Phoenix

We put the SQL back in NoSQL

James Taylor
jtaylor@salesforce.com

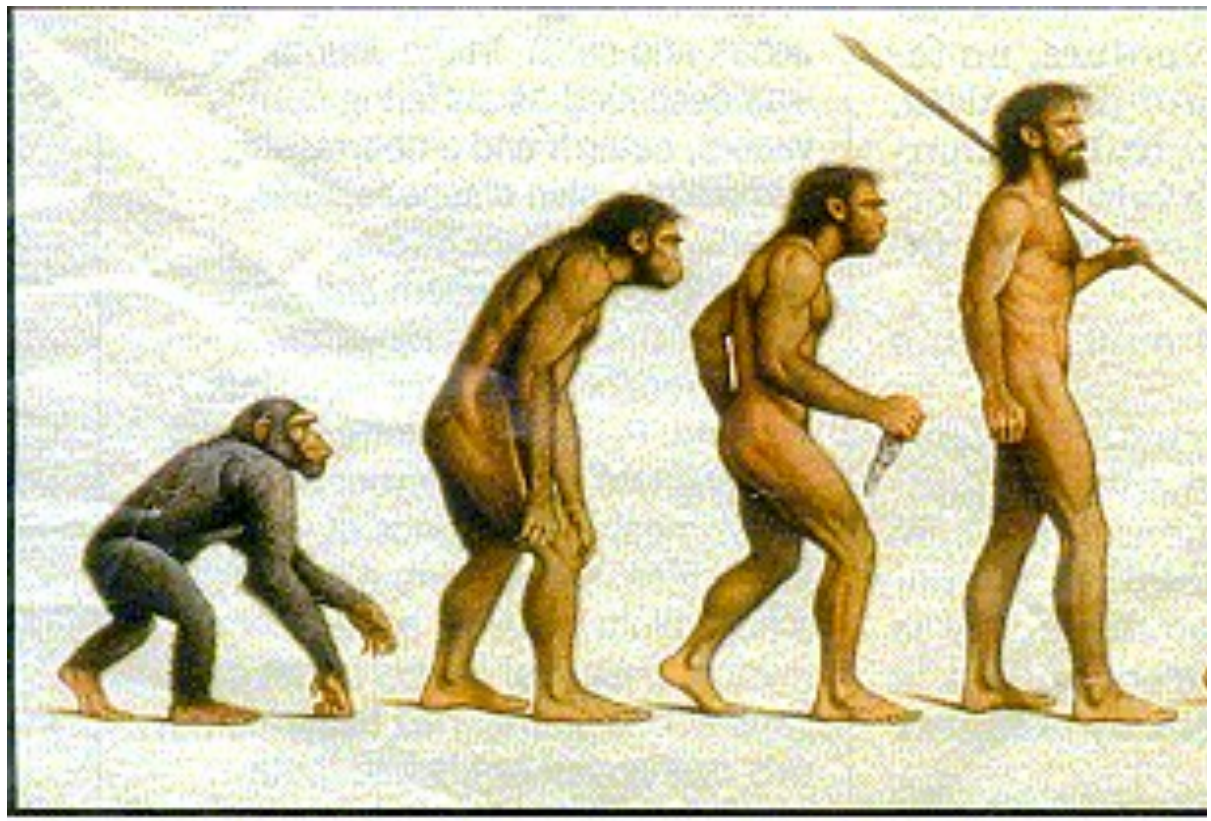
salesforce.com

In the dawn of time…

Relational Databases were invented

But we all know the problems folks ran into

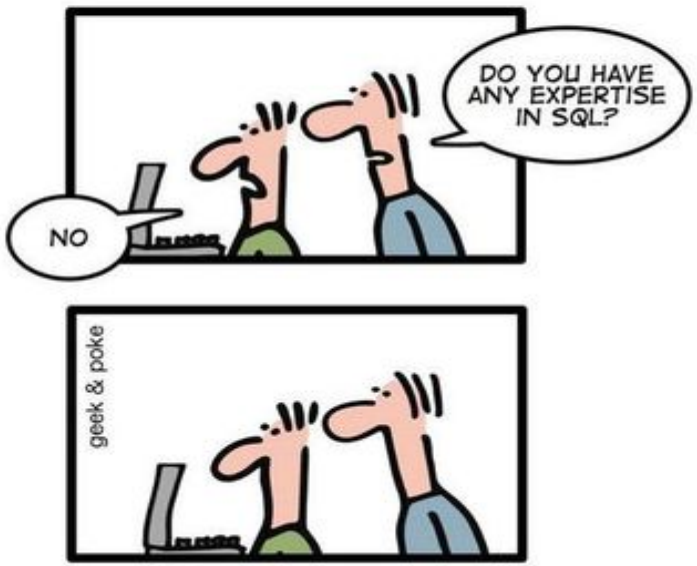
And then there was HBase



And it was good

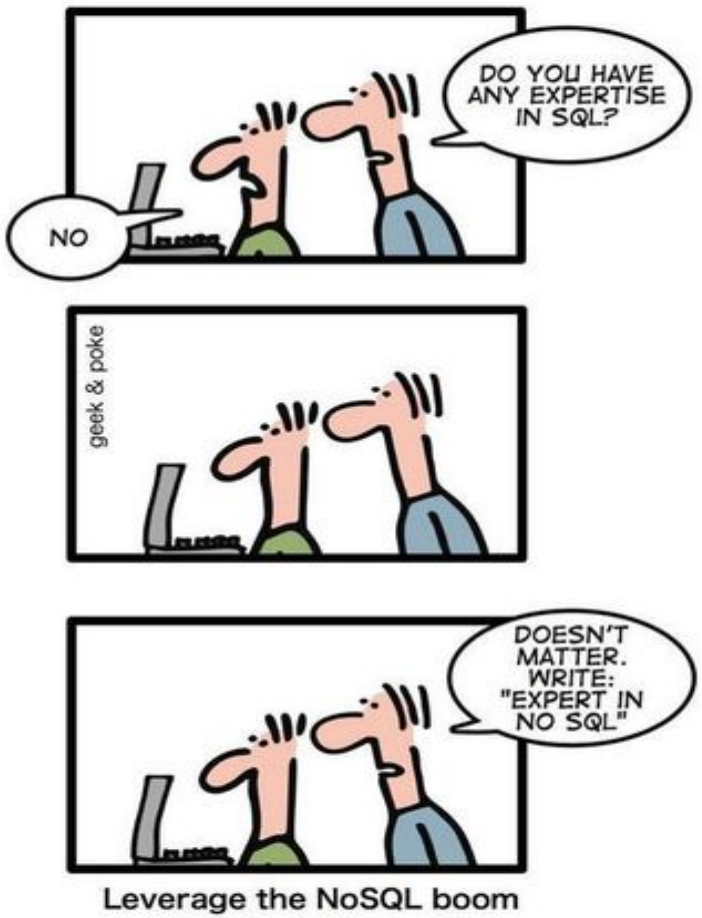
1. Horizontally scalable

