IN PURSUIT OF MESSAGING BROKER(S)

eHarmony®

DAVID GEVORKYAN

@davidgev

davidgevorkyan

eHarmony®

PRIMARY GOAL

- STABLE
- ACTIVE & HAS PROPER SUPPORT
- SATISFIES PERFORMANCE REQUIREMENTS
  - OUR PEAK LOAD PER NODE 1000 MSG / SEC, 60 MLN / DAY
  - MUS MESSAGE SIZE ~4KB
  - SINGLES PS MESSAGE SIZE ~2.2KB
  - MDS MESSAGE SIZE ~615KB
- BETTER THAN

eHarmony®

RabbitMQ

NSQ
BILLIONS AND BILLIONS OF MESSAGES SERVED

kafka

ØMQ

ActiveMQ

Apollo

RestMQ
Redis based message queue

twitter / kestrel

NATS

akka

eHarmony®

Comparison of Different Messaging Brokers

Feature Name	Apache Kafka	ActiveMQ	Apollo	RabbitMQ	NSQ	NATS	RestMQ / Redis	Kestrel by Twitter	ZeroMQ	Akka
Brokered	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗
Open Source	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Commercial Support	Confluent	Kind of	✗	Pivotal	✗	Apcera	✗	✗	Kind Of	Typesafe
Large Community	✓	✓	✗	✓	✓	✗	✗	✗	✗	✓
Management UI	✓	✓	✗	✓	✓	✗	✗	✗	✗	Cloudy Akka
Multi-protocol	Binary Protocol over TCP, WebSocket Interface	AMQP, STOMP	AMQP, STOMP, MQTT, Openwire, WebSockets	AMQP, STOMP, MQTT, WebSockets, JSON-RPC	Binary Protocol over TCP / data format agnostic	-	REST / Comet / WebSockets over HTTP	Text protocol / memcache / thrift	Unbounded TCP Sockets	-
JMS Compliant	✗	✓	✓	✓	✗	✗	✗	✗	✗	✓
Metric Exposure (maybe through JMX)	✓	✓	✓	HTTP API	HTTP API	-	HTTP API	-	-	✓
Persistence / Durability	✓	KahaDB	✓	✓	✓	✗	✓	✓	Memory Swap	✓
Guaranteed Delivery	Kind Of	✓	✓	✓	✓	✗	-	✓	✗	✓
Guaranteed Order of Delivery	Order Within Partition	✓	✓	✓	✗	✗	-	Order Per Machine	-	✓
Delivery Acknowledgements	✓	✓	✓	✓	✓	✗	-	✓	✓	✓
Tracing	✗	✗	✗	✓	✗	✗	✗	✗	✗	✗
Variety of Driver Support	✓	✓	✓	✓	✓	✓	✓	✗	✓	✗
Item Expiration	✓	✓ + DLQ support	✓	✓ + DLQ support	✗	✗	-	✓	✗	DLQ
High Availability	✓	✓	through LevelDB	✓	✓	✓	✗	✓	-	-
Hadoop Ecosystem	✓	HBase persistence	sync LevelDB to HDFS	Flume-to-HDFS	✗	✗	✗	Spark Kestrel	✗	✗
Virtualization Friendly	AWS	AWS, OpenShift, Azure	✗	AWS, CloudAmqp, Google Compute Engine	AWS	Cloud Foundry	✗	✗	AWS	AWS, Google Compute Engine
Project Age	2010	2005	2012	2009	2012	2014	2010	2008	2007	2009
Active Development	✓	✓	✓	✓	✓	✗	✗	✗	✗	✓
Bad experiences	-	-	-	-	-	-	-	✓	-	-

eHarmony®

21 PARTICIPANTS
7 FEATURES EACH

Matching / ... / Matching Engineering Projects

Prioritizing Messaging Broker Features

Edit Watch Sh

Created and last modified by David Gevorkyan on Apr 30, 2015

Please choose any seven features that you think are important for your ideal Messaging Broker.

Note that once we identify all important features, we will narrow down the list of messaging brokers that we are looking at and perform thorough benchmark tests to compare their throughput and latency characteristics.

CSV Export | Voter Column

Feature Name	Description	Result	Your Vote
High Availability / Replication	Queues can be mirrored across several machines in a cluster, ensuring that even in the event of hardware failure your messages are safe.	20	☑
Persistence / Durability	Persist messages to disk, database or any other means to support survival of restarts and failures.	19	☑
Developer friendly	Having a good developer community support (documentation, forums) is always important when something "goes south".	16	☑
Brokered	No application is speaking directly to the other application. All the communication is passed through the broker.	15	☑
Guaranteed Delivery & Delivery Acknowledgements	*At least once* — Messages are never lost but may be redelivered (this is bare minimum requirement if you choose this item). Use of acknowledgements guarantees *at-least-once* delivery. Without acknowledgements, message loss is possible during publish and consume operations and only *at-most-once* delivery is guaranteed.	15	☐
Easy to configure	Deployment and configuration should be very simple, we do not want something where it is rocket science to make it usable.	11	☐
Stable Enough for Production	It sometimes gives peace of mind when you know that project has been around for a while and some large companies are using it in production.	10	☐
Metric Exposure	It is useful to know different stats being exposed by the Messaging Broker, such as MessageCount, ConsumerCount, TotalMessagesAdded. Most common way would be through JMX.	7	☐
Management UI	To have some kind of management console, preferably web based, for monitoring health of the cluster. e.g. which nodes are up and which topics they are hosting.	5	☐
Hadoop Ecosystem	Out of the box integration of different Hadoop ecosystem components (HDFS, Flume, Spark, Storm, Hive, Presto). Note that most of the time, messaging brokers support data loading into HDFS, which is good enough to support others.	5	☑
Open Source	Some people enjoy going into the code to understand specific behavior.	4	☐
WebSocket Support	WebSockets are a cool new HTML5 technology which allows you to asynchronously send and receive messages.	4	☐

eHarmony®

VERY IMPORTANT FEATURES

VOTED BY AT LEAST 30%

- HIGH AVAILABILITY / REPLICATION
- PERSISTENCE / DURABILITY
- DEVELOPER FRIENDLY
- BROKERED
- GUARANTEED DELIVERY & DELIVERY ACKNOWLEDGEMENTS
- EASY TO CONFIGURE
- STABLE ENOUGH FOR PRODUCTION
- METRIC EXPOSURE

LESS IMPORTANT FEATURES

VOTED BY AT LEAST 5%

- MANAGEMENT UI
- HADOOP ECOSYSTEM
- OPEN SOURCE
- WEBSOCKET SUPPORT
- ITEM EXPIRATION
- OPERATIONS FRIENDLY
- VARIETY OF DRIVER SUPPORT
- JMS COMPLIANT
- ACTIVE DEVELOPMENT / CUTTING EDGE

eHarmony®

NOT IMPORTANT FEATURES

VOTED BY LESS THAN 5%

- BROKERLESS (PEER-TO-PEER)
- GUARANTEED ORDER OF DELIVERY
- TRACING
- DEAD LETTER EXCHANGE SUPPORT
- VIRTUALIZATION FRIENDLY

eHarmony®

RabbitMQ

NSQ
BILLIONS AND BILLIONS OF MESSAGES SERVED

kafka

ActiveMQ

after considering "Very Important Features"

