



Ceph RGW Cache Prefetching for Batch Jobs

Xun Lin
Yang Qiao
Tianyi Tang
Gang Wei
Zhangyu Wan



Vision and Goals

When developer using Spark to deal with batch jobs, the jobs are done in sequence which means one job cannot start until all of its dependencies are done. Goal of this project is to establish a mechanism to extract the dependencies and prefetch the data from Ceph RGW beforehand so that the overall runtime can be speed up.

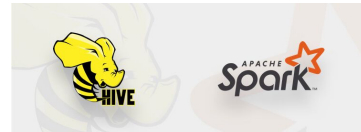


User Story

For Spark developers:

Accelerate the computation speed of batch jobs by prefetching data from file systems to Ceph RGW high speed cache instead of directly fetching the data from low speed file systems.

System components



- **Spark**

- high-performance open source data processing engine that can perform batch processing.
- **DAG:** a set of Vertices and Edges which represents tasks dependencies.

- **Hive**

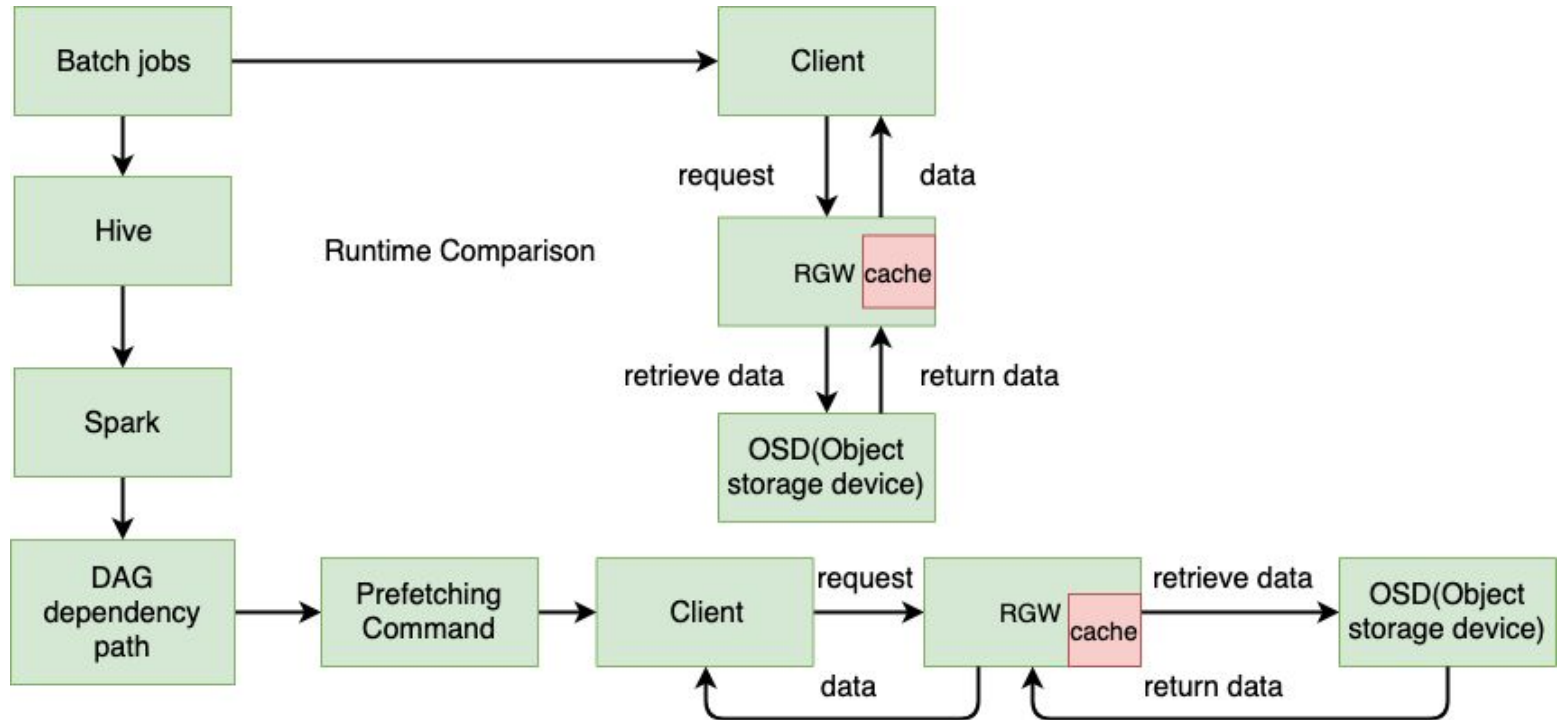
- data warehouse framework for querying and analysis of data that is stored in HDFS.

