-structured Design
    - Enables data compression and EC during GC
  - Disk segmentation
    - Alleviate the hotspot accessing in VDs
  - Network (2x25 Gbps)
    - Frontend: Luna(user-space TCP implementation)
    - Backend: RDMA network



EBS2: Speedup with Space Efficiency

- Disk Segmentation
  - A VD is divided into 128 GiB segment groups
  - A segment group comprises 32 GiB segments
    - Allocated in a round-robin fashion
  - Associates one segment with multiple DataFiles
    - Supports concurrent write

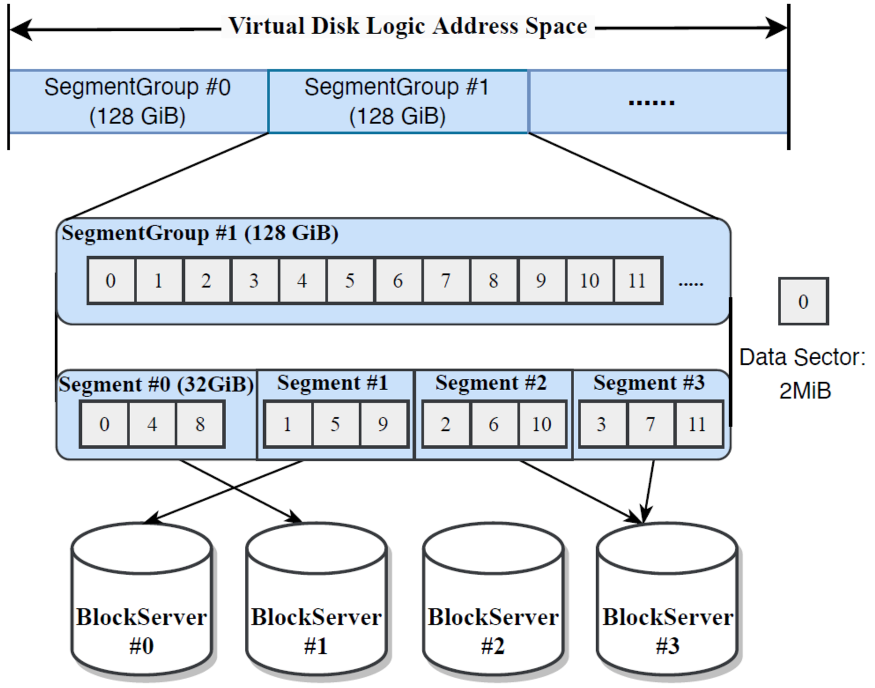
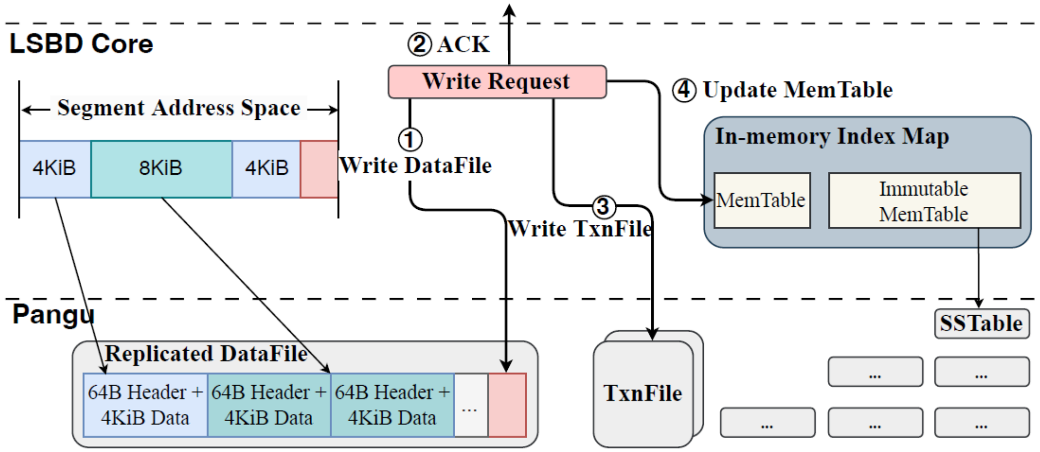


Figure 4: The Disk Segmentation Design of EBS2 (§2.2).

