Acronis Cyber Cloud: Percona to MariaDB migration experience

Alexander Andreev
Chief Architect
25+ years in cloud software industry. Experienced in the Linux Kernel, Hypervisors, Storages, Databases, Cloud services architecture

Mikhail Balaian
Chief Database Architect
20 years in IT. Experienced in designing and building highly scalable and HA systems, zero downtime migrations, ETL processes
Agenda

- Acronis Cyber Cloud overview
- Percona-to-MariaDB migration project
- Acronis contribution to OpenSource
  - Contribution to MariaDB
  - Contribution to Percona
  - The acronis-db-bench (DB performance swiss-knife)
- Future Plans / Points of Acronis Interest
Agenda

- Acronis Cyber Cloud overview
- Percona-to-MariaDB migration project
- Acronis contribution to OpenSource
  - Contribution to MariaDB
  - Contribution to Percona
  - The acronis-db-bench (DB performance swiss-knife)
- Future Plans / Points of Acronis Interest
Acronis Cyber Cloud Architecture

Production environment:
- 50+ data centers
- 2K DB instances
- 5K logical DBs
- 0.35M tables
- 50TB data in DB
- 50K connections
- 10B transactions per day
- 300K updated rows per second
- 5K microservices – DB clients (globally)
- 1M rows per second returned to DB clients
- Millions of agents

Agents report their telemetry to SaaS services and so to DBs
Agenda

- Acronis Cyber Cloud overview
- Percona-to-MariaDB migration project
- Acronis contribution to OpenSource
  - Contribution to MariaDB
  - Contribution to Percona
  - The acronis-db-bench (DB performance swiss-knife)
- Future Plans
Percona-to-MariaDB Migration Project

Acronis Context, as of 2023:

1. Acronis Cyber Cloud Platform database stack:
   - Percona 5.7 database (EOL Q4'2023)
   - Galera replication
   - HAProxy
   - Percona Monitoring and Management
   - Backup, automation, and deployment scripts

2. Michael “Monty” Widenius joined Acronis

3. NOTE: Acronis run Percona on it’s own and did not use Percona support services

Migration project objectives:

1. To migrate from Percona 5.7 database to the latest stable MariaDB 10.11 with lowest possible downtime (< 5min)
2. To keep the environment as is: Galera; HA-proxy; Percona Monitoring; Backup, automation and deployment scripts
3. To ensure migration would not introduce unexpected performance problems in production
4. To be able to contribute to MariaDB and use the features in Production afterwards

Project timeline 2023’Q4 – 2024’Q1
Percona Management and Monitoring (Query Analytics)

Why we ❤️ PMM
Percona-to-MariaDB Migration Project

Original Percona cluster:

- Acronis Cyber Cloud Microservice
- Main SQL traffic
- HAPROXY
  - Main SQL traffic
- Replication
- Data files

Migration procedure:

1. Stop applications access on HAPROXY
2. Backup old Percona configs
3. Set `innodb_fast_shutdown = 0`
4. Stop `mysql`
5. Delete Percona packages on all nodes, in reverse order
6. Delete old configs (`/etc/my.cnf` and `/etc/my.cnf.d`)
7. Install new mariadb+galera packages
8. Deploy new mariadb configs in `/etc/my.cnf.d`/
9. Configure mariadb systemd unit:
   - increase max open files
   - increase systemd startup timeout
10. Start mariadb nodes one-by-one, in direct order
11. Sequentially Run `mysql_upgrade` on all three nodes
12. Enable applications access on haproxy

Total downtime is just a few minutes
Percona-to-MariaDB Migration Project

Original Percona cluster:
- Acronis Cyber Cloud Microservice
- HAPercona
- Main SQL traffic
- Percona MySQL 5.7
- Galera 3
- Main SQL traffic
- Replication
- Data files
- Deployment Scripts
- Percona Monitoring
- Backup Routines

Target MariaDB cluster:
- Acronis Cyber Cloud Microservice
- HAPercona
- Main SQL traffic
- MariaDB 10.11
- Galera 4
- Main SQL traffic
- Replication
- Data files
- Deployment Scripts
- Percona Monitoring
- Backup Routines
Agenda

- Acronis Cyber Cloud overview
- Percona-to-MariaDB migration project
- Acronis contribution to OpenSource
  - Contribution to MariaDB
  - Contribution to Percona
  - The acronis-db-bench (DB performance swiss-knife)
- Future Plans
Acronis Contribution Into MariaDB (1/3)

Convert MySQL 5.7 partitioned tables (10.6)

1. Stop server, replace MySQL binaries with MariaDB
2. Start server
3. Try to access table

MariaDB [acronis_db]> select count(*) from temp_table;
ERROR 1932 (42S02): Table 'acronis_db.temp_table' doesn't exist in engine

Support for Zulu format (10.11.7)

- ISO 8601, dates like '2023-05-21T14:40:39.046Z'

MariaDB [testdb]> insert into example_table (timestamp_column) values ('2023-05-21T14:40:39.046Z');
ERROR 1292 (22007): Incorrect datetime value: '2023-05-21T14:40:39.046Z' for column `testdb`.`example_table`.`timestamp_column` at row 1

