Apache Kafka

A distributed publish-subscribe messaging system

Neha Narkhede, LinkedIn
@nehanarkhede, 11/11/11

Presented by

Produced by

LinkedIn Account Type: Basic

Home | Profile | Contacts | Groups | Jobs | Inbox 66 | Companies | News | More

Neha Narkhede

Senior Software Engineer at LinkedIn

San Francisco Bay Area | Information Technology and Services

Neha Narkhede

5 Bay Area firms make list of best workplace food sfgate.com · via Anmol Bhasin

Free beer and a Willy Wonka-themed lunch menu have some employees praising the food served at their companies. Five Bay Area firms - Facebook, Google, LinkedIn, Marvell Technology Group and Zynga - made the list of the...

Like (2) · Comment · Share · See all activity · 10 days ago

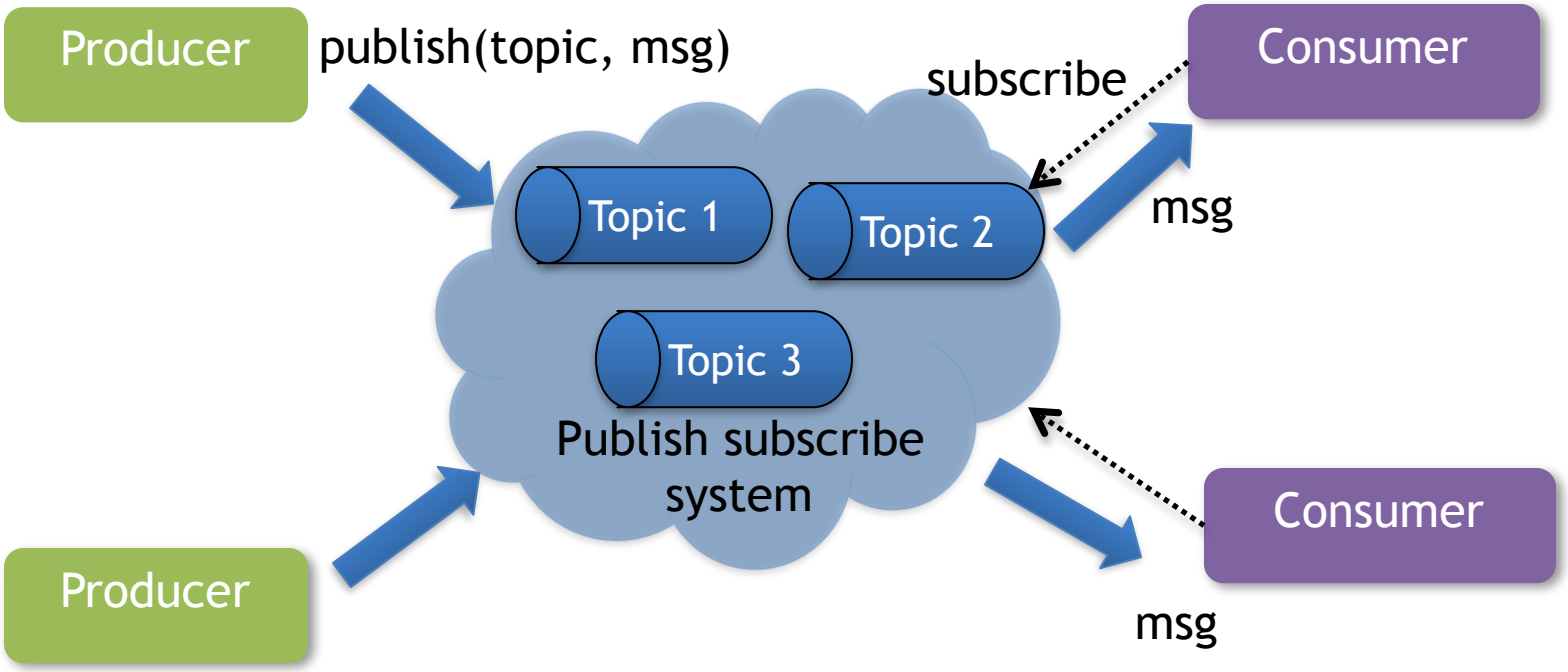
Current	**Senior Software Engineer, Distributed Systems** at **LinkedIn**
Past	Member of Technical Staff at **Oracle Corporation** Summer Intern at **Oracle Corporation**
Education	Georgia Institute of Technology University of Pune St. Anne's School
Recommendations	**3** people have recommended Neha
Connections	**403** connections
Twitter	nehanarkhede
Public Profile	http://www.linkedin.com/in/nehanarkhede

Share | PDF | Print

Outline

- ***Introduction to pub-sub***
- Kafka at LinkedIn
- Hadoop and Kafka
- Design
- Performance

What is pub sub ?



Outline

- Introduction to pub-sub
- ***Kafka at LinkedIn***
- Hadoop and Kafka
- Design
- Performance

Motivation

- Activity tracking

- Operational metrics

Kafka

- Distributed
- Persistent
- High throughput

All Updates · LinkedIn Coworkers · Shares · More ▾ Recent · Top ·

Peter Skomoroch

cloudera Ignition, Accel, Greylock Put $40M In Apache Hadoop Distribution... techcrunch.com

Like · Comment · Send a message · Share · 36 seconds ago

Trending in Venture Capital & Private Equity

Add a comment…

Peter Skomoroch

Common Crawl Foundation Announces 5 Billion Page Web Index, Available... readwriteweb.com

Like · Comment · Send a message · Share · 2 minutes ago

Trending in Computer Software and Online Media

Add a comment…

Neha Desai is now connected to Eli Smaga

Send a message · 19 minutes ago

Ajay Choudhari is now connected to Ward Wilson [LION]

Send a message · 33 minutes ago

Ashwin Ram via Twitter

ashwinram RT @daniel_kraft : @tgoetz at #FutureMed on the role feedback loops in #health http://t.co/O0mzJw5l #in

Retweet · Favorite · Reply | Like · Comment · Share · 40 minutes ago

Create snapshot:

Customize Host Sele

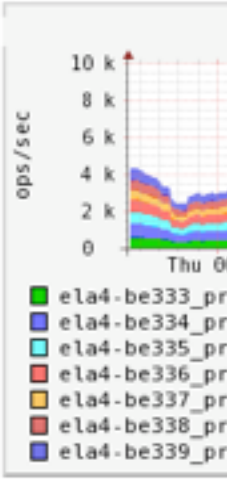
Duration: 2

Start Time: YY

Timezone: US/Paci

Stack: | Consol

Hide Controls



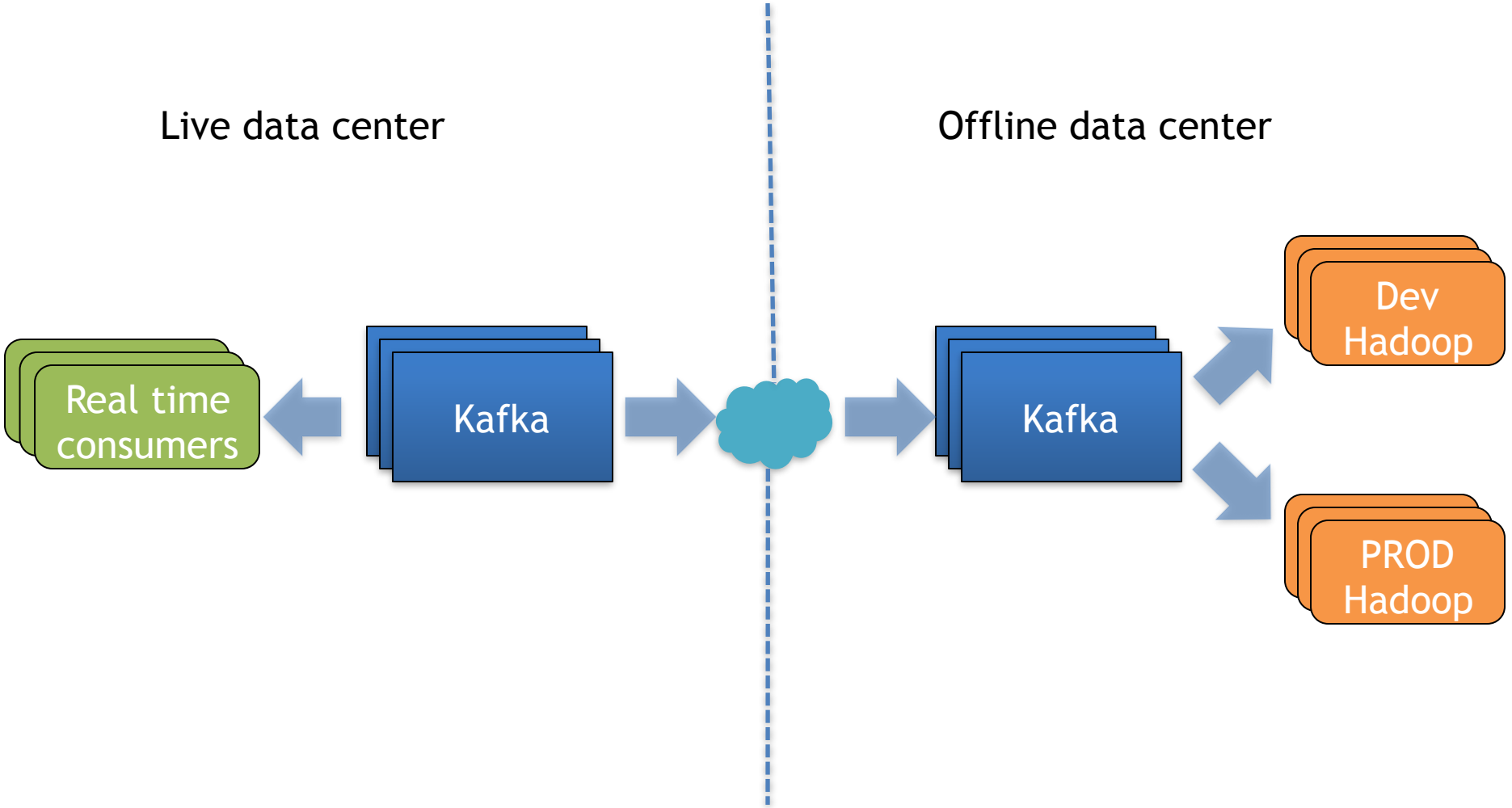
oughput

Apache Con North America 2011

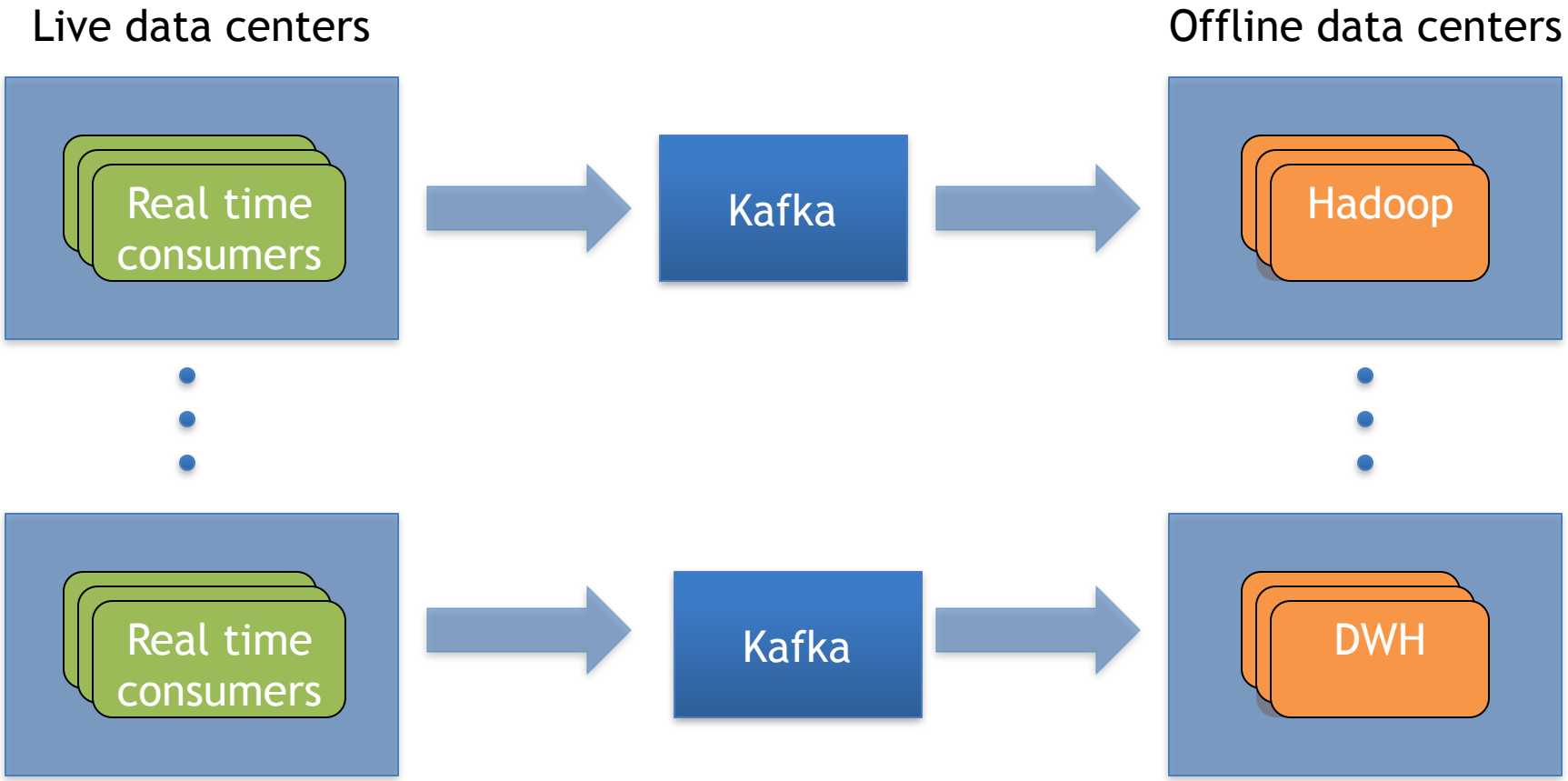
Outline

- Introduction to pub-sub
- Kafka at LinkedIn
