



InnoDB: Status, Architecture, and New Features

MySQL User Conference
April 15, 2008

Heikki Tuuri
CEO Innobase Oy
Vice President, Development
Oracle Corporation

Ken Jacobs
Vice President, Product Strategy
Server Technologies Division,
Manager, InnoDB group
Oracle Corporation

INNOBASE





Today's Topics

- InnoDB Overview

INNOBASE



Today's Topics

- InnoDB Overview
- ANNOUNCEMENT:

InnoDB Plugin 1.0 for MySQL 5.1



Today's Topics

- InnoDB Overview

- ANNOUNCEMENT:

InnoDB Plugin 1.0 for MySQL 5.1

- InnoDB Plugin Features & Performance



Today's Topics

- InnoDB Overview
- ANNOUNCEMENT:

InnoDB Plugin 1.0 for MySQL 5.1

- InnoDB Plugin Features & Performance
- Summary and Q&A

INNOBASE

InnoDB -

FAST. RELIABLE. PROVEN.



- FAST:
 - row-level locking, MVCC -> high concurrency & throughput
 - High performance CPU, memory and I/O architecture
 - Efficient indexing (covering indexes)
- RELIABLE:
 - Automatic crash recovery, page-level double-write and checksums
 - Integrated referential integrity and transactions
 - Online backup with InnoDB Hot Backup
 - Well written, well tested code
- PROVEN:
 - Distributed by MySQL since 2001
 - Wide use in large-scale customer deployments

INNOBASE



InnoDB in MySQL 5.1

- Fully usable with new MySQL features like partitioning, row-based replication
- Reduced “next-key” locking with READ COMMITTED isolation (requires RBR)
- Mostly bug fixes from MySQL 5.0
- **Significant** performance and scalability improvement with AUTO_INCREMENT

