EVALUATING INFLUXDB AND CLICKHOUSE DATABASE TECHNOLOGIES FOR IMPROVEMENTS OF THE ATLAS OPERATIONAL MONITORING DATA ARCHIVING

Introduction

The *Trigger and Data Acquisition (TDAQ)* system of the *ATLAS* experiment the *Large Hadron Collider (LHC)* at *CERN* currently is composed of a large number of distributed hardware and software components (about 3000 machines and more than 25000 applications) which, in a coordinated manner, provide the data-taking functionality of the overall system.

During data taking runs, a huge flow of operational data is produced in order to constantly monitor the system and allow proper detection of anomalies or misbehaviors. The *Persistent Back-End for the ATLAS Information System of TDAQ (P-BEAST)*[1] is a system based on a custom-built time-series database and it is used to archive and retrieve for applications any operational monitoring data. *P-BEAST* stores about 18 TB of highly compacted and compressed raw monitoring data per year acquired at 200 kHz average information update rate during *ATLAS* data taking periods.

Why

Since *P-BEAST* has been put into production, 4 years ago, several promising database technologies for fast access to time-series and column-oriented data have become available. **InfluxDB**[2] and **ClickHouse**[3] were the most promising candidates for improving the performance and functionality of the current implementation of *P-BEAST*.

A series of synthetic tests have been ran on both database systems which try to leverage the best possible options for storage of the *P-BEAST* data using their respective data model capabilities.

These performance tests have been performed using a subset of archived *ATLAS* operational monitoring data.

How (strategy)

Alternatives

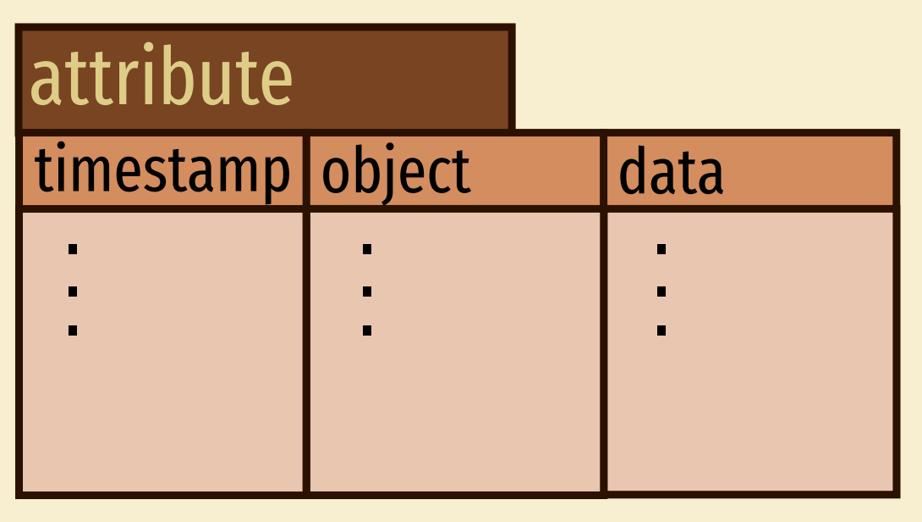
P-BEAST can store integers, floats, strings and arrays and structures of the above. It also has support for schema evolution/modifications.

InfluxDB:
- Time-series database
- Support for storing integers, floats, strings and no control over signedness and word size
- Schema-less
- High write speed (reportedly)

ClickHouse:
- Columnar database
- Support for storing integers, floats, strings, arrays and control over signedness and word size
- High write speed (reportedly)