eHarmony®

RabbitMQ

NO OFFICIAL
JAVA DRIVER

kafka

ActiveMQ

eHarmony®

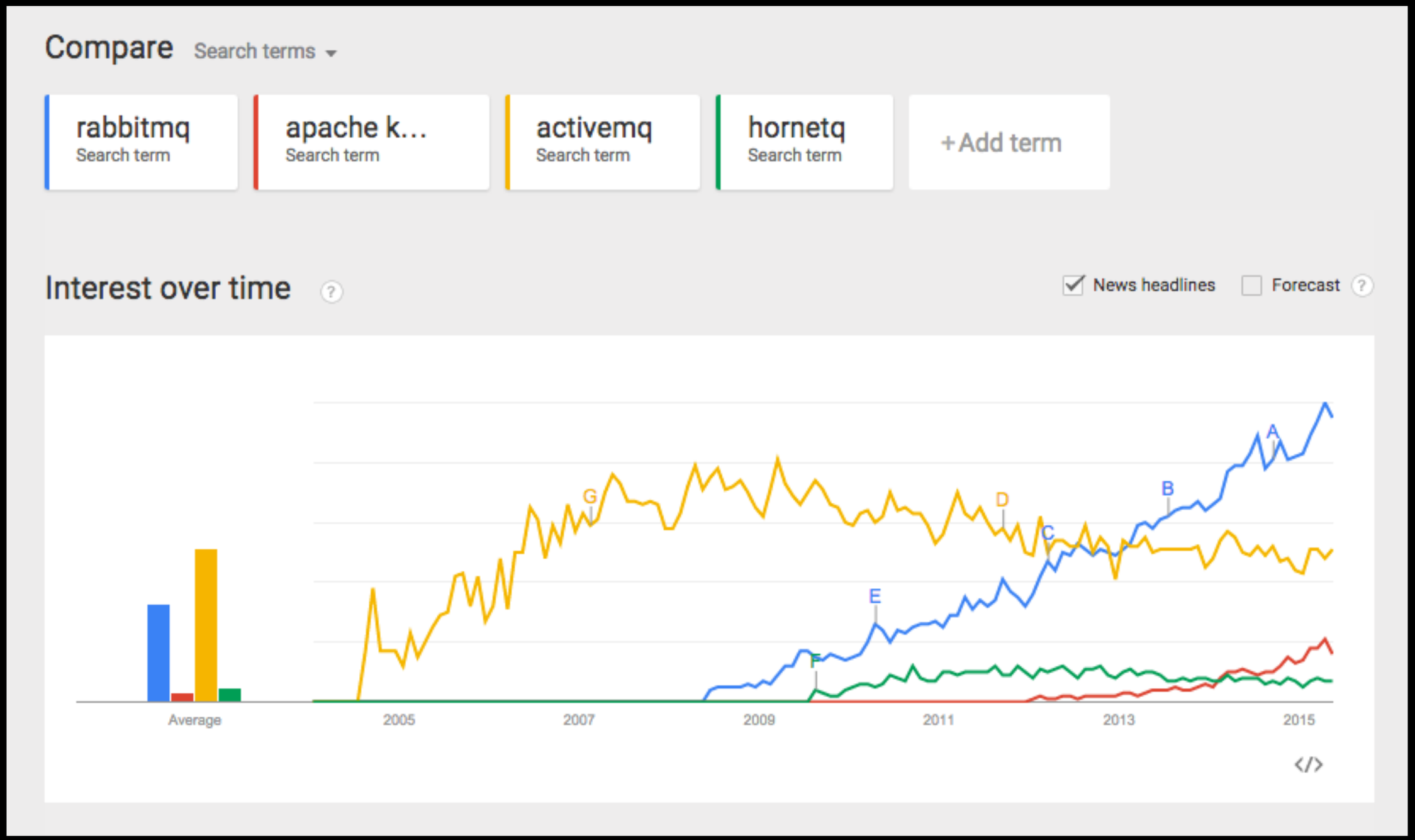
RabbitMQ

NO OFFICIAL
JAVA DRIVER

kafka

RISK OF BEING
REPLACED BY APOLLO

eHarmony®


Compare
Search terms
rabbitmq
Search term
apache k...
Search term
activemq
Search term
hornetq
Search term
+ Add term
Interest over time
News headlines
Forecast
A
B
C
D
E
F
G
Average
2005
2007
2009
2011
2013
2015

eHarmony®

RabbitMQ

kafka

eHarmony®

RabbitMQ

RABBITMQ IS AN EFFICIENT, HIGHLY SCALABLE, AND EASY-TO-DEPLOY QUEUING SYSTEM THAT MAKES HANDLING MESSAGE TRAFFIC VIRTUALLY EFFORTLESS

Source: http://stackshare.io and others

eHarmony®

kafka

KAFKA IS A DISTRIBUTED, PARTITIONED, REPLICATED COMMIT LOG SERVICE. IT PROVIDES THE FUNCTIONALITY OF A MESSAGING SYSTEM, WITH A UNIQUE DESIGN

Source: http://stackshare.io and others

eHarmony®

BENCHMARK.IO

eHarmony®

ROCKET LAUNCHER

```
usage: benchmarkio.controlcenter.LaunchRocket
    --benchmark-type <arg>              benchmark type, one of
                                        PRODUCER_ONLY | CONSUMER_ONLY |
                                        PRODUCER_AND_CONSUMER |
                                        PRODUCER_NO_CONSUMER_THEN_CONSUMER
    --broker-type                       broker type => KAFKA | RABBITMQ |
                                        ACTIVEMQ
    --durable <arg>                     boolean value, indicates whether we
                                        should test with durability
    --host <arg>                        host of the broker
    --kafka-producer-type <arg>         This parameter specifies whether
                                        the messages are sent
                                        asynchronously in a background
                                        thread. Valid values are (1) async
                                        for asynchronous send and (2) sync
                                        for synchronous send. By setting
                                        the producer to async we allow
                                        batching together of requests
                                        (which is great for throughput) but
                                        open the possibility of a failure
                                        of the client machine dropping
                                        unsent data.
    --msg-size-in-kb <arg>              message size in KB
    --num-consumers <arg>               consumer count
    --num-producers <arg>               producer count
    --port <arg>                        port of the broker
    --total-number-of-messages <arg>    total number of messages
-u,--usage                              show usage
    --zookeeper <arg>                   zookeeperHost:zookeeperPort
```

eHarmony®

BENCHMARKS CAN PREDICT LOAD UNDER CERTAIN CONDITIONS, BUT ARE FAR FROM BEING PERFECT.

