Autonomous workload rebalancing in Kafka

Indrajeet Kumar
Site Reliability Engineer - LinkedIn
Agenda

- Workload distribution problem
- Manual - Built-in tools
- Semi-automated - Kafka-assigner
- Autonomous - Cruise Control
Workload distribution problem

Important for Distributed Systems
Harder to work around with Stateful systems
Kafka Overview

App 1

Broker X
- P1
- R P3

Broker Y
- P2
- R P1

Broker Z
- P3
- R P2

Topic A
- P1
- P2
- P3
Workload in Kafka

Leader Partitions

Total Partitions

Partition Sizes
Workload in Kafka

**Leader Partitions**

**Total Partitions**

**Partition Sizes**
Workload in Kafka

Leader Partitions

Total Partitions

Partition Sizes
Workload in Kafka

Leader Partitions

Total Partitions

Partition Sizes
Workload distribution problem - Some causes

Major factors which affect workload balance are:

- Bad partition distribution
- Hard host failures
- Soft host failures
- Traffic patterns
Kafka workload distribution - Solution

Rebalance the partitions!

Disk usage
Network usage
Number of partitions
Partition leadership count
Usual operations in Kafka

Preferred Leader Election
Partition rebalance
Bump Partition counts
Add/Remove brokers
Kafka at LinkedIn
Kafka at LinkedIn

4.5 Trillion messages a day

2500+ kafka brokers

1 PB In

3.9 PB Out
Kafka admin utilities

Out of the box tools:
bin/kafka-reassign-partitions.sh
bin/kafka-preferred-replica-election.sh
Example run of built-in tools

Rebalancing Partitions:

```
$ cat topics-to-move.json
{"topics":
  [{"topic": "foo1"},{"topic": "foo2"}],
  "version":1
}

$ ./bin/kafka-reassign-partitions.sh --topics-to-move-json-file topics-to-move.json --broker-list "5,6,7" --generate

$ cat partitions-to-move.json
{"partitions":
  [{"topic": "foo",
    "partition": 1,
    "replicas": [1,2,4] }],
  "version":1
}

$ ./bin/kafka-reassign-partitions.sh --reassignment-json-file partitions-to-move.json --execute
```
Problems with stock tools

Manual
Less optimal
Slow
Kafka Assigner
Kafka assigner

High level administrative commands

Under the hood, it uses the ‘kafka-utils/bin/’ scripts

It also allows to do complex rebalances with multiple goals
<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>reorder</td>
<td>Reelect partition leaders using replica reordering</td>
</tr>
<tr>
<td>balance</td>
<td>Rebalance partitions across the cluster</td>
</tr>
<tr>
<td>elect</td>
<td>Reelect partition leaders using preferred replica election</td>
</tr>
<tr>
<td>trim</td>
<td>Remove partitions from some brokers (reducing RF)</td>
</tr>
<tr>
<td>remove</td>
<td>Move partitions from one broker to one or more other brokers (maintaining RF)</td>
</tr>
<tr>
<td>set-replication-factor</td>
<td>Increase the replication factor of the specified topics</td>
</tr>
<tr>
<td>clone</td>
<td>Copy partitions from some brokers to a new broker (increasing RF)</td>
</tr>
</tbody>
</table>
Preferred Leader election
Case of URPs
Kafka assigner

Pros:

- High level admin commands
- Simple to use
- Allows chaining rebalance goals
- Easy to remove all partitions from a broker
Kafka assigner

Cons:
Where did you run it?
In-optimal balances in certain cases
Needs manual invocation and supervision
Cruise Control
Cruise Control

Central System
Complete live health of the cluster
Manual/Automatic management of workload
<table>
<thead>
<tr>
<th>Cruise Control State</th>
<th>Kafka Cluster State</th>
<th>Kafka Cluster Load</th>
<th>Cruise Control Proposals</th>
<th>Administration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Monitor</td>
<td>Analyzer</td>
<td>Executor</td>
<td>Training</td>
<td></td>
</tr>
<tr>
<td>RUNNING</td>
<td>PROPOSALS_READY</td>
<td>NO_TASK_IN_PROGRESS</td>
<td>TRAINING (0.00 %)</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Total Kafka Partitions</th>
<th>Valid Kafka Partitions</th>
<th>Flawed Kafka Partitions</th>
<th>Snapshots</th>
</tr>
</thead>
<tbody>
<tr>
<td>123</td>
<td>123</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

