Kafka Monitoring

Hardware and OS

- Dual quad-core Intel Xeon machines with 24GB of memory or higher
  - for production mission critical system
  - 24 GB total but only 25% of that for JVM (6 GB)
- Kafka Broker needs memory to buffer active readers and writers
  - to buffer for 30 seconds and memory needed is write_throughput*30
- Disk throughput is important
  - 8x7200 rpm SATA drives
  - Disk throughput is often performance bottleneck
  - JBOD - more disks is better

OS

Tos

- Kafka production usually runs on Linux
- Ensure you have enough file descriptors
  - Kafka uses file descriptors for log segments and open connections
  - (number_of_partitions)*(partition_size/segment_size) + number_of_producer_connections + number_of_consumer_connections
  - Start with 100,000 or more file descriptors
- Max socket buffer size:
  - increased to enable high-performance data transfer between data centers
- Use JBOD instead of RAID, RAID ok, JBOD better
- Check flusher threads and PDFlush but defaults should be ok
- Prefer filesystem XFS (largeio, nobarrier), EXT4 ok too (data=writeback, commit=num_secs, nobh, delalloc)

not preserve data ordering when writing to the disk

Monitoring

- Kafka uses Yammer Metrics
  - metrics reporting for Kafka Broke, Consumers and Producers
  - Reports stats using pluggable stats reporters
- Metrics exposed via JMX
- You can see what metrics are available with jconsole

Tos

Kafka Broker Metrics -1 of 3

Description	JMX MBEAN NAME
Message in rate	Kafka.server:type=BrokerTopicMetrices, name=MessagesInPerSec
Byte in rate	Kafka.server:type=BrokerTopicMetrics,name=BytesInPerSec
Request rate	Kafka.network:type=RequestMetrics, name=RequestPeSec, request ={Producer| FetchConsumer| FetchFollower}
Byte out rate	Kafka.server:type=BrokerTopicMetrics,name=BytesOutPerSec
Log flush rate and time	Kafka.log:type=LogFlushStats,name=LogFlushRateAndTimeMs
Time request waits in request queue	Kafka.network:type=RequestMetrics,name=RequestQueueTimeMs, Request={Produce|FetchConsumer|FetchFollower}
Time request is processed at leader	Kafka.network:type=RequestMetrics,name=LocalTimeMs, request={producer|Fetchconsumer|FetchFollower}
Messages count consumer lags behind producer	Kafka.consumer:type=consumer-fetch-manager-metrics, client-id ={client-id} Attribute:records-lag-max

Kafka Broker Metrics - 2 of 3

Under replicated Count partitions	**Kafka.server:type=ReplicaManager,name=Under ReplicatedPartitions**	**0**
Is controller active on broker?	Kafka.controller:type=kafkaController,name =ActiveControllerCount	Only 1 Kafka Broker is controller and has 1. All else should have 0.
Leader election rate	Kafka.controller:type=ControllerStats,name=LeaderElectionRate AndTimeMs	>0 if failures
Unclean leader election rate	Kafka.c ontroller:type=ControllerStats,name= UncleanLeaderElectionsPerSec	0
Partition counts	Kafka.server:type=ReplicaManager, name= PartitionCount	Mostly even across brokers
Leader replica counts	Kafka.server:type=ReplicaManager,name= LeaderCount	Mostly even across brokers
ISR shrink rate	Kafka.server:type=ReplicaManager, name=IsrShrinksPerSec	If a broker dies,ISR shrinks for some partitions.ISR expands when brokers come back
ISR	Kafka.server:type=ReplicaManage,name=IsrExpandsPerSec	Opposite of ISR shrink rate

Kafka Broker Metrics - 3 of 3

Max follower lag	Kafka.server:type=ReplicaFetcherManager,name=MaxLag,client id=Replica	Lag usually proportional to produce maximum batch size
Messages lag per follower	Kafka.server:type=FetcherLagMetrics,name=ConsumerLag,clientId=([-.\W]+),topic=([-.\W]+),partition=([0-9]+)	Lag usually proportional to produce maximum batch size
Requests waiting in producer purgatory	kafka.server:type=DelayedOperationPurgatory,name=Purgatory Size,delayedOperation=Produce	>0 if ack=all is used
Requests waiting in fetch purgatory	kafka.server:type=DelayedOperationPurgatory,name=Purgatory Size,delayedOperation=Fetch	size depends on consumer config fetch.wait.max.ms
Request total time	kafka.server:type=DelayedOperationPurgatory,name=Purgatory Size,delayedOperation=Fetch	broken into queue, local, remote and response send time
Leader replica counts	kafka.server:type=ReplicaManager,name=LeaderCount	Should be even

