TEXAS TECH UNIVERSITY™

DISCL

Two-Mode Data Distribution Scheme for Heterogeneous Storage in Data Centers

Authors: Wei Xie, Jiang Zhou, Mark Reyes, Jason Noble and Yong Chen
Presenter: Wei Xie

IEEE BigData 2015, Santa Clara, CA, USA
October 30th, 2015

Cloud and Autonomic Computing Center

TEXAS TECH UNIVERSITY

NIMBOXX

NSF

Outline

- Challenges in distributed storage
- Motivation of Two-Mode data distribution
- Design of Two-Mode data distribution
- Analysis of Two-Mode distribution
- Conclusion and future work

Distribution Storage Challenge

- Big Data needs distributed storage system to handle PB-scale data on tens of thousands of storage nodes
- A highly scalable, high-performance, reliable and efficient distributed storage is needed
- How to manage data distribution in such scale

✓ Table management: Lustre, HDFS

- *A centralized or distributed table that contains data-node mapping information*
- *Table can be large and enormous to synchronize among nodes*
- *SPOF and performance bottleneck*

✓ Algorithm management: Ceph, Dynamo, Sheepdog

- *Data are distributed to nodes based on some type of algorithm, e.g. Consistent Hash*
- *Datum id -> hash value -> node id*
- *Minimal metadata, no central node, better scalability, node addition/removal*

Distribution Storage Challenge (cont'd)

- Requirement of data distribution

1. Scalability
   - *Algorithm management much better*
2. Load balance (based on capacity)
   - *Data need to be proportionally distributed according to nodes' capacity*
   - *More data on large-capacity HDDs*
3. Handles node addition/removal
4. Data replication for fault-tolerance
5. High performance
   - *Throughput of storage nodes need to be fully exploited*
   - *More data on faster SSDs*

- Consistent hashing and CRUSH handle 1-4 well but not on 5

✓ Load balance vs. high performance

Motivation

- Conflict between load balancing and high performance in heterogeneous storage environment

✓ SSD: high throughput, small capacity

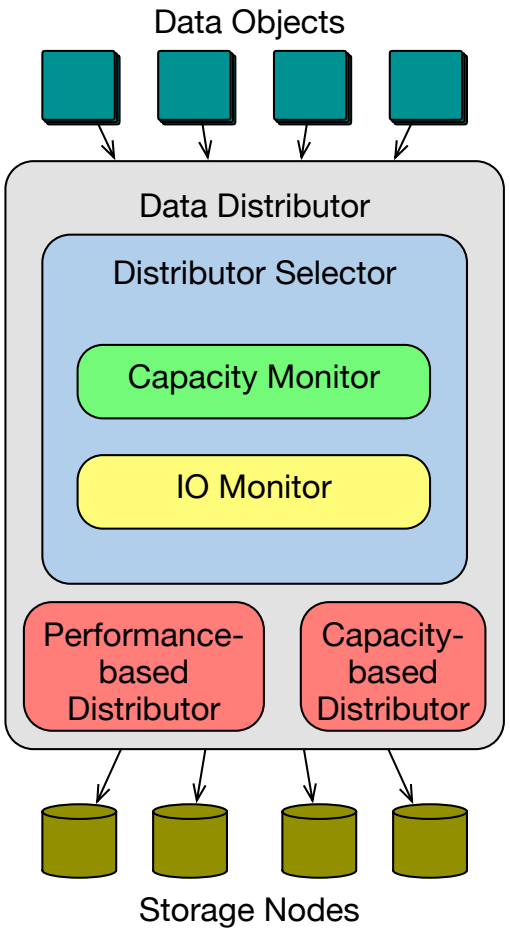
✓ HDD: low throughput, large capacity

Device name	Capacity(GB)	Throughput (MB/s)
Raw Seagate hard disk	1000	146
WD Red RAID5 with 4 disks	500	263
Samsung 850 EVO	256	540

Motivation

- Traditional way of using SCMs (i.e. SSD) as cache is not ideal
  - ✓ Caching/buffering generates extensive writes to SSD, which wears out the device
  - ✓ Need SSD-aware caching/buffering scheme
  - ✓ Not fully utilize capacity of SSDs
  - ✓ The capacity of SSDs is growing fast
- Put SCMs at the same level as HDDs in distributed storage system
  - ✓ Another faster tier of storage
  - ✓ No need to do cache replacement or buffer flushing
  - ✓ User sees the storage system with combined capacity and maximized performance
  - ✓ Less write to SSDs
  - ✓ Conflict between load balancing and high performance could be a problem

