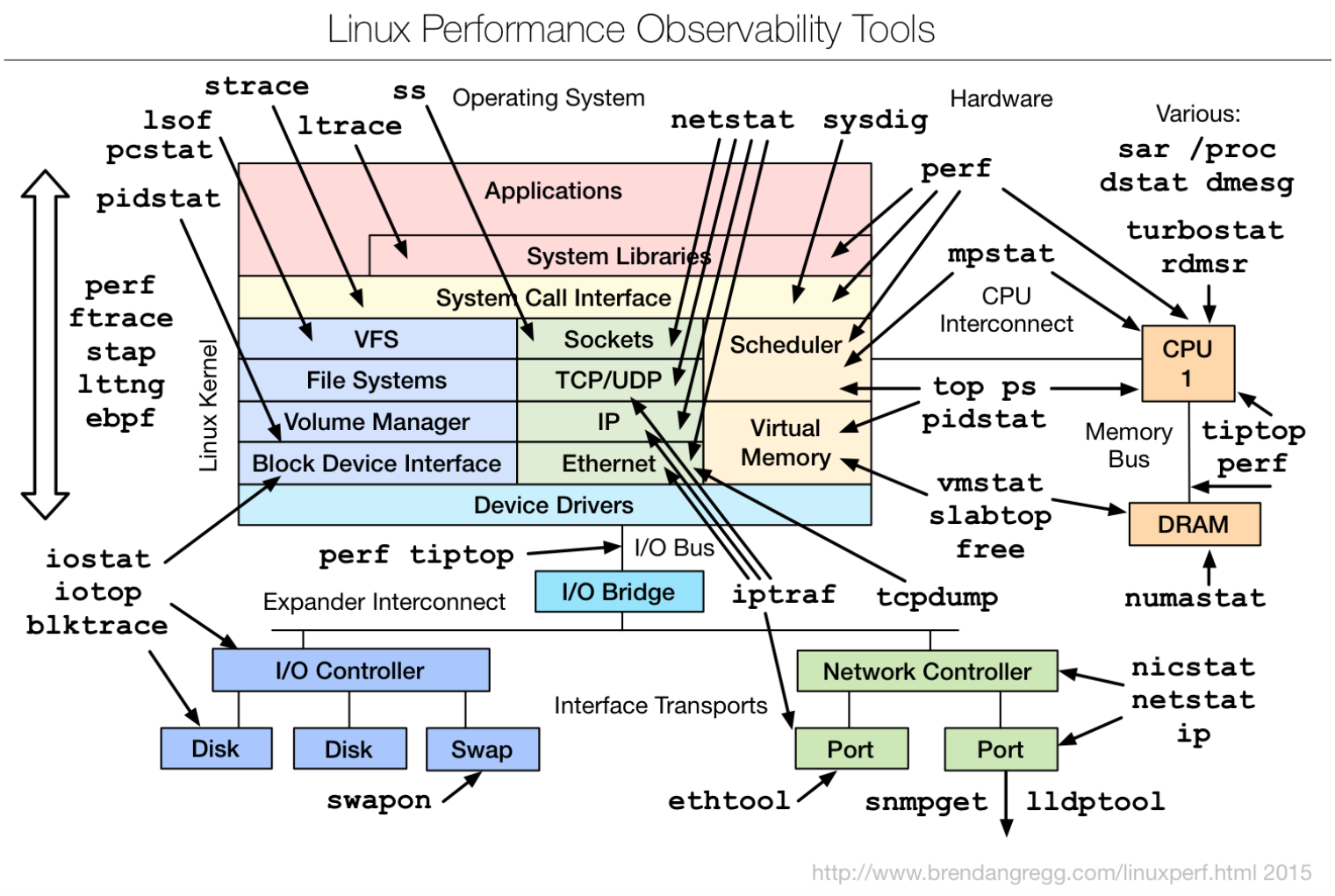
Kwapi: A Unified Monitoring Framework for Energy Consumption and Network Traffic