© Acronis 2024
Log warnings into sql_error_log (10.11.7)

```sql
sql_error_log_warnings = ON
```

More statistics in slow log (11.7)

- Pages_accessed, Pages_read, Pages_updated
- Pages_read_time, Engine_time
- Tmp_tables, Tmp_disk_tables, Tmp_tables_sizes
- Full_scan, Full_join, Tmp_tables, Tmp_table_on_disk
- Filesort, Filesort_on_disk, Merge Passes, Priority queue
Ignore log_slow_rate_limit for slow queries (11.7)

log_slow_rate_limit=10
# log each 10th occurrence of the query

log_slow_always_query_time=0.1
# log unconditionally queries slower than 100ms

Limits on binlog size (11.7)

binlog_space_limit MDEV-31404
# limiting disk space used by binlogs

max_binlog_total_size
# considers if binlogs are needed by replica node

Extended statistics for userstat module (11.7)

- information_schema.table_statistics:
  ROWS_INSERTED, ROWS_UPDATED, ROWS_DELETED,
  KEY_READ_HITS, KEY_READ_MISSES
- information_schema.client_statistics:
  KEY_READ_HITS, KEY_READ_MISSES
- information_schema.user_statistics:
  KEY_READ_HITS, KEY_READ_MISSES
- information_schema.index_statistics: MDEV-33151
  QUERIES

Temp usage quotas (11.7)

max_tmp_space_usage
# per user

max_total_tmp_space_usage
# for all users

Acronis Contribution Into MariaDB (3/3)
Agenda

- Acronis Cyber Cloud overview
- Percona-to-MariaDB migration project
- Acronis contribution to OpenSource
  - Contribution to MariaDB
  - Contribution to Percona
  - The acronis-db-bench (DB performance swiss-knife)
- Future Plans / Points of Acronis Interest
Acronis Contribution Into Percona


- New statistics for MariaDB: PMM-12886

- Issues with Explain Plan PMM-12151
Acronis Contribution Into Percona

Before:

After:
Agenda

- Acronis Cyber Cloud overview
- Percona-to-MariaDB migration project
- Acronis contribution to OpenSource
  - Contribution to MariaDB
  - Contribution to Percona
  - The acronis-db-bench (DB performance swiss-knife)
- Future Plans / Points of Acronis Interest
Open Sourced Benchmark With Multi-tenant-like SQL

Acronis Context:

1. Acronis Cyber Cloud instance consists of 150+ microservices
2. Each service stores all the customers & services data in single (own) database
3. Such model requires multi-tenant and multi-service access check logic to be implemented directly in SQL
4. These SQL queries could be complicated and require specific optimization techniques, DB tuning or use of no-SQL

The acronis-db-bench idea:

1. To collect all the typical SQL query patterns used by Acronis multi-tenant services in single benchmark
2. To develop a benchmark which can test simple-to-complex scenarios – SELECT, INSERT, UPDATE, etc
3. Can be used to compare different DB configurations or solutions (Percona, MariaDB, PostgreSQL, Cassandra, etc)
4. To make it open source to get feedback, submit performance tickets to community or verify optimisations

Sources available @ https://github.com/acronis/perfkit/ - written in go-lang, ~10K LOC
Acronis Multi-Tenancy and Multi-Service Access Check

Acronis Cyber Cloud hierarchical multi-tenancy model:

Distributor 1
- Partner 1.1
  - Folder 1.1.1
    - Customer 1.1.1.1
    - Customer 1.1.1.2
  - Folder 1.1.2
    - Customer 1.1.2.1
    - Customer 1.1.2

Distributor 2
- Partner 2.1
  - Customer 2.1.1
  - Customer 2.1.2

Tenant scope
Service scope
Partner scope

Service scope
(1) Tenants isolation requires object in database to be associated with a tenant ID
(2) Services isolation requires objects in database to be associated with a service ID
Service #1 data visible to Customer 1.1.1.1

Acronis Cyber Cloud DB data

SaaS Service #1 Data
SaaS Service #2 Data
The acronis-db-bench - testing methodology

1. Use several type of tables (simple-to-complex)
   - Synthetic 'light' table – 2 columns, 0 indexes
   - Synthetic 'medium' table – 5 columns, 1 index
   - Synthetic 'heavy' table – 40 columns, 20 indexes
   - A bunch of realistic tables (with multi-tenancy support)

2. Pre-defined set of queries flavours (INSERT, SELECT, UPDATE, etc) gradually increases the complexity, like:
   - BEGIN; INSERT; INSERT ; ... ; COMMIT
   - INSERT INTO () VALUES (...,...)
   - INSERT INTO () VALUES (...,...) // with prepared statement
   - COPY ()
   - SELECT 1
   - SELECT ... WHERE {data filter}
   - SELECT ... WHERE {data filter} AND {tenant filter}
   - SELECT ... WHERE {data filter} AND {tenant filter} AND {service filter}
   - ...

