ISBN: 978-3-95450-238-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2023-TUPDP035

NEW DEVELOPMENTS FOR eGiga2m HISTORIC DATABASE WEB VISUALIZER

L. Zambon*, R. Passuello, Elettra Sincrotrone Trieste, Trieste, Italy

Abstract

eGiga was an historic database web visualizer from 2002. At the beginning it was connected to a proprietary database schema, support for other schemas was added later, for example HDB and HDB++. eGiga was deeply refactored in 2015 becoming eGiga2m. Between 2022 and 2023 a few improvements have been made, among them, optimization of large data extraction, improvement of images and pdf exports, substitution of 3D chart library with a touch screen enabled one; the addition of: logger status info, a new canvas responsive chart library, adjustable splitter, support for TimescaleDB and HDF5 data format, correlations and time series analysis, and ARIMA (AutoRegressive Integrated Moving Average) forecast.

INTRODUCTION

eGiga2m is a web application for displaying time series as charts. It was designed to be easily embedded in other applications and also to include other web pages. In order to get this result, it is possible to display only the chart without any menu, bar or configuration dialog and all configuration parameters are reported in the address bar (they are HTTP GET parameters).

There are also a few configuration parameters which are configurable only by the address bar, for example multiple X axes as shown in Fig. 1.

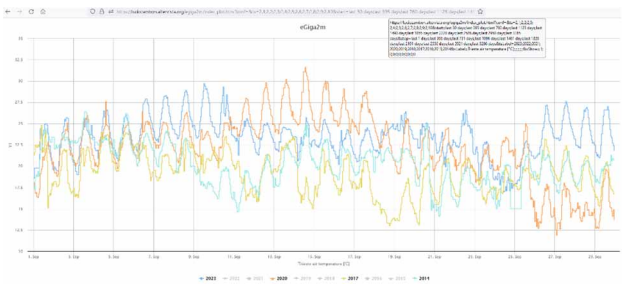

Figure 1: Multiple X axes.

Time series are provided by several data sources such as databases (for example HDB++), CSV files or numerical elaboration micro services.

The core of eGiga2m is written in JavaScript and doesn't receive data directly from a database, but it uses a web service which serves JSON encoded data.

JSON data schema is defined in JSON according to standards defined by json-schema.org [1].

Source code can be downloaded from https://gitlab.elettra.eu/puma/client/egiga2m

eGiga2m architecture is reported in Fig. 2.

* lucio.zambon@elettra.eu

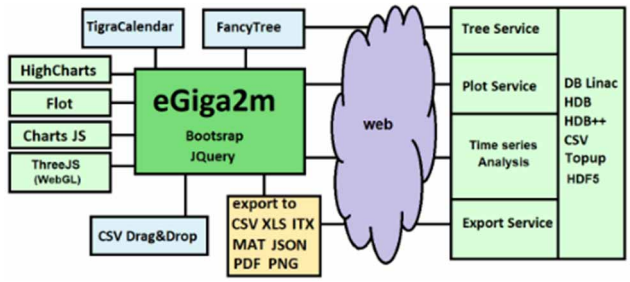

Figure 2: eGiga2m architecture.

Recent improvements are described in the following paragraphs.

NEW FEATURES

Canvas Chart Library

A new chart library, Chart.js [2], was added recently. This library isn't based on SVG but on HTML5 Canvas. SVG was defined by the W3C consortium and integrates perfectly in HTML, CSS and JavaScript, but if the chart requires a certain complexity, the browser must manage a very complex DOM and it can cause a sensible slowness. Using a canvas directly is significantly faster and makes it possible to be truly responsive; i.e. when the size of the charting area is modified the chart adapts instantly to the new dimensions.

Adjustable Separator

In the main page on the left side there is a configurator dialog composed by the start and stop time selector and the time series tree selector; on the right side there is the chart. Between the configurator dialog and the chart there is a separator bar which was made adjustable recently. It is more useful if used with Chart.js because the chart adjusts instantly moving the bar. On mobile the ideal setup is obtained by rotating the device horizontally and adjusting the separation bar as shown in Fig. 3.

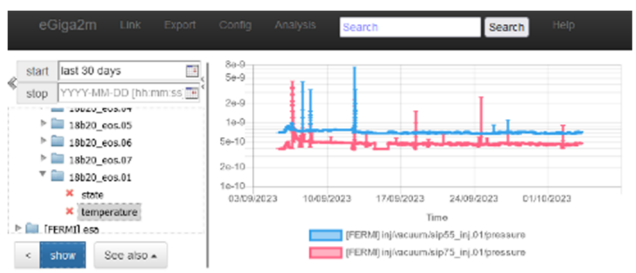

Figure 3: eGiga2m on a mobile device.