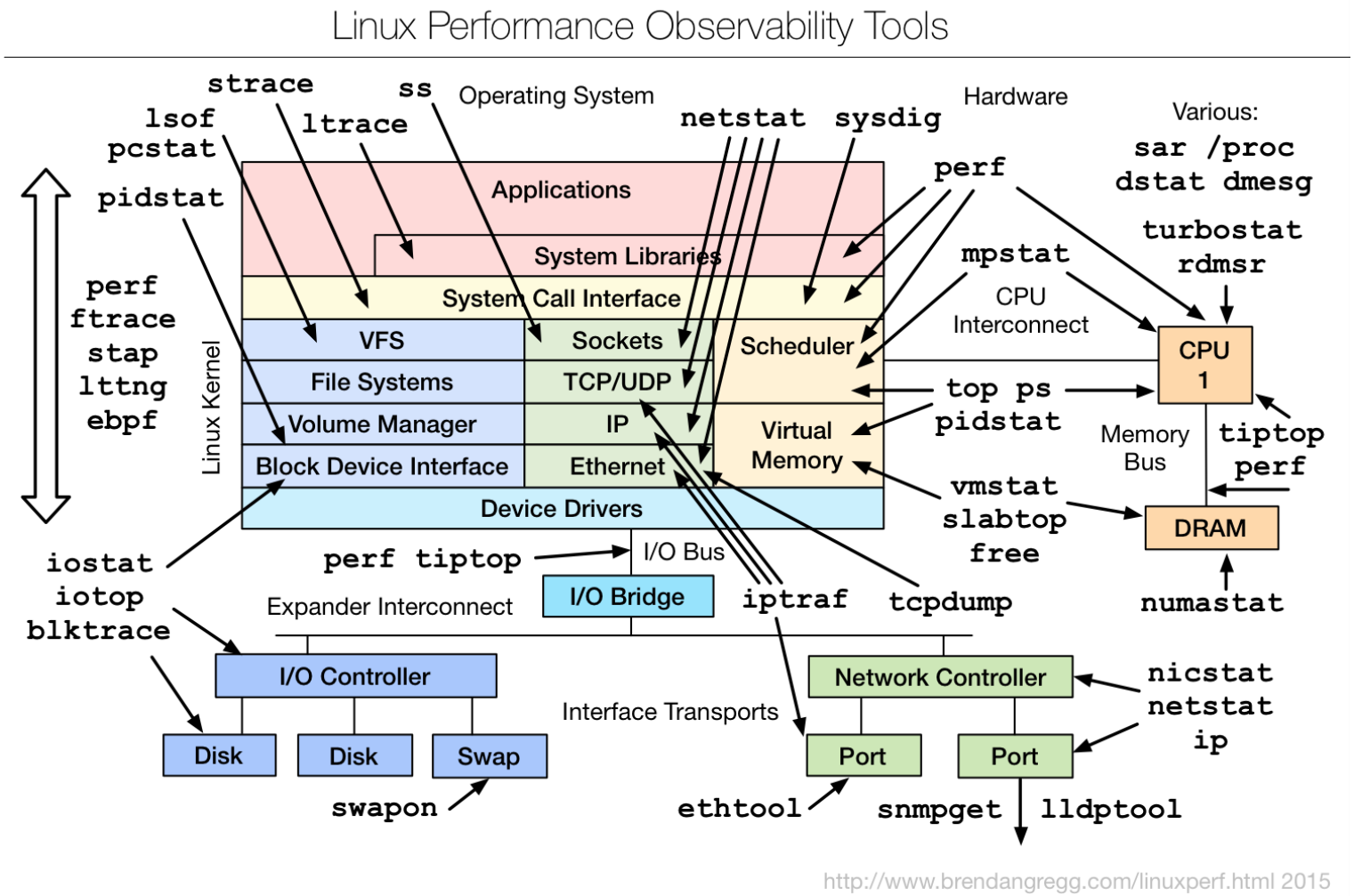
Florentin Clouet, Simon Delamare, Jean-Patrick Gelas, Laurent Lefèvre,
Lucas Nussbaum, Clément Parisot, Laurent Pouilloux, François Rossigneux