- ***Hadoop and Kafka***
- Design
- Performance

Hadoop Data Load for Kafka

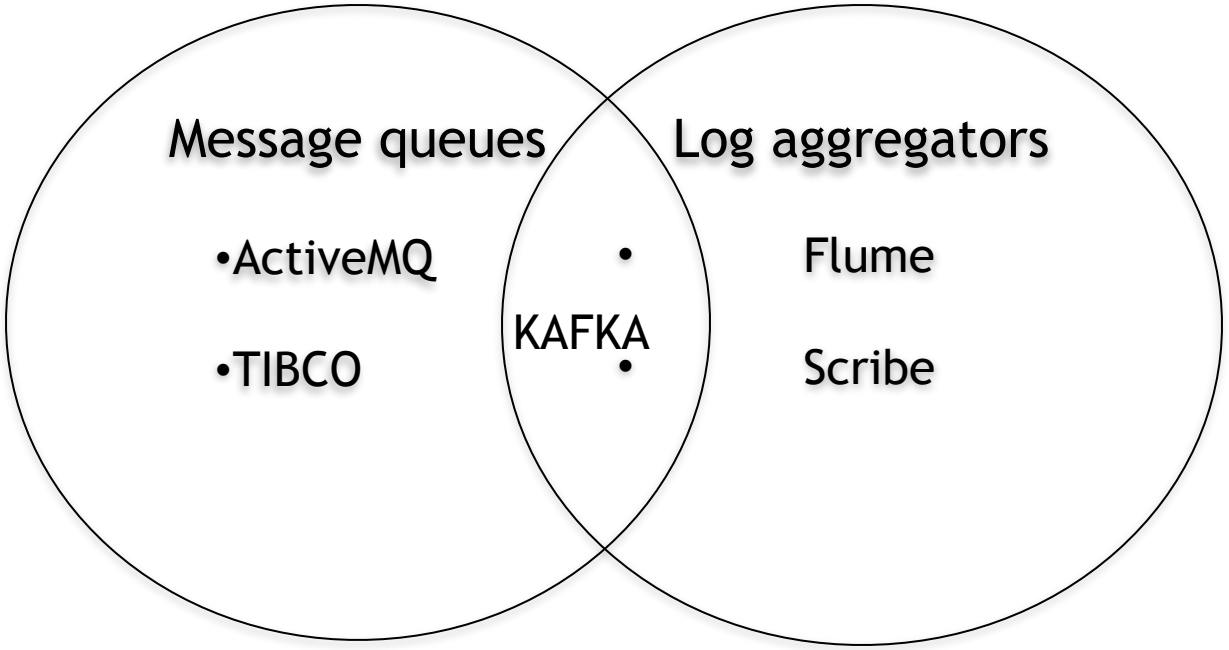


Multi DC data deployments



Volume

- 20B events/day
- 3 terabytes/day
- 150K events/sec



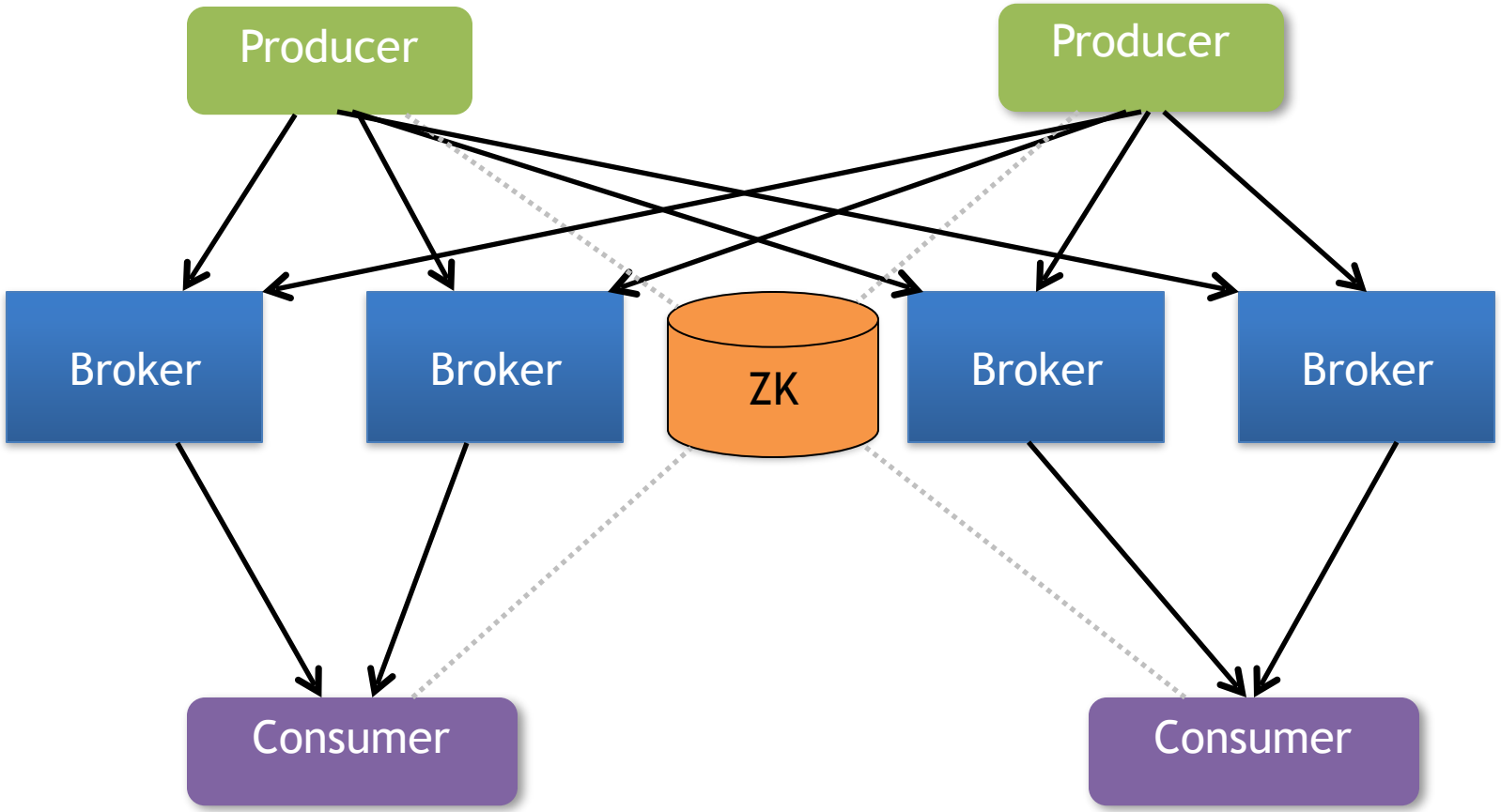
- Low throughput
- Secondary indexes
- Tuned for low latency

