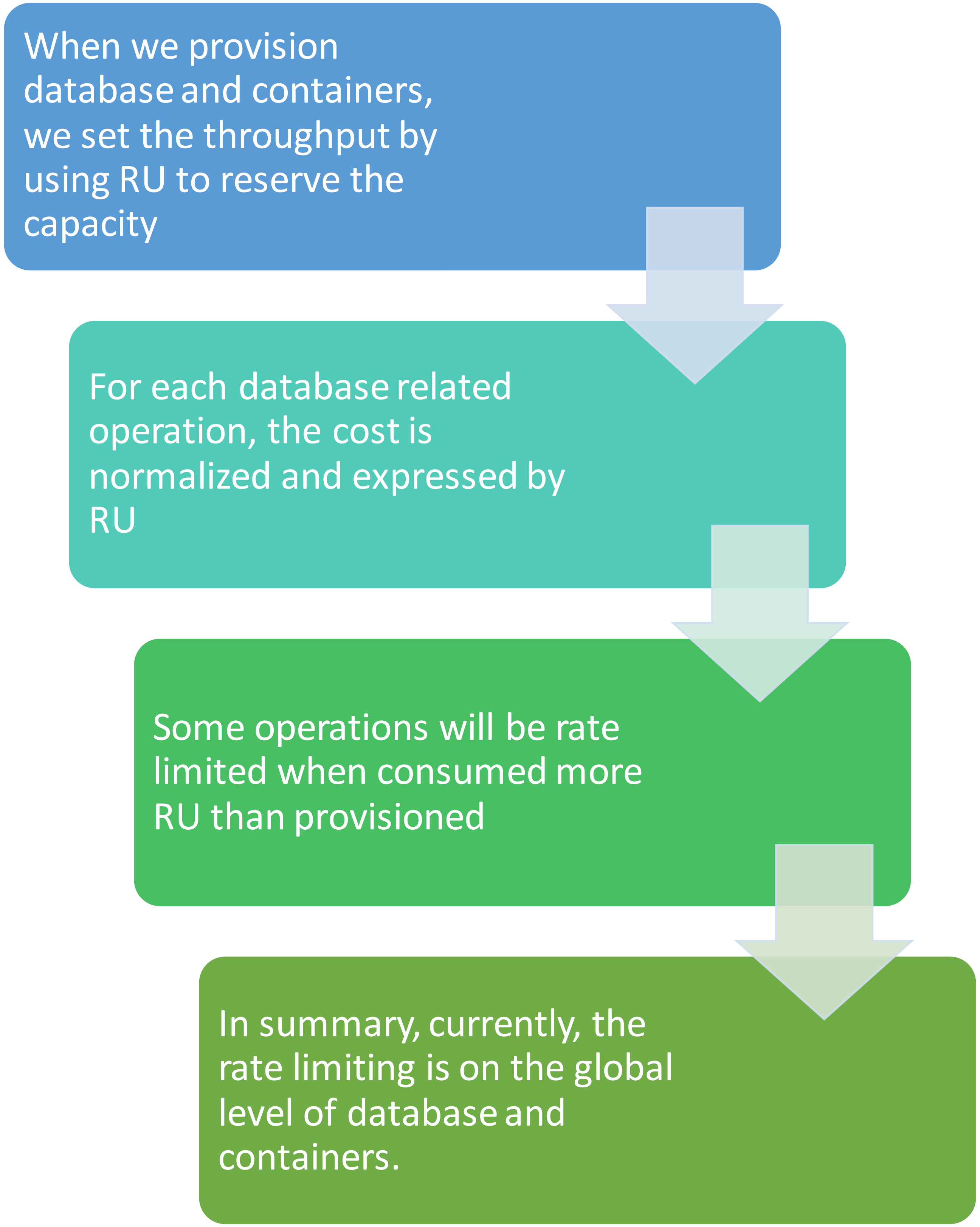
THROUGHPUT CONTROL

CLIENT RU LIMITING

How rate limiting works today



When we provision database and containers, we set the throughput by using RU to reserve the capacity

For each database related operation, the cost is normalized and expressed by RU

Some operations will be rate limited when consumed more RU than provisioned

In summary, currently, the rate limiting is on the global level of database and containers.

Advanced Scenarios

Different operation/task has different priority

- Customer would want to prevent normal transactions being throttled due to data ingestion/copy
  - Bulk upload
  - ADF data copy
- Some operation/task is not sensitive to latency and is more tolerant to be throttled compared to others.

Provide fairness/isolation to different end user/tenant

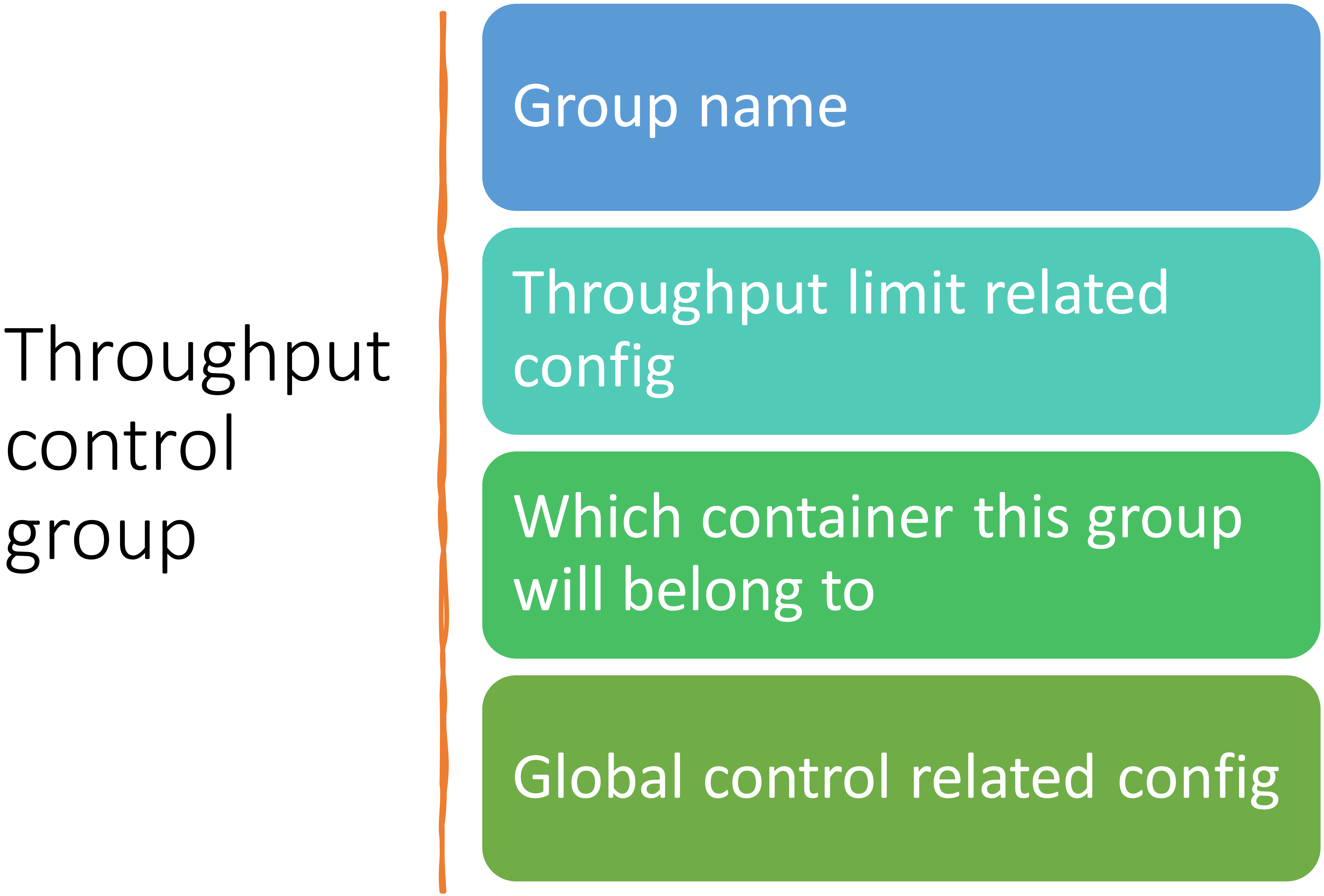
- An application usually will have many end users, due to intentionally or unintentionally, an end user may send too many requests which consume all the throughput which causing others to get throttled.
- LinkedIn premium/normal end user-based rate limiting

Throughput load balancing between different cosmos clients

For some cases, it is important to make sure all the clients get fair share of the throughput