THE REAL PREDICTOR IS RUNNING IN SHADOW MODE

eHarmony®

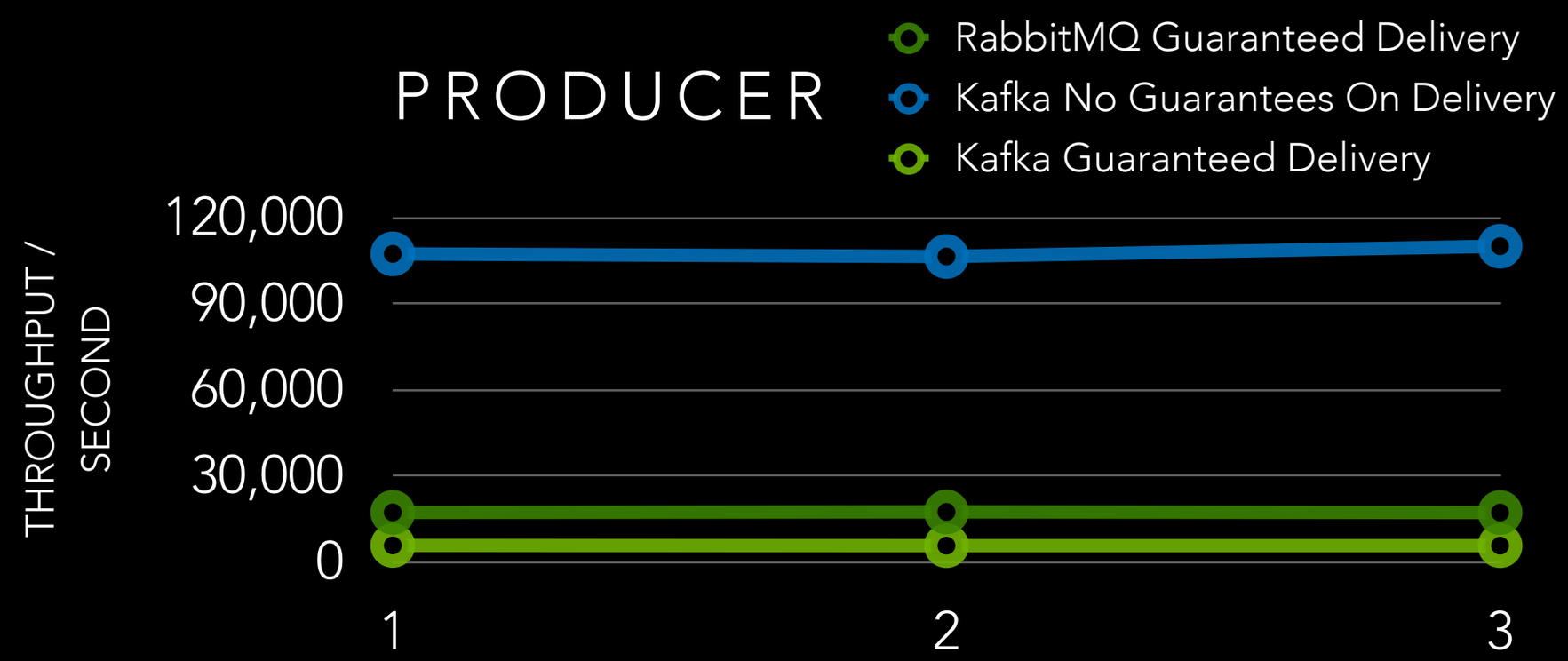
MACHINE DETAILS

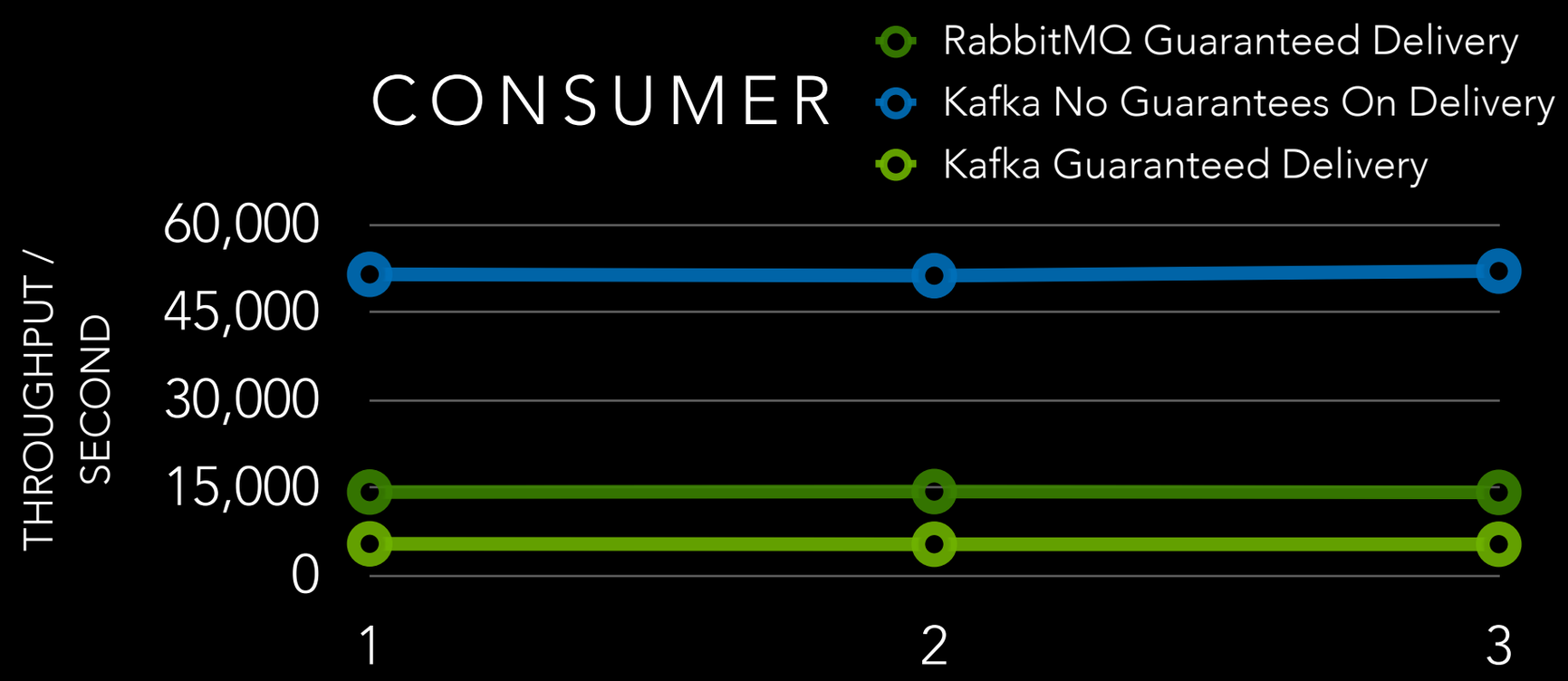
r3.2xlarge - *Optimized for memory-intensive applications*
8 cores - *Intel Xeon E5-2670 v2 (Ivy Bridge) Processors*
61GB of RAM
160GB SSD Memory

Ubuntu: 14.04 64bit AMD
Java: 1.7.0_79 64bit
RabbitMQ: 3.5.2
Apache Kafka: 2.9.2-0.8.2.1