Short version of a TRIDENTCOM'2015 talk
Paper + slides: `http://deb.li/kwapi`

OTS monitoring and measurement tools



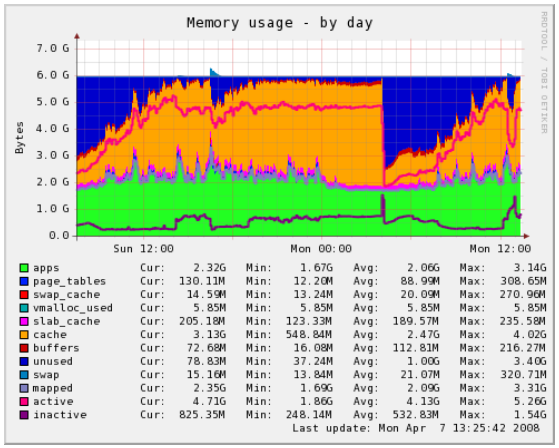
OTS monitoring and measurement tools

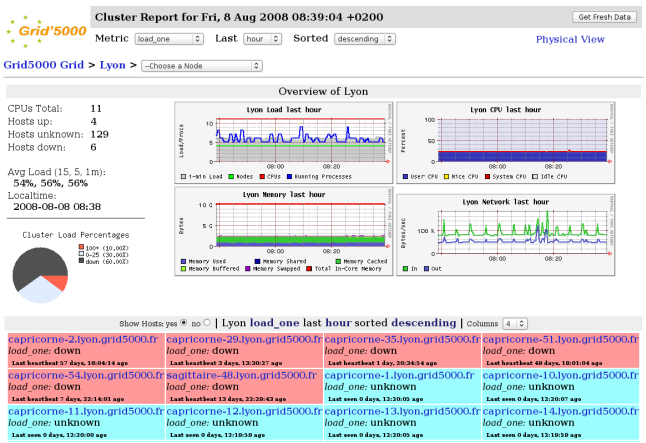


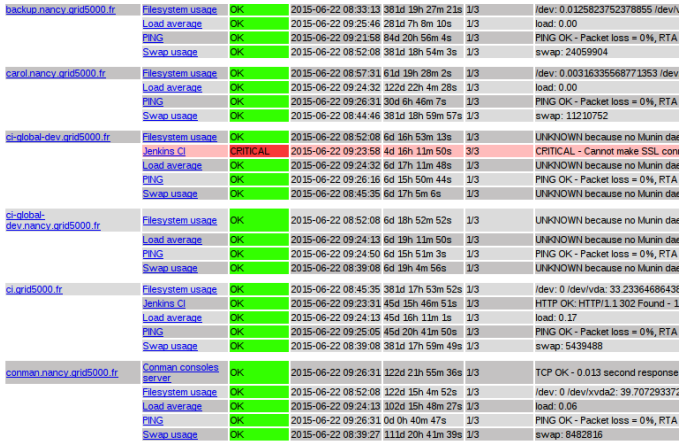
Many tools available, but:

- Need to be configured by the experimenters
- Often intrusive (running on users' nodes, non-negligible overhead)

Monitoring solutions for system administration







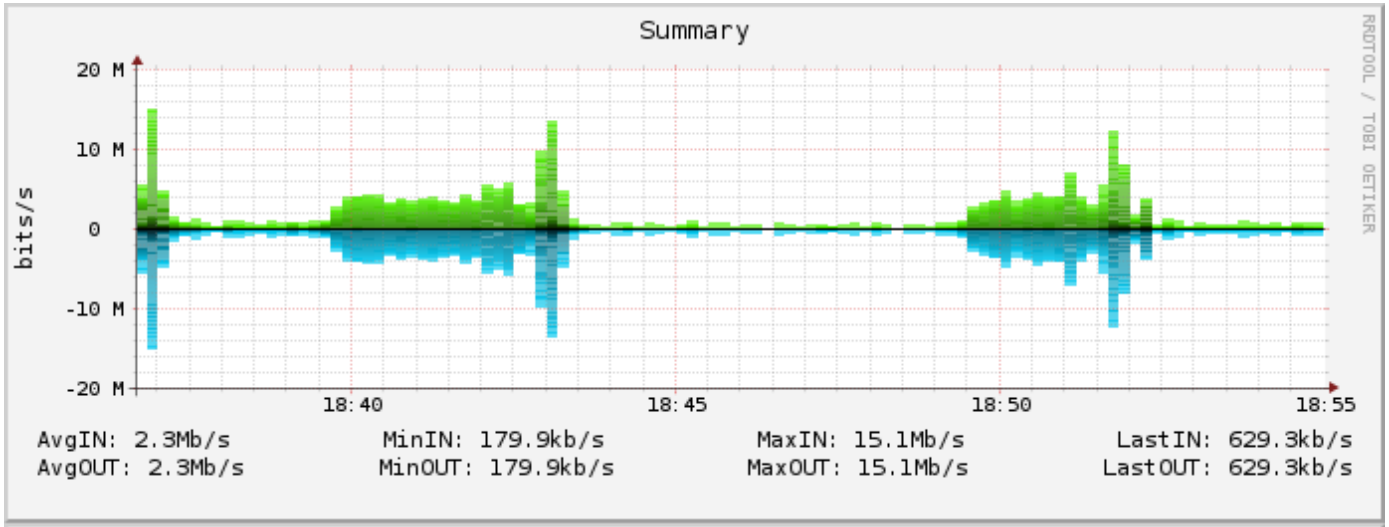
- MRTG, Munin, Ganglia, Nagios, etc.
- Main focus: monitor long term variations, tendencies
- Designed for low resolution (5 mins) $\rightsquigarrow$ unsuitable for experimenters

