Building LinkedIn’s Real-time Data Pipeline

Jay Kreps

Jay Kreps

Principal Staff Engineer at LinkedIn

Mountain View, California (San Francisco Bay Area) | Internet

Current	**Principal Staff Engineer** at **LinkedIn**
Past	Principal Engineer and Engineering Manager at **LinkedIn**
	Senior Software Engineer at **LinkedIn**
	Principal Software Engineer at **LinkedIn**
	see all
Education	University of California, Santa Cruz
	University of California, Santa Cruz
Recommendations	1 person has recommended Jay
Connections	**500+** connections
Websites	Blog
	Project Voldemort
	Apache Kafka

What is a data pipeline?

What data is there?

- Database data
- Activity data
  - Page Views, Ad Impressions, etc
- Messaging
  - JMS, AMQP, etc
- Application and System Metrics
  - Rrdtool, graphite, etc
- Logs
  - Syslog, log4j, etc

"One Size Fits All": An Idea Whose Time Has Come and Gone

Michael Stonebraker
Computer Science and Artificial Intelligence Laboratory, M.I.T., and StreamBase Systems, Inc.
stonebraker@csail.mit.edu

Uğur Çetintemel
Department of Computer Science Brown University, and StreamBase Systems, Inc.
ugur@cs.brown.edu

Abstract

The last 25 years of commercial DBMS development can be summed up in a single phrase: "One size fits all". This phrase refers to the fact that the traditional DBMS architecture (originally designed and optimized for business data processing) has been used to support many data-centric applications with widely varying

of multiple code lines causes various practical problems, including:

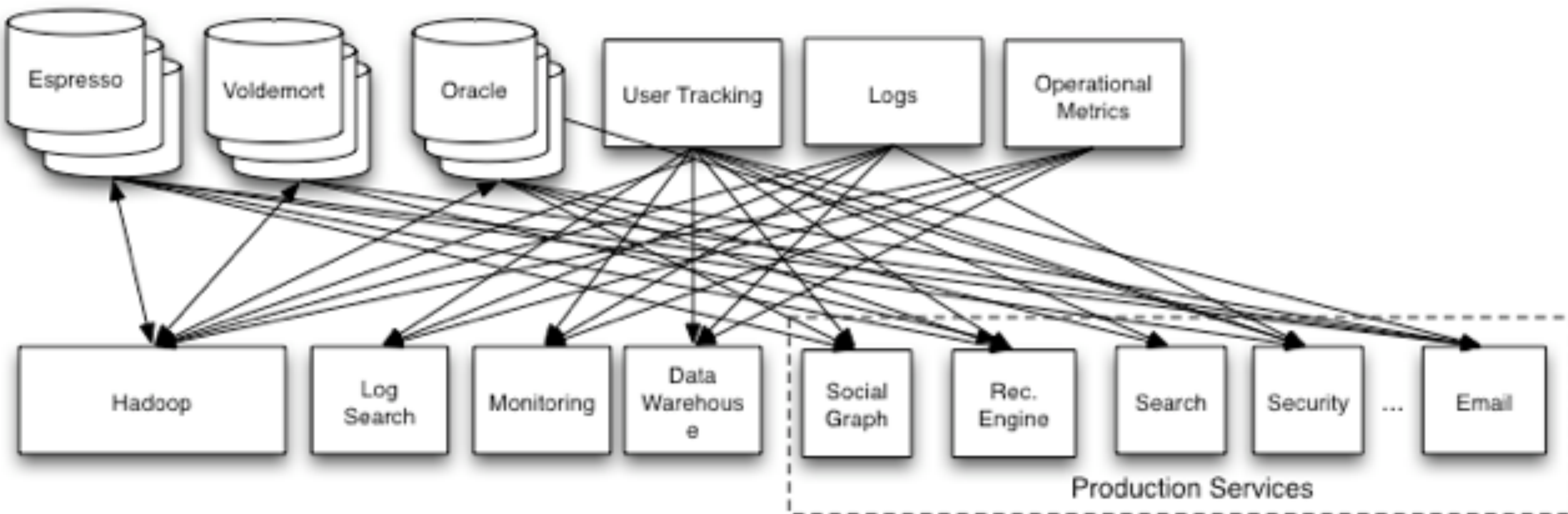
- *a cost problem*, because maintenance costs increase at least linearly with the number of code lines;
- *a compatibility problem*, because all applications have to run against every code line;
- *a sales problem*, because salespeople get confused

