[Kafka] Performance

Tools

Kafka package provides performance following testing tools:

- kafka-producer-perf-test.sh - Performance at Producer End (write data)
- kafka-consumer-perf-test.sh - Performance at Consumer End (read data)

https://cwiki.apache.org/confluence/display/KAFKA/Performance+testing

The main intent of this tests is to find out the following stats:

1. Throughput(messages/sec) on size of data

2. Throughput(messages/sec) on number of messages

3. Total data

4. Total Messages

We collected node performance statistics by prometheus node exporter.

It could be also used specialized performance testing tool like Apache JMeter and Pepper-Box - Kafka Load Generator.

Clusters configuration

Servers

We have installed 4 VMs in Yandex Cloud with following configurations:

- CPU Cores = 8
- Memory = 16GB
- Data Disk Size = 1536 GB (disk performance depends on size: https://cloud.yandex.com/docs/compute/concepts/limits#limits-disks)
- disk_type = "network-hdd"

Disk limits

Network SSD | Network HDD

Type of limit	Value
Maximum disk size	4 TB
Maximum disk snapshot size	4 TB
Allocation unit size	256 GB
Maximum* IOPS for writes, per disk	11,000
Maximum* IOPS for writes, per allocation unit	300
Maximum** bandwidth for writes, per disk	240 MB/s
Maximum** bandwidth for writes, per allocation unit	30 MB/s
Maximum* IOPS for reads, per disk	300
Maximum* IOPS for reads, per allocation unit	100
Maximum** bandwidth for reads, per disk	240 MB/s
Maximum** bandwidth for reads, per allocation unit	30 MB/s

Maximum disk performance must be 1536GB/256GB(Allocation unit)*30MB/s = **180MB/s** read write per node. 100 * 6 = **600 IOPS** read write per node.

Software

We have installed zookeeper version 3.5.5 on the first 3 nodes.

We have installed latest available Apache Kafka 2.3.1 https://kafka.apache.org/downloads#2.3.1 with following non default params:

```
auto.create.topics.enable=True  auto creation for testing purposes
default.replication.factor=3  prod-like RF
num.partitions=20  5 partitions per node, provides best consuming possibilities.
log.dirs=/data1/kafka  external data disk formatted in xfs with default mount options.
```

Pricing

Total cluster costs:
4 x 8755.23 per month = **35020.92** per month

Per unit, per resource type:

Intel Cascade Lake. 100% vCPU 4306.18
Intel Cascade Lake. RAM 2280.96
Standard Storage (HDD) 2168.09

Tests

Producer

```
Example test config:

./kafka-producer-perf-test.sh \
  --topic test.topic \
  --num-records 10000000 \
  --record-size 2048 \
  --throughput -1 \
  --producer-props acks=1 \
  bootstrap.servers=10.80.66.43:9092,10.80.66.51:9092,10.80.66.6:9092,10.80.66.7:9092 \
  buffer.memory=67108864 \
  compression.type=gzip \
  batch.size=1
```

Consumer

```
./kafka-consumer-perf-test.sh \
--broker-list=10.80.66.43:9092,10.80.66.51:9092,10.80.66.6:9092,10.80.66.7:9092 \
--messages 50000000 \
--topic test.topic \
--group test1 \
--threads 5
```

Results

Producer long test

params: nocomp, acks=1, messagesize=2KB

100000000 records sent, 127346.518792 records/sec (**242.89 MB/sec**), 255.25 ms avg latency, 2427.00 ms max latency, 7 ms 50th, 106 ms 95th, 241 ms 99th, 387 ms 99.9th.

Total time 12 mins. Total data produced: 190GB (2KB*100.000.000). Average IOPS per server - 400.

