
CERN

InfluxDB for Monitoring Data

Luca Magnoni, for the MONIT team

Outline

- The Monitoring use case
- InfluxDB Workflow
  - Data preparation
  - How we write
  - Reading from Grafana
- Lessons Learned

The Monitoring use case

MONIT / DBOD InfluxDB story

- ~ early 2017 we were investigating time series storage for Collectd and WLCG metrics
  - with automatic aggregation
  - and good Grafana support
- InfluxDB was growing as reference TSDB
- At that time pilot @ CERN IT DBOD
- The good technology at the good moment

MONIT / InfluxDB data flow

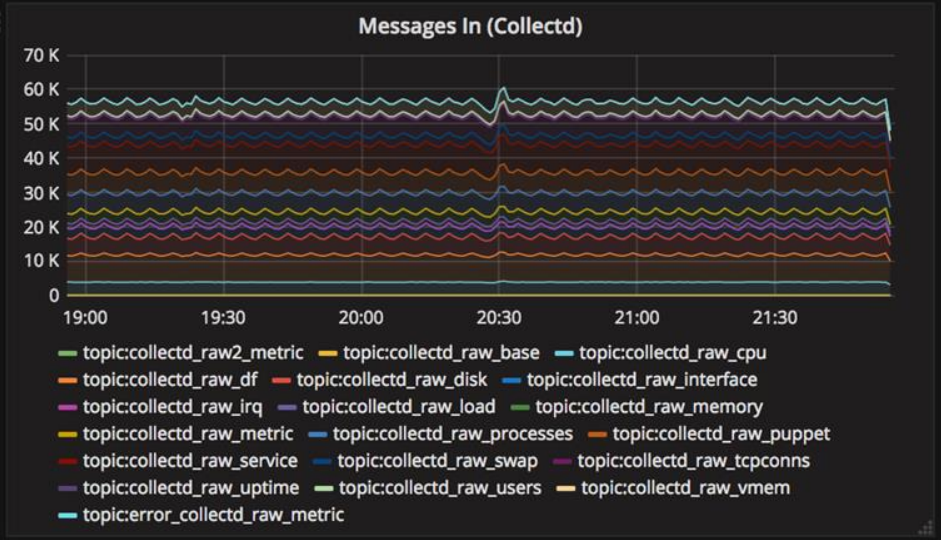
- Collectd and WLCG metrics
- Current flow to InfluxDB:
  - ~ 65 k documents per second
  - 1.6 TB / day
- Increases with new data sources and new collectd plugin (e.g. puppet)