Data Systems at LinkedIn

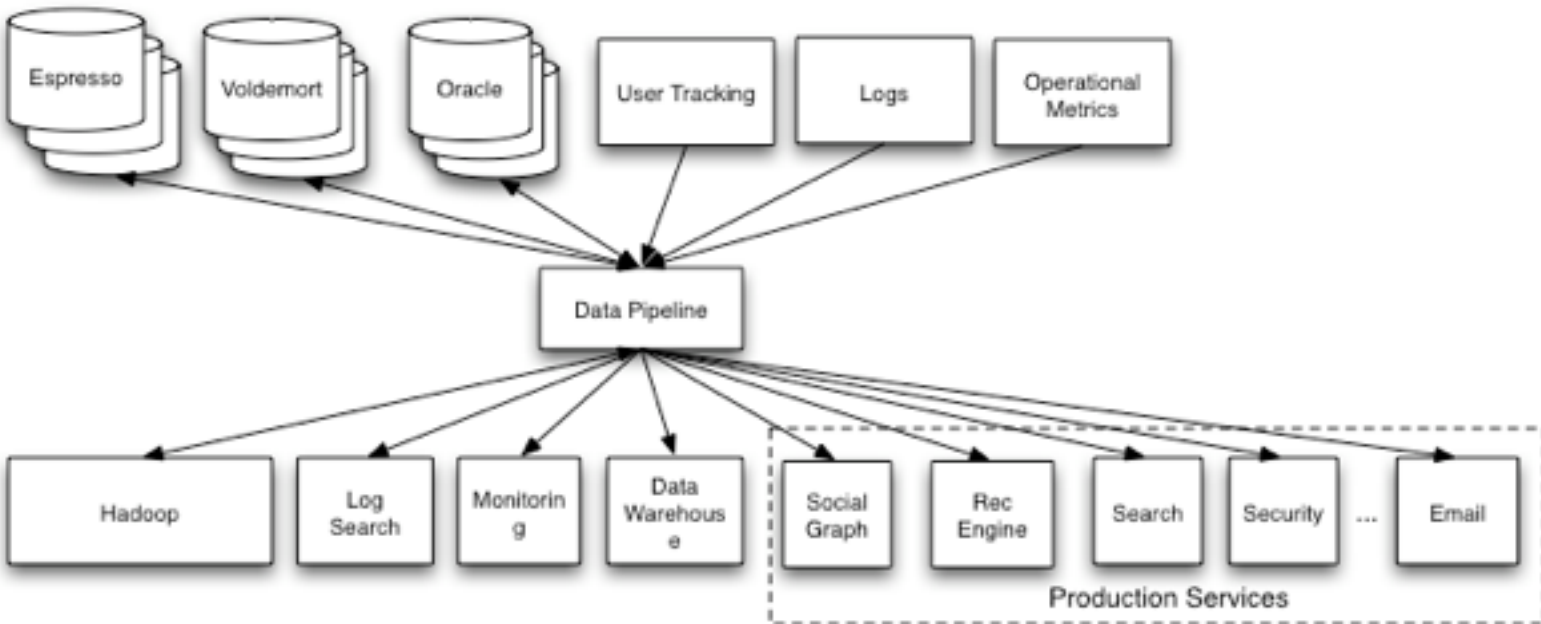
- Search
- Social Graph
- Recommendations
- Live Storage
- Hadoop
- Data Warehouse
- Monitoring Systems
- …

Problem: Data Integration

Point-to-Point Pipelines



Centralized Pipeline



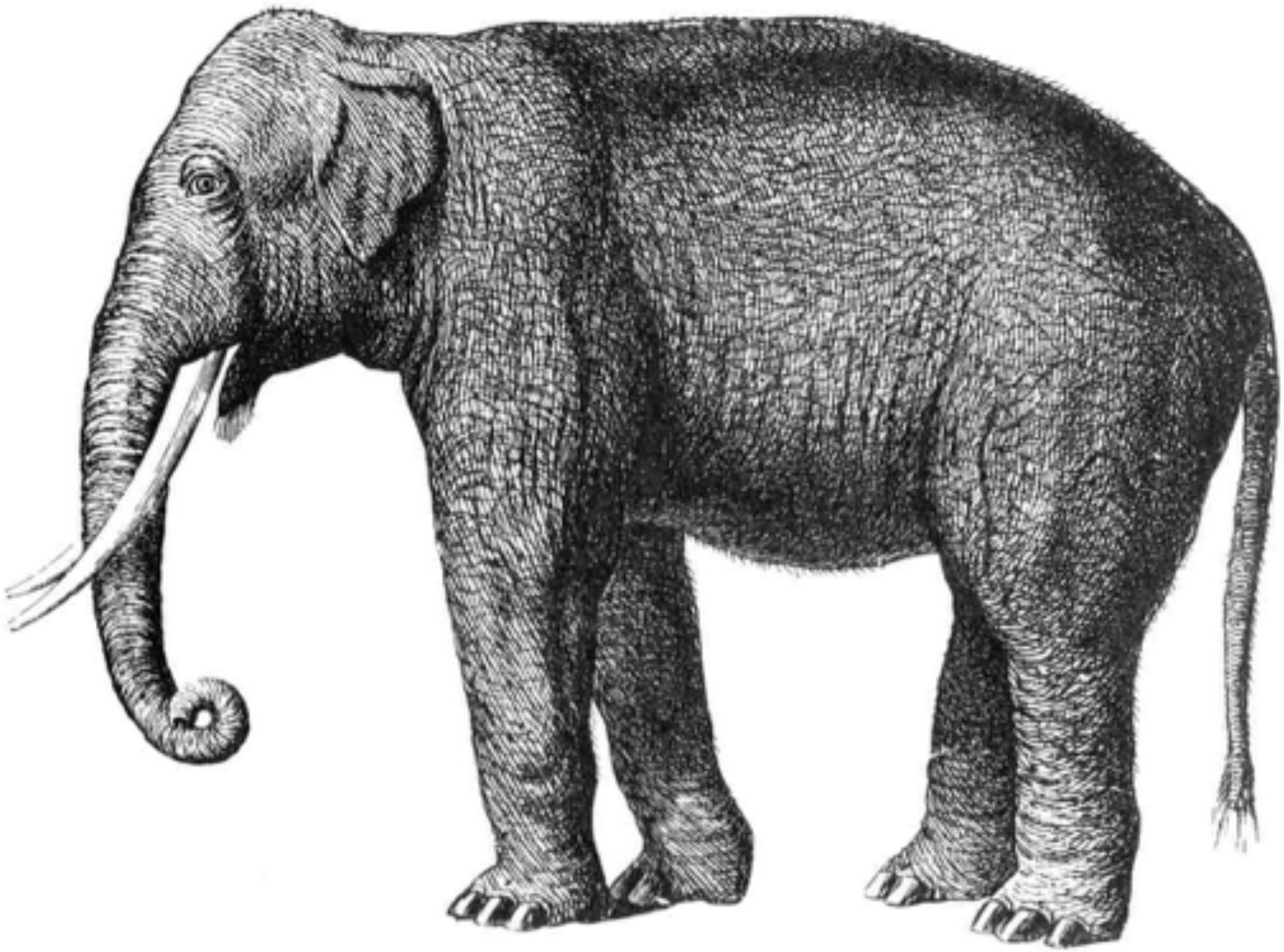
How have companies solved this problem?

