Design and Modelling Cloud-based Burst Buffers

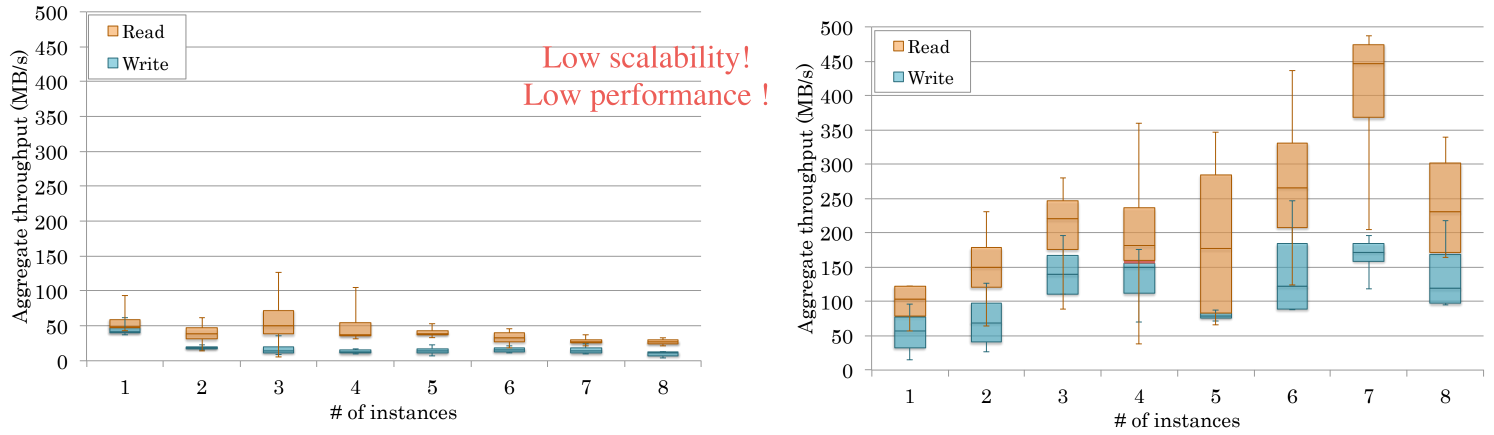
Tianqi Xu†, Kento Sato‡ and Satoshi Matsuoka†

† Tokyo Institute of Technology ‡ Lawrence Livermore National Laboratory

Background

- Public cloud: high scalability, high computational resources on-demand usage available.
- Such cloud environments are suitable for large scale data intensive computation.
- However there are two major challenges in cloud storages:
  - Low I/O performance. Loose consistency model,
- We have proposed Cloud-based Burst Buffers (CloudBB) [1] as a new tier in cloud storage hierarchy to improve I/O performance and consistency while using cloud storages.

