



MySQL Maatkit

By Pradeep Chandru



About Us

- Founded by a group of enthusiastic and dedicated individuals
- Hands on experience in MySQL versions 3.x, 4.0, 4.1, 5.0, & 5.1
- Experience in working with large set ups like Sify, Yahoo, Genpact, Cricinfo, BSNL, & TCS
- Have 7 Certified MySQL DBAs and a cluster certified MySQL professional with an average of 6 years of experience and 4 DBAs with an average of 3 years of experience
- Managed MySQL database driving Platforms with database volume ranging from 10GB to 6 TB



Our Services

- Design and architecture services
- Query tuning and performance optimization
- Migration from any database to MySQL
- Monitoring and remote maintenance of MySQL databases
- Design, Implementation and Maintenance of MySQL high availability solutions



Maatkit



Why do I need Maatkit?

- Ensuring slaves have the same data as masters
- Re-syncing data efficiently on slaves
- Creating a delayed slave
- Finding objects and executing commands on them
- Data ETL, archiving and purging
- Dumping and restoring data in many threads at once
- Piping EXPLAIN through a filter to make it more readable



MySQL Archiver

- `wget http://www.maatkit.org/get/mk-archiver`
- Copies rows from one table to another
- Minimal locking, minimal blocking, efficient queries
- Does not do `INSERT... SELECT`, to avoid locking
- Archive and/or purge
 - Delete instead of copy
 - Use option `-purge`
 - Write to file and then load using `LOAD DATA INFILE`



... archive continued

- Logic of the archiver
 - Keep transactions small
 - Split based on primary key, which reduces the lock and maintains the speed.
 - Keep contents minimal in transaction log, so flush to disk happens in smaller blocks.
 - Even if deadlock occurs, keep the rollback to be minimal
 - Commit in blocks, not tiny transactions.



... archive continued

- Scenario
 - `select min(id) from table where name = 'dba'`
 - `select min(id) from table where name = 'dba' and id > 0`
 - `select id from table where name = 'dba' limit 1;`



To do

- Problem with limit, when indefinite scan to match the number of rows that match the condition



Example

- Before
- **\$ mysql -s -e 'select count(*) from test.source'- 8192**
- **\$ mysql -s -e 'select count(*) from test.dest' – 0**
- After Executing the command
- **\$ mk-archiver --source h=localhost, D=test,t=source --dest t=dest --where 1=1**
- **\$ mysql -s -e 'select count(*) from test.dest' - 8191**
- **\$ mysql -s -e 'select count(*) from test.source' – 1**
- **Why there was 1 row in source?**



Example2

- **Previous example took one by one**
- **To increase the speed**
 - `$ mk-archiver --source h=localhost,D=test,t=source --dest t=dest --where 1=1 --commit-each --limit 100`



Options

- `--analyze`
 - Run analyze after completion on completion in source or the destination
 - Example `--analyze=ds`
- `--buffer`
 - Write the output to file and then flush at commit the changes.
- `bulk-delete`
 - Can be faster and also slower!!!



Options continued

- `--bulk-insert`
 - creating a temporary file for each chunk of rows, and writing the rows to this file instead of inserting them
- `--columns`
 - Take only specific columns to copy to another table.
- `--commit-each`
 - Commit after each archive or insert
 - Can avoid search that can cause nil result and avoid scans



Options continued

- --dry-run
 - Test run- don't do any thing
- --for-update
 - Adds for update modifier to select statements.
- --ignore
 - User insert ignore for insert
- --limit
 - Limit the rows in select
- --optimize
 - Optimize after delete and insert



Options continued

- --plugin
 - Can write custom plugin to create new features or new checks
- --quick
 - Uses delete quick for delete. This is good when used with optimize.
- --replace
 - User replace for insert
- --retries
 - No of times to retry after deadlock



Options continued

- `--run-time`
 - Time to run
- `--set-vars`
 - Set command line arguments for mysql like `wait_timeout`.
- `--share-lock`
 - Ads lock in share mode
- `--statistics`
 - Print stats on the archive process
- `--txn-size`
 - No of rows to be part of transaction



THANK YOU