Ceph – a scalable distributed storage system

Nico Wang, Zuriel Avilez, Christopher Poon with Mentor Laura Flores

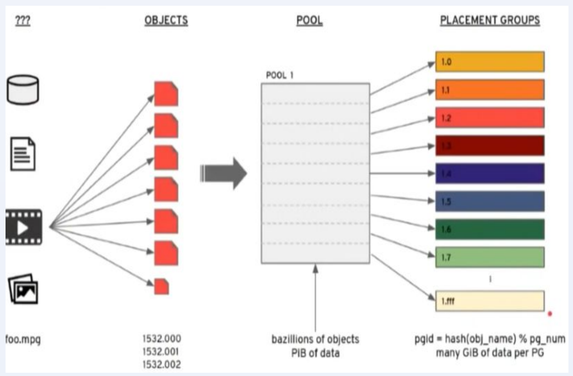
IBM and Rensselaer Polytechnic Institute

Introduction

Ceph is an open source distributed storage system that is reliable, scalable, and unified. It is built on an underlying software system Reliable Autonomic Distributed Object Store (RADOS) which manages the distribution of data within the Ceph cluster.

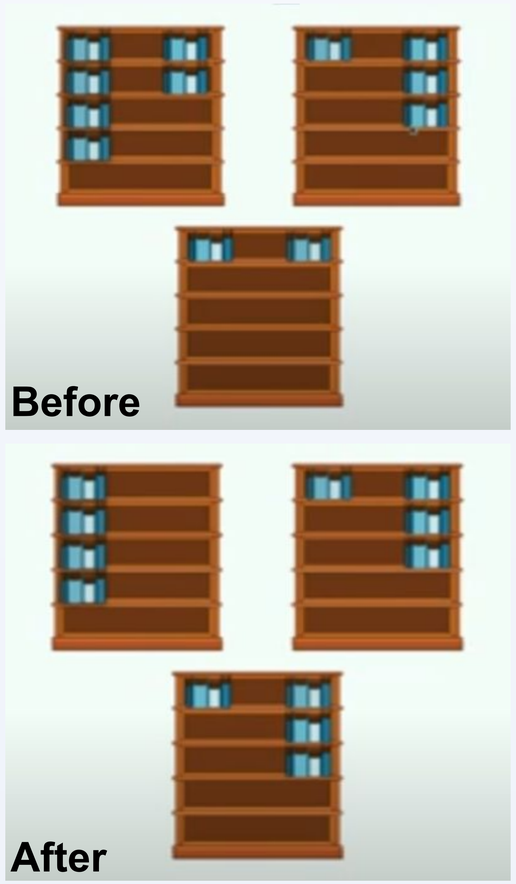
RADOS is comprised of storage daemons. The monitor (ceph-mon) is the central authority and coordination point for other cluster components. The manager (ceph-mgr) aggregates real-time metrics such as the cluster health or performance. Object storage daemons (OSDs) store the data on an HDD or SSD and replicates/rebalances data.

Our objective focused mainly on the storage aspect of Ceph. The general hierarchy consists of a file being broken up into RADOS objects and put into a single RADOS pool. Then it is broken down further into Placement Groups (PGs) and stored into OSDs for redundancy.



Balancer Module

The Balancer Module is a component of the Ceph storage system that balances the distribution of data (PGs) across OSDs. By continuously monitoring the storage capacity of the OSDs, the balancer can prevent imbalances that lead to performance bottlenecks or overloaded HDDs and SSDs.



The specific component we were tasked with was the ‘ceph balancer status’ command that can be called via the Command Line Interface. This command provides real-time information about the current state of the balancer, such as its last optimization, duration, and method of automatic balancing.

```
"active": true,
"last_optimize_duration":
"0:00:00.003861",
"last_optimize_started": "Fri Nov 10
21:57:33 2023",
"mode": "upmap",
"no_optimization_needed": true,
"optimize_result": "Unable to find
further optimization, or pool(s) pg_num
is decreasing, or distribution is
already perfect",
"plans": []
```

Objective

Our objective is to improve the 'ceph balancer status' command to be more informative of the changes it made when it last optimized a Ceph cluster. This will be used to help customers understand what changed in their Ceph cluster as a result of the balancer, allowing for better communication between Ceph as a storage system and its users.

Setting Up

Setting up Ceph onto our local environment proved to be a difficult challenge, given our operating systems. The various problems each of us ran into are documented in our cloned Ceph repo. In the end, our project mentor Laura advised us to request for VPN access via a Sepia Lab Access Request ticket. Following the Wiki instructions, we installed OpenVPN and generated our client credentials. We provided Laura the necessary information required for the ticket and she filed the ticket for us.

After being granted access and fixing a few more issues, we were able to successfully remotely access a setup with Ceph installed. Now finally able to build a test cluster, we went to work familiarizing ourselves with the cluster layout and understanding the balancer personally, not just through code.

Methods

To improve the ‘ceph balancer status’ command, we decided to implement another option to the command that prints out more detailed information on what exactly the balancer had done on its previous iteration. More specifically, we would add another key to display which PGs moved from which OSDs. We listed other ideas of more information to include on our Wiki, though most are already implemented in this command or other balancer commands.

Results (Progress)

So far, we were able to create the ‘ceph balancer status detailed’ command. It contains the same information as the regular status command does, with the addition of the ‘pg_upmap_items’ key that outputs a list of PGs and the OSDs that it was moved from and to.

```
“pg_upmap_items”: {
  “2.a”: [
    1,
    2
  ]
},
“plans”: []
```

Above is our latest attempt (simplified) in formatting the output of the key.

Conclusion

In conclusion, we learned about the Ceph storage system and how to successfully setup a test cluster. We also implemented an extension of an existing command to provide more detail to users about the balancer module’s last operation. It was our first step into a project with a complexity and scale of this level. Being able to observe its functions and modify its code alongside a mentor is a huge opportunity and learning experience.

References

Intro to Ceph:
https://www.youtube.com/watch?v=PmLPbrf-x9g
Balancer Module
https://docs.ceph.com/en/latest/rados/operations/balancer/
Cloned Github Repo:
https://github.com/WangWNico/ceph

Acknowledgements

Neha Ojha, Anthony Lewitt

RESEARCH POSTER PRESENTATION TEMPLATE © 2015
www.PosterPresentations.com