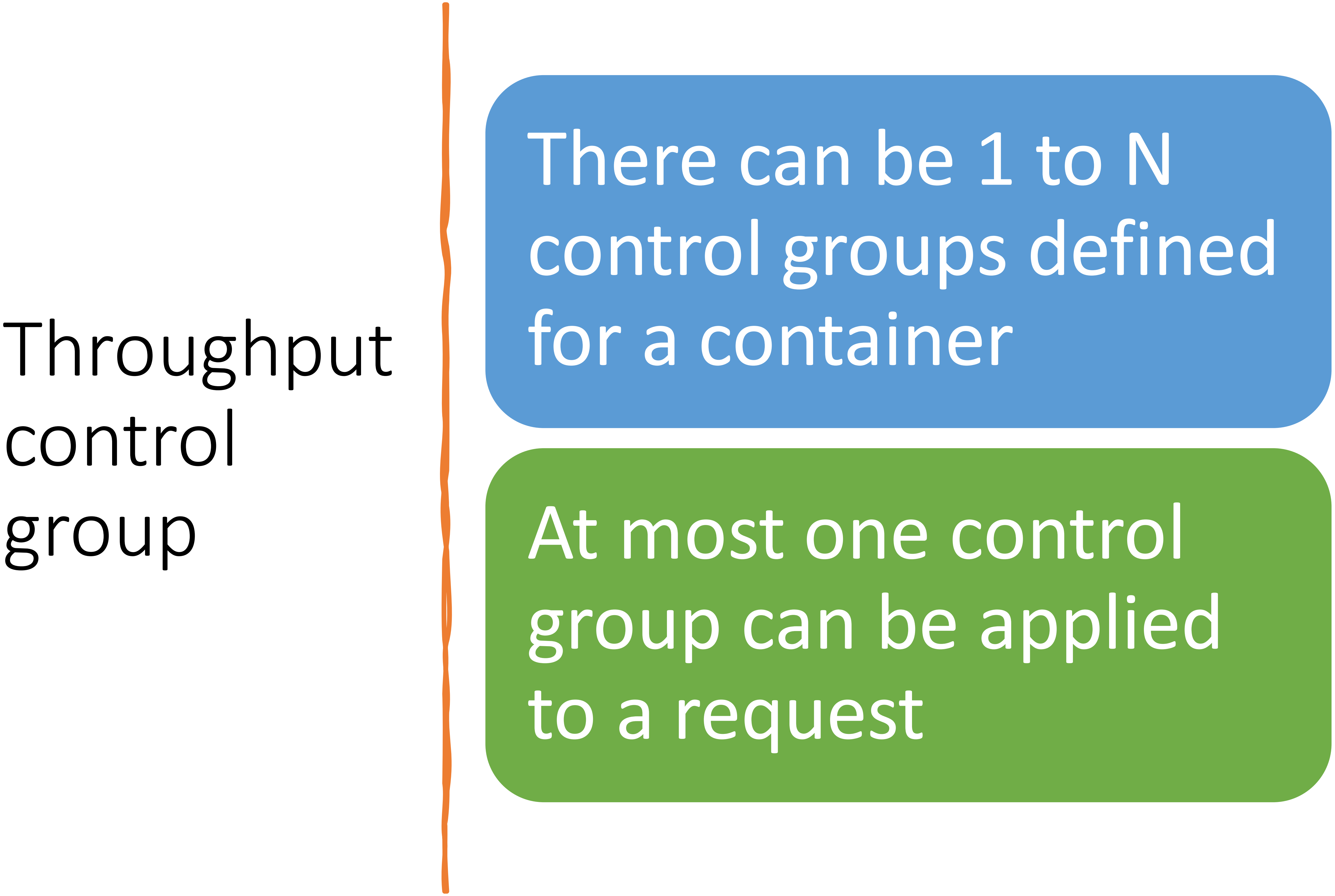
- Spark jobs

Throughput control enables capability for more granular level RU rate limiting as needed. For example, scenario-based rate limiting



Throughput control group

- Group name
- Throughput limit related config
- Which container this group will belong to
- Global control related config

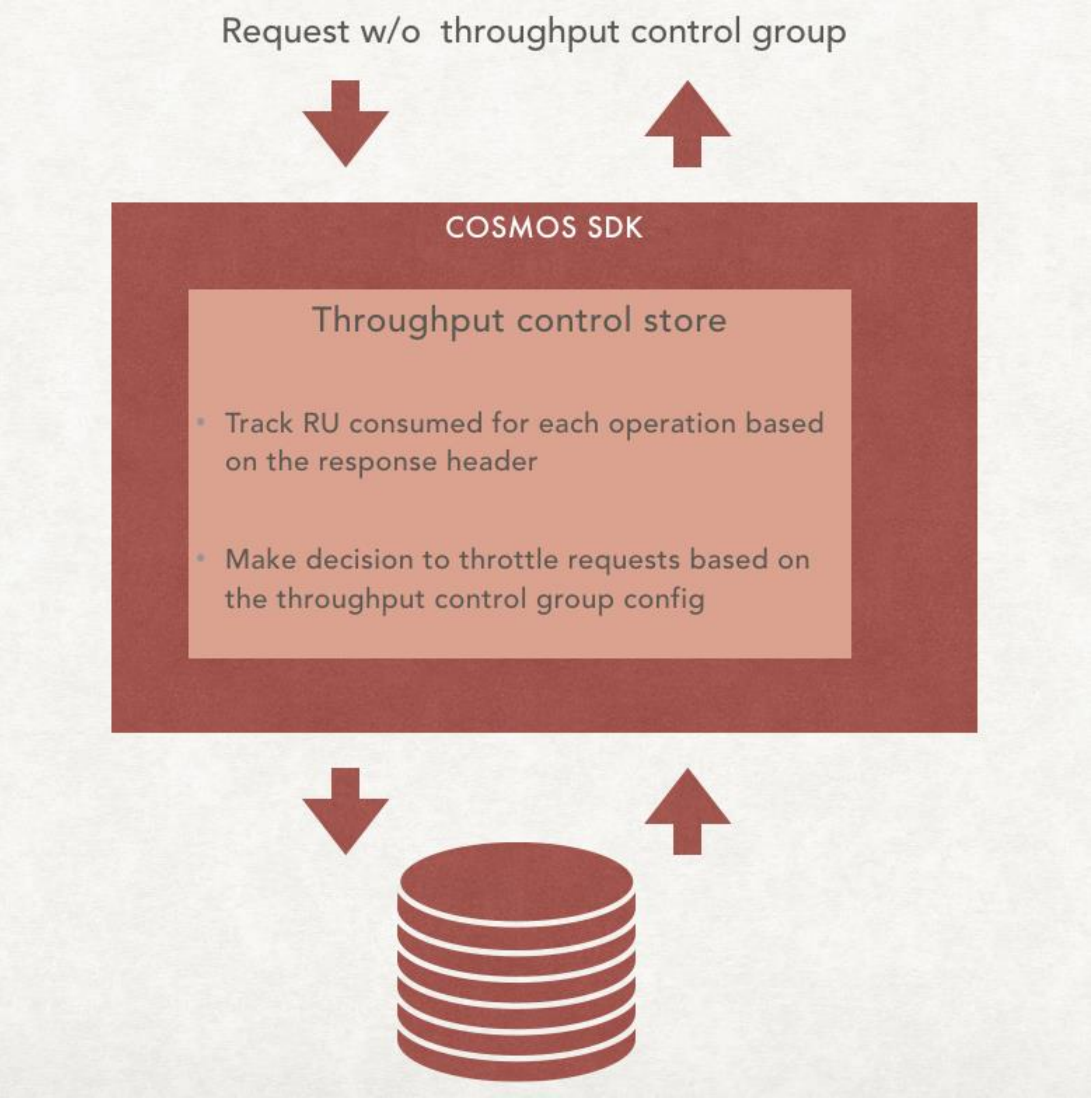


Throughput control group

There can be 1 to N control groups defined for a container

At most one control group can be applied to a request

High level overview



Request w/o throughput control group

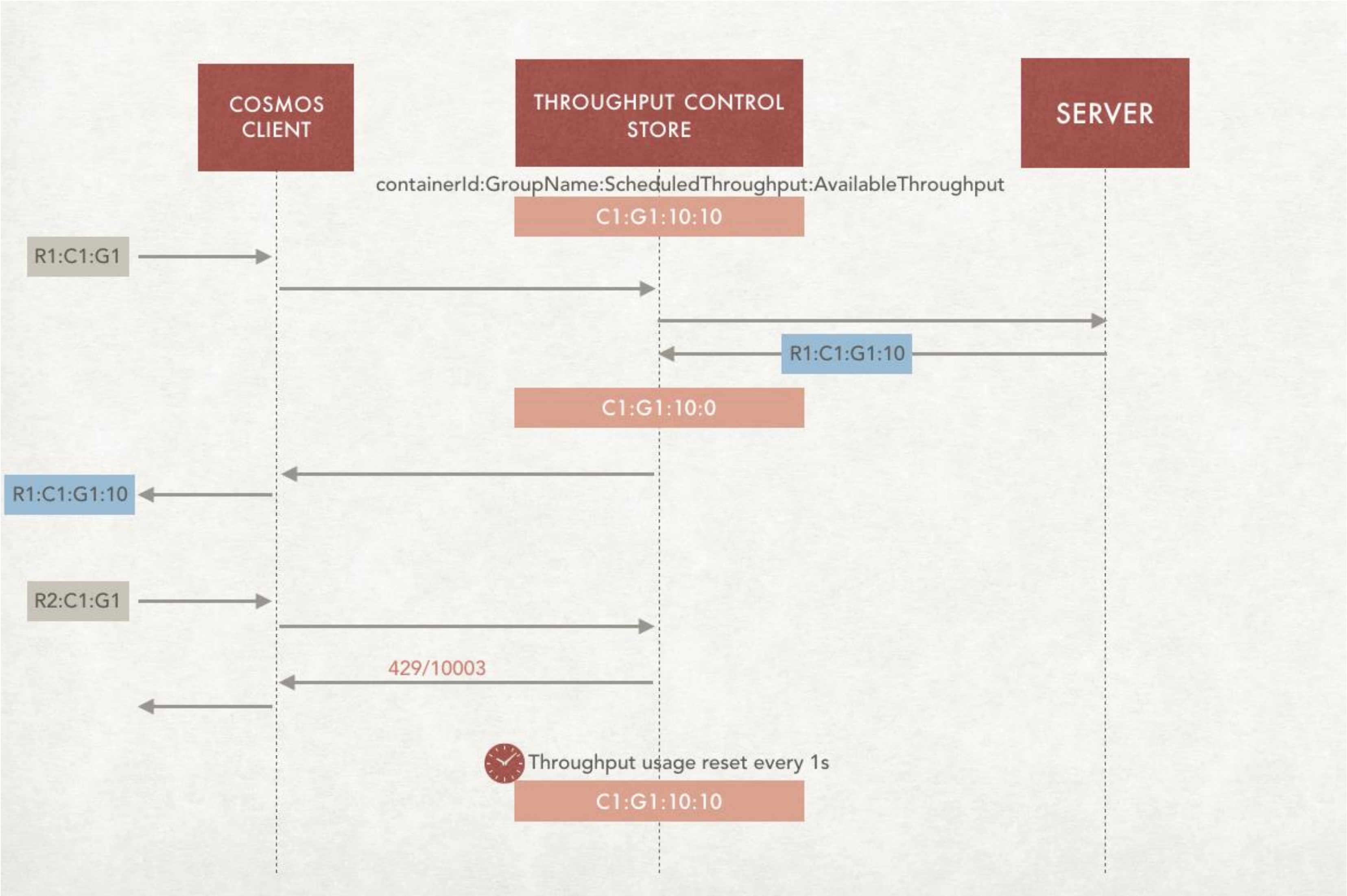
COSMOS SDK

Throughput control store

- Track RU consumed for each operation based on the response header
- Make decision to throttle requests based on the throughput control group config