3. Supported patterns:
   - INSERT, UPDATE, SELECT queries
   - JSON insert / search
   - Large blobs data INSERT / SELECT
   - Sequence generation simulation
   - Custom query from command line

4. Benchmark parameters:
   - Test duration
   - Number of queries
   - Concurrency
   - Customizable cardinality
   - Optional logging / EXPLAIN
   - Raw SQL vs DBR query builder
   - Different DB vendors support:
     - Percona / MariaDB / MySQL
     - PostgreSQL
     - SQLite
     - NoSQL – Cassandra, ClickHouse
The acronis-db-bench – cmdline example

Single quick test run example: insert-light vs insert-heavy

```bash
$ acronis-db-bench --driver mysql --dsn $DSN -t insert-light -i 100000 -c 16
```

```
Acronis Database Benchmark: version v1.0.0
Connected to 'mysql' database: 10.11.4-MariaDB (Source distribution)
```

```
mysql database settings checks:
- innodb_buffer_pool_size (aka primary DB cache)................. 12884901888 OK
- innodb_log_file_size (aka InnoDB redo log size)............... 2147483648 OK
- max_connections (aka max allowed number of DB connections).... 2048 OK
- query_cache_type (aka query cache policy)...................... OFF OK
- performance_schema (aka performance-related server metrics).... ON OK
```

test: insert-light; rows before test: 120171; time: 5.7 sec; workers: 16; loops: 100000; batch: 1; rate: 17457 rows/sec

```
$ acronis-db-bench --driver mysql --dsn $DSN -t insert-heavy -i 100000 -c 16
```

```
Acronis Database Benchmark: version v1.0.0
Connected to 'mysql' database: 10.11.4-MariaDB (Source distribution)
```

```
... test: insert-heavy; rows before test: 60; time: 9.8 sec; workers: 16; loops: 100000; batch: 1; rate: 10246 rows/sec
```
The acronis-db-bench – Integration with the Perftracker

Integration with the Perftracker (https://github.com/perfguru87/perftracker)
The acronis-db-bench – explore your databases

- Compare **simple table** performance vs **complex** ones
- Compare **simple queries** vs **complex** ones
- Compare single **worker** vs N workers
- Compare single query vs N queries in **transaction**
- Compare DB **engines** (InnoDB vs Aria)
- Compare DB **vendors** (MariaDB vs PostgreSQL)
- Compare DB **versions** (10.11 vs 11.4)
- Compare **different hardware**
- Compare single DB **instance** vs **cluster**
- Compare default DB **config** vs **tuned**
- Compare **enough-memory** vs **memory-shortage**
- Compare standalone **DB** vs **DB-in-kubernetes**
- Compare your **on-prem DB** vs **AWS-hosted**
- Compare **tables/indexes** disk consumption space on different storage/DB engines
- …
Agenda

- Acronis Cyber Cloud overview
- Percona-to-MariaDB migration project
- Acronis contribution to OpenSource
  - Contribution to MariaDB
  - Contribution to Percona
  - The acronis-db-bench (DB performance swiss-knife)
- Future Plans
# Future Plans / Points of Acronis Interest

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>1</strong></td>
<td>Non-blocking backup in Community Server <a href="10.11.8">MDEV-20644</a></td>
</tr>
<tr>
<td><strong>2</strong></td>
<td>Extended error messages</td>
</tr>
<tr>
<td></td>
<td>Adding table, column and index names to error messages (10.11)</td>
</tr>
<tr>
<td><strong>3</strong></td>
<td>Extended temp usage monitoring and quotas</td>
</tr>
<tr>
<td></td>
<td>• TEMPORARY TABLE relations (11.7)</td>
</tr>
<tr>
<td><strong>4</strong></td>
<td>Better monitoring capabilities</td>
</tr>
<tr>
<td></td>
<td>• More metrics (11.7)</td>
</tr>
<tr>
<td></td>
<td>• Visualizing in Percona Management and Monitoring (11.7)</td>
</tr>
<tr>
<td><strong>5</strong></td>
<td>Catalogs</td>
</tr>
<tr>
<td></td>
<td>HW resources isolation (11.7)</td>
</tr>
<tr>
<td><strong>6</strong></td>
<td>Built-in analog of pt-osc, gh-ost</td>
</tr>
<tr>
<td></td>
<td>Zero downtime migrations (11.7)</td>
</tr>
<tr>
<td><strong>7</strong></td>
<td>Avoid stall in Galera during huge table drop</td>
</tr>
<tr>
<td><strong>8</strong></td>
<td>Create foreign keys without downtime</td>
</tr>
<tr>
<td><strong>9</strong></td>
<td>Performance</td>
</tr>
<tr>
<td></td>
<td>MDEV-6096 (Parallel query execution), multi-tenant-like queries</td>
</tr>
<tr>
<td><strong>10</strong></td>
<td>No SQL</td>
</tr>
<tr>
<td></td>
<td>Vector database for AI models, indexable JSON</td>
</tr>
</tbody>
</table>
Q&A

Contacts:
Alexander.Andreev@acronis.com
Mikhail.Balayan@acronis.com