- Focus on HDFS
- Push model
- No rewindable consumption

What Kafka offers

- Very high performance
- Elastically scalable
- Low operational overhead
- Durable, highly available (coming soon)

Architecture



Outline

- Introduction to pub-sub
- Kafka at LinkedIn
- Hadoop and Kafka
- ***Design***
- Performance

Efficiency #1: simple storage

- Each topic has an ever-growing log
- A log == a list of files
- A message is addressed by a log offset

```
[nnarkhed@nnarkhed-md kafka-logs]$ tree -s
.
|-- [      4096]  novels-0
|   `-- [119878441]  00000000000000000000.kafka
`-- [      4096]  short stories-0
    `-- [119817774]  00000000000000000000.kafka

2 directories, 2 files
```

Efficiency #2: careful transfer

- Batch send and receive
- No message caching in JVM
- Rely on file system buffering
- Zero-copy transfer: file -> socket

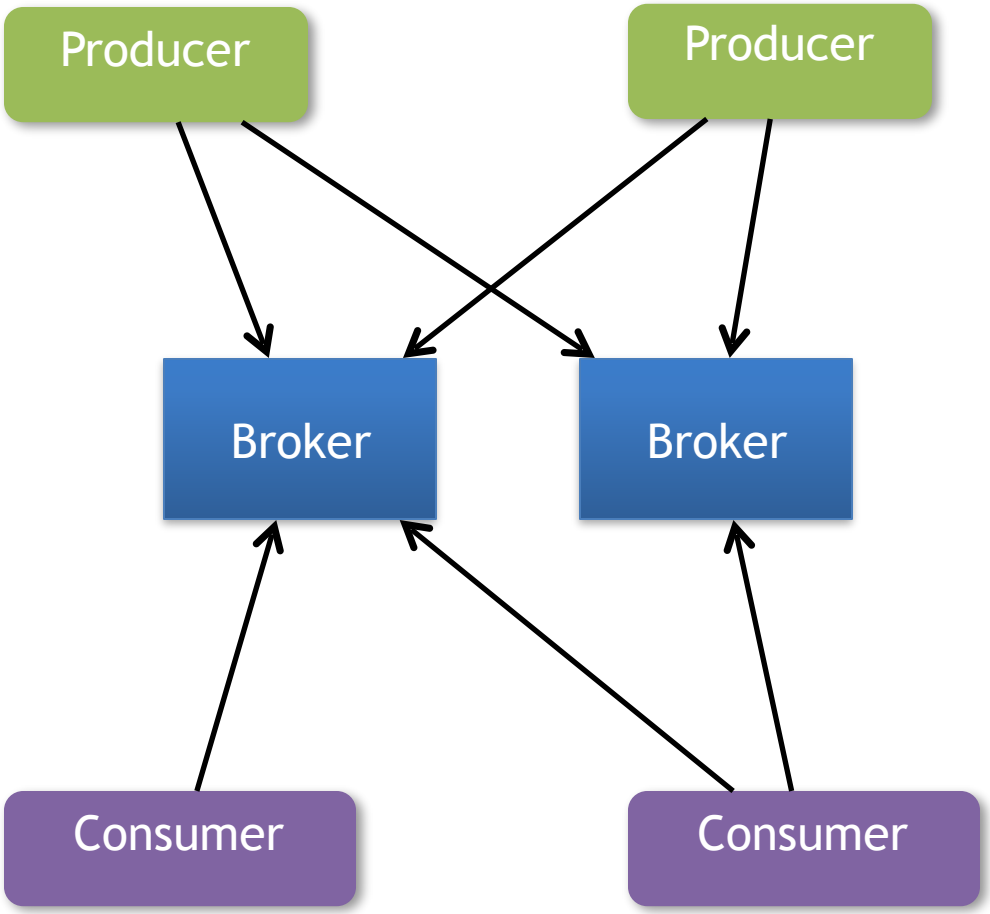
Multi subscribers

- 1 file system operation per request
- Consumption is cheap
- SLA based message retention
- Rewindable consumption