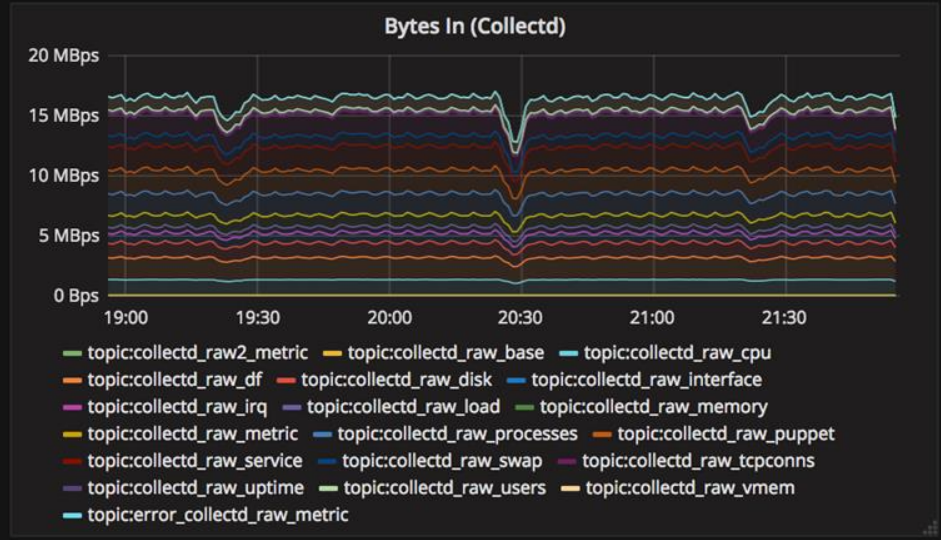
Kafka

Zoom Out | Last 3 hours

Data Range | one_week | Environment | production

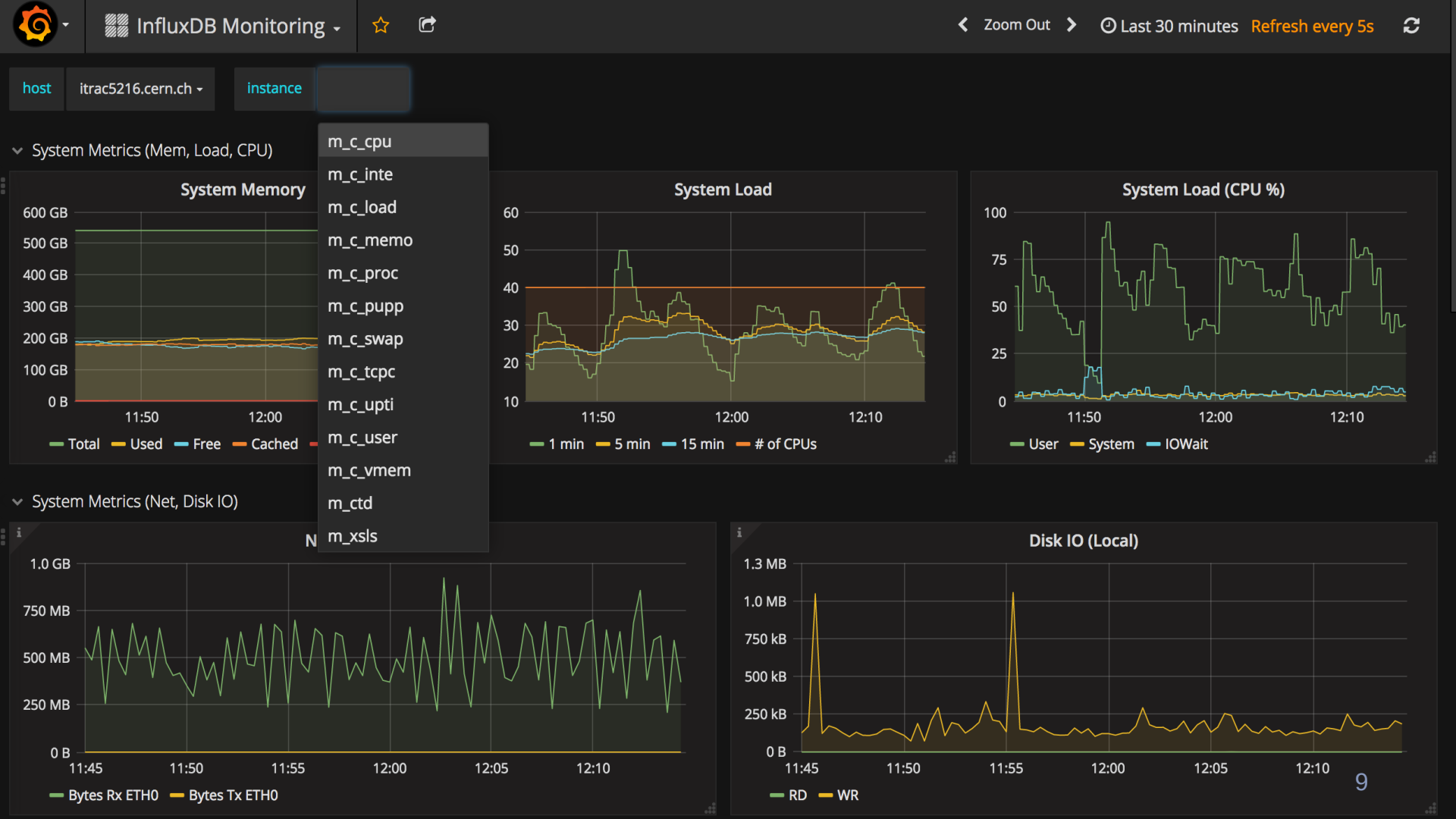
- Overview
- Cluster
- Topics (Raw Metrics)
- Topics (Raw Logs)
- Topics (Raw Collectd)





