Fighting with Unknowns: Estimating the Performance of Scalable Distributed Storage Systems with Minimal Measurement Data

Moo-Ryong Ra and Hee Won Lee[1]

AT&T Labs Research

May 23, 2019

[1]Presenter at MSST 2019

Motivation

- Goal
  - To estimate the performance of scalable distributed storage systems (e.g., Ceph and Swift) that use consistent hashing to distribute the workload as evenly as possible across all available compute resources
- Problem
  - Mathematical modeling or black-box approach needs a significant amount of efforts and data collection processes
- Our Approach
  - We propose a simple, yet accurate performance estimation technique for scalable distributed storage systems
  - Our technique aims to identify max IOPS for an arbitrary read/write ratio with a minimal evaluation process

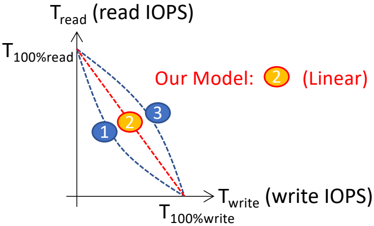
Our Model

Claim: If HW/SW/workload settings remain unchanged, the total processing capability (C) of a distributed storage system is invariant for a given IO size.

$$C = T_{read} + T_{write} \cdot f_{rw}$$

We can acquire f_{rw} value with just two data points:

$$f_{rw} = \frac{T_{100\%read}}{T_{100\%write}}$$



Our Model: arbitrary read/write ratio

Given that read/write ratio $= R_{read} : R_{write}$,

- read IOPS: $T_{read} = k \cdot R_{read}$
- write IOPS: $T_{write} = k \cdot R_{write}$

$$k \cdot R_{read} + k \cdot R_{write} \cdot f_{rw} = C$$

$$k = \frac{T_{100\% read}}{R_{read} + \{100 - R_{read}\} \cdot f_{rw}}$$

Once we get the value of k, it is trivial to obtain T_{read} and T_{write}.

Our Model: mixed IO sizes

Suppose that we have heterogeneous IO sizes, $S_1, S_2, \cdots, S_N$ and know the proportion of each IO size to the total IOs, $P_1, P_2, \cdots, P_N$ where $\sum_{i=0}^{N} P_i = 1$.

$$\bar{k}^{S_1} = \frac{P_1 \cdot T_{100\%read}^{S_1}}{R_{read} + \{100 - R_{read}\} \cdot f_{rw}^{S_1}} = P_1 \cdot k^{S_1}$$

$$\vdots$$

$$\bar{k}^{S_N} = \frac{P_N \cdot T_{100\%read}^{S_N}}{R_{read} + \{100 - R_{read}\} \cdot f_{rw}^{S_N}} = P_N \cdot k^{S_N}.$$

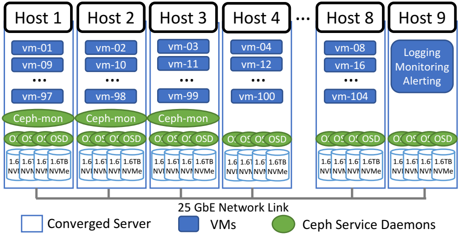
Total IOPS can be obtained by:

$$T_{total} = \sum_{i=1}^{N} \{R_{read}^{S_i} + R_{write}^{S_i}\} \cdot \bar{k}^{S_i} = 100 \cdot \sum_{i=1}^{N} P_i \cdot k^{S_i}$$

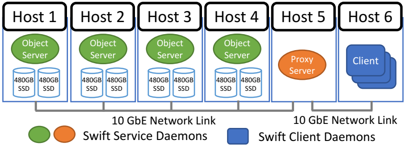
Evaluation

We set up two different distributed storage systems:

- Ceph
  - Block Storage, Strong Consistency, 3x Replication
  - FIO: 104 OpenStack VMs, each running 8 FIO jobs



- Swift
  - Object Storage, Eventual Consistency, 3x Replication
  - COSBench: 32 workers

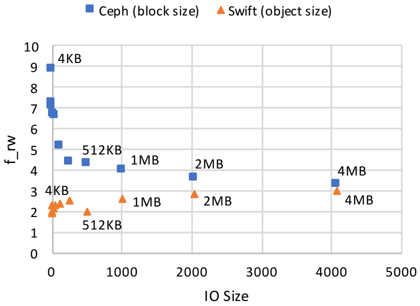


Meaning of f_{rw}

[Our Model] Total processing cap.(C) is invariant per IO size:

$$C = T_{read} + T_{write} \cdot f_{rw}$$

where $f_{rw} = \frac{T_{100\% read}}{T_{100\% write}}$.