- **Ceph**

- an open source storage platform



System Diagram





Features

This project provides an efficient mechanism to accelerate Spark application running time by prefetching batch jobs data into cache. Below is an overview of the project features:

- Create the DAG of operations and data from user's Spark applications.
- Prefetch the data from OSDs into cache based on the DAG.



DAG

Spark creates a DAG in order to schedule the operation

Ways to retrieve DAG without actually performing those operations

- If using dataframes (spark sql), we can use `df.explain(true)` to get the plan and all operations.
- If using rdd, we can use `rdd.toDebugString` to get a string representation and `rdd.dependencies` to get the tree itself.



Minimum Valuable Product

Demonstrating speed improvements with our prefetching mechanism using common benchmarks. (e.g. TPC-DS/TPC-H) as well as other common jobs.



Sprint 1

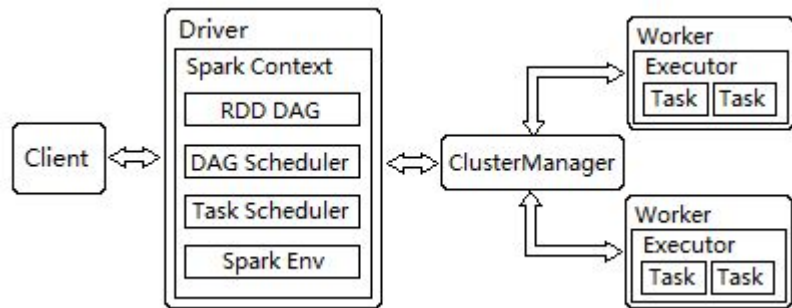
- Setting up infrastructures
- Understanding Ceph Prefetching
- Running spark applications
- Extra: Building Docker containers

Spark Architecture

The Spark architecture uses the Master-Slave model in distributed computing. The node running the Master process in the cluster is called the Master. Similarly, the node containing the Worker process in the cluster is the Slave.

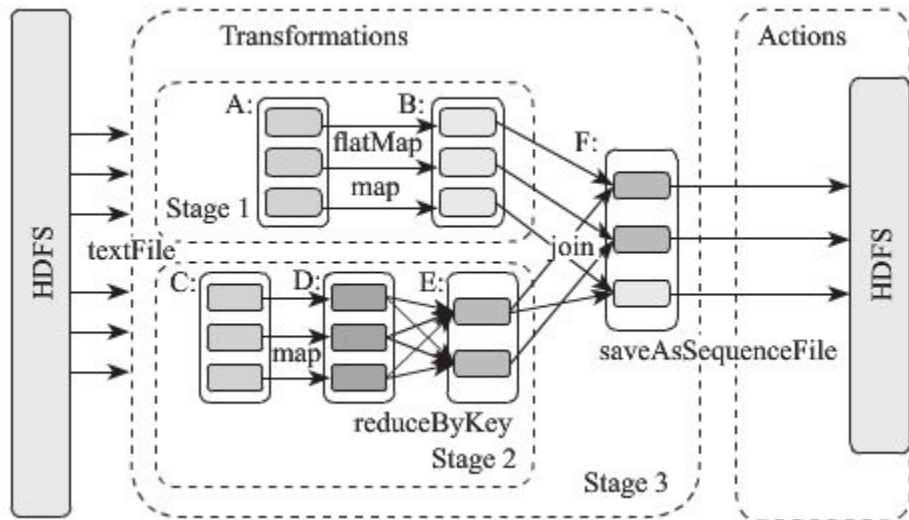
The Master is responsible for controlling the operation of the entire cluster; the Worker node is equivalent to the computing node in the distributed system, which receives the Master node commands and reports the calculation process to the Master;