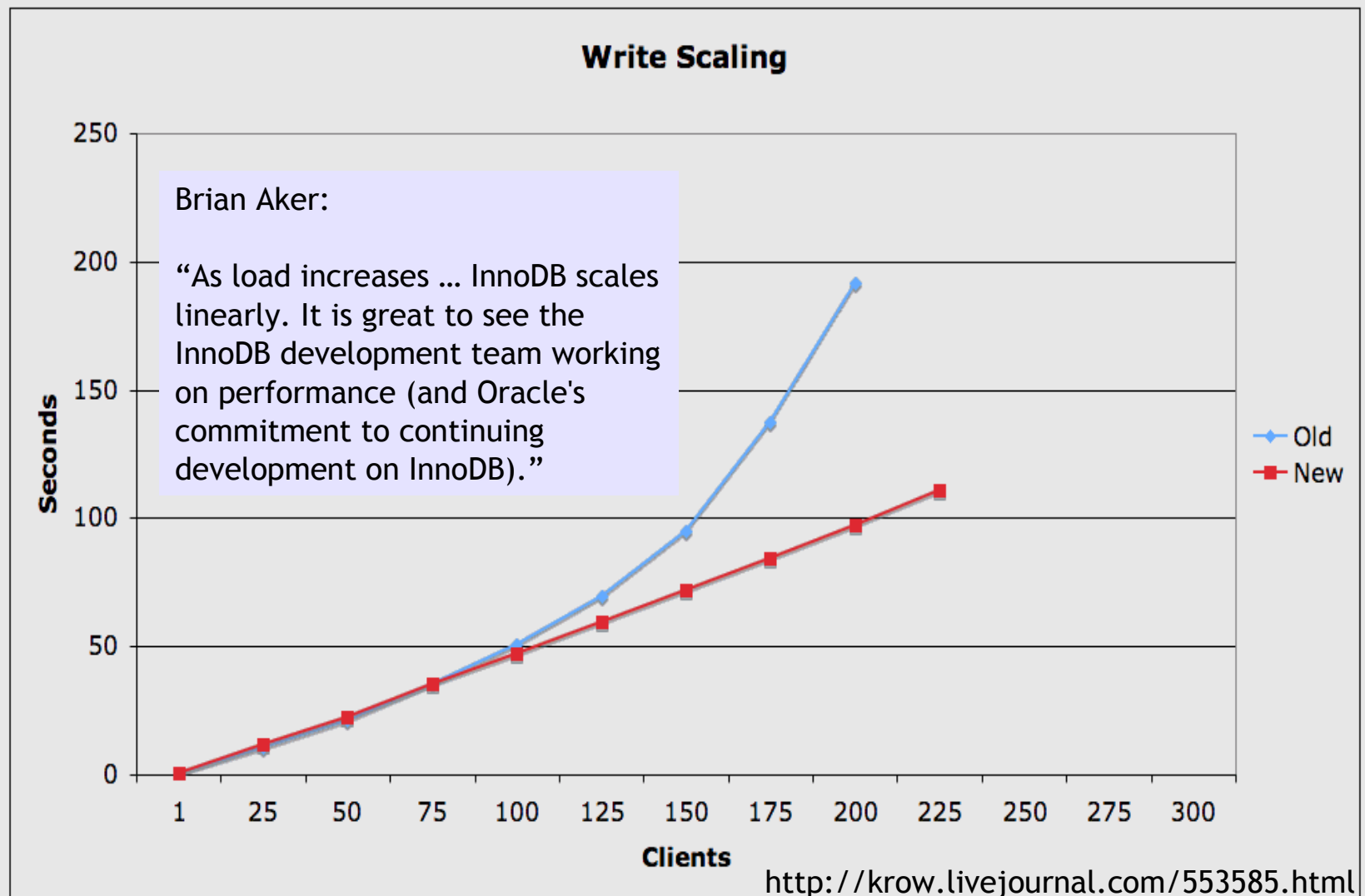
INNOBASE



InnoDB Auto-Increment in 5.1

- Before 5.1.22, InnoDB held a lock on the AUTOINC counter until the end of the SQL statement
- OK for short, single-row inserts; was required for reliable replay with replication
- New in 5.1: much faster, lighter-weight locking
- New parameter `innodb_autoinc_lock_mode`
 - 0 (“Traditional”) for backward compatibility
 - 1 (“Consecutive”) for multi-row INSERTs, precludes gaps in numbers generated by one SQL statement
 - 2 (“Interleaved”) even faster, requires row-based replication
- Documentation:
13.5.6.3. How AUTO_INCREMENT Handling Works in InnoDB

Auto-Increment Performance



INNOBASE



ANNOUNCEMENT

InnoDB Plugin 1.0 for MySQL 5.1

Early Adopter Release
Now Available on www.innodb.com

INNOBASE



InnoDB Plugin New Features

- **Fast index creation**
 - add or drop indexes without copying the data
- **Data compression**
 - shrink tables, to significantly reduce storage and I/O
- **New row format**
 - off-page storage of long BLOB, TEXT, and VARCHAR columns
- **File format management**
 - protects compatibility of databases and InnoDB versions
- **INFORMATION_SCHEMA tables**
 - information about compression and locking
- **Changes for flexibility, ease of use and reliability**
 - Dynamic control of innodb_file_per_table
 - TRUNCATE TABLE re-creates the *.ibd file to reclaim space
 - “Strict mode” to prevent mistakes



About the InnoDB Plugin

- Dynamically INSTALLED w/o relinking MySQL
 - Linux available now, Unix-like & Windows to follow
- Available in source and binary shared library
- Licensed under the GPL V2, just like MySQL
- Supports MySQL 5.1.x; 6.0 TBA
- Compatible with existing InnoDB databases
- Supports temporary use
 - can downgrade to standard InnoDB ... carefully!
- Supported via forums.innodb.com

INNOBASE



Now Presenting ...

Dr. Heikki Tuuri

Founder of Innobase Oy

Chief Architect of InnoDB

VP Software Development, Oracle

INNOBASE



InnoDB Plugin

New Features & Performance

- **Obtaining and Installing the InnoDB Plugin**
- **Description of New Features**
 - Fast index creation
 - Data compression
 - New row format
 - File format management
 - INFORMATION_SCHEMA tables
 - Other changes

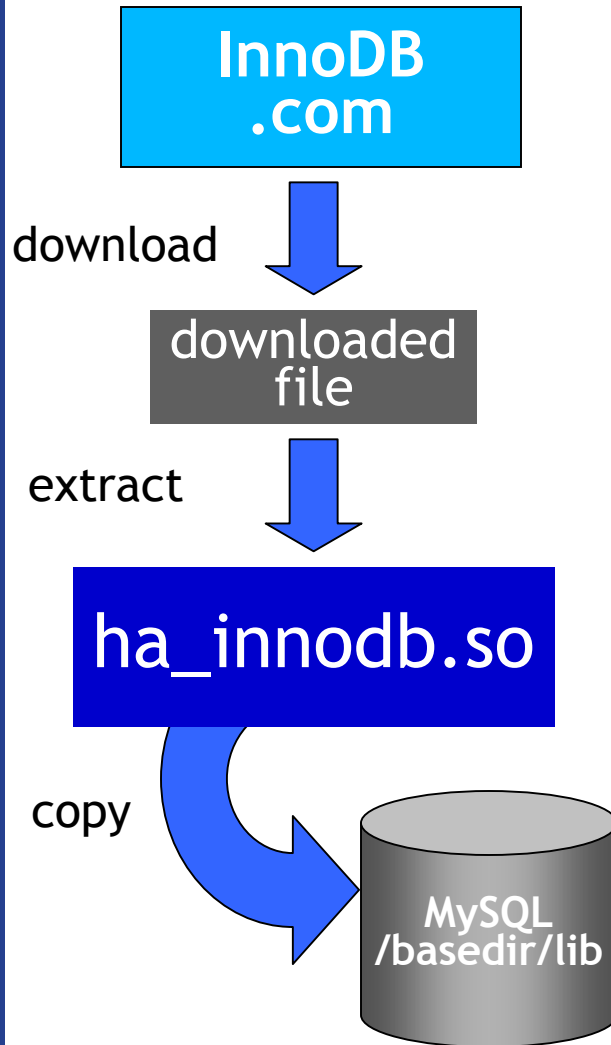
INNOBASE



Obtaining and Installing The InnoDB Plugin for MySQL

- Download, extract, copy
InnoDB Plugin to MySQL directory
- Shut down MySQL, add InnoDB Plugin
parameters to my.cnf, start MySQL
- **INSTALL PLUGIN**
- Start using the InnoDB Plugin!