eHarmony®

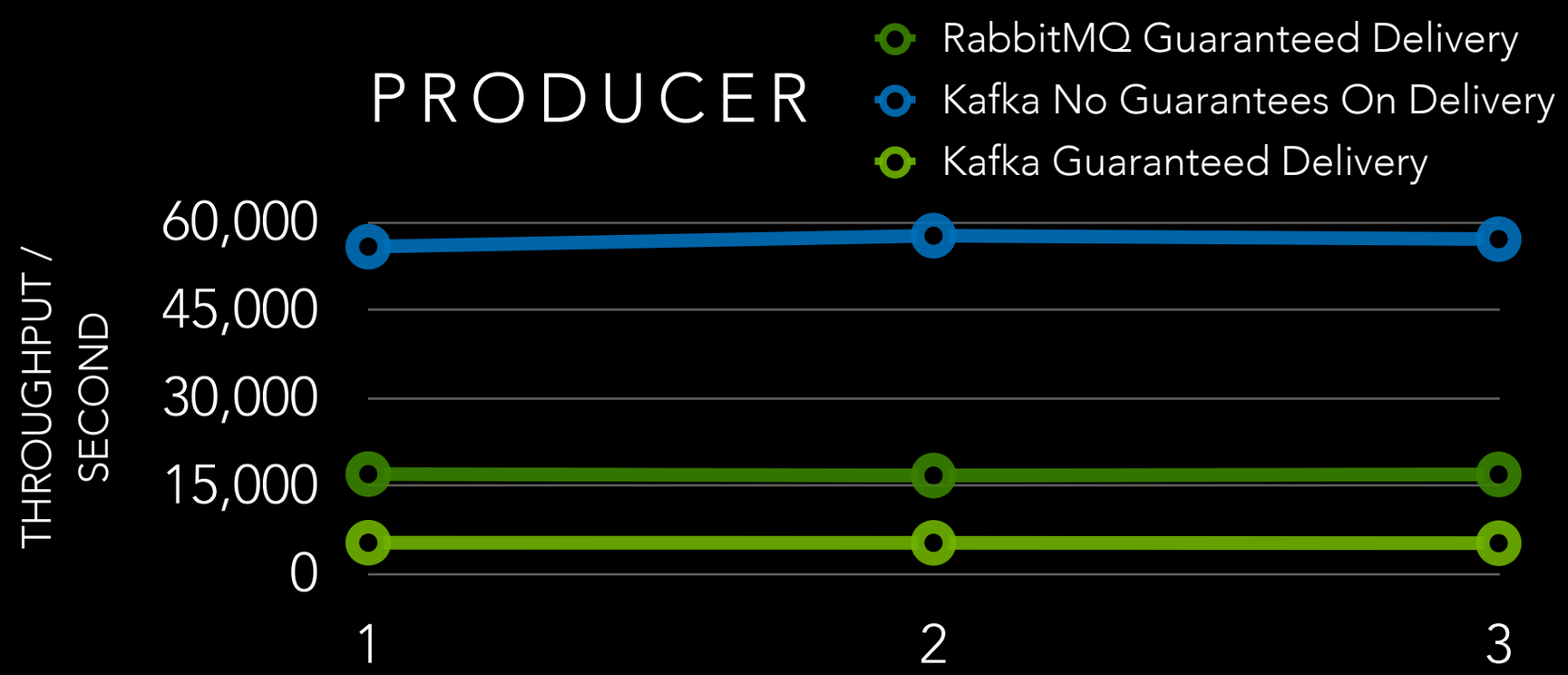
1P / 1C / 1KB / TOPIC

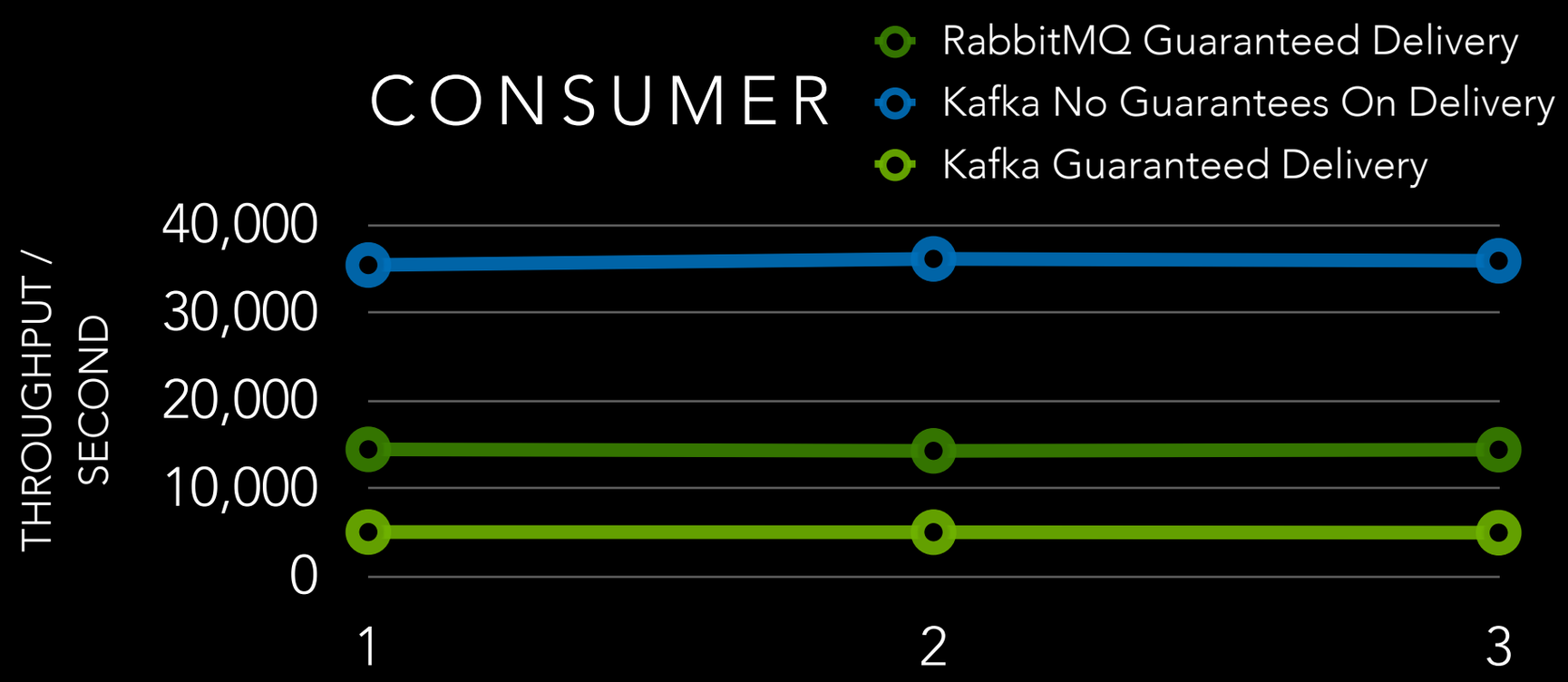




eHarmony®

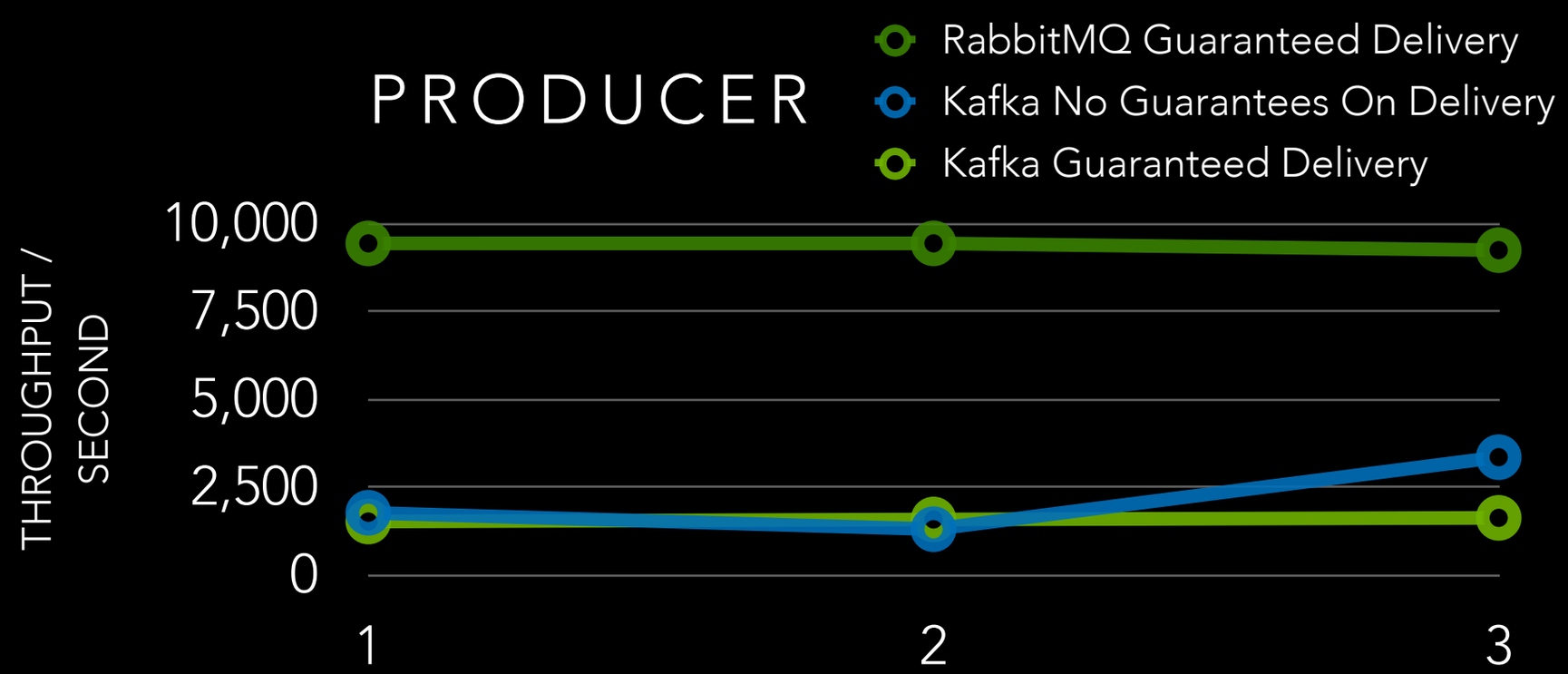
1P / 1C / 4KB / TOPIC

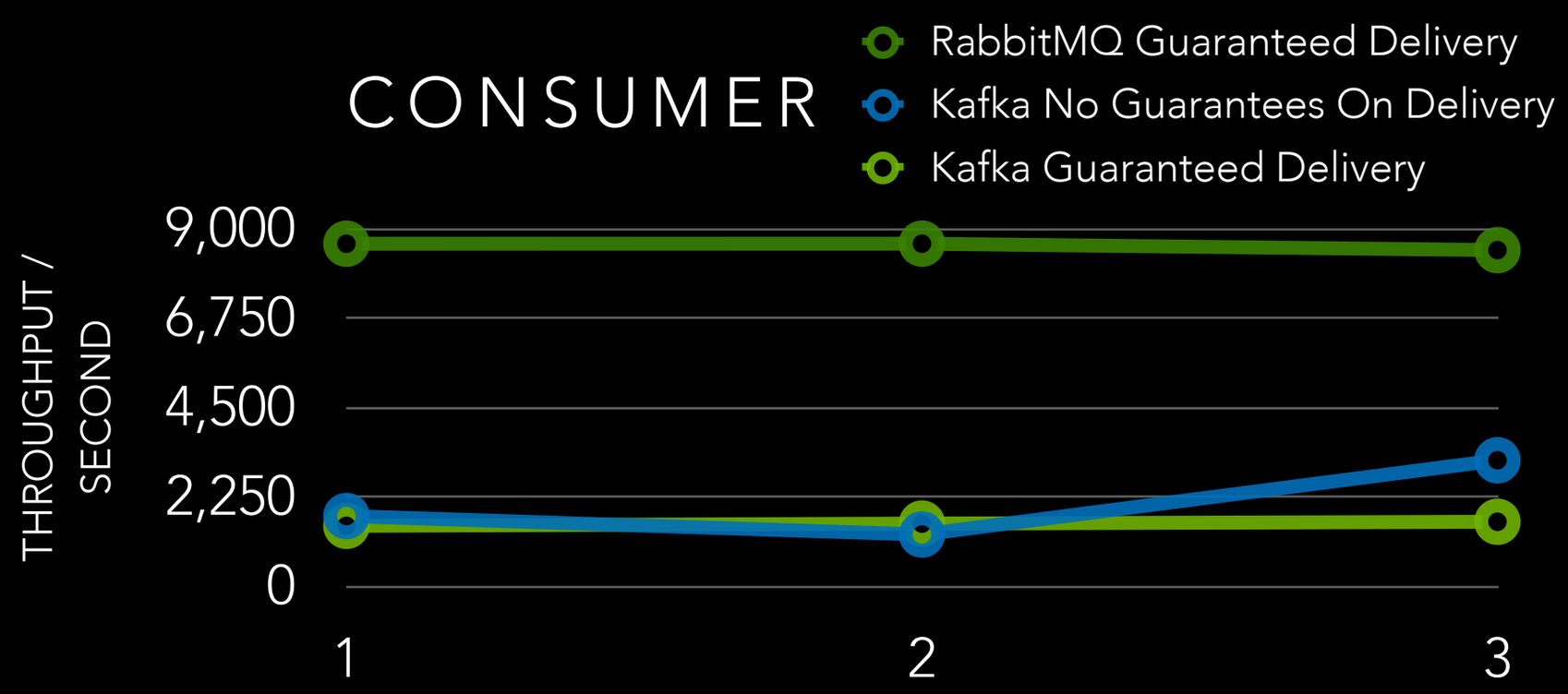




eHarmony®

1P / 1C / 100KB / TOPIC

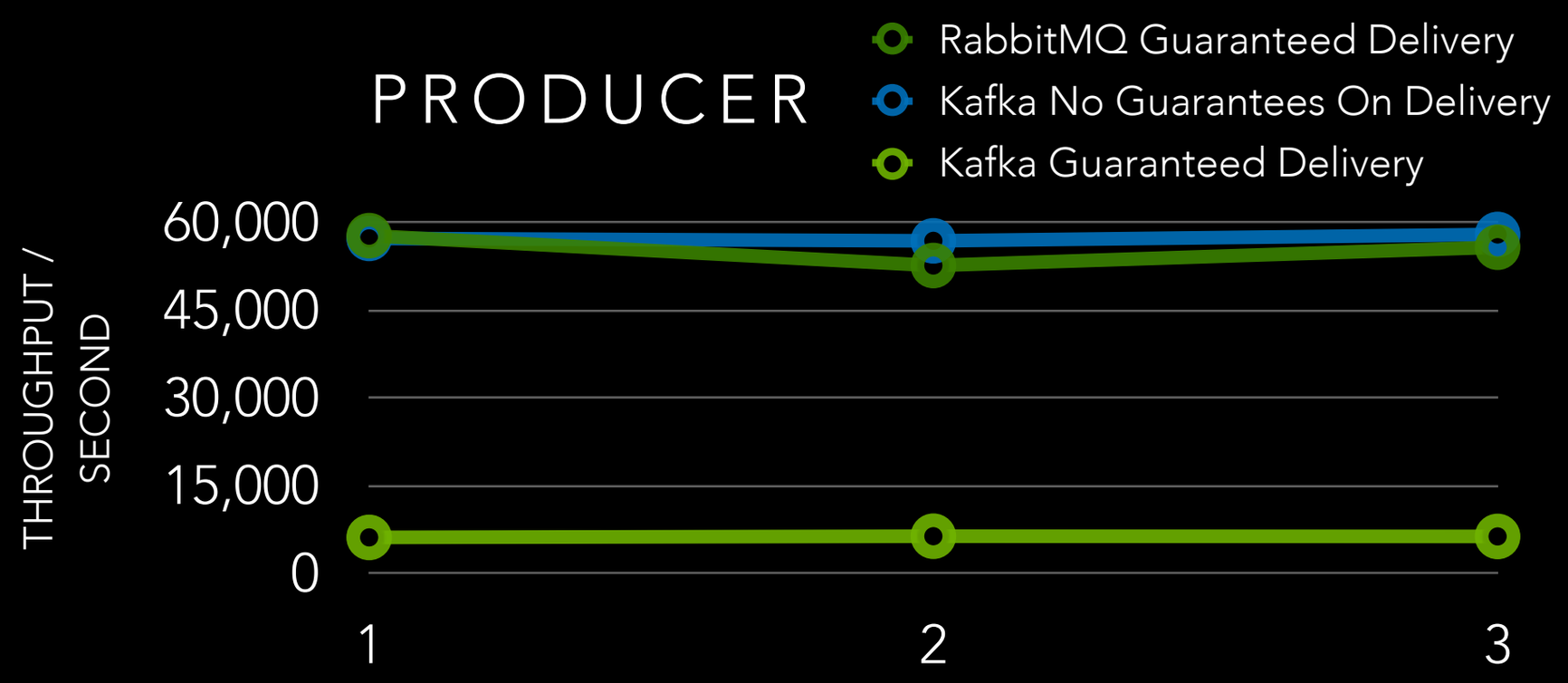




eHarmony®

1P / 0C / 4KB / TOPIC

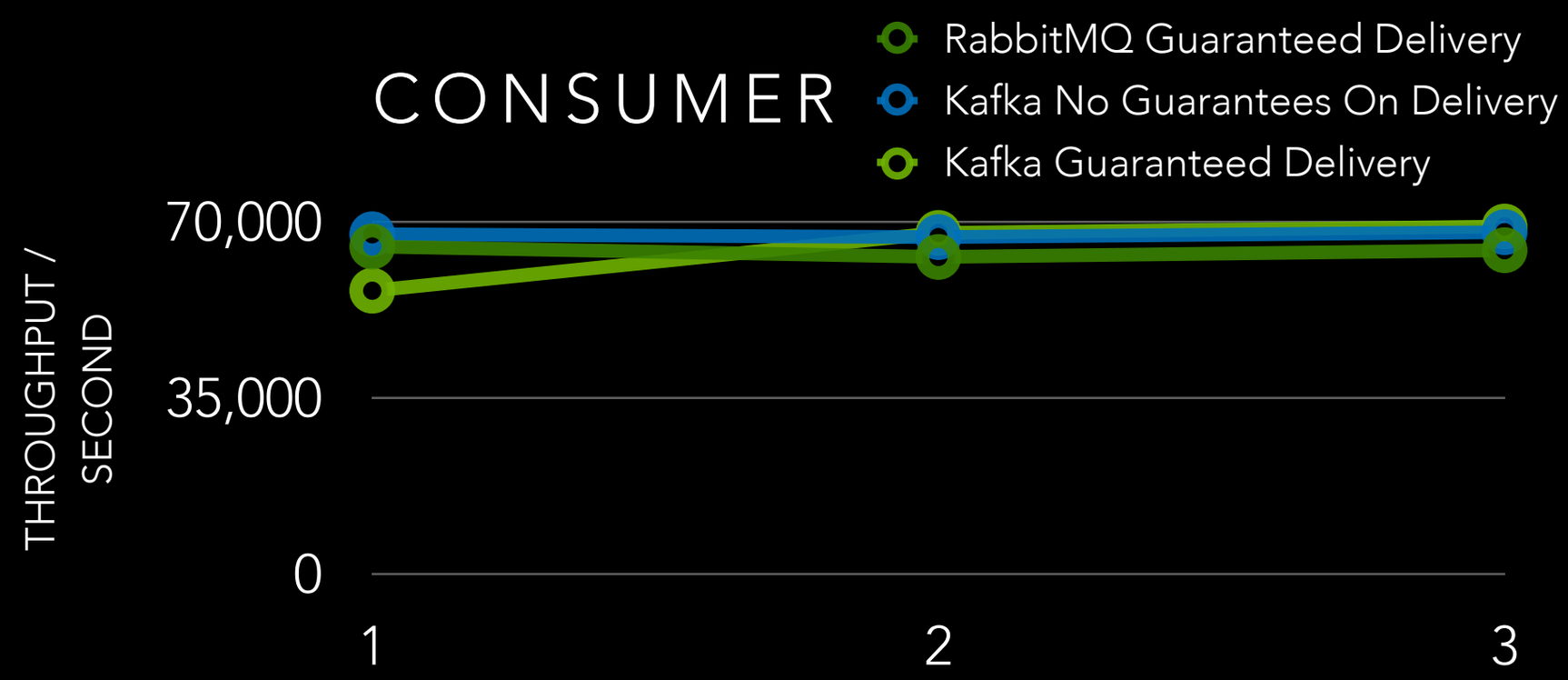
LOAD



eHarmony®

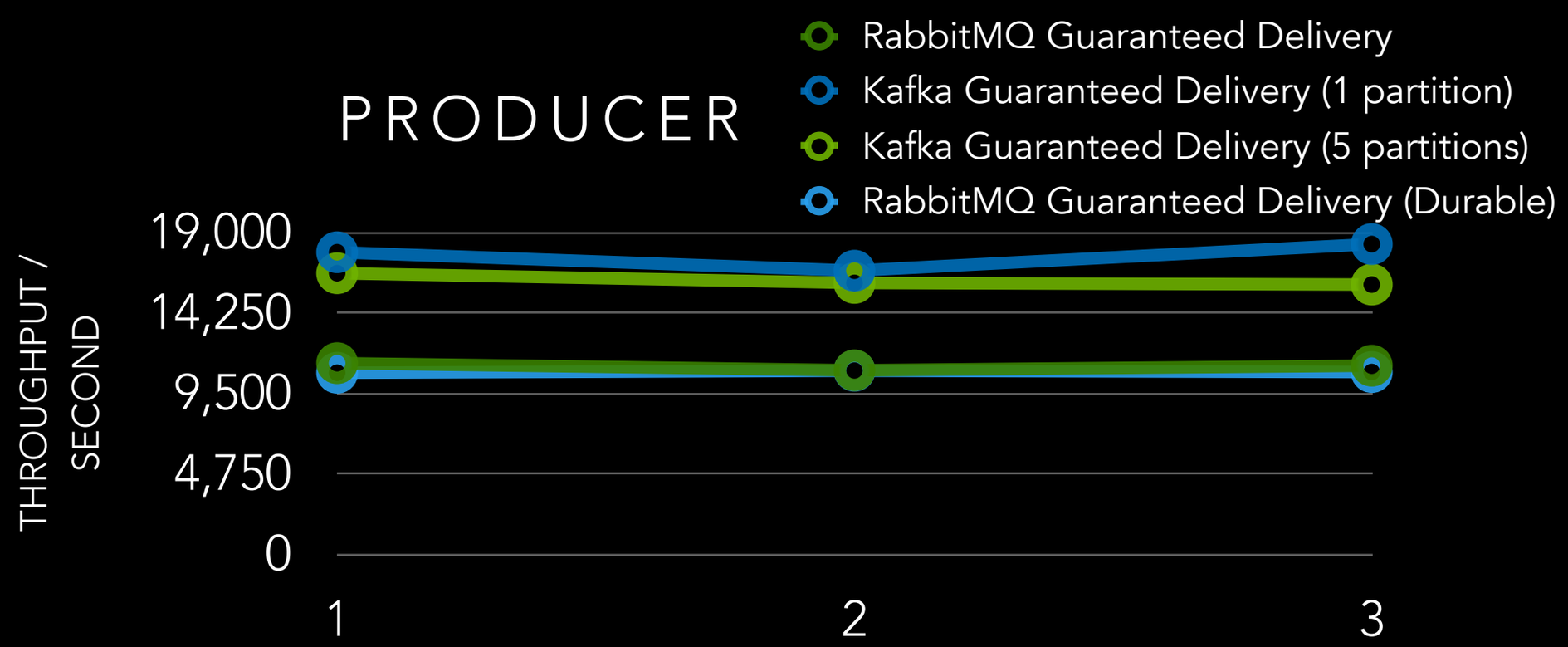
0P / 1C / 4KB / TOPIC

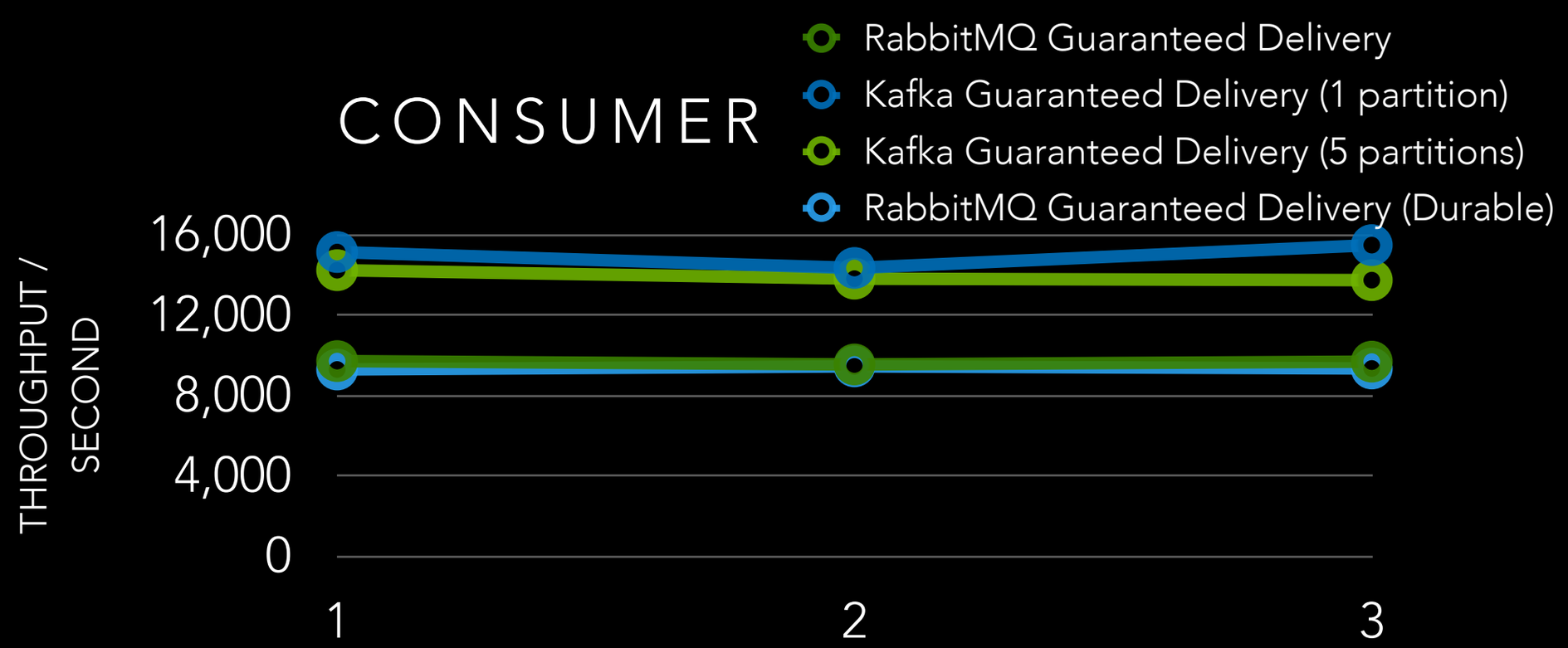
UNLOAD



eHarmony®

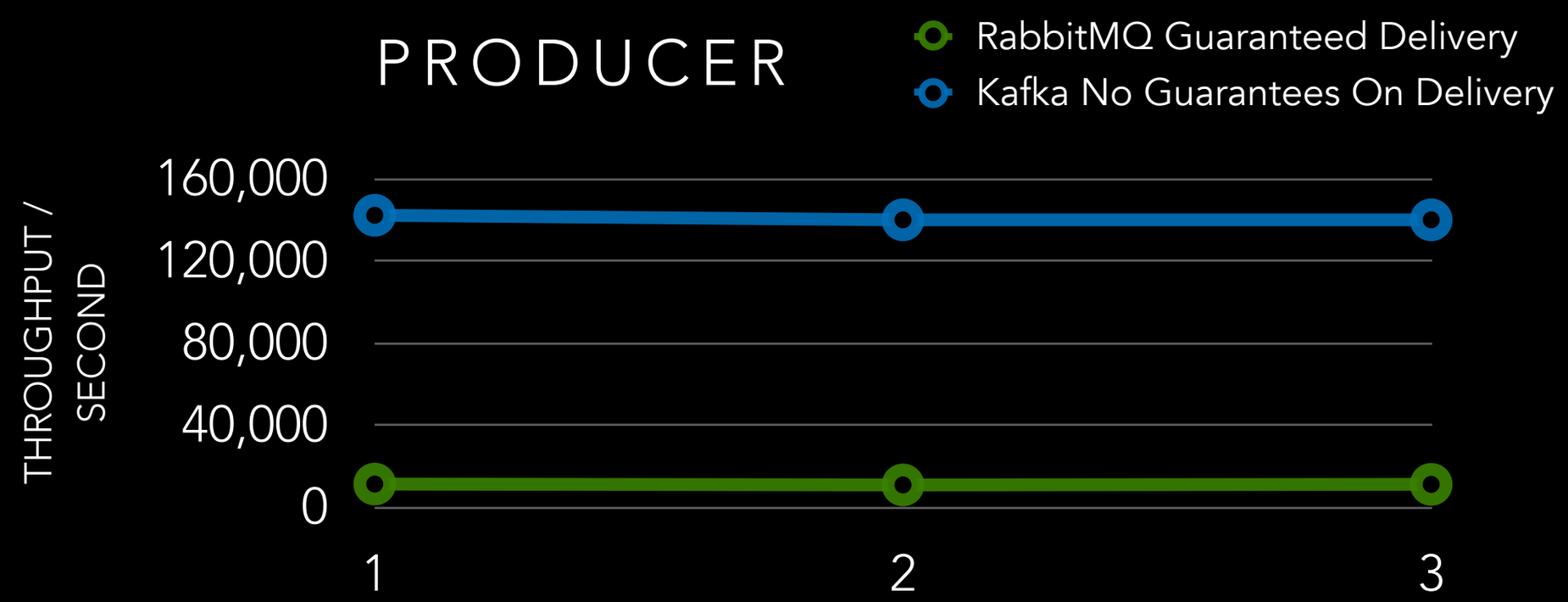
5P / 5C / 4KB / TOPIC

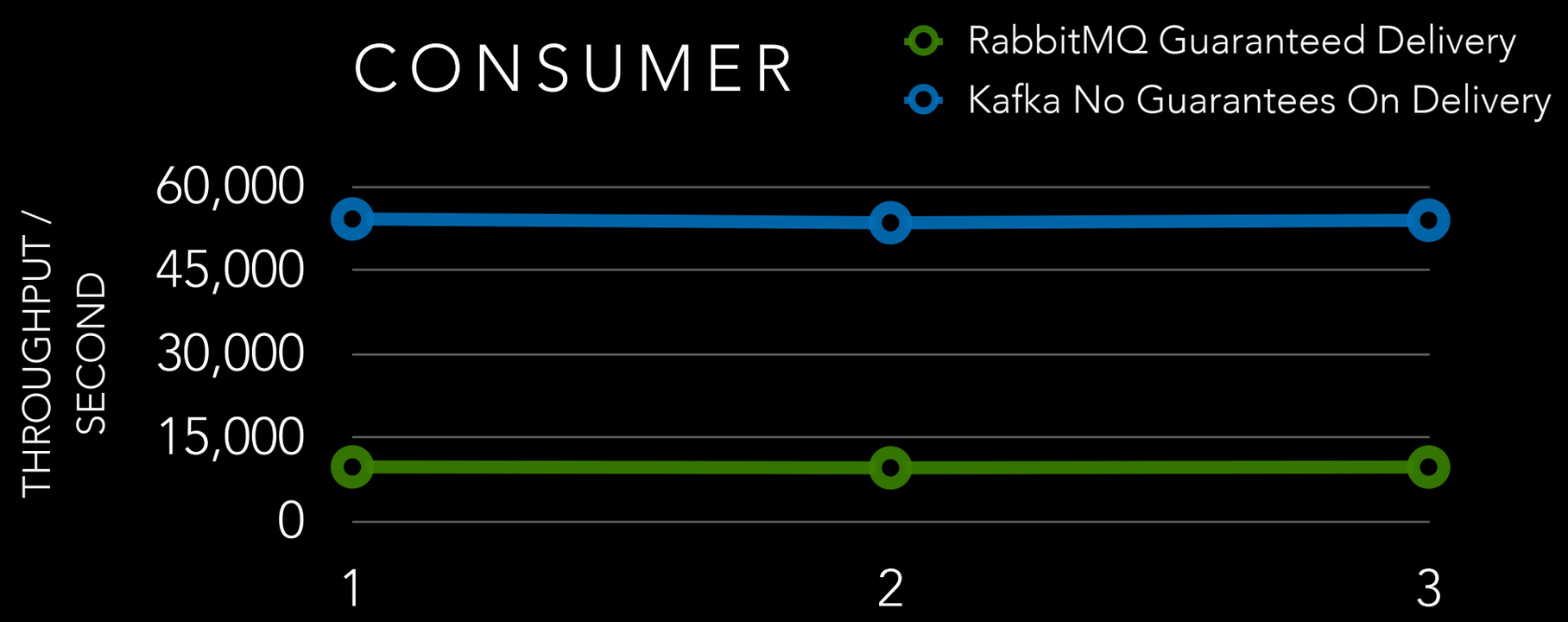




eHarmony®

5P / 5C / 4KB / TOPIC





eHarmony®

LATENCY FOR BOTH

MEAN < 1MS

MAX < 40MS

eHarmony®

RabbitMQ Management

localhost:15672/#/

RabbitMQ

User: guest
RabbitMQ 3.0.0, Erlang R15B02
Log out

Overview | Connections | Channels | Exchanges | Queues | Admin

Overview

Totals

Queued messages (?)

Ready	Unacknowledged	Total
51581	0	51581
-115 msg/s		-115 msg/s

Message rates (?)

Publish	Deliver (noack)
5135 msg/s	5281 msg/s

Global counts (?)

Connections: 3 | Channels: 3 | Exchanges: 9 | Queues: 2 | Consumers: 1

Nodes

Name	File descriptors (?)	Socket descriptors (?)	Erlang processes	Memory	Disk space	Uptime	Type
rabbit@Thomas-Bayers-MacBook	36 256 available	4 138 available	225 1048576 available	212.2MB 3.0GB high watermark	86.9GB 953.7MB low watermark	29m 21s	Disc Stats *

