

Scaling MySQL with TokuDB®

Tim Callaghan and Lawrence Schwartz

Webinar

Tokutek®

Scaling MySQL with TokuDB®

TokuDB[®] Product Overview

Unmatched Speed. Maximum Scalability. Exceptional Agility.

Lawrence Schwartz
Lawrence@tokutek.com

MySQL Performance Issues

The Problem

MySQL enables your business;
its use continues to grow...



...but over time performance,
manageability suffer



- Inserts, indexes can't keep up
- Query performance lags
- Storage requirements increase
- Fragmentation leads to dump/reloads
- Schema changes result in downtime
- Scalability remains limited

Common Compromises

Partitions / Shards

- ✓ Fast drops
- Challenging to manage
- Poor performance

Operational headache

HW (RAM/SSD)

- ✓ Faster than disks
- Expensive
- Inefficient DB bandwidth use

Inefficient, cost prohibitive

Data Warehouse

- ✓ Handles big data
- Hard to update
- ACID compliance

Non-interactive, lack of ACID

NoSQL

- ✓ Handles unstructured data
- Not ACID, often no indexes
- No SQL language, expertise

No ACID, Lack of indexes, SQL

A True Solution

TokuDB®

- ✓ 20x to 80x Insertion Speed
- ✓ Hot Schema Changes
- ✓ Full MySQL Compatibility
- ✓ Ad-Hoc Query Performance
- ✓ 5x to 15x compression
- ✓ Scalability to tens of TBs
- ✓ No de-fragmentation
- ✓ ACID, MVCC, SAVEPOINTS
- ✓ MariaDB

*Turns MySQL to NewSQL with
revolutionary Fractal Tree®*

<intent>
MEDIA

JAWA

Limelight
NETWORKS

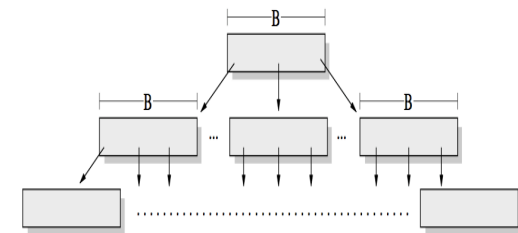
PROFILE TECHNOLOGY

KAYAK

Tokutek

The Need for a Better Storage Engine

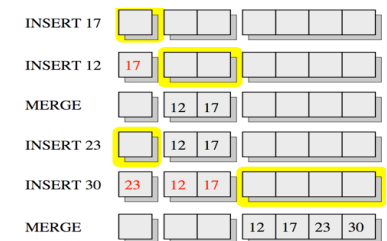
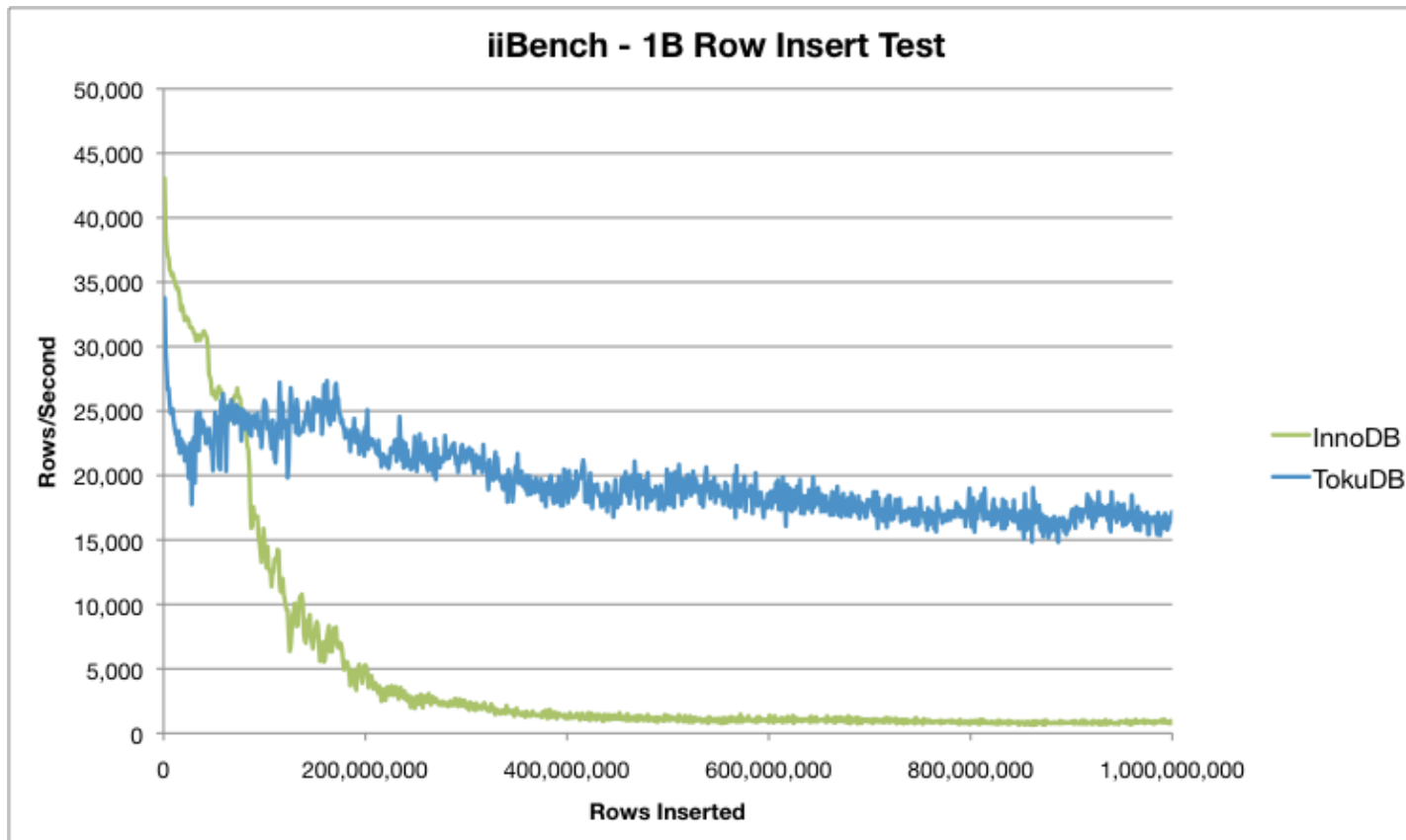
- MySQL is critical part of Web 2.0 LAMP
 - *MySQL continues to have the largest mindshare in the open source database market* - Forrester
- However, all major vendors use obsolete B-tree indexes
 - Developed in the 1970s
 - Struggle under high insertion rates
 - Age, fragment, requiring off-line maintenance
- MySQL often kept to 0.5 TB or less
 - Larger sizes lead to severe operational problems
 - Hours to days for index rebuilding
 - Downtime for schema changes
 - Offline fragmentation clean up
 - Difficult to maintain partitions