The Enterprise Data Warehouse

Problems

- Data warehouse is a batch system
- Central team that cleans all data?
- One person’s cleaning…
- Relational mapping is non-trivial…

My Experience

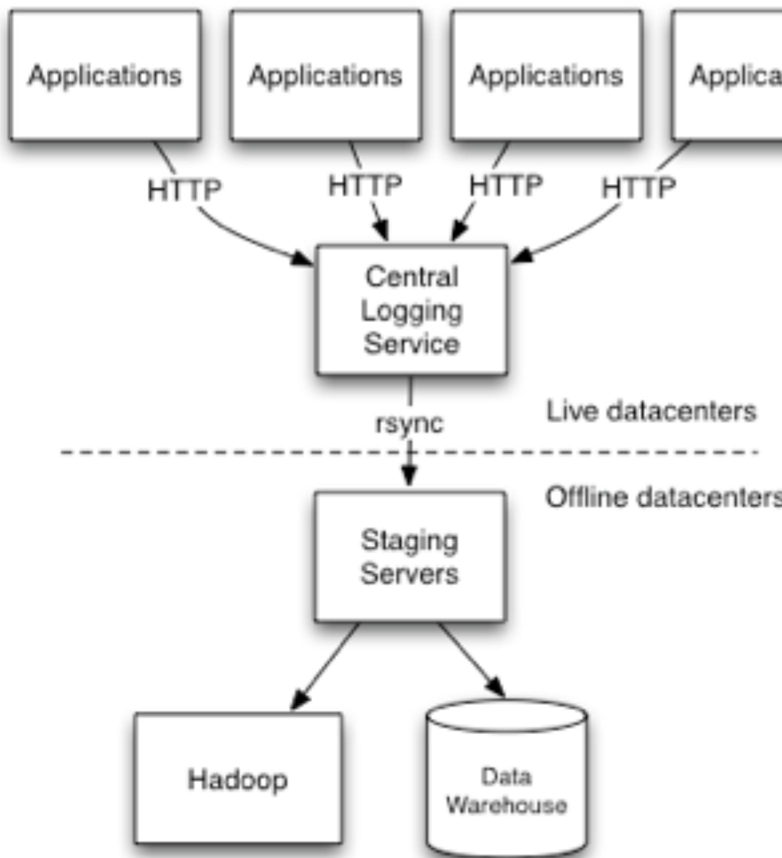


LinkedIn’s Pipeline

LinkedIn Circa 2010

- Messaging: ActiveMQ
- User Activity: In house log aggregation
- Logging: Splunk
- Metrics: JMX => Zenoss
- Database data: Databus, custom ETL

2010 User Activity Data Flow



Problems

- Fragility
- Multi-hour delay
- Coverage
- Labor intensive
- Slow
- Does it work?