eHarmony®

Queued messages (chart: last ten minutes) (?)

Ready	0
Unacked	0
Total	0

Message rates (chart: last ten minutes) (?)

Publish	15,408/s
Deliver (noack)	15,392/s

Global counts (?)

Connections: 2 | Channels: 2 | Exchanges: 9 | Queues: 1 | Consumers: 1

Node

Node: rabbit@uniavatar **(More about this node)**

File descriptors (?)	Socket descriptors (?)	Erlang processes	Memory	Disk space	Uptime	Rates mode	Info	+/-
33 256 available	3 138 available	211 1048576 available	49MB 5.9GB high watermark	98GB 48MB low watermark	1m 40s	basic	Disc 1 Stats	

eHarmony®

RabbitMQ

User: queue
RabbitMQ 3.2.3, Erlang R15B01
Log out

Overview | Connections | Channels | Exchanges | Queues | Admin

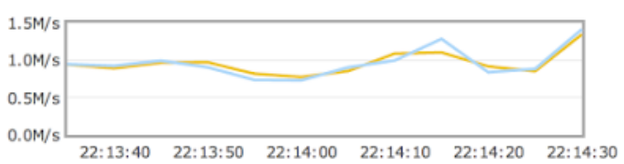
Overview

Totals

Queued messages (chart: last minute) (?)

Ready	2,343 msg
Unacknowledged	0 msg
Total	2,343 msg

Message rates (chart: last minute) (?)



Publish	1,345,531/s
Deliver (noack)	1,413,840/s

Global counts (?)

Connections: 12690 | Channels: 12690 | Exchanges: 194 | Queues: 186 | Consumers: 10304

Nodes

Name	File descriptors (?)	Socket descriptors (?)	Erlang processes	Memory	Disk space	Uptime	Type
rabbit@b-rabbitmq-queue-1te2	445 262144 available	395 235837 available	4594 1048576 available	252MB 12GB high watermark	6.2GB 48MB low watermark	1h 33m	RAM

eHarmony®

