

Rethinking monitoring with Prometheus



Martín Ferrari

Based on a previous talk prepared with
Štefan Šafár - @som_zlo

Who is Prometheus?

A dude who stole fire from Mt. Olympus
and gave it to humanity

<http://prometheus.io/>

What is Prometheus?

NOT Nagios

What is Prometheus?

Only good/bad/worse states

Does not really scale

No understanding of underlying problems

What is Prometheus?

Systems like NewRelic are the new cool stuff™

Automatically instrumented services!

A lot of data!

Not easy to do something useful with it

Cloud-based, you lose control of your data

What is instrumentation?



What does Prometheus do?

It collects and process data:

- From everywhere
- A lot of data
- Very efficiently



Encourages instrumentation

Has really nice graphs™

Intermission: Go packaging

A few challenges to get Prometheus into Debian

Go is a new language, especially in Debian - most dependencies were not packaged

Small group, best practices still in flux

Come help the team!

Prometheus architecture

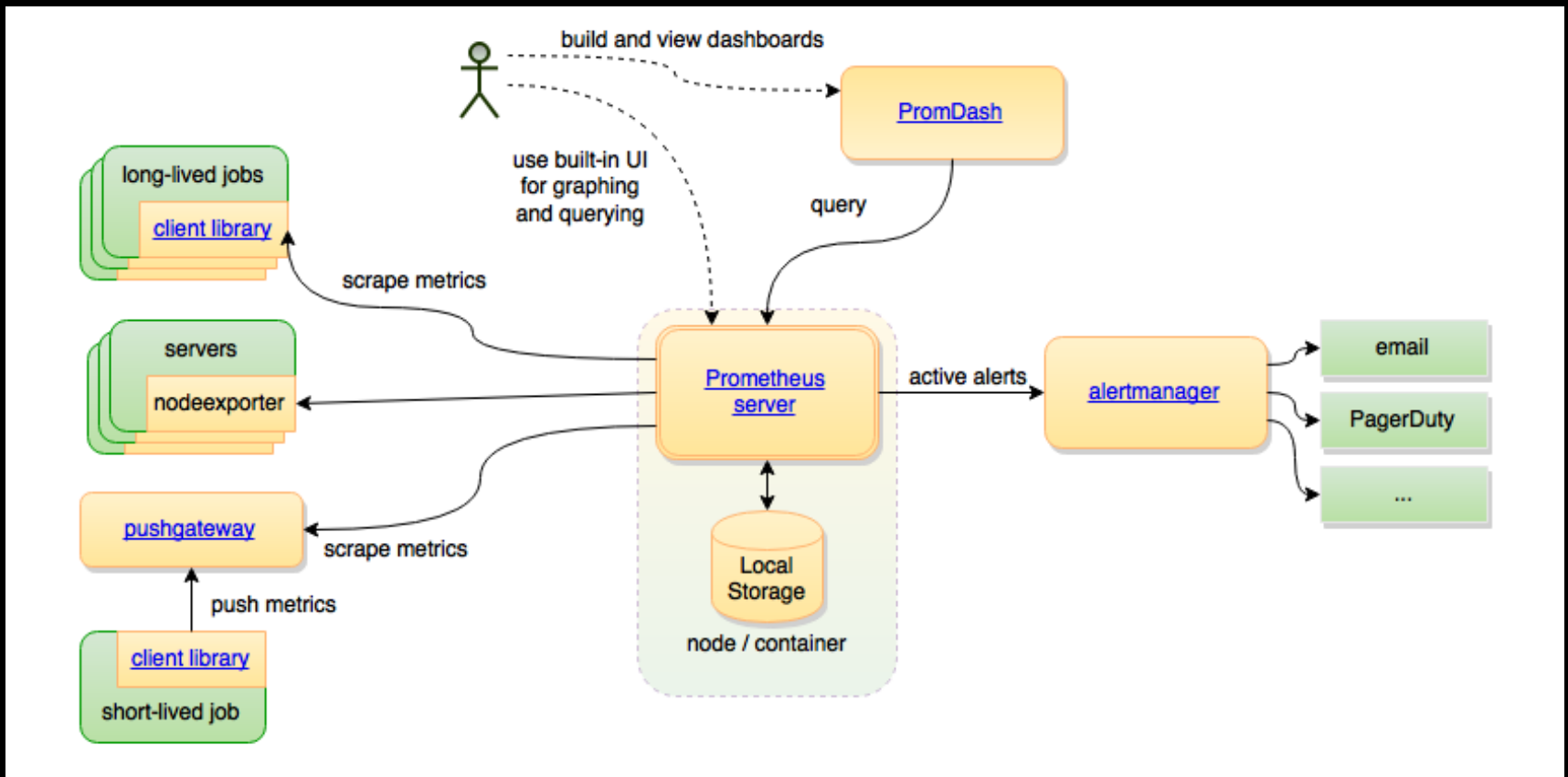


Image based on diagram at <http://prometheus.io/docs/introduction/overview/>

Data ingestion: protocol

Simple protocol:

- HTTP transport
- Plain text content (protobuf optional)
- Pull-based collection

Data ingestion: implementation

Very efficient implementation:

- Hundreds of 1000s of metrics/s per server
- Disk-efficient storage
- Tunable retention
- Sane defaults!
Both in Debian and upstream

Data ingestion: sources (I)

node_exporter

- Network, disk, cpu, ram, etc
- Add your custom metrics (text file)

push_gateway

- Cron jobs, short-lived services
- Data that has to be pushed

Data ingestion: exporters

Official

- Node/system metrics
