facebook

facebook

Realtime Apache Hadoop at Facebook

Jonathan Gray & Dhruba Borthakur
June 14, 2011 at SIGMOD, Athens

Agenda

1. Why Apache Hadoop and HBase?
2. Quick Introduction to Apache HBase
3. Applications of HBase at Facebook

Why Hadoop and HBase?

For *Realtime* Data?

facebook

Problems with existing stack

- MySQL is stable, but…
  - Not inherently distributed
  - Table size limits
  - Inflexible schema
- Hadoop is scalable, but…
  - MapReduce is slow and difficult
  - Does not support random writes
  - Poor support for random reads

Specialized solutions

- High-throughput, persistent key-value
  - Tokyo Cabinet
- Large scale data warehousing
  - Hive/Hadoop
- Photo Store
  - Haystack
- Custom C++ servers for lots of other stuff

What do we need in a data store?

- Requirements for Facebook Messages
  - Massive datasets, with large subsets of cold data
  - Elasticity and high availability
  - Strong consistency within a datacenter
  - Fault isolation
- Some non-requirements
  - Network partitions within a single datacenter
  - Active-active serving from multiple datacenters

HBase satisfied our requirements

- In early 2010, engineers at FB compared DBs
  - Apache Cassandra, Apache HBase, Sharded MySQL
- Compared performance, scalability, and features
  - HBase gave excellent write performance, good reads
  - HBase already included many nice-to-have features
    - Atomic read-modify-write operations
    - Multiple shards per server
    - Bulk importing
    - MapReduce

HBase uses HDFS

We get the benefits of HDFS as a storage system for free

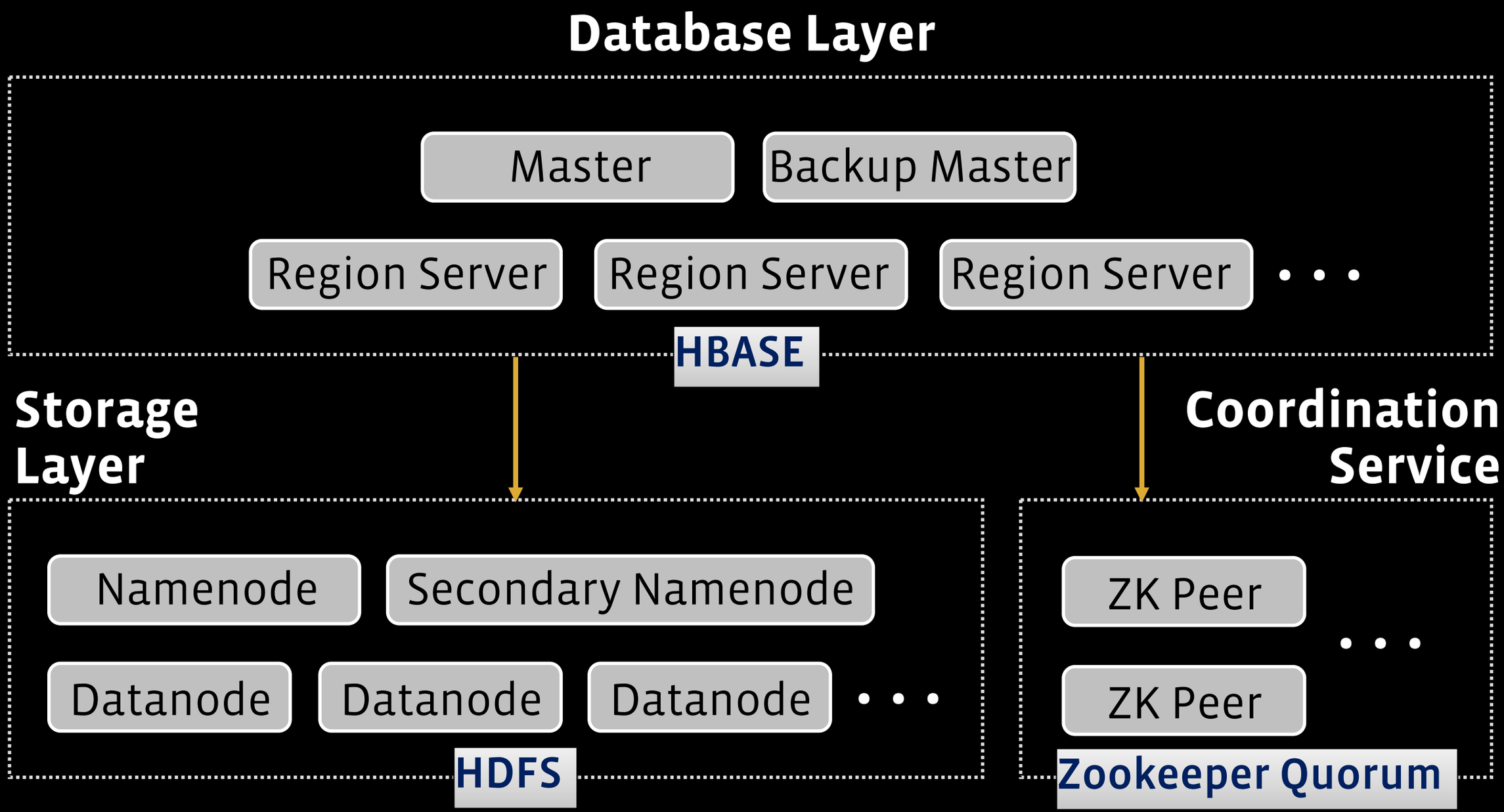
- Fault toleranceScalabilityChecksums fix corruptionsMapReduce
- Fault isolation of disksHDFS battle tested at petabyte scale at Facebook

