PANDORA OR 'HOW TO TAME THE PLUG JUMBLE'

by Patrick Michl

23. March, 2019

Online analytical processing and predictive analytics in combination with machine learning provides a new challenge in data-warehousing: The response time for large transactions of data from different domains.

Today's data analysis and enterprise anlytical applications, increasingly utilize complex statistical models, like artificial neural networks, which demand large amounts of raw unaggregated data. Since for many of such applications, however, the response time is critical, it is becoming quite clear, that the traditional approach, to dump day-to-day data into a huge repository for data analysis, needs to be revised.

So how to create a high throughput data transaction structure with low latency? Of course, the key is decentralization! The simple idea is to directly "plug" the analysis applications into the source data systems they require. On a closer perspective, however, this idea turns out to be horrible! It not only spawns an absolutely unmanageable jumble of data interfaces (which first of all have to be implemented), but also does not provide any flexibility to the underlying structure. … Nevertheless - At no time people have been deterred of any simple idea by the argument of "a bad idea"! The result is, where we find ourselves today: In a Plug Jumble!

What is Pandora?

> " *In order to bring a little more order into the chaos, we have decided not to follow the most simple idea, but that one right after it: A multi plug!* "

Pandora is a universal data interface and SQL-Database engine, that mediates between data source systems (like operational databases) and data analysis applications. To this end Pandora implements the two fundamental layers of a data warehouse.

The **integration layer** of Pandora is implemented by a modular plugin-system, which allows it to stay light-weight, while flexibly supporting a wide variety of different data sources. The included data support comprises an SQL-plugin, which utilizes SQLAlchemy to allow it's connection to a variety of SQL-Databases (IBM Db2, Oracle Database, SAP HANA, Microsoft SQL, MySQL, Postgesql, …). There above Pandora as aimed to ship with integrated support for the most common laboratory measurement devices, flat files and data generators that appear in the wild.

The **staging layer** of Pandora is currently implemented as a native SQL-Database engine, featuring a DB-API 2.0 interface with full SQL:2016 support, a vertical data storage manager and real-time encryption. On this foundation Pandora is aimed to support sampling in common data analysis formats: NumPy-Arrays and R-Tables.

According to our convictions, Pandora is free and open-source, based on the Python programming language and actively developed as part of our Liquid Coding framework. For more information please visit GitHub.