A Tale of 8T
Transportable Tablespaces Vs Mysqldump

Kristofer Grahn
Verisure Innovation
whoami?

Kristofer Grahn (kristofer.grahn@verisure.com)

- Verisure (Sweden)
- Senior Systems Specialist
  - But mostly Dba :)
  - Mysql
  - Cassandra
<table>
<thead>
<tr>
<th>Product Lineup</th>
<th>Service Lineup</th>
</tr>
</thead>
</table>
| ![Image of products](image1.png) | • 24/7 prof. monitored intruder alarm handling  
• Professionally installed and maintained  
• Smoke & fire, water leakage, SOS alarm handling  
• Alarm centre can remotely see pictures taken by cameras  
• Direct user access via App, Web:  
  • Monitor temperature and control heatpump  
  • Smart plugs |

**Europe's most widely installed home alarm**

| ![Image of products](image2.png) | • 24/7 prof. monitored intruder alarm handling  
• Professionally installed and maintained with yearly customer visits +  
  24/7 technical service  
• Smoke & fire, water leakage, SOS alarm handling  
• Alarm centre, user can remotely see pictures taken by cameras  
• Alarm centre calls back via alarm panel  
• Direct user access via App, Web:  
  - See pictures from security cameras |
Datawarehouse setup

- Used for BI + Troubleshooting
- 45 shards ≡ Channels and counting (+10 next week)
- Challenges
  - Physical backups can't be used to merge several instances
  - TB sized databases and `mysqldump`, not efficient
  - Load of data must be fast, or replication will never catch up.
  - Poor catchup when channels > No of Cores
  - For 5.6
    - Partitioned tables not supported for `IMPORT TABLESPACE`.
MysqlDump

- Export / Import to .sql files
- Need to lock data on source server!
  - --single-transaction
  - --master-data (for replication)
- Both Source and Destination parse the Sql
- Flexible
  - Will solve DDL AutoMagically
  - But Sloooow (And heavy on the instance)
- Example
  - 186Gb
  - Dump : 80m
  - Prepare/Export : NA
  - Import : 802m
  - Import no replication : 652m
Dump and Load using mysqldump

- Dump the schema using mysqldump
  - `--single-transaction --master-data=2 --triggers --routines`
- Restore
  - `mysql < dump.sql`
- Wait, Wait, Wait

- Tip
  - `pv`, will give a nice processbar..
Transportable tablespaces

• Export / Import .ibd files between Instances
• But xtrabackup ‘–export’ is easier
• Import takes time
  • But main limit is ‘cp’ bandwith
• Import only read’s .ibd metadata fast for big files
• Example
  • 186Gb
  • Dump : 18m
  • Prepare/Export : 5m
  • UnStream : 34m
  • Import : 628m
  • Import No Replication : 162m
Dump and Load using TTS

- Dump the data using xtrabackup
  - `--export --prepare`
- Dump the schema and/or table using mysqldump
  - `--no-data --triggers --routines`
- Restore the DDL
  - `mysql < ddl.sql`
- Load the data
  - `discard tablespace`
  - `- cp`
  - `- import tablespace`
Tip’s and tricks for TTS

• Import and replication is not friends
  • Congestion ?
  • Metadata lock ?
• ‘STOP SLAVE’ while running import helps
  • But keep track of Binlog retention
• ‘REMOVE PARTITIONING’ is slow
  • Try running only once if many partitions
Dump and Load Partitions

• 5.6
  • Not supported, but
    • Import each partition as a separate table
    • Add to table using ‘EXCHANGE PARTITION’

• 5.7
  • Supported !
  • Discard
  • Cp
  • Import
Dump and Load Partitions in 5.6

• Create table on destination instance
  • ‘Show create table’
• Get list of partitions
  • ‘SELECT partition_name FROM information_schema.PARTITIONS where table_schema=inTableSchema and table_name=inTableName;’
• For each partition in (part1 part2)
  • ‘CREATE TABLE inTableSchema.inTableName_part1 LIKE inTableSchema.inTableName;’
  • ‘ALTER TABLE inTableSchema.inTableName_part1 REMOVE PARTITIONING;’
  • ‘ALTER TABLE inTableSchema.inTableName_part1 DISCARD TABLESPACE;’
  • ‘cp part1.ibd -> inTable_name_part1.ibd’
  • ‘ALTER TABLE inTableSchema.inTableName_part IMPORT TABLESPACE;’
  • ‘ALTER TABLE inTableSchema.inTableName EXCHANGE PARTITION part1 WITH TABLE inTableSchema.inTableName_part1;’
Row format misery

Diffrent default innodb row_format in Versions (Partitions created over time)
• 5.5 : COMPACT
• 5.6 : COMPACT
• 5.7 : DYNAMIC
• 8.0 : DYNAMIC

Howto Check :
`select @@innodb_default_row_format;`

```
+--------------------------------+
| @@innodb_default_row_format |)
+--------------------------------+
| dynamic                   |
+--------------------------------+
```

Error on import for mismatch :
Starting load of tablespace : ERROR 1808 (HY000) at line 1: Schema mismatch (Table flags don't match, server table has 0x29 and the meta-data file has 0x1)
Checking for rowformat Misery
For table log in logdb

```sql
USE information_schema;
SELECT substring_index(name,'
',1) AS Tbl ,file_format,row_format,count(1) Partitions
FROM INNODB_SYS_TABLES
WHERE name like 'logdb/logtable%'
GROUP BY 1,file_format,row_format order by 1;
```

<table>
<thead>
<tr>
<th>Tbl</th>
<th>file_format</th>
<th>row_format</th>
<th>Partitions</th>
</tr>
</thead>
<tbody>
<tr>
<td>logdb/logtable</td>
<td>Antelope</td>
<td>Compact</td>
<td>48</td>
</tr>
<tr>
<td>logdb/logtable</td>
<td>Barracuda</td>
<td>Dynamic</td>
<td>25</td>
</tr>
</tbody>
</table>

```
Solving the rowformat misery
Long way...

- ALTER TABLE inTableSchema.inTableName_part FORCE;
- Create the destination table one partition at a time
  - `set innodb_default_rowformat='dynamic';`
  - `create partition`
  - `set innodb_default_rowformat='compact';`
  - `create partition`
- Then run load as usual..
Some related bugs/Feature requests
Regarding create table like and row_format

- Big Thank’s to Jean-François Gagné for taking the time to write them :)

- Links
Rate My Session

Introducing gh-ost: triggerless, painless, trusted online schema migrations

Rate & Review

Tap a star to rate

Feedback (optional)

Anonymously

Submit