B Trees
Proven for sequential loads, but performance drops off with age and size

Addressing the Root Issue - Better Indexes

- TokuDB® Fractal Tree® indexes change the game with huge insertion rates in scalable systems



Fractal Tree indexes
Continuous rebalancing and aggregation eliminates fragmentation, maximizes disk I/O

Indexed insertions

Insert 1 billion rows into a table maintaining three multi-column secondary indexes.

Terminal rate (last 10MM rows):

InnoDB®: 876 inserts/sec

TokuDB: 16,507 inserts/sec

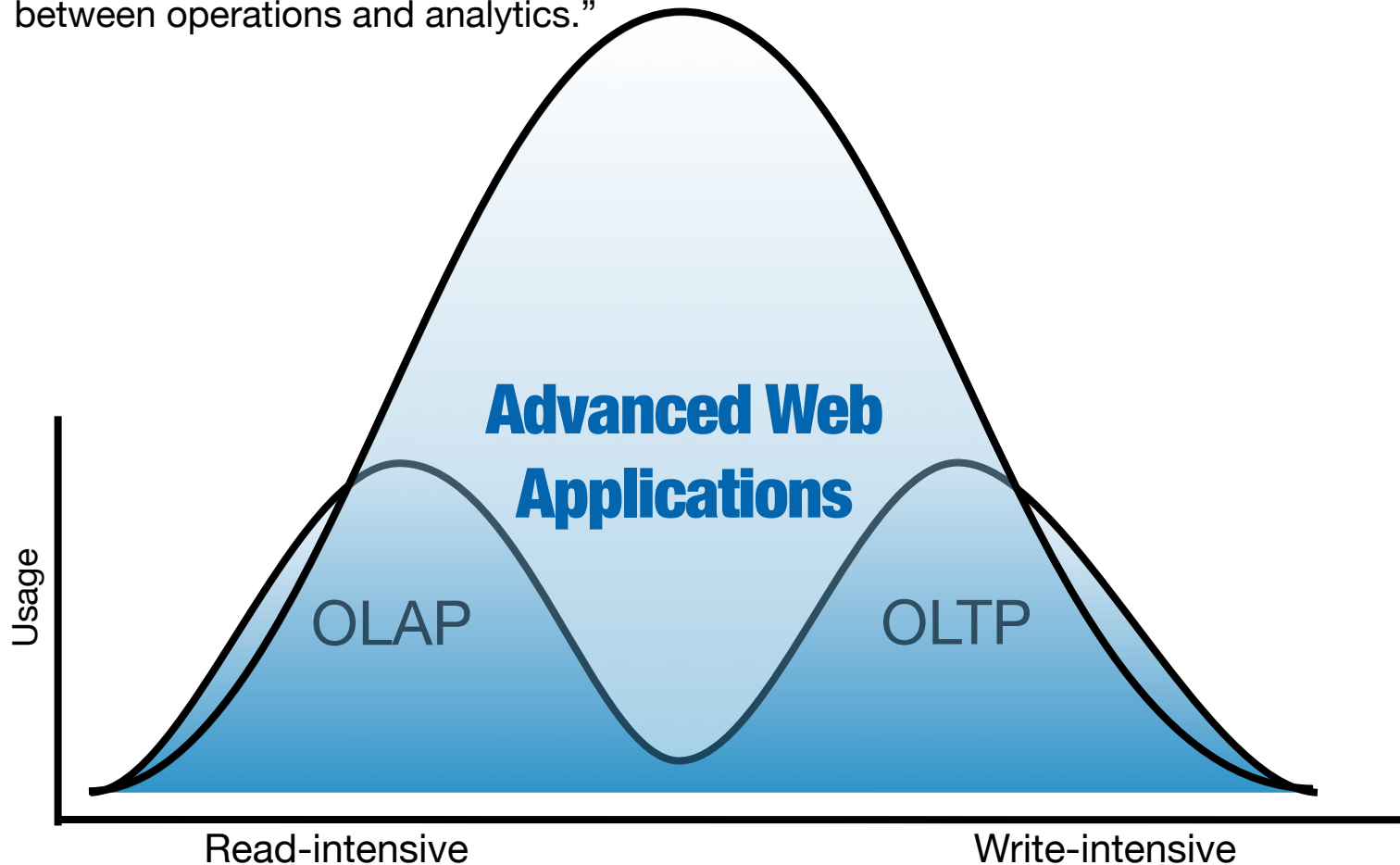
(19x faster!)