InfluxDB Setup / Instances

- 20 production instances (7 dev)
  - initially started with few *big* ones
    - with several databases/measurements each
    - difficult to isolate/debug problems
  - decided to split into many ~*small* ones
    - e.g. collectd: one per plugin, several per services
    - better load distribution and control
    - It scales (up to the resources behind… :))
    - best fit for DBOB model
- Currently using both 1.1 and 1.3 (with TSI)


InfluxDB Monitoring
Zoom Out
Last 30 minutes
Refresh every 5s
host
itrac5216.cern.ch
instance
m_c_cpu
m_c_inte
m_c_load
m_c_memo
m_c_proc
m_c_pupp
m_c_swap
m_c_tcpc
m_c_upti
m_c_user
m_c_vmem
m_ctd
m_xsls
System Metrics (Mem, Load, CPU)
System Memory
600 GB
500 GB
400 GB
300 GB
200 GB
100 GB
0 B
11:50
12:00
Total
Used
Free
Cached
System Load
60
50
40
30
20
10
11:50
12:00
12:10
1 min
5 min
15 min
# of CPUs
System Load (CPU %)
100
75
50
25
0
11:50
12:00
12:10
User
System
IOWait
System Metrics (Net, Disk IO)
1.0 GB
750 MB
500 MB
250 MB
0 B
11:45
11:50
11:55
12:00
12:05
12:10
Bytes Rx ETH0
Bytes Tx ETH0
Disk IO (Local)
1.3 MB
1.0 MB
750 kB
500 kB
250 kB
0 B
11:45
11:50
11:55
12:00
12:05
12:10
RD
WR

InfluxDB Setup / RP

- Using Retention Policies (RP) to manage raw and downsampled data.
  - one_week : raw (1 minute sampling)
  - one_month : 5 minute aggregation
  - five_years : 1 hour aggregation

InfluxDB Setup / CQs

- Continuous Queries (CQ)
  - We're using CQs to aggregate data over time
    - 5min, 1hour (but also 1 day, 1w, 1M in some cases)
  - With backreferencing
    - abstracts the aggregation from the data format
    - very useful for the Collectd use case
      - 1 generic query for all data types / measurements
  - Chaining CQs to reduce IO load

Generic CQs (e.g. 1 for all services)