Common Metrics for Clients 1 of 2

Metric	Description
connection-close-rate	Connections closed per second JMX MBean Name kafka.[producer|consumer|connect]:type=[producer|consumer|connect]-metrics,client-id=([-.\w]+)
connection-creation-rate	New connections established per second
network-io-rate	Average network operations count on all connections per second.
outgoing-byte-rate	Average outgoing bytes count sent per second to all servers.
request-rate	Average requests count sent per second.
request-size-avg	Average size of all requests
request-size-max	Maximum size of any request

Common Metrics for Clients 2 of 2

Tos

Metric	Description
incoming-byte-rate	Average incoming byte count received by all sockets JMX MBean Name (kafka.[producer|consumer|connect]:type=[producer|consumer|connect]-metrics,client-id=([-.\w]+))
response-rate	Responses received sent per second.
select-rate	I/O layer checked for new I/O to perform per second count
io-wait-time-ns-avg	Average duration I/O thread spent waiting for a socket ready for reads/writes
io-wait-ratio	Fraction of time the I/O thread spent waiting
io-time-ns-avg	Average duration for I/O per select call in nanoseconds.
io-ratio	Fraction of time I/O thread spent doing I/O.
connection-count	Current number of active connections.

Per Kafka Broker Client Monitoring

Tos

Metric	Description
outgoing-byte-rate	Average outgoing byte count sent per second for node JMX MBean Name: kafka.producer:type=[consumer|producer|connect]-node-metrics,client-id=([-.\w]+),node-id=([0-9]+)
request-rate	Average requests count sent per second for a node.
request-size-avg	Average size of all requests for node
request-size-max	Maximum size of any request sent for node
incoming-byte-rate	Average responses received count per second for node
request-latency-avg	Average request latency in ms for node
request-latency-max	Maximum request latency in ms for node
response-rate	Responses received sent per second for node

Kafka Producer Monitoring - 1 of 3

Tos

Metric	Description
waiting-threads	User threads blocked count waiting for buffer memory to enqueue their records. JMX MBean Name kafka.producer:type=producer- metrics,client-id=([-.\w]+)
buffer-total-bytes	Maximum buffer memory size client can use
buffer-available-bytes	Total buffer memory size that is not being used
bufferpool-wait-time	Fraction of time an appender waits for space allocation
batch-size-avg	Average byte count sent per partition per-request.
batch-size-max	Max byte count sent per partition per-request.
compression-rate-avg	Average compression rate of record batches
record-queue-time-avg	Average time in ms record batches spent in record accumulator.
record-queue-time-max	The maximum time in ms record batches spent in the record accumulator.

Kafka Producer Monitoring - 2 of 3

Tos

Metric	Description
request-latency-avg	Average request latency in ms. JMX MBean Name kafka.producer:type=producer- metrics,client-id=([-.\w]+)
request-latency-max	Maximum request latency in ms.
record-send-rate	Average record count sent per second
records-per-request-avg	Average record count per request
record-retry-rate	Average per-second retried record send count
record-error-rate	Average per-second record send count that resulted in errors.
record-size-max	Maximum record size.
record-size-avg	Average record size.
requests-in-flight	Current number of in-flight requests - waiting for a response.

Kafka Producer Monitoring - 3 of 3

Tos

Metric	Description
metadata-age	Age in seconds of current producer metadata being used
record-send-rate	Average records sent count per second for topic
byte-rate	Average bytes sent count per second for topic
compression-rate	Average record batches compression rate for topic
record-retry-rate	Average per-second retried record send count for a topic
record-error-rate	Average per-second record sends that resulted in errors count for topic
produce-throttle-time- max	Maximum time in ms a request was throttled by a broker
produce-throttle-time- avg	Average time in ms a request was throttled by a broker
requests-in-flight	Current number of in-flight requests - waiting for a response.

