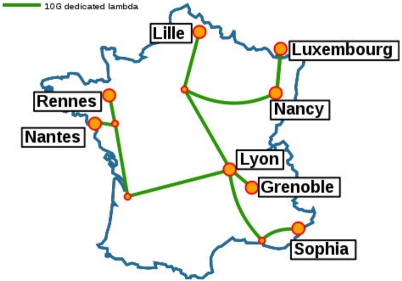
The data-centers facet of SILECS
(A.K.A. Grid'5000)

Frédéric Desprez & Lucas Nussbaum
Grid'5000 Scientific & Technical Directors

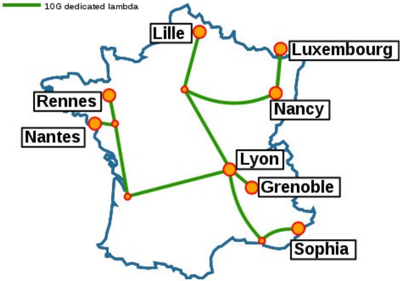
2019-04-25

The Grid'5000 testbed

- **A large-scale testbed for distributed computing**
  - 8 sites, 31 clusters, 828 nodes, 12328 cores
  - Dedicated 10-Gbps backbone network
  - 550 users and 120 publications per year



The Grid'5000 testbed



- **A large-scale testbed for distributed computing**
  - 8 sites, 31 clusters, 828 nodes, 12328 cores
  - Dedicated 10-Gbps backbone network
  - 550 users and 120 publications per year
- **A meta-cloud, meta-cluster, meta-data-center**
  - Used by CS researchers in HPC, Clouds, Big Data, Networking, AI
  - To experiment in a fully controllable and observable environment
  - Similar problem space as Chameleon and Cloudlab (US)
  - Design goals
    - Support high-quality, reproducible experiments
    - On a large-scale, distributed, shared infrastructure

Landscape – cloud & experimentation[1]

- Public cloud infrastructures (AWS, Azure, Google Cloud Platform, etc.)
  - ☹ No information/guarantees on placement, multi-tenancy, real performance

- Private clouds: Shared observable infrastructures
  - ☺ Monitoring & measurement
  - ☹ No control over infrastructure settings
  - ⇝ Ability to understand experiment results

- Bare-metal as a service, fully reconfigurable infrastructure (Grid'5000)
  - ☺ Control/alter all layers (virtualization technology, OS, networking)
  - ⇝ *In vitro* Cloud

And the same applies to all other environments (e.g. HPC)

---

[1] Inspired from a slide by Kate Keahey (Argonne Nat. Lab.)

Some recent results from Grid’5000 users

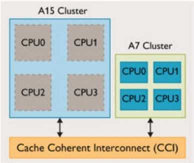
- Portable Online Prediction of Network Utilization (Inria Bdx + US)
- Energy proportionality on hybrid architectures (LIP/IRISA/Inria)
- Maximally Informative Itemset Mining (Miki) (LIRM/Inria)
- Damaris (Inria)
- BeBida: Mixing HPC and BigData Workloads (LIG)
- HPC: In Situ Analytics (LIG/Inria)
- Addressing the HPC/Big-Data/IA Convergence
- An Orchestration Syst. for IoT Applications in Fog Environment (LIG/Inria)
- Toward a resource management system for Fog/Edge infrastructures
- Distributed Storage for Fog/Edge infrastructures (LINA)
- From Network Traffic Measurements to QoE for Internet Video (Inria)

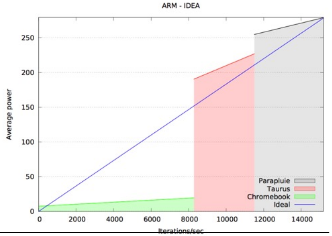
Portable Online Prediction of Network Utilization

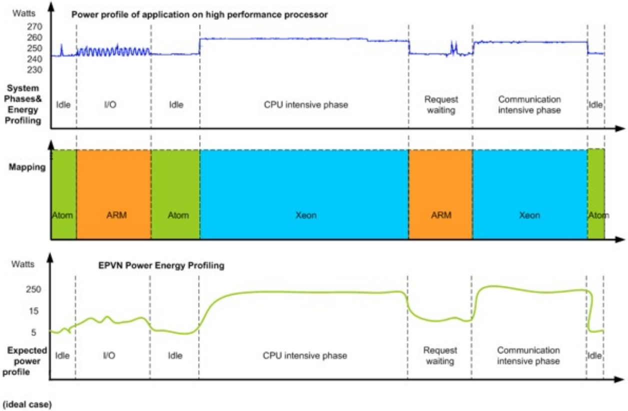
- **Problem**
  - Predict network utilization in near future to enable optimal utilization of spare bandwidth for low-priority asynchronous jobs co-located with an HPC application
- **Goals**
  - High accuracy, low compute overhead, learn on-the-fly without previous knowledge
- **Proposed solution**
  - Dynamic sequence-to-sequence recurrent neural networks that learn using a sliding window approach over recent history
  - Evaluate the gain of a tree-based meta-data management
  - INRIA, The Univ. of Tennessee, Exascale Comp. Proj., UC Irvine, Argonne Nat. Lab.
- **Grid'5000 experiments**
  - Monitor and predict network utilization for two HPC applications at small scale (30 nodes)
  - Easy customization of environment for rapid prototyping and validation of ideas (in particular, custom MPI version with monitoring support)
  - Impact: Early results facilitated by Grid'5000 are promising and motivate larger scale experiments on leadership class machines (Theta@Argonne)