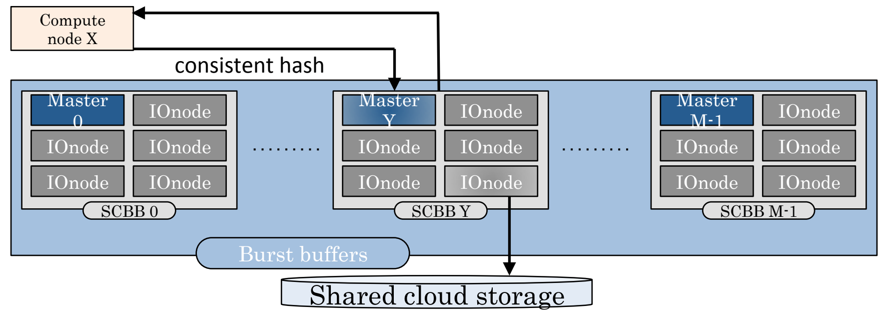
I/O throughput of Amazon S3



N-1 throughput
N clients access to the different part of a shared file

N-N throughput
Each client access to its own file

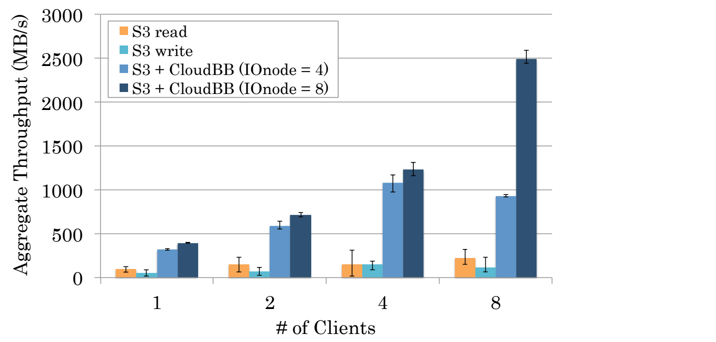
Overview of Cloud-based Burst Buffers



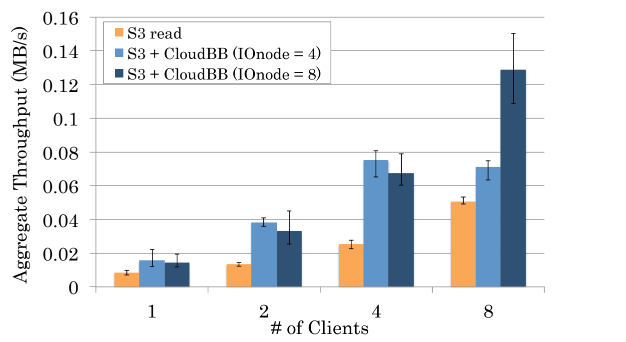
- Burst buffers are several dedicate nodes to provide remote data cache with high throughput and low latency.
- Our system consists of several SCBBs (Sub CloudBB).
- In each SCBB, there is a Master and several IOnodes
  - Masters control the IOnodes in the same SCBB, manage file metadata and handle I/O requests from Compute Nodes.
  - IOnodes store actual data and transfer data with Compute Nodes.
- Consistent hash of file path is used to distribute workload among each SCBBs.

Problem

- Trade off: more IOnodes -> high performance & high cost
  less IOnodes -> low performance & low cost
- Since we use several additional nodes as burst buffer nodes, it is important to choose the number of IOnodes carefully to achieve high performance as well as save cost.



Sequential I/O Performance Comparison



Random I/O Performance Comparison

Proposal

We propose performance model for our Cloud-based Burst Buffers system.
- predict the performance.
- help to determine the optimal configuration.

Performance Model of Cloud-based Burst Buffers

In our model, we make two assumptions:
- The master node evenly distributes data of applications across burst buffers nodes evenly so that I/O workloads are balanced across burst buffer nodes;
- If multiple compute nodes access to a single burst buffer node, the bandwidth is divided by the number of compute nodes accessing to the node.

Optimal Configuration

First, in order to achieve both cost and time optimal, we define the overall optimal configuration as follow:

$$\text{Optimal Configuration} = \min\{Time \times Cost\}$$

Cost

According to cloud pay-as-you-go pricing policy:

$$Cost = Time \times (P_C \times N_C + P_I \times N_I + P_M \times N_M)$$

Time

The overall *Time* can be computed as:

$$Time = (T_C + \frac{D_{input}}{Thr_{Cloud}} + \frac{D_{buff}}{Thr_{CloudBB}}) \times (r + \frac{1}{N_C} \times (1-r))$$

Since the throughput of IOnode are shared by multiple compute nodes. Thus, overall average I/O throughput, $Thr_{CloudBB}$, can be computed as:

$$Thr_{CloudBB} = \sum_{i=0}^{N_C - 1} P_i \times Thr_m$$

P_i denotes probability where i number of compute nodes accessing the same IOnode. Hence, the Thr_m and P_i can be computed as:

$$Thr_m = \max\{Thr_{\text{Compute Node}}, Thr_{\text{IOnode}}\}$$

$$P_i = \frac{\binom{N_C - 1}{i}(N_I - 1)^{N_C - i - 1}}{N_I^{N_C - 1}}$$

P_C	Unit price of Compute Node
P_I	Unit price of IOnode
P_M	Unit price of Master Node
N_C	The number of Compute Node
N_I	The number of IOnode
N_M	The number of Master Node
T_C	The total time in computation
D_{input}	The total input size
D_{buff}	The total data size can be buffered in burst buffer
r	The radio of tasks must be executed serially in total tasks
Thr_{Cloud}	The average throughput of cloud storage
$Thr_{CloudBB}$	The throughput of CloudBB under the given configuration
Thr_m	The maximum throughput of IOnode

variables used in model

Reference

[1] T. Xu, K. Sato, and S. Matsuoka. "Cloud-based Burst Buffers for I/O Acceleration". In Summer United Workshops on Parallel, Distributed and Cooperative Processing (SWoPP), 2015, July 2015.

[2] Montage. [Online]. Available: http://montage.ipac.caltech. edu/docs/grid.html

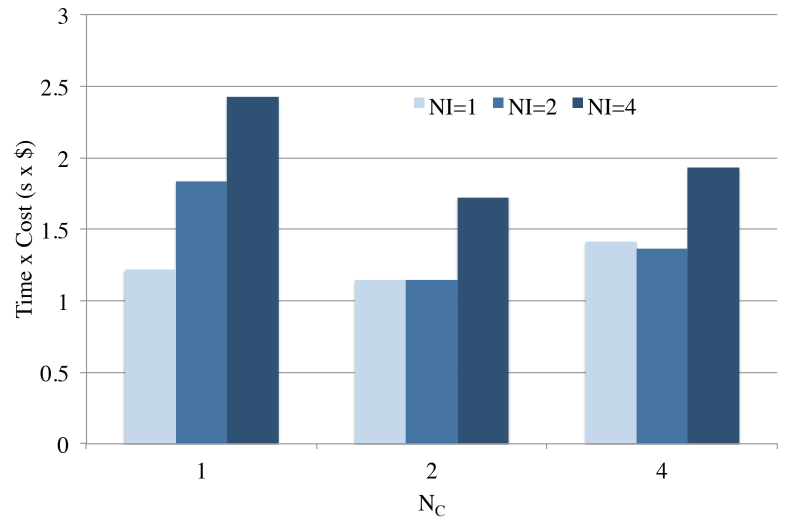
Evaluation and Model Prediction

System	Amazon EC2
Region	Tokyo
Instance Type	m3.xlarge
vCPUs	4
Memory	15GiB
Instance Storage	2*40 GB(SSD)
Network	135 MB/s
Cost	$0.405 per hour
Cloud Storage	Amazon S3
Mount Method	s3fs

Experiment Environment

D_{input} (MB)	25
D_{buff} (MB)	215
Total I/O Size (MB)	224
Total Read Size(MB)	147
Total Write Size(MB)	76
T_C (s)	2.638
r	0.45
P_C=P_I=P_M ($)	0.405
N_C	{1, 2, 4}
N_I	{1, 2, 4}
N_M	1
Thr_{Cloud}	18 MB/s
Thr_m	135 MB/s

Dataset and Experiment Setting Details



Montage [2] Results

# of Compute Nodes	Optimal number of IOnodes (Model Prediction)	Optimal number of IOnodes (Experiment Result)
1	1	1
2	1	1
4	2	2

Montage [2] Prediction results

According to the results of Montage [2] and the prediction, our model can predict the performance and optimal configuration while using our CloudBB system.

Conclusion

- We propose performance model for our Cloud-based Burst Buffers system.
- We validate our model using the experiment results of a HPC application on real public cloud, Amazon EC2.

Acknowledgement

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344. (LLNL-POST-676072)

This research was supported by JST, CREST (Research Area: Advanced Core Technologies for Big Data Integration).

This research made use of Montage, funded by the National Aeronautics and Space Administration's Earth Science Technology Office, Computation Technologies Project, under Cooperative Agreement Number NCC5-626 between NASA and the California Institute of Technology. Montage is maintained by the NASA/IPAC Infrared Science Archive.