SOLARWINDS

Clickhouse Austin Meetup

Tony Burke

Clickhouse at SolarWinds

What are the characteristics of our Clickhouse use

SaaS Platform engineering

- Ingestion
  - Received Messages
  - Transformation
  - Queueing
- Services
  - Ingress Endpoint
  - Clickhouse Writes
  - Clickhouse Query
- Clickhouse
  - Schema
  - Management
  - Support

About our data

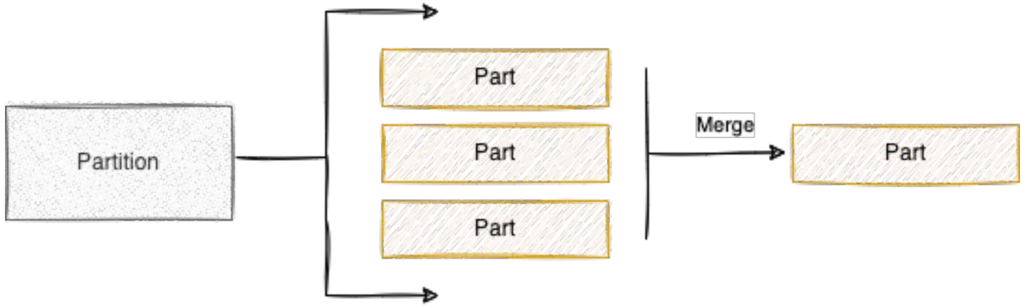
- Telemetry (Metrics, Logs and Traces)
  - Devices (hosts, firewalls, switches, logs)
  - Applications (traces, logs, and metrics)
  - Databases (SQL statements, metrics)
  - Kubernetes (metrics, logs)
  - . . .
- Real-time customer-facing multi-tenant
- ~3M messages ingested per second (1 cluster)
  - ~550MB – 1G per second

Research: Metrics Primary Key

Can we change the primary key time boundary to improve hot queries

First, Understand How Clickhouse

- Stores and manages data
  - MergeTree table engine
  - Physical layout
- Resolve queries
  - Select Rows
  - Indexing
- What are the query characteristics
  - Frequent time-range queries

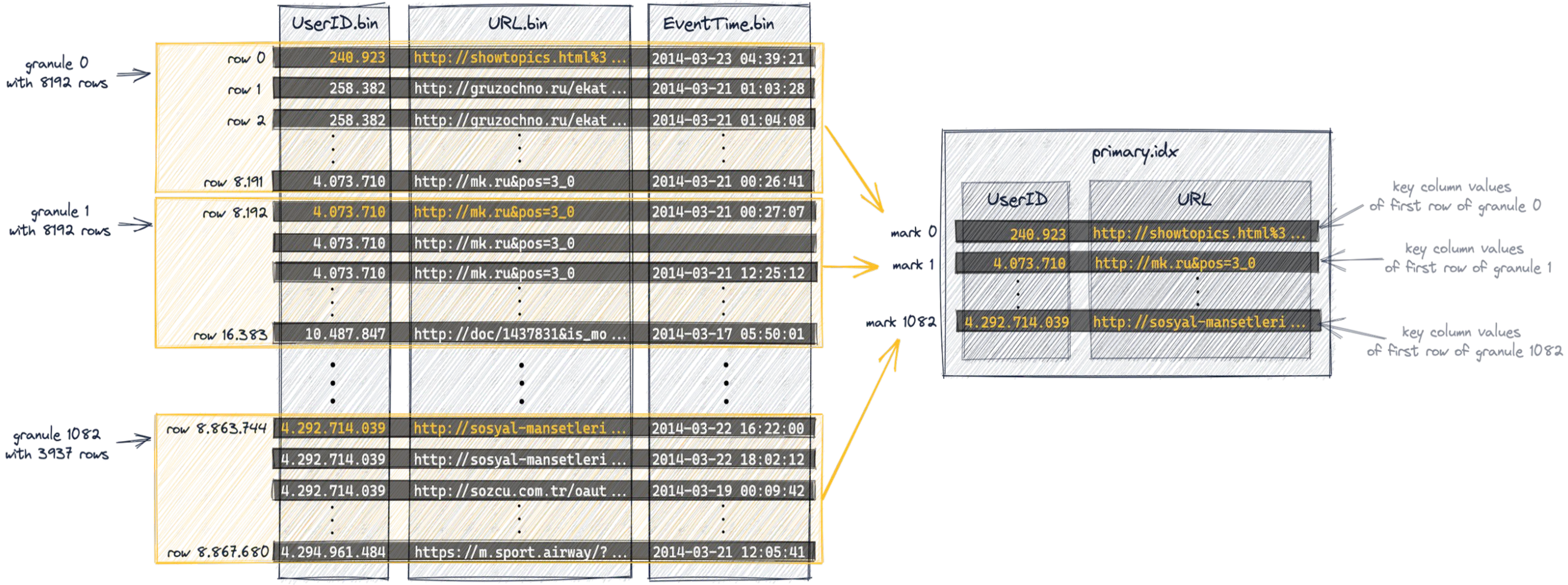


Most queries are for the latest 60 minutes

- Interactive users
- Alert evaluations

Clickhouse Indexing

Contents of a part with index_granularity of 8192 (default)