Four Ideas

1. Central commit log for all data
2. Push data cleanliness upstream
3. O(1) ETL
4. Evidence-based correctness

Four Ideas

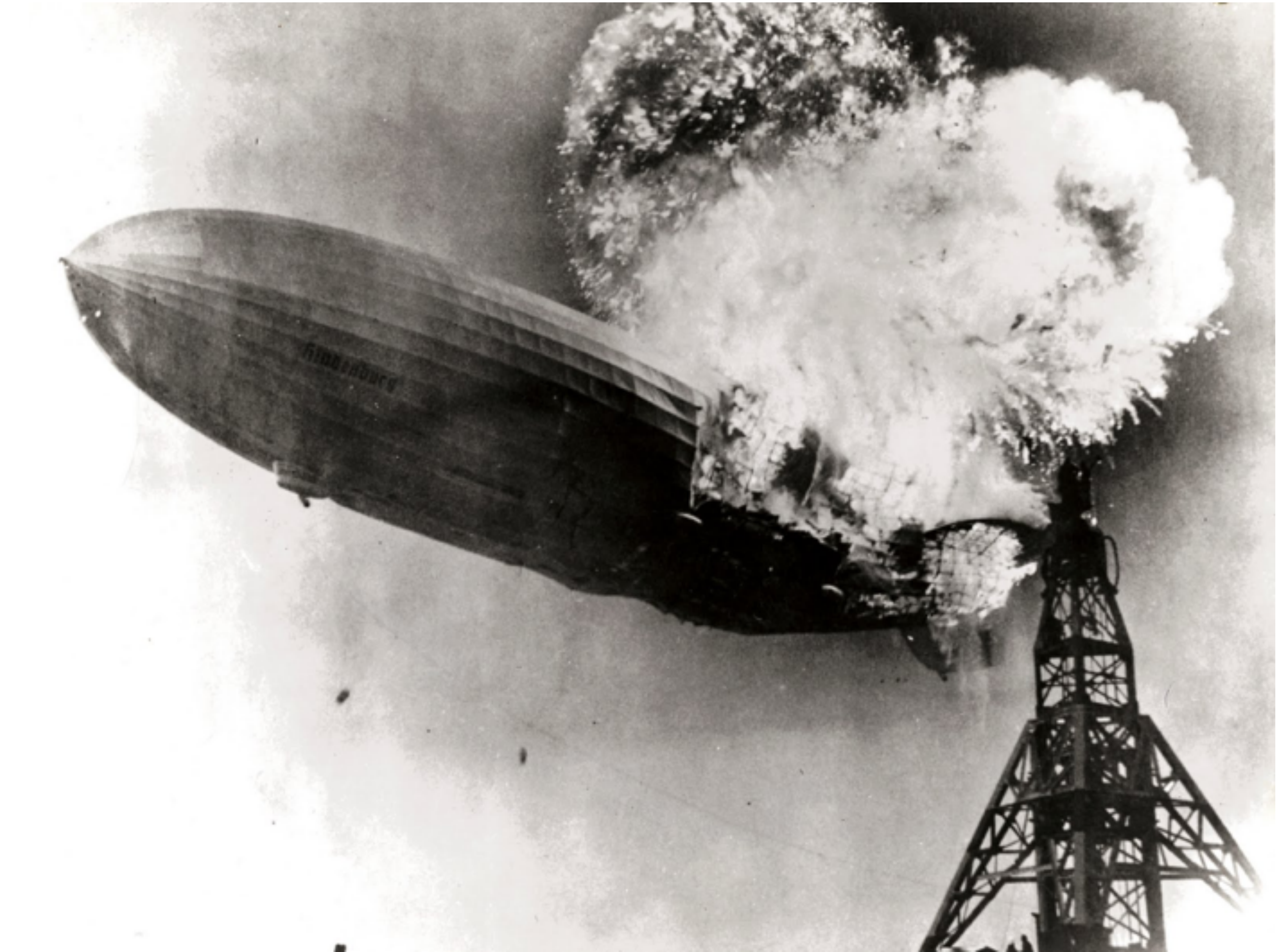
1. Central commit log for all data
2. Push data cleanliness upstream
3. O(1) ETL
4. Evidence-based correctness

What kind of infrastructure is needed?

Very confused

- Messaging (JMS, AMQP, …)
- Log aggregation
- CEP, Streaming

First Attempt:
Don’t reinvent the wheel!



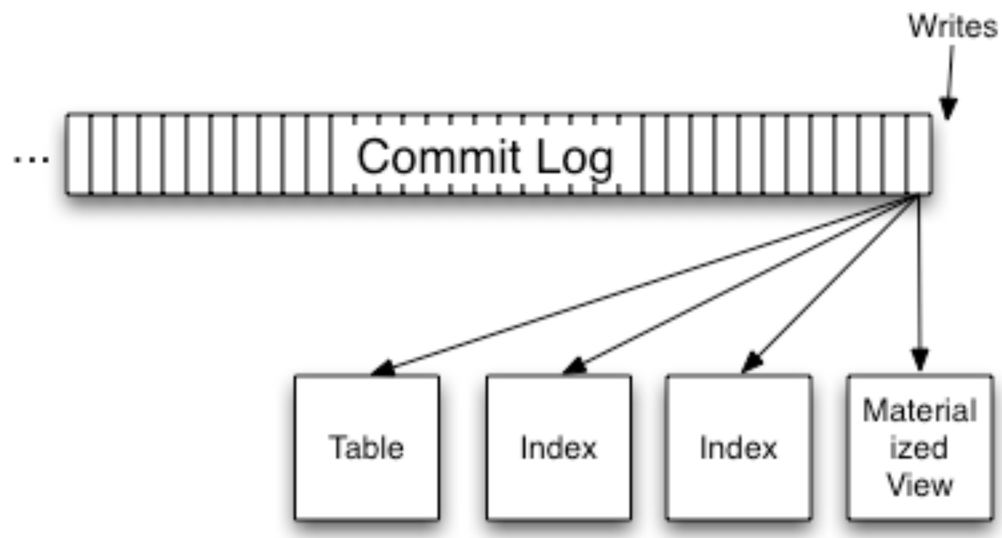
Problems With Messaging Systems

- Persistence is an afterthought
- Ad hoc distribution
- Odd semantics
- Featuritis