EBS2: Speedup with Space Efficiency

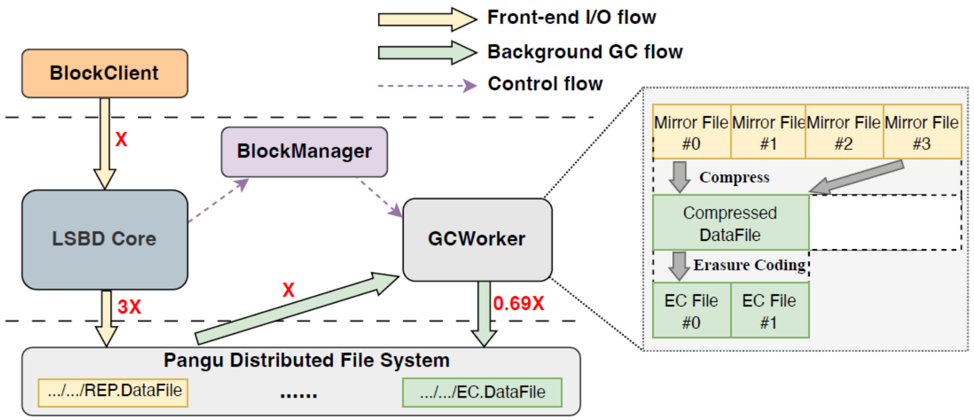
- Log-Structured Block Device
  - Supports the append-only semantics
  - Splits traffic into frontend and backend
  - Index Map(an LSM-tree)
    - Speeds up the locating process
    - Maps the VD's LBA to the DataFile ID, offset and length
  - TxnFile accelerates the index map rebuild



EBS2: Speedup with Space Efficiency

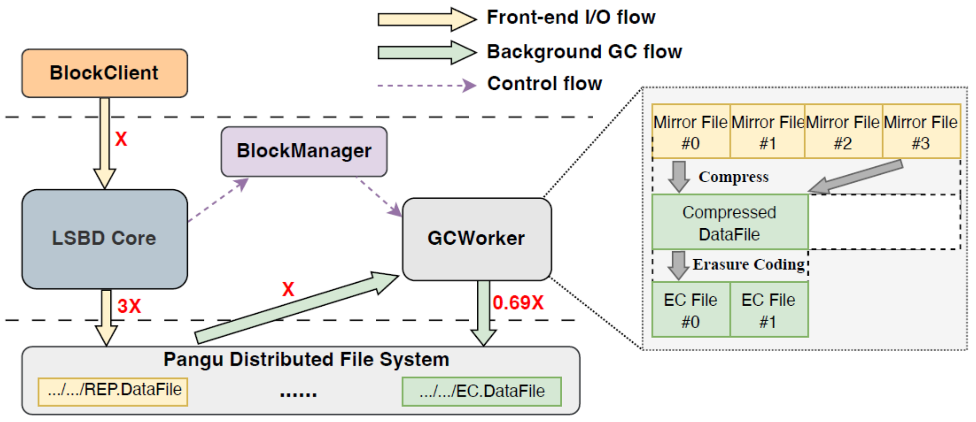
- GC with EC/Compression(LZ4/ZSTD)
  - Runs at the granularity of DataFile
  - Triggered when stale data within DataFiles reaches the threshold
  - Update the TxnFile and the in-memory index map