Optimization of Large Data Extraction

Some PHP servers extract data from Databases and the whole data structure is returned in a unique JSON data stream; a quite fast way to produce it is to load all data in

ISBN: 978-3-95450-238-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2023-TUPDP035

memory and JSON-ify in a single call to a function called json_encode(); but memory is limited; the first action was to configure Apache server in order to increase allowed memory, then the main query was split in chunks and JSON data was sent chunk by chunk paying great attention not to produce an invalid JSON file. But smaller chunks imply slower throughput (more overhead in transmission), so the chunks must be as large as allowed by memory (with a security margin).

3D Charts

From 2015 there was a WebGL based library for 3D plots, it worked fine but it didn't support gestures on touch screens. So in 2022 it was replaced by ThreeJS [3], it is based on WebGL, too. But it performs very efficiently also on touch screens (Fig. 4).

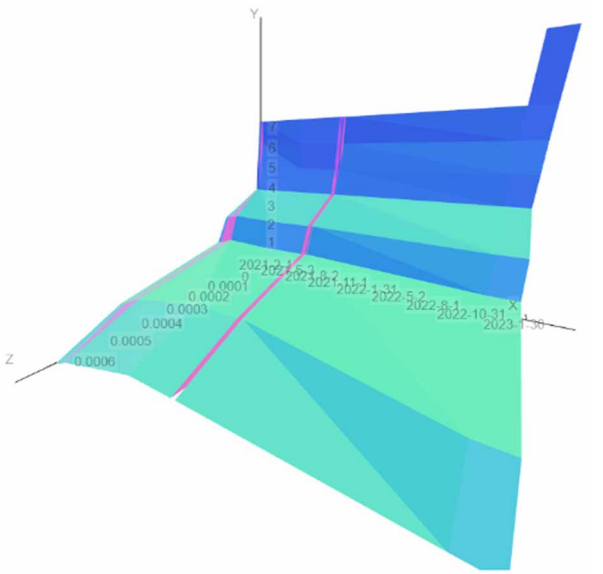

Figure 4: 3D chart.

CSV Upload and Persistent Storage

CSV file has been draggable on the plot since a few years ago. Now there is a new dialog which allows to support many CSV formatted files. Expert users can configure an external server to be used to store data coming from a CSV file in a Postgres TimescaleDB with hypertables and continuous aggregation so that decimated data are continuously stored also as decimated time series; an overview over a long time period is really much faster.

Time Series Correlation

By default time is displayed on X axis and time series values are displayed on the Y axis. A correlation instead deploys values of first time series on X axis and values of second time series on Y axis.

Archiver Status Info

Data is saved on a database by an archiver. Some archivers provide some information about their working status. Time series not in the acquisition state are marked with a line through and a tooltip shows the last date when the acquisition stopped, otherwise a tooltip shows the last date when the acquisition started. This information is shown only if available.

Support for TimescaleDB

Data can be extracted from 2 schemas in PostgresQL TimescaleDB: HDB++ and hypertables with continuous aggregation [4]

Support for HDF5 Data Format

An external library has been used to convert from HDF5 [5] to JSON, than some data treatment are performed, i.e. *NaN* are substituted with *null*, data can be normalized, the order of samples is inverted in respect of time (Fig. 5).

Figure 5: Data extracted from HDF5 format.

Vertical Lines

Some vertical lines may be added as separators using a parameter called *plotLines* which is available only as a GET parameter. (Fig. 5)

Image Exports

Also the export tool was refactored. Inkscape CLI was used to convert from SVG to PNG and PDF (in PDF images are still vectorial and zoomable lossless). Canvas plots are exported too. Charts are created also on server side (as in eGiga) in order to export them directly from a specific URL.

Configurable Max Number of Errors

Time series can include errors too. An error is a null value accompanied by an error message. Errors may be very numerous and require a huge data transfer. To prevent this problem there is a maximum number of errors, the defalt is 1000, it can be adjusted up to an unlimited number of errors.

Time Series Analysis

There was a user request for interpolation in order to compare unevenly spaced time series with evenly spaced time series. This feature was the starting point for developing a set of time series analysis micro services. They are written in PHP or in R (but they could be written in Python or Julia as well), each service is independent of the others, but they all share the same JSON data interface. They are modular in the sense that they can be used in cascade (the output of one service becomes the input of another service) or grouped in a complex formula.