Throughput control store

Scenario: Enable throughput control group G1 on Container C1 with target throughput 10 RU



Throughput Control Mode

Local Control:

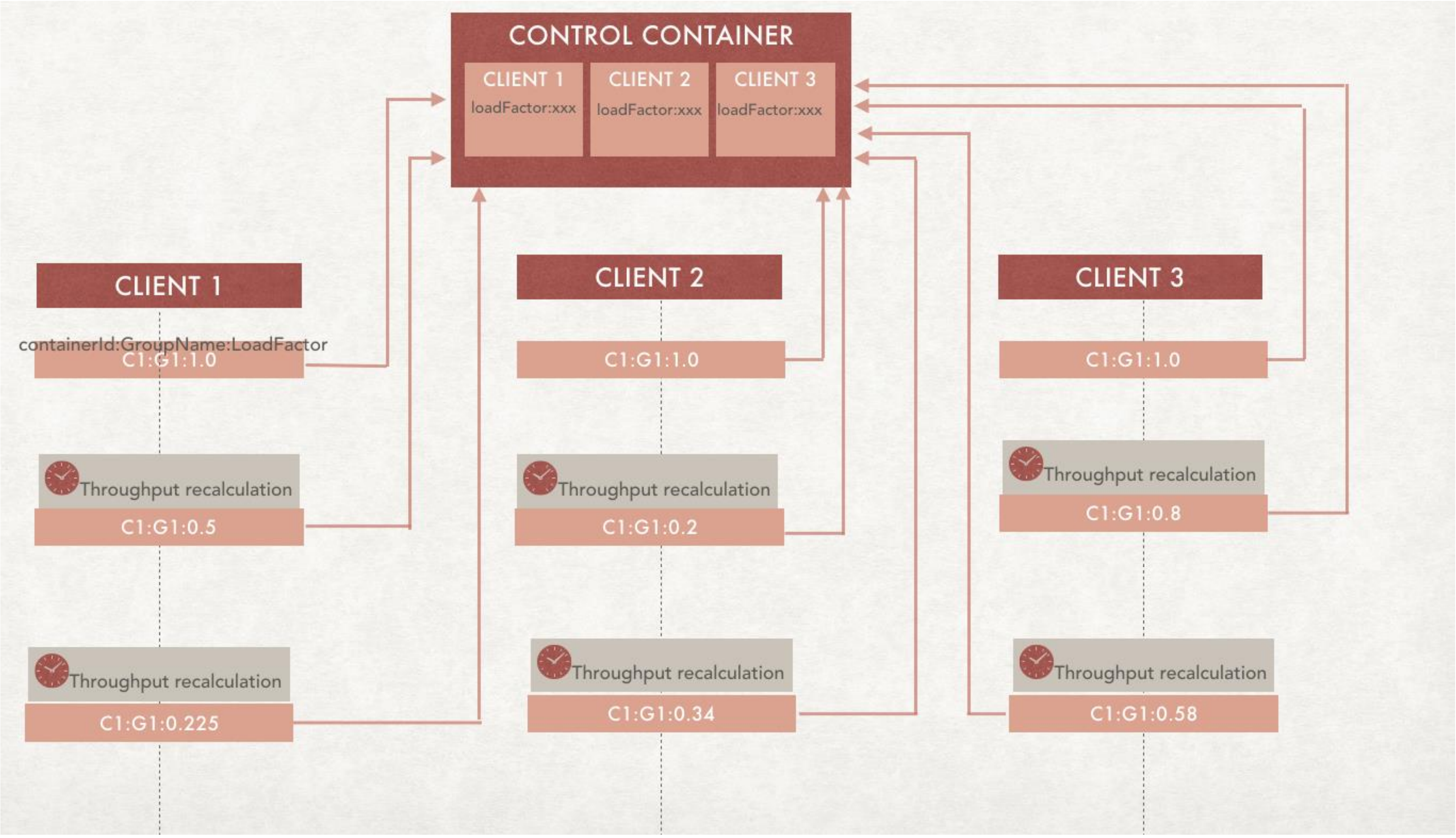
- The throughput control group only applies to the client that creates it.

Global Control:

- Multiple cosmos client instances will coordinate with each other to share the throughput of a throughput control group. For example, cosmos client instances on different machines.
- Throughput is constantly load balancing between cosmos client instances .
- Tracking client's load (calculation based on throughput usage) in a separate container -> customer provided.

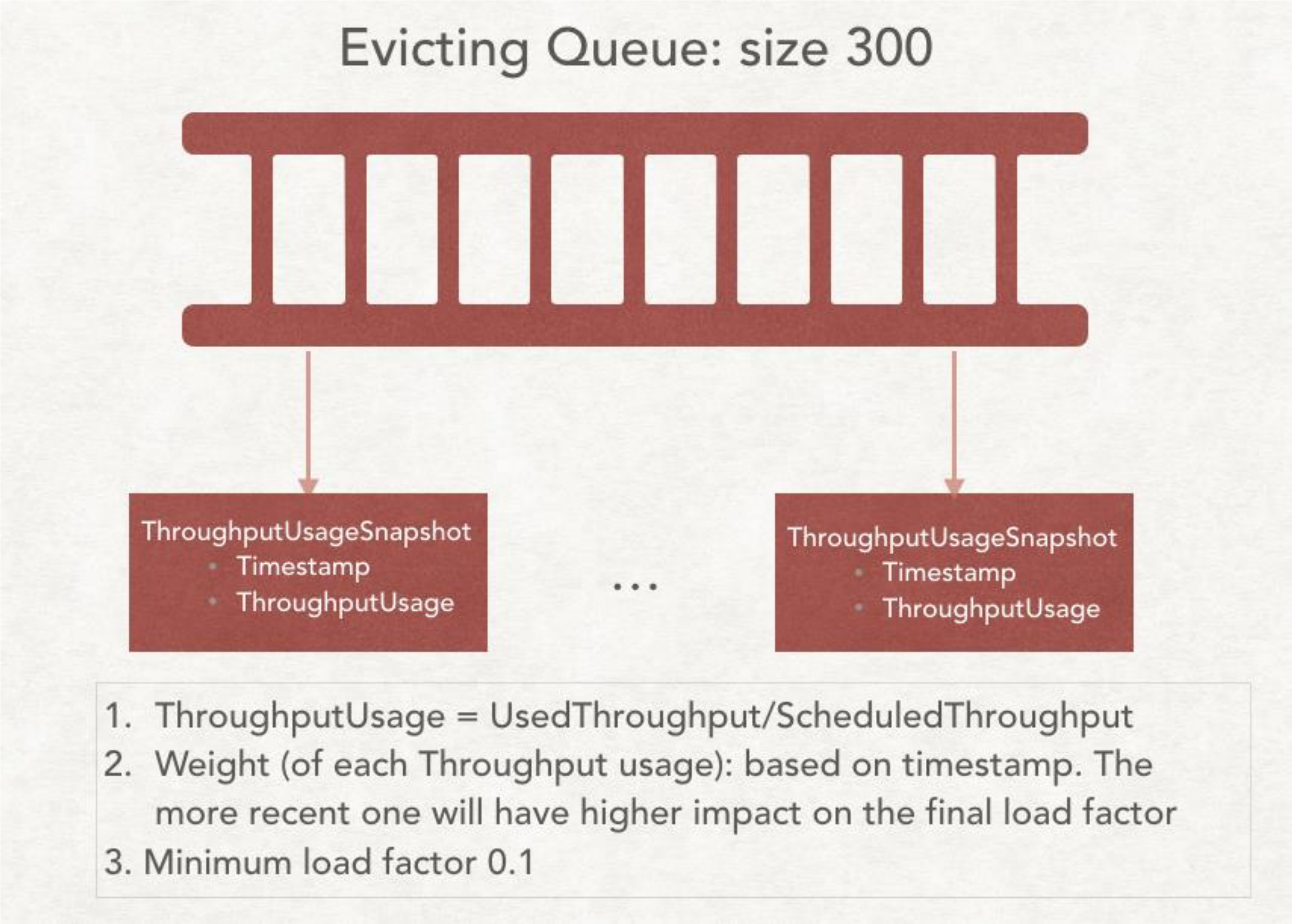
Throughput global control

Scenario: Enable throughput control group G1 on Container C1 with target throughput 10 RU



Load Factor

A measure of how throughput is being used during a certain period



1. ThroughputUsage = UsedThroughput/ScheduledThroughput
2. Weight (of each Throughput usage): based on timestamp. The more recent one will have higher impact on the final load factor
3. Minimum load factor 0.1