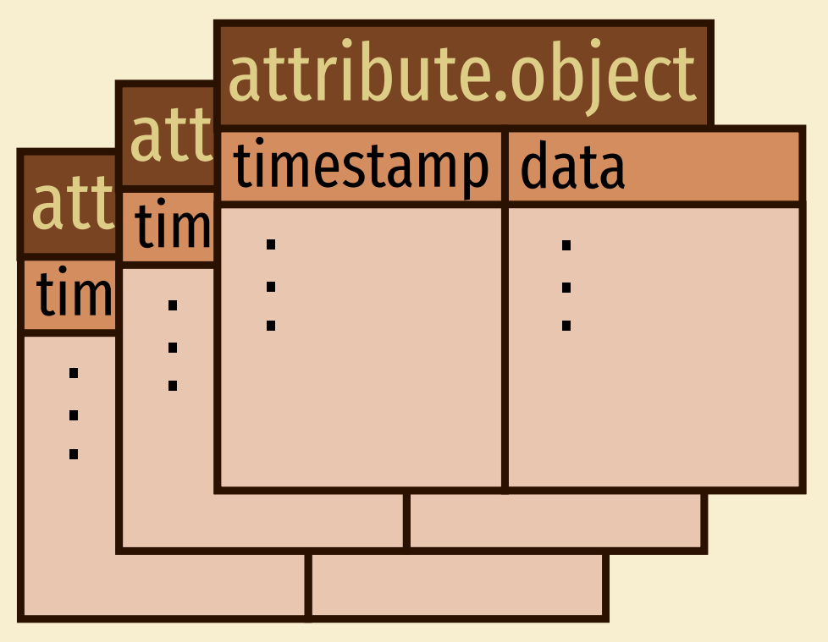
Tested data models

InfluxDB #1



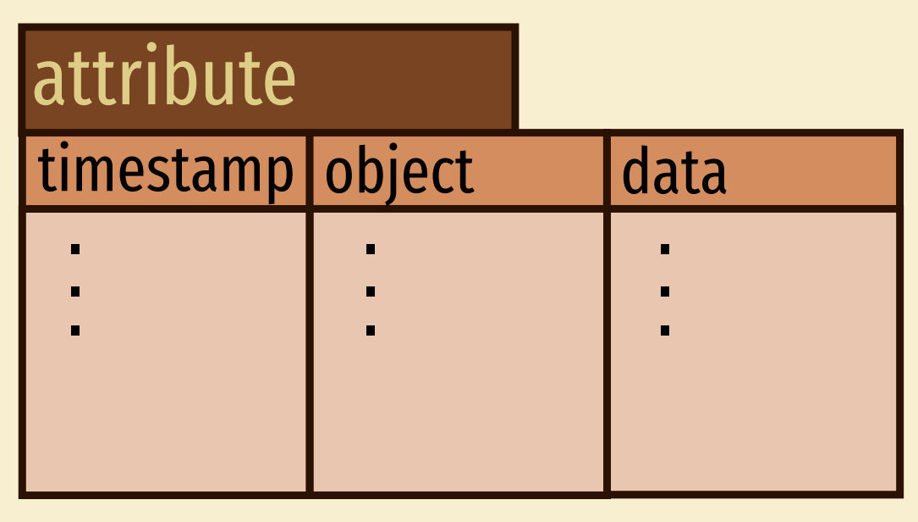
- Single table (*measurement* in *InfluxDB* terminology)
- Timestamp and a *tag* (*InfluxDB* indexed column) containing the object name make up the primary key

InfluxDB #2



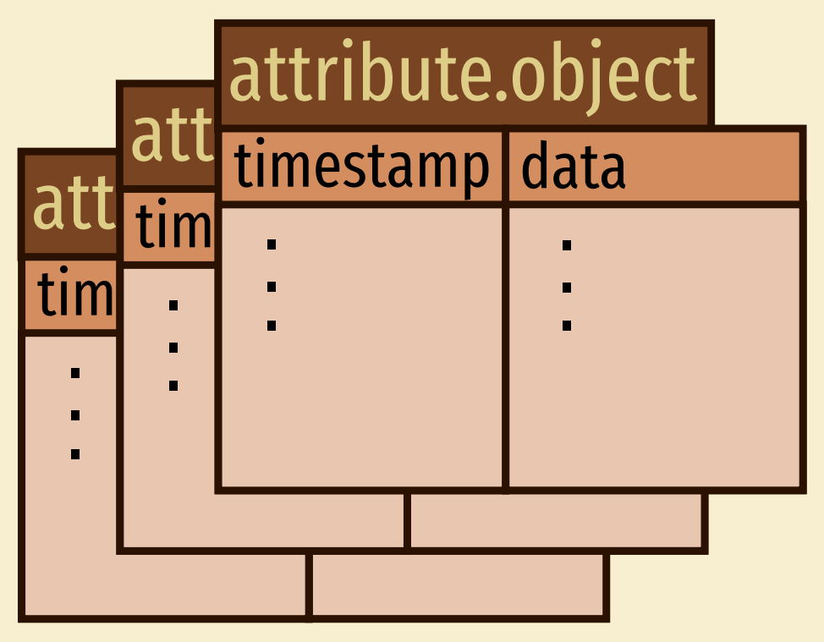
- Multiple tables (*measurements* in *InfluxDB* terminology)
- Timestamp is the primary key in each table
- Each table contains data from a single object
- The object name is stored as part of the table name

ClickHouse #1



- Single table
- Columns containing the timestamp and the object name make up the primary key
- The object name column is used to create *partitions* (*ClickHouse* concept)

ClickHouse #2



- Multiple tables
- Column containing the timestamp is the primary key in each table
- Each table contains data from a single object
- The object name is stored as part of the table name