Tokutek

More on how FT indexes work: <http://goo.gl/Smu6H>

Tokutek's Target Market

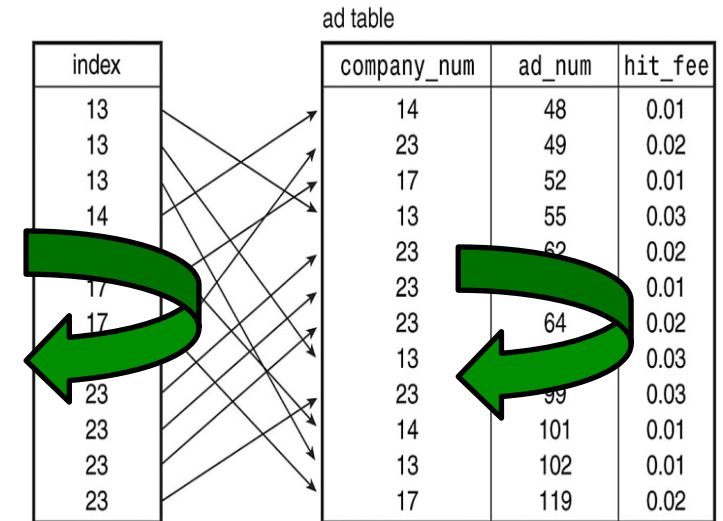
- An End to the Either-Or
 - Matt Aslett / The 451 Group:
 - “While TokuDB is effectively an operational database technology, it does blur the lines between operations and analytics.”





Agility at Scale: Hot Indexing

- Allows for **concurrent operation** on the database and index
- Enables **ad hoc queries** to run fast with real-time, **optimized index support**
- Brings familiar **Enterprise Database** online operations to to MySQL



Hot Index

Concurrent DB read/write operations while the index is being created

TokuDB® v5.0: The only MySQL Storage Engine with Hot Indexing



Agility at Scale: Hot Column Addition

- Provides capability to **add or drop a column** to a database **as a fast operation**
- Enables database administrators to **rapidly define and add new fields**
- Allows for **much larger tables** to be created given **simplified maintenance**



Hot Column Addition

Online column addition allows for quick online data re-organization

TokuDB® v5.0: The only MySQL Storage Engine with Hot Column Addition

TokuDB® – A True Solution

	InnoDB	TokuDB
Index Type	B-tree	Fractal Tree® index
Insertion Rate at Scale	100s / second	10,000s / second
Compression	~2x	5 - 15x or more
Hot Indexing	No (hours+)	Yes
Hot Column Addition/Deletion	No (hours+)	Yes
Fast Loader	No	Yes (parallelized for full multi-core utilization)
Fragmentation Immunity	No	Yes (no dump/reload downtime)
Clustering Indexes	Primary key Only	Multiple
Fast Recovery Time	No	Yes
MariaDB Compatible	Yes	Yes
ACID	Yes	Yes
MVCC	Yes	Yes

From: <http://www.tokutek.com/resources/tokudb-vs-innodb/>

Feedback on TokuDB® v5.0

[Josh Hartman, CTO at Intent Media](#)

"Managing terabytes of data now is as easy as managing 50 gigabytes was at the beginning."

[Ernie Souhrada, Chief IT Architect at Jawa](#)

"With the introduction of Hot Schema Changes in TokuDB v5.0, Tokutek makes deriving value out of a large MySQL database simple, giving us the option to much more easily analyze our data and generate value for our business."

Customers

Press

[Joab Jackson, IT News](#)

"Tokutek [now] offers an alternative MySQL storage engine for on-the-fly schema changes"

[Mike Vizard, CTO Edge](#)

"The company has developed an index engine for MySQL based on Fractal Tree® technology that allows MySQL performance to scale from gigabytes to terabytes."

[Joseph Martins, Data Mobility Group](#)

"With the introduction of agile and online operations (Hot Schema Changes) in TokuDB v5.0, Tokutek ensures MySQL can be used in environments that were previously limited to the largest of enterprise database vendors."