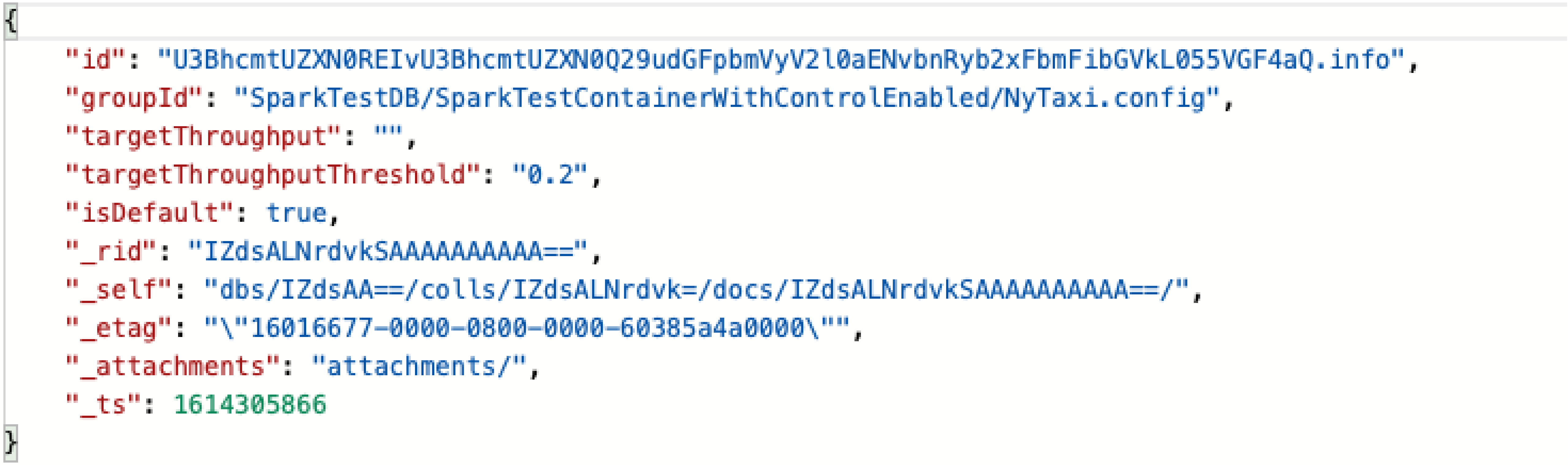
Throughput share

The throughput percentage the client should take from each throughput control group

= selfLoadFactor / SumLoadFactorsFromAllClients

Control Container

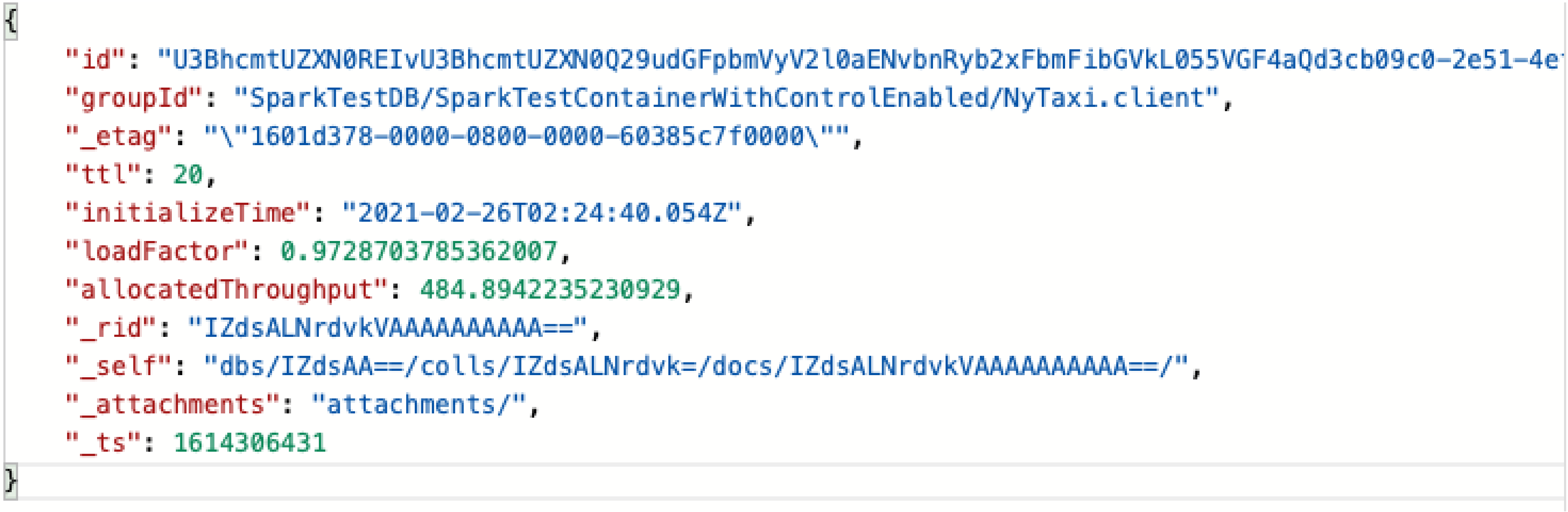
Config Item



```
{
    "id": "U3BhcmtUZXN0REIvU3BhcmtUZXN0Q29udGFpbmVyV2l0aENvbnRyb2xFbmFibGVkL055VGF4aQ.info",
    "groupId": "SparkTestDB/SparkTestContainerWithControlEnabled/NyTaxi.config",
    "targetThroughput": "",
    "targetThroughputThreshold": "0.2",
    "isDefault": true,
    "_rid": "IZdsALNrdvkSAAAAAAAAAA==",
    "_self": "dbs/IZdsAA==/colls/IZdsALNrdvk=/docs/IZdsALNrdvkSAAAAAAAAAA==/",
    "_etag": "\"16016677-0000-0800-0000-60385a4a0000\"",
    "_attachments": "attachments/",
    "_ts": 1614305866
}
```

Control Container

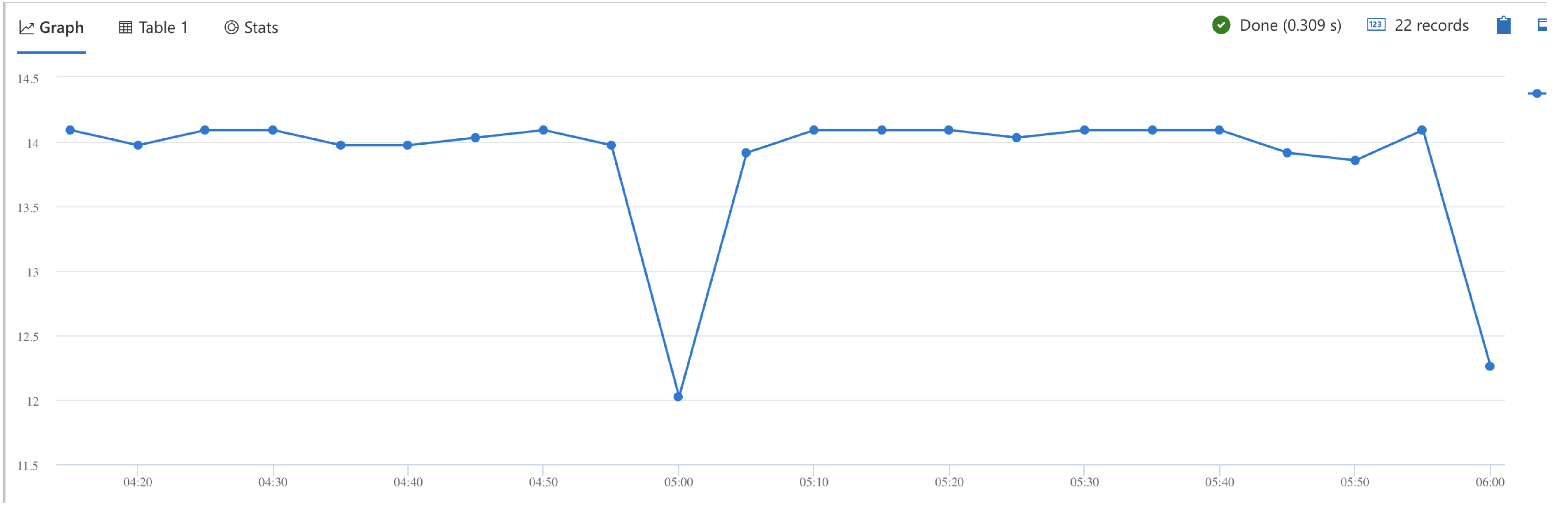
Client Item



```
{
    "id": "U3BhcmtUZXN0REIvU3BhcmtUZXN0Q29udGFpbmVyV2l0aENvbnRyb2xFbmFibGVkL055VGF4aQd3cb09c0-2e51-4e
    "groupId": "SparkTestDB/SparkTestContainerWithControlEnabled/NyTaxi.client",
    "_etag": "\"1601d378-0000-0800-0000-60385c7f0000\"",
    "ttl": 20,
    "initializeTime": "2021-02-26T02:24:40.054Z",
    "loadFactor": 0.9728703785362007,
    "allocatedThroughput": 484.8942235230929,
    "_rid": "IZdsALNrdvkVAAAAAAAAAA==",
    "_self": "dbs/IZdsAA==/colls/IZdsALNrdvk=/docs/IZdsALNrdvkVAAAAAAAAAA==/",
    "_attachments": "attachments/",
    "_ts": 1614306431
}
```

