Practical MySQL

for web applications

Domas Mituzas
MySQL @ Sun Microsystems
Wikimedia Foundation

This is different world

- Not OLAP, Not OLTP
- OLWP. On line web processing
- OLP?

Web

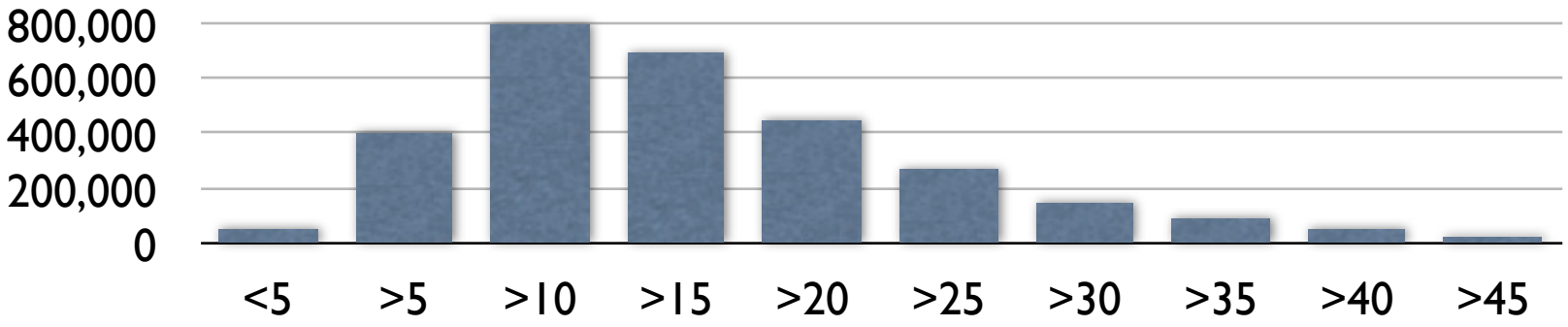
- Need data now. Right now. Instantly.
- High scale, huge churns
- Repetitive reads
- Click, click, click!

MySQL

- Great network bit-bucket
- Lightweight and fast
- Likes reads a lot
  - Data clustering
  - Covering indexes
  - Index-order reads

Data

- Mostly variable length
- Web users are lazy to type
- Tends to be short:
- Example of varchar(255) web supplied data length distribution:



Access

- Writes are small and rare
  - Inserts mostly
  - Text changes uncommon
- Reads are bigger and everywhere
  - Ordered data
  - Range scans
  - Pages

View patterns

- User
- Tag
- Time
- Related, friends, etc

Ordering

- Filesorts are bad.
  - TEXT/BLOB go to filesystem immediately
  - ORDER BY a LIMIT x will still scan whole range, 1000x costs
  - Filesort alone is more than 2x expensive operation, than just fetching rows
  - In Web, order is known and rarely changed

Indexing

- Provide index for every view pattern
- Composite indexes are a must
- varchar(255) cries for index
- Overindexing sometimes is worth it
  - As long as indexes get used

InnoDB

- Data is clustered by PK
  - Composite PKs group logical sets
- PK value is in secondary keys
  - Adding PK to index definition - cheap
  - Reading PK+SK values doesn't read PK (table) itself - very very cheap
- Every read is in index-order

Covering index

- A (partial) copy of table for different access pattern/view
- Especially useful if table doesn't fit in memory
- 100x less seeks for cold data
- Probably no need with SSDs :-)

JOIN orders

- 5.1 fixes this
- In 5.0, order on joined tables caused filesorts, even if indexes exist
- Might be faster fetching single row as separate SELECT, instead of joining to it
- May use FORCE INDEX and rely on implicit order, instead of ORDER BY. <-- DIRTY!
- Can use app-defined ordering columns in keys

Simple, not elegant

- Keep data where needed, not where elegant
- WHERE and ORDER BY fields - in same table
- JOIN for single static value not worth it
  - (usernames are short too!)
- Used to be heresy in other worlds

Query cache

- Does not work for big apps (transactions, lots of changes)
- Does great job for small ones
  - Needs uniform queries
  - Less dynamic queries - better hitrate
- Primary tweak on shared hosting

Counts

- MyISAM count(*) works, sometimes
- Scanning rows - inefficient
- Maintaining counts is easy and cheap
  - Aggregate data is easier to keep hot
  - INSERT ... ON DUPLICATE KEY ...

BLOBs

- Keep in separate tables
- Keep outside search query logic, display only
  - Cheaper metadata updates, especially InnoDB
  - Less data to read in big scans, orders, grouping, etc
  - Can always revisit blobs in second query
  - Easier to move out afterwards

Paging

- Bad: OFFSET .. LIMIT ..
- Good: WHERE id>xxx LIMIT ..
  - Who cares about 15th page?

Character sets

- Lucky if latin1 (or other 8bit charset) is enough
- Even luckier if binary
- Have to be very careful with utf8 and friends
  - More expensive filesorts!

Connections

- Persistent - bad
  - Connect is cheap, connection is not
  - Applications may spend less than 20% time in db
  - Prepare application, do all db work, disconnect, continue with other work
  - ___QQQ___ instead of _Q_Q_Q_Q_

Locking

- SELECT ... FOR UPDATE - necessary evil
- Good: Hot potato transactions
- Bad: waiting for anything
  - Especially streaming of files, etc
  - Use output buffers!
    - ob_flush() after COMMIT, not before
- Delay and split big changes

Replication

- Number one scaling technique
- Lag is not lag, if users don't notice it:
  - Check master status on change
  - Wait for it on slaves

Understanding

- Poor man's backtrace:
  - SELECT /* __METHOD__ */ ...
- Automatic, or manual, tagging of queries helps
- Profile!
  - Thats why query wrappers help

Main rule

- Your needs, your design
- Don’t listen to others

Questions? Lunch?

- Ask now!
- or domas @ #mysql freenode
- or domas at mysql dot com

Bon appétit

- I hope lunch is warm today
- Thanks for coming by