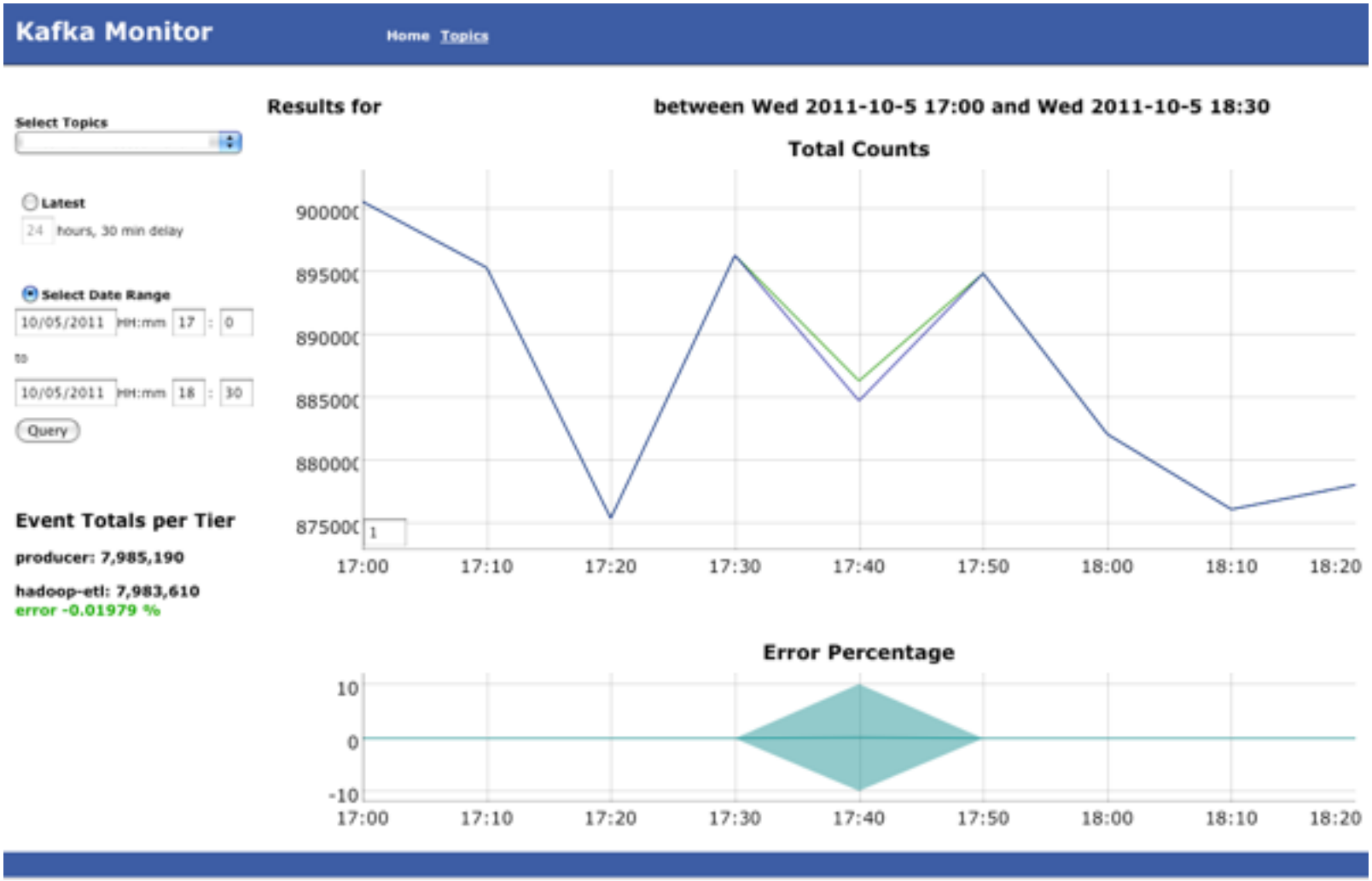
Guarantees

- Data integrity checks
- At least once delivery
- In order delivery, per partition

Automatic load balancing



Auditing


Kafka Monitor
Home Topics
Select Topics
Latest
24 hours, 30 min delay
Select Date Range
10/05/2011 HH:mm 17 : 0
to
10/05/2011 HH:mm 18 : 30
Query
Event Totals per Tier
producer: 7,985,190
hadoop-etl: 7,983,610
error -0.01979 %
Results for
between Wed 2011-10-5 17:00 and Wed 2011-10-5 18:30
Total Counts
90000
89500
89000
88500
88000
87500
17:00
17:10
17:20
17:30
17:40
17:50
18:00
18:10
18:20
Error Percentage
10
0
-10
17:00
17:10
17:20
17:30
17:40
17:50
18:00
18:10
18:20

Outline

- Introduction to pub-sub
- Kafka at LinkedIn
- Hadoop and Kafka
- Design
- ***Performance***

Performance

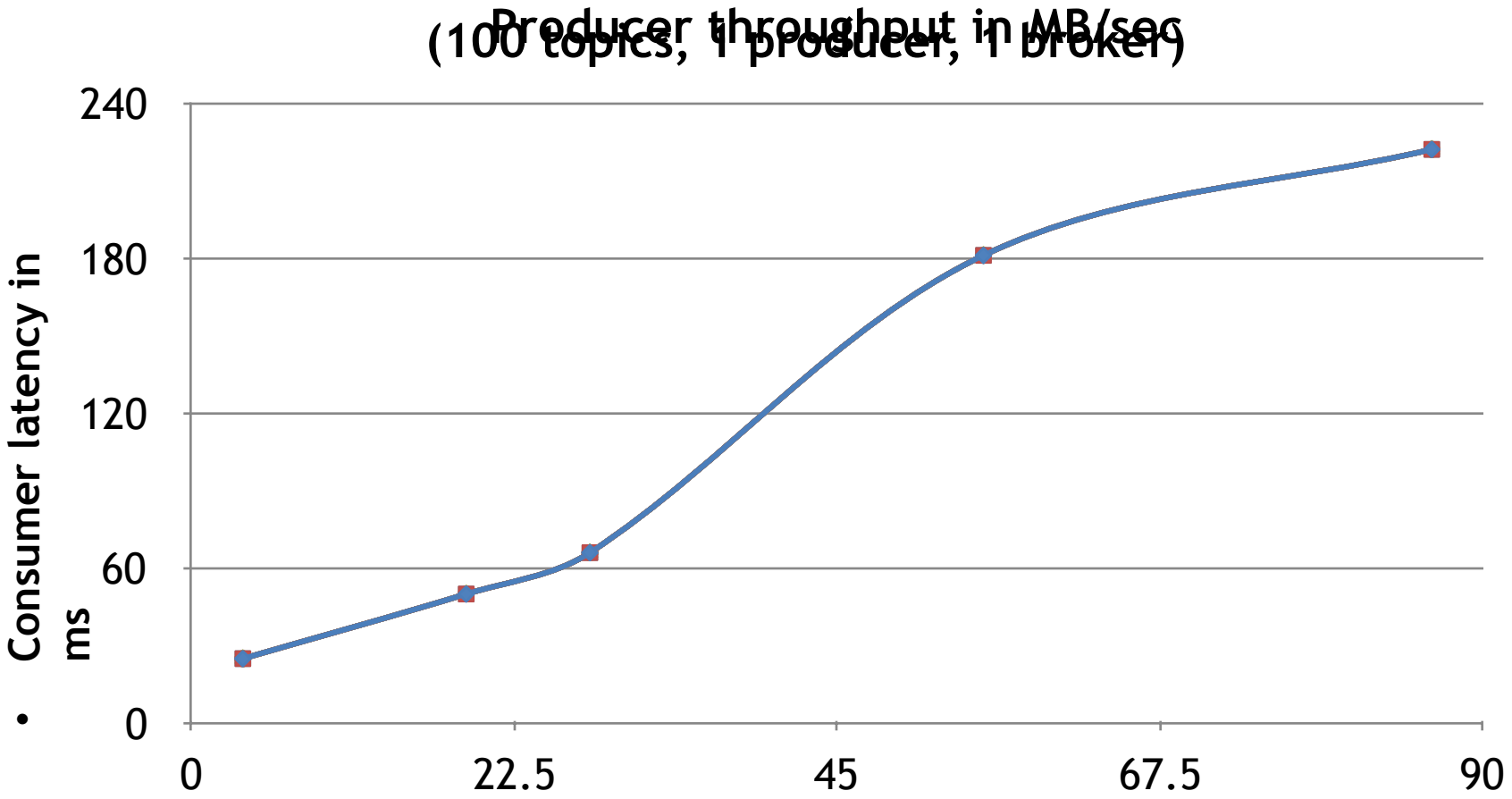
- 2 Linux boxes
  - 16 2.0 GHz cores
  - 6 7200 rpm SATA drive RAID 10
  - 24GB memory
  - 1Gb network link
- 200 byte messages
- Producer batch size 200 messages