  - Avg. compression ratio ≈ 50.1%
  - Avg. # of replicas ≈ 1.29



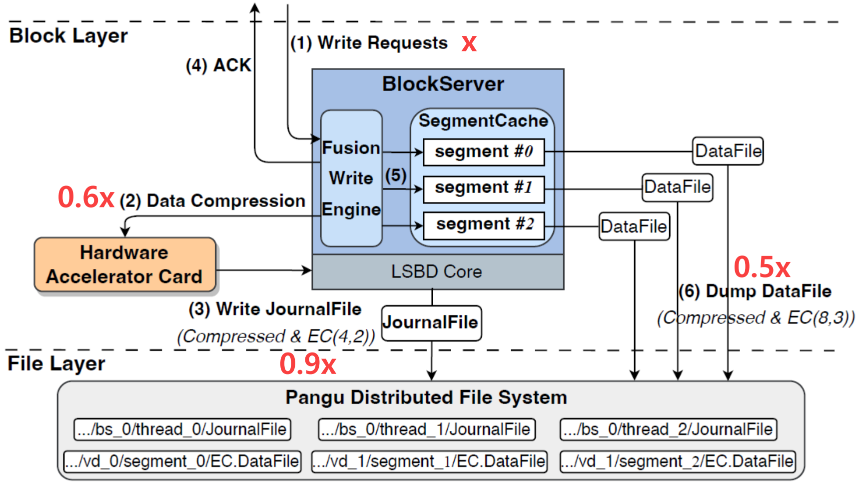
EBS2: Speedup with Space Efficiency

- Limitations
  - Heavy traffic amplification
    - Increased the overall traffic from 3 to 4.69
    - 3 from three-way replication + 1.69 from backend GC
    - Yields only 15.5% of the network bandwidth
  - Unable to adopt online EC/Compression
    - EC requires the data block to be at least 16 KiB
    - 70% of write request are smaller than 16 KiB



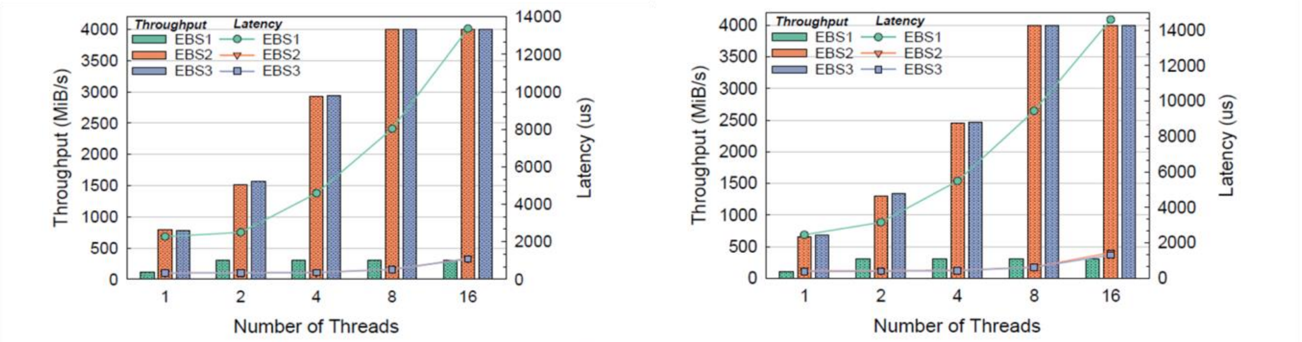
EBS3: Foreground EC/Compression

- Reducing network traffic amplification
  - Fusion Write Engine
    - Merge small writes until forms a 16 KiB DataBlock
    - Infrequent small writes are directly appended with three-way replication
  - FPGA-based compression offloading
    - Comp. data is verified with end-to-end CRC check
  - Network (2 x 100 Gbps)
    - adopts Solar(a UDP-based protocol) for both
  - Traffic amplification reduced to 1.59



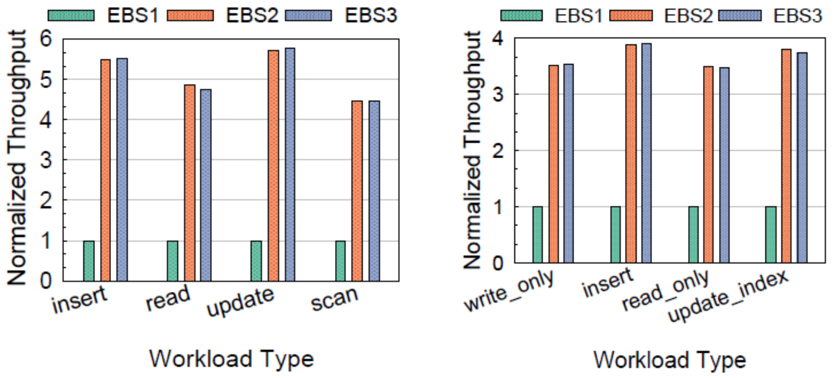
Evaluation

- Throughput under
  - Microbenchmark(by stressing the VD using FIO)
  - Application-based microbenchmark(RocksDB with YCSB / MySQL with Sysbench)



(a) Throughput and Latency of Random **Write** on Thread-to-core Pinning (b) Throughput and Latency of Random **Read** on Thread-to-core Pinning

Figure 9: Random Write/Read Latency of Each Generation EBS under Multiple Threads and 4 KiB-sized I/O. Thread-to-core pinning means that each thread occupies one CPU core exclusively.