Kafka Consumer Group Monitoring - 1 of 2

Tos

Metric	Description
commit-latency-avg	Average duration for commit request **kafka.consumer:type=consumer-coordinator-metrics,client-id=([-.\w]+)**
commit-latency-max	Max duration for a commit request
commit-rate	Commit call count per second
assigned-partitions	Partition count currently assigned to consumer
heartbeat-response-time-max	Max duration for heartbeat request to receive response
heartbeat-rate	Average heartbeat count per second
join-time-avg	Average duration for a group rejoin
join-time-max	Max duration for a group rejoin
join-rate	Group join count per second

Kafka Consumer Group Monitoring - 2 of 2

Tos

Metric	Description
sync-time-avg	Average duration for a group sync
sync-time-max	Max duration for a group sync
sync-rate	Group sync count per second
last-heartbeat- seconds-ago	Second count since last controller heartbeat

Kafka Consumer Monitoring

Tos

Metric	Description
fetch-size-avg	Average byte size fetched per request
fetch-size-max	Maximum byte size fetched per request
bytes-consumed-rate	Average byte count consumed per second
records-per-request-avg	Average record count in each request
records-consumed-rate	Average record count consumed per second
fetch-latency-avg	Average fetch request duration
fetch-latency-max	Max fetch request duration
fetch-rate	Fetch request count per second
records-lag-max	Max lag of record count for any partition
fetch-throttle-time-avg	Average throttle time in ms
fetch-throttle-time-max	Maximum throttle time in ms