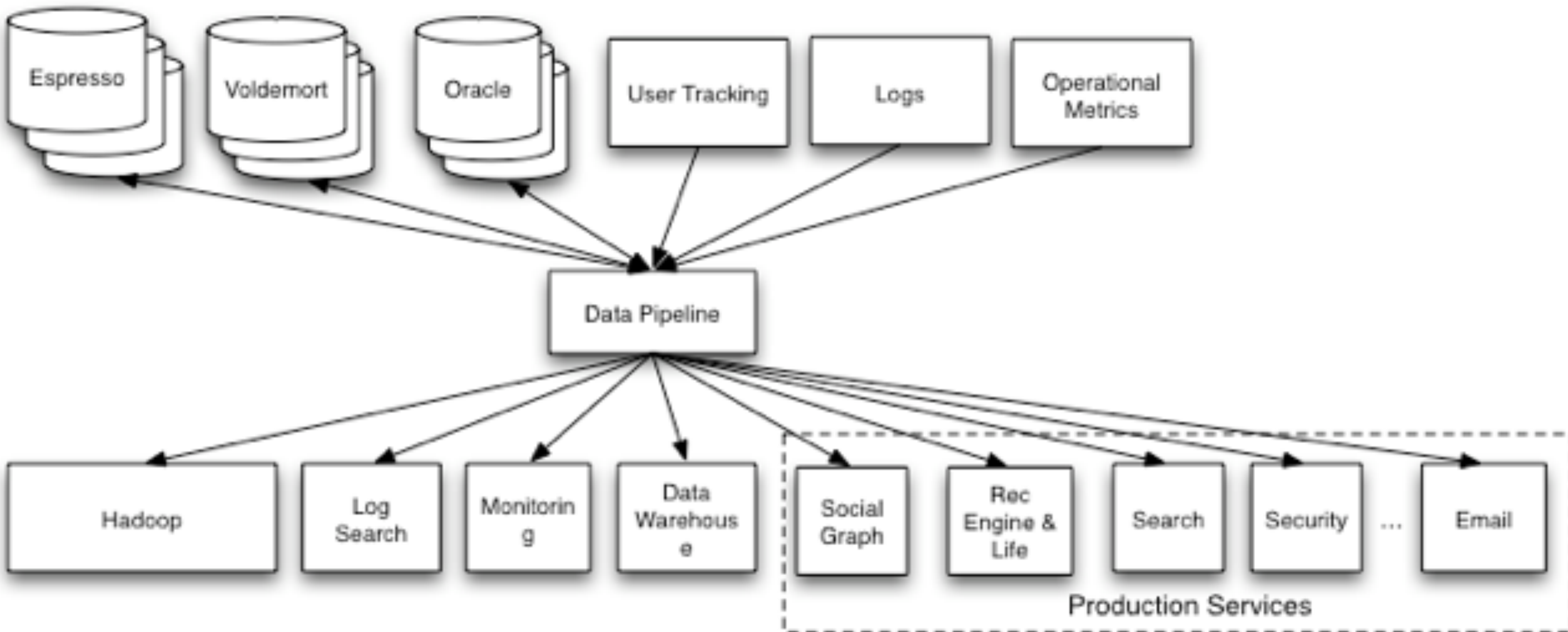
Second Attempt:
Reinvent the wheel!

Idea: Central, Distributed Commit Log

What is a commit log?



Data Flow



Espresso
Voldemort
Oracle
User Tracking
Logs
Operational Metrics
Data Pipeline
Hadoop
Log Search
Monitorin g
Data Warehous e
Social Graph
Rec Engine & Life
Search
Security
...
Email
Production Services

