

# *Slave Setup and Syncing Issues*

Presented by:  
Sheeri K. Cabral

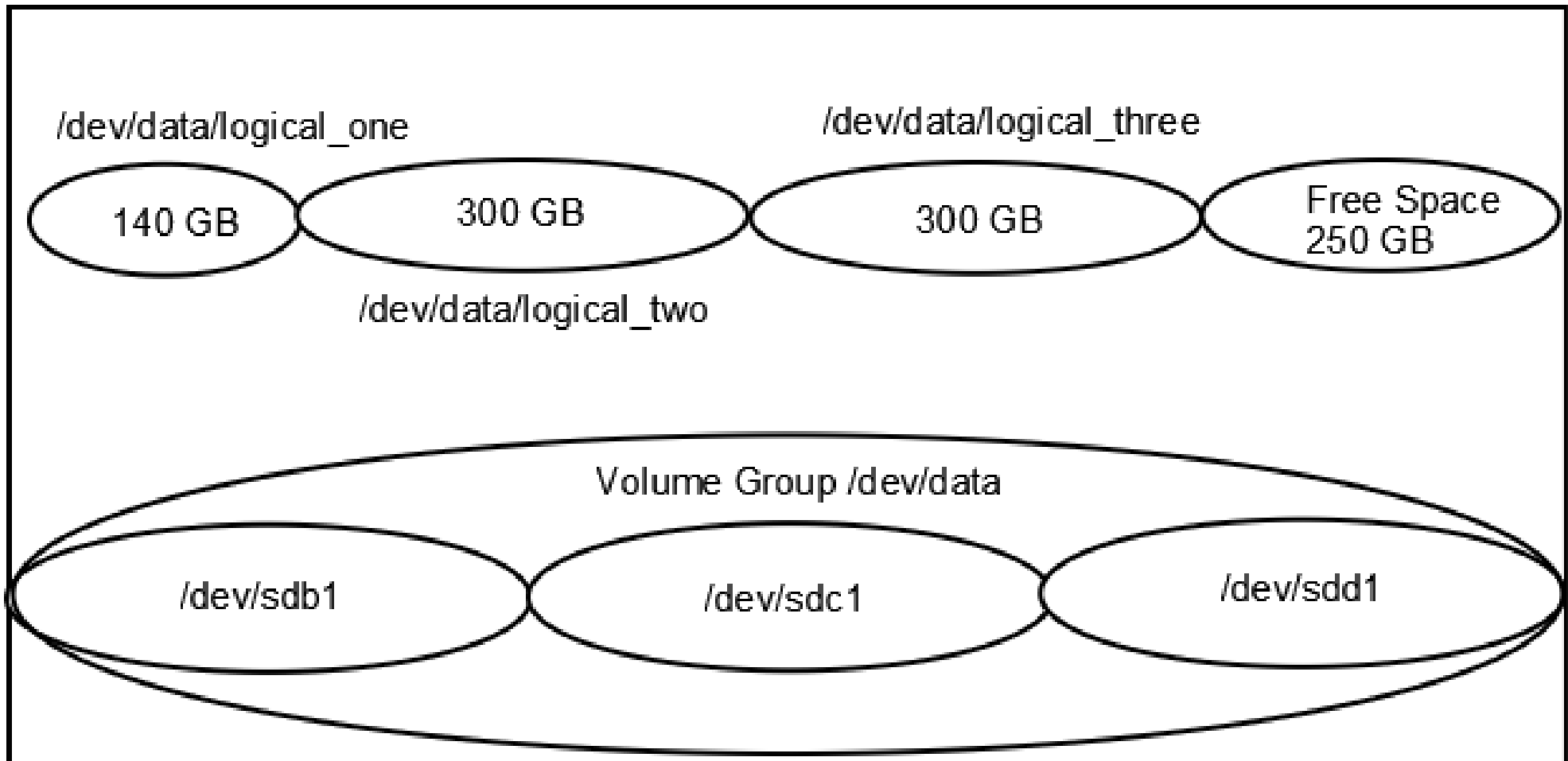
Pythian  
love your data

# Topics Covered

- Slave setup
- Getting the dataset
  - LVM snaps hots
  - ZFS snapshots
  - Locking tables and running mysqldump
  - Innodb backup
  - mysqlhotcopy
- Other information needed for setup
- What causes a slave to get out of sync
- How to sync a slave with its master

# LVM Background

LVM (Logical Volume Manager) can be used to manage storage on Linux servers.



# LVM Snapshot

- Uses the LVM commands:
  - pvcreate
  - pvdisplay
  - vgcreate
  - vgdisplay
  - vgextend
  - lvcreate
  - lvdisplay
  - lvextend
- File system considerations

# LVM Snapshot How-to

- Requires a small amount of database operation interruption. It's an “Almost Hot” backup.
- On Master:
  1. `mysql> FLUSH TABLES WITH READ LOCK;`
    - `mysql> SHOW MASTER STATUS;`
    - 2. `# lvcreate -L 5G -s -n snap /dev/vgname/data`
    - 3. `mysql> UNLOCK TABLES;`

and then...

# LVM Snapshot How-to

- 4. # mount /dev/vgname/snap /mnt/back\_snap
  - # scp -R /mnt/snap slave\_ip:/mysql\_data\_dir
    - this step takes a while
  - # umount /dev/vgname/snap
  - #! lvremove /dev/vgname/snap
- These MUST be done in the specified order.