- AWS CloudWatch
- Collectd
- Consul
- Graphite
- HAProxy
- Hystrix metrics
- JMX
- Mesos tasks
- MySQL server
- StatsD bridge

Unofficial

- CouchDB
- Django
- Memcached
- Meteor JS framework
- Minecraft module
- MongoDB
- Munin
- New Relic
- RabbitMQ
- Redis
- Rsyslog
- ...

Data ingestion: instrumentation

Language-specific libraries for instrumentation

Go, Java, Scala, Python, Ruby

Bash, Haskell, Node.js, .NET / C#

Already instrumented: etcd, kubernetes, ...

Or roll your own! (it's easy)



Data processing

Powerful query language. Use it to:

- Browse data: interactive console
- Synthesise metrics from complex calculations:
- Create cute graphs
- Wake you up at 3am

Query language: example

Source data:

<code>node_cpu{cpu="cpu0",instance="here.cz:9000",mode="idle"}</code>	16312937.7
<code>node_cpu{cpu="cpu0",instance="here.cz:9000",mode="iowait"}</code>	182080.66
<code>node_cpu{cpu="cpu0",instance="here.cz:9000",mode="system"}</code>	282463.23
<code>node_cpu{cpu="cpu0",instance="here.cz:9000",mode="user"}</code>	552748.8
<code>node_cpu{cpu="cpu0",instance="there.org:9100",mode="idle"}</code>	17914450.35
<code>node_cpu{cpu="cpu0",instance="there.org:9100",mode="iowait"}</code>	81386.28
<code>node_cpu{cpu="cpu0",instance="there.org:9100",mode="system"}</code>	47401.76
<code>node_cpu{cpu="cpu0",instance="there.org:9100",mode="user"}</code>	124549.65
<code>node_cpu{cpu="cpu1",instance="there.org:9100",mode="idle"}</code>	18005086.74
<code>node_cpu{cpu="cpu1",instance="there.org:9100",mode="iowait"}</code>	12934.74
<code>node_cpu{cpu="cpu1",instance="there.org:9100",mode="system"}</code>	44634.8
<code>node_cpu{cpu="cpu1",instance="there.org:9100",mode="user"}</code>	86765.05