Basic performance metrics

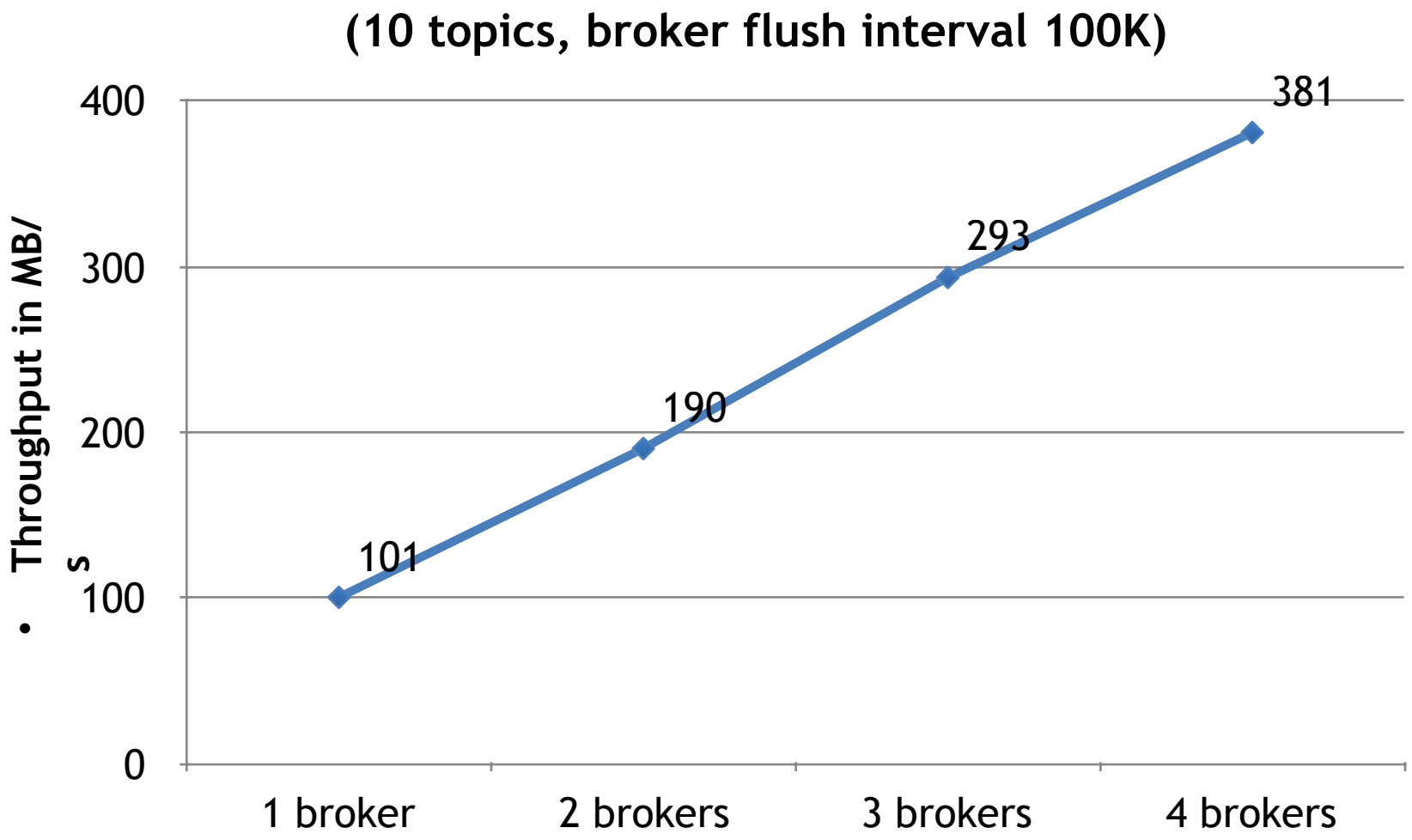
- Producer batch size = 40K
- Consumer batch size = 1MB
- 100 topics, broker flush interval = 100K

  - Producer throughput = 90 MB/sec
  - Consumer throughput = 60 MB/sec
  - Consumer latency = 220 ms