Producer multi tests

		Kafka Version		2.1.3	2.1.3	2.1.3	2.1.3	2.1.3	2.1.3	2.1.3	2.1.3	2.1.3
		Replication Factor		3	3	2	3	3	2	3	3	2
		Acks		1 (leader only)	-1 (all ISR)	-1	1	-1	-1	1	-1	-1
		Compression		Uncomp	Uncomp	Uncomp	Gzip	Gzip	Gzip	Snappy	Snappy	Snappy
		Record size		2048 (2KB)	2048 (2KB)	2048 (2KB)	2048 (2KB)	2048 (2KB)	2048 (2KB)	2048 (2KB)	2048 (2KB)	2048 (2KB)
Producer location	**Batch size**											
Ya.Cloud C zone	1			60833.54 records /sec 118.82 MB/sec 514.72 ms avg latency	8773.54 records /sec 17.14 MB/sec 3463.81 ms avg latency	10996.26 records /sec 21.48 MB/sec 2443.30 ms avg latency	24218.93 records/sec 47.30 MB/sec **9.89 ms avg latency**	8763.16 records /sec 17.12 MB/sec 3409.19 ms avg latency	10355.18 records /sec 20.22 MB/sec 2811.29 ms avg latency	54487.00 records /sec 106.42 MB/sec 551.90 ms avg latency	8984.21 records /sec 17.55 MB/sec 3527.2 ms avg latency	10411.56 records /sec 20.34 MB/sec 2945.42 ms avg latency
Ya.Cloud C zone	32			61560.24 records /sec 120.24 MB/sec 505.6 ms avg latency	8796.15 records /sec 17.18 MB/sec 3458.83 ms avg latency	...	23000.13 records/sec 44.92 MB/sec 25.52 ms avg latency	...	...	...	...	...
Ya.Cloud C zone	1024			57321.37 records /sec 111.96 MB/sec 539.42 ms avg latency	8798.63 records /sec 17.18 MB/sec 3468.89 ms avg latency	10323.11 records /sec 20.16 MB/sec 2672.79 ms avg latency	...	...	...	...	...	...
Ya.Cloud C zone	4096			74694.21 records /sec 142.47 MB/sec 435.3 ms avg latency	8920.28 records /sec 17.42 MB/sec 1800.33 ms avg latency	10023.05 records /sec 19.58 MB/sec 1476.08 ms avg latency	...	...	...	...	...	...

Ya.Cloud C zone	16384			143374.31 records/sec 280.03 MB/sec 193.01 ms avg latency	48388.65 records /sec 94.51 MB/sec 567.32 ms avg latency	55732.04 records /sec 108.85 MB/sec 489.02 ms avg latency	65155.06 records/sec **127.26 MB/sec** 12.14 ms avg latency	63379.38 records /sec 123.79 MB/sec 12.87 ms avg latency	65595.27 records /sec **128.12 MB/sec** 12.30 ms avg latency,	760745.53 records/sec **1485.83 MB/sec** 4.47 ms avg latency	314911.03 records/sec 615.06 MB/sec 568.31 ms avg latency	382065.82 records/sec 746.22 MB/sec 463.24 ms avg latency
Ya.Cloud C zone	32768			173913.04 records/sec **339.67 MB/sec** 170.58 ms avg latency	78845.69 records /sec 154.00 MB/sec 370.03 ms avg latency	100220.48 records/sec 195.74 MB/sec 285.00 ms avg latency	...	...	...	...	...	...
Ya.Cloud C zone	65536			171328.22 records/sec 334.63 MB/sec 177.40 ms avg latency	97285.72 records /sec **190.01 MB/sec** 305.52 ms avg latency,	121418.16 records/sec **237.14 MB/sec** 237.79 ms avg latency,	63488.03 records/sec 124.00 MB/sec 12.31 ms avg latency,	66489.36 records /sec **129.86 MB/sec** 12.52 ms avg latency	64316.95 records /sec 125.62 MB/sec 21.31 ms avg latency	730513.55 records/sec 1426.78 MB/sec 4.20 ms avg latency	644080.89 records/sec **1257.97 MB/sec** 23.89 ms avg latency	691969.69 records/sec **1351.50 MB/sec** 17.87 ms avg latency

Fields description:

Acks: acknowledgments - producer will not set message as produced until receive approve from leader or all ISR (in-sync replicas) or none of them (1, -1, 0):