CREATE CONTINUOUS QUERY "60min_agg" ON monit_production_collectd_service

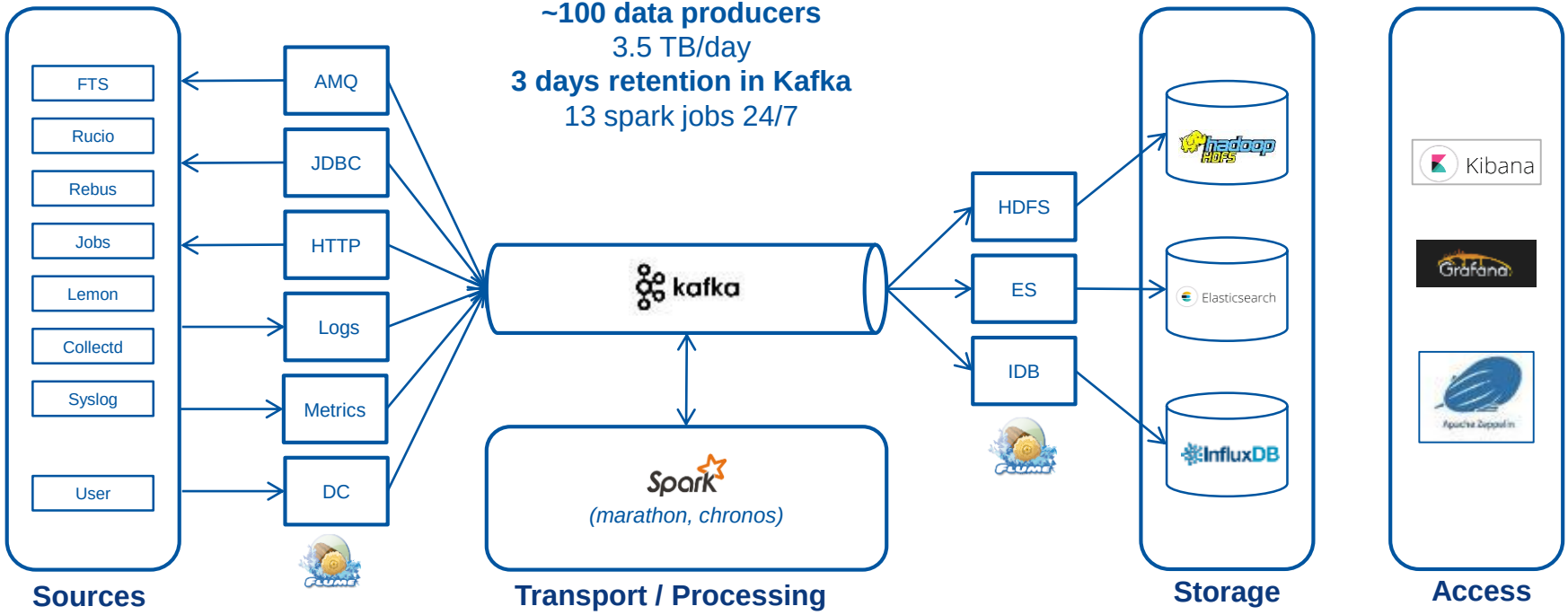
BEGIN SELECT mean(mean_value) AS mean_value, sum(sum_value) AS sum_value, count(count_value) AS count_value, max(max_value) AS max_value, min(min_value) AS min_value

INTO ***monit_production_collectd_service.five_years.:MEASUREMENT*** *FROM monit_production_collectd_service.one_month./.*/*

GROUP BY time(1h), ***** *END*

Workflow

MONIT Architecture: quick recap



Data Preparation

”One does not simply write data to InfluxDB…”

Data preparation / analysis

- Not all data can fit
- Carefully identify TAGs and FIELDs
  - Use case specific
  - They define searches and visualizations capability
- Check TAGs cardinality (twice…)
  - We’re living with several millions cardinality
    - memory grows non-linearly with cardinality…

Data preparation / transformation

- Extract TAGs, FIELDs, TIME from JSON
- Validate and Transform, if needed
- Prepare data in InfluxDB format
- Write via HTTP API

e.g. CPU Collectd data

{ metadata: {
submitter_environment: qa
toplevel_hostgroup: monitoring TAGS
submitter_hostgroup: monitoring/kafka
event_timestamp: 1505744792000 TIME
}

data: {
host: monit-kafka
plugin: cpu
plugin_instance: TAGS
type: percent
type_instance: idle
value: 0.021 VALUE
} }

cpu_percent MEASUREMENT
host=monitkafka,toplelvel_hostgroup=monitoring,type=cpu,submitter_hostgroup=monitoring/kafka,plugin=cpu,plugin_insent, tance=UNKNOWN,type=percent,type_instance=idle
mean_value= 0.021,max_value=0.021,min_value=0.021,sum_value=0.021
1505744792000

How we write data

Flume / InfluxDB sinks

- Several (7) Flume agents writing to InfluxDB
  - m2.large VMs
- Single agent:
  - Reads from all Kafka topics
    - starts multiple sources per topic
  - Writes to multiple InfluxDB instances
- Scale horizontally very easily

Flume HTTP sink

- POST requests to the /write endpoint
  - with specific data content
- We use Flume HTTP sink
  - patched to use HTTPS
  - available here [ADD LINK]
- Interceptor to parse & transform data
- Batches of 5k metrics (recommended)
- We've also a sampled flow for QA/dev
  - e.g. writes 10% of docs, configurable

Flume / InfluxDB Interceptor

[...]
type=ch.cern.monit.flume.interceptors.InfluxDBInterceptor$Builder

tags=host,plugin,plugin_instance,type,type_instance,toplevel_hostgroup,producer,type_prefix,submitter_environment,submitter_hostgroup,value_instance

fields=mean_value,sum_value,max_value,min_value
measurementField=measurement

timeField=event_timestamp

[...]

