

## **Eines TSDB**

### **Introducció**

Time series data can be defined as data points indexed by their temporal order, where the distance between two data points may or may not be equal. If the frequency at which data points are taken is constant (e.g., sampling the data every 10 ms) then the series is called a discrete time data series.

In computer systems, all user data can potentially be represented as time series data, as all stored information has a time component that can provide different metrics in different scenarios. For example, Twitter, Facebook, and LinkedIn have data on the user's registration date, as well as the dates and times at which various actions were performed (tweet or article posted, activity liked, etc.). Other examples are data collected from physical sensors (temperature, humidity, etc) or from performance's computer monitoring tools (RAM use, CPU use, disk space use, network bandwidth use, etc) and/or specific service monitoring tools (apache2 status, mysql status, etc).

Having data arrive at a higher frequency can create challenges, including having to handle a greater number of write requests per second and needing to store all the data. One sensor, with a sampling frequency of 30 requests per second and a payload of 1KB, can generate 86MB of information each day, meaning 100 sensors would create a data load of 8GB per day. Querying and aggregating such a large amount of data to extract useful information is another issue to be considered.

### **Eines**

No todos los programas que tratan datos de tipo "series de tiempo" son iguales: algunos tan solo se encargan de recopilar los datos en sí; otros se encargan de recibir por la red esos datos para recopilarlos y/o guardarlos de forma permanente en una base de datos específica (las llamadas Time Series BD) y otros se encargan de ofrecer esos datos gráficamente en forma de "dashboard" editable con gráficas y colorines.

Históricamente, las primeras TSBDs estaban mayormente basadas en un formato de datos llamado RRD, manejado principalmente por una herramienta central llamada **RRDTool** (<https://oss.oetiker.ch/rrdtool>), la cual integra tanto la base de datos en sí (de tipo "Round Robin", de ahí el nombre) como diversos comandos de terminal para su manejo e incluso también un graficador, pudiéndose utilizar diversos recolectores, como por ejemplo:

**Collectd** (<https://collectd.org>)

**Collectl** (<http://collectl.sourceforge.net>)

**MRTG** (<http://oss.oetiker.ch/mrtg>) -Eina més específica per obtenir dades de tipus SNMP-

**SmokePing** (<http://oss.oetiker.ch/smokeping>) -Eina específica per obtenir dades de latència xarxa-

En el caso de no querer utilizar el graficador que RRDTool incorpora, también se podrían utilizar otros graficadores especializados externos, como:

**GnuPlot** (<http://www.gnuplot.info>)

**Drraw** (<http://web.taranis.org/drraw>)

O bien optar por soluciones "todo-en-uno", las cuales incorporan un recolector de datos y un graficador propio convenientemente configurados con el motor RRDTool para funcionar de forma unificada. Ejemplos son:

**Cacti** (<http://www.cacti.net>) -Su recolector se llama Spine (<https://github.com/Cacti/spine>)-

**Munin** (<http://munin-monitoring.org>)

**Monit** (<https://mmonit.com/monit>)

**Monitorix** (<http://www.monitorix.org>)

Por otro lado, existe una segunda generación de herramientas de monitoraje que van más allá e incorporan todo un ecosistema de plugins que multiplican la versatilidad y funcionalidad de las herramientas RRDTool iniciales. En este sentido se tratan de completos "packs integrales" con multitud de opciones, tanto a nivel de recolección de datos como de visualización (y también de alertas/notificaciones, disparadas al detectar valores fuera de los umbrales previamente definidos para determinados datos). El "abanderado" de esta segunda generación es **Nagios** (<https://www.nagios.org/downloads/nagios-core>) pero hay muchos más:

**Zenoss** (<http://www.zenoss.org>)

**Icinga** (<https://www.icinga.com/download>)

**Zabbix** (<http://www.zabbix.com/download>)

**OpenNMS** (<https://www.opennms.org/en/install>)

**LibreNMS** (<http://www.librenms.org>)

**PandoraFMS** (<https://pandorafms.org/en/features/free-download-monitoring-software>)

**Observium** (<http://www.observium.org/docs>)

**Centreon** (<https://www.centreon.com>)

**Bosun** (<http://bosun.org>). Especializado en alertas

Se pueden consultar más en [https://en.wikipedia.org/wiki/Comparison\\_of\\_network\\_monitoring\\_systems](https://en.wikipedia.org/wiki/Comparison_of_network_monitoring_systems)

