Introduction to the BLOB Streaming Project

MySQL Conference & Expo 2008

Paul McCullagh
PrimeBase Technologies GmbH
www.blobstreaming.org
Contents

• Basics: BLOBs & Streaming
• Overview of the BLOB Streaming Project
• The BLOB Streaming engine & Demos
• Database operations: INSERT/SELECT
• Advanced topics: Backup & Replication
Definition: BLOB

- Invented by Jim Starkey when working for DEC on Rdb, JRD (later InterBase).
- “A blob is the thing that ate Cincinnatti, Cleveland, or whatever” - Jim Starkey
- **Basic Large Object** is a “backronym” - Terry McKiever, Apollo Computer Marketing Dept.
- **Binary Large Object** - Informix
- Examples: photos, films, MP3 files, PDFs, text files, programs, documents and multimedia.
How MySQL handles BLOBs
Streaming a BLOB...
Goals of BLOB Streaming Project

- Stream BLOB data directly in and out of the database.
- Store BLOBs of any size (> 4GB) in the database.
- Create a scalable back-end that can handle any throughput and storage requirements.
- Provide an open system that can be used by all engines.
- Provide extensions for BLOB streaming to existing MySQL clients (C/C++/PHP/Java, etc).
Why put BLOBs in the Database?

- BLOB operations are transactional
  - No invalid references
- All data in one place - good for testing
- Small BLOBs are handled better by databases
  - Convenient to handle all data identically
- Backups are consistent.
- BLOBs in the DB can be Replicated - become part of the HA solution
Why "not to BLOB":

• A BLOB column makes a table slow
  ▪ Big rows in memory
  ▪ Sequential scans are not possible

• The database becomes too big
  ▪ Cannot be easily copied
  ▪ Backups become too slow
  ▪ Space not freed on delete
  ▪ The database does not scale well

• Replication is too slow
  ▪ BLOB data must be written to the binary log
The Solution to these Problems

- The “BLOB repository”:
  - A collection of BLOBs stored outside of the database rows
  - References are stored in the table

- Advantages:
  - Allows for incremental backup
  - Automatic defragmentation and compaction
  - Repository BLOB data not written to the binary log
  - The repository can be scaled-out
BLOB Streaming Architecture

User Application

MySQL Client API

MySQL Client/Server Protocol

MySQL Server Front-end

MySQL Client/Server Protocol

Stream-based Protocol (HTTP)

Other Storage Engine

Streaming Enabled Engine (PBXT)

BLOB Streaming Engine (MyBS)

Server-side Streaming API

HTTP-based client API Extension

API Ext.
The BLOB Streaming Engine:

- Does not provide conventional table storage:
  - CREATE TABLE (...) ENGINE=MyBS only possible for MyBS system tables.
- Built-in HTTP (Web) server
  - Publishes its own TCP port (8080 by default).
- BLOB Repository
  - BLOBs can be referenced by any table.
- Provides a server-side API for any engine to reference/dereference a BLOB.
A View of the Repository

One entry for each BLOB in the repository

mybs_reference
- Table_name
- Column_name
- Row_condition
- Blob_id
- Blob_url
- Repository_id
- Repo_blob_offset
- Deletion_time
- Remove_in

One entry for each Reference to a BLOB

mybs_repository
- Repository_id
- Repo_blob_offset
- Blob_size
- Access_code
- Creation_time
- Last_ref_time
- Last_access_time
- Content_type
- Blob_data

BLOB Repository

MyBS
The Temp BLOB Timeout

- BLOBs that are not referenced are deleted from the repository
- Deleted BLOB references are not removed immediately.
- `mybs_temp_blob_timeout` determines delete timeout (in seconds).
MyBS Repository Demo

- Create a table
- Upload a BLOB for the table to the repository
- View the result in a browser
- Upload a text BLOB for the table to the repository
- Examine the repository using the system tables
INSERTING a BLOB...

1. HTTP PUT
2. 200 OK
3. Retrieve file
4. SQL INSERT
5. Retain blob

MyBS

PBXT

MySQL Front-end

BLOB Repository

notes_tab

null ~.0 ...

http://.../test/my_test_tab

~*test2/*3–128–47f8d688–0
SELECTING a BLOB...

1. SQL SELECT

2. Notes:
   - 5 ~.0 ...

3. HTTP GET
   - http://.../~test/~1-128-3dd49f08-0

4. 200 OK
INSERT & SELECT Demo

- Upload a BLOB
- INSERT the BLOB URL (reference)
- View the BLOB Repository
- SELECT the BLOB URL from the table.
- Use the URL to retrieve the BLOB.
Client-side Extensions

- JDBC - Connector/J SE 5.0.7:
  - Supports transparent streaming of BLOB data to and from the MyBS BLOB Repository: using `get/setBinaryStream()` and `get/setBlob()`

- Google Summer of Code 2008
  - Extension to Connector/PHP

- Still to be done:
  - mysqlclient library - simple extension to do HTTP GET and PUT.
  - ODBC - Support using existing functions: `SQLGetData()` and `SQLPutData()`
  - Other Languages (Perl, Ruby, etc)
How will BACKUP work?

• How does mysqldump work if SELECT returns a reference, not the data?

```
mysql> select * from my_test_tab;
+---------------------------------+
| n_id | n_text                       |
+---------------------------------+
| 1    | ~*test/~2-128-1680fce2-0     |
+---------------------------------+
1 row in set (0.00 sec)
```

• mysqldump will also dump `mybs_repository` and `mybs_reference`.

• For import of dump, MyBS system tables must be writeable.
Replication Mechanism

Master Server

(4) INSERT

(3)

(2)

~*test/~1–12...08–1

(1) PUT

notes_tab

5 ~.1 ...

5 ~.1 ...

Slave Server

(6) Replicate

(5) retain()

(8) GET

(7) retain()

notes_tab

5 ~.1 ...

5 ~.1 ...

(9)
Distributed Repository

Repository Servers

Host 1

Host 2

Host 3

Database Host

retain()

PUT

INSERT
Q&A

Thanks for Listening!

http://www.blobstreaming.org

http://sourceforge.net/projects/mybs

http://pbxt.blogspot.com