The Executor is responsible for the execution of the task; the Client is the client that submits the application to the user; the Driver is responsible for submitting Coordination of post distributed applications



Spark Process

After the Action operator is triggered, all operators form a DAG. Spark will form a Stage based on the different dependencies between RDDs using the DAG. Each Stage contains a series of function execution pipelines. In the figure, A, B, C, D, E, and F are different RDDs, and the boxes in the RDD are partitions of the RDD.



Spark-Master & Worker

```
docker -- docker run --rm -it --name spark-master --hostname spark-master -p 7077:7077 -p 8080:8080 --network spark_network tty/spark:latest /bin/sh -- 120x40
Tianyis-MacBook-Pro:docker tty$ docker run --rm -it --name spark-master --hostname spark-master -p 7077:7077 -p 8080:8080 --network spark_network tty/spark:latest /bin/sh
/ # /spark/bin/spark-class org.apache.spark.deploy.master.Master --ip 'hostname' --port 7077 --webui-port 8080
Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties
19/09/26 00:34:03 INFO Master: Started daemon with process name: 7@spark-master
19/09/26 00:34:03 INFO SignalUtils: Registered signal handler for TERM
19/09/26 00:34:03 INFO SignalUtils: Registered signal handler for HUP
19/09/26 00:34:03 INFO SignalUtils: Registered signal handler for INT
19/09/26 00:34:03 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
19/09/26 00:34:04 INFO SecurityManager: Changing view acls to: root
19/09/26 00:34:04 INFO SecurityManager: Changing modify acls to: root
19/09/26 00:34:04 INFO SecurityManager: Changing view acls groups to:
19/09/26 00:34:04 INFO SecurityManager: Changing modify acls groups to:
19/09/26 00:34:04 INFO SecurityManager: authentication disabled; ui acls disabled; users with view permissions: Set(root); groups with view permissions: Set(); users with modify permissions: Set(root); groups with modify permissions: Set()
19/09/26 00:34:04 INFO Utils: Successfully started service 'sparkMaster' on port 7077.
19/09/26 00:34:04 INFO Master: Starting Spark master at spark://spark-master:7077
19/09/26 00:34:04 INFO Master: Running Spark version 2.4.4
19/09/26 00:34:05 INFO Utils: Successfully started service 'MasterUI' on port 8080.
19/09/26 00:34:05 INFO MasterWebUI: Bound MasterWebUI to 0.0.0.0, and started at http://spark-master:8080
19/09/26 00:34:05 INFO Master: I have been elected leader! New state: ALIVE
19/09/26 00:34:14 INFO Master: Registering worker 172.18.0.3:45483 with 6 cores, 1024.0 MB RAM
19/09/26 00:34:29 INFO Master: Registering worker 172.18.0.4:40249 with 6 cores, 1024.0 MB RAM
```

```
docker -- docker run --rm -it --name spark-worker2 --hostname spark-worker2 --network spark_network tty/spark:latest /bin/sh -- 120x40
Tianyis-MacBook-Pro:docker tty$ docker run --rm -it --name spark-worker2 --hostname spark-worker2 --network spark_network tty/spark:latest /bin/sh
/ # /spark/bin/spark-class org.apache.spark.deploy.worker.Worker --webui-port 8080 spark://spark-master:7077
Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties
19/09/26 00:34:27 INFO Worker: Started daemon with process name: 6@spark-worker2
19/09/26 00:34:27 INFO SignalUtils: Registered signal handler for TERM
19/09/26 00:34:27 INFO SignalUtils: Registered signal handler for HUP
19/09/26 00:34:27 INFO SignalUtils: Registered signal handler for INT
19/09/26 00:34:28 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
19/09/26 00:34:28 INFO SecurityManager: Changing view acls to: root