Installing the InnoDB Plugin



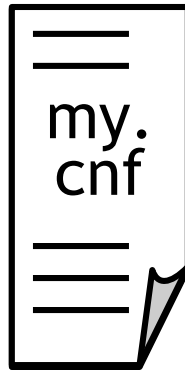
- Download precompiled binary (Linux x86 32-bit & Linux x86_64 available now)
- Extract with `gzip -d` and `tar -xvf`
- Copy the file `ha_innodb.so` to the directory where the `mysqld` server finds plugins (usually dir `lib` under the `basedir` of MySQL)

INNOBASE

...Installing the InnoDB Plugin



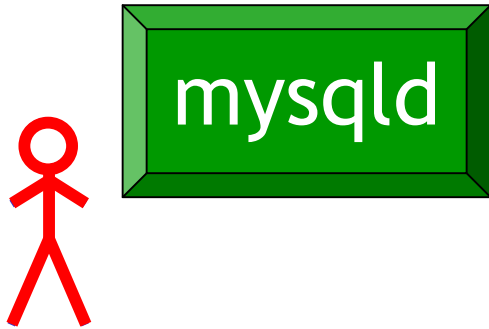
edit



- Shut down **mysql** server
- Add to my.cnf
 - **skip_innodb**
 - **innodb_file_per_table**
 - **innodb_file_format =Barracuda**
- Start **mysql**

INNOBASE

...Installing the InnoDB Plugin

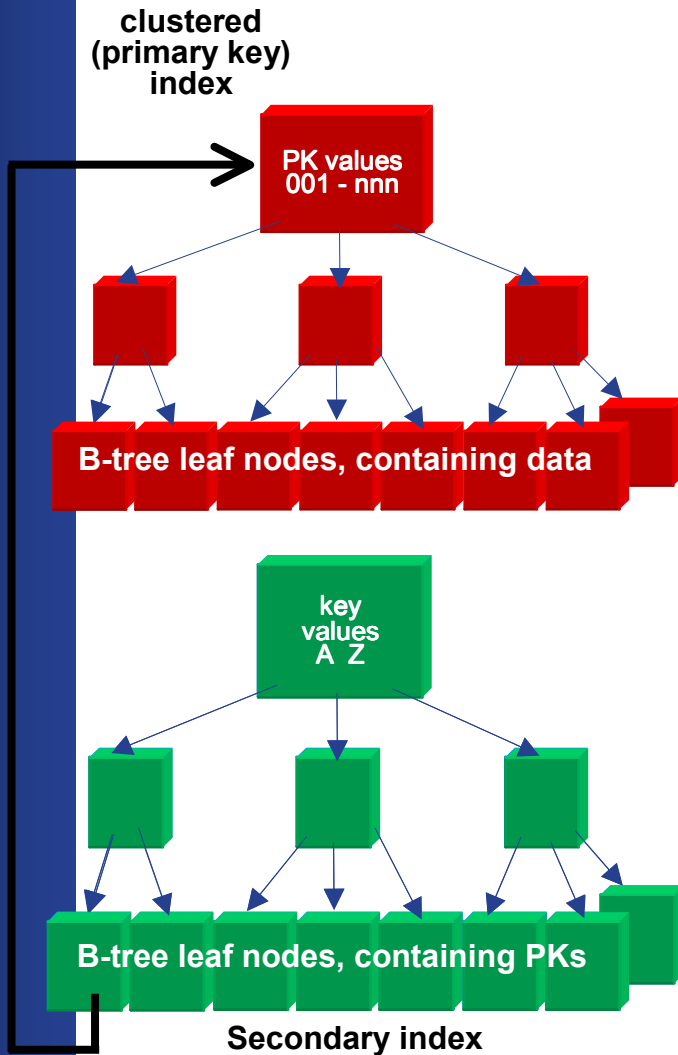


Install
Plugin

Install
Info Schema
tables

- Start mysql client as superuser:
`mysql -uroot -p...`
- **INSTALL PLUGIN INNODB SONAME 'ha_innodb.so'**
 - if this fails, check the mysql `.err` log
- To use InnoDB InfoSchema tables:
INSTALL PLUGIN INNODB_LOCKS SONAME 'ha_innodb.so', etc.
- Verify successful installation with
SHOW PLUGINS