Kafka Consumer Topic Fetch Monitoring

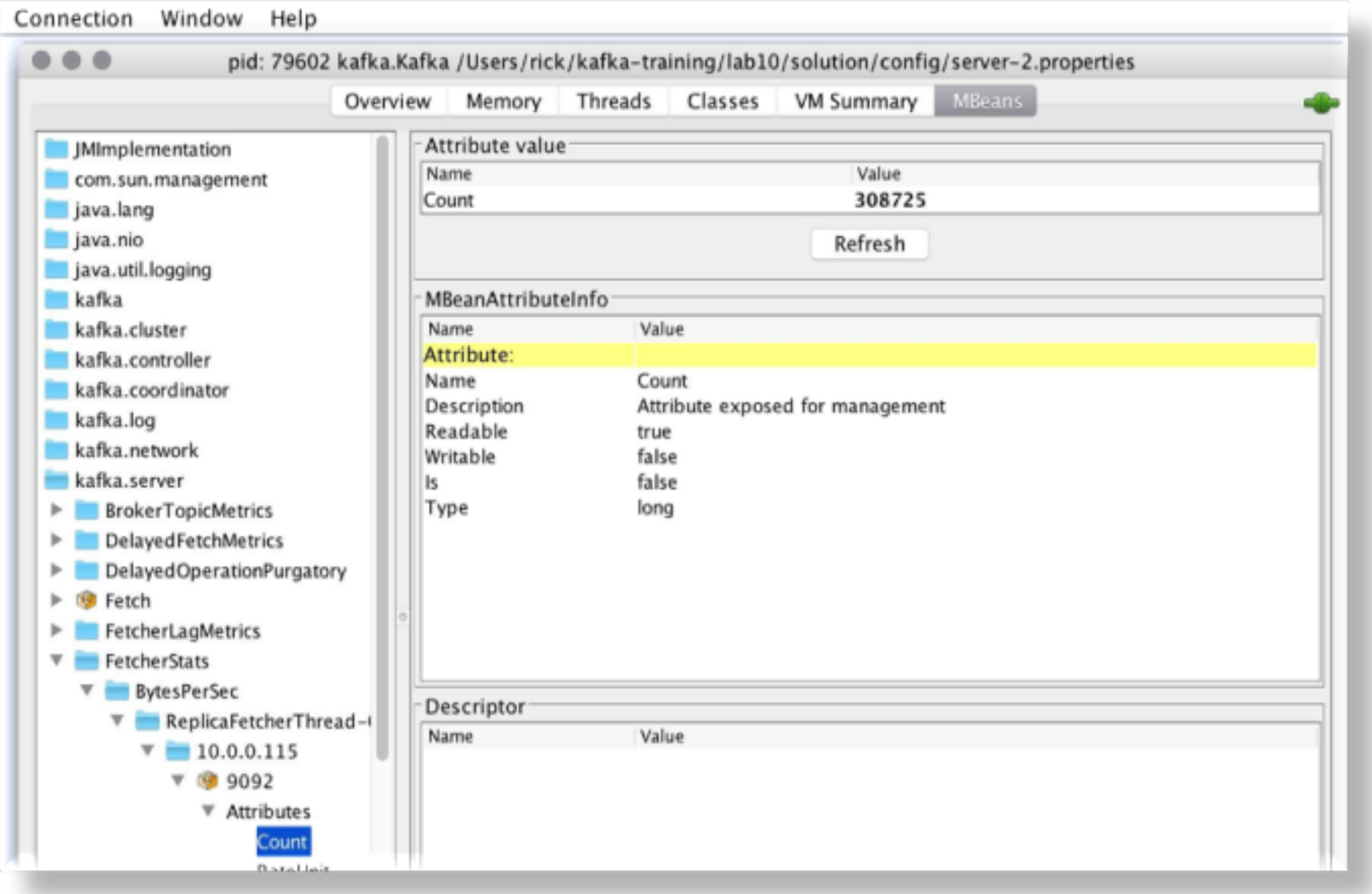
Tos

Metric	Description
fetch-size-avg	Average byte size fetched per request for specific topic
fetch-size-max	Max byte size fetched per request for specific topic
bytes-consumed-rate	Average byte size consumed per second for specific topic
records-per-request-avg	Average record count per request for specific topic
records-consumed-rate	Average record count consumed per second for specific topic

Other Metrics

- Low level metrics
- Thread metrics
