Too Much Data

Yves Trudeau
Percona
Who Am I?

• Principal architect at Percona since 2009 (10 years already…)
• With Sun Microsystems and MySQL before Percona
• Physicist by training
• I like to understand how things work
Why This Talk?

- I often see customers struggling with high inflow of data and large datasets
- A lot can be done to improve the situation
- It is a nice case to study, with many options
- Allows us to uncover the inner behavior of MySQL
Plan

• The issue
• Benchmarking procedure
• Iterations/results/investigation
• Data compression options
Typical Metric Oriented Schema

Columns:
• A device identifier (deviceld)
• A metric identifier (metricld)
• A metric value (value)
• A timestamp (TS)
Insert Rate Versus Report Queries?

- High insert rate, likely many 1000 rows/s
- Queries will need to retrieve and aggregate “value” by (device|Id, metric|Id)
- Size >> memory
- Wide versus narrow (wide is 1 column/metric)
- This talk = only narrow
The Benchmarking Procedure
A Synthetic Data Set

- A python script creates 14k devices each with 1 to 10 metrics
- On average, 5.5 metrics/device
- 77000 metrics
- 5000 insert periods
- 385M rows
- Goals: insert rate AND decent query time
- VM capped at 500 iops, 2GB Ram (BP 1GB), 2 vcpu, PS 5.7.25
The Report Query

- A very simple query:

```sql
select deviceId, avg(value), min(value), max(value), sum(value), count(*)
from Metrics
where deviceId in (102,103,104,105,106)
   and metricId = 1
group by deviceId;
```

- Looking at the execution time
Results/Investigation
Iteration #1: auto-increment id PK

CREATE TABLE `CollectorMetrics` (  
`ID` int(10) unsigned NOT NULL AUTO_INCREMENT,  
`deviceId` int(10) unsigned NOT NULL,  
`metricId` smallint(5) unsigned NOT NULL,  
`Value` float NOT NULL,  
`TS` int(10) unsigned NOT NULL,  
PRIMARY KEY (`ID`),  
KEY `idx_nat` (`deviceId`,`metricId`,`TS`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1
Iteration #1: Results

Query time: 48.5s  (24235 read Iops)  Size: 23GB
Iteration #2: Natural PK

CREATE TABLE `CollectorMetrics` (  
`deviceId` int(10) unsigned NOT NULL,  
`metricId` smallint(5) unsigned NOT NULL,  
`Value` float NOT NULL,  
`TS` int(10) unsigned NOT NULL,  
PRIMARY KEY (`deviceId`,`metricId`,`TS`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1
Iteration #2: Results

Not an option, but… stopped at 67M rows, 2.76 GB
CREATE TABLE `CollectorMetrics` (  `deviceId` int(10) unsigned NOT NULL,  `metricId` smallint(5) unsigned NOT NULL,  `Value` float NOT NULL,  `TS` int(10) unsigned NOT NULL,  PRIMARY KEY (`deviceId`,`metricId`,`TS`) ) ENGINE=InnoDB DEFAULT CHARSET=latin1  /*!50100 PARTITION BY RANGE (TS)  */  PARTITION p1555691218 VALUES LESS THAN (1555691218) ENGINE = InnoDB,  PARTITION p1555692118 VALUES LESS THAN (1555692118) ENGINE = InnoDB,  PARTITION p1555693018 VALUES LESS THAN (1555693018) ENGINE = InnoDB,  ...  PARTITION p1555719118 VALUES LESS THAN (1555719118) ENGINE = InnoDB,  PARTITION pMax VALUES LESS THAN (4294967295) ENGINE = InnoDB) */

32 partitions
Iteration #3: Results

Query time: 0.57s  (320 read Iops)  
Size: 21.7GB
Can This Really Work?

- Partition sizes tuned for ~ BP size
- On the fly
- Only one partition really used at a time
- Up to 8192 partitions…
- 100GB BP → up to 800 TB
Compression Options
InnoDB Compression (8KB)

Query time: 0.79s (416 read Iops)
Size: 11.5 GB
ZFS (Lz4)

Query time: 0.97 s  (316 read Iops)  
Size: 9.4 GB
ZFS (Gzip)

Query time: 0.66s  (192 read Iops)  
Size: 7.0 GB
Summary
Thank You to Our Sponsors
Rate My Session

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

11:20 → 12:10, Matterhorn 2

Rate & Review

Tap a star to rate

Feedback (optional)

Anonymously

Submit