InnoDB Fast Index Create

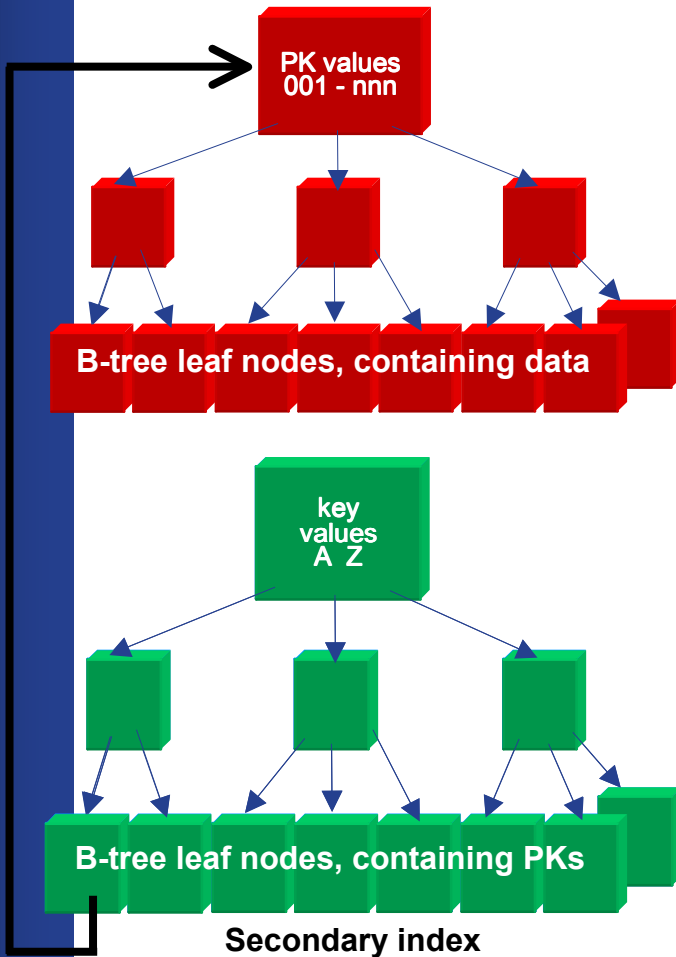


INNOBASE

- MySQL/InnoDB 5.1 rebuilds the entire table, row-by-row, to create a new secondary index
- The InnoDB Plugin builds just the new indexes, not the entire table
 - Sorts data on secondary key
 - Inserts the rows into the index
- MUCH faster, since the table is not re-created and because the data are inserted in order
- DROP INDEX for secondary index is even FASTER; data dictionary change only

Fast Index Create - Practical Considerations

clustered
(primary key)
index

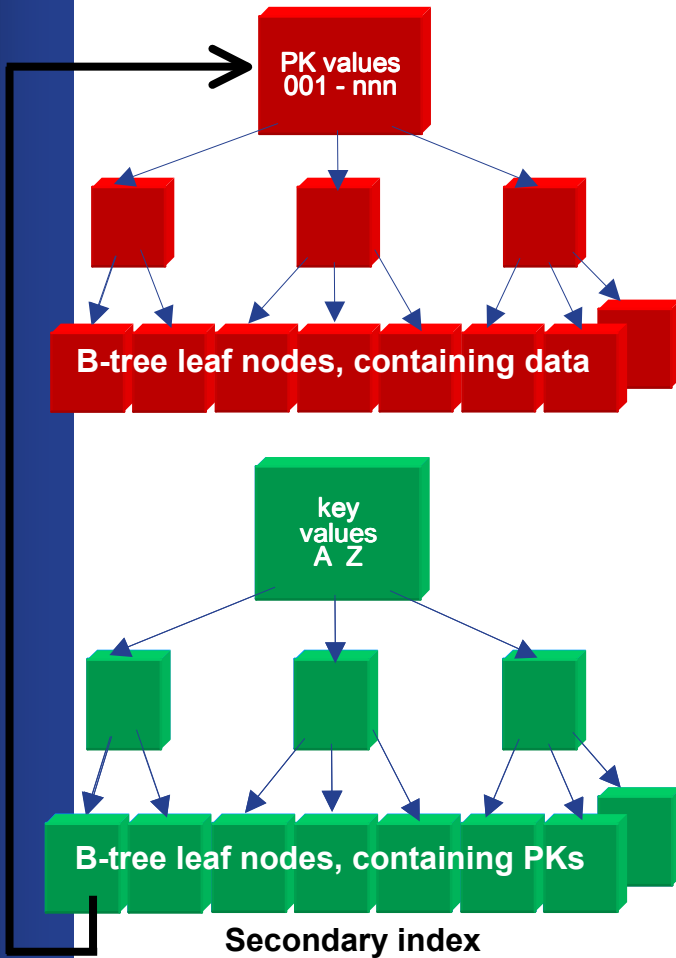


INNOBASE

- InnoDB must still recreate the whole table to add or change **PRIMARY KEY**
 - Slow, but faster than 5.1!
- Use one **ALTER TABLE** command to create several secondary indexes; faster than creating separately: fewer table scans
- Adding/dropping **FOREIGN KEY** constraints is NOT fast; still requires table rebuild 🙄

Fast Index Create - Concurrency Considerations

clustered
(primary key)
index

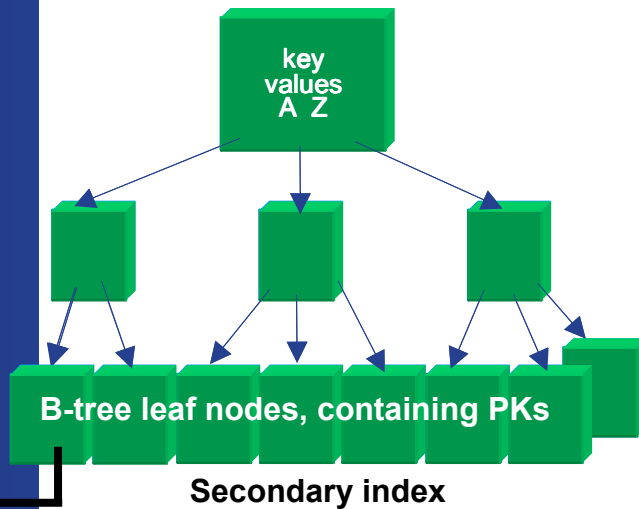
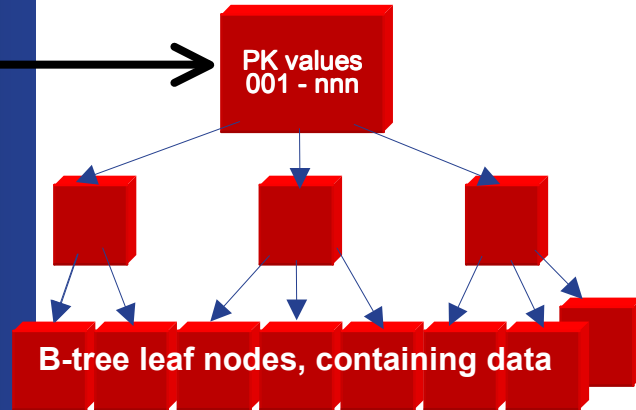