- Task Metrics
- Processor Node Metrics
  - Forwarding to other nodes
- State Store Metrics
- Good idea to monitor GC, JVM threads, etc.
- See metrics available with JConsole

Kafka Broker Metrics via JConsole 1 of 2



pid: 79602 kafka.Kafka /Users/rick/kafka-training/lab10/solution/config/server-2.properties

Attribute value

Name	Value
Count	308725

MBeanAttributeInfo

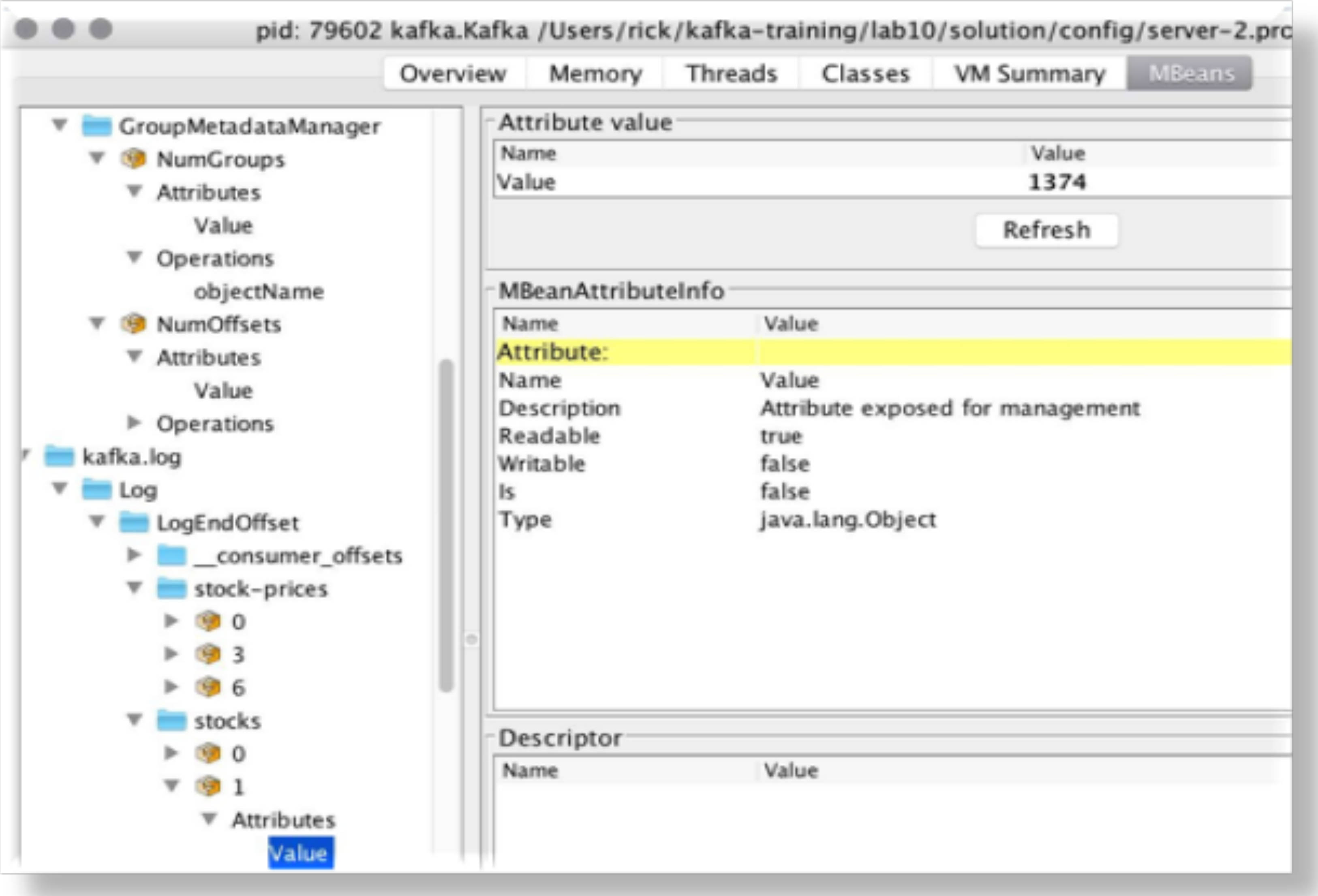
Name	Value
Attribute:	
Name	Count
Description	Attribute exposed for management
Readable	true
Writable	false
Is	false
Type	long