No obstante, recientemente ha aparecido una "tercera generación", en la cual se vuelve a la filosofía de "divide y vencerás" (es decir, utilizar una herramienta especializada independiente para cada cosa - recolectar, agregar/almacenar, visualizar- en vez de una macro aplicación "todo-en-una", teniendo entonces que preocuparse, eso sí, de configurarlas para que se comuniquen entre sí correctamente pero ganando a cambio versatilidad y "desacople"). En este sentido, volvemos a encontrarnos con recolectores (que, insistimos, no proporcionan persistencia ya que para ello deberán conectarse a una TSDB)...:

**NetData** (<https://my-netdata.io>)

**Performance Co-Pilot** (<http://pcp.io>) Incluye un gestor de TSDB propio básico y un alertador

**Sensu** (<https://sensuapp.org>)

**Statsd** (<https://github.com/etsy/statsd>)

**Diamond** (<https://github.com/python-diamond/Diamond>)

...con los gestores de base de datos de serie de tiempo propiamente dichos...:

**Graphite** (<https://graphiteapp.org>). Formado por un aglutinador de datos provenientes de diversos recolectores llamado Carbon y un gestor de BD propiamente dicho llamado Whisper, además de un graficador propio llamado Graphite-web.

**InfluxDB** (<https://www.influxdata.com>). Este gestor de BD puede venir acompañado (aunque no tiene por qué) de un recolector de datos desarrollado por el mismo equipo llamado Telegraph y un graficador también propio llamado Chronograf, además, si se desea, de un gestor de alertas llamado Kapacitor. A la suite completa se le suele llamar TICK.

**ElasticSearch** (<https://www.elastic.co>). Este gestor de BD (especializado en la indexación y búsqueda de texto -de ahí el nombre-, por lo que es una herramienta muy interesante para el manejo de logs) puede venir acompañado (aunque no tiene por qué) de un recolector de datos desarrollado por el mismo equipo llamado Beats, un aglutinador/parseador/limpiador de datos provenientes de diferentes orígenes llamado Logstash y un graficador también propio llamado Kibana.

**Prometheus** (<https://prometheus.io>) . Alternativa que incluye aglutinador+almacenador+graficador el cual tiene la particularidad de utilizar un sistema "push" en vez de "pull" para obtener los datos del cliente. Por tanto, no usa por defecto ningún recolector (aunque se puede utilizar uno oficial, [https://github.com/prometheus/node\\_exporter](https://github.com/prometheus/node_exporter))

**Graylog** (<https://www.graylog.org>). Alternativa completa (recolector+aglutinador+almacenador+graficador)

**OpenTSDB** (<http://opentsdb.net>). Necesita HBase para funcionar  
**KairosDB** (<https://kairosdb.github.io>). Necesita Cassandra para funcionar  
**Cyanite** (<http://cyanite.io>). Necesita Cassandra para funcionar  
**Gnocchi** (<https://gnocchi.xyz>). Parte de OpenStack

...y con graficadores que muestran al usuario -via navegador- los datos allí almacenados:

**Grafana** (<http://grafana.org>)  
**Graphsky** (<http://www.graphsky.org>)  
**Cubism.js** (<http://square.github.io/cubism>)

Se pueden consultar más graficadores en <https://github.com/obazoud/awesome-dashboard>

Y también (hay más en general) <https://github.com/graphite-project/graphite-web/blob/master/docs/tools.rst>

También existen herramientas específicas para generar y enviar alertas a partir de determinadas condiciones detectadas en los datos recopilados, como por ejemplo:

**Seyren** (<https://github.com/scobal/seynen>)  
**Cabot** (<https://github.com/arachnys/cabot>)

Incluso existen servicios SaaS (es decir, servicios online) que ofrecen el almacenamiento, gestión de alertas y visualización de los datos que hayamos obtenido mediante nuestros recolectores, de forma que no tengamos que preocuparnos de la gestión de dichos datos. Ejemplos son:

<https://www.hostedgraphite.com>  
<https://cloud.influxdata.com>  
<https://www.elastic.co/cloud>  
<http://pnda.io>

Per saber más sobre las similitudes y diferencias técnicas entre las herramientas anteriores (y muchas otras) recomiendo la visita a <http://db-engines.com>