Lots of existing operational experience

Apache HBase

- Originally part of Hadoop
  - HBase adds random read/write access to HDFS
- Required some Hadoop changes for FB usage
  - File appends
  - HA NameNode
  - Read optimizations
- Plus ZooKeeper!

HBase System Overview



HBase in a nutshell

- Sorted and column-oriented
- High write throughputHorizontal scalability Automatic failoverRegions sharded dynamically

Applications of HBase at Facebook

Use Case 1

Titan

(Facebook Messages)

The New Facebook Messages

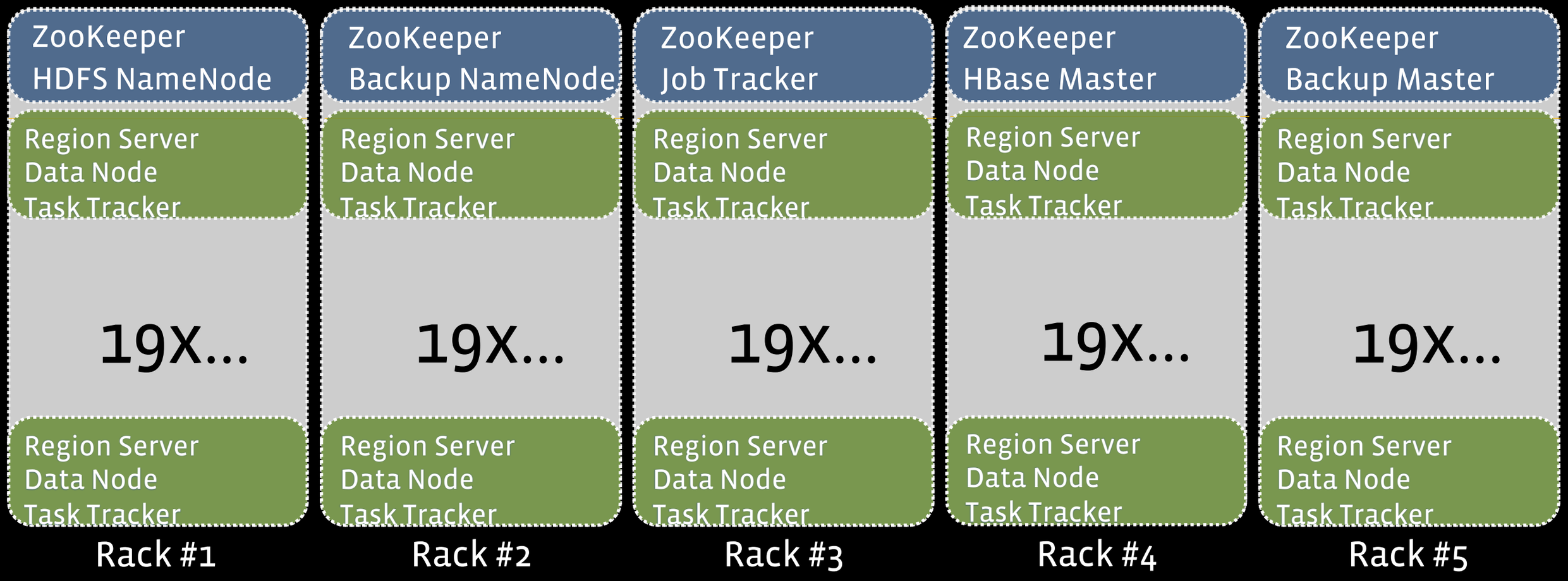
Messages **IM/Chat** **email** **SMS**

Facebook Messaging

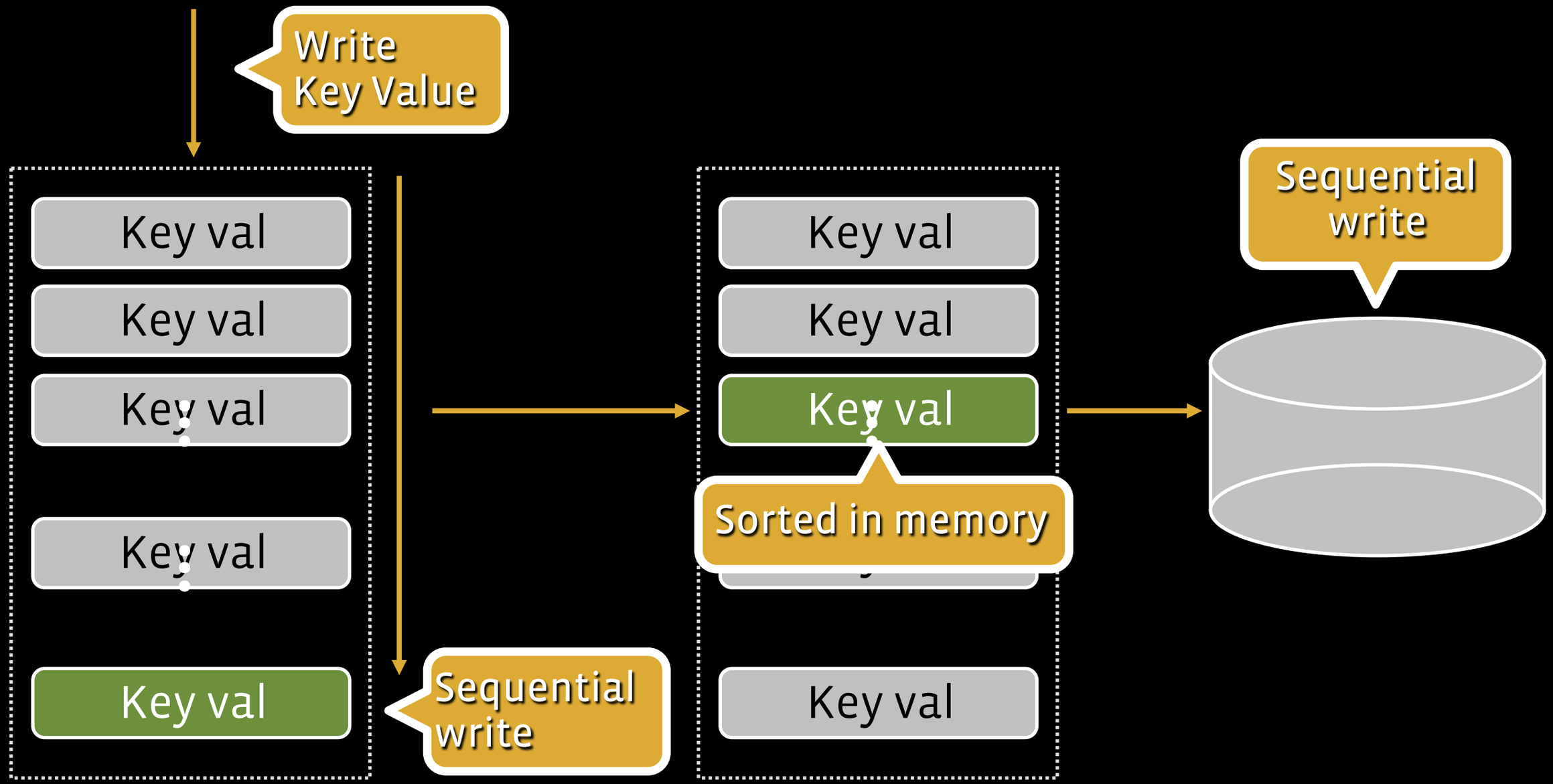
- **High write throughput** Every message, instant message, SMS, and e-mail Search indexes for all of the above
  - Denormalized schema
- **A product at massive scale on day one**
  - 6k messages a second
  - 50k instant messages a second
  - 300TB data growth/month compressed