[Stuart Miniman, Wikibon](#)

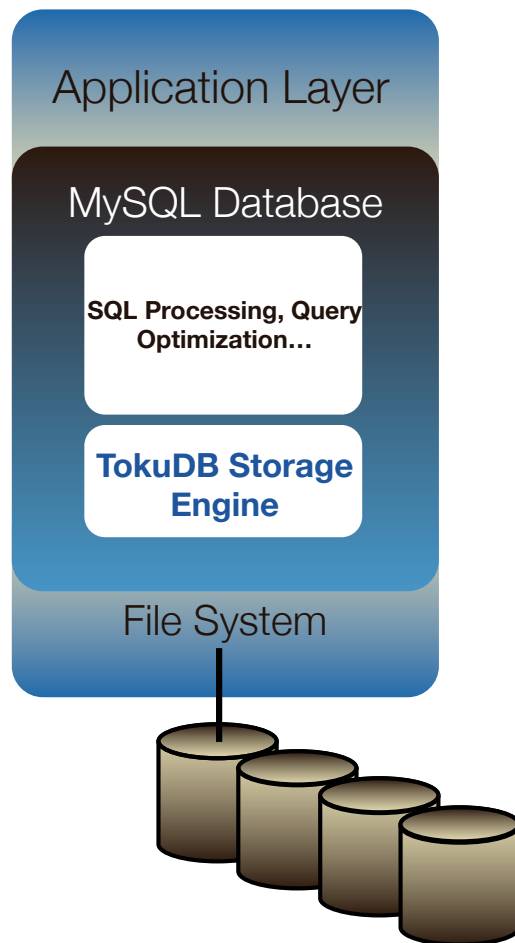
"The growth of data, especially in real-time Web 2.0 environments, can be a burden or an opportunity for new products and new revenue. MySQL users should consider TokuDB to increase agility, speed and scalability."

Analysts

Tokutek®

TokuDB®: Speed, Scalability and Agility

A highly scalable, zero maintenance downtime, MySQL Storage Engine that offers on-the-fly schema modification and powerful indexing-based query acceleration.



- Use your existing MySQL code, application logic
 - TokuDB is a drop-in *storage engine* for MySQL and MariaDB
- Fast, Agile, and scalable for Big Data
 - **Insertion Speed:** 80x faster index inserts (customer measurement)
 - **Query Performance:** Fast ad-hoc and rich queries
 - **Hot Schema Changes:** Hot indexing and hot column addition
 - **Compression:** 5 – 15x at customer sites
 - **Scalability:** Predictable performance at 10+ billion rows
 - **Fast Dataset Loading:** 1+ million rows / second
 - **Continuous Operation:** Avoid downtime for de-fragmentation
 - **Support:** ACID and MVCC Compliant, SAVEPOINTS **Tokutek**

TokuDB[®] Technical Overview

Tim Callaghan
Tokutek Technical Services
tim@tokutek.com

A Few Myths about Big* Databases

Big* = significantly larger than RAM

Myth: Big Databases are Inflexible

- Applications running on large transactional and operational databases often require maintenance windows
 - Adding indexes
 - Adding columns
 - Defragmenting tables and indexes

Important Notice

The system will be unavailable on
Saturday, November 19
from 1:00am to 3:00am
for maintenance.