ISBN: 978-3-95450-238-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2023-TUPDP035

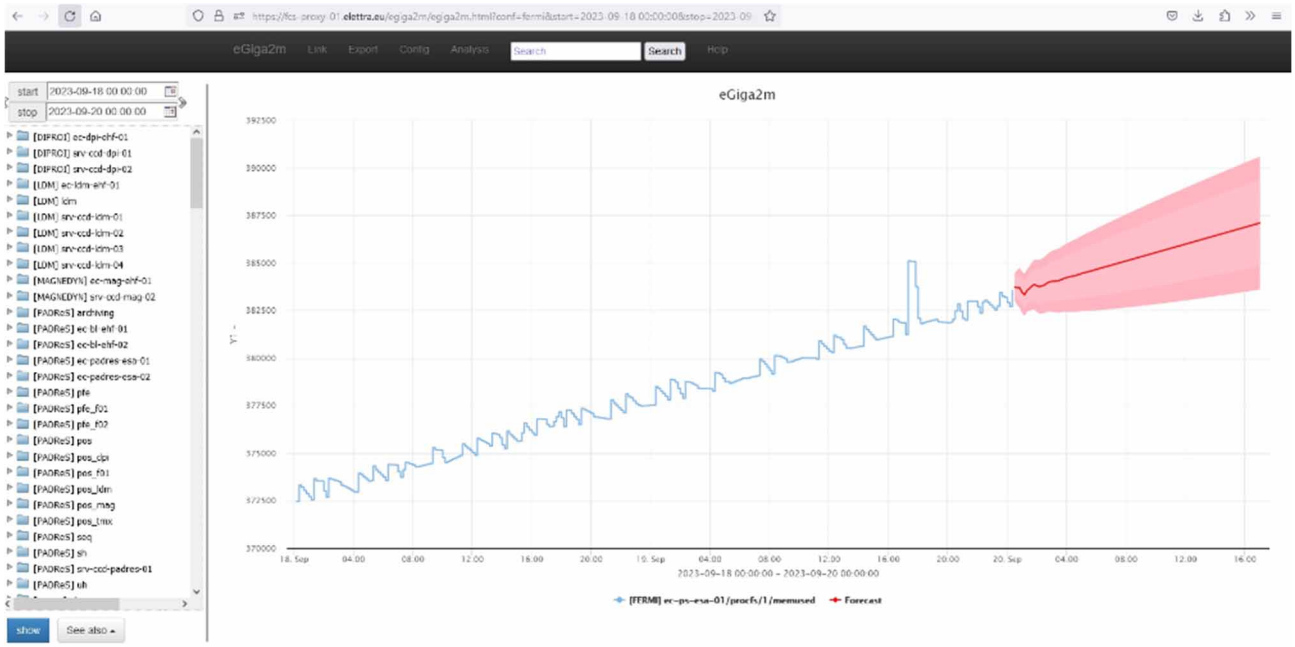

Figure 6: Forecasts are evaluated by an R library implementing an ARIMA algorithm. The red line indicates the most likely forested values, the light pink area an interval of confidence of 80% and the dark pink area 95% confidence.

Each service can be replaced by a new one because it is faster or more robust. Now they are deployed only outside the control system because they are used for post processing of stored data; but they could be useful also inside the control system, a proof of concept was implemented inside a Tango device server, but this would require a second installation also inside the control system from the same versioning repository.

Interpolation

Unevenly spaced time series may require much less space on disk, but it is difficult to compare data with evenly spaced time series and most time series analysis algorithms do not support unevenly spaced time series. Three algorithms for interpolation have been used: ZOH (zero-order hold), linear and spline

ARIMA Forecast

Forecasts are evaluated using the ARIMA package written in R [6]

A fundamental parameter is the interpolation period, by modifying this parameter the result can change significantly. The second parameter is the number of values returned, in most cases it shouldn't be more than 100. The last parameter is the seasonal frequency as the number of samples, the default value of 0 means no seasonality.

The results are proposed to the user but they must be validated by the user because they can be wrong or even misleading. (Fig. 6)

CONCLUSION

eGiga2m in the last 2 years has improved in many aspects. There is still a lot of work to do and it is not excluded a restart from scratch as already happened in 2015.

Although competitors are incredibly strong, eGiga2m is still alive and growing step by step.

REFERENCES

[1] Json-schema, `https:json-schema.org`

[2] Chart.js, `https://www.chartjs.org`

[3] Three.js, `https://threejs.org`

[4] TimescaleDB, `https://www.timescale.com`

[5] HDF5, `https://www.hdfgroup.org/solutions/hdf5`

[6] Forecasting time series, `https://www.rdocumentation.org/packages/forecast/versions/8.16/topics/forecast`