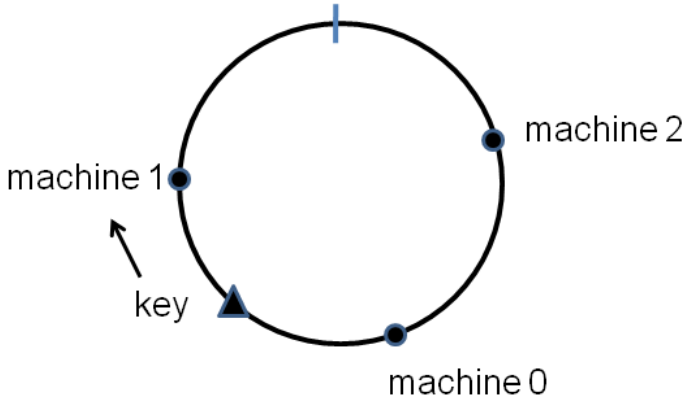
Two-Mode Data Distribution



- Traditional data distribution only cares about load-balance, i.e. uses capacity-based distribution
- We propose to use performance-based and capacity-based distributor alternately
- Two system modes/states
- Switch between two mode is based on the use of capacity and IO workload

Background: Consistent Hashing

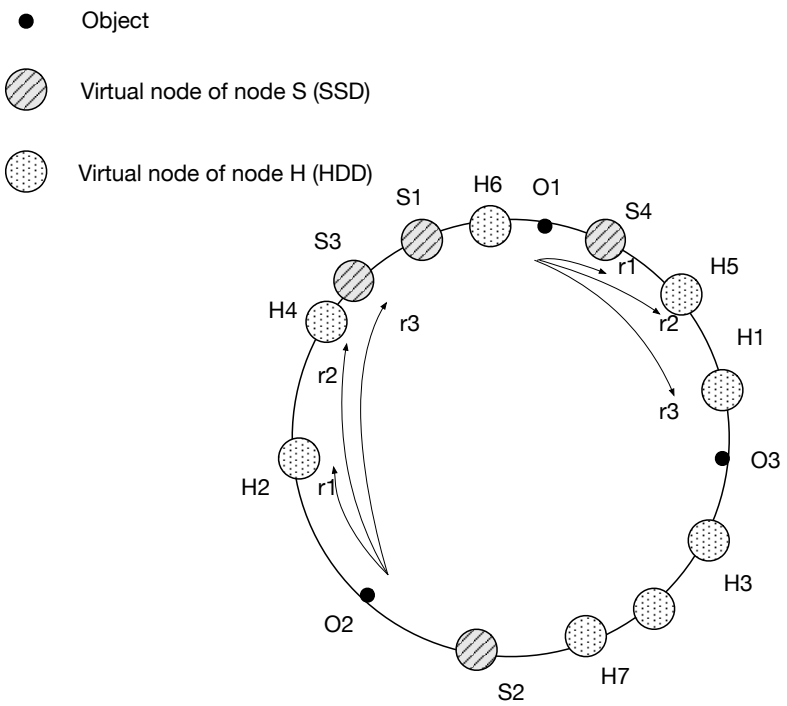
- Widely used in distributed systems and P2P network
- Both objects and nodes are hashed and placed on a hash ring: object is placed based on the node next to the object in clockwise direction
- Load-balance and node weight achieved by virtual nodes



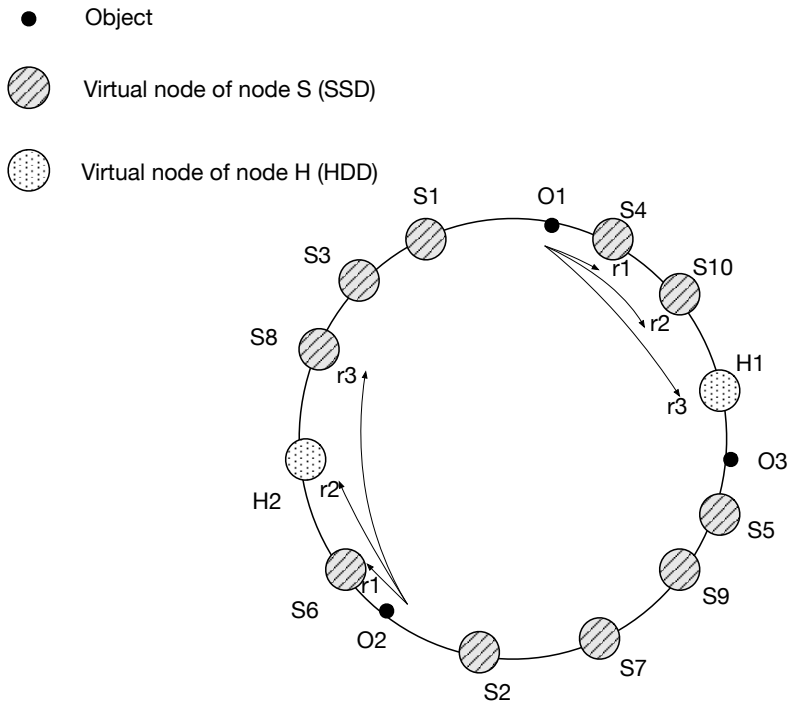
http://michaelnielsen.org/blog/consistent-hashing/