Kafka Manager dev Cluster Brokers Topic Preferred Replica Election Reassign Partitions

Clusters / dev / Topics / content

content

Topic Summary

Replication	1
Number of Partitions	4
Total number of Brokers	1
Number of Brokers for Topic	1
Preferred Replicas %	100
Brokers Skewed %	0
Brokers Spread %	100
Under-replicated %	0

Operations

Generate Partition Assignments | Reassign Partitions | Delete Topic

Partitions by Broker

Broker	# of Partitions	Partitions	Skewed?
0	4	(2,1,3,0)	false

Partition Information

Partition	Leader	Replicas	In Sync Replicas	Preferred Leader?	Under Replicated?
0	0	(0)	(0)	true	false
1	0	(0)	(0)	true	false

eHarmony®


Kafka Web Console
Zookeepers
Brokers
Topics
Settings
Offset
Lag
117576
100000
80000
60000
40000
20000
0
Messages
22:17:09
Offset 117576
Lag 26478
22:11:48
22:12:30
22:13:20
22:14:10
22:15:00
22:15:50
22:16:40
22:17:30
22:18:20
22:19:09
Time
Consumer
Producer/s
115
100
80
60
40
20
0
Messages per second
22:11:58
22:12:30
22:13:20
22:14:10
22:15:00
22:15:50
22:16:40
22:17:30
22:18:20
22:19:09
Time

eHarmony®

kafka

STRENGTHS

- REALLY NICE DESIGN FOR LOG SHIPPING, E.G. IF YOU WANT TO WRITE A VOLATILE EVENT FIREHOSE TO DISK, AND NEED 'ELASTIC' CONSUMPTION
- IMPLEMENTS BULKING AND BUFFERING (WHEN WRITING TO DISK) HENCE REALLY PERFORMANT
- VARIETY OF TECHNOLOGY INTEGRATIONS IN HADOOP ECOSYSTEM, SPECIFICALLY FOR STREAM PROCESSING

kafka

WEAKNESSES