Grafana & InfluxDB

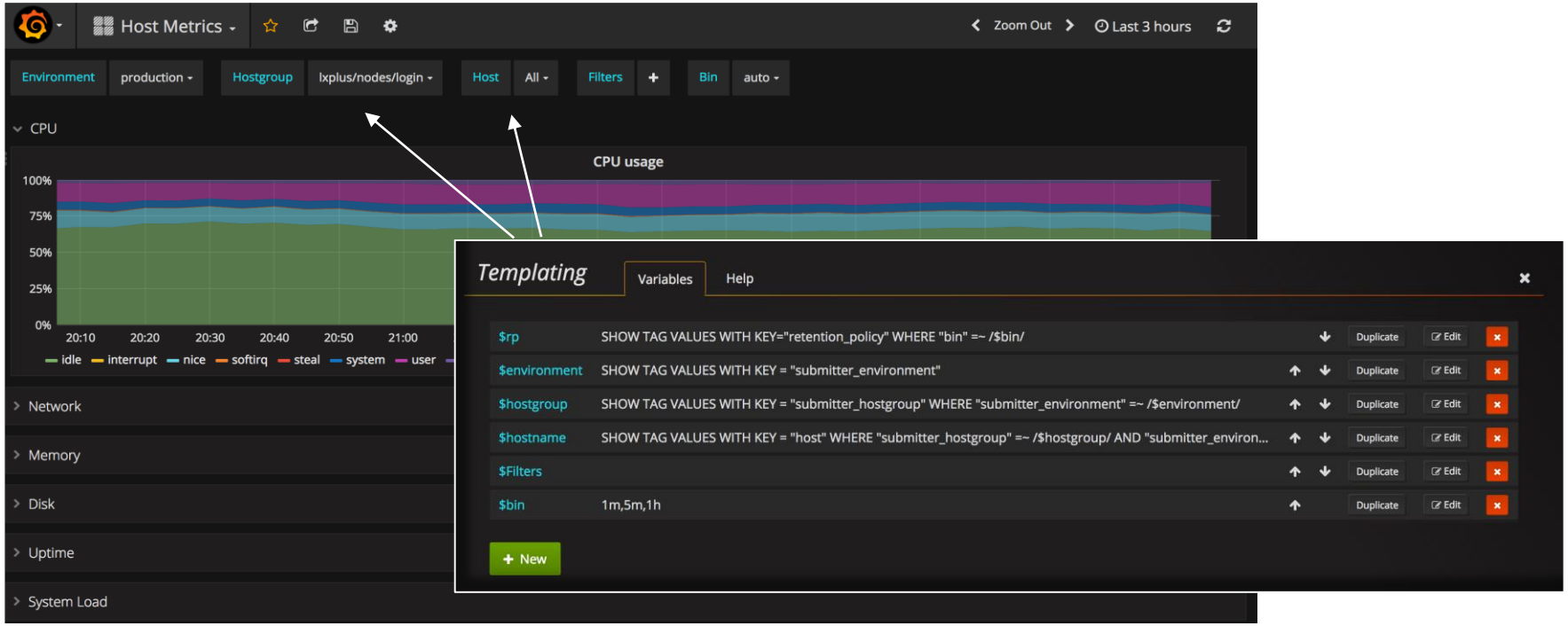
Grafana / InfluxDB integration

- Grafana comes with built-in InfluxDB support
  - Template / Ad-hoc filters / Autocompletion
  - Advanced SQL-like query syntax
  - Alarms
- Focus next on some of the main features