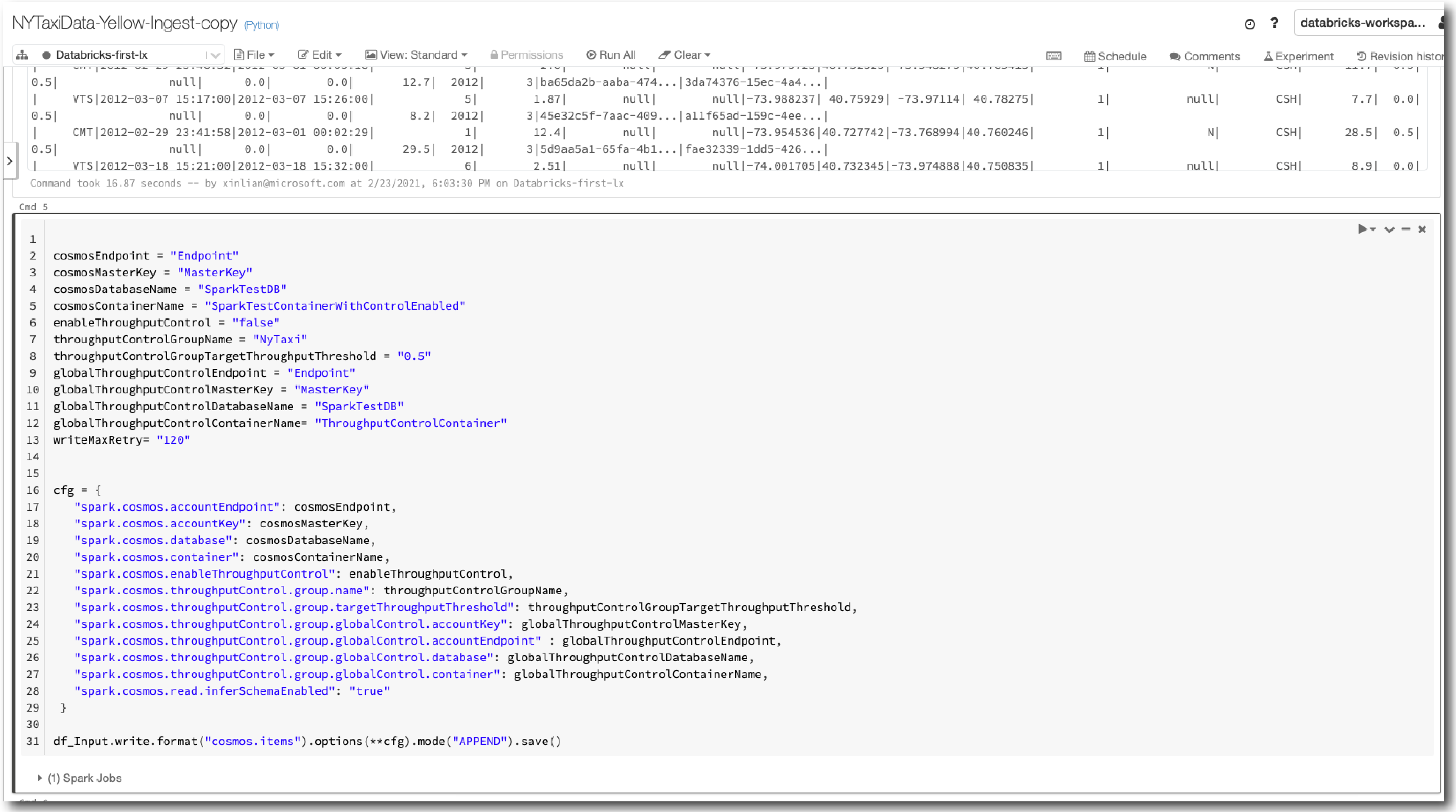
Control Container – RU requirement

- Depends on cosmos client instances total count
- Customer can configure how often do they want to update the client item. By default, it is 5s.
- Example for 8 clients using default configuration



Example with Spark Connector

NYTaxiData Ingestion



```
0.5|               null|       0.0|          0.0|        12.7|  2012|         3|ba65da2b-aaba-474...|3da74376-15ec-4a4...|
|     VTS|2012-03-07 15:17:00|2012-03-07 15:26:00|              5|         1.87|          null|          null|-73.988237| 40.75929| -73.97114| 40.78275|          1|          null|         CSH|        7.7|  0.0|
0.5|               null|       0.0|          0.0|         8.2|  2012|         3|45e32c5f-7aac-409...|a11f65ad-159c-4ee...|
|     CMT|2012-02-29 23:41:58|2012-03-01 00:02:29|              1|         12.4|          null|          null|-73.954536|40.727742|-73.768994|40.760246|          1|             N|         CSH|       28.5|  0.5|
0.5|               null|       0.0|          0.0|        29.5|  2012|         3|5d9aa5a1-65fa-4b1...|fae32339-1dd5-426...|
|     VTS|2012-03-18 15:21:00|2012-03-18 15:32:00|              6|         2.51|          null|          null|-74.001705|40.732345|-73.974888|40.750835|          1|          null|         CSH|        8.9|  0.0|
Command took 16.87 seconds -- by xinlian@microsoft.com at 2/23/2021, 6:03:30 PM on Databricks-first-lx
```