- NOT BUILT FOR USE IN TRANSACTIONAL SITUATIONS
- WEAK DELIVERY GUARANTEES
- AMATEUR CONSUMER AND PRODUCER CODE IMPLEMENTATIONS
- DOESN'T HAVE MESSAGE ACKNOWLEDGEMENTS BETWEEN BROKER -> CONSUMER
- EXTRA COMPLEXITY BECAUSE OF ZOOKEEPER (BAD RECOVERY WHEN EITHER THE ZOOKEEPER OR BROKER DIE)
  - *EXPERIENCED BY ELEVATED TEAM*
- DOESN'T HAVE A NOTION OF NON-PERSISTENT QUEUE / TOPIC
- VERY CLOSE TIE BETWEEN CONSUMER COUNTS AND PARTITIONS (IF LESS CONSUMERS THAN PARTITIONS, SOME CONSUMERS WILL DO NOTHING)
- LOG CLEANER NEEDS TUNING, OTHERWISE YOU QUICKLY RUN OUT OF SPACE
- DOESN'T PERFORM WELL FOR LARGE MESSAGES
- NEED TO BUY SUPPORT
- ROUTING NOTIONS ARE NOT SUPPORTED
- LEARNING CURVE IS STEEP

eHarmony®

RabbitMQ

STRENGTHS

- EASY TO SETUP WITH VIRTUALLY NO INTERVENTION (IT JUST WORKS)
- IMPLEMENTS PROPER DELIVERY GUARANTEES
- COMPREHENSIVE DOCUMENTATION (ANYTHING I WANTED TO FIND WAS AT MY FINGERTIP)
- MATURE PRODUCT
- PROVIDES A VARIETY OF MESSAGE ROUTING CAPABILITIES (EXCHANGE, BINDING AND QUEUING MODEL)
- VARIETY OF LARGE COMPANIES USE IT AS A MESSAGE BUS
- EASY MIGRATION FROM EXISTING JMS COMPLIANT SYSTEMS
- AVAILABILITY OF CHEF COOKBOOK
- SUPPORTS NOTION OF ROUTING EXCHANGES
- RABBITMQ AS A SERVICE: HTTP://WWW.CLOUDAMQP.COM/

RabbitMQ

WEAKNESSES

- MAXIMUM OUT OF THE BOX THROUGHPUT IS 50K / SEC (ALTHOUGH IT IS FAIR TO MENTION THAT LARGE INSTALLATIONS, UP TO 1.3 MLN / SEC EXIST)