Typical Cell Layout

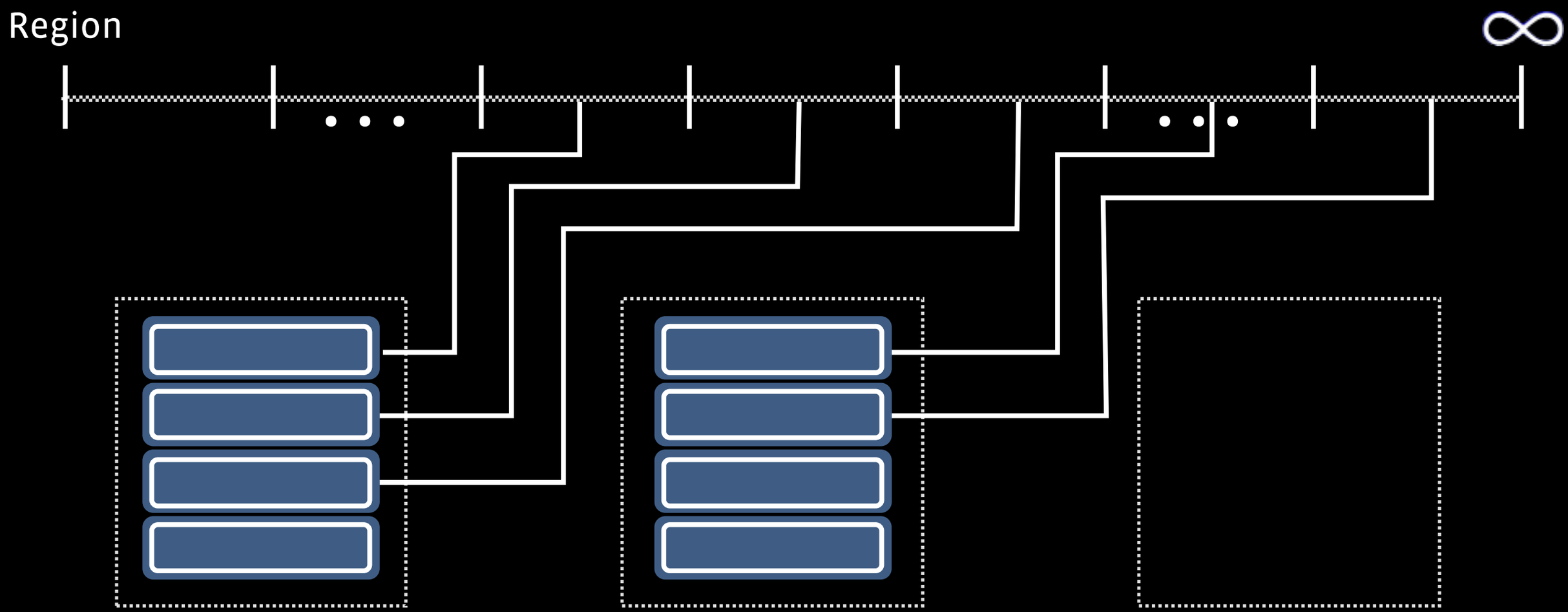
- Multiple cells for messaging
- 20 servers/rack; 5 or more racks per cluster
- Controllers (master/Zookeeper) spread across racks