And it was good



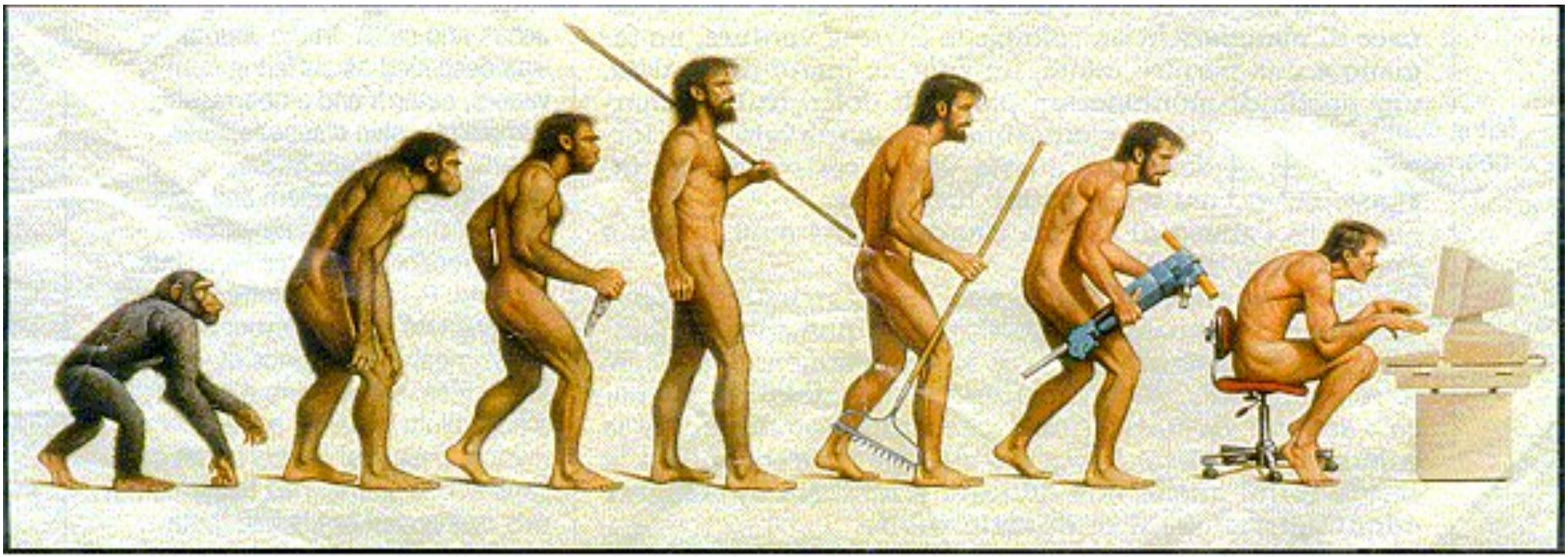
1. Horizontally scalable
2. Maintains data locality

And it was good



1. Horizontally scalable
2. Maintains data locality
3. Runs on commodity hardware

But somewhere, something terrible went wrong



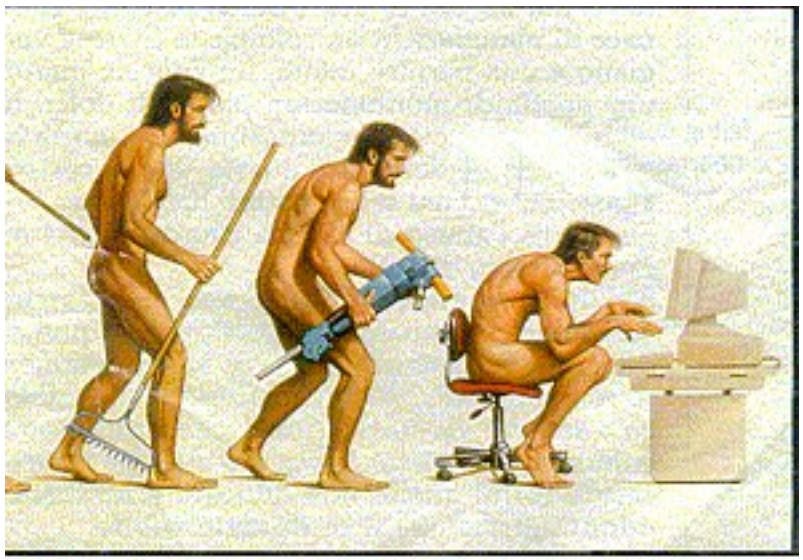
But somewhere, something terrible went wrong

1. It takes too much expertise to write an application

But somewhere, something terrible went wrong

1. It takes too much expertise to write an application
2. It takes too much code to do anything

But somewhere, something terrible went wrong



1. It takes too much expertise to write an application
2. It takes too much code to do anything
3. Your application is tied too closely with your data model

What is Phoenix?