Apache Kafka

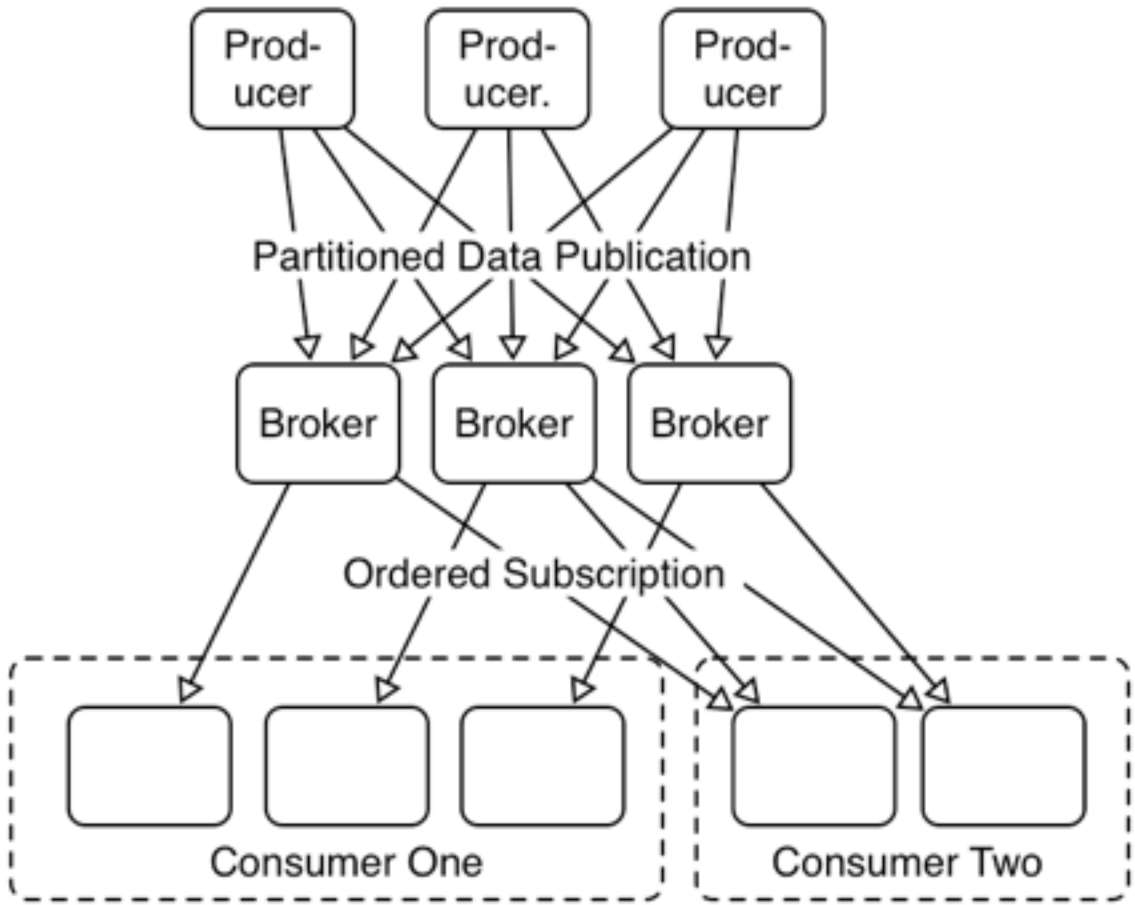
Some Terminology

- Producers send messages to Brokers
- Consumers read messages from Brokers
- Messages are sent to a Topic
- Each topic is broken into one or more ordered partitions of messages

APIs

- send(String topic, String key, Message message)
- Iterator<Message>

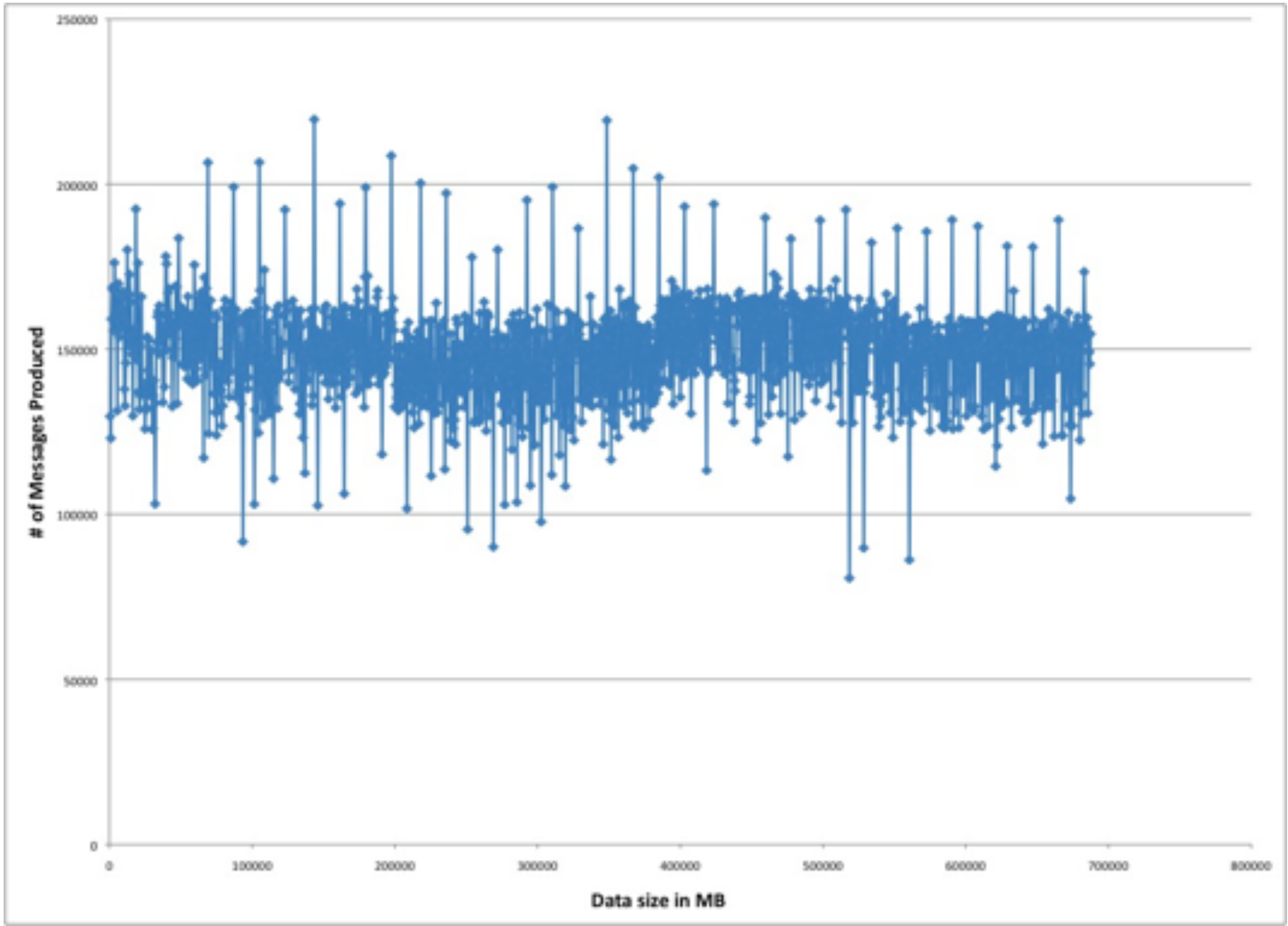
Distribution



Performance

- 50MB/sec writes
- 110MB/sec reads

Performance



Performance Tricks

- Batching
  - Producer
  - Broker
  - Consumer
- Avoid large in-memory structures
  - Pagecache friendly
- Avoid data copying
  - sendfile
- Batch Compression