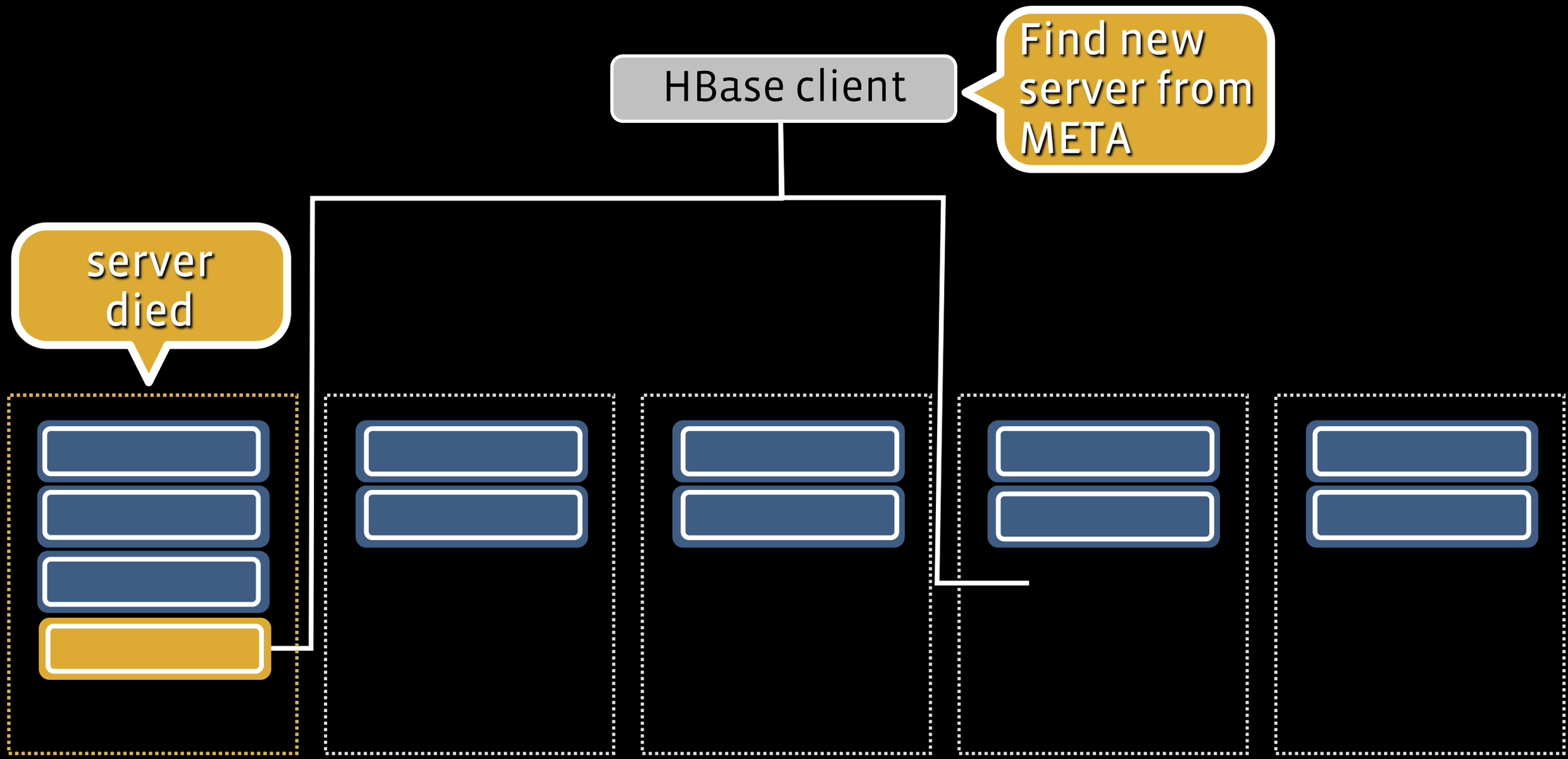
High Write Throughput



Horizontal Scalability



Automatic Failover



No physical data copy because data is in HDFS

Use Case 2

Puma

(Facebook Insights)

facebook

facebook

Puma

- Realtime Data Pipeline
  - Utilize existing log aggregation pipeline (Scribe-HDFS)
  - Extend low-latency capabilities of HDFS (Sync+PTail)
  - High-throughput writes (HBase)