- SQL skin for HBase

What is Phoenix?

- SQL skin for HBase
- An alternate client API

What is Phoenix?

- SQL skin for HBase
- An alternate client API
- An embedded JDBC driver that allows you to run at HBase native speed

What is Phoenix?

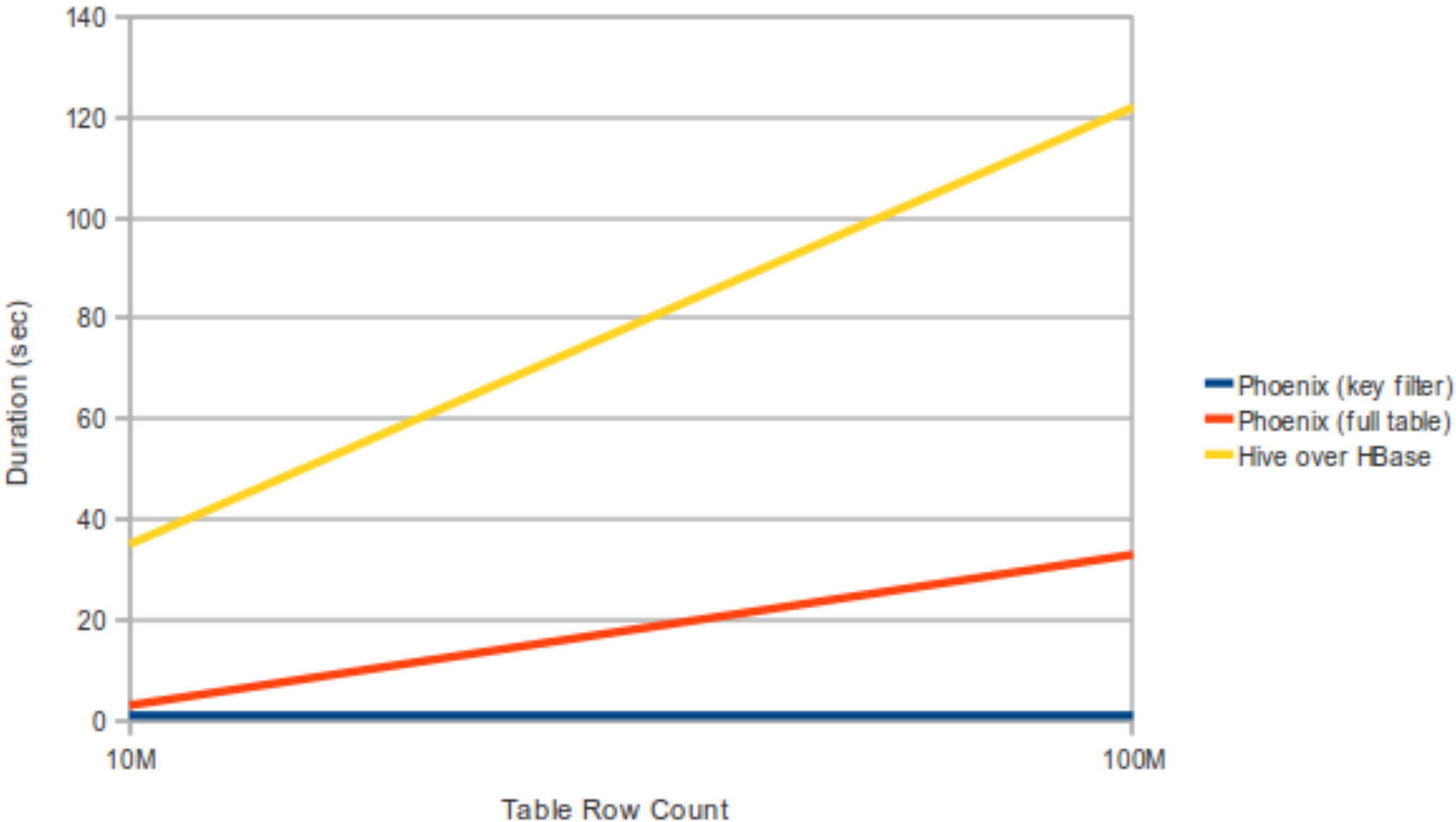
- SQL skin for HBase
- An alternate client API
- An embedded JDBC driver that allows you to run at HBase native speed
- Compiles your SQL into native HBase calls

