



Figure 4-4. A sample plot from Grafana. This plot is taken from the sample instance that can be accessed from [the project home page](#).

Installation of Grafana is quite simple because it runs entirely on the client side using JavaScript. All you need to run Grafana is a web server that can serve static files such as Twisted or nginx. You will also have to make sure that your users' browsers can access the Open TSDB REST interface either directly or through a proxy. Using a proxy is a good idea if you want to ensure that users see data but can't modify it. If you want to allow users to define new dashboards, you will need to install and run an Elasticsearch instance as well. Grafana is available at <http://grafana.org/>.

Another option for nicer dashboards with Open TSDB is Metrilyx, a package recently open sourced by Ticketmaster. Installing Metrilyx is a bit more involved than installing Grafana because there are additional dependencies (on nginx, Elasticsearch, Mongo and, optionally, Postgres), but there are some benefits such as the use of websockets in order to improve the responsiveness of the display. Keep in mind that while Metrilyx has been in use inside Ticketmaster for some time, it has only recently been released as open source. There may be some teething issues as a result due to the change in environment. Metrilyx is available at <https://github.com/Ticketmaster/metrilyx-2.0>.

## Possible Future Extensions to Open TSDB

The bulk API extension to Open TSDB's REST interface assumes that data can be buffered in memory by the TSD. This violates the design assumptions of Open TSDB by making the TSD keep significant amounts of state information in memory. This has several negative effects, the most notable being that a failure of the TSD will likely cause