Energy efficiency in OpenStack clouds

François Rossigneux
francois.rossigneux@inria.fr

Inria

January 28, 2013 - Université du Luxembourg

Summary

Context

Telemetry architecture

Scheduling / sleep modes (future works)

Summary

Context

Telemetry architecture

Scheduling / sleep modes (future works)

Context

XLcloud:

- HPC-as-a-Service (based on OpenStack)
- Funded by the "Fonds national pour la Société Numérique"
- Three-year long collaborative project
- Open source license

Some features:

- GPU virtualization
- Green scheduling
- Power consumption based billing

xlcloud
Powering Computation in the Cloud

Context

Consortium:

Bull

ServiWare
A Bull Group Company

MINES TELECOM
INSTITUT
Mines-Télécom

SILKAN
Embedded simulation for a safer world

EISTI
Ecole Internationale des Sciences du Traitement de l'Information

ateme

inria

cea list

OW2

AMG.lab
A Bull Group Company

Context

Our team is working on energy topics:

- Telemetry (taking measurements)
- Scheduling (placing virtual machines)
- Turning off unused machines (sleep modes)

Summary

Context

Telemetry architecture

Scheduling / sleep modes (future works)

Telemetry architecture
OpenStack overview

OpenStack main components:

- Compute (Nova)
- Object Storage (Swift)
- Block Storage (Cinder)
- Networking (Quantum)
- Identity (Keystone)
- Dashboard (Horizon)

Recently added:

- Metering / billing (Ceilometer)

Incubation:

- Energy (Kwapi)

Telemetry architecture
OpenStack overview

OpenStack main components:

- Compute (**Nova**)
- Object Storage (Swift)
- Block Storage (Cinder)
- Networking (Quantum)
- Identity (**Keystone**)
- Dashboard (Horizon)

Recently added:

- Metering / billing (**Ceilometer**)

Incubation:

- Energy (**Kwapi**)

Telemetry architecture
OpenStack overview

OpenStack main components:

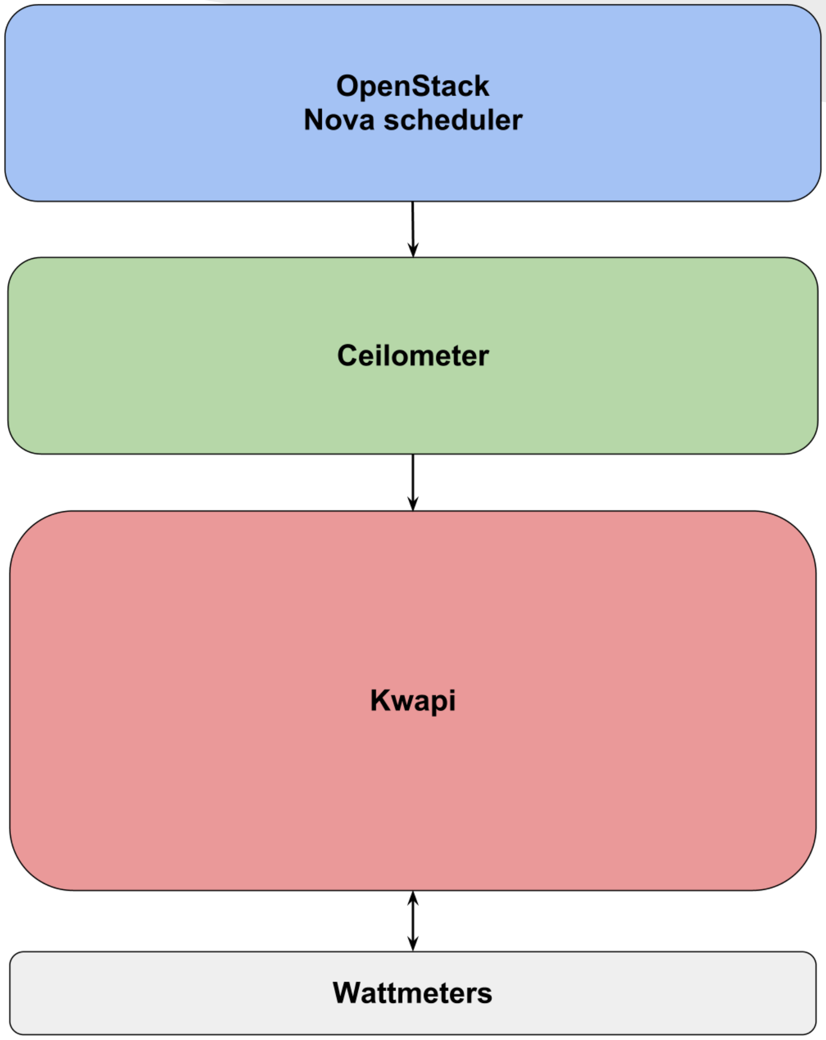
- Compute (**Nova**)
- Object Storage (Swift)
- Block Storage (Cinder)
- Networking (Quantum)
- Identity (**Keystone**)
- Dashboard (Horizon)

Recently added:

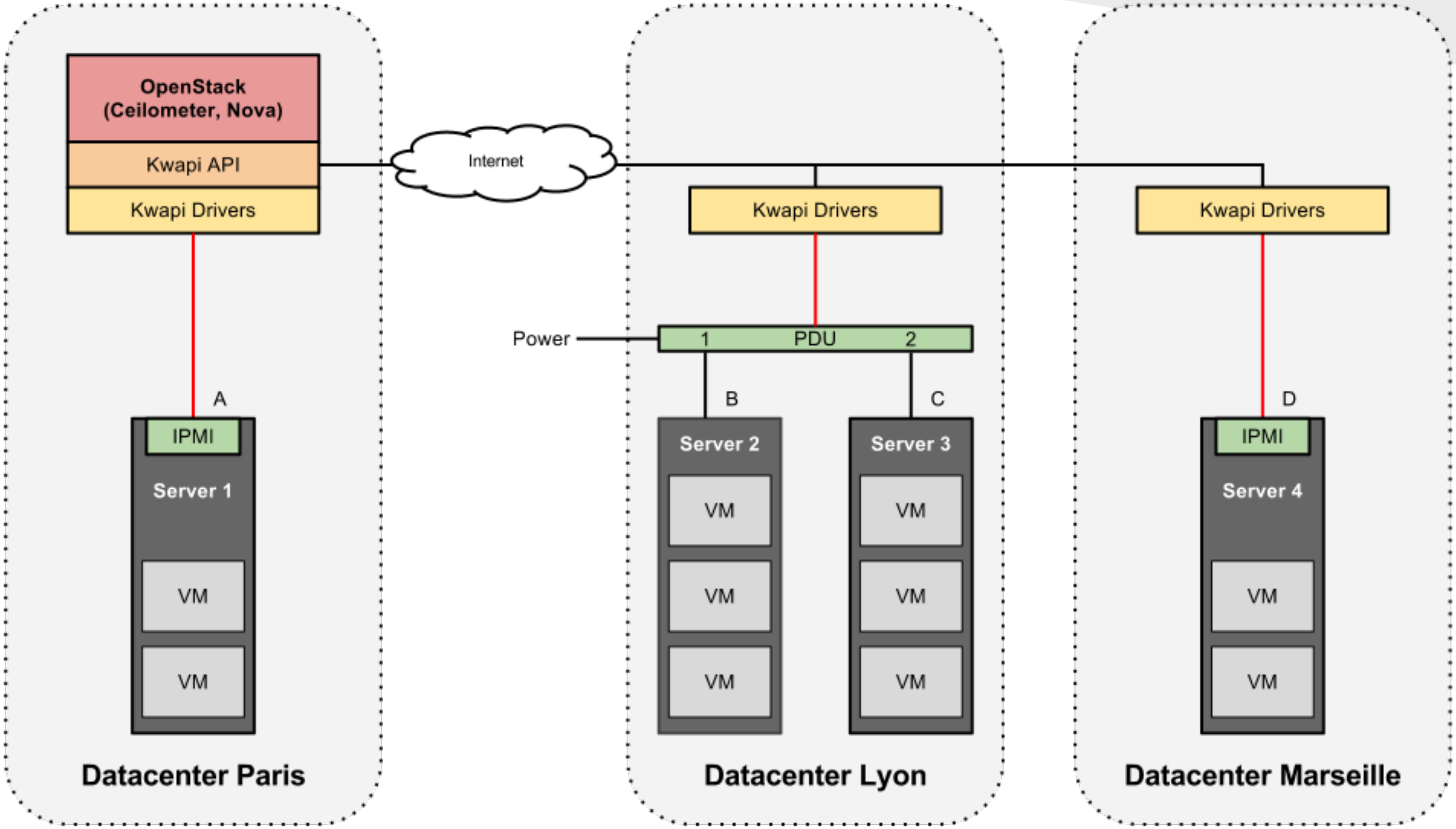
- Metering / billing (**Ceilometer**)