Myth: Big Databases are Slow

- Existing technologies provide a no-win situation

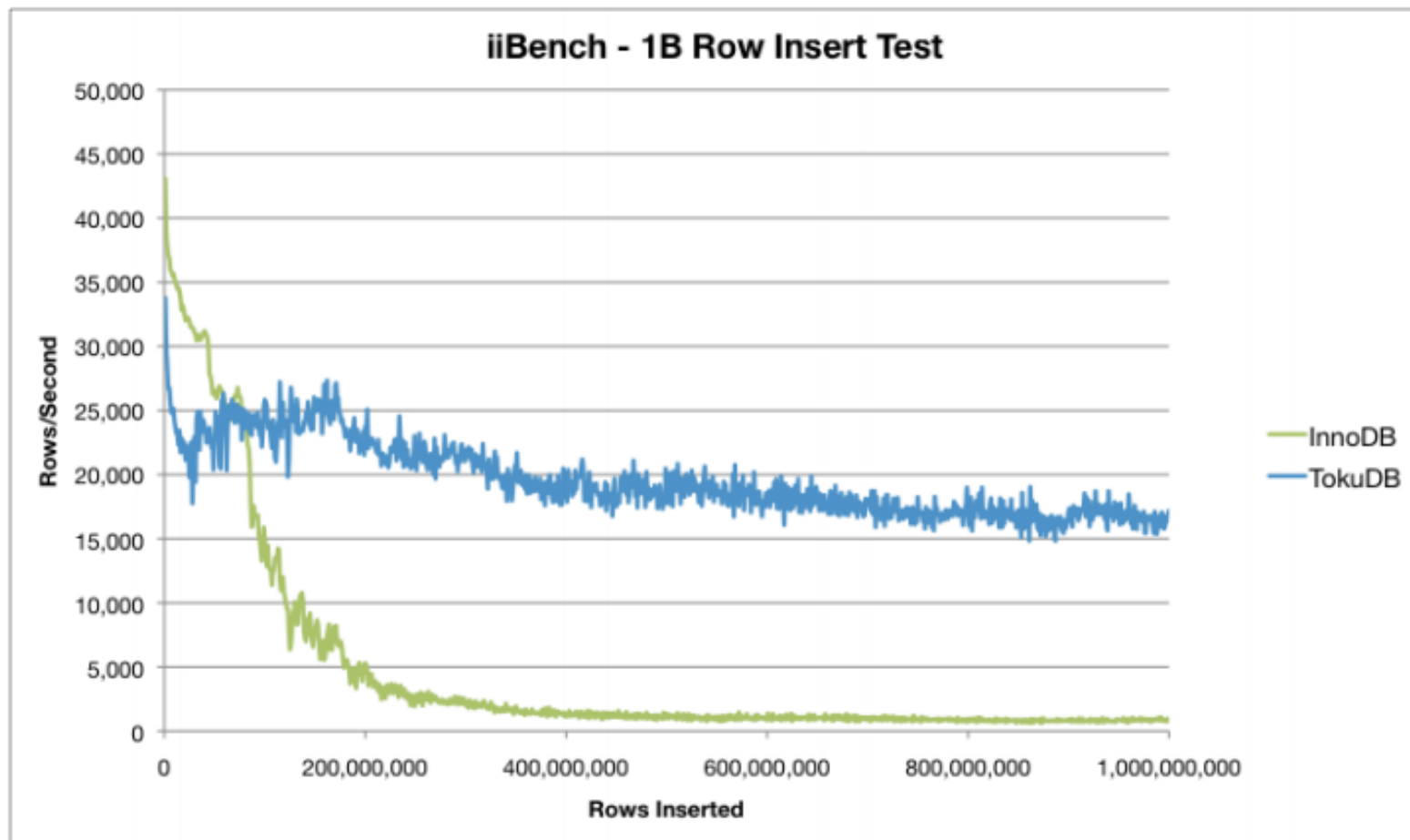
	Insert Performance	Query Performance
More Indexes	Down	Up
Less Indexes	Up	Down

TokuDB Benefits

- Performance
 - Storage engine capabilities
 - Multiple clustering keys
 - Data loader
- Agility
 - Online operations
- Management at Scale
 - OLTP and OLAP
 - Compression
 - No fragmentation
 - Checkpoints and recovery
 - Progress tracking

Performance: Storage engine

- High-performance insert/update/delete for large databases while maintaining indexes



Performance: Storage engine

- TokuDB uses Tokutek's Fractal Tree[®] technology
 - Large block size enables high compression and excellent range query performance
 - Internal nodes are similar to B-trees (keys and pointers) but also contain message buffers
 - A great writeup is <http://goo.gl/Smu6H>
 - Lots of great information available on our website at tokutek.com/resources/technology

Performance: Multiple clustering keys

- Like InnoDB
 - All row data is stored by primary key (clustered)
 - Secondary keys store primary key
 - Lookups by secondary key actually require a “join” to retrieve the rest of the row
- Unlike InnoDB, TokuDB allows secondary keys to be clustered
 - Compression saves space, indexes are small
 - All columns of the table are immediately available on secondary key lookup

Performance: Data loader

- Traditional MySQL loader is single threaded
- TokuDB loader uses all available cores to parallelize load process
- Index creation uses same technology
- We benchmarked on an 8 core machine and measured an 8x improvement in TPC-C data load time
 - <http://goo.gl/CSh5L>

Management: Online operations

- Common schema changes can take hours in MySQL
 - Adding or dropping a column
 - Adding an index
- Also, the table is unavailable to all other operations during the process
- As a workaround, people generally
 - Create the new index on a replication slave
 - Once complete, allow it to catch up to the master
 - then promote the slave to master
- Many are considering NoSQL (schema-less) technologies to overcome these limitations

Management: Hot column addition/deletion

- “alter table t1 add column c4 bigint;”
- Traditional MySQL
 - Locks the table and performs a “select into ...” to the new table structure
 - Indexes get rebuilt as well
 - No access to the table allowed during the process
- TokuDB
 - Creates and addcolumn() message and returns
 - Over time, the column is physically added to the actual rows

Management: Hot column addition/deletion

- 122mm row "air traffic data" table, add column
 - InnoDB 5.1 plugin took 17 hours, 44 minutes
 - table was locked the entire time
 - TokuDB 5.0 took 3 seconds
- details at <http://goo.gl/sEzx1>

Management: Hot indexing

- `“create index c4_idx on t1(c4);”`
- Traditional MySQL
 - Locks the table and creates the index
 - No access to the table allowed during the process
- TokuDB
 - Begins creating the index in the background
 - Uses TokuDB parallelized loader technology
 - Index is available to MySQL when finished
 - Accurate progress via `“show processlist;”`

Management: Hot indexing

- 122mm row "air traffic data" table, create new index
 - InnoDB 5.1 plugin took 31 minutes, 34 seconds
 - table was locked the entire time
 - TokuDB 5.0 took 9 minutes, 30 seconds
 - table was locked for under 2 seconds
 - more CPU cores would build the new index even faster
- details at <http://goo.gl/ffW4E>