What is Phoenix?

- SQL skin for HBase
- An alternate client API
- An embedded JDBC driver that allows you to run at HBase native speed
- Compiles your SQL into native HBase calls

so you don't have to!

Phoenix Performance



Why SQL for HBase?

- Broaden HBase adoption
  - Give folks an API they already know

Why SQL for HBase?

- Broaden HBase adoption
  - Give folks an API they already know
- Reduce the amount of code users need to write

```
SELECT TRUNC(date,'DAY'), AVG(cpu_usage)
FROM web_stat
WHERE domain LIKE 'Salesforce%'
GROUP BY TRUNC(date,'DAY')
```

Why SQL for HBase?

- Broaden HBase adoption
  - Give folks an API they already know
- Reduce the amount of code users need to write

```
SELECT TRUNC(date,'DAY'), AVG(cpu_usage)
FROM web_stat
WHERE domain LIKE 'Salesforce%'
GROUP BY TRUNC(date,'DAY')
```

- Performance optimizations transparent to the user
  - Aggregation
  - Skip Scan
  - Secondary indexing (soon!)

Why SQL for HBase?

- Broaden HBase adoption
  - Give folks an API they already know
- Reduce the amount of code users need to write

```
SELECT TRUNC(date,'DAY'), AVG(cpu_usage)
FROM web_stat
WHERE domain LIKE 'Salesforce%'
GROUP BY TRUNC(date,'DAY')
```

- Performance optimizations transparent to the user
  - Aggregation
  - Skip Scan
  - Secondary indexing (soon!)
- Leverage existing tooling
  - SQL client/terminal
  - OLAP engine

Query Processing

Over metrics data for clusters of servers with a schema like this:

Server Metrics	
HOST	VARCHAR
DATE	DATE
RESPONSE_TIME	INTEGER
GC_TIME	INTEGER
CPU_TIME	INTEGER
IO_TIME	INTEGER
…	

Row Key

Query Processing

With data that looks like this:

SERVER METRICS			
HOST	DATE	RESPONSE_TIME	GC_TIME
ny1-s1	Jun 5 10:10:10.234	1234	
ny1-s1	Jun 5 11:18:28.456	4560	
…			
sf1-s1	Jun 5 10:10:10.234	2345	
sf1-s1	Jun 6 12:46:19.123	1003	
sf7-s20	Jun 4 08:23:23.456	5002	
…			

Scenario 1
Chart Response Time Per Cluster

```
SELECT host, trunc(date,'HOUR'),
    min(response_time), max(response_time)
FROM server_metrics
WHERE date > CURRENT_DATE() – 1
AND substr(host, 1, 3) IN ('sf1', 'sf3, 'sf7')
GROUP BY substr(host, 1, 3), trunc(date,'HOUR')
```

Phoenix Roadmap

- Secondary Indexing
- Hash Joins
- Apache Drill integration
- Count distinct and percentile
- Derived tables
    - SELECT * FROM (SELECT * FROM t)
- Cost-based query optimizer
- OLAP extensions
    - WINDOW, PARTITION OVER, RANK
- Monitoring and management
- Transactions

Thank you!
Questions/comments?