Two-Mode on Consistent Hashing

Node name	Storage type	Capacity(GB)	Throughput (MB/s)	Number of Vnodes (capacity mode)	Number of Vnodes (performance mode)
Node S	SSD	250	70	4	10
Node H	HDD	500	350	8	2



Capacity-mode



Performance-mode

Two-Mode on Consistent Hashing

Mode	O1's replicas placed on	O2's replicas placed on
Capacity	S4, H5, H1	H2, H4, S3
Performance	S4, S10, H1	S6, H2, S8

- One replica of O1 and O2 will migrate between the two nodes
- Even virtual nodes might change, the placement of physical node might still be the same

Read and Write Operation in the Two-Mode Data Distribution

- Two mode may introduce two possible distribution choices

E.g. write in capacity mode but read in performance mode

- Read scheme: read based on current mode first and then the other mode;
- Chances to retrieve data in current mode is high because it is very likely there is overlap in the nodes chosen in two modes

Read and Write Operation in the Two-Mode Data Distribution

- Write scheme is more complicated

✓ find obj on nodes determined by current mode ->

1. obj not in any nodes ->
   1. *find on nodes determined by the other distributor->*
      a. not found -> new data write to nodes in current mode
      b. found -> write to current nodes but invalidate obj on the nodes in the other mode
2. obj in some nodes ->
   1. *write to current nodes but invalidate obj on the nodes in the other mode*
3. obj in all nodes ->
   1. *write to nodes in current mode*

- The write will need double time for mapping if object was not lastly written in current mode

Algorithm Analysis

- Two-mode on data distribution time
  - ✓ Worst case: double the data distribution time to calculate two sets of hash value, double local search time
  - ✓ For read, it is less likely to need double time because only one replica is needed for read
  - ✓ For write, it is more likely to need double time if mode mismatch occurs
  - ✓ Need more evaluation
- Two mode on memory footprint
  - ✓ Double the memory footprint to memorize node information in two modes
- Two mode could potential complicate data recovery process due to node failure

Algorithm Analysis

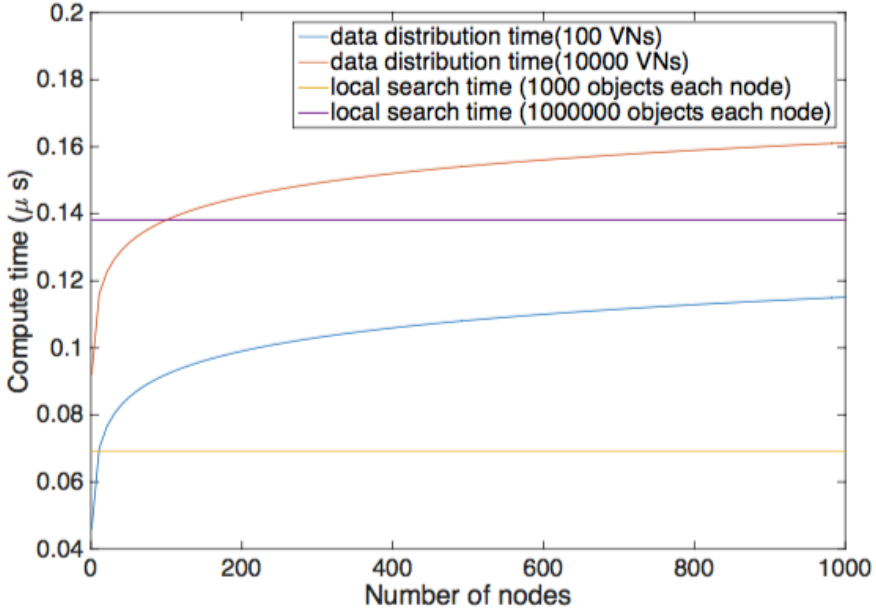


Fig. 4. The data distribution time in Consistent Hash. The two lines for data distribution time are with 100 virtual nodes and 10000 virtual nodes, respectively. Local search time is the time to search an object on a data node with known id. Some data are gained from [18].

- Data distribution time: compute hash value/random number to determine the node - O(log(n))
- Local search time: find data object in local node – O(1)
- Doubling distribution and local search time is not impractical for CH

Conclusions and Future Work

- Reconsider data distribution with heterogeneous storage devices with distinct performance metrics
- Existing algorithm-based data distribution schemes focus on load-balancing but ignores performance
- Two-mode scheme targets at providing maximized performance while still maintain load-balance, without drastic change to existing data distribution algorithms
- Analysis shows potential of the scheme
- Implementation and evaluation of two-mode scheme in Sheepdog
- Design mode transition scheme to minimize data migration overhead

Acknowledgement

This research is supported by the National Science Foundation under grant IIP-1362134 (through the Nimboxx membership contribution) and CNS-1338078.

Thanks!

Any questions?