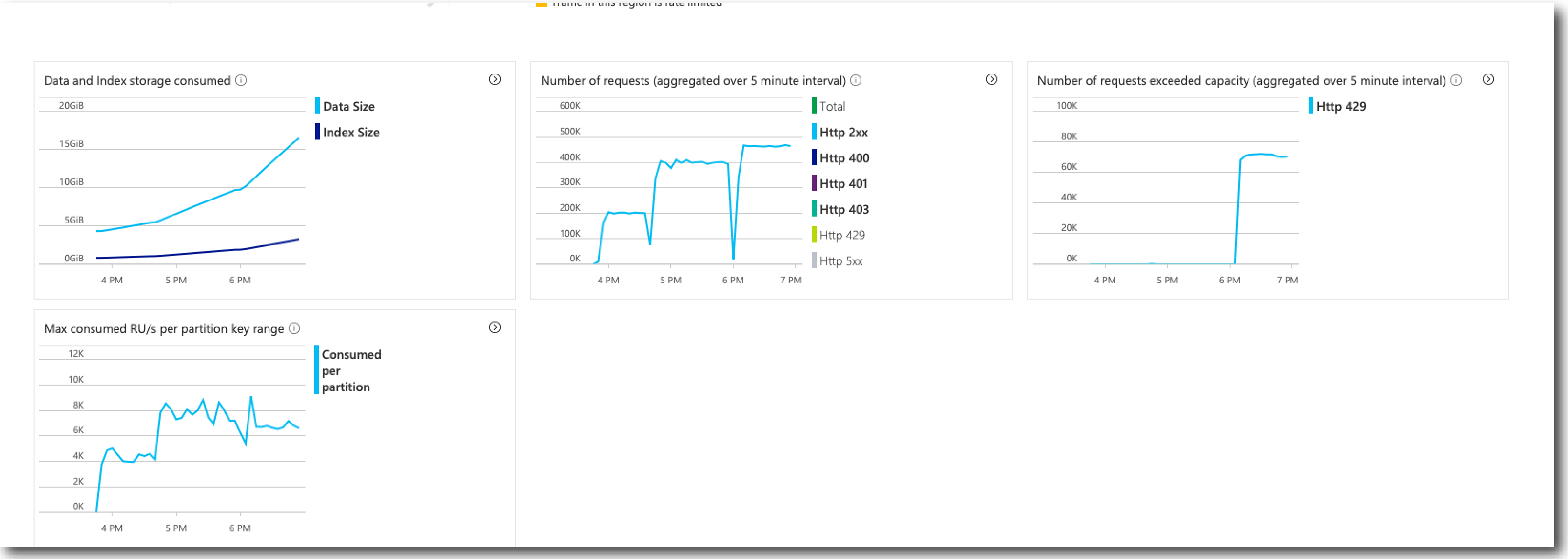
Cmd 5

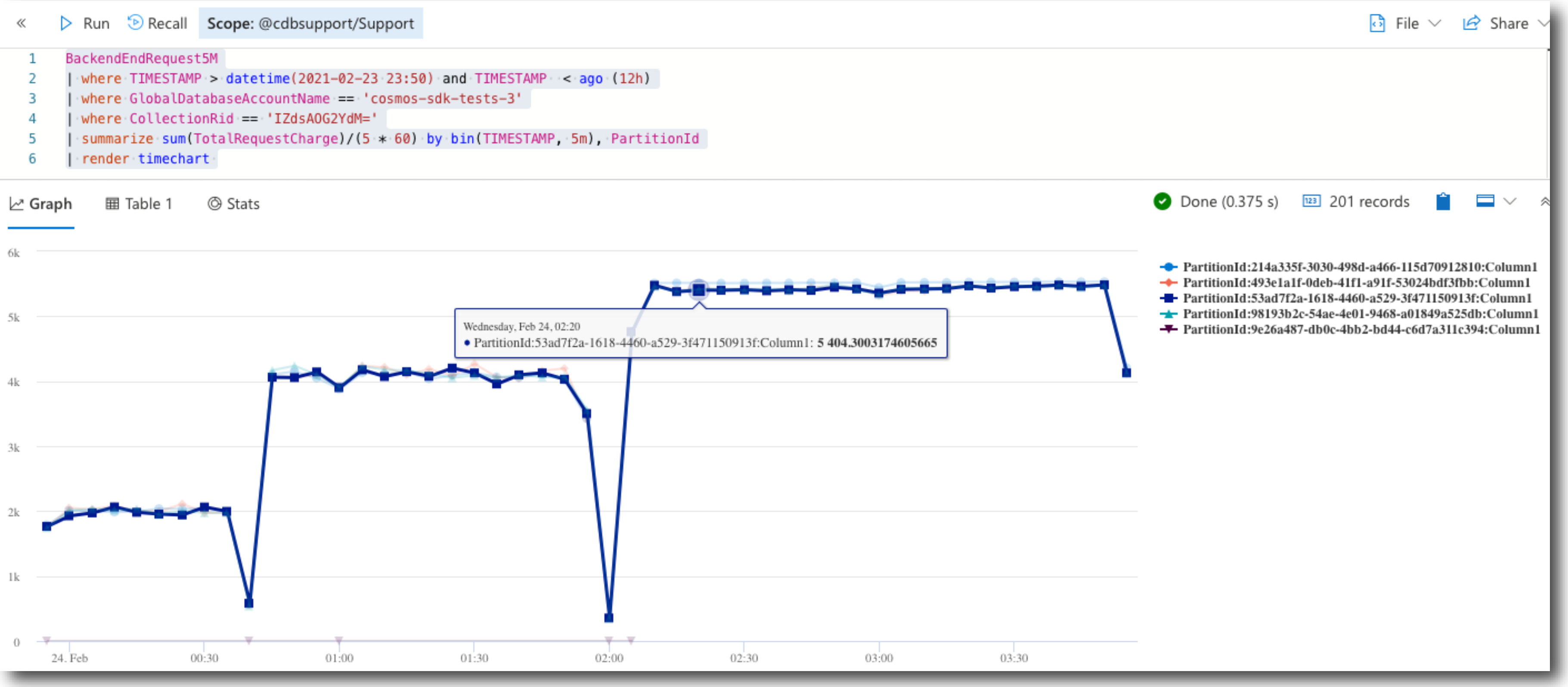
```python

cosmosEndpoint = "Endpoint"
cosmosMasterKey = "MasterKey"
cosmosDatabaseName = "SparkTestDB"
cosmosContainerName = "SparkTestContainerWithControlEnabled"
enableThroughputControl = "false"
throughputControlGroupName = "NyTaxi"
throughputControlGroupTargetThroughputThreshold = "0.5"
globalThroughputControlEndpoint = "Endpoint"
globalThroughputControlMasterKey = "MasterKey"
globalThroughputControlDatabaseName = "SparkTestDB"
globalThroughputControlContainerName= "ThroughputControlContainer"
writeMaxRetry= "120"


cfg = {
   "spark.cosmos.accountEndpoint": cosmosEndpoint,
   "spark.cosmos.accountKey": cosmosMasterKey,
   "spark.cosmos.database": cosmosDatabaseName,
   "spark.cosmos.container": cosmosContainerName,
   "spark.cosmos.enableThroughputControl": enableThroughputControl,
   "spark.cosmos.throughputControl.group.name": throughputControlGroupName,
   "spark.cosmos.throughputControl.group.targetThroughputThreshold": throughputControlGroupTargetThroughputThreshold,
   "spark.cosmos.throughputControl.group.globalControl.accountKey": globalThroughputControlMasterKey,
   "spark.cosmos.throughputControl.group.globalControl.accountEndpoint" : globalThroughputControlEndpoint,
   "spark.cosmos.throughputControl.group.globalControl.database": globalThroughputControlDatabaseName,
   "spark.cosmos.throughputControl.group.globalControl.container": globalThroughputControlContainerName,
   "spark.cosmos.read.inferSchemaEnabled": "true"
 }

df_Input.write.format("cosmos.items").options(**cfg).mode("APPEND").save()
```

(1) Spark Jobs

Test Configuration:

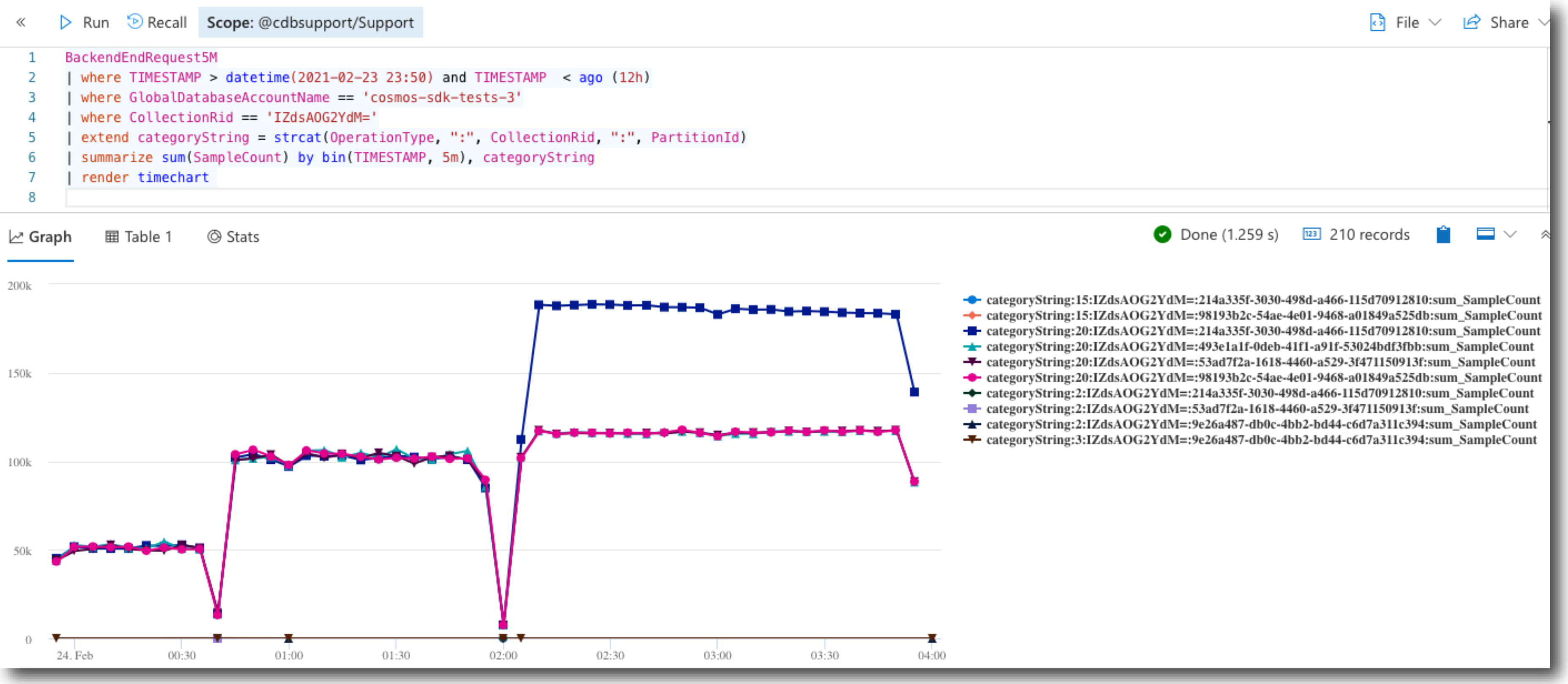
- SparkTestContainerWithControlEnabled, West US2, 20,000 RU/s
- Three runs:
  - Run with throughput control enabled with threshold 0.2
  - Run with throughput control enabled with threshold 0.5
  - Run with throughout control disabled





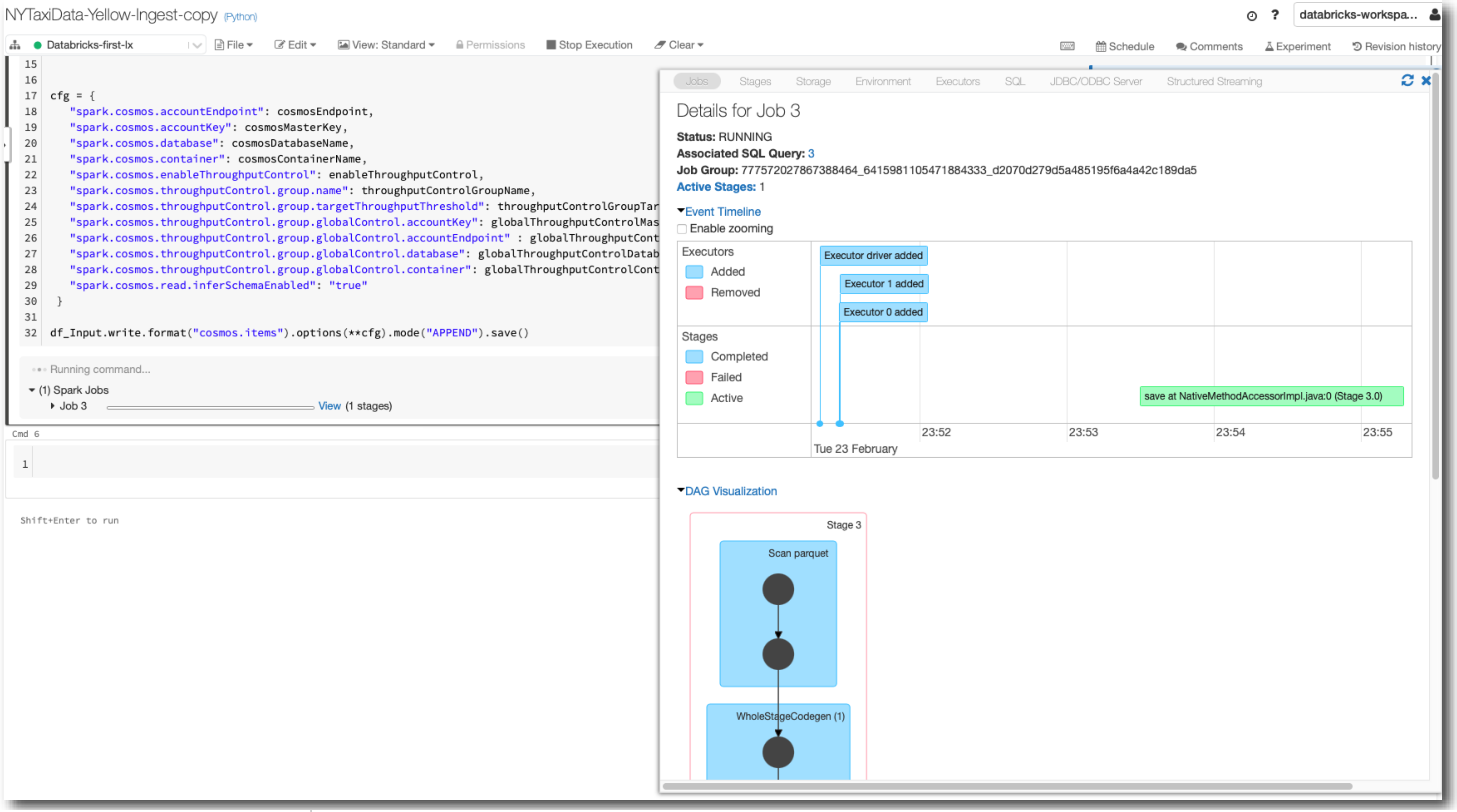
```
BackendEndRequest5M
| where TIMESTAMP > datetime(2021-02-23 23:50) and TIMESTAMP  < ago (12h)
| where GlobalDatabaseAccountName == 'cosmos-sdk-tests-3'
| where CollectionRid == 'IZdsAOG2YdM='
| summarize sum(TotalRequestCharge)/(5 * 60) by bin(TIMESTAMP, 5m), PartitionId
| render timechart
```