Incubation:

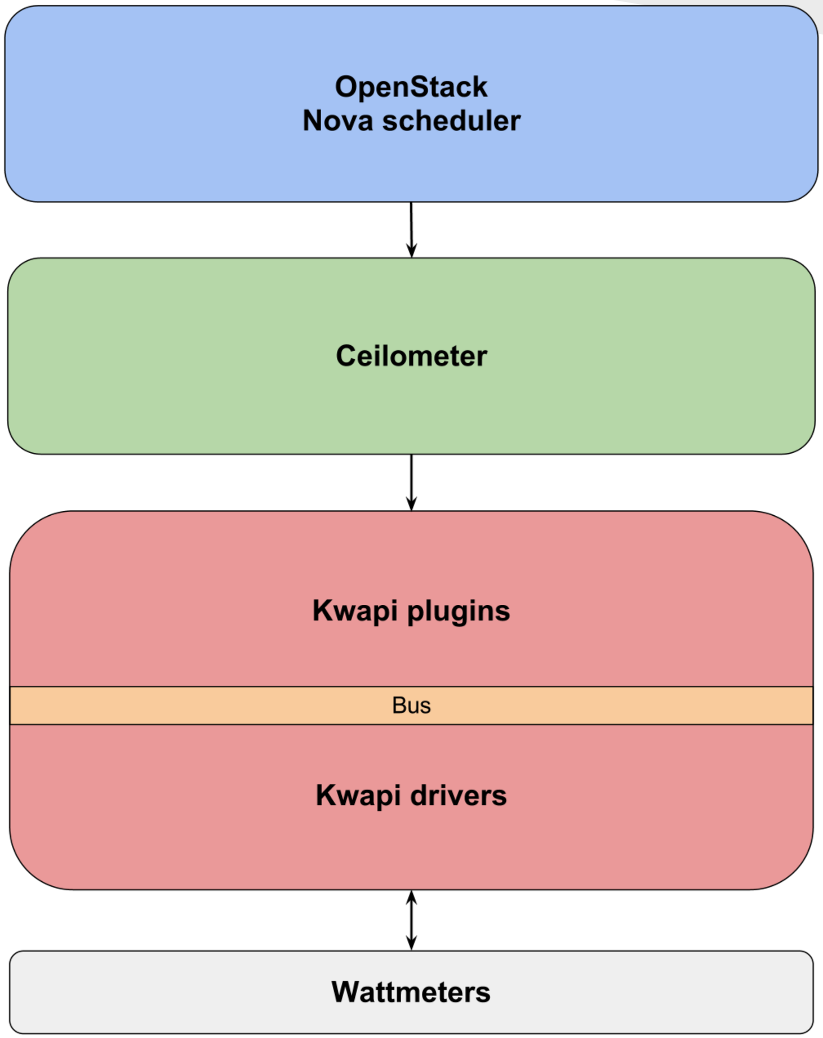
- Energy (**Kwapi**)



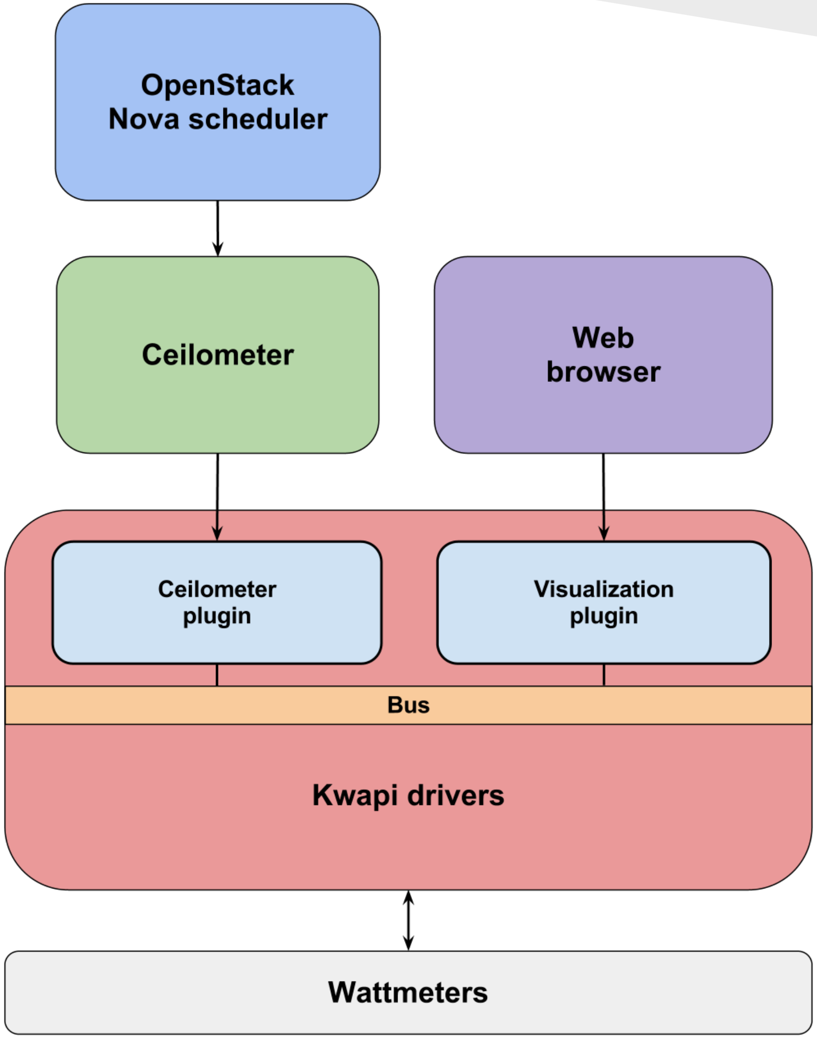
Telemetry architecture
Datacenter overview



