Storing Images in MySQL

Boston MySQL Meetup Group
Monday June 12th, 2006

Sheeri Kritzer
awfief@gmail.com
http://www.sheeri.com

A BLOB By Any Other Name

- Text is data
- BLOB is data
- Images are BLOBs

Similarities

- Syncing images across multiple servers
- Web server image caching
- Performance
- Adding another server

Similarities

- A filesystem is a DB
- Isolation – locking
- Durability – transactions committed not lost

Why Not?

- Filesystems store files
- Hard to backup/restore/repair
- Partial restore tedious

Why Not?
(continued)

- Binary logging doubles space
- More db connections, queries
- Bottleneck

Why Not?
(continued)

- Database fragmentation
- Portability
- Single storage engine for all data ???

Why?

- Snapshots easier, less expensive
- Scalability
- Indexing and sorting
- Access Control

Why?
(continued)

- Atomicity
- Consistency
- Database features
- Remote storage easier

Why?
(continued)

- Replication easy
- Single storage engine for all data
- Dual-level caching

Why?
(continued)

Oracle recommends it

http://www.oracle.com/technology/products/intermedia/htdocs/why_images_in_database.html

Caveats

- SELECT *
- Full table scan
- max_allowed_packet

Compromise

- Do both
- Originals in db
- Written to disk
- Transformations also written to disk

What We Chose

- 700,000 users, 6 pictures each
- 20,000 online at peak time
- Online user image retrieval

What We Chose

- Add 17-19,000 users a week
- Possible 102-114,000 pictures a week
- Need referential integrity

BLOB Sizes

- TINYBLOB 2^8 bytes 256 bytes
- BLOB 2^16 bytes 64 Kb
- MEDIUMBLOB 16 Mb
- LONGBLOB 4 Gb
- variable-length

What Storage Engine?

- 1 datafile per table?
- Table-level or row-level locking?
- Compressed, read-only storage?
- Fulltext indexing?

What Storage Engine?
(continued)

- InnoDB for row-level locking
- Can specify 1 datafile per table if desired
- No captions or text

Architecture

- 2 tables – 1 fixed-length
- Single write master to many read slaves

Creating the Table

```
CREATE TABLE ImageMd (
  imageNum int unsigned not null primary key,
  ownerUid int unsigned not null,
  modified timestamp,
  uploaded datetime) ENGINE=InnoDB
```

Creating the Table

```
CREATE TABLE Images (
  imageNum int unsigned not null primary key
   auto_increment,
  modified timestamp,
image mediumblob) ENGINE=InnoDB
```

Populating the Tables

LOAD_FILE('path/to/file.jpg')
(max_allowed_packet)

Application code

Retrieving the Images

Browser needs to know the data is an image

Questions?

Who's done this?

Other stories....