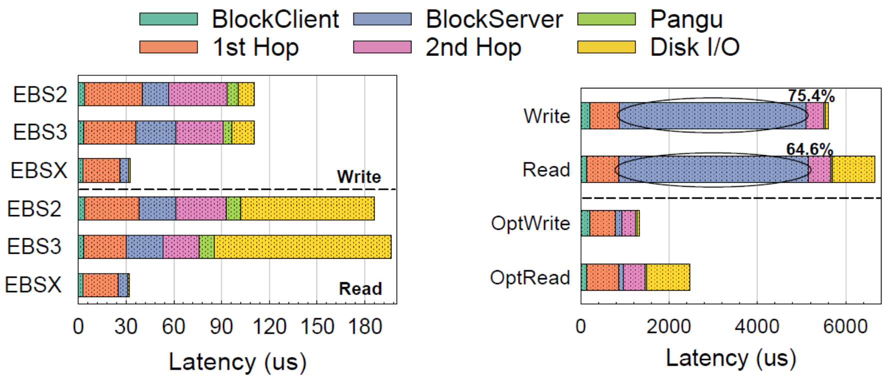
(a) Throughput Comparison of each EBS with **YCSB**

(b) Throughput Comparison of each EBS with **Sysbench**

Figure 10: Throughput Comparison (Normalized with EBS1).

Achieving Elasticity: Latency

- Is determined by the architecture
  - Networks: frontend(1st Hop) and backend(2nd Hop)
  - Software stacks: BlockClient, BlockServer and Pangu
  - Hardware: SSD I/O
- Optimizing hardware is straightforward
  - EBSX: BlockServer stores the data in PMem
- Tail latency by software stack may noise
  - OPT: Segregate the I/O flow from other tasks



(a) Average Latency Breakdown of EBS2, EBS3 and EBSX

(b) 99.999th Tail Latency Breakdown of EBS3

Figure 11: 8 KiB-Sized Avg. and Tail Latency Breakdown of EBS. ***1st hop:*** *network latency from compute to storage end.* ***2nd hop:*** *network latency from BlockServer to Pangu.*

Achieving Elasticity: Throughput and IOPS

- The upper bound is determined by the BlockClient
  - Backend can easily scale with parallelism
  - In EBS1: implemented as a kernel module
  - In EBS2: move to the user space (with user-space TCP stack)
  - In EBS3: offload to hardware (FPGA)
- High performance is not always needed
  - Base+Burst strategy
    - BaseIO: pre-defined
    - BurstIO: allocated based on available capability

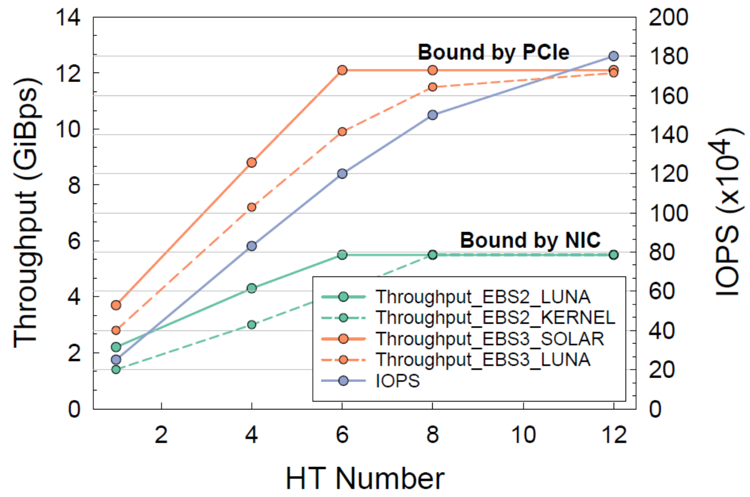


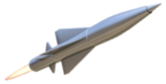
Figure 12: The maximum throughput and IOPS changes of Block-Client with different HT numbers.