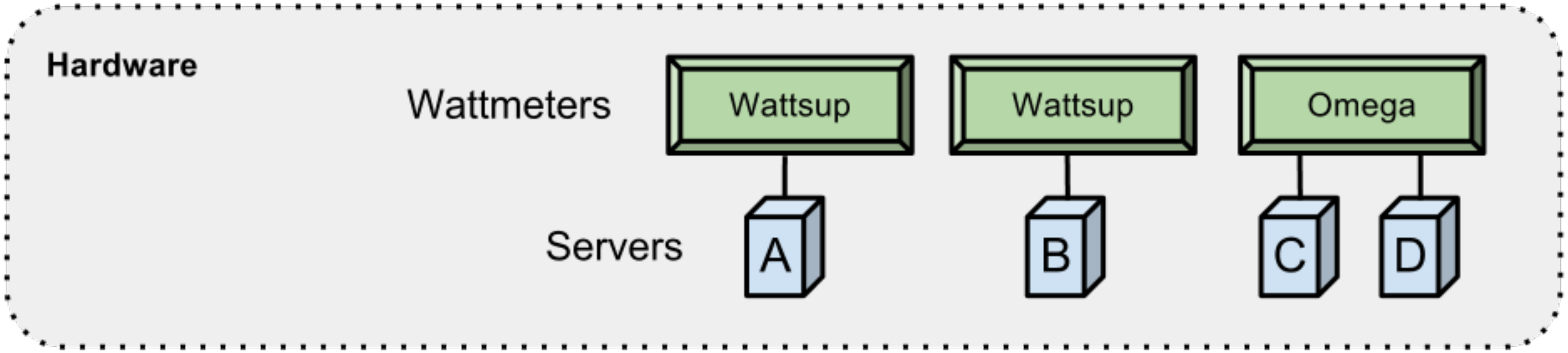
Telemetry architecture
Software layers



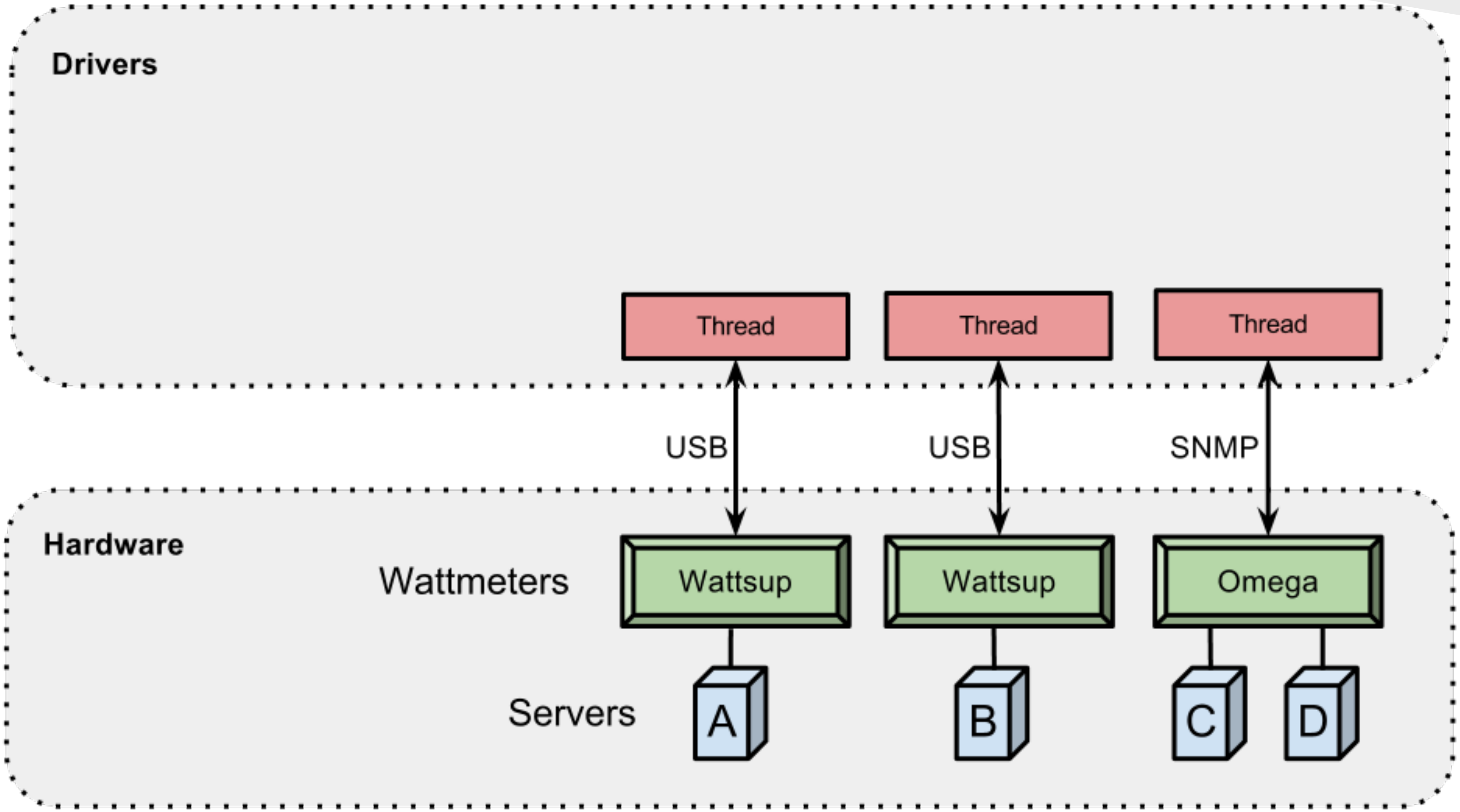
Telemetry architecture
Software layers



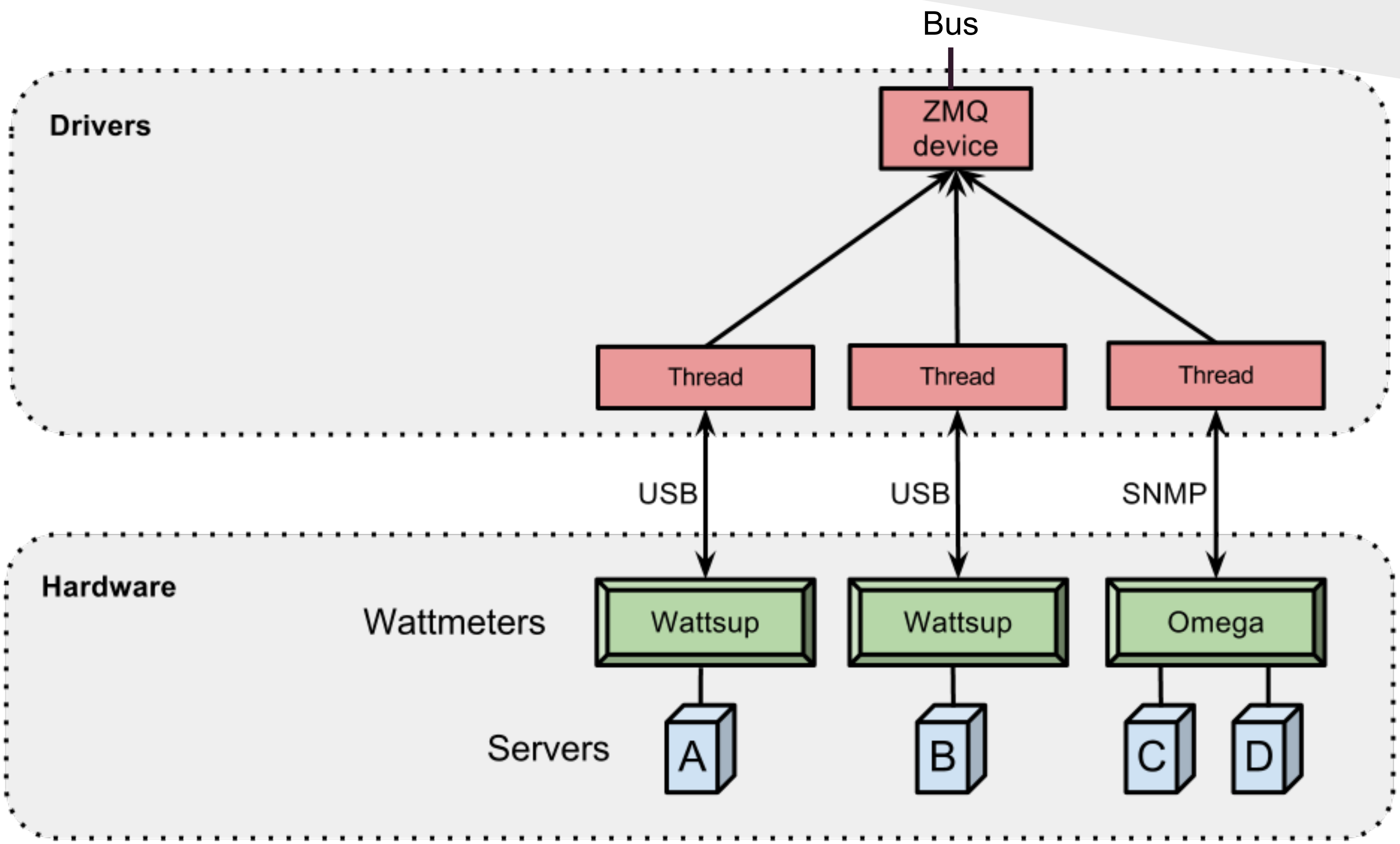
Telemetry architecture
Drivers layer



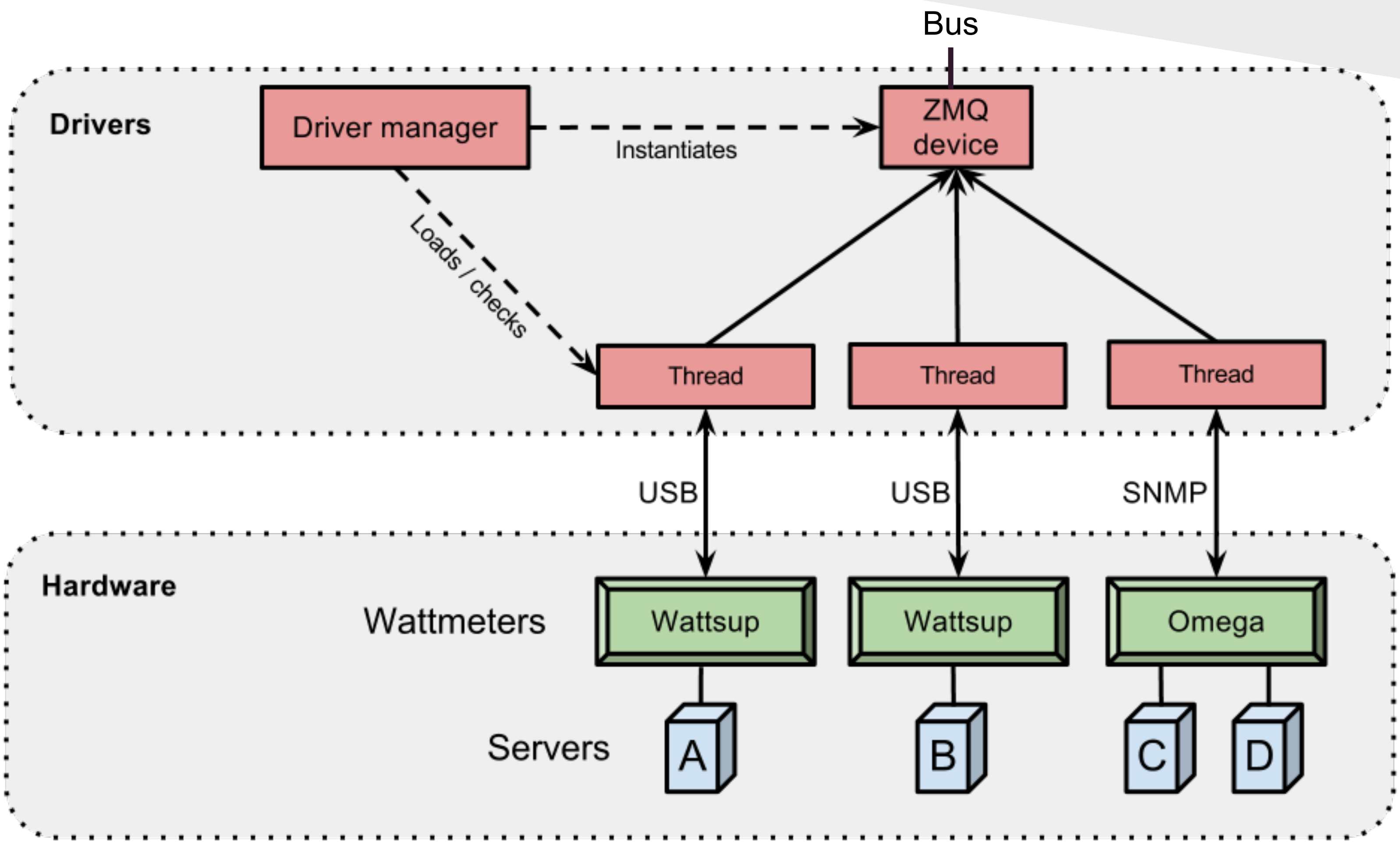
Telemetry architecture
Drivers layer



Telemetry architecture
Drivers layer



Telemetry architecture
Drivers layer



Telemetry architecture
Bus frameworks

ZeroMQ (used in Kwapi):

- Very fast
- Small (1.6 Mo)
- Written in C++ (provide a Python wrapper)
- Socket types: inproc, ipc, tcp
- Reliable / preserves order of messages
- Simple to use design patterns (publish/subscribe, request/response, ...)
- Brokerless

RabbitMQ (used in OpenStack):