Latency vs throughput

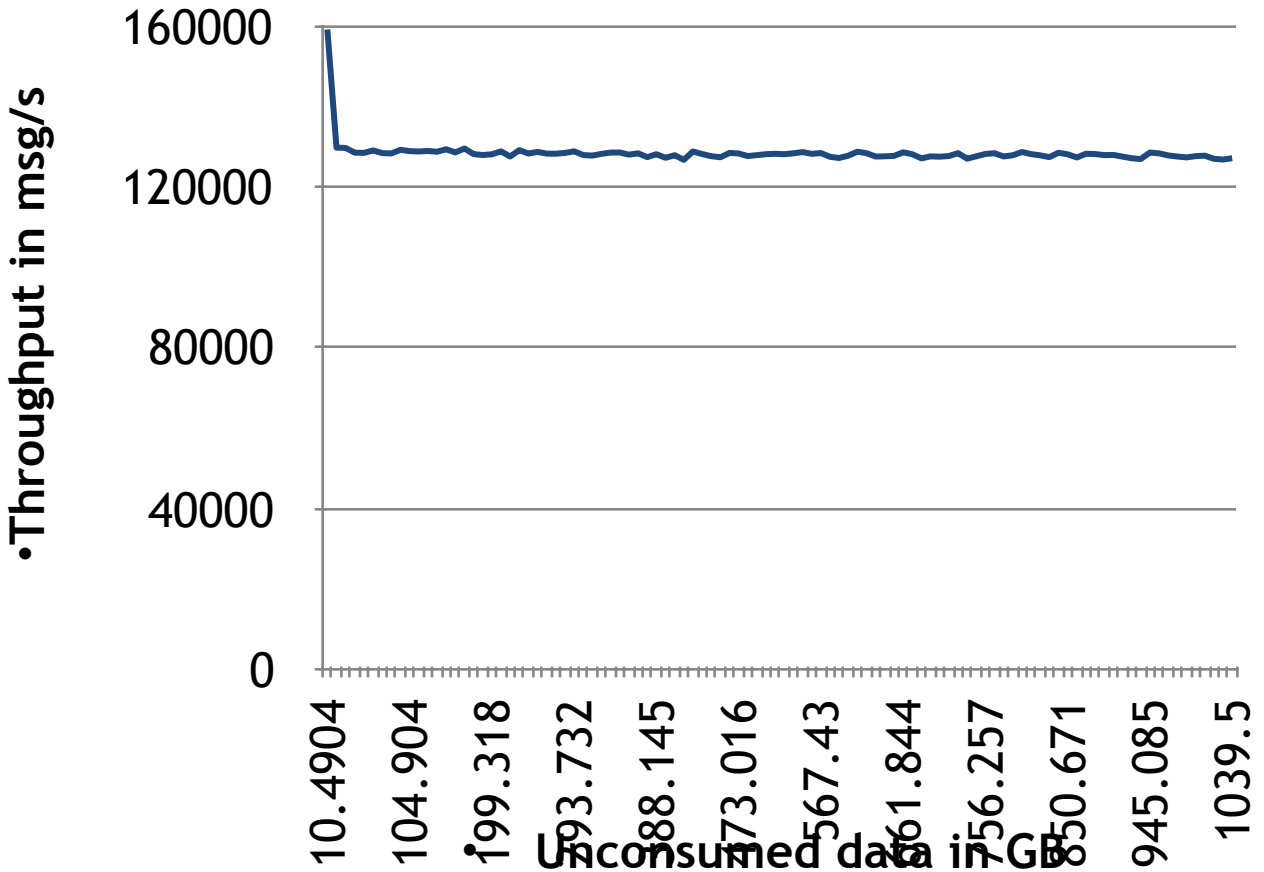


Scalability



Throughput vs Unconsumed data

- (1 topic, broker flush interval 10K)



State of the system

- 4 clusters per colo, 4 servers each
- 850 socket connections per server
- 20 TB
- 430 topics
- Batched frontend to offline datacenter latency => 6-10 secs
- Frontend to Hadoop latency => 5 min

State of the system

- Successfully deployed in production at LinkedIn and other startups
- Apache incubator inclusion
- 0.7 Release
  - Compression
  - Cluster mirroring

Replication

Some project ideas

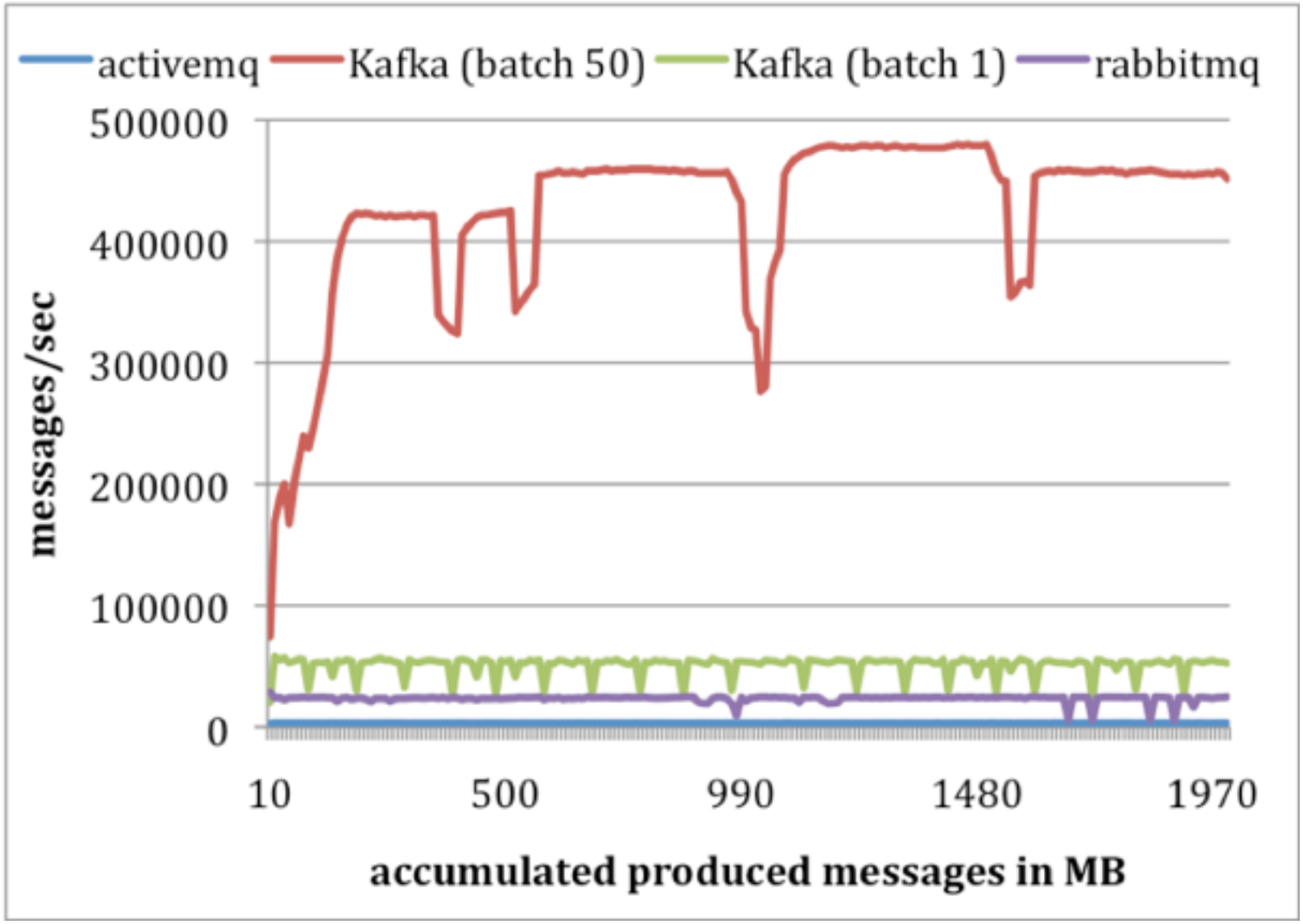
- Security
- Long poll
- More compression codecs
- Locality of consumption