	UserID.bin	URL.bin	EventTime.bin
row 0	240.923	http://showtopics.html%3 ...	2014-03-23 04:39:21
row 1	258.382	http://gruzochno.ru/ekat ...	2014-03-21 01:03:28
row 2	258.382	http://gruzochno.ru/ekat ...	2014-03-21 01:04:08
	⋮	⋮	⋮
row 8.191	4.073.710	http://mk.ru&pos=3_0	2014-03-21 00:26:41
row 8.192	4.073.710	http://mk.ru&pos=3_0	2014-03-21 00:27:07
	4.073.710	http://mk.ru&pos=3_0	
	4.073.710	http://mk.ru&pos=3_0	2014-03-21 12:25:12
	⋮	⋮	⋮
row 16.383	10.487.847	http://doc/1437831&is_mo ...	2014-03-17 05:50:01
	⋮	⋮	⋮
row 8.863.744	4.292.714.039	http://sosyal-mansetleri ...	2014-03-22 16:22:00
	4.292.714.039	http://sosyal-mansetleri ...	2014-03-22 18:02:12
	4.292.714.039	http://sozcu.com.tr/oaut ...	2014-03-19 00:09:42
	⋮	⋮	⋮
row 8.867.680	4.294.961.484	https://m.sport.airway/? ...	2014-03-21 12:05:41

primary.idx

	UserID	URL
mark 0	240.923	http://showtopics.html%3 ...
mark 1	4.073.710	http://mk.ru&pos=3_0
	⋮	⋮
mark 1082	4.292.714.039	http://sosyal-mansetleri ...

Primary Key Effectiveness

How would a change to the primary key (time-bound) affect performance

Analysis

- The last 60min is the hot query data
  - Not a lot of time for parts to merge
- The primary.idx
  - Few marks for the latest data
  - Cardinality of the first 2 columns are high.

    (tenant, namespace, hour)
- Conclusion:
  - Any change to the primary key would have little effect on reducing the marks selected.

Reduce Index_granularity

- Use a Materialized View to mirror table data
  - The only difference is index_granularity
- Use our Clickhouse query service to
  - Alternate queries between the two tables
- Measure
  - Query Performance
  - Background Tasks (Merges)
  - Memory usage

query_log metrics

Measure queries for both tables and compare results

index_granularity=1024

```
┌─queries─┬─avg_cpu_seconds─┬─avg_query_time_ms─┬─avg_read_parts─┬─avg_read_marks─┬─avg_read_rows─┬─time_window_in_mins─┐
│  265602 │            0.07 │             76.52 │           9.41 │           9.45 │       9555.86 │                   5 │
│  186923 │            0.09 │             98.38 │           9.53 │          10.31 │      10553.26 │                  15 │
│  160668 │            0.09 │              92.2 │           9.43 │           9.84 │      10073.83 │                   2 │
│   59489 │            0.08 │              79.8 │           9.37 │           10.3 │      10528.28 │                   1 │
│   49987 │            0.09 │             82.16 │           9.21 │           11.2 │      12191.19 │                  60 │
│   17793 │             0.1 │            117.82 │           9.78 │          13.19 │       14224.2 │                  30 │
└─────────┴─────────────────┴───────────────────┴────────────────┴────────────────┴───────────────┴─────────────────────┘
```

index_granularity=32

```
┌─queries─┬─avg_cpu_seconds─┬─avg_query_time_ms─┬─avg_read_parts─┬─avg_read_marks─┬─avg_read_rows─┬─time_window_in_mins─┐
│  200246 │            0.03 │             27.88 │           9.91 │          11.63 │        374.05 │                   5 │
│  140138 │            0.07 │             59.05 │          10.13 │          32.15 │       1378.26 │                  15 │
│  120856 │            0.05 │             44.88 │           9.93 │          23.64 │        957.41 │                   2 │
│   45007 │            0.07 │             50.58 │           9.86 │          26.26 │       1069.82 │                   1 │
│   38591 │            0.12 │             93.55 │           9.84 │           73.5 │       3542.48 │                  60 │
│   15520 │            0.11 │            106.97 │          10.15 │           79.7 │       3126.16 │                  30 │
└─────────┴─────────────────┴───────────────────┴────────────────┴────────────────┴───────────────┴─────────────────────┘
```

Observations

The side effects of reducing the index_granularity

What should I watch?

- An increase in merge time
  - peak_memory_usage and duration_ms in **system.part_log**
- Primary index file size
  - primary_key_bytes_in_memory and primary_key_bytes_in_memory_allocated in **system.parts**
- Column mark size
  - mark_cache_size (default 5G) LRU cache
  - MarkCacheHits and MarkCacheMisses in **system.events**

Other recent tuning

- Physical file reads/writes
  - local_filesystem_read_method
  - The default setting is pread_threadpool is used to decrease the chances of exhausting the open files limit.
  - This negatively impacted filesystem reads in our environments. We have high number of concurrent queries and fast SSD drives. We set this value to pread and saw a nice performance improvement.

THANK YOU