INNOBASE

- CREATE & DROP INDEX must wait for active update txns to commit
- CREATE of secondary index locks table in shared mode
 - **SELECT** queries can run concurrently
- CREATE & DROP of clustered (primary key) index or DROP of secondary index requires an exclusive lock
 - Not even consistent read **SELECT**s are allowed
- Newly created indexes lack historical version info for rows (for that index)
 - Consistent read **SELECT**s of older transactions may return inconsistent results

Fast Index Create - Performance

clustered
(primary key)
index



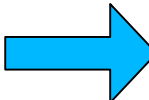
INNOBASE

**Creation
Time**

File size **Approx 3 GB**

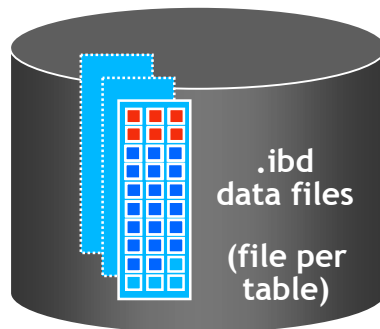
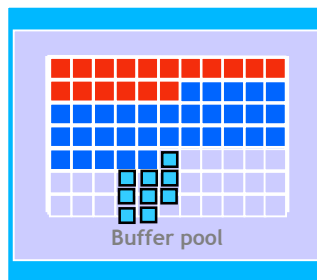
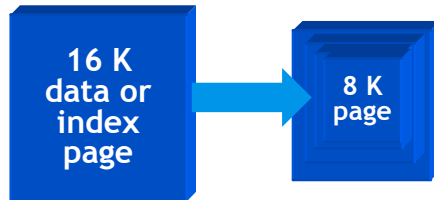
MySQL 5.1 **88 minutes**

InnoDB Plugin **8 minutes** 😊

- Fast index creation reduced size of secondary indexes by 30%
-  Faster index scans!
- Even faster speedup for larger tables!

Test performed by
Peter Zaitsev of Percona

InnoDB Table Compression



INNOBASE

- You choose compressed page size per table
`CREATE TABLE t ...
KEY_BLOCK_SIZE=8`
- InnoDB tries to compress data & index pages from normal 16 kB size to specified compressed page size
- Typical values for the compressed page size are 8 kB and 4 kB
- InnoDB keeps some uncompressed pages in the buffer pool with compressed copy
- Requires `innodb_file_per_table=1` in `my.cnf` and new “**Barracuda**” `innodb_file_format`

How InnoDB Compression Works



16 K
data or
index
page

The diagram illustrates the compression process. It starts with a large blue box labeled '16 K data or index page'. A large blue arrow points downwards from this box to a smaller blue box labeled '8 K page'. To the left of the diagram is a vertical dark blue bar with a series of small light blue squares, resembling a database page structure.



8 K
page

- A novel method using zlib
- Patterns in your data will determine compression ratio, can often be >50%
- Secret: InnoDB's ability to PREDICT if a 16 kB page will compress to fit in 8 kB
- Unique: compresses all data in table *and* indexes

INNOBASE

How InnoDB Compression Works

16 K
data or
index
page



8 K
page

- Tries to minimize uncompression & recompression of pages when they change
- InnoDB keeps a “modification log” in each page, recording changes
- Updates & inserts of small records are written to the log w/o page reconstruction; deletes don’t even require uncompression
- Log also tells InnoDB if the page will compress to fit page size
- When log space runs out, InnoDB uncompresses the page, applies the changes and recompresses the page

INNOBASE

How InnoDB Compression Works

16 K
data or
index
page

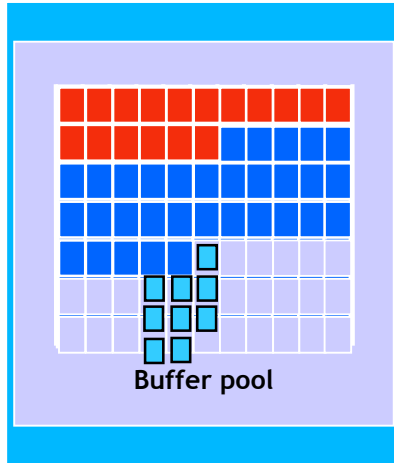


8 K
page

- If recompression fails, the B-tree node is split; the process repeats until update or insert succeeds
- A row must always fit on a single page (except BLOB, VARCHAR “overflow” pages)
- Splits can waste resources: the B-tree fill factor is lower, and CPU is used
- Watch ratio of successful to unsuccessful compressions in the Info Schema table `INNODB_CMP`

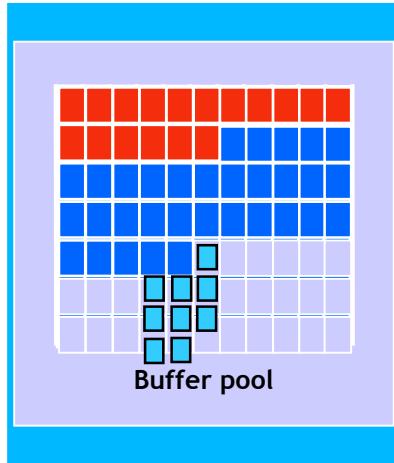
INNOBASE

Compression & the Buffer Pool



- InnoDB caches compressed (disk) pages in the buffer pool as usual
- If a page is frequently accessed, then InnoDB also keeps its uncompressed form in the buffer pool
- An adaptive LRU algorithm balances memory use based on the workload to save CPU time in uncompressing and compressing
 - With I/O-bound workload, InnoDB can allocate up to 90 % of the buffer pool to compressed pages
 - With CPU-bound workload, InnoDB may allocate most of the buffer pool for the uncompressed versions of compressed pages