- Support for Realtime Aggregation
  - Utilize HBase atomic increments to maintain roll-ups
  - Complex HBase schemas for unique-user calculations
  - Store checkpoint information directly in HBase

Puma as Realtime MapReduce

- Map phase with PTail
  - Divide the input log stream into N shards
  - First version only supported random bucketing
  - Now supports application-level bucketing
- Reduce phase with HBase
  - Every row+column in HBase is an output key
  - Aggregate key counts using atomic counters
  - Can also maintain per-key lists or other structures

Puma for Facebook Insights

- Realtime URL/Domain Insights
  - Domain owners can see deep analytics for their site
  - Clicks, Likes, Shares, Comments, Impressions
  - Detailed demographic breakdowns (anonymized)
  - Top URLs calculated per-domain and globally

- Massive Throughput
  - Billions of URLs
  - > 1 Million counter increments per second

Day | Week | Month | 05/08/2011 - 06/06/2011 | Export

Site Engagement

[redacted] Total Actions?

118x

Distribution on Facebook

[redacted] Total Impressions?

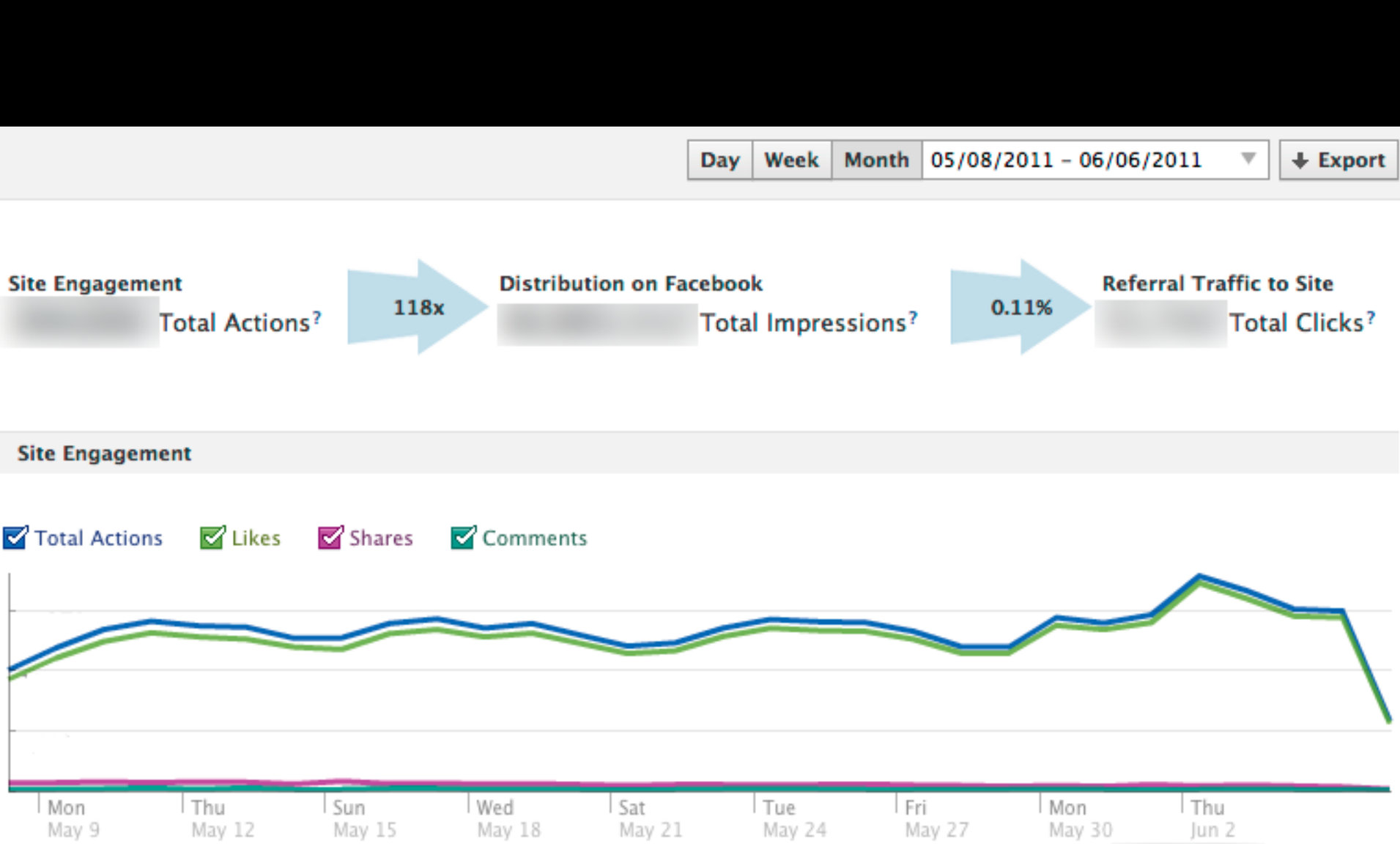
0.11%

Referral Traffic to Site

[redacted] Total Clicks?

Site Engagement

Total Actions | Likes | Shares | Comments



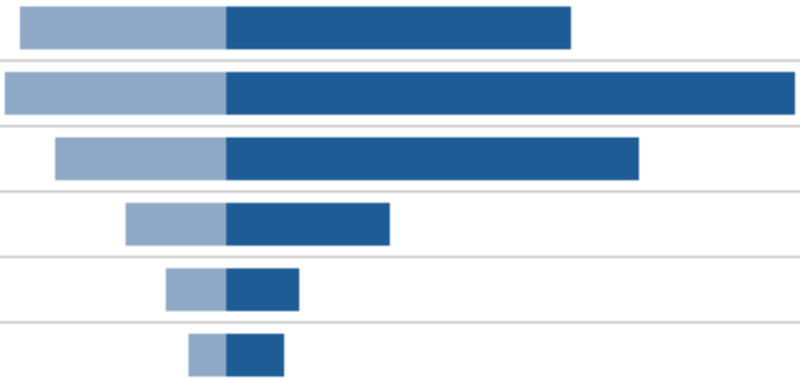
[redacted] Total Actions?

[redacted] Likes?

Shares?

Comments?

Age	Female (31%)	Male (62%)
13–17	7.9%	13%
18–24	8.5%	22%
25–34	6.6%	16%
35–44	3.9%	6.3%
45–54	2.3%	2.8%
55+	1.5%	2.3%



Language	
English (US)	(34%)
Spanish	(12%)
Arabic	(6.1%)
Turkish	(6%)
English (UK)	(5.3%)
French (France)	(5.2%)
Spanish (Spain)	(4.6%)
Indonesian	(3.9%)
Italian	(3.7%)
German	(2.7%)
Portuguese (Brazil)	(2.6%)
Thai	(1.2%)
Russian	(0.96%)
Polish	(0.91%)
More	

Country	
United States	(13%)
Turkey	(5.9%)
Philippines	(5%)
Indonesia	(4.5%)
India	(4.4%)
Mexico	(4.3%)
Egypt	(4.1%)
Italy	(3.8%)
Brazil	(2.7%)
Argentina	(2.7%)
Germany	(2.6%)
Malaysia	(2.6%)
United Kingdom	(2.3%)
Spain	(1.8%)
More	