Ready Goals

- NetworkInboundUsageDistributionGoal
- CpuUsageDistributionGoal
- PotentialNwOutGoal
- ReplicaDistributionGoal
- DiskCapacityGoal
- NetworkInboundCapacityGoal
- LeaderBytesInDistributionGoal
- RackAwareDistributionGoal
- TopicReplicaDistributionGoal
**Alert:** Any actions that you do in this section will have consequences on your Kafka Cluster. Please think twice before executing these actions.

1. Add broker to Kafka cluster
2. Remove broker from Kafka cluster
3. Demote broker from Kafka cluster
4. Rebalance Kafka cluster
5. Stop Execution

### Broker Administration

<table>
<thead>
<tr>
<th>Replicas</th>
<th>Host</th>
<th>Broker</th>
<th>Status</th>
<th>Disk</th>
<th>CPU</th>
<th>Leader In</th>
<th>Follower In</th>
<th>Out</th>
<th>Potential Out</th>
</tr>
</thead>
<tbody>
<tr>
<td>115</td>
<td>nareshv-mn1</td>
<td>1</td>
<td>ALIVE</td>
<td>24 KB</td>
<td>0 %</td>
<td>48 Bps</td>
<td>26 Bps</td>
<td>97 Bps</td>
<td>150 Bps</td>
</tr>
<tr>
<td>104</td>
<td>nareshv-mn1</td>
<td>2</td>
<td>ALIVE</td>
<td>65 KB</td>
<td>0 %</td>
<td>107 Bps</td>
<td>34 Bps</td>
<td>149 Bps</td>
<td>218 Bps</td>
</tr>
<tr>
<td>18</td>
<td>nareshv-mn1</td>
<td>3</td>
<td>ALIVE</td>
<td>10 KB</td>
<td>0 %</td>
<td>9 Bps</td>
<td>24 Bps</td>
<td>18 Bps</td>
<td>66 Bps</td>
</tr>
</tbody>
</table>

**Flags:**
- Dryrun
- Kafka Assigner Mode

**Remove Broker(s) URL:**
http://localhost:8080/kafka/cruisecontrol/remove_broker?kafka_assigner=true&dryrun=true&brokerid=3&json=true

**Demote Broker(s) URL:**
http://localhost:8080/kafka/cruisecontrol/demote_broker?kafka_assigner=true&dryrun=true&brokerid=3&json=true

[Remove 1 Broker] [Demote 1 Broker]
Balancing Performance:
- Racks: 10
- Brokers: 40
- Entities: 50K
- Topics: 3K
- Replication Factor: 3
- Entity distribution: Exponential
- Balance percentage (for all resources): 1.05

Cruise Control

Pre-Balance:
- Graph showing workload distribution across different broker IDs.

Post-Balance:
- Graph showing workload distribution across different broker IDs.

Pre-Balance Broker Load Distribution:
- Box plots for CPU, Disk, Inbound, and Outbound workload.

Post-Balance Broker Distribution:
- Box plots for CPU, Disk, Inbound, and Outbound workload.
CC setup requirements

Kafka > 0.11.0.0
Drop in jar
Features already built-in

- Resource utilization tracking
- Multi-goal rebalance
- Anomaly detection
- Admin operations
How is CC doing?

Save SRE’s time to debug/fix kafka workload issues
Very fast operations
Central place to look at for globally distributed teams
Self-heal !!
Resources

Kafka shipped admin-tools:
https://github.com/apache/kafka/tree/trunk/bin

Kafka Assigner:
https://github.com/linkedin/kafka-tools/wiki/Kafka-Assigner

Cruise Control:
https://github.com/linkedin/cruise-control

Connect with me: https://www.linkedin.com/in/indrajeetkm/
Questions