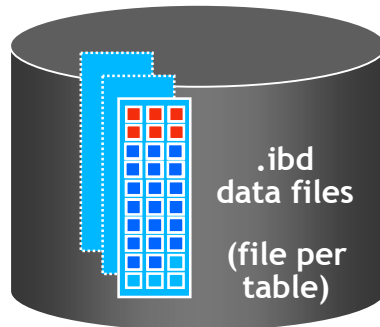
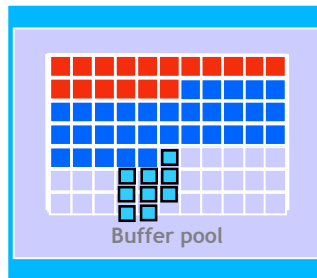
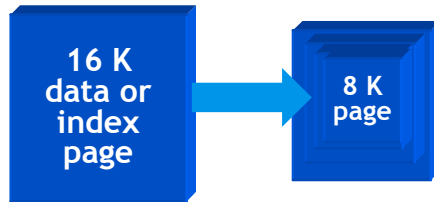
Compression & the Buffer Pool



- InnoDB caches compressed (disk) pages in the buffer pool as usual
- If a page is frequently accessed, then InnoDB also keeps its uncompressed form in the buffer pool
- An adaptive LRU algorithm balances memory use based on the workload to save CPU time in uncompressing and compressing

- A heuristic controls the ratio of compressed to uncompressed pages
- Future: we may tune that heuristic and also allow users to tune the ratio

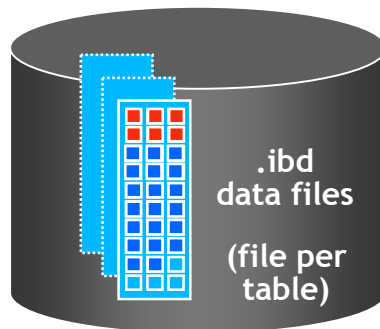
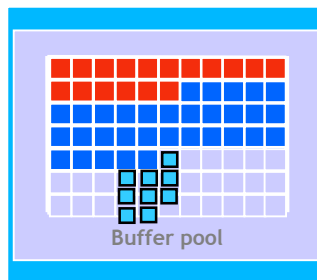
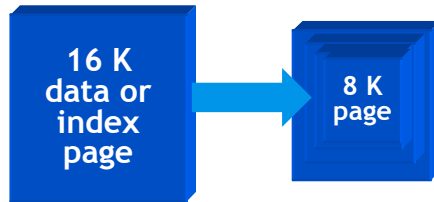
When to Use Compression



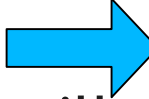
INNOBASE

- With I/O-bound workloads
- Compression means more data pages in the buffer pool
- → the buffer pool hit rate may be somewhat better
- → table scans may be faster because compressed data pages cause less I/O

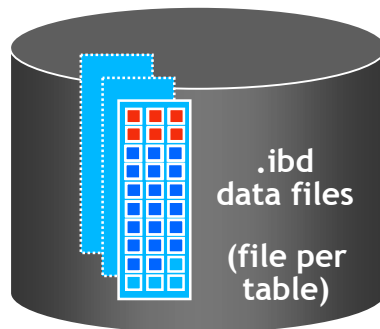
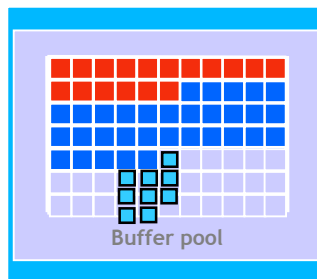
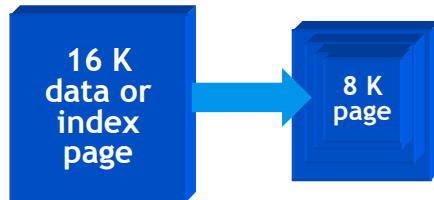
When NOT to Use Compression



INNOBASE

- With cpu-bound workloads
-  compressing & uncompressing will spend even more CPU!
- Do not compress very small, frequently accessed tables that easily fit in the buffer pool anyway
- If your data does not compress well; test by gzipping your .ibd data file
 - If the file does not compress to significantly less than 50 % of the original size, compression will probably not pay off