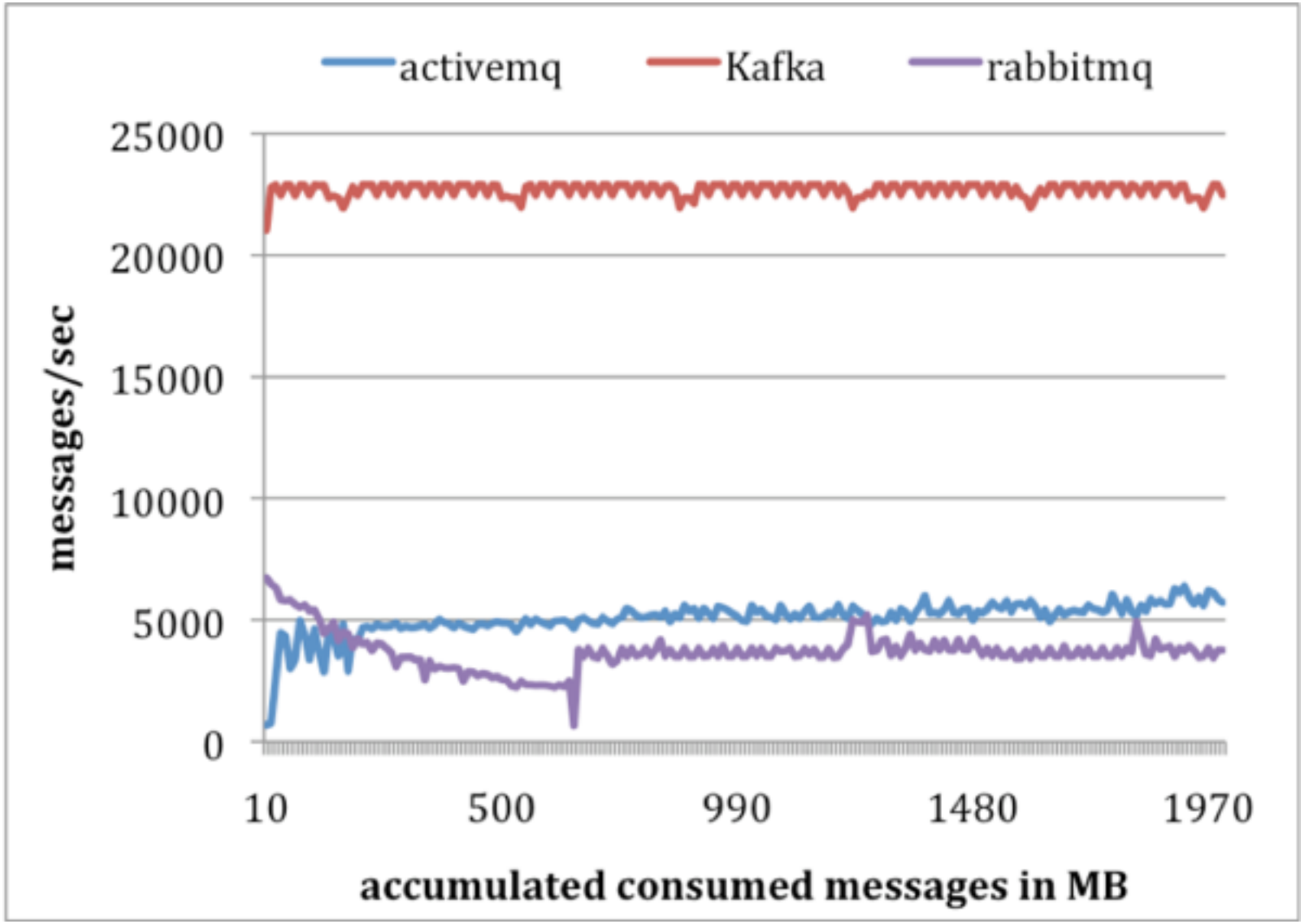
Team

- Jay Kreps
- Jun Rao
- Neha Narkhede
- Joel Koshy
- Chris Burroughs

THANK YOU

http://incubator.apache.org/kafka/index.html
kafka-users@incubator.apache.org
http://www.linkedin.com/in/nehanarkhede
@nehanarkhede
#kafka


activemq
Kafka (batch 50)
Kafka (batch 1)
rabbitmq
500000
400000
300000
200000
100000
0
messages/sec
10
500
990
1480
1970
accumulated produced messages in MB


activemq
Kafka
rabbitmq
25000
20000
15000
10000
5000
0
messages/sec
10
500
990
1480
1970
accumulated consumed messages in MB

API

```
// set the right config options
Properties props = new Properties();
props.put("zk.connect", "127.0.0.1:2181");
props.put("groupid", "kafka-fans");
String topic = "Kafka-Novels";
// pull all the messages in ONE stream
// Kafka is not a the serialization business
int numConsumerStreams = 1;
// retrieve the message streams for topics
Iterable streams = client.createMessageStreams(...).get(topic);

byte[] data = "The Metamorphosis".getBytes();

for(message : streams)
{
// send data to Kafka
producer.send(topic, data);
// process message
}
```