This talk: Kwapi

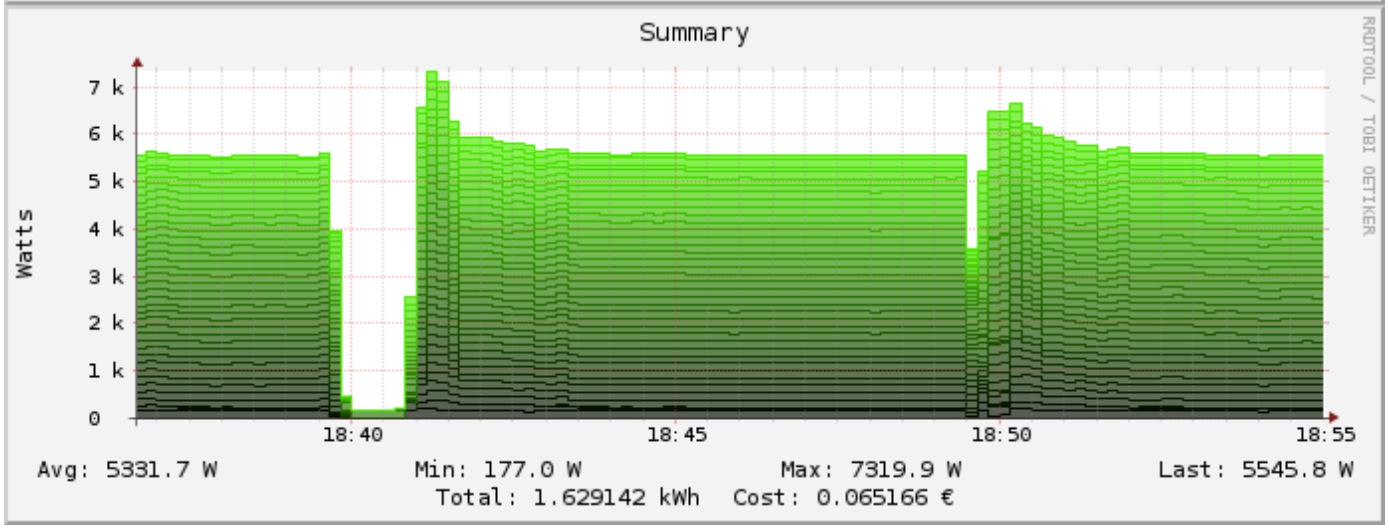
- **Monitoring and measurement framework for the Grid'5000 testbed**
- Initially designed as a power consumption measurement framework for OpenStack – then adapted to Grid'5000's needs and extended
- For energy consumption and network traffic
- Measurements taken at the infrastructure level
  (SNMP on network equipment, power distribution units, etc.)
- High frequency (aiming at 1 measurement per second)

Multi-metrics support: energy and networking

Network traffic



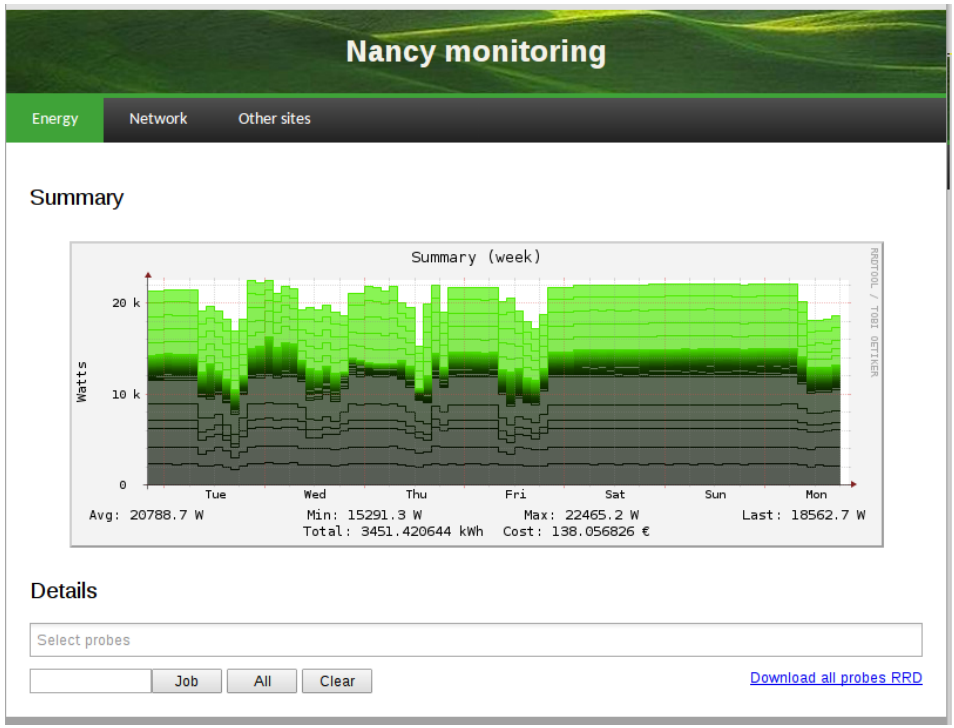
Power consumption



- 18:39:28 – machines are turned off
- 18:40:28 – machines are turned on again and generate network traffic as they boot via PXE
- 18:49:28 – machines reservation is terminated, causing a reboot to the default system