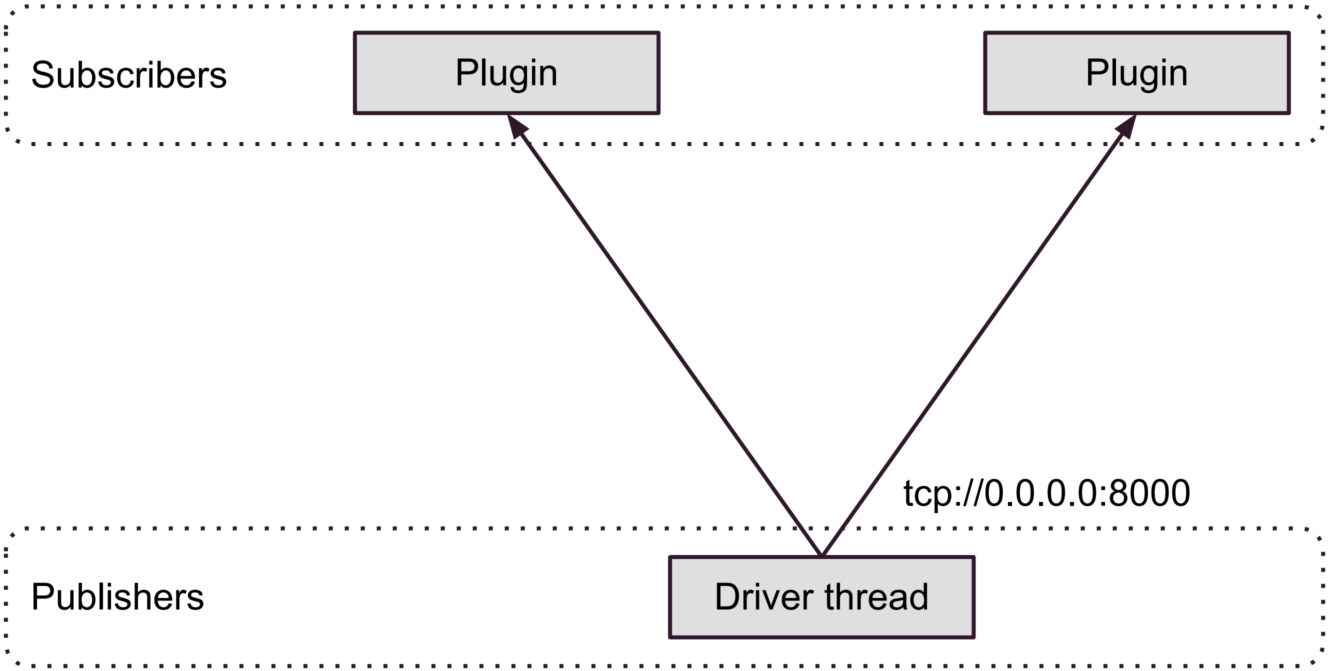
- Much more slower (10x)
- Require Erlang (70 Mo)
- Broker

Sockets (without framework):

- Why reinvent the wheel?

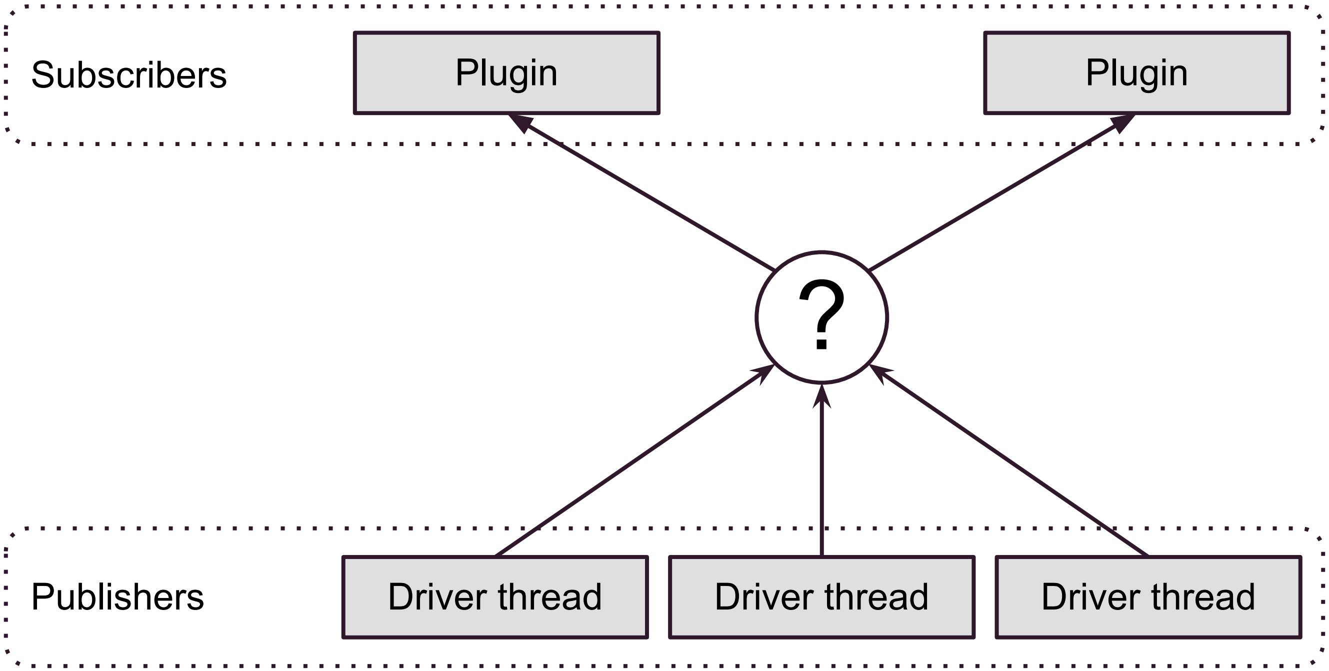
Telemetry architecture
ZeroMQ design pattern

Publish/subscribe design pattern



Telemetry architecture
ZeroMQ design pattern

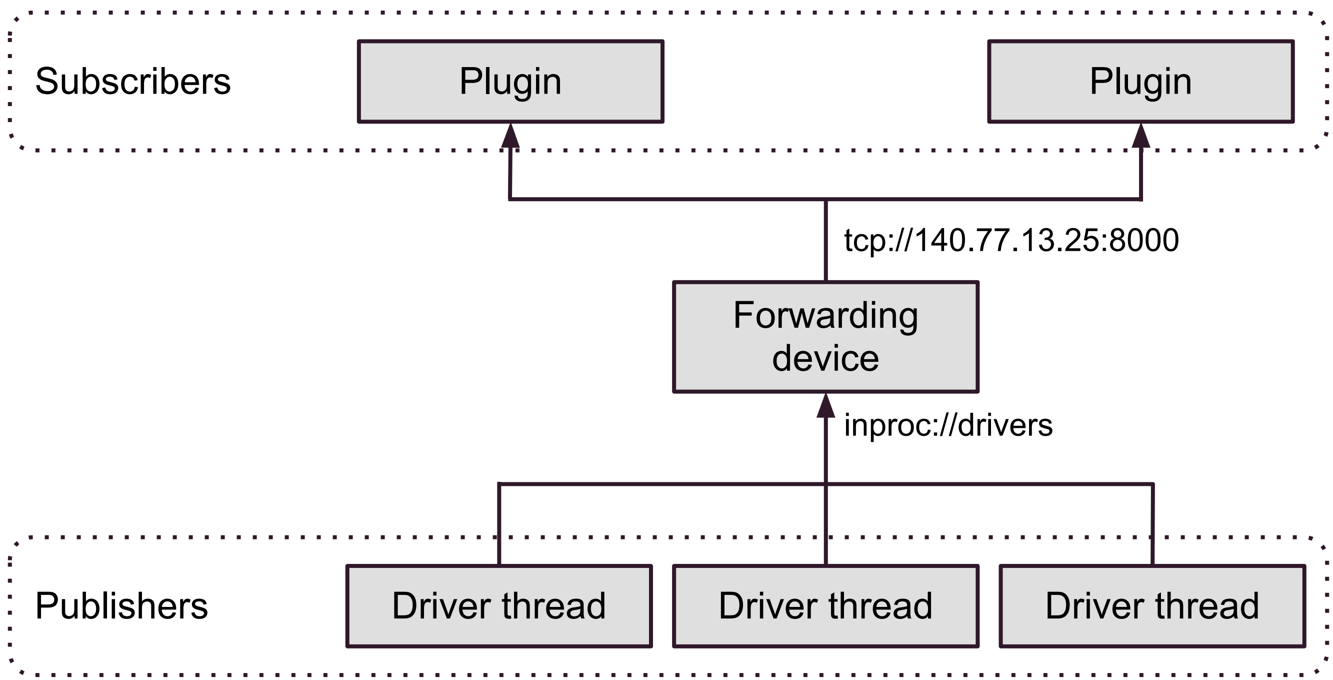
Publishers and subscribers need common endpoints