Query language: example

```
sum by (instance, mode) (rate(node_cpu[1m]))
```

```
{instance="here.cz:9000",mode="idle"}           0.89222
{instance="here.cz:9000",mode="iowait"}         0.00911
{instance="here.cz:9000",mode="system"}         0.03444
{instance="here.cz:9000",mode="user"}           0.05799

{instance="there.org:9100",mode="idle"}         1.8464
{instance="there.org:9100",mode="iowait"}       0.0217
{instance="there.org:9100",mode="system"}       0.0211
{instance="there.org:9100",mode="user"}         0.107
```

Query language: example



|rate(node_network_transmit_bytes{host =~ "^..fwdb\d+\\.\\.\\.","device="eth1"})[5m]

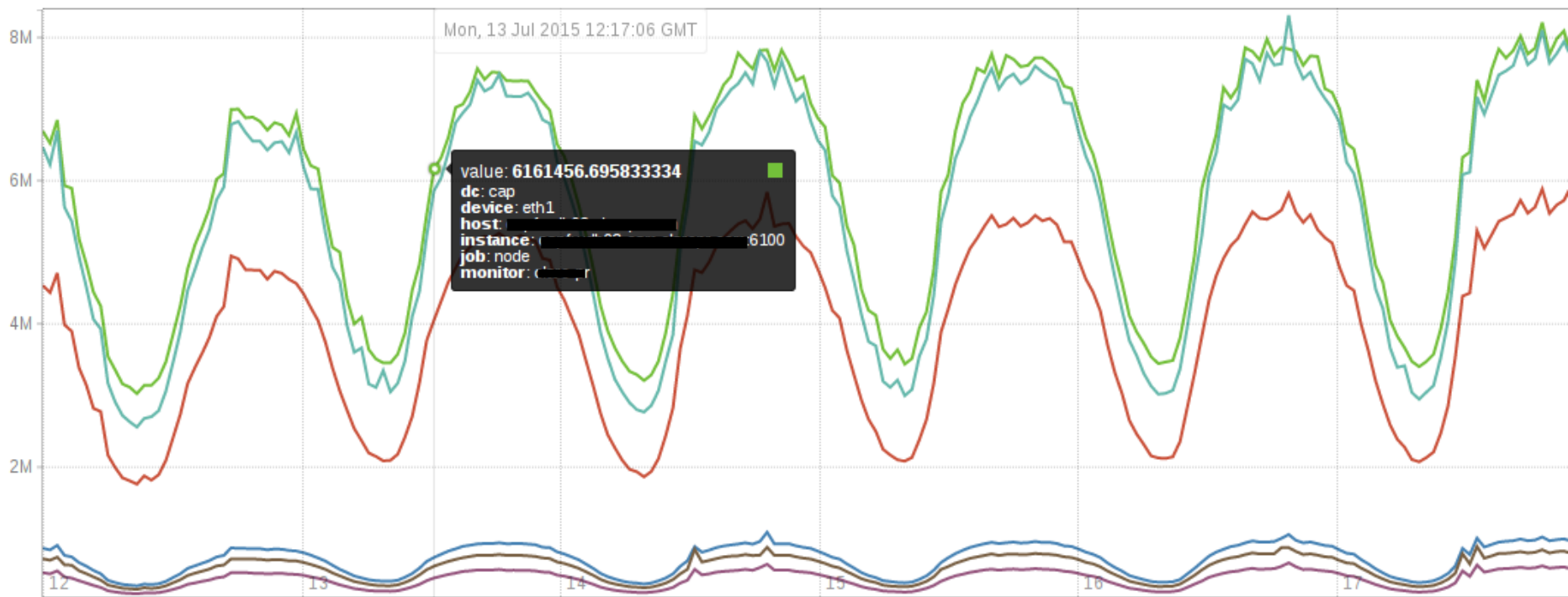
Execute

- insert metric at cursor -

Graph

Console

- 1w +
◀ 2015-07-19 00:00 ▶
Res. (s)
 stacked



- {dc="san",device="eth1",host="[REDACTED]",instance="[REDACTED]:6100",job="node",monitor="[REDACTED]"} (1)
