



# **Benchmarking tools for NextGen Archiver for WinCC OA**

**August 2019**

**AUTHOR(S):**  
Jayaditya Gupta

BE-ICS-FD

**SUPERVISOR(S):**  
Rafal Lukasz Kulaga

Anthony Hennessey



## PROJECT SPECIFICATION

.....

The objective of the project is to develop benchmarking tools for database technologies (InfluxDB, Oracle and Apache Kudu) used by WinCC OA Archiver used in CERN control systems. The goal of the project is to help in selecting the technology that will be used to store history of signal value changes and alarms in hundreds of control systems at CERN, including the ones in the accelerator complex and experiments.

Particularly we are more focused towards benchmarking InfluxDB vs Oracle database. We are using/developing two tools namely ArchivePump and QueryBenchmark. ArchivePump will measure the write speed of the database. QueryBenchmark tool will measure read speed of the database.

.....





## ABSTRACT



On this project we focused on benchmarking Influx against Oracle database. One of the primary reason is ETM/Seimens were already working on Influx database backend. To perform benchmarking using the Query Benchmark Tool we needed to have same data in both databases (Influx and Oracle). The Archive pump tool was used to do that task. After that we used the Query Benchmark Tool to get the initial test results and performance of the database. A good amount of time was spent on understanding and learning InfluxDB and also how to translate Oracle based SQL queries to InfluxQL queries.





# TABLE OF CONTENTS



- 1 Introduction**
  - a. Archiving from Wincc OA
  - b. Next Generation Archiver
  - c. Prospective storage technologies
  - d. Benchmarking
  
- 2 Method**
  - a. Architecture
  - b. Test data
  - c. Queries
  - d. Results capture
  
- 3 Results**
  - a. Understanding queries
  - b. Workflow
  
- 4 Conclusion & Future work**
  
- 5 References**





## 1. INTRODUCTION

### a. Archiving from WinCC OA

WinCC OA is a SCADA system for visualizing and operating of processes, production flows, machines and plants in all lines of business[1]. WinCC OA provides archiving functionality that stores the history of signal value changes and alarms for approximately 700 control systems at CERN. The systems that use WinCC OA include CERN electric grid, cryogenics and interlocking systems. Signal values are stored primarily for reason of to study the state of the control system, if the machines were working fine or not and also for historical retrieval of control system values. Current WinCC OA archiver writes to Oracle database. Most important queries correspond to three data retrieval scenarios

- Trending: trending gives a trend for signal values for a specified time range.
- Event Replay : event replay give information about the state of all datapoints in a given time range
- Value Change: value change gives information about specific datapoint in time range.

### b. Next Generation Archiver

CERN along with ETM/Seimens are developing the Next Generation Archiver (NGA). The NGA will support multiple pluggable backend storage technologies including Oracle and InfluxDB storage solutions are currently in development and apache kudu, InfluxDB are under consideration. There will also be the potential to write custom backends.

### c. Prospective storage technologies

CERN is currently considering InfluxDB and Apache Kudu as prospective storage technologies.

**InfluxDB:** InfluxDB is an open-source time series database which is built for high-availability storage and retrieval of time series data in fields such as operations monitoring, application metrics, Internet of Things sensor data, and real-time analytics [2]. InfluxDB integrates with Grafana to provide visualizing metrics and dashboards to easily study data.

**Apache Kudu :** Apache Kudu is also considered due to its good balance between real-time and batch processing performance and integration with data analytics tools such as Apache Spark and SQL query engines such as Apache Impala.

### d. Benchmarking

Before considering a backend storage technology for use at CERN we will benchmark the technology under conditions that reproduce as closely as possible to the operating environment.

The testing must consider reading and writing data and the data itself must reflect the quantity and frequency of production of data. In order to do these performance tests 2 tools were developed namely the Archive Pump Tool and the Query Benchmark Tool.