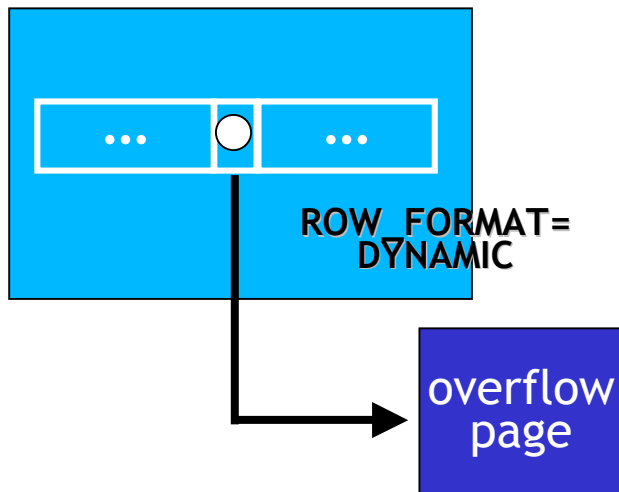
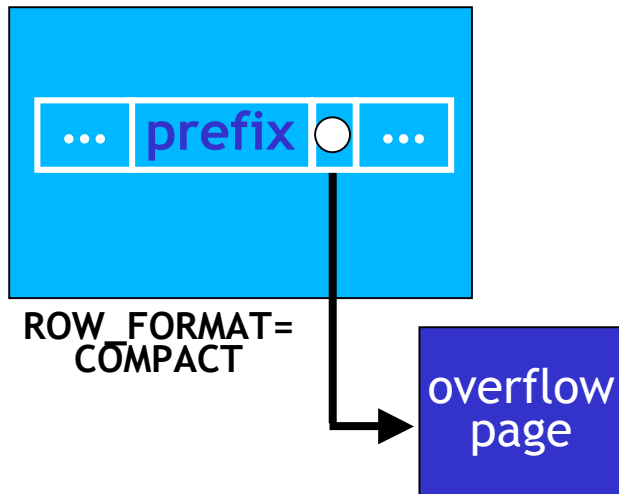
Performance of Compression



INNOBASE

- Internal tests: sysbench transactional workload may run up to 50% faster compressed vs. non-compressed (sysbench data compresses well)
- But, DBT2 runs 30 % slower with compressed tables (DBT2 data does not compress well)
- Peter Zaitsev's Percona has run initial benchmarks on the InnoDB plugin
 - Typical MySQL workloads: Forum, Social Network, Clickstream
- Data set size reduced by 50%, load time increased by 2x (not using fast index creation!)
- Most queries were 2x or more faster

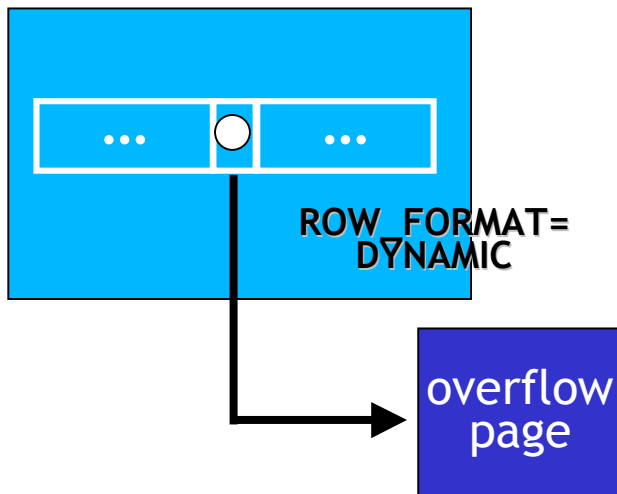
ROW_FORMAT=DYNAMIC



INNOBASE

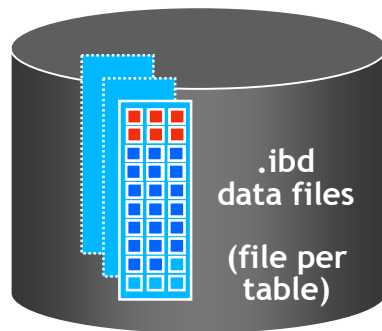
- If row does not fit in clustered index page, some long BLOB or VARCHAR column(s) may be stored on an overflow page
- **COMPACT** mode (5.1 default) always stores up to 768-byte prefix of such columns in the clustered index page
- New **DYNAMIC** mode stores long columns entirely “off-page”, with only a 20-byte prefix in the clustered index page (no prefix like COMPACT mode)

ROW_FORMAT=DYNAMIC



- Implied when KEY_BLOCK_SIZE is used on CREATE TABLE (compression)
- Requires [innodb_file_format=Barracuda](#)
- Even KEY_BLOCK_SIZE=16 can save space, compressing only data that does not fit on the B-tree page
- Reduces occurrence of 'record too long' errors, since no prefix is stored “on-page”

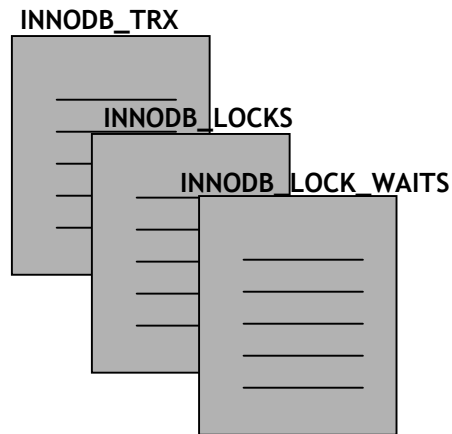
File Format Management



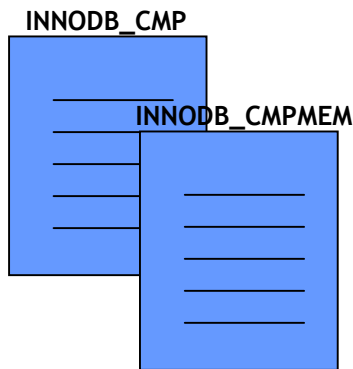
- Pre-plugin format: “Antelope”
- Enable a file format with new dynamic parameter `innodb_file_format`
- New “Barracuda” format enables compression, `ROW_FORMAT=DYNAMIC`
 - Fast index creation & other features do not require Barracuda file format
 - Default “Antelope” precludes new features
- Built-in InnoDB can access “Antelope” databases, but not “Barracuda” databases
 - Future: will check file format tag in system tablespace on startup
- Preserves ability to downgrade easily

