Grafana and MySQL Benefits and Challenges

INGRAM

About me

Philip Wernersbach

Software Engineer

Ingram Content Group

https://github.com/philip-wernersbach

https://www.linkedin.com/in/pwernersbach

- I work in Ingram Content Group’s Automated Print On Demand division
- We have an automated process in which publishers (independent or corporate) request books via a website, and we automatically print, bind, and ship those books to them
- This process involves lots of hardware devices and software components

The Problem

The Problem

"How do we aggregate and track metrics from our hardware and software sources, and display those data points in a graph format to the end user?"

→Grafana!

Which data store should we use with Grafana?

- Out of the box, Grafana supports Elasticsearch, Graphite, InfluxDB, KairosDB, OpenTSDB

Which data store should we use with Grafana?

- *We compared the options and tried InfluxDB*
- *There were several sticking points with InfluxDB, both technical and organizational, that caused us to rule it out*

Which data store should we use with Grafana?

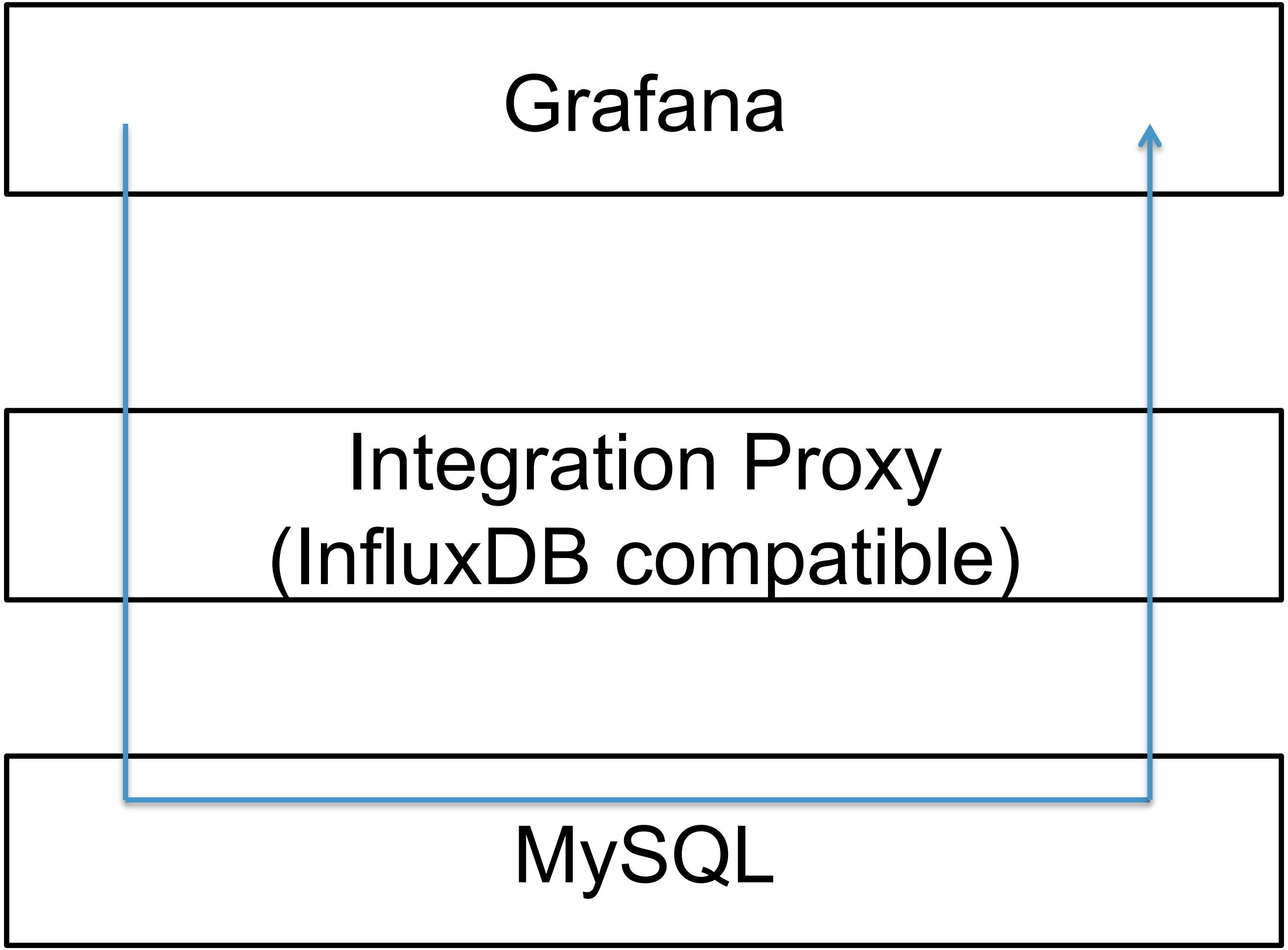
- ***We already have a MySQL cluster deployed, System Administrators and Operations know how to manage it***
- ***Decided to go with MySQL as a data store for Grafana***