Data access and storage

- Metrics collected by Kwapi are stored:
  - In RRD files (typical for monitoring systems)
  - In HDF5 files, for long-term loss-less archival
    - One year of Grid'5000 monitoring = 720 GB
- Visualization via a web interface (selection by nodes or job numbers)

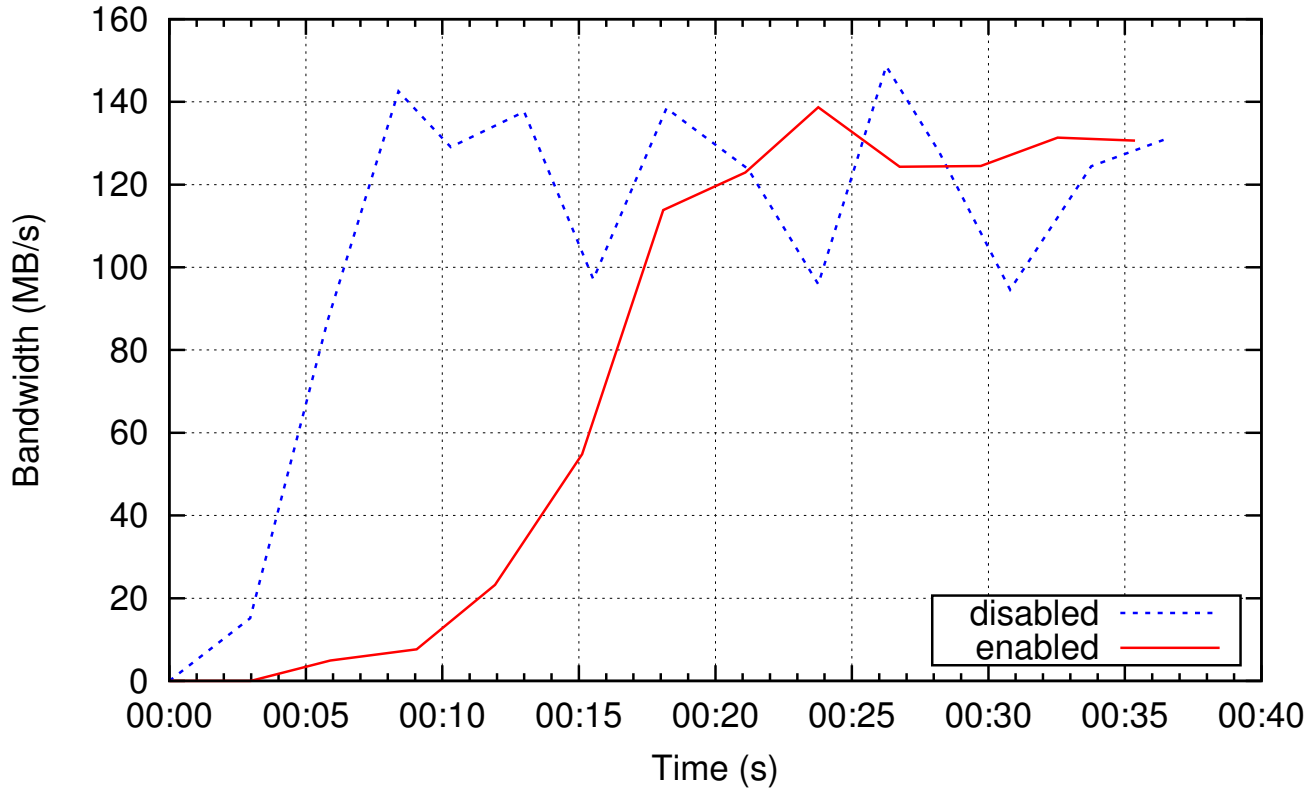


- Data also exported via the Grid'5000 REST API

Some example use cases

Visualizing TCP congestion control

- Linux's implementation of TCP CUBIC includes the Hystart heuristic
  - Detects congestion by measuring RTT
  - Broken until Linux 2.6.32