Note:

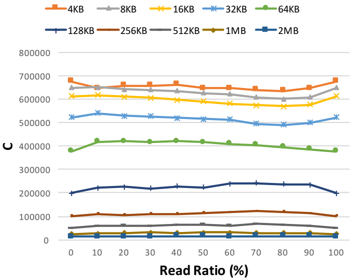
- f_{rw} reflects the load difference b/w read and write operations
- The amount of work required for read and write operations can be very different per storage system implementation and their configurations

Total Processing Capacity (C) per IO Size

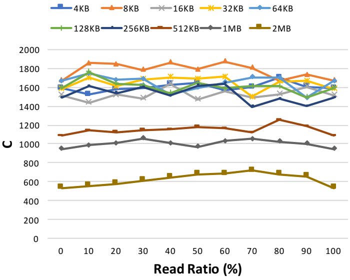
[Our Model] Total processing cap.(C) is invariant per IO size:

$$C = T_{read} + T_{write} \cdot f_{rw}$$

where $f_{rw} = \frac{T_{100\%read}}{T_{100\%write}}$.



(a) Ceph



(b) Swift

Figure: C value

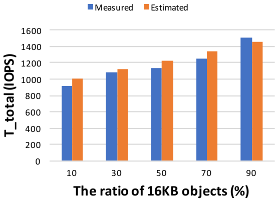
Performance Estimation

For obj size S_i, when read/write ratio $= R_{read} : R_{write}$:

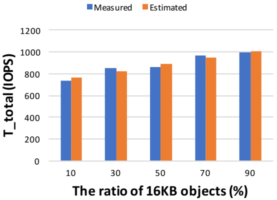
$$k^{S_i} = \frac{T^{S_i}_{100\%read}}{R_{read} + \{100 - R_{read}\} \cdot f^{S_i}_{rw}}$$

For mixed obj sizes ($P_i =$ proportion of obj size S_i to total objs):

$$T_{total} = 100 \cdot \sum_{i=1}^{N} P_i \cdot k^{S_i}$$



(a) 16KB Read+1MB Read



(b) 16KB RW+512KB RW

Figure: IO workloads with mixed object sizes on Swift cluster

Performance Estimation Error

The errors between estimated and measured total IOPS are less than 9%.

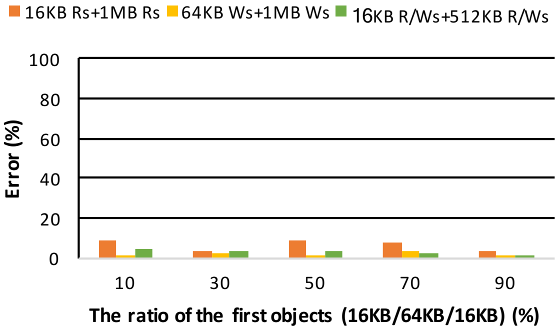


Figure: Estimation error on Swift Cluster

Conclusion

1. We proposed a novel technique to accurately estimate the performance of an arbitrarily mixed workload, in terms of read/write ratio and IO size
2. Our simple technique requires only a few data points – i.e., 100% read IOPS and 100% write IOPS for each IO size
3. Our technique can be applicable to any distributed storage systems that distribute the load evenly across the available hardware resources

Any Questions?

We are hiring a couple of systems researchers:

- Senior Inventive Scientist (for fresh PhDs)
- Principal Inventive Scientist (for mid-career professionals)

Contact: Hee Won Lee, PhD

Email: knowpd@research.att.com

Location: Bedminster, New Jersey