(Chained) Template Variables

- *Templates* are used to build dropdown filters
- *Query variables* can be populated querying InfluxDB dynamically
- Template relations can be defined so that values are updated when other values change
  - e.g. select hosts from selected hostgroups

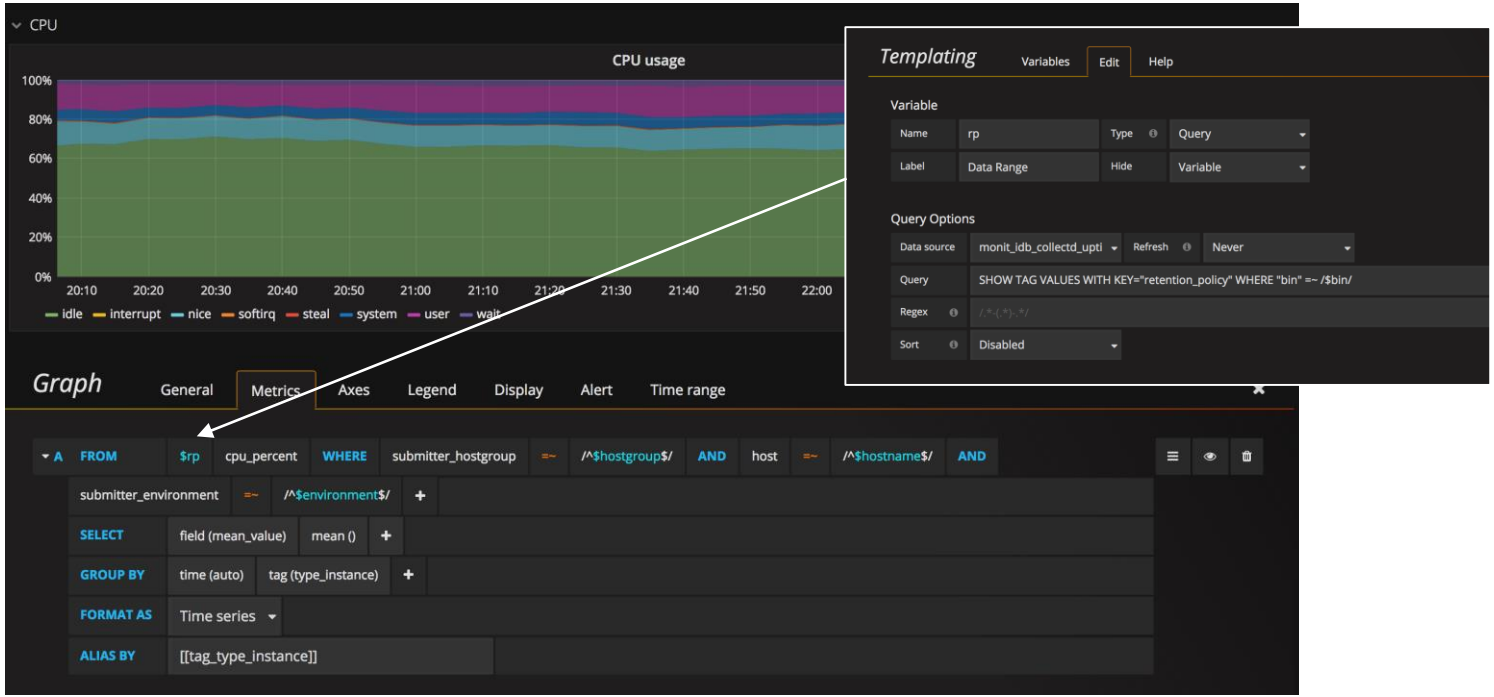
(Chained) Template Variables



Dynamic RP selection

- Same dashboard can show data from multiple aggregation bins (e.g. retention policies)
- Retention Policy can be parametrized as user-selected variable
- With some more tricks, RP selection can be linked directly to the Binning interval

(Hidden) Dynamic RP selection



Data Exploration

- Possibility to build a generic Table view to explore raw data
- Useful to discover metrics tags and field values
  - ad-hoc filters can be added to narrow selection
- e.g. Collectd browser to inspect plugin data types