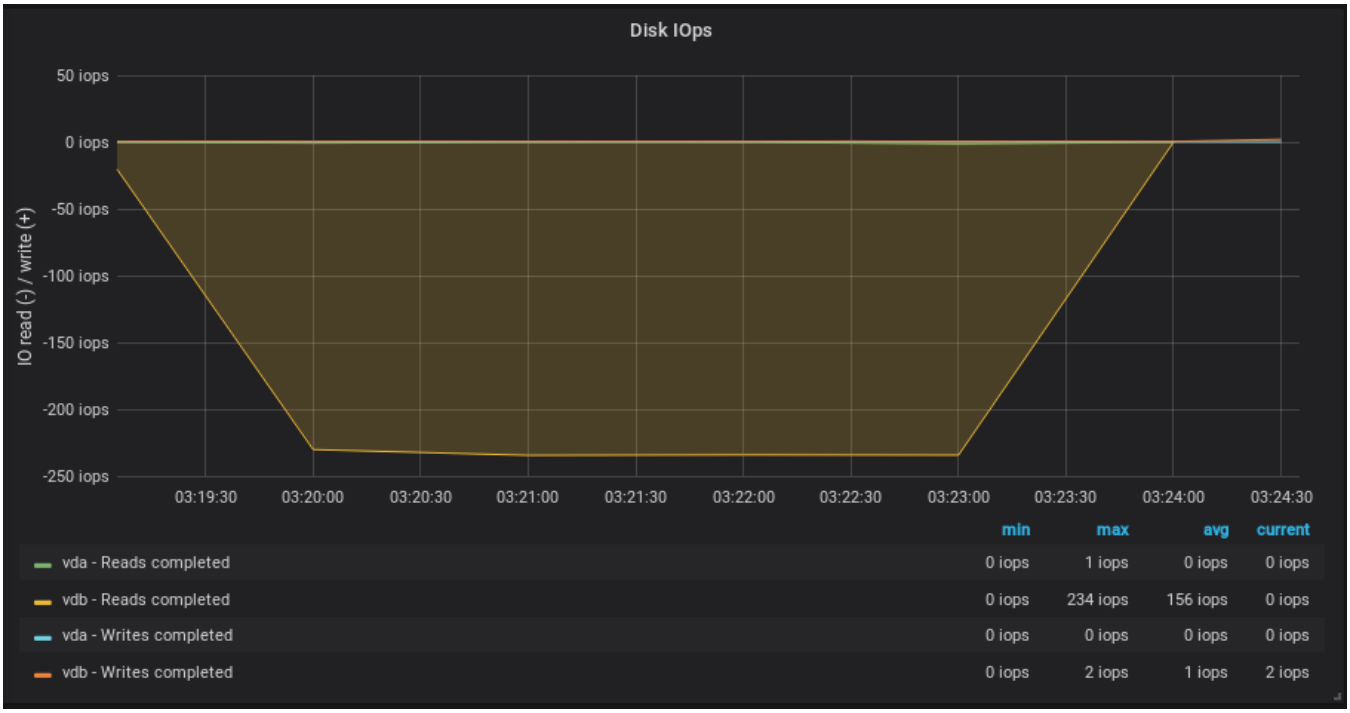
Acks	Throughput	Latency	Durability
0	High	Low	No Gurantee
1	Medium	Medium	Leader
-1	Low	High	ISR

Consumer long test

params: 5 threads, consume 50.000.000 messages

start.time, end.time, data.consumed.in.MB, MB.sec, data.consumed.in.nMsg, nMsg.sec, rebalance.time.ms, fetch.time.ms, fetch.MB.sec, fetch.nMsg.sec
2019-12-14 03:19:34:879, 2019-12-14 03:23:37:756, 98464.1030, 405.4073, 50000199, 205866.3398, 132, 242745, 405.6277, 205978.2859

Total time 4 mins. Total data consumed: 95GB (2KB*50.000.000). **405.62MB/s.** Average IOPS per server - 234.



Possible configuration improvements

Have not tested:

- Filesystem (https://www.confluent.io/kafka-summit-sf18/kafka-on-zfs/)

- Disk mount options (noatime, etc)
- -Xmx8g -Xms8g (default is -Xmx1G -Xms1G)
- Changing socket buffer size (+netstack tuning)
- Multiple dedicated disks (with multiple data dirs)

https://community.cloudera.com/t5/Community-Articles/Kafka-Best-Practices/ta-p/249371

https://www.slideshare.net/JiangjieQin/producer-performance-tuning-for-apache-kafka-63147600