Energy proportionality on hybrid architectures[2]

- Hybrid computing architectures : low power processors, co processors, GPUs. . .
- Supporting a “Big, Medium, Little” approach : the right processor at the right time







---

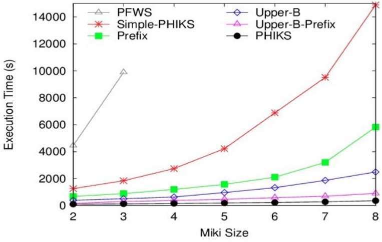
[2] V. Villebonnet, G. Da Costa, L. Lefèvre, J.-M. Pierson and P. Stolf. "Big, Medium, Little" : Reaching Energy Proportionality with Heterogeneous Computing Scheduler", Parallel Processing Letters, 25 (3), Sep. 2015

Maximally Informative Itemset Mining (Miki)[3]

Extracting knowledge from data

Miki: measures the quantity of information (e.g., based on joint entropy measure) delivered by the itemsets of size k in a database (i.e., k denotes the number of items in the itemset)

- **PHIKS, a parallel algorithm for mining of maximally informative k-itemsets**
  - Very efficient for parallel miki discovery
  - High scalability with very large amounts of data and high size of the itemsets
  - Includes several optimization techniques
  - Communication cost reduction using entropy bound filtering
  - Incremental entropy computation
  - Prefix/Suffix technique for reducing response time
- **Experiments on Grid'5000**
  - Hadoop/Map Reduce on 16 and 48 nodes
  - Datasets of 49 Gb (English Wikipedia, 5 millions articles), 1 Tb (ClueWeb, 632 millions articles)
  - Metrics: Response time, communication cost, energy consumption



Performance on ClueWeb Data Set

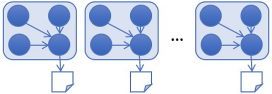
---

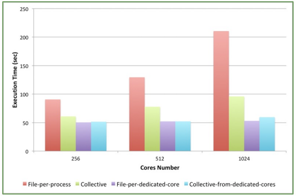
[3] S.Salah, R. Akbarinia, F. Masseglia. A Highly Scalable Parallel Algorithm for Maximally Informative k-Itemset Mining. Knowledge and Information Systems (KAIS), Springer, 2017, 50 (1)

Damaris

Scalable, asynchronous data storage for large-scale simulations using the HDF5 format

- **Traditional approach**
  - All simulation processes (10K+) write on disk at the same time synchronously
  - Problems: 1) I/O jitter, 2) long I/O phase, 3) Blocked simulation during data writing
- **Solution**
  - Aggregate data in dedicated cores using shared memory and write asynchronously
- **Grid'5000 used as a testbed**
  - Access to many (1024) homogeneous cores
  - Customizable environment and tools
  - Repeat the experiments later with the same environment saved as an image
  - The results show that Damaris can provide a jitter-free and wait-free data storage mechanism
  - G5K helped prepare Damaris for deployment on top supercomputers (Titan, Pangea (Total), Jaguar, Kraken, etc.)
  - https://project.inria.fr/damaris/

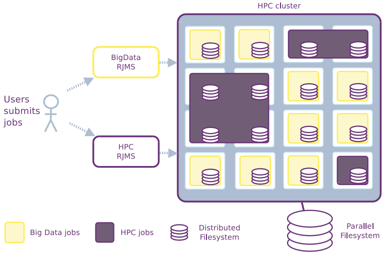


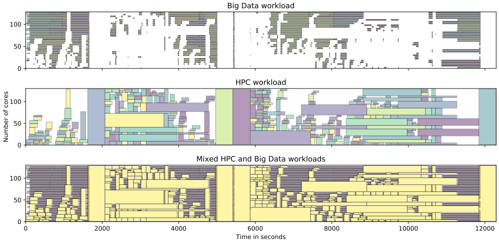


BeBida: Mixing HPC and BigData Workloads

Objective: Use idle HPC resources for BigData workloads

- **Simple approach**
  - HPC jobs have priority
  - BigData Framework: Spark/Yarn, HDFS
  - Evaluating costs of starting/stopping tasks (Spark/Yarn) and data transferts (HDFS)
- **Results**
  - It increases cluster utilisation
  - Disturbance of HPC jobs is small
  - Big Data execution time varies (WIP)





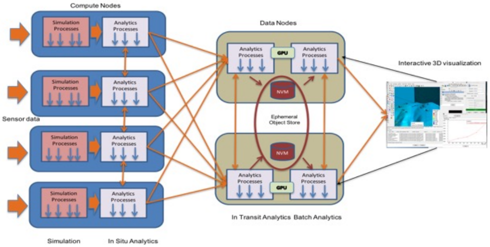
HPC: In Situ Analytics

Goal: improve organization of simulation and data analysis phases

- Simulate on a cluster; move data; post-mortem analysis
  - Unsuitable for Exascale (data volume, time)
- Solution: analyze on nodes, during simulation
  - Between or during simulation phases? dedicated core? node?

Grid'5000 used for development and test, because control

- of the software environment (MPI stacks),
- of CPU performance settings (Hyperthreading),
- of networking settings (Infiniband QoS).



Then evaluation at a larger scale on the Froggy supercomputer (CIMENT center/GRICAD, Grenoble)

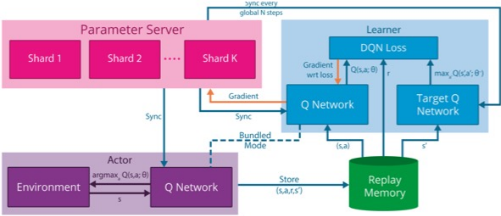
Addressing the HPC/Big-Data/IA Convergence[4]

Gathering teams from HPC, Big Data, and Machine Learning to work on the convergence of

- Smart Infrastructure and resource management
- HPC acceleration for AI and Big Data
- AI/Big Data analytics for large scale scientific simulations

Current work

- Molecular dynamics trajectory analysis with deep learning
  - Dimension reduction through DL, accelerating MD simulation coupling HPC simulation and DL
- Flink/Spark stream processing for in-transit on-line analysis of parallel simulation outputs
- Shallow Learning
  - Accelerating Scikit-Learn with task-based progamming (Dask, StarPU)
- Deep Learning
  - TensorFlow graph scheduling for efficient parallel executions
  - Linear algebra and tensors for large scale machine learning
  - Large scale parallel deep reinforcement learning



---

[4] https://project.inria.fr/hpcbigdata/

An Orchestration Syst. for IoT Applications in Fog Environment

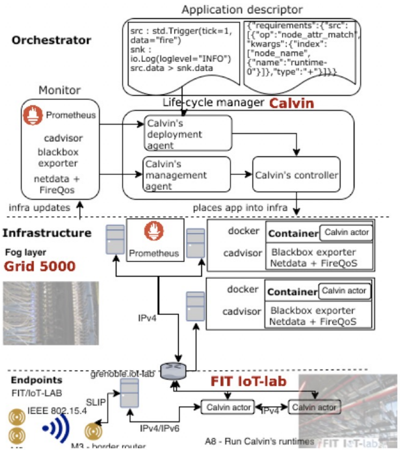
Objective: Design a Optimized Fog Service Provisioning strategy (O-FSP) and validate it on a real infrastructure

Contributions

- Design and implementation of FITOR, an orchestration framework for the automation of the deployment, the scalability management, and migration of micro-service based IoT applications
- Design of a provisioning solution for IoT applications that optimizes the placement and the composition of IoT components, while dealing with the heterogeneity of the underlying Fog infrastructure

Experiments

- Fog layer = 20 servers from Grid5000 which are part of the genepi cluster, Mist layer = 50 A8 nodes from IOTLab
- Use of a software stack made of open-source components (Calvin, Prometheus, Cadvisor, Blackbox exporter, Netdata)
- Experiments show that the O-FSP strategy makes the provisioning more effective and outperforms classical strategies in terms of: i) acceptance rate, ii) provisioning cost, and iii) resource usage

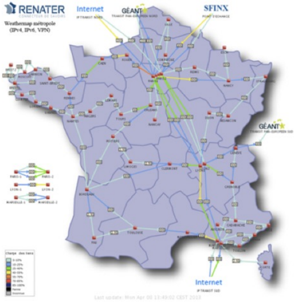


```
src : std.Trigger(tick=1,
data="fire")
snk :
io.Log(loglevel="INFO")
src.data > snk.data
```