Telemetry architecture
ZeroMQ design pattern

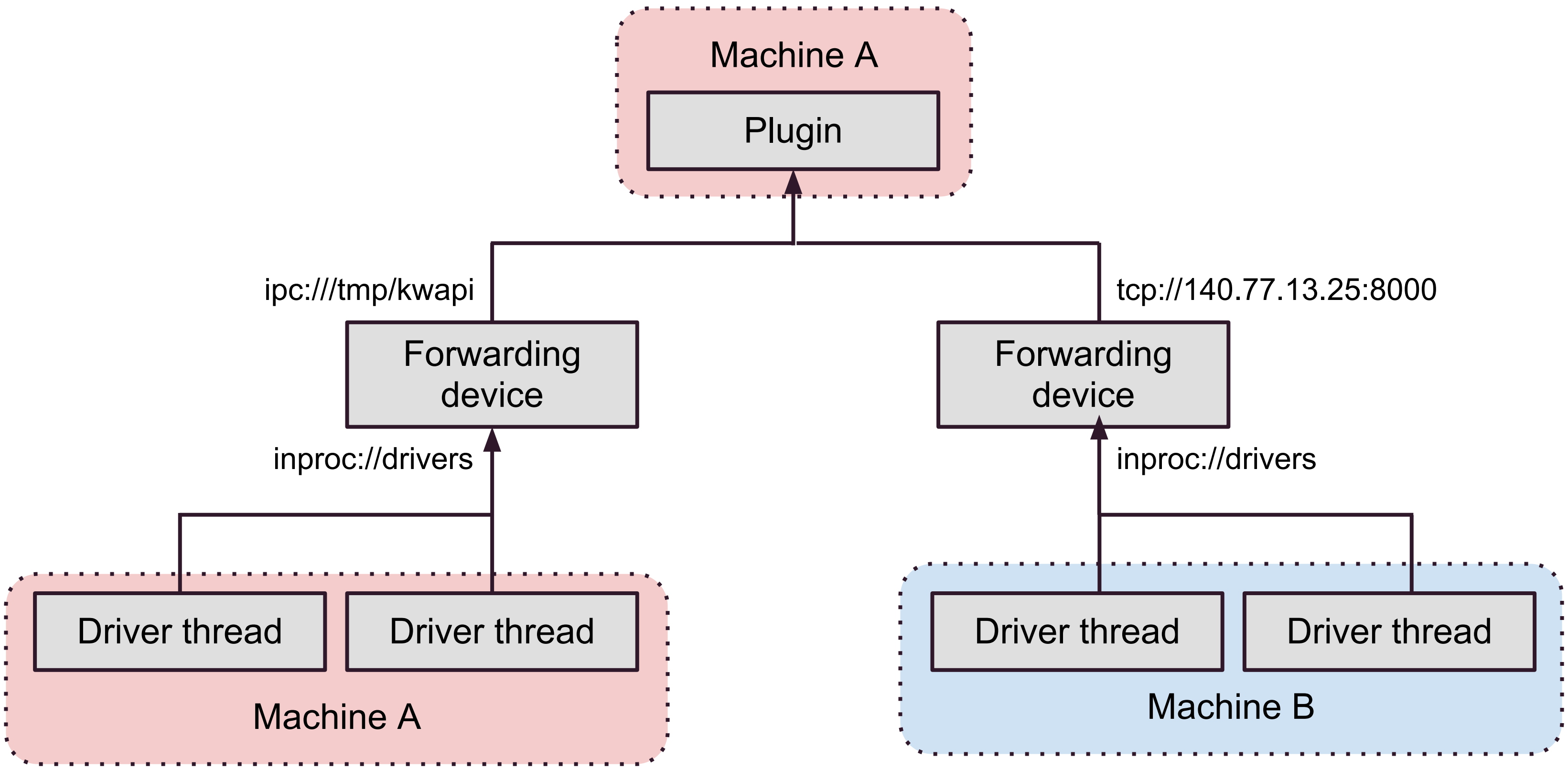
Forwarding device:

- Subscribes to inproc://drivers
- Publishes all received packets on tcp://140.77.13.25:8000



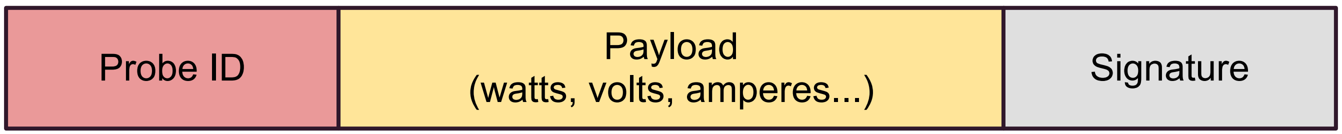
Telemetry architecture
ZeroMQ design pattern

Subscribers can listen multiple endpoints



Telemetry architecture
Bus messages format

Python dictionary:

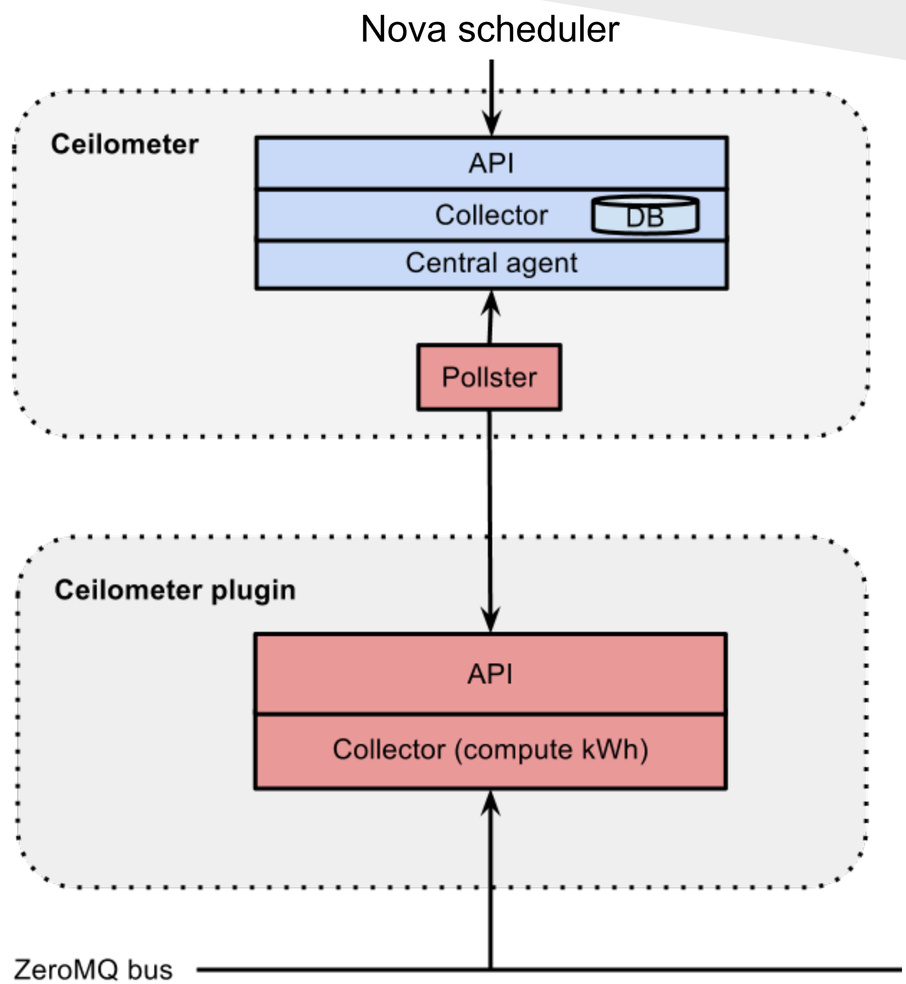


Three mandatory fields:

- Probe ID
- Watts
- Signature

Signature based on a shared secret key

Telemetry architecture
Ceilometer overview



Telemetry architecture
API plugin

Collector:

- Collects power consumption data
- Computes kWh and stores the last value (watts)

API (based on Flask):

/v1/probe-ids/	The list of probe ids
/v1/probes/	All detailed information about all probes
/v1/probes/A/	Detailed information about probe A
/v1/probes/A/kwh	Energy consumed by probe A

Authentication:

- The pollster provides a token (X-Auth-Token)
- The plugin checks the token (Keystone request)
- If the token is valid, requested data are sent

Telemetry architecture
Ceilometer pollster

Pollster:

- Is run periodically by Ceilometer central agent
- Asks to Keystone the Ceilometer plugin address
- Retrieves data
- Publishes kWh and watts counters

Collector stores published counters

API is queried by the Nova Scheduler to make a placement decision

Telemetry architecture
Visualization plugin

Telemetry architecture
Visualization plugin

Writes power consumption into RRD files:

- Severals archived periods with different resolutions
- RRD file size = 10 Ko (1000 probes = 10 Mo)

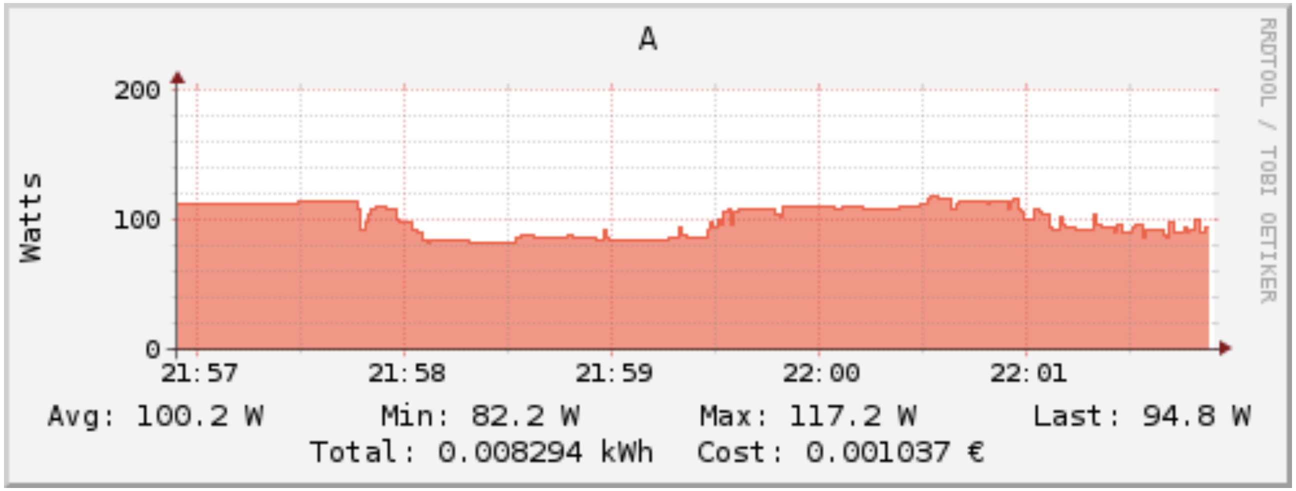
Webpage based on Flask:

- Two visualization modes (per periods and per probes)
- Summary graphs
- Cache mechanism (rebuild graph only if outdated)

Telemetry architecture
Visualization plugin

API example

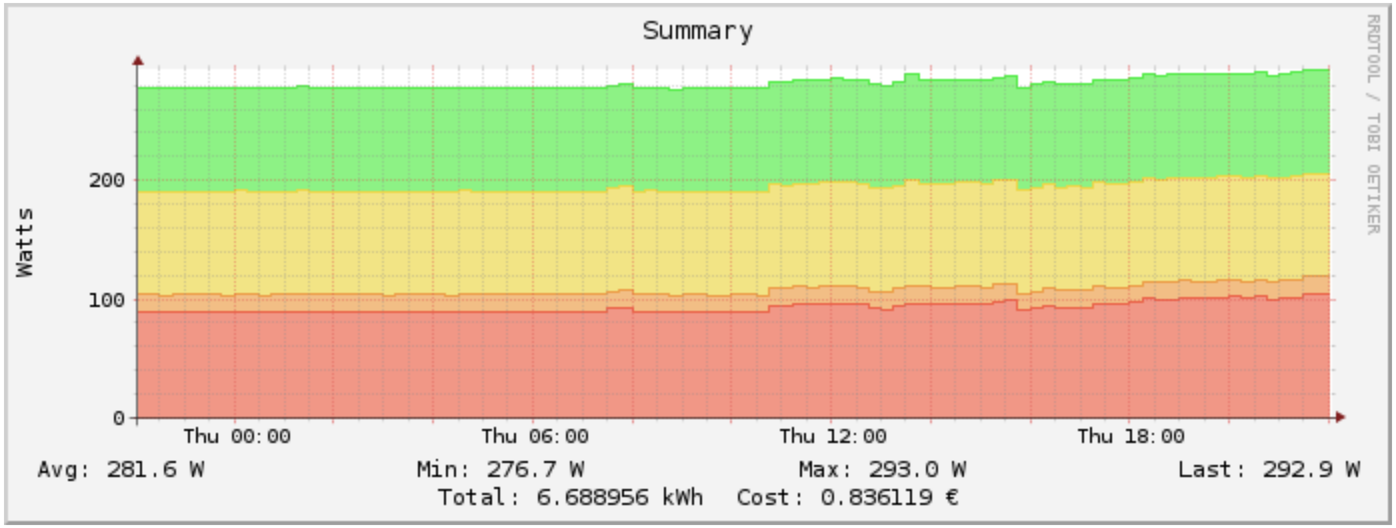
/graph/minute/A



Telemetry architecture
Visualization plugin

API example

/graph/day/



Summary

Context

Telemetry architecture

Scheduling / sleep modes (future works)