Run
Recall
Scope: @cdbsupport/Support
File
Share
Graph
Table 1
Stats
Done (0.375 s)
201 records
6k
5k
4k
3k
2k
1k
0
24. Feb
00:30
01:00
01:30
02:00
02:30
03:00
03:30
Wednesday, Feb 24, 02:20
● PartitionId:53ad7f2a-1618-4460-a529-3f471150913f:Column1: 5 404.3003174605665
PartitionId:214a335f-3030-498d-a466-115d70912810:Column1
PartitionId:493e1a1f-0deb-41f1-a91f-53024bdf3fbb:Column1
PartitionId:53ad7f2a-1618-4460-a529-3f471150913f:Column1
PartitionId:98193b2c-54ae-4e01-9468-a01849a525db:Column1
PartitionId:9e26a487-db0c-4bb2-bd44-c6d7a311c394:Column1



```
BackendEndRequest5M
| where TIMESTAMP > datetime(2021-02-23 23:50) and TIMESTAMP  < ago (12h)
| where GlobalDatabaseAccountName == 'cosmos-sdk-tests-3'
| where CollectionRid == 'IZdsAOG2YdM='
| extend categoryString = strcat(OperationType, ":", CollectionRid, ":", PartitionId)
| summarize sum(SampleCount) by bin(TIMESTAMP, 5m), categoryString
| render timechart
```

Run
Recall
Scope: @cdbsupport/Support
File
Share
Graph
Table 1
Stats
Done (1.259 s)
210 records
200k
150k
100k
50k
0
24. Feb
00:30
01:00
01:30
02:00
02:30
03:00
03:30
04:00
categoryString:15:IZdsAOG2YdM=:214a335f-3030-498d-a466-115d70912810:sum_SampleCount
categoryString:15:IZdsAOG2YdM=:98193b2c-54ae-4e01-9468-a01849a525db:sum_SampleCount
categoryString:20:IZdsAOG2YdM=:214a335f-3030-498d-a466-115d70912810:sum_SampleCount
categoryString:20:IZdsAOG2YdM=:493e1a1f-0deb-41f1-a91f-53024bdf3fbb:sum_SampleCount
categoryString:20:IZdsAOG2YdM=:53ad7f2a-1618-4460-a529-3f471150913f:sum_SampleCount
categoryString:20:IZdsAOG2YdM=:98193b2c-54ae-4e01-9468-a01849a525db:sum_SampleCount
categoryString:2:IZdsAOG2YdM=:214a335f-3030-498d-a466-115d70912810:sum_SampleCount
categoryString:2:IZdsAOG2YdM=:53ad7f2a-1618-4460-a529-3f471150913f:sum_SampleCount
categoryString:2:IZdsAOG2YdM=:9e26a487-db0c-4bb2-bd44-c6d7a311c394:sum_SampleCount
categoryString:3:IZdsAOG2YdM=:9e26a487-db0c-4bb2-bd44-c6d7a311c394:sum_SampleCount



```
cfg = {
   "spark.cosmos.accountEndpoint": cosmosEndpoint,
   "spark.cosmos.accountKey": cosmosMasterKey,
   "spark.cosmos.database": cosmosDatabaseName,
   "spark.cosmos.container": cosmosContainerName,
   "spark.cosmos.enableThroughputControl": enableThroughputControl,
   "spark.cosmos.throughputControl.group.name": throughputControlGroupName,
   "spark.cosmos.throughputControl.group.targetThroughputThreshold": throughputControlGroupTar
   "spark.cosmos.throughputControl.group.globalControl.accountKey": globalThroughputControlMas
   "spark.cosmos.throughputControl.group.globalControl.accountEndpoint" : globalThroughputCont
   "spark.cosmos.throughputControl.group.globalControl.database": globalThroughputControlDatab
   "spark.cosmos.throughputControl.group.globalControl.container": globalThroughputControlCont
   "spark.cosmos.read.inferSchemaEnabled": "true"
 }

df_Input.write.format("cosmos.items").options(**cfg).mode("APPEND").save()
```

Running command...

(1) Spark Jobs
Job 3 View (1 stages)

Cmd 6

Shift+Enter to run

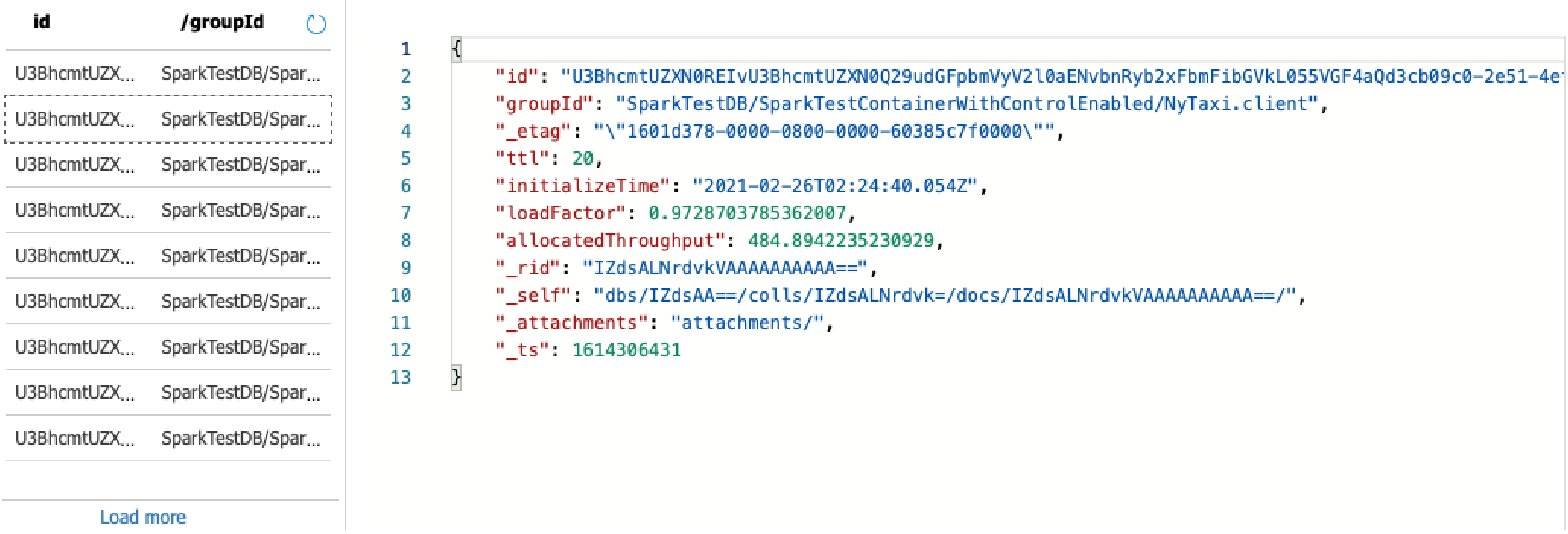
Details for Job 3

Status: RUNNING
Associated SQL Query: 3
Job Group: 7775720278673884 64_6415981105471884333_d2070d279d5a485195f6a4a42c189da5
Active Stages: 1