INNOBASE

Information Schema Tables



Locking info



Compression info

INNOBASE

- Query InnoDB locks and other status
 - Previously available only in cryptic output of `SHOW INNODB STATUS`
- Info Schema tables must be installed:
`INSTALL PLUGIN INNODB_LOCKS SONAME 'ha_innodb.so', etc.`
- Info Schema tables reside in MySQL's "INFORMATION_SCHEMA" pseudo-database
- Example:
`mysql -uroot -p...`
`mysql> USE INFORMATION_SCHEMA;`
`mysql> SELECT * FROM INNODB_TRX;`

Info Schema Locking Tables

Query to see who is waiting, and for whose locks

```
mysql> SELECT
  trx_blocking.trx_mysql_thread_id AS blocking_thread,
  trx_blocking.trx_query AS blocking_query,
  trx_requesting.trx_mysql_thread_id AS requesting_thread,
  trx_requesting.trx_query AS requesting_query

FROM innodb_lock_waits
INNER JOIN innodb_trx AS trx_blocking
  ON innodb_lock_waits.blocking_trx_id = trx_blocking.trx_id
INNER JOIN innodb_trx AS trx_requesting
  ON innodb_lock_waits.requesting_trx_id = trx_requesting.trx_id;
```

Blocking thread	Blocking query	Requesting thread	Requesting query
5	SELECT SLEEP(100)	6	SELECT b FROM t FOR UPDATE
5	SELECT SLEEP(100)	7	SELECT c FROM t FOR UPDATE
6	SELECT b FROM t FOR UPDATE	7	SELECT c FROM t FOR UPDATE

User 6 is waiting for user 5.
User 7 is waiting for both user 5 and 6.



Other Improvements

- **TRUNCATE TABLE** reclaims .ibd file space
 - deletes the .ibd file of the table
 - creates a new empty one
 - returns space to the operating system
 - previously required DROP + CREATE TABLE
- InnoDB Strict Mode respects SQL syntax
 - Set with `innodb_strict_mode=on`
 - Warnings on CREATE or ALTER TABLE become errors; prevents silently ignoring specified options
 - Catch problems sooner!
 - Will become default; make sure scripts are correct!

INNOBASE



Try the new

InnoDB Plugin 1.0
for MySQL 5.1

Today!

INNOBASE



InnoDB Plugin Summary

- New features for performance, flexibility & reliability
 - Fast index creation
 - Data compression
 - New DYNAMIC row format
 - File format management
 - Info Schema tables
 - Dynamic innodb_file_per_table
 - TRUNCATE reclaims space
 - “Strict mode”
- Binary and source, under the GPL
- Early Adopter release available now
- Compatible w/ MySQL 5.1.x, existing databases
- Visit www.innodb.com and forums.innodb.com

INNOBASE

New InnoDB Forums

forums.innodb.com

INNODB®

 [Announcing the InnoDB Plugin 1.0 for MySQL 5.1](#)

Goto: [Search](#) · [My Control Center](#) · [Private Messages](#) · [Log Out \(KenJacobs\)](#)

Forums

Announcements

News and announcements about InnoDB, the InnoDB Plugin and InnoDB Hot Backup. You may also find here announcements and news about Innobase or the www.innodb.com website.

Options: [Mark Forum Read](#) ·  [RSS](#)

InnoDB Storage Engine

Get support for using the standard built-in InnoDB storage engine distributed by MySQL from other users and from InnoDB developers themselves.

Options: [Mark Forum Read](#) ·  [RSS](#)

InnoDB Plugin

Discussions about installation and use of the plugin version of InnoDB

Options: [Mark Forum Read](#) ·  [RSS](#)

InnoDB Hot Backup

Discuss and get help for the InnoDB Hot Backup utility

Options: [Mark Forum Read](#) ·  [RSS](#)

Suggestion Box

This is the place to make suggestions and request new features for any of the InnoDB products: the InnoDB storage engine (built-in or the plugin) and for InnoDB Hot Backup.

Options: [Mark Forum Read](#) ·  [RSS](#)

INNOBASE

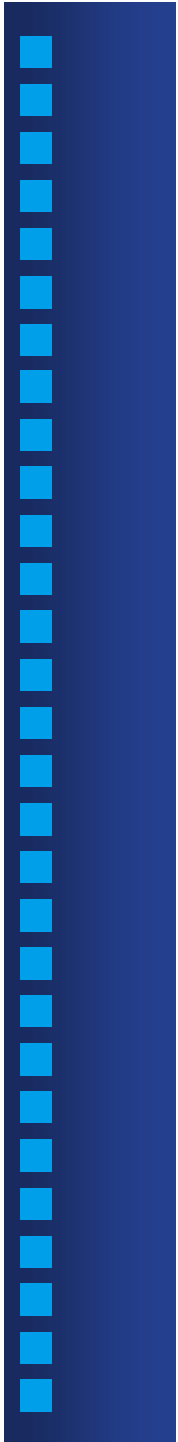
Got
Data?



Innobase is Looking for YOUR Help ...

- Got data ... and application workloads?
 - We'd love to use them to test InnoDB!
- Got a good story about using InnoDB?
 - We'd love to feature you on our website!
- Please see Heikki Tuuri or Ken Jacobs or email heikki.tuuri@oracle.com or ken.jacobs@oracle.com

INNOBASE



Q&A

QUESTIONS
ANSWERS

INNOBASE



INNOBASE

developer of

INNODB[®]

and the

INNODB[®] Plugin

and

INNODB[®] Hot Backup

ORACLE

Innbase is an Oracle company