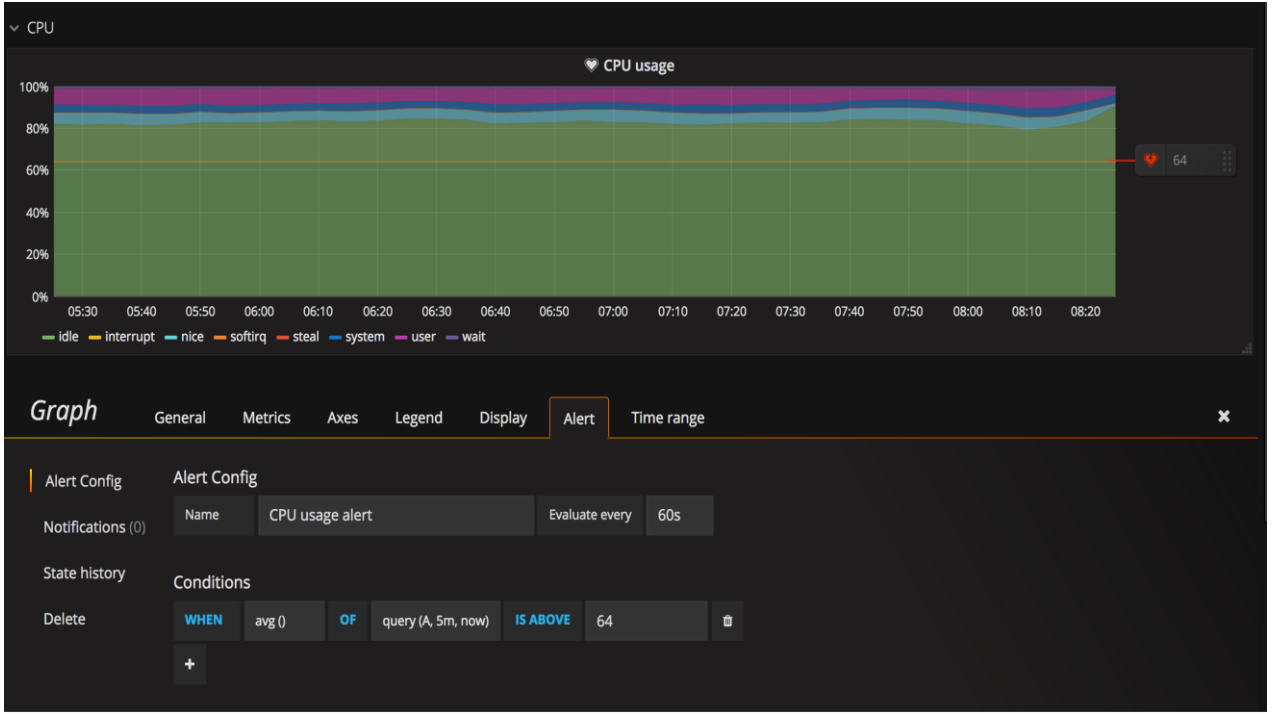
Grafana *fill(null)* on new plots

- When query grouping time is smaller than sampling time, InfluxDB allow several fill() functions to be used to handle missing bins (i.e. *none, null, 0 previous, linear*)
  - Grafana set a fine-grained granularity by default
  - and uses *fill(null),* unfortunately
  - witch may lead to confusing (empty) plots…
- Solutions:
  - Set a low limit to the query grouping time so that is >= sampling
  - Or choose a different fill strategy e.g. fill(none)
- #7253

Grafana Alarms

- Users can create a threshold-based rule on a plot via the Grafana UI
- Grafana server queries InfluxDB to evaluate the rule and trigger a notification in case of issue

Grafana Alarms



Lessons Learned

Deletion is hard

- Careful with DELETE
  - Slow and heavy
  - Data actually removed by shard, may lead to surprises (e.g. deletion of 1hr removes 2 days)
  - Do not consider RP
- Prefer DROP SHARD or MEASUREMENT
- DROP DATABASE is the fastest…