Event Timeline

□ Enable zooming

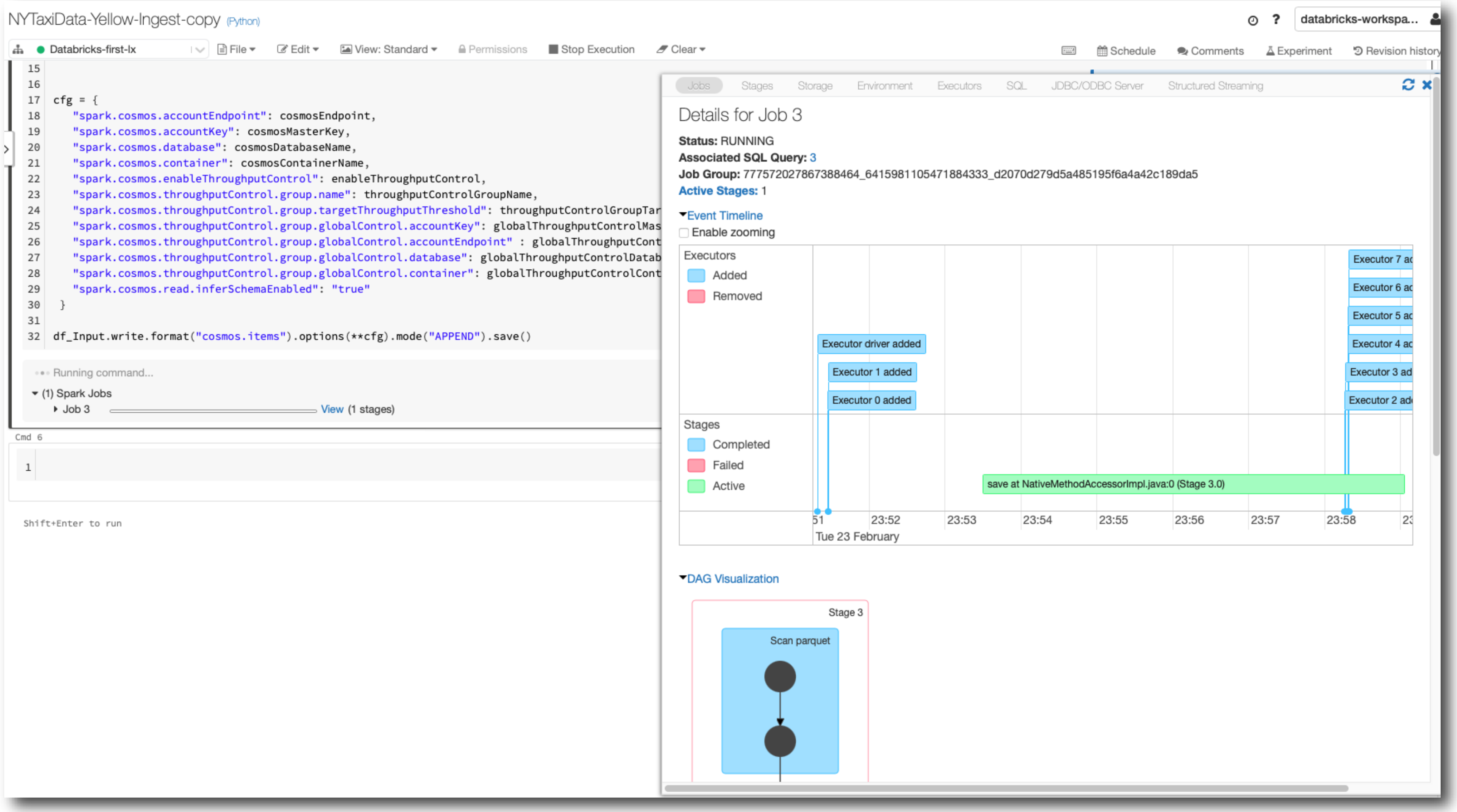
DAG Visualization



id	/groupId
U3BhcmtUZX...	SparkTestDB/Spar...
U3BhcmtUZX...	SparkTestDB/Spar...
U3BhcmtUZX...	SparkTestDB/Spar...
U3BhcmtUZX...	SparkTestDB/Spar...
U3BhcmtUZX...	SparkTestDB/Spar...
U3BhcmtUZX...	SparkTestDB/Spar...
U3BhcmtUZX...	SparkTestDB/Spar...
U3BhcmtUZX...	SparkTestDB/Spar...
U3BhcmtUZX...	SparkTestDB/Spar...

Load more

```
{
    "id": "U3BhcmtUZXN0REIvU3BhcmtUZXN0Q29udGFpbmVyV2l0aENvbnRyb2xFbmFibGVkL055VGF4aQd3cb09c0-2e51-4e
    "groupId": "SparkTestDB/SparkTestContainerWithControlEnabled/NyTaxi.client",
    "_etag": "\"1601d378-0000-0800-0000-60385c7f0000\"",
    "ttl": 20,
    "initializeTime": "2021-02-26T02:24:40.054Z",
    "loadFactor": 0.9728703785362007,
    "allocatedThroughput": 484.8942235230929,
    "_rid": "IZdsALNrdvkVAAAAAAAAAA==",
    "_self": "dbs/IZdsAA==/colls/IZdsALNrdvk=/docs/IZdsALNrdvkVAAAAAAAAAA==/",
    "_attachments": "attachments/",
    "_ts": 1614306431
}
```



```
cfg = {
   "spark.cosmos.accountEndpoint": cosmosEndpoint,
   "spark.cosmos.accountKey": cosmosMasterKey,
   "spark.cosmos.database": cosmosDatabaseName,
   "spark.cosmos.container": cosmosContainerName,
   "spark.cosmos.enableThroughputControl": enableThroughputControl,
   "spark.cosmos.throughputControl.group.name": throughputControlGroupName,
   "spark.cosmos.throughputControl.group.targetThroughputThreshold": throughputControlGroupTar
   "spark.cosmos.throughputControl.group.globalControl.accountKey": globalThroughputControlMas
   "spark.cosmos.throughputControl.group.globalControl.accountEndpoint" : globalThroughputCont
   "spark.cosmos.throughputControl.group.globalControl.database": globalThroughputControlDatab
   "spark.cosmos.throughputControl.group.globalControl.container": globalThroughputControlCont
   "spark.cosmos.read.inferSchemaEnabled": "true"
 }

df_Input.write.format("cosmos.items").options(**cfg).mode("APPEND").save()
```

Running command...

(1) Spark Jobs
Job 3 View (1 stages)

Cmd 6

Shift+Enter to run

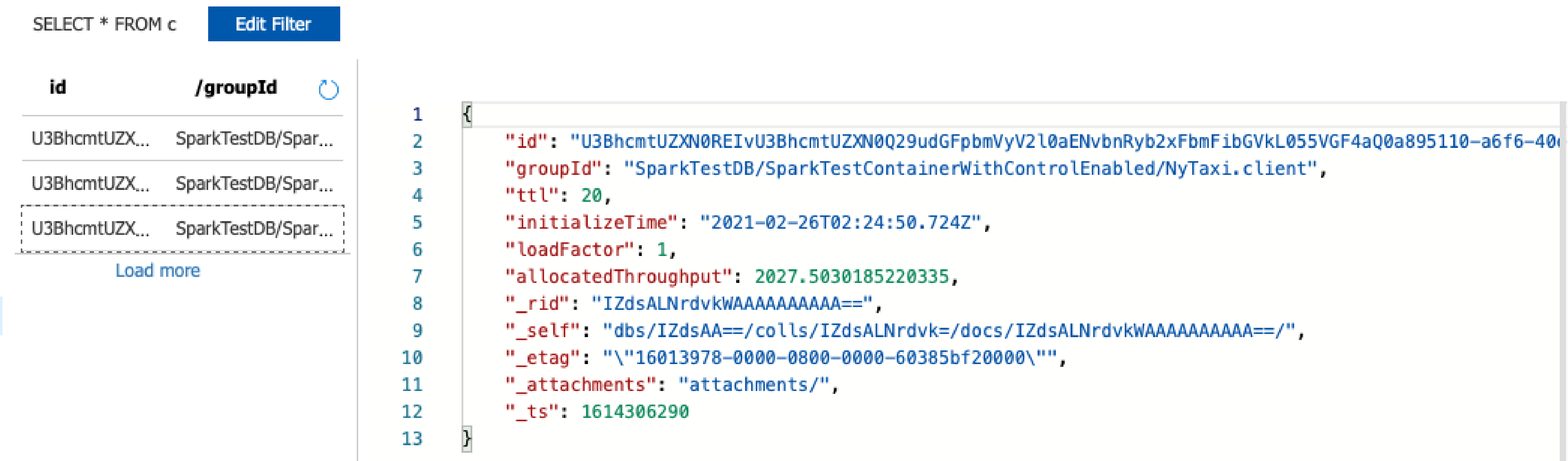
Details for Job 3

Status: RUNNING
Associated SQL Query: 3
Job Group: 7775720278673884 64_6415981105471884333_d2070d279d5a485195f6a4a42c189da5
Active Stages: 1

Event Timeline

☐ Enable zooming

DAG Visualization


SELECT * FROM c

id	/groupId
U3BhcmtUZX...	SparkTestDB/Spar...
U3BhcmtUZX...	SparkTestDB/Spar...
U3BhcmtUZX...	SparkTestDB/Spar...

```
{
    "id": "U3BhcmtUZXN0REIvU3BhcmtUZXN0Q29udGFpbmVyV2l0aENvbnRyb2xFbmFibGVkL055VGF4aQ0a895110-a6f6-40
    "groupId": "SparkTestDB/SparkTestContainerWithControlEnabled/NyTaxi.client",
    "ttl": 20,
    "initializeTime": "2021-02-26T02:24:50.724Z",
    "loadFactor": 1,
    "allocatedThroughput": 2027.5030185220335,
    "_rid": "IZdsALNrdvkWAAAAAAAAAA==",
    "_self": "dbs/IZdsAA==/colls/IZdsALNrdvk=/docs/IZdsALNrdvkWAAAAAAAAAA==/",
    "_etag": "\"16013978-0000-0800-0000-60385bf20000\"",
    "_attachments": "attachments/",
    "_ts": 1614306290
}
```

Limitations & Future Work

Limitations & Future Work

- Throughput control does not do RU pre-calculation of each operation, it tracks the RU usages after the operation based on the response header.
- Throughput control works on its best effort.
- Support for serverless account
- Add extended metrics support for throughput control group