Descriptor

Name	Value

Kafka Broker JConsole Metrics 2 of 2

Tos



pid: 79602 kafka.Kafka /Users/rick/kafka-training/lab10/solution/config/server-2.pro

Attribute value

Name	Value
Value	1374

MBeanAttributeInfo

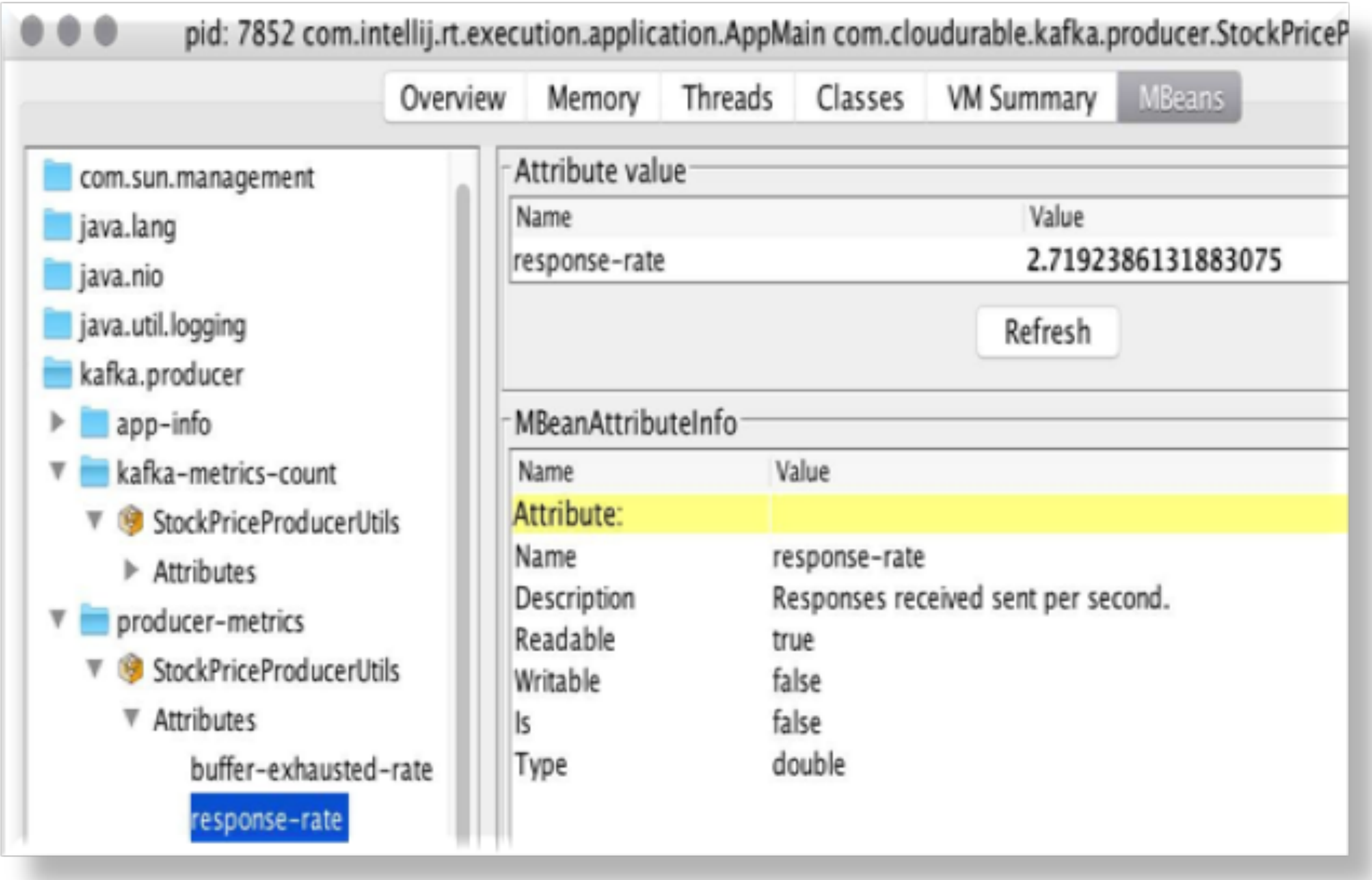
Name	Value
Attribute:	
Name	Value
Description	Attribute exposed for management
Readable	true
Writable	false
Is	false
Type	java.lang.Object