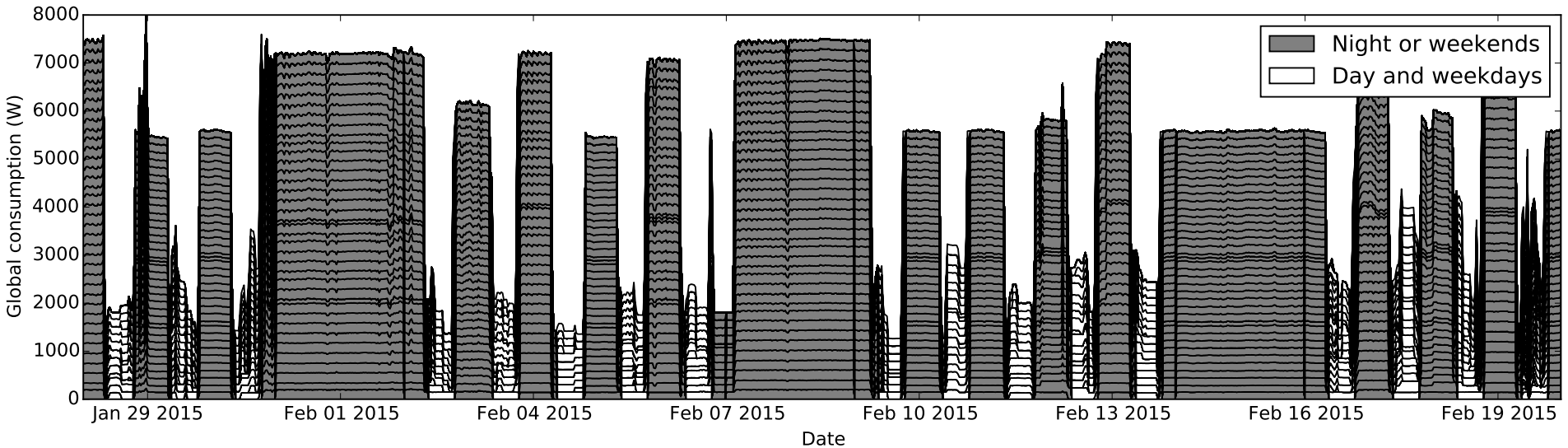
- Not as accurate as `nuttcp` or `iperf` but:
  - Measurements are completely passive from the experiment POV
  - No instrumentation required on nodes

Extracting power consumption trends

- Grid'5000 distinguishes between two time periods:
  - daytime – shared usage to prepare experiments
  - nights and week-ends – large scale experiments
- As a result, there are often free resources during the day
- Also, nodes are automatically shut down when not used
- **Does this reflect in power consumption as seen by Kwapi?**