- {dc="san",device="eth1",host="[REDACTED]",instance="[REDACTED]:6100",job="node",monitor="[REDACTED]"} (1)
- {dc="san",device="eth1",host="[REDACTED]",instance="[REDACTED]:6100",job="node",monitor="[REDACTED]"} (1)
- {dc="cap",device="eth1",host="[REDACTED]",instance="[REDACTED]:6100",job="node",monitor="[REDACTED]"} (1)
- {dc="cap",device="eth1",host="[REDACTED]",instance="[REDACTED]:6100",job="node",monitor="[REDACTED]"} (1)

Consoles

Templates rendered and served by prometheus

Convenient for version control

Can include graphs, metric values, alerts

Customise your dashboard!

Promdash

Rails app

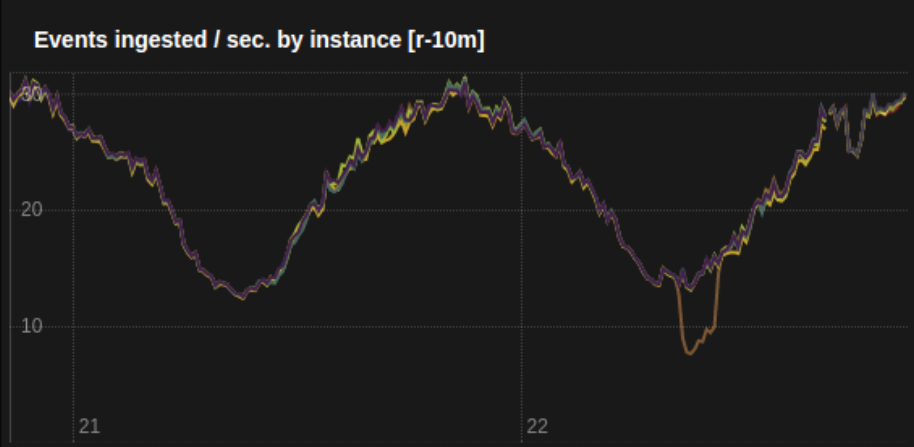
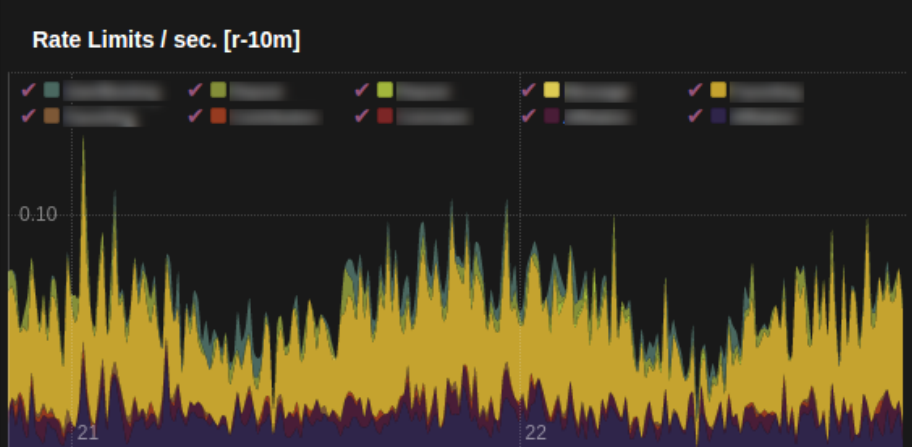
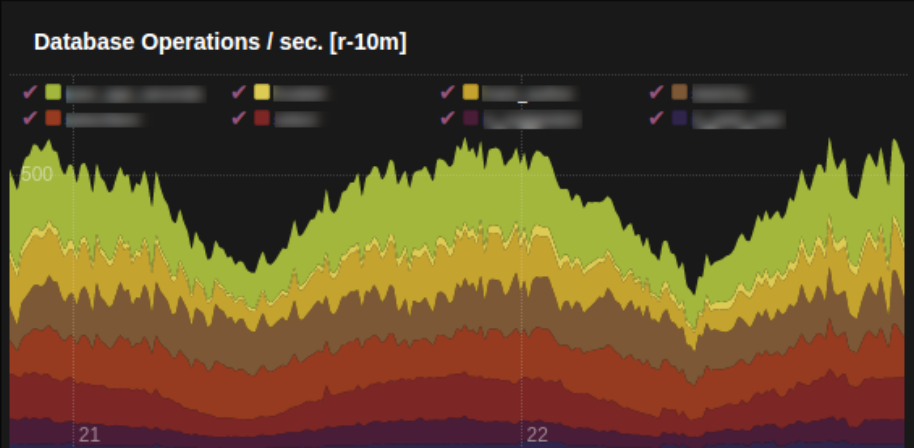
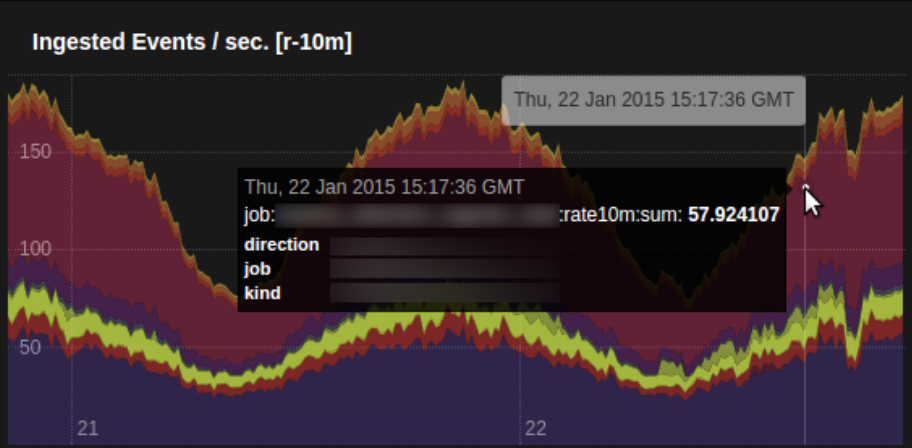
Browser-based building of consoles