RECOMMENDATION

eHarmony®

RabbitMQ

- USE RABBITMQ AS A STRAIGHT REPLACEMENT FOR OUR CURRENT HORNETQ INSTALLATION
  - EASE OF SETUP
  - SIMPLICITY OF MIGRATION
  - SUPPORT OF ROUTING CAPABILITIES AND JMS COMPLIANCY
  - MEETS OUR PERFORMANCE REQUIREMENTS
  - GREAT DOCUMENTATION AND COMMUNITY SUPPORT
  - PROFESSIONAL SUPPORT IS AVAILABLE FROM PIVOTAL

kafka

- USE KAFKA FOR LOG PROCESSING CASES THAT REQUIRE NO GUARANTEES ON DELIVERY
  - GOOD FOR STREAM PROCESSING
  - GOOD FOR METRICS LOADING AND PROCESSING
  - LOG DELIVERY AND PROCESSING
    - EXCEPTION TRACKING
    - FRAUD ANALYSIS
  - HORTONWORKS CERTIFIED KAFKA DISTRIBUTION IS AVAILABLE

eHarmony®

RabbitMQ

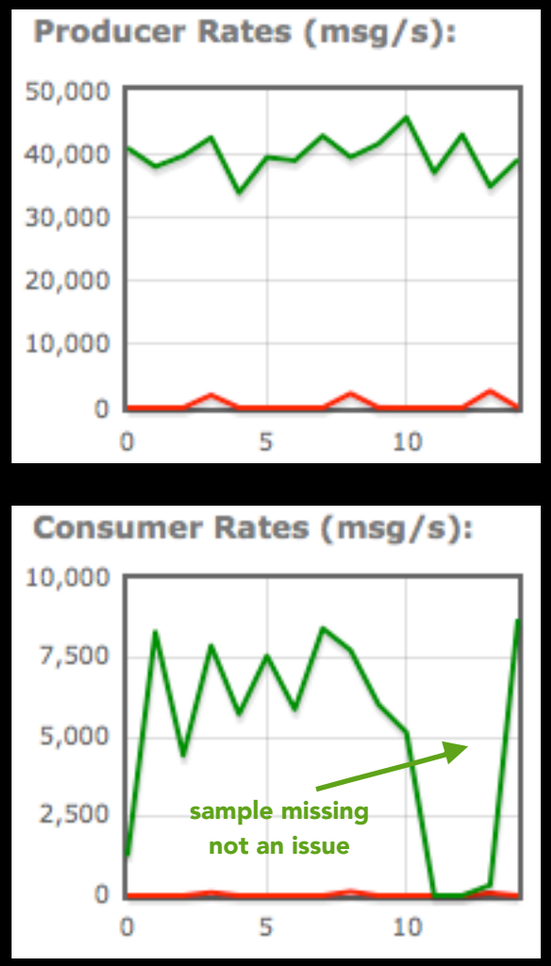
HornetQ

eHarmony®

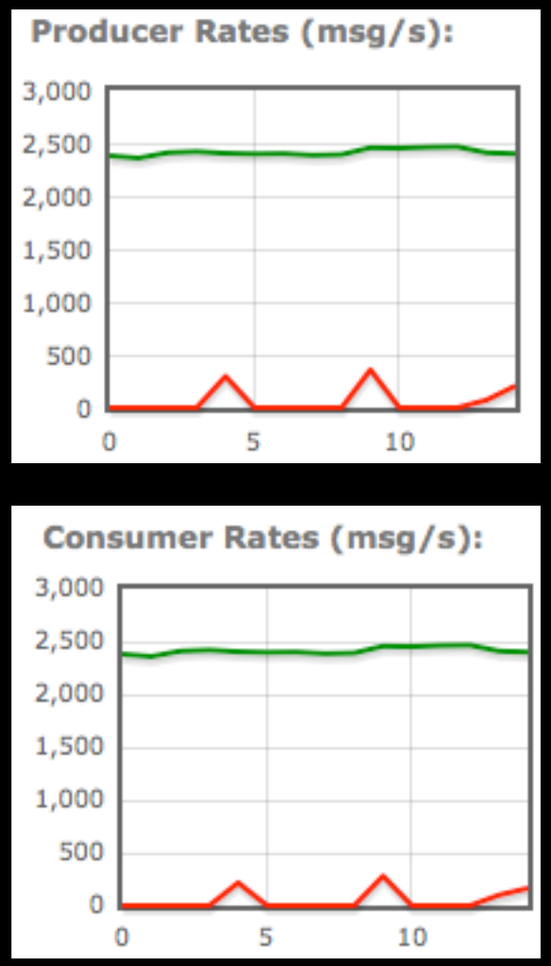
3.5.2 IS MUCH FASTER

rabbitmq-2.7.0
hornetq-2.2.5

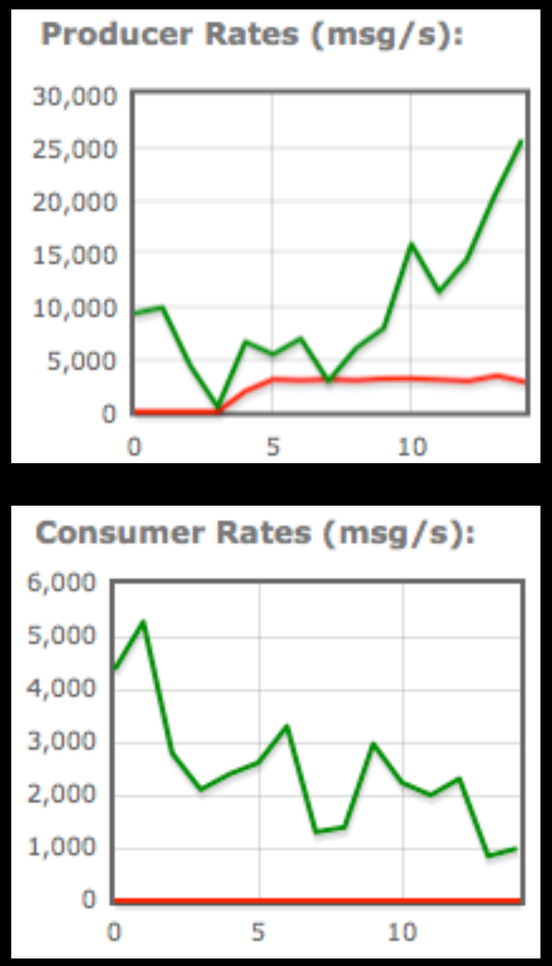
1P / 1C / 1KB / QUEUE NON PERSISTENT



1P / 1C / 1KB / QUEUE PERSISTENT



1P / 1C / 1KB / TOPIC NON PERSISTENT

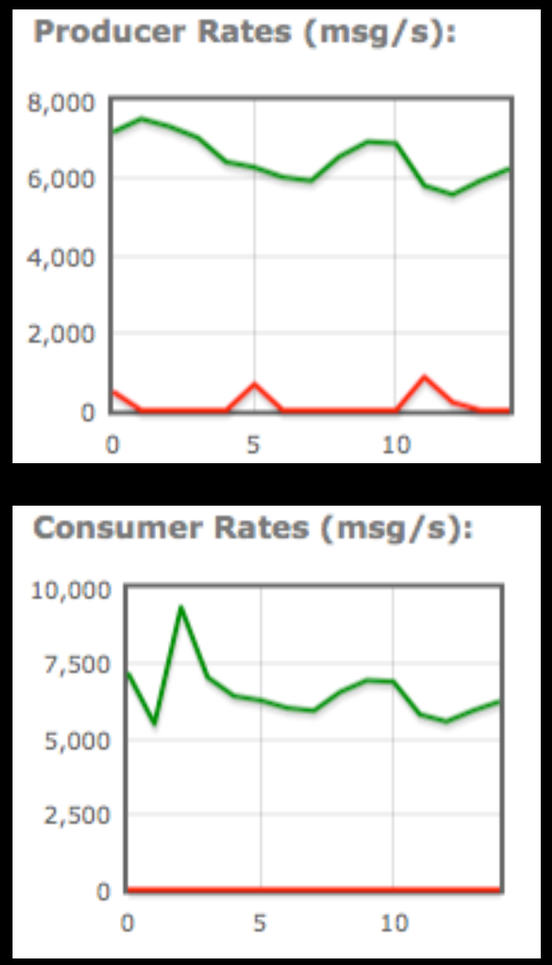


eHarmony®

3.5.2 IS MUCH FASTER

rabbitmq-2.7.0
hornetq-2.2.5

5P / 5C / 1KB / TOPIC PERSISTENT



5P / 5C / 256KB / TOPIC PERSISTENT

eHarmony®

RabbitMQ

eHarmony®

REFERENCES

- http://www.rabbitmq.com/blog/2012/04/25/rabbitmq-performance-measurements-part-2/
- http://hiramchirino.com/stomp-benchmark/ec2-c1.xlarge/index.html
- https://engineering.linkedin.com/kafka/benchmarking-apache-kafka-2-million-writes-second-three-cheap-machines
- http://blog.pivotal.io/pivotal/products/rabbitmq-hits-one-million-messages-per-second-on-google-compute-engine

eHarmony®

THANK YOU

QUESTIONS?

eHarmony®

CREDITS:

Visual Elements From

The Noun Project
http://thenounproject.com

eHarmony®