19/09/26 00:34:28 INFO SecurityManager: Changing modify acls to: root
19/09/26 00:34:28 INFO SecurityManager: Changing view acls groups to:
19/09/26 00:34:28 INFO SecurityManager: Changing modify acls groups to:
19/09/26 00:34:28 INFO SecurityManager: authentication disabled; ui acls disabled; users with view permissions: Set(root); groups with view permissions: Set(); users with modify permissions: Set(root); groups with modify permissions: Set()
19/09/26 00:34:28 INFO Utils: Successfully started service 'sparkWorker' on port 40249.
19/09/26 00:34:29 INFO Worker: Starting Spark worker 172.18.0.4:40249 with 6 cores, 1024.0 MB RAM
19/09/26 00:34:29 INFO Worker: Running Spark version 2.4.4
19/09/26 00:34:29 INFO Worker: Spark home: /spark
19/09/26 00:34:29 INFO Utils: Successfully started service 'WorkerUI' on port 8080.
19/09/26 00:34:29 INFO WorkerWebUI: Bound WorkerWebUI to 0.0.0.0, and started at http://spark-worker2:8080
19/09/26 00:34:29 INFO Worker: Connecting to master spark-master:7077...
19/09/26 00:34:29 INFO TransportClientFactory: Successfully created connection to spark-master/172.18.0.2:7077 after 59 ms (0 ms spent in bootstraps)
19/09/26 00:34:29 INFO Worker: Successfully registered with master spark://spark-master:7077
19/09/26 00:34:29 INFO TransportClientFactory: Successfully created connection to spark-master/172.18.0.2:7077 after 57 ms
19/09/26 00:34:29 INFO Worker: Successfully registered with master spark://spark-master:7077

docker -- docker run --rm -it --name spark-worker1 --hostname spark-worker1 --network spark_network tty/spark:latest /bin/sh -- 120x40
Tianyis-MacBook-Pro:docker tty$ docker run --rm -it --name spark-worker1 --hostname spark-worker1 --network spark_network tty/spark:latest /bin/sh
/ # /spark/bin/spark-class org.apache.spark.deploy.worker.Worker --webui-port 8080 spark://spark-master:7077
Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties
19/09/26 00:34:12 INFO Worker: Started daemon with process name: 7@spark-worker1
19/09/26 00:34:12 INFO SignalUtils: Registered signal handler for TERM
19/09/26 00:34:12 INFO SignalUtils: Registered signal handler for HUP
19/09/26 00:34:12 INFO SignalUtils: Registered signal handler for INT
19/09/26 00:34:12 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
19/09/26 00:34:13 INFO SecurityManager: Changing view acls to: root
19/09/26 00:34:13 INFO SecurityManager: Changing modify acls to: root
19/09/26 00:34:13 INFO SecurityManager: Changing view acls groups to:
19/09/26 00:34:13 INFO SecurityManager: Changing modify acls groups to:
19/09/26 00:34:13 INFO SecurityManager: authentication disabled; ui acls disabled; users with view permissions: Set(root); groups with view permissions: Set(); users with modify permissions: Set(root); groups with modify permissions: Set()
19/09/26 00:34:13 INFO Utils: Successfully started service 'sparkWorker' on port 45483.
19/09/26 00:34:14 INFO Worker: Starting Spark worker 172.18.0.3:45483 with 6 cores, 1024.0 MB RAM
19/09/26 00:34:14 INFO Worker: Running Spark version 2.4.4
19/09/26 00:34:14 INFO Worker: Spark home: /spark
19/09/26 00:34:14 INFO Worker: I have been elected leader! New state: ALIVE
19/09/26 00:34:14 INFO WorkerWebUI: Bound WorkerWebUI to 0.0.0.0, and started at http://spark-worker1:8080
19/09/26 00:34:14 INFO Worker: Connecting to master spark-master:7077...
19/09/26 00:34:14 INFO TransportClientFactory: Successfully created connection to spark-master/172.18.0.2:7077 after 57 ms (0 ms spent in bootstraps)
19/09/26 00:34:14 INFO Worker: Successfully registered with master spark://spark-master:7077
```