- **InfluxDB**: Arrays are stored using one column (*field* in *InfluxDB* terminology) for each element of the array
- **ClickHouse**: Arrays are stored in a single column, by using *ClickHouse*'s support for arrays
- For versioning purposes, the data types are stored as part of the column name for all tested data models

How (implementation)

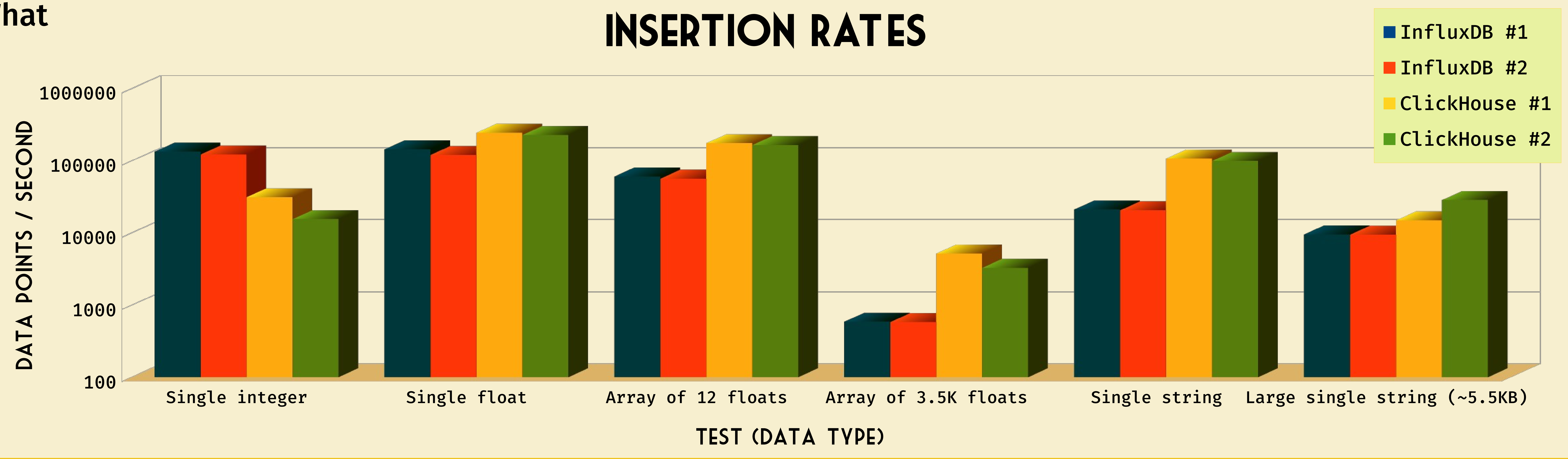
Software

- The 4 tested data models have each been implemented as a separate test that was run on 6 types of *ATLAS* operational monitoring data (see below);
- All the tests have been implemented in the **Go! Programming Language**;
- **InfluxDB** is itself written in Go!, so it comes with a Go! client library from the developer;
- **ClickHouse** has multiple third-party Go! libraries.

Hardware

- All the tests have been run on a dual-CPU machine with:
  - 2 Intel Xeon E5-2630 v2 @ 2.60GHz CPUs (each with 6 cores and Hypethreading, for a total of **24 threads**) and
  - **32 GB of RAM** and
  - An 18 TB **RAID0** array using hard disk drives

What



Status and Outlook

- In the best case scenario, **ClickHouse**'s insert rate is faster than that of **InfluxDB** (all tests except the one for *Single integer* indicate this);
- What happened with the *Single integer* test? The number of objects for that attribute was very large (over 45k) and **ClickHouse**'s *partitions* feature is recommend only for up to 1k partitions, otherwise the read speed decreases significantly. It seems that the insert speed is negatively impacted as well;
- The single table approach almost always provide a better result;
- **Future work:** What happens when it comes to read speed? Or storage performance? How about those server startup times issues that were noticed? We will investigate and get back to you!

References

[1] P-BEAST - Avolio G, d'Ascanio M, Lehmann-Miotto G, Soloviev I, *"A web-based solution to visualize operational monitoring data in the Trigger and Data Acquisition system of the ATLAS experiment at the LHC"* in J. Phys.: Conf. Ser. 898 (2017) 032010 https://iopscience.iop.org/article/10.1088/1742-6596/898/3/032010
[2] InfluxDB - https://www.influxdata.com/time-series-platform/
[3] ClickHouse - https://clickhouse.yandex/

MATEI-EUGEN VASILE[1], GIUSEPPE AVOLIO[2], IGOR SOLOVIEV[3]
[1] IFIN-HH, [2] CERN, [3] UNIVERSITY OF CALIFORNIA, IRVINE

ATLAS EXPERIMENT