Storage Efficiency in Clustered Storage Environment

Ankur Saran
Tata Consultancy Services
Abstract

- Data Storage Efficiency mainly by de-duplicating and compressing data is very popular and indeed it is one of the unique selling points for storage vendors. Such feature has some inherent problem, which results in running them as background application on the storage systems, because of the amount of workload generated, due to the CPU’s intensive scanning operations, which eats up huge amount of CPU resources at run time. Despite of multiple de-dupe and compression techniques present, none can efficiently deal with handling data in production environment.

As data is growing with exponential rate, almost all storage vendors are coming up with scale-out/clustered solutions to address the big data problem for their production environment. In clustered environment however, there is scope in leftover CPU bandwidth which can be used to improve data Efficiency ratio without affecting other operation significantly. This works on idea of distributing load from already busy Node to the relatively free Nodes using few distribution algorithms, map reduce being one of them.
Topics to be covered

- Storage trend
- Storage Efficiency & Challenges
- Storage Efficiency in Cluster Environment
- Benefits & Challenges
- Conclusion
Storage Trend

- Storage need has been continuously growing
- Next Generation Storage needs
  - System to handle exponential growth in data from
    - Medical sciences i.e. data from DNA sequence etc
    - Finance data storage and analysis
    - Scientific data storage and analysis i.e. Handle big data handling
- Need of extremely scalable data storage system
- Clustered Storage is definitely one of the next generation Storage system.
Clustered Storage Structure

Fig1: One of the Clustered Storage Architecture
Structure of Clustered Storage

- Common Name space is shared as point of transaction
- All the Nodes in system have Storage Space and Computing ability.
- All the Node have tightly coupled storage with processing unit.
- All the Node are loosely coupled cluster.
Storage Efficiency & Challenges

Storage Efficiency
Storage Efficiency

- Storage Efficiency comprises of storage features e.g.
  - Deduplication
  - Compression
  - Snap shots
  - Thin Provisioning etc.

Storage and Storage Efficiency

- Storage market players compete on point of features they provided such as dedupe.
- Many processes in Storage Efficiency such as fingerprint searching of duplicate data present in storage are CPU intensive, hence these processes run as low priority processes.
- Design in which CPU intensive processes are distributed among the Node (with leftover CPU) can be an asset to capitalize.
Storage Efficiency in Cluster Environment

In
Clustered Environment
Structure of Clustered Storage Efficiency

Fig2: Setup with Global Fingerprint Database
Global Fingerprint Database (GFDB) contains unique Fingerprint of all data across Nodes.

All the Data storage Node can query and update GFDB to keep track of data in the system.

Duplicate Fingerprint in GFDB mean duplicate data block in the system.
Clustered Storage System Setup and GFDB

Fig3: Initial Storage setup without data
Important Components

- **Storage Master**: Controls the block flow and maintain information about the stored blocks, CPU utilization
- **Storage Slaves**: Storage units of system
- **Fingerprints**: Unique identification key for the data block
- **Fingerprint Database**: Database to store unique key with value as data block address
Clustered Storage System with Single Instance of Data

Fig4: System after first stream of data is stored in system and related changes in FPDB
System with duplicate Entries

Fig5: System after second stream of data is stored in system and related changes in FPDB
Processing of Duplicate data

- Mapping of fingerprint database having key as fingerprint and value as path id of the fingerprint

- Mapper( fingerprint , path)
  //For each fingerprint
  emit(fingerprint, 1)
Reducer Function

- Reducer Collects the fingerprints and reduces the duplicate count if any

- Reducer(fingerprint, iteration value)
  - if fingerprint == new_fingerprint
  - emit(result)
Cluster Level Deduplication

<table>
<thead>
<tr>
<th>FP</th>
<th>Slave Name</th>
<th>Block Id</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Slave1</td>
<td>1</td>
</tr>
<tr>
<td>B</td>
<td>Slave1</td>
<td>2</td>
</tr>
<tr>
<td>C</td>
<td>Slave2</td>
<td>1</td>
</tr>
<tr>
<td>D</td>
<td>Slave2</td>
<td>2</td>
</tr>
<tr>
<td>E</td>
<td>SlaveN,Slave1</td>
<td>1,3</td>
</tr>
<tr>
<td>F</td>
<td>SlaveN,Slave1</td>
<td>2,4</td>
</tr>
</tbody>
</table>

Fig6: Before and after Deduplication
### Fingerprint Database

<table>
<thead>
<tr>
<th>FP</th>
<th>Slave Name</th>
<th>Block Id</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Slave1</td>
<td>1</td>
</tr>
<tr>
<td>B</td>
<td>Slave1</td>
<td>2</td>
</tr>
<tr>
<td>C</td>
<td>Slave2</td>
<td>1</td>
</tr>
<tr>
<td>D</td>
<td>Slave2</td>
<td>2</td>
</tr>
<tr>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>E</td>
<td>SlaveN</td>
<td>1</td>
</tr>
<tr>
<td>F</td>
<td>SlaveN</td>
<td>2</td>
</tr>
</tbody>
</table>

**Before Deduplication**

<table>
<thead>
<tr>
<th>FP</th>
<th>Slave Name</th>
<th>Block Id</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Slave1,Slave2</td>
<td>1,3</td>
</tr>
<tr>
<td>B</td>
<td>Slave1</td>
<td>2</td>
</tr>
<tr>
<td>C</td>
<td>Slave2</td>
<td>1</td>
</tr>
<tr>
<td>D</td>
<td>Slave2</td>
<td>2</td>
</tr>
<tr>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>E</td>
<td>SlaveN,Slave1</td>
<td>1,3</td>
</tr>
<tr>
<td>F</td>
<td>SlaveN,Slave1</td>
<td>2,4</td>
</tr>
</tbody>
</table>

**After Deduplication**

**Fig7:** FPDB before and after dedupe
Benefits & Challenges

Benefits & Challenges With Clustered
PoC: Scenario & Results

Graph1: Data type vs. Dedupe saving in percentage across Cluster

Constants for the PoC
- No of Nodes: 2
- Block Size: 64m
- Replication Factor: 1
Results

- Storage Efficiency in cluster environment reduces the storage requirement by many folds.

- Searching the Global fingerprint database worked as a boon to design as it can be used to reduce the time required to process duplicate data in cluster.
Challenges

- Creation of Global Finger Print Database (GFDB)
- Removal of data blocks and updating it on GFDB and Storage System
- Master Node- Single point of failure.
Conclusion

- Multi Node storage is the storage of future
- Next Generation storage can be more efficient with utilization of distributed architecture
- Features like dedupe will be the key to success in the industry.
- Storage Efficiency will help in optimizing Storage requirement and reducing carbon footprints
Mail us @
Ankur.saran@tcs.com

Special Thanks to Arvind Kumar and Gaurav Gupta, Suraj Bhatnagar
Thanks to Nishi Gupta, Udayan Singh and my project members