Independent of prometheus server

Shiny!!1!

Test Event Processor

2d + << Until >> ↻ never ▾ ⚙ 🗄 Save Changes



Alerting: simple

```
ALERT InstanceDown
  IF up == 0
  FOR 5m
  WITH { severity="page" }
  SUMMARY "Instance {{$labels.instance}} down"
  DESCRIPTION "{{$labels.instance}} of job
                {{$labels.job}} has been down
                for more than 5 minutes."
```

Alerting: more complex

```
ALERT ApiHighRequestLatency
  IF api_http_request_latencies_ms{quantile="0.5"}
    > 1000
  FOR 1m
  SUMMARY "High request latency on {{$labels.instance}}"
  DESCRIPTION "{{$labels.instance}} has a median request
    latency above 1s (current value:
    {{$value}}) "
```




Martín Ferrari
<http://tincho.org>

Bonus: Push vs Pull

centrally coordinated

easy reconfiguration / sharding / adding servers

parallel / redundant servers are trivial

developers can run their own instances

Bonus: demo queries

```
sum by (instance) (  
    rate(http_response_size_bytes_sum{job="node"}[1m])  
)
```

```
http_requests_total{code=~"^[45]..$"}  
rate(process_cpu_seconds_total[1m])
```

```
sum by (mode) (  
    rate(node_cpu{instance="brie.tincho.org:9100", mode =~  
    "^(idle|user|system|iowait)"}[1h])  
) or sum (  
    rate(node_cpu{instance="brie.tincho.org:9100", mode !~  
    "^(idle|user|system|iowait)"}[1h])  
)
```