Achieving Elasticity: Capacity

- Flexible space resizing
  - Segmentation design enables seamless support for VD resizing
  - Supports VD size ranging from 1 GiB to 64 TiB

- Fast VD cloning
  - Needs for a large volume of resources to be allocate in a short time
  - Uses the Hard Link of Pangu files
  - Enables the creation of up to 10,000 VD(each 40 GiB) in 1 min

Improving Availability

- Blast Radius
  - Individual 
    - When only one VD is influenced
    - e.g., An uncorrectable error inside the disk and a software bug
  - Regional 
    - When incurs deny of service for several VDs (e.g., BlockServer crash)
  - Global

Minimize Blast Radius: Control Plane

- BlockManager in EBS2
  - Single leader serves all the VDs in the cluster
  - Single metadata table hosts the metadata of VDs in the cluster

- Federated BlockManager
  - Multiple BlockManager for each cluster
  - Manages hundreds of VD-level partitions

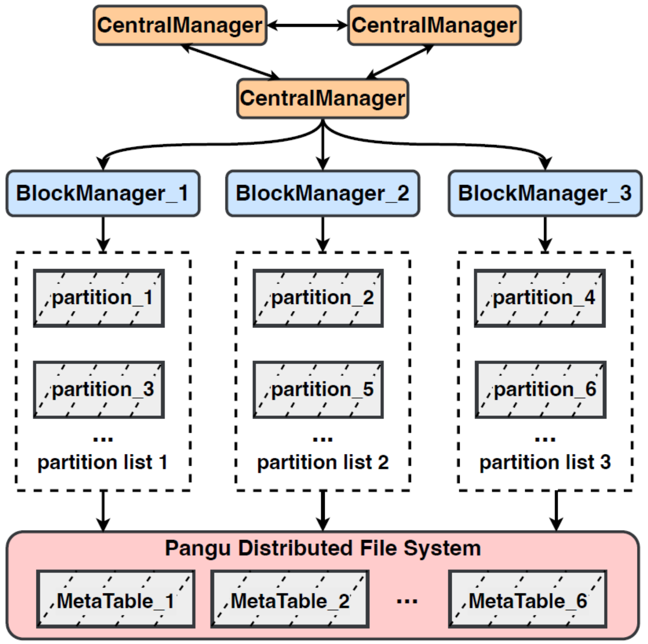


Figure 13: The architecture of Federated BlockManager.

Minimize Blast Radius: Data Plane

- When BlockServer in EBS2 crash
  - BlockManager migrates the segments to other BlockServer
  - Resume and crash again if the crash is caused by an error segment
  - Lesson learned
    - Failure typically originates from a single VD or segment
    - The root causes of the failure are mostly due to software errors
    - Cascading failure can propagate to the whole cluster

- Logical Failure Domain
  - Isolate suspicious segments into a small set of BlockServers

To Whom the EBS offloads

- Offloading BlockClient
  - BlockClient in EBS2 has become bottleneck
    - Calculating CRC
    - Encryption
    - Performing per-I/O table lookups
  - Using FPGA-based solution?
    - 37% of data corruption incidents was identified by CRC mismatches
    - Overheating, signal interference, timing issues
    - Later move on to adopt the ASIC-based solution

To Whom the EBS offloads

- Offloading BlockServer
  - To reduce costs while maintaining performance
  - Still remains 25 us latency with latency-optimized LZ4 compression
  - Using FPGA-based solution?
    - Faces similar instability issues
    - Reorienting the target of offloading toward server ARM CPUs

What if?

- The log-structured design was never adopted?
  - Foreground EC/compression necessitates a sufficient amount of data
- Built EBS with open-source software?
  - Tailored software stacks are needed to achieve low I/O latency
- Pangu and EBS were never separated?
  - Exceedingly complexed interfaces

Conclusion

- EBS: Cloud block store serviced by ALIBABA
  - Revisiting architecture evolutions
    - EBS1 → EBS2 → EBS3 → EBSX
  - Summarize develop lessons
    - High elasticity in latency, throughput, IOPS and capacity
    - Improving availability
    - Identifying the motivations and key tradeoffs in hardware offloading solutions
    - Identifying the pros/cons of alternative solutions

What's the Story in EBS Glory: Evolutions and Lessons in Building Cloud Block Store

Weidong Zhang, Erci Xu, Qiuping Wang, Xiaolu Zhang, Yuesheng Gu, Zhenwei Lu, Tao Quyang, Guanqun Dai, Wenwen Peng, Zhe Xu, Shuo Zhang, Dong Wu, Yilei Peng, Tianyun Wang, Haoran Zhang, Jiasheng Wang, Wenyuan Yan, Yuanyuan Dong, Wenhui Yao, Zhongjie Wu, Lingjun Zhu, Chao Shi, Yinhu Wang, Rong Liu, Junping Wu, Jiaji Zhu, Jiesheng Wu

USENIX FAST'24.

Thank You !

2024. 10. 16

Presented by Jeyeon Lee

jeyeonlee@dankook.ac.kr