Kafka Replication

- In 0.8 release
- Messages are highly available
- No centralized master

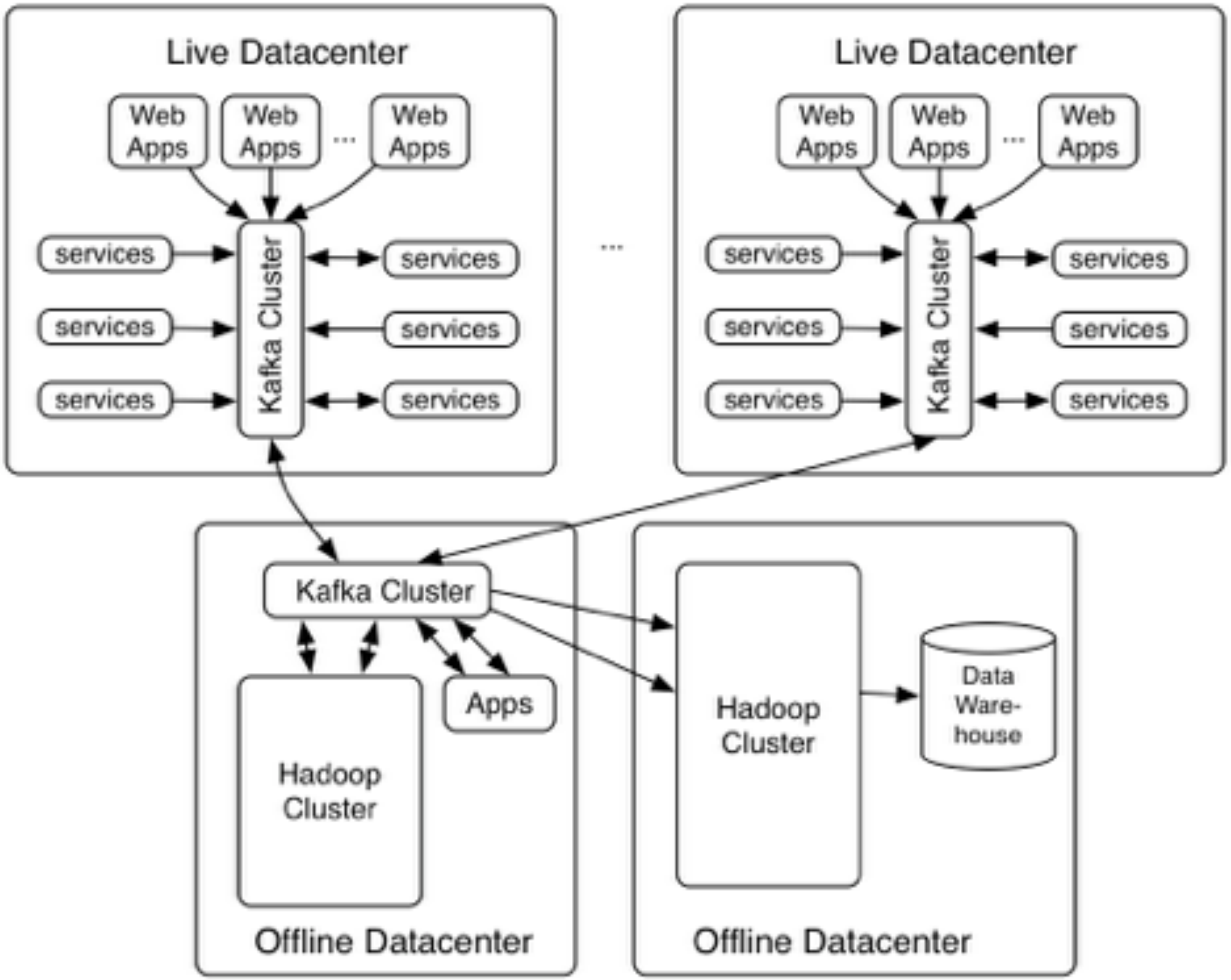
Kafka Info

http://incubator.apache.org/kafka

Usage at LinkedIn

- 10 billion messages/day
- Sustained peak:
  - 172,000 messages/second written
  - 950,000 messages/second read
- 367 topics
- 40 real-time consumers
- Many ad hoc consumers
- 10k connections/colo
- 9.5TB log retained
- End-to-end delivery time: 10 seconds (avg)

Datacenters



Four Ideas

1. Central commit log for all data
2. Push data cleanliness upstream
3. O(1) ETL
4. Evidence-based correctness

Problem

- Hundreds of message types
- Thousands of fields
- What do they all mean?
- What happens when they change?