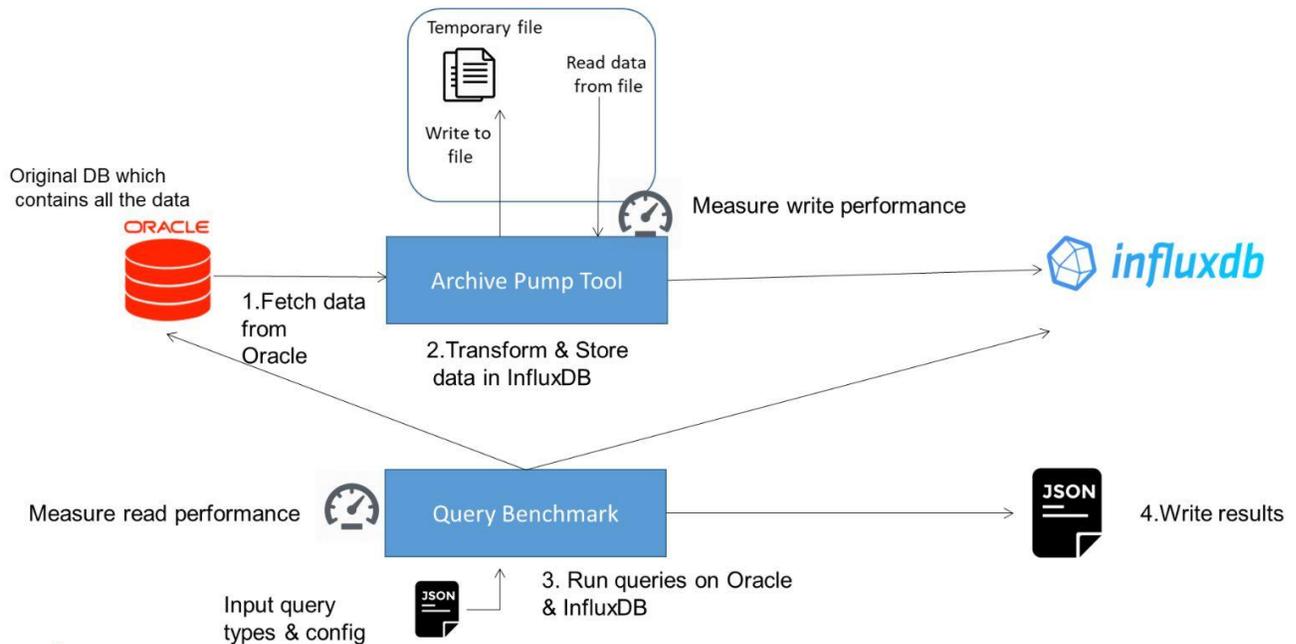
The Archive Pump Tool primary use is to copy the data from one database instance to another tool. It is also used to measure write performance of the storage technologies and the Query Benchmark Tool is used to measure read performance. Both of these tools were in an advanced stage of development and my part in the project was to add support for InfluxDB in the Query Benchmark Tool; support for Oracle in the query benchmark tool is already implemented.





## 2. Method

### a. Architecture



1. We start with fetching the data from the original oracle database(which is actual production data) and send it to archive pump tool
2. Next we extract the data from Oracle and store it in temporary files; this allows us to measure the raw write performance to influxDB without the results being influenced by oracle read performance.
3. Once the storage technologies have the same data, we use query benchmark tool to perform the same set of queries against the database technologies.
4. We measure the read performance of database technologies and save it in a json file.

### b. Test data

In order to perform tests we need actual data that is observed at CERN. We used previous 2.5 years of PSEN data. We used the Archive Pump Tool to populate that data in InfluxDB, in doing so we also recorded the write performance for InfluxDB. Our test data has

- Total signals count: ~1.4 million
- Total rows: ~11.5 billion
- Total unique datapoints/elements : 533318



### c. Queries

The queries correspond to three data retrieval scenarios: trends, event replay and value change (explained above).

In order to effectively test the storage technologies we use randomization to create more realistic scenario. If we keep on performing same queries again and again we might get unrealistic results because of caching, by doing randomization we try to avoid any unfair advantage a technology might have. First we get the list of all the data points and store it in a file, during runtime a set of datapoints are randomly chosen and the Query Benchmark Tool uses those same data points for all storage technologies.

This ensures we are running same set of queries and on same set of datapoints and time range for both the databases. It helps in effectively measuring how each database performed. In results file we store datapoints, time range of queries and read time of queries.

There is one catch in this whole process. SQL queries does not directly map to InfluxDB. InfluxDB has its own query language called InfluxQL. We had to convert oracle SQL queries to InfluxQL queries.

### d. Results capture

We only store certain metrics in our results JSON file. A sample result.json looks like this

**db:** stores the name of the database for which the test was performed.

**DurationMillisec:** stores how much time query took to execute.

**queriedDpes:** set of datapoints used in queries.

**Queried Time range(start/end):** stores the time range( start and end time) duration for the query.

**QueryTest:** stores various configurations for type of query test performed

We verify the results of the queries by matching the results of the both queries i.e. queries performed on InfluxDB and Oracle database.





### 3. Results

In this project my main contribution was to the development of the Query Benchmark Tool. Specifically I extended the functionality to include InfluxDB. Most of my time was invested in development and testing of The Query Benchmark Tool. I studied and later on converted Oracle SQL queries to InfluxQL queries, these queries were particularly challenging and took up large proportion of the project time. I also fixed various bugs and submitted upgrades to already existing Query Benchmark codebase.

We saw a write rate of roughly ~50,000 events/s in InfluxDB and that too when InfluxDB was not pushed to its limits mainly because InfluxDB has a tendency to use up all memory when faced with high load, leading to it getting killed by OOM handler.

### 4. Conclusion and future work

In order to draw out some concrete results and numbers for read and write performance of Oracle vs InfluxDB we need more enhancement and time but in our first initial tests we were able to get some interesting results such as ~50,000 event/s write rate for InfluxDB.

We need more time and testing, enhancements to decide on which database technology will be best suited for CERN use case /NGA. The nature of project is quite complex which will need several iterations on each run to get concrete results.

We plan to investigate InfluxDB configuration to make it more performant. Running The Query Benchmark Tool on full scale i.e. using it on whole PSEN database. We also plan to extend the Query Benchmark Tool to test other storage technology mainly Apache Kudu.

### 5. References

1. <https://w3.siemens.com/mcms/human-machine-interface/en/visualization-software/simatic-wincc-open-architecture/pages/default.aspx>
2. <https://www.influxdata.com/>