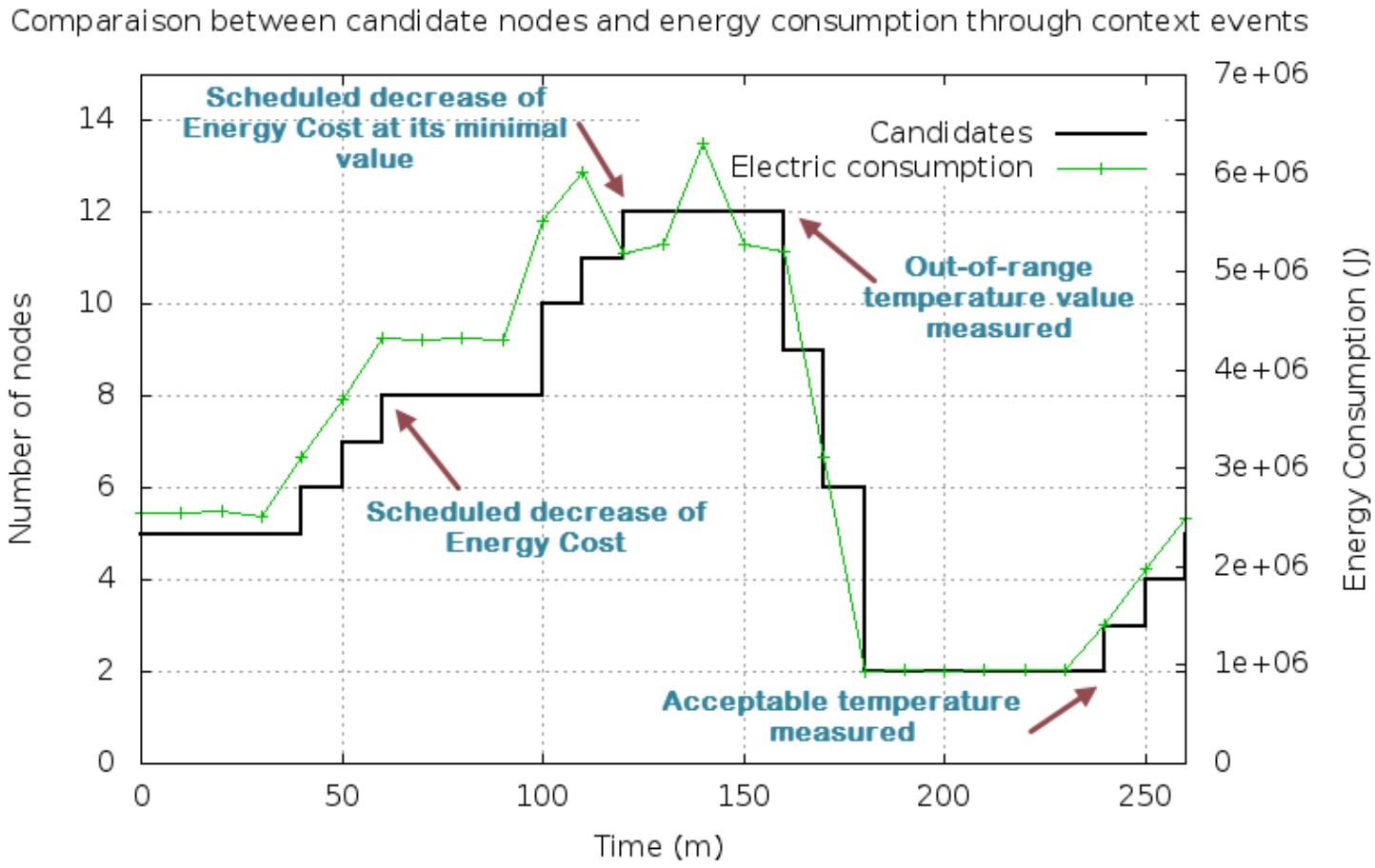
Extracting power consumption trends

- Grid'5000 distinguishes between two time periods:
  - daytime – shared usage to prepare experiments
  - nights and week-ends – large scale experiments
- As a result, there are often free resources during the day
- Also, nodes are automatically shut down when not used
- **Does this reflect in power consumption as seen by Kwapi?**



Evaluating energy-aware schedulers

- DIET: energy-aware distributed computing middleware
- Scheduler starts computing nodes based on energy cost
- Kwapi provides a feedback loop



Conclusions

- **Kwapi: the integrated monitoring solution of the Grid'5000 testbed**
- Already widely used on Grid'5000
- Available as free software
- Try it on your testbed, or on Grid'5000 (Open Access program)
- Future work (collaboration opportunities?)
  - Additional metrics: reactive power, network errors, Infiniband, storage systems, server room temperature, etc.
  - Integrate with other monitoring solutions (sFlow/NetFlow, collectd)
  - OML support: expose measurement points

Backup slides

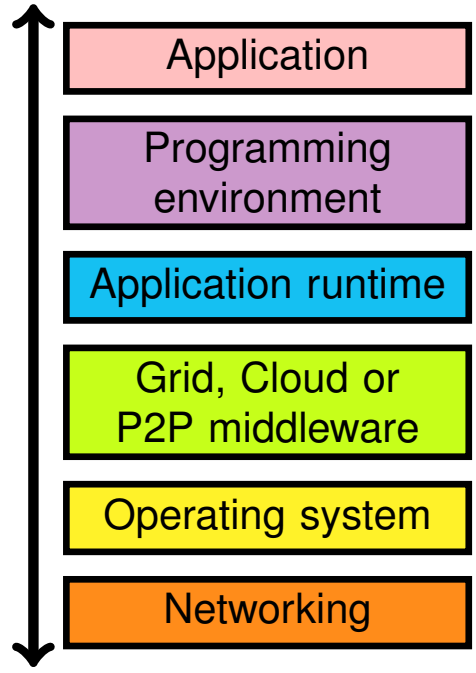
Context: Grid'5000

- Versatile testbed for research on **HPC, Clouds, Big Data**
- 10 sites (1 outside France)
- 24 clusters, 1000 nodes, 8000 cores
- 10-Gbps backbone (RENATER)
- Widely used since 2005:
  - **500+ users per year**
  - 700+ publications since 2009