Make activity data part of the domain model

Schema free?

```
LOAD 'student' USING PigStorage()
AS (name:chararray, age:int, gpa:float)
```

Schemas

- Structure can be exploited
  - Performance
  - Size
- Compatibility
- Need a formal contract

Avro Schema

- Avro – data definition and schema
- Central repository of all schemas
- Reader always uses same schema as writer
- Programatic compatibility model

Workflow

1. Check in schema
2. Code review
3. Ship

Four Ideas

1. Central commit log for all data
2. Push data cleanliness upstream
3. O(1) ETL
4. Evidence-based correctness

Hadoop Data Load

- Map/Reduce job does data load
- One job loads all events
- Hive registration done automatically
- Schema changes handled transparently
- ~5 minute lag on average to HDFS

Four Ideas

1. Central commit log for all data
2. Push data cleanliness upstream
3. O(1) ETL
4. Evidence-based correctness

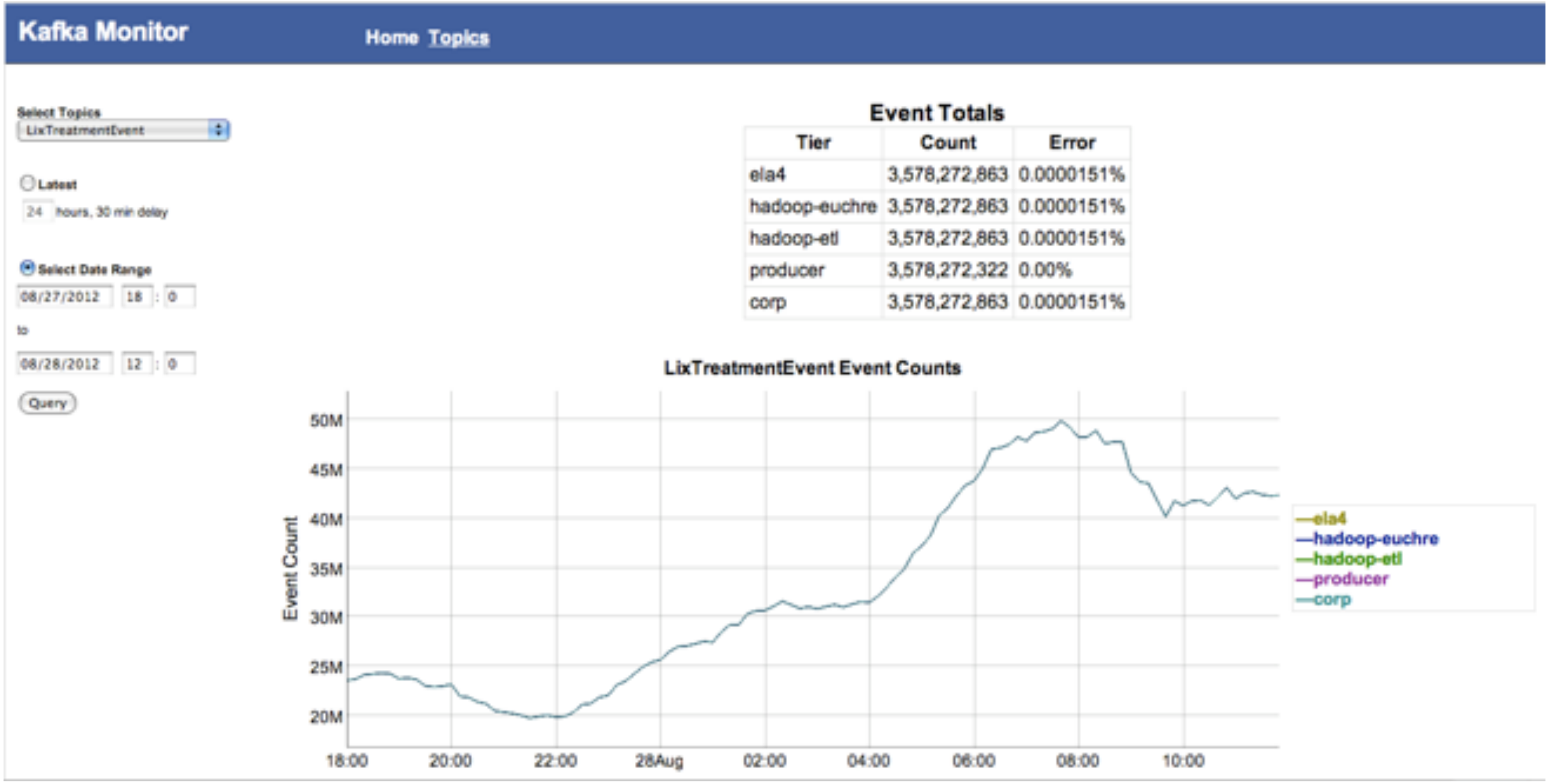
Does it work?

All messages sent must be delivered to all consumers (quickly)

Audit Trail

- Each producer, broker, and consumer periodically reports how many messages it saw
- Reconcile these counts every few minutes
- Graph and alert

Audit Trail


Kafka Monitor
Home Topics
Select Topics
LixTreatmentEvent
Latest
24 hours, 30 min delay
Select Date Range
08/27/2012 18 : 0
to
08/28/2012 12 : 0
Query
Event Totals
Tier Count Error
ela4 3,578,272,863 0.0000151%
hadoop-euchre 3,578,272,863 0.0000151%
hadoop-etl 3,578,272,863 0.0000151%
producer 3,578,272,322 0.00%
corp 3,578,272,863 0.0000151%
LixTreatmentEvent Event Counts
Event Count
50M
45M
40M
35M
30M
25M
20M
18:00
20:00
22:00
28Aug
02:00
04:00
06:00
08:00
10:00
—ela4
—hadoop-euchre
—hadoop-etl
—producer
—corp

Questions?