RP and CQ

- Retention Policies
  - Chose RP names wisely
    - Duration can be changed, not names
- Continuous Queries
  - CQ execution serialized per instance :(
  - Lack of more time literals (Week, Month) #2071
  - Resample (e.g. CQ continuously evaluating long past intervals to catch late arriving events) with care
    - We've experienced some issue with 1.3 using CQ Advanced Syntax

Some useful tricks

- 2 colliding data points, same time, but different attribute that cannot be tag (e.g. ID) ?
  - Add an artificial random part to time
  - Hash those attribute and add the hash as time, for a reproducible insertion
- Poor’s man ’SHOW CARDINALITY’

Whish List

- Intelligent rollups/queries #7198
- SHOW CARDINALITY #7195
- Log access on DBOB interface

On Performance

- 70/100 k pps
- Memory footprint is critical
  - 1.3 with TSI improved, but we don’t have
- Instance Isolation

Conclusions

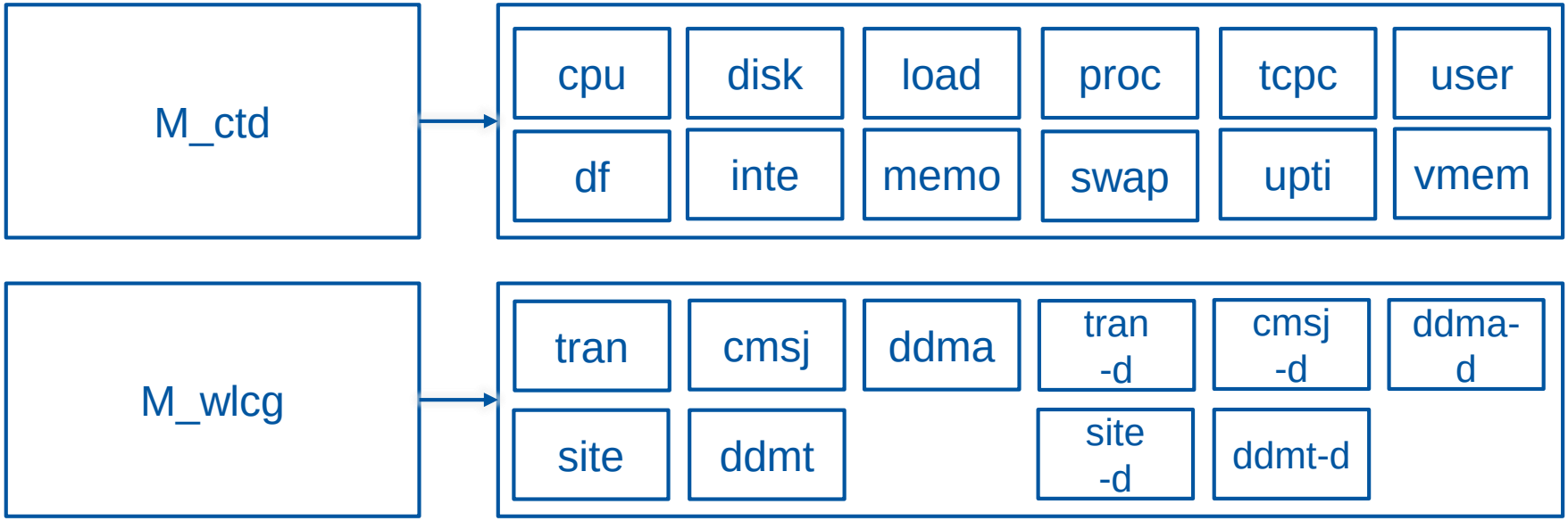
- InfluxDB now used as backend for CERN Data Centre and WLCG monitoring dashboards
- Very positive feedback for DBOD service
- Important to have prompt support and expertise
- Resources


CERN

MONIT InfluxDB setup

- Initially a couple of instances, decided to go for several instances
  - Probably a bigger split will be done
- Whenever possible different instances for production and development
  - Different resources

MONIT InfluxDB setup



Starting point

Current point