Scheduling
Choosing the greenest place

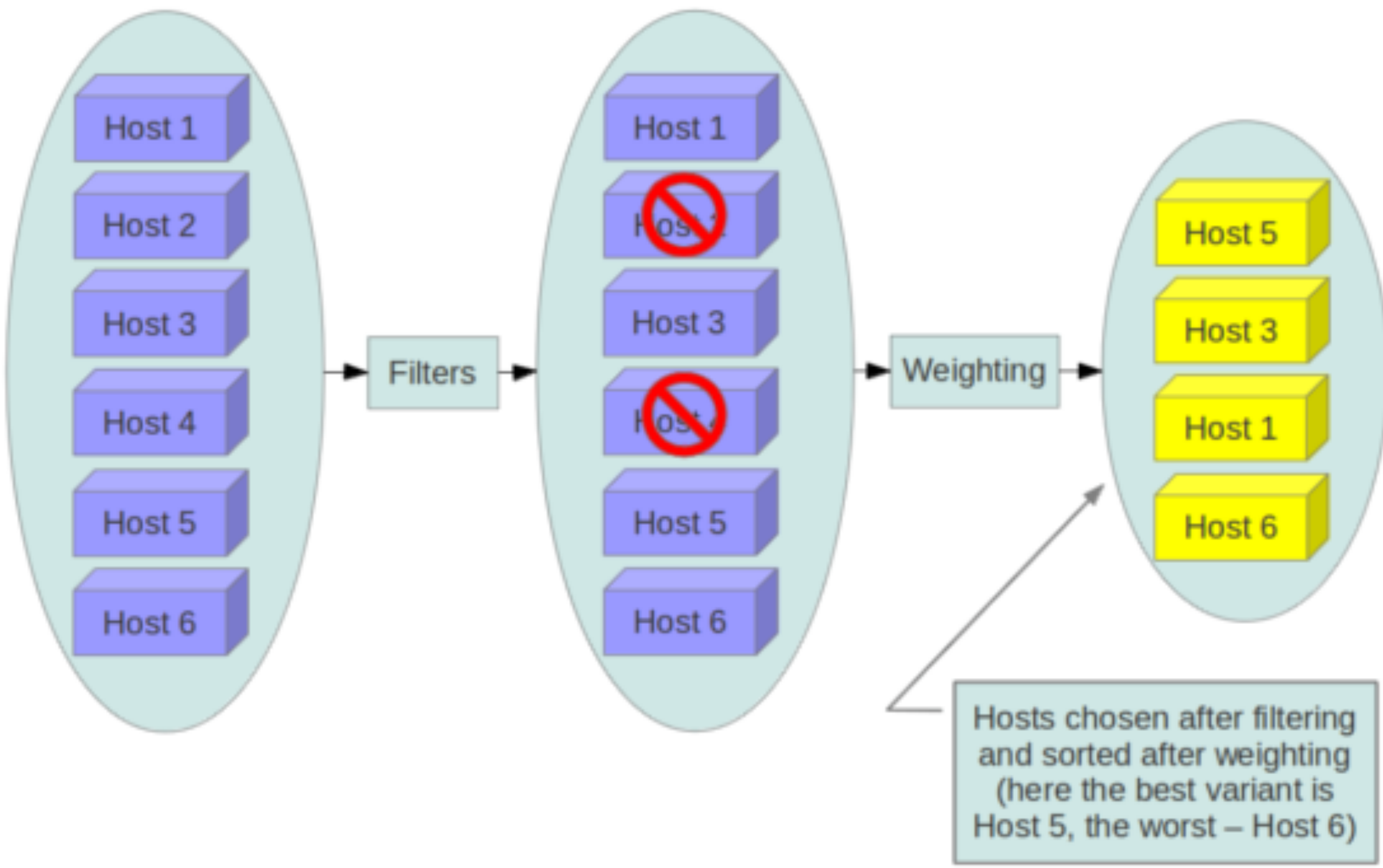
Where is the greenest place to run your job?

It depends on your job:

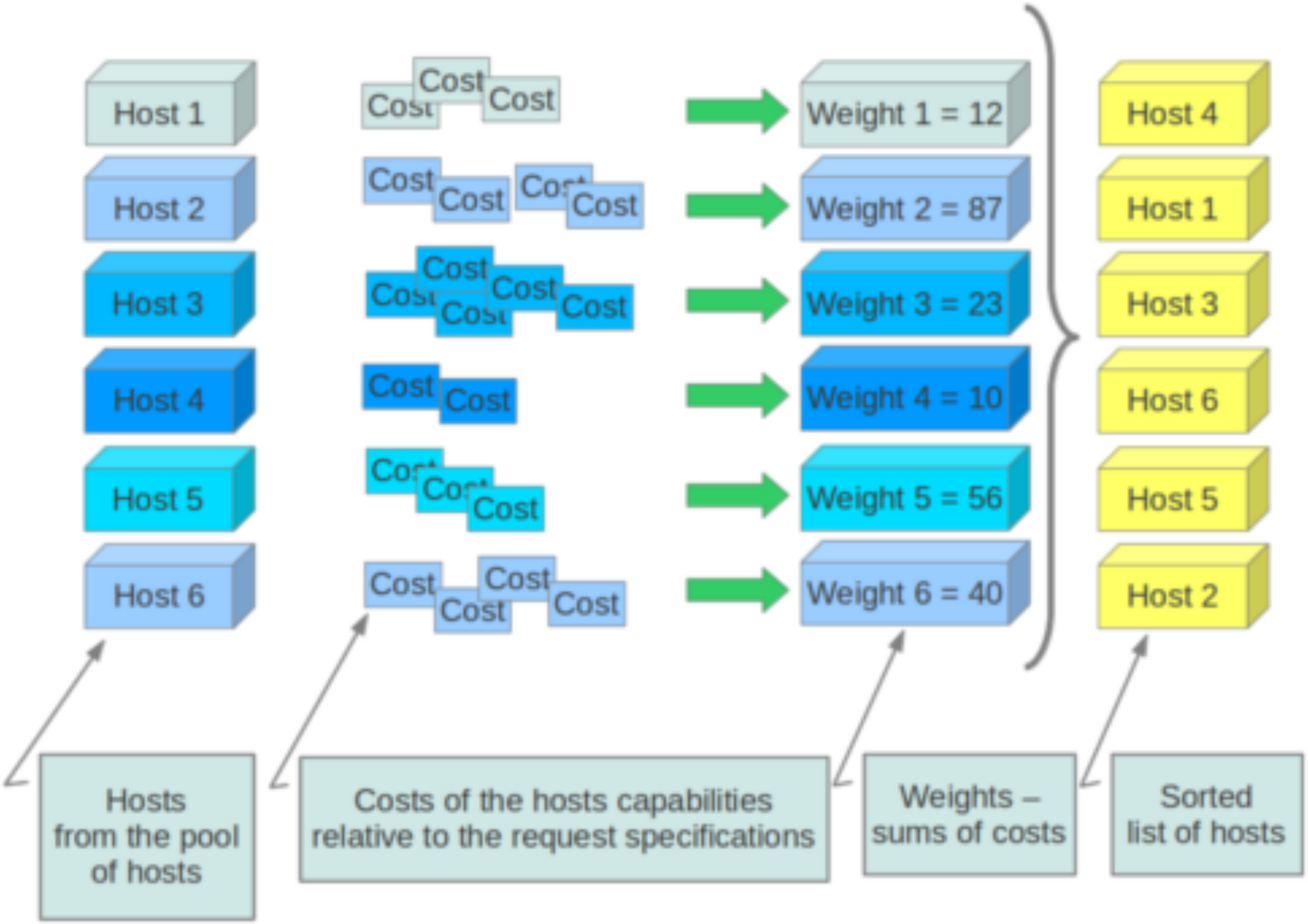
- CPU / GPU / memory / storage / network intensive ?
- Hard to estimate: vary over time, external events...

Approach: use a benchmark for efficiency rating.

Scheduling
Nova scheduler



Scheduling
Nova scheduler



Turning off unused machines

Using power saving modes:

- Which mode to choose?
  => Standby / hibernation

- How many machines should be turned off?
  => Anticipating demand and avoiding frequent shutdown / start-up cycles

- How much energy does it save? Is it profitable?
  => Peak start-up power

- Avoiding too frequent shutdown / start-up cycles
  => Sparing the old computers, but they are the least efficient ones

Conclusion

Telemetry:

- Writing more drivers
- Improving scalability

Scheduling and sleep modes:

- Implementing the strategies
- Live VM migration

Measuring energy with wattmeters is not all:

- What about the energy needed to build or recycle the servers?
- What about PUE (on a distributed architecture ?)

Thank you
for your attention