# ZFS Snapshot

- On Master:

```
-mysql> FLUSH TABLES WITH READ LOCK;
```

```
-# zfs snapshot pool_name/part_name@date
```

```
-mysql> UNLOCK TABLES;
```

```
-# scp -R /pool_name/part_name/.zfs/snapshot/date  
slave_ip:/mysql_data_dir
```

```
-# zfs destroy pool_name/part_name@date
```

- These **MUST** be done in the specified order.

# For any snapshot

- Make sure you get binary log positions for the machine itself
  - SHOW MASTER STATUS
  - binlogs and index file
- Make sure you get any binary log positions for the machine's master, if there is one
  - SHOW SLAVE STATUS
  - master.info



# MySQL Dump

- Entire database
- `# mysqldump -user=root -pPASSWORD db > db.sql`
- Single table
- `# mysqldump -user=root -p db table_name > db.table.sql`

# MySQL Dump

- Important Options

- --databases

- --all-databases

- --single-transaction

- --opt

- (same as --add-drop-table --add-locks --create-options --disable-keys --extended-insert --lock-tables --quick --set-charset)

- --lock-tables

# InnoDB Hot Backup

- Backup
- ```
perl innobackup --user='username'  
  --password='password' --compress /etc/my.cnf  
  /backup_dir
```

# InnoDB Hot Backup

- Restore:
- [08:21:53] root@db02
- /mysql/data> ./ibbackup --apply-log --uncompress /etc/my.cnf
- 
- Contents of /etc/my.cnf:
- innodb\_data\_home\_dir /mysql/data
- innodb\_data\_file\_path ibdata1:2000M;ibdata2:10M:autoextend
- datadir /mysql/data
- innodb\_log\_group\_home\_dir /mysql/data
- innodb\_log\_files\_in\_group got 2
- innodb\_log\_file\_size 1073741824

# InnoDB Hot Backup

- Documentation:  
<http://www.innodb.com/support/documentation/innodb-hot-backup-manual/>
- InnoDB Hot Backup has a 30-day free trial
- Xtrabackup works similarly, and is completely free

# mysqlhotcopy

- If there are only MyISAM tables this tool can be used.
- It locks tables for read and copies the .FRM, .MYD, MIY files to another location

```
shell> mysqlhotcopy dbname /backup_dir
```

# Cold Backups

- First you shut down the MySQL daemon...
- ...then you copy or archive files

# Do Not Forget...

- my.cnf
- triggers and views and stored procs
- crontab scripts
- users permissions
- binary logs



# Other Recommendations

- Choose low load time if you are going to lock tables.
- Especially if you are using mysqlhotcopy.
- The lvm and zfs option provide low overhead and do not impact performance greatly.

# What causes slave to get out of sync

- The main reason is statement-based replication.
- Some SQL statements are non-deterministic – thus produce different results on different machines.
- Network errors (this is hard to prove)
- Human errors. Someone updates data on slave by mistake.

# What causes slave to get out of sync

- Missing temporary tables. If slave is restarted it loses all temporary and in-memory tables
- Dependencies on none-replicated data
- Non-unique `server_id` on two or more slaves

# How to sync slave with master

- First of all determine, what tables are out of sync.
- `mk-table-checksum` is a good tool for this.

- Usage:

```
perl mk-table-checksum --replicate=maatkit.checksum  
  --algorithm=BIT_XOR localhost --user=root -askpass
```

- You can also specify chunks

# mk-table-checksum and chunking

- Chunks use numeric indexes
  - date/time counts as a number

- Usage:

```
perl mk-table-checksum --replicate maatkit.checksum  
  --algorithm BIT_XOR --modulo 21 --chunk-size 1000  
  --offset 7 localhost
```

- If there is no numeric index, it will checksum the whole table

# mk-table-checksum

- Need to use `--create-replicate-table` on the first run
- If there is no numeric index, it will checksum the whole table

# mk-table-checksum

- Need to use `--create-replicate-table` on the first run
- If there is no numeric index, it will checksum the whole table
- Creates a table and stores information
- To find tables/chunks not in sync, query the table

# mk-table-checksum

```
mysql> select * from checksum1 where this_crc!=master_crc\G
***** 1. row *****
      db: production
      tbl: foo
      chunk: 1176
boundaries: `id` >= 11841145 AND `id` < 11851214
      this_crc: 9139bf59
      this_cnt: 295
master_crc: f91afa73
master_cnt: 245
      ts: 2009-11-08 16:23:40
```



# When mk-table-checksum fails

- A suitable key can't be found on a table
- As any piece of software it is not free from bugs
- In this case you can use `mysqldump --order-by-primary --lock-tables`
- Do this on both servers and use `diff`

# Ok, slave is out of sync. What's next?

- 1. Try to use `mk-table-sync` tool.
- Use `-verbose` option first to see what operations it is going to perform.
- Use `-print` to display the actual SQL statements. Good idea to save them to a log file
- Looks good to you? Apply the log to the slave. Since this tool once died silently in the middle of syncing and left inconsistent data, we prefer to apply logs manually (one command at a time)

# Ok, slave is out of sync. What's next?

- mk-table-sync doesn't always work right. If you see that it doesn't bring tables in sync, leave it and use another strategy.
- If there are only a few differences, use mysqldump and diff to find and resolve them manually
- If there are a 100 tables with complex FK checks – forget it, just do a full refresh.