The Solution: Ingram Content's Grafana-MySQL Integration

The Integration

- Written in Nim
- Emulates an InfluxDB server
- Connects to an existing MySQL server
- Protocol compatible with InfluxDB 0.9.3
- Acts as a proxy that converts the InfluxDB protocol to the MySQL protocol and vice-versa

The Integration



Grafana

Integration Proxy (InfluxDB compatible)

MySQL

The Challenges!

The Challenges!

Database Engine Comparison

InfluxDB

- Time Series database
- “time” is a special value, has special meaning
- Applies special logic to “time”

SMART!

MySQL

- Generic relational database
- “time” is a generic data value
- Doesn’t apply special logic

DUMB!

The Challenges!

Query Comparison

InfluxDB

“SELECT count(bar1) FROM foo WHERE bar1 > 0 AND bar2 > 0 AND time > now() - 7d GROUP BY time(1h)”

The Challenges!

Query Comparison

MySQL

“SELECT time, count(bar1) FROM foo WHERE bar1 > 0 AND bar2 > 0 AND time > NOW(6) - INTERVAL 7 DAY GROUP BY YEAR(time), MONTH(time), DAY(time), HOUR(time) ORDER BY time ASC”

InfluxQL != SQL

- InfluxQL is SQL-like, but different enough that it can’t be passed through to MySQL

The Challenges!

InfluxQL != SQL

- “time” is SELECT’d automatically implicitly in InfluxDB
  - “SELECT bar FROM foo” → “SELECT time, bar FROM FOO”
- GROUP’ing on “time” is smart in InfluxDB, and dumb in MySQL
  - See slides 16 and 17
- “time” in epoch format with millisecond precision is a float in MySQL
  - “FROM_UNIXTIME(1444667802.145)”

The Challenges!

InfluxQL != SQL

- InfluxDB series are roughly equivalent to MySQL tables, but the management syntax is sometimes different
  - “DROP SERIES FROM foo” → “DELETE FROM foo”
- InfluxDB allows quoted identifiers, MySQL is more restrictive
  - “SELECT * FROM foo WHERE bar > ‘0’” → Silently returns garbage in MySQL
  - Grafana quotes identifiers aggressively, so we had to patch it

The Challenges!

InfluxQL != SQL

There's probably more InfluxQL-SQL incompatibilities, but these are the ones that we've run into with Grafana at Ingram Content Group.

The Benefits!

The Benefits!

- Allows us to leverage System Administrator’s and Operation’s existing knowledge of MySQL
- Allows us to process large amounts of data
  - InfluxDB had issues when we tried to insert about a million points at a time, MySQL handles this with no problem

The Benefits!

- We can use full, regular SQL for our Grafana queries in addition to InfluxQL
  - Think JOINs, etc.
- We're doing massive batch inserts, and MySQL's query cache makes Grafana queries significantly faster
  - This is obviously use case specific, but shows how you can tune MySQL to your specific use case