Spark Cluster Visualization



Spark Master at spark://spark-master:7077

URL: spark://spark-master:7077

Alive Workers: 2

Cores in use: 12 Total, 0 Used

Memory in use: 2.0 GB Total, 0.0 B Used

Applications: 0 Running, 0 Completed

Drivers: 0 Running, 0 Completed

Status: ALIVE

Workers (2)

Worker Id	Address	State	Cores	Memory
worker-20190926003413-172.18.0.3-45483	172.18.0.3:45483	ALIVE	6 (0 Used)	1024.0 MB (0.0 B Used)
worker-20190926003428-172.18.0.4-40249	172.18.0.4:40249	ALIVE	6 (0 Used)	1024.0 MB (0.0 B Used)

Running Applications (0)

Application ID	Name	Cores	Memory per Executor	Submitted Time	User	State	Duration
----------------	------	-------	---------------------	----------------	------	-------	----------

Completed Applications (0)

Application ID	Name	Cores	Memory per Executor ▲	Submitted Time	User	State	Duration
----------------	------	-------	-----------------------	----------------	------	-------	----------

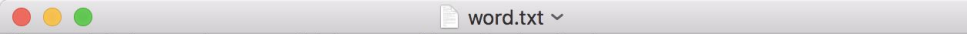


Virtualization: Docker Container

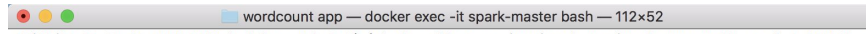
Why are we using Docker instead of running spark locally?

- Containers: other than virtual machine
- Deployment: manage using Kubernetes, Rancher, etc
- Configuration: write dockerfiles

Demo: word count



Thomas A Anderson is a man living two lives By day he is an average computer programmer and by night a hacker known as Neo Neo has always questioned his reality but the truth is far beyond his imagination Neo finds himself targeted by the police when he is contacted by Morpheus a legendary computer hacker branded a terrorist by the government Morpheus awakens Neo to the real world a ravaged wasteland where most of humanity have been captured by a race of machines that live off of the humans body heat and electrochemical energy and who imprison their minds within an artificial reality known as the Matrix As a rebel against the machines Neo must return to the Matrix and confront the agents super powerful computer programs devoted to snuffing out Neo and the entire human rebellion



```
19/09/26 01:35:34 INFO DAGScheduler: Job 0 finished: collect at /app/wordcount/wordcount.py:22, took 1.345082 s
19/09/26 01:35:34 INFO TaskSchedulerImpl: Removed TaskSet 1.0, whose tasks have all completed, from pool
the: 10
a: 7
Neo: 6
by: 5
and: 5
is: 4
to: 3
computer: 3
of: 3
reality: 2
his: 2
machines: 2
hacker: 2
Matrix: 2
known: 2
he: 2
Morpheus: 2
an: 2
as: 2
contacted: 1
human: 1
captured: 1
devoted: 1
awakens: 1
police: 1
day: 1
must: 1
questioned: 1
has: 1
By: 1
real: 1
return: 1
government: 1
far: 1
snuffing: 1
legendary: 1
world: 1
humanity: 1
entire: 1
Thomas: 1
confront: 1
race: 1
truth: 1
night: 1
where: 1
energy: 1
humans: 1
Anderson: 1
programmer: 1
beyond: 1
```



Q & A



Thanks