Use Case 3

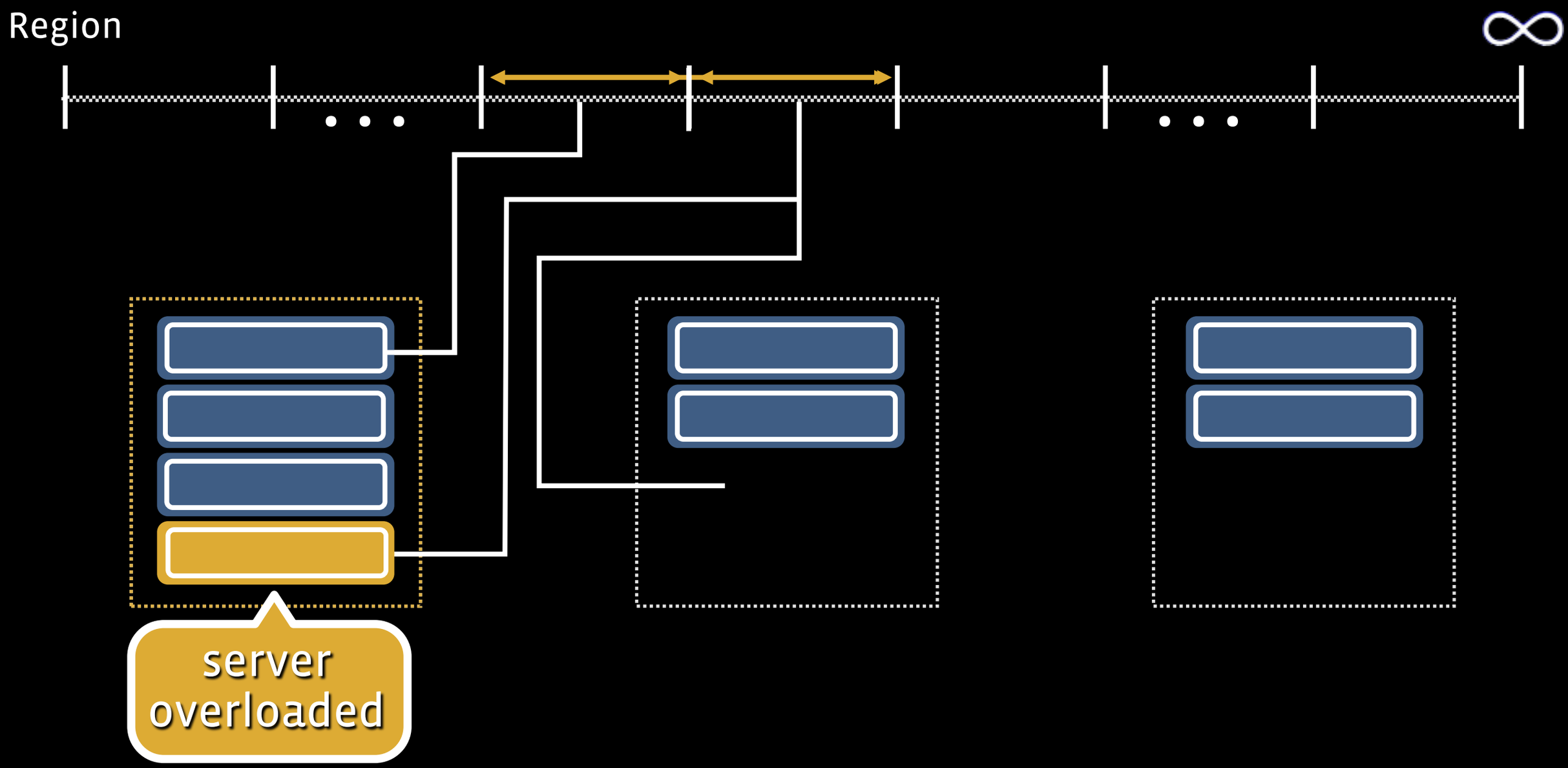
ODS

(Facebook Internal Metrics)

ODS

- **Operational Data Store**
  - System metrics (CPU, Memory, IO, Network)
  - Application metrics (Web, DB, Caches)
  - Facebook metrics (Usage, Revenue)
    - Easily graph this data over time
    - Supports complex aggregation, transformations, etc.
- **Difficult to scale with MySQL**
  - Millions of unique time-series with billions of points
  - Irregular data growth patterns

Dynamic sharding of regions



Future of HBase at Facebook

User and Graph Data in HBase

HBase for the important stuff

- Looking at HBase to augment MySQL
  - Only single row ACID from MySQL is used
  - DBs are always fronted by an in-memory cache
  - HBase is great at storing dictionaries and lists
- Database tier size determined by IOPS
  - HBase does only sequential writes
  - Lower IOPs translate to lower cost
  - Larger tables on denser, cheaper, commodity nodes

Conclusion

- Facebook investing in Realtime Hadoop/HBase
  - Work of a large team of Facebook engineers
  - Close collaboration with open source developers
- Much more detail in Realtime Hadoop paper
  - Technical details about changes to Hadoop and HBase
  - Operational experiences in production

Questions?

jgray@fb.com

dhruba@fb.com