Descriptor

Name	Value

Kafka Producer Metrics JConsole

Tos



pid: 7852 com.intellij.rt.execution.application.AppMain com.cloudurable.kafka.producer.StockPriceP

- com.sun.management
- java.lang
- java.nio
- java.util.logging
- kafka.producer
- app-info
- kafka-metrics-count
  - StockPriceProducerUtils
    - Attributes
- producer-metrics
  - StockPriceProducerUtils
    - Attributes
      - buffer-exhausted-rate
      - response-rate

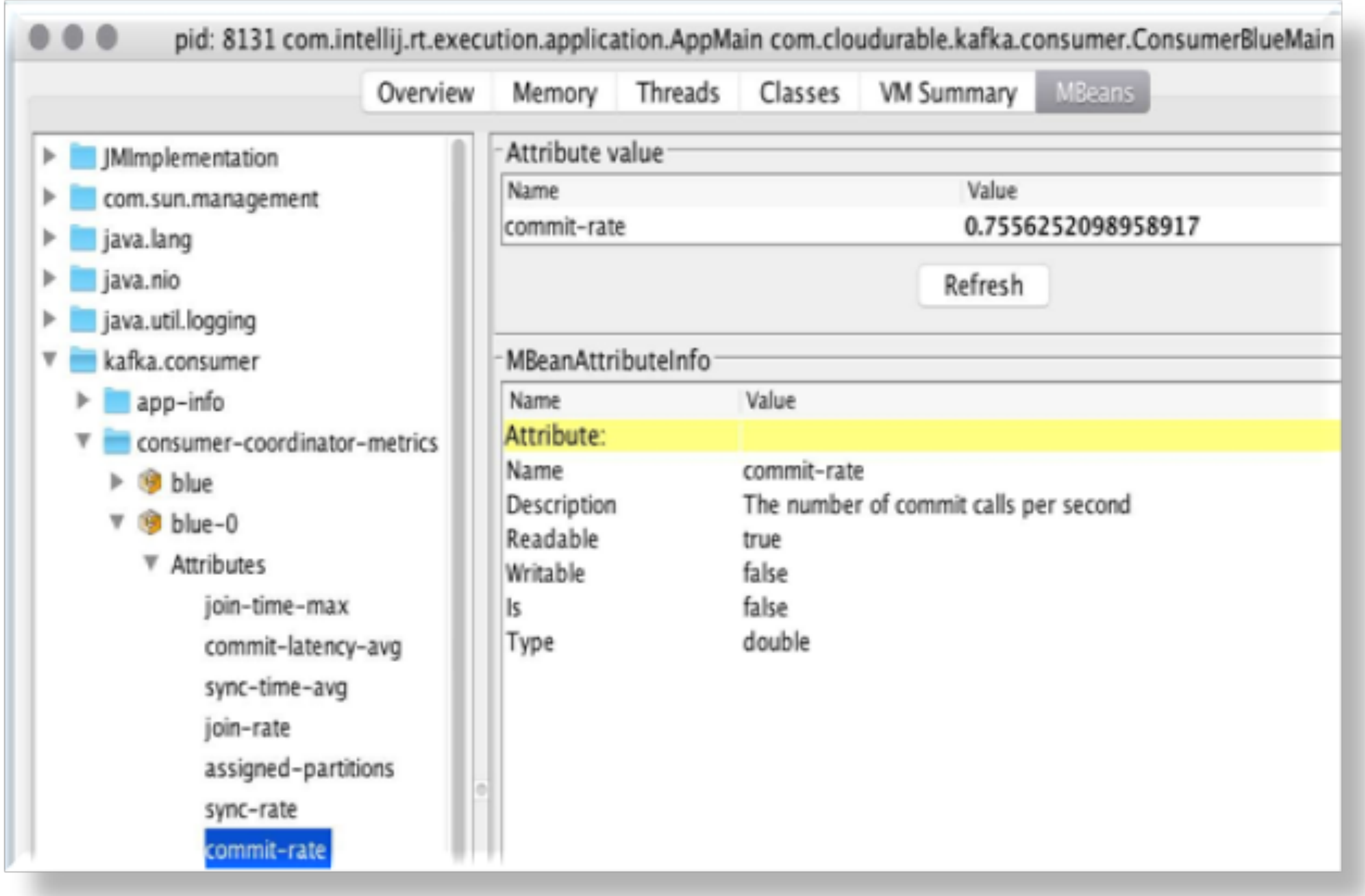
Attribute value

Name	Value
response-rate	2.7192386131883075

MBeanAttributeInfo

Name	Value
Attribute:	
Name	response-rate
Description	Responses received sent per second.
Readable	true
Writable	false
Is	false
Type	double

Kafka Consumer JConsole Metrics



pid: 8131 com.intellij.rt.execution.application.AppMain com.cloudurable.kafka.consumer.ConsumerBlueMain

- JMImplementation
- com.sun.management
- java.lang
- java.nio
- java.util.logging
- kafka.consumer
  - app-info
  - consumer-coordinator-metrics
    - blue
    - blue-0
      - Attributes
        - join-time-max
        - commit-latency-avg
        - sync-time-avg
        - join-rate
        - assigned-partitions
        - sync-rate
        - commit-rate

Attribute value

Name	Value
commit-rate	0.7556252098958917

MBeanAttributeInfo

Name	Value
Attribute:	
Name	commit-rate
Description	The number of commit calls per second
Readable	true
Writable	false
Is	false
Type	double

ZooKeeper Setup 1 of 3

Tos

- Don’t put all ZooKeeper nodes in same same rack or in a single AWS availability Zones
- Decent hardware; don’t use T2 Micro
- Use 5 to 7 servers for production tolerates 2 to 3 servers down
- For small deployment using 3 servers is ok (only 1 allowed down)
- Put transaction logs on dedicated disk group (***dataLogDir***)
- Put snapshots, message log, and OS on another disk/disk group (***dataDir***)
- Writes to transaction log are synchronous batches
  - Concurrent writes can significantly affect performance

ZooKeeper Setup 2 of 3

- Use dedicated ZooKeeper cluster for Kafka
- ZooKeeper needs 3 to 5GB of heap with some room for OS (30% to 50% of System total)
- Monitoring ZooKeeper use JMX and or 4 letter words
- Keep ZooKeeper cluster small
  - Reduce quorums on the writes and subsequent cluster member updates
  - But don't go too small either
  - More ZooKeeper servers increases read capacity of ZooKeeper

ZooKeeper Setup 3 of 3

- ZooKeeper requires little administration, but…
- ZooKeeper takes periodic snapshots of its data
  - snapshot plus log can rebuild ZooKeeper state
- ZooKeeper does not purge snapshots by default
  - Let's you back up snapshots
- You want to purge snapshots so disk does not fill up
  - ***autopurge.snapRetainCount*** *(*how many snapshots to keep)
  - ***autopurge.purgeInterval*** (duration in hours)
- Make sure you use rolling log files for logging

Tos

Lab : Monitoring, Performance & Tuning