Management: OLTP and OLAP

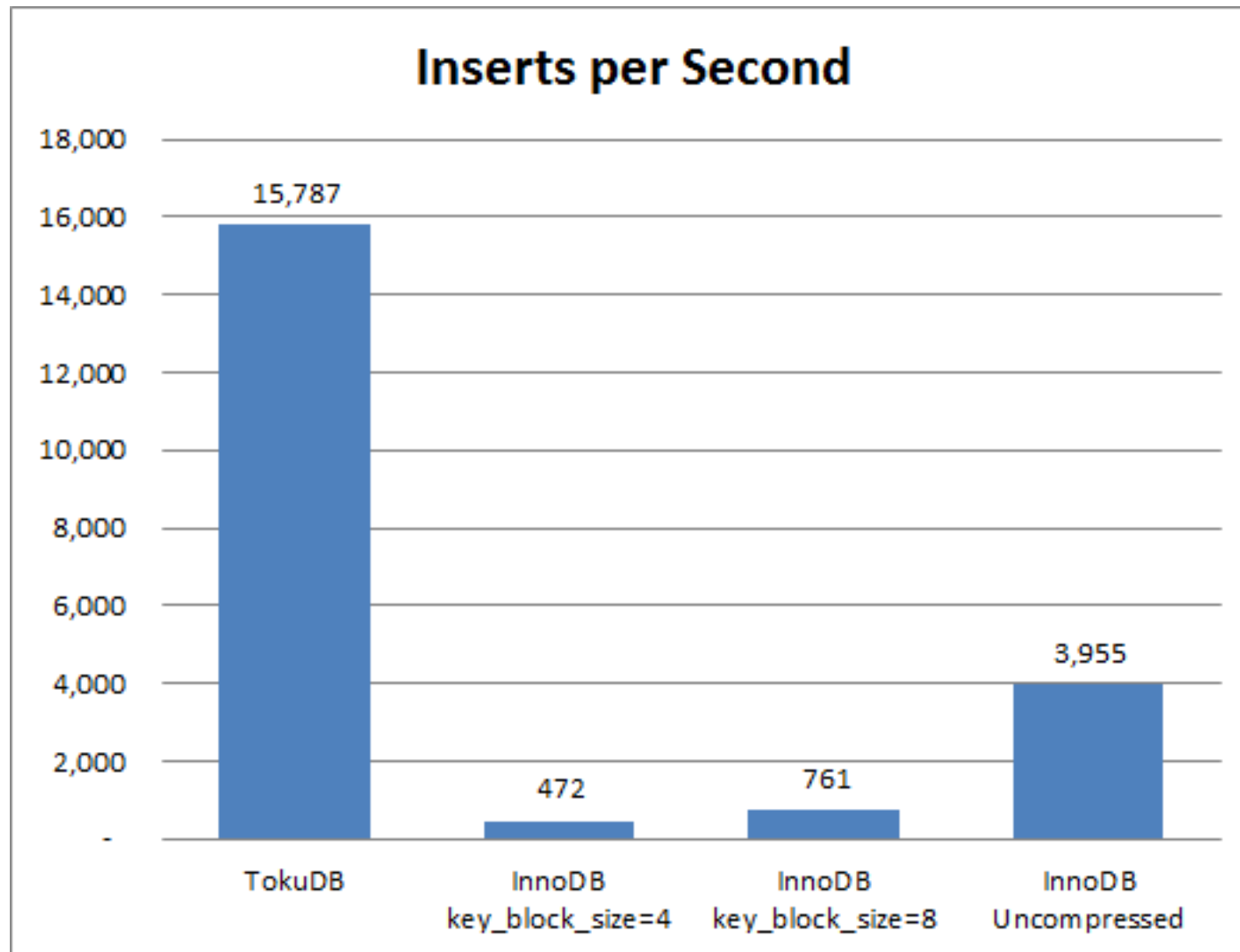
- “Hybrid” data implementations are becoming the norm
 - 1 database for OLTP, another for OLAP
- Usually because performance degrades as the database gets large
- Running a single TokuDB database means more timely analytical information and less moving parts

Management: Compression

- TokuDB has a larger block size than InnoDB, compression ratios are higher
- 5x to 15x can be achieved
- TokuDB compression is always enabled, no knobs or performance penalties

