Consistent Hashing-based Algorithm for Heterogeneous Storage Systems

Jiang Zhou, Wei Xie, and Yong Chen
Department of Computer Science, Texas Tech University

Center for Cloud and Autonomic Computing

Abstract

- To design a data placement algorithm for heterogeneous storage composing with hard disk drivers (HDDs) and solid state disk (SSDs)
- Based on hashing algorithms to keep its inherent features while making better utilization of heterogeneous storage devices
- Two novel data distributed placement algorithms based on consistent hashing for heterogeneous storage systems
- Improve the I/O performance while keeping storage space load balanced and well utilized for capacity, bandwidth and cost

Motivation and Goals

- Data explosion in big data era exposes significant challenges to the underlying storage systems
- Heterogeneous storage system is a very promising trend in data centers
- Object storage and hashing-based distribution design are usually adopted to address the scalability and availability problems
- Provide a uniform and balanced data placement based on consistent hashing algorithm, which is widely used in modern distributed storage systems, such as Dynamo, Chord, Cassandra, Ceph and Sheepdog

Methods and Techniques

Based on consistent hashing to keep its inherent features and make better use of different devices

(1) StrategyCHT

- Adopt a *unified* hashing ring to manage heterogeneous nodes
- Maintain attributes of each node
- Use a selection strategy for mapping nodes

(2) HiCH (hierarchical consistent hashing)

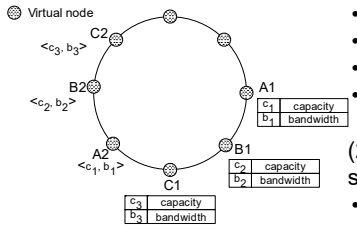
- Divide heterogeneous nodes into buckets
- Apply *separate* hashing rings for each bucket
- Place data into various hashing rings according to the hotness, access time, and other data access patterns

Current Status

- Proposed two different ideas based on consistent hashing for heterogeneous data placement
- Completed the design details of the algorithms
- Implemented the algorithms on real distributed storage systems
- Conducted extensive analysis and evaluation to verify the effectiveness

Methodology and Results

- StrategyCHT



(1) Attributed consistent hashing ring
- Add two attributes to nodes on the hashing ring
- Capacity: the overall data capacity of a node
- Bandwidth: the maximum bandwidth of a node
- Virtual node has the same attributes with the physical node. (e.g., A1 and A2)

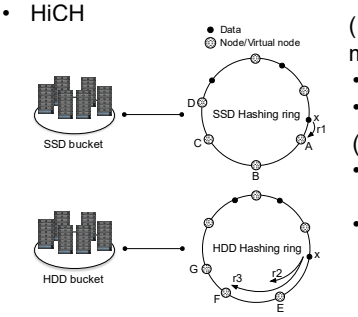
(2) Divide the nodes into sectors and use a selection strategy to select the node in each sector
- Uniform strategy: f(x) = (hash(x) + p) mod s
- Performance strategy: val <= f(x, seed, 0, u)

(a) calculation time and memory usage

Node number	Consistent hashing	CRUSH (straw buckets)	Strategy CHT
64	*0.0124*	*1.4591*	*0.0131*
512	*0.0204*	*9.1039*	*0.0218*
4096	*0.0332*	*70.2185*	*0.0361*
32768	*0.1783*	*582.4153*	*0.1948*

Node No. × Vnode No.	Consistent hashing (MB)	StrategyCHT (MB)
1000×256	*7.81*	*9.77*
10000×256	*78.13*	*97.65*
1000×1024	*31.25*	*39.06*
10000×4096	*1562.4*	*1953.0*

- HiCH

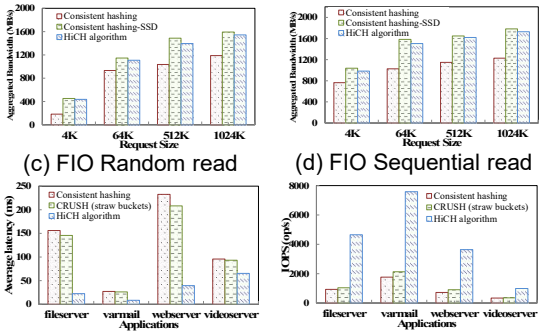


(a) model

(1) Divide HDD and SSD nodes into two buckets and maintains a consistent hashing ring for each bucket
- Construct a hierarchical consistent hashing
- SSD ring is considered the cache for HDD ring

(2) Data placement strategy
- When new data objects come, the first choice is to place the data on SSD ring
- Data movement between SSD and HDD rings

(b) Algorithm illustration



(c) FIO Random read

(d) FIO Sequential read

(e) Filebench average latency

(f) Filebench IOPS

Discussion and Future Work

- Introduce two consistent hashing-based algorithms specifically for data placement on heterogeneous storage systems

We plan to conduct further research on

- Replication strategy and reliability mechanism on heterogeneous storage systems
- Automatic data access pattern discovery for cloud computing applications, also to achieve I/O performance optimization with data prefetching and distribution
- Distributed object storage in heterogeneous cloud environment, such as virtual machine image storage
- High throughput I/O path for HPC in large-scale data centers
- Distributed cache and distributed memory for data-intensive computing

Acknowledgements

We are grateful to the Nimboxx and the Cloud and Autonomic Computing site at Texas Tech University for the valuable support for this project.

NSF CAC Semi-Annual Meeting, October 5-6, 2016
The Center for Cloud and Autonomic Computing is supported by the National Science Foundation under Grant No. 1362134.

TEXAS TECH UNIVERSITY