`https://www.grid5000.fr/`

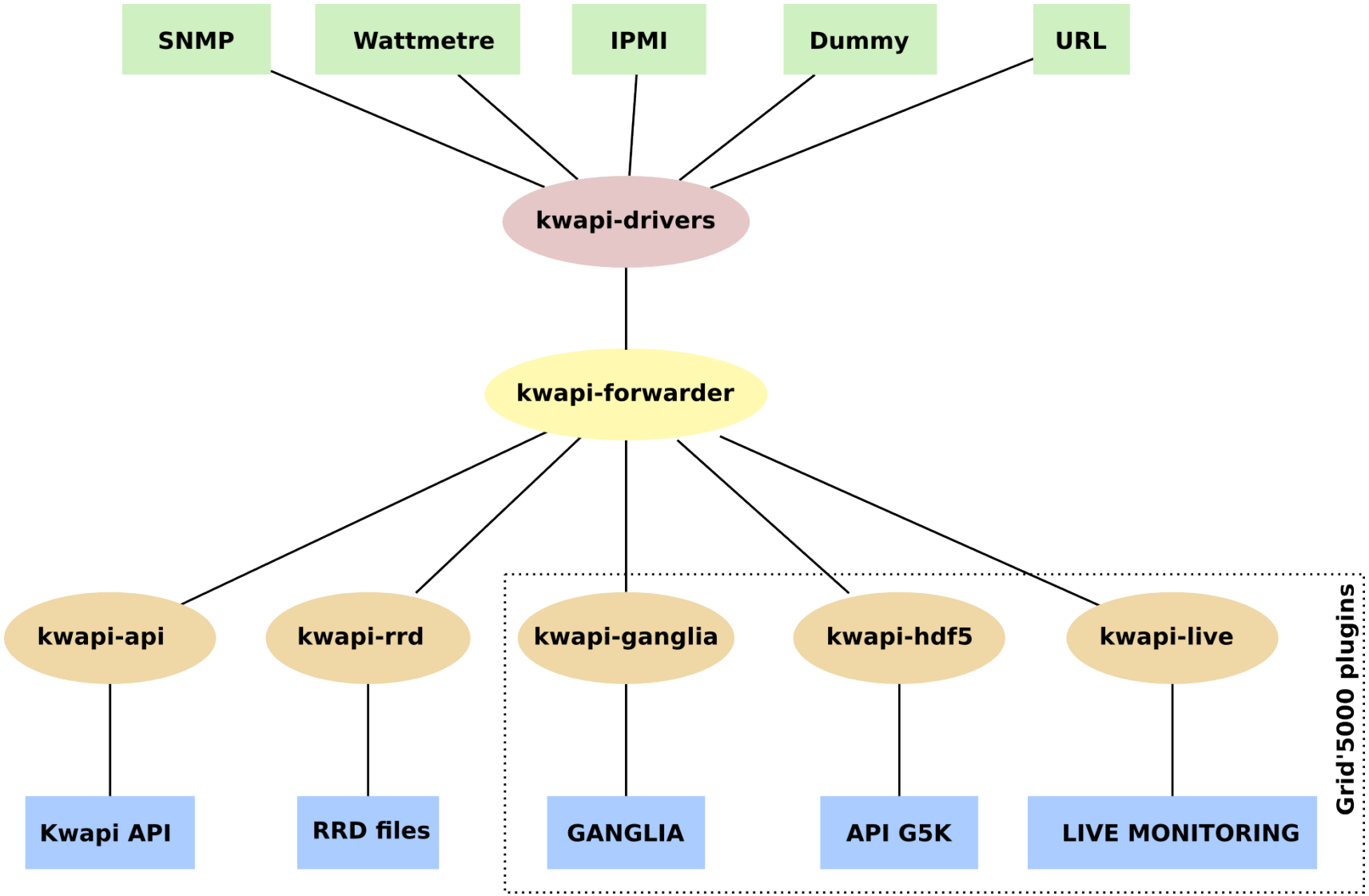
Maximizing support for advanced experiments

- **Complete control** of the testbed's resources, over the whole stack:
  - Bare-metal system image deployment
    $\rightsquigarrow$ Customize your kernel, use your own Cloud stack
  - Network isolation using KaVLAN
    $\rightsquigarrow$ no perturbation; protect rest of the testbed
- **Trustworthiness**: automatic inventory and verification of resources (TRIDENTCOM'2014 paper)
- **Fully programmable** through a REST API
  $\rightsquigarrow$ Automating experiments $\rightsquigarrow$ reproducible research
- **Higher level tools** to facilitate HPC, Clouds, Big Data experiments



This paper: observability, monitoring, measurement

Architecture



Development and deployment challenges

- SNMP:
  - `GetBulkRequest` to fetch all metrics at once
  - 64 bits counters (32 bits cycle in 4s on a 10 Gbps network)
- Configuration generated automatically from Grid'5000 Reference API
  - Describes each node's hardware, including where it is connected (network switch port, PDU port)
  - Format of SNMP's *IF-Descr* fields
    ```
    GigabitEthernet1/%LINECARD%/%PORT%
    TenGigabitEthernet%LINECARD%/%PORT%
    Unit:  %LINECARD% Slot:  0 Port:  %PORT% Gigabit - Level
    ```
  - Includes handling of complex cases (2+ NIC, 2 PSU, shared PDU)
- Configuration is automatically tested
  (Stress CPU and network $\rightsquigarrow$ compare data retrieved from REST API)

Monitoring overhead

- Network traffic: all monitoring traffic on a separate network (also used for e.g. remote control of nodes)
- Load on network equipment: no visible impact on performance