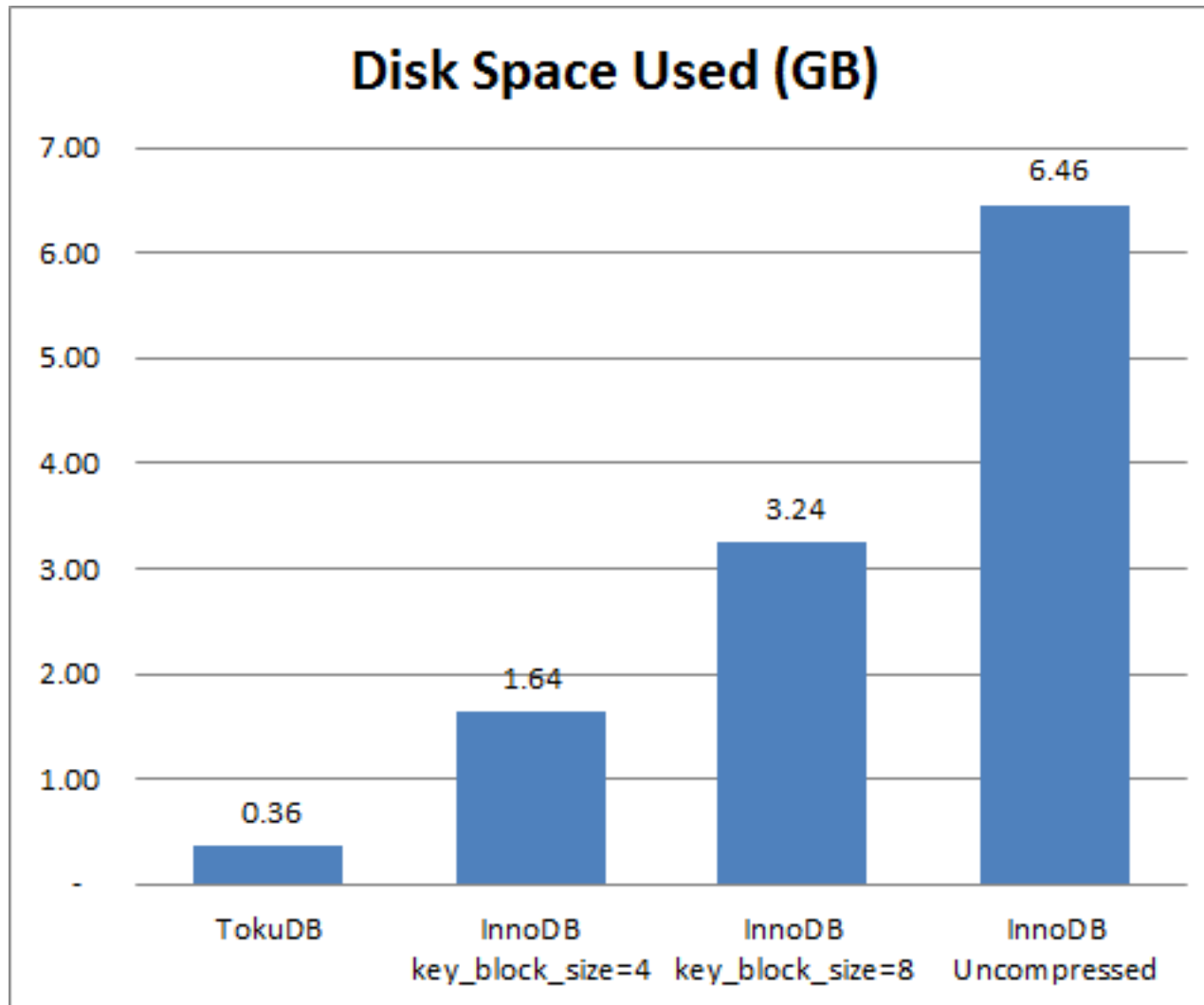
Management: Compression

- Compression performance penalties, iiBench



Management: Compression

- Compression space savings, log data



Management: Fragmentation

- Over time, B-trees with small block sizes suffer from fragmentation
- Best practices are dump/reload or “optimize table”
- Requires maintenance windows during which table is unavailable
- Fractal Trees do not fragment
 - 4MB compressed blocks vs. 16KB uncompressed
 - Far less random IO for range queries

Management: Checkpoints and recovery

- TokuDB performs a checkpoint every 60 seconds
 - Cost of checkpoint work is reduced
 - Also enables fast recovery time (less than 1 minute)
 - 60 second time period is user definable
- Be careful, I've run benchmarks on large servers (> 48GB) with other storage engines that show long periods of inactivity when checkpointing
 - see Vadim Tkachenko's blog at <http://www.mysqlperformanceblog.com/2011/09/18/disaster-mysql-5-5-flushing/>

Management: Progress tracking

- “show processlist”
- long running processes show accurate % complete
 - Data loader
 - Index creation

Deployment and Other Details

- 64-bit Linux only
 - Developed and tested on CentOS 5.5
 - Other Linux distros in production
- Physical hardware or virtualized
 - Well suited for Amazon EC2/EBS
- Windows/Mac development via Virtual Machines
- MySQL 5.1 and MariaDB 5.2
- Commercial usage free for data < 50GB
- Also free for academic, research, evaluation

Use Cases

Use-case #1 - The Problem

- Want to perform ad-hoc analytics on clickstream data
 - Daily “canned” reports helpful
 - Data loaded daily into HDFS
 - Reports generated using Hadoop Map/Reduce jobs
- Needs
 - Ingest constant flow of data while maintaining indexes
 - Ad-hoc reporting capabilities

Use-case #1 - The Solution

- Why TokuDB?
 - High insertion rates, billions of rows
 - InnoDB was not sufficient
 - Query performance
 - Didn't remove indexes to maintain insert performance
 - SQL Interface
 - Business analysts already know SQL, don't want to learn Map/Reduce or NoSQL access methods
 - Schema flexibility
 - Adding indexes and columns while still accessing table data
 - Bonus: Compression
 - Clickstream data is highly compressible

Customer feedback - Intent Media:

"Column additions in the past simply were not practical, taking days to complete. They now take a matter of seconds, and can be accomplished in a non-disruptive fashion. This has dramatically improved our ability to adapt to the changing needs of our business, without fear that a schema change would lock up a table for a week or more, blocking other time-sensitive analyses."

Use-case #2 - The Problem

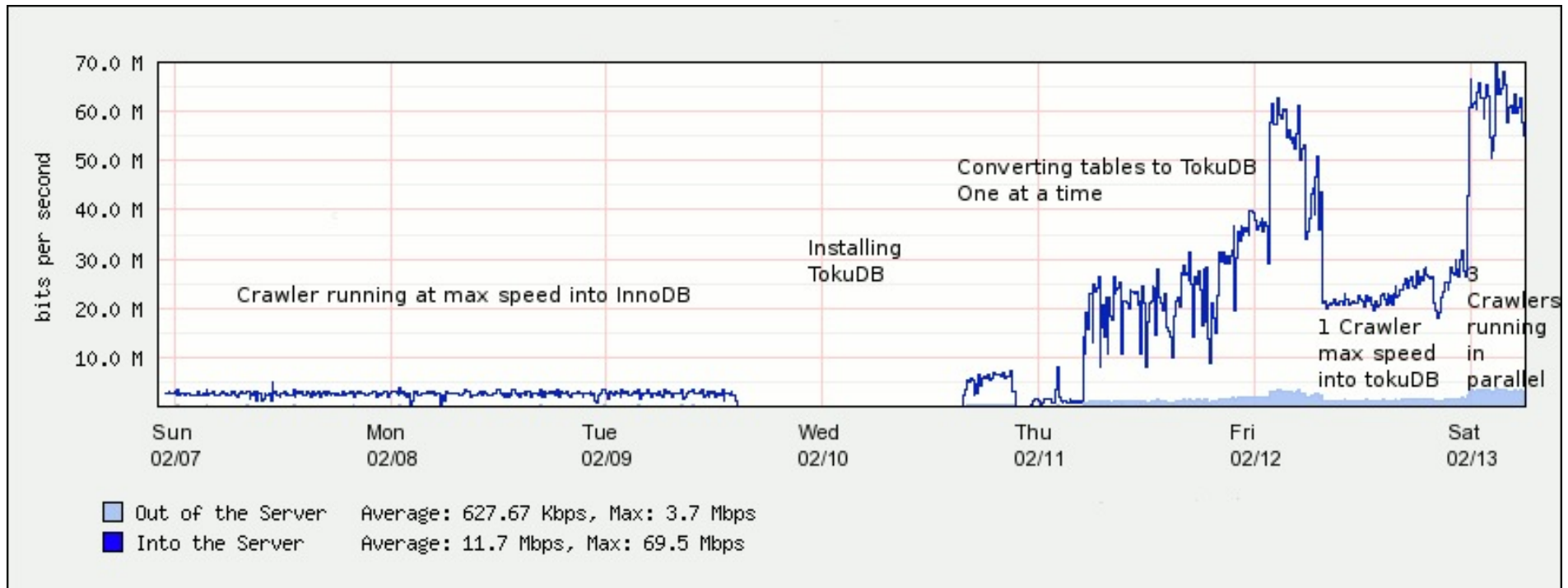
- Web crawler speed severely limited by existing database
 - Random insert performance of InnoDB insufficient
 - Estimated between 3 and 6 months to do a full crawl
 - Experienced 30 hour crash-recovery times
- Needs
 - High speed indexed inserts
 - Improved recovery time

Use-case #2 - The Solution

- Why TokuDB?
 - Measured 80x improved insertion rate
 - Shifted focus to improving crawler application performance
 - Full crawls now complete in under 2 weeks
 - Improved recovery time
 - Recovery now measured in minutes
 - Bonus: Compression
 - Crawl data highly compressible
 - Saving on expensive RAID storage
 - Bonus: Schema flexibility
 - Just added an index on the largest table, 24 hours to complete, table was fully available

Use-case #2 - The Picture

- Migration from InnoDB to TokuDB



Customer feedback - Profile Technology:

"I measured an 80x improvement on our crawler's insertion rate, boosting our overall performance by 20x, and we expect our 2 billion row database will now quickly grow to 8 billion or more. Insertion speed is no longer the bottleneck and we can now complete a full crawl in just 1-2 weeks."

Additional Resources

- **Contacting the Presenters**
 - **Lawrence@tokutek.com, @schwartzlaws** on Twitter
 - **Tim@tokutek.com, @tmcallaghan** on Twitter
- **Downloads and Support**
 - **tokutek.com/products/downloads/**
 - Will need to register to get a login
 - Tarballs offer either MySQL or MariaDB with TokuDB
 - Commercial use free for < 50GB. Also free for academic, research, eval
 - **support@tokutek.com, @tokutek** on Twitter
- **Technical Info**
 - **tokutek.com/technology/**
 - **Understanding Indexing**
 - » **<http://vimeo.com/26454091>**
 - **How TokuDB Fractal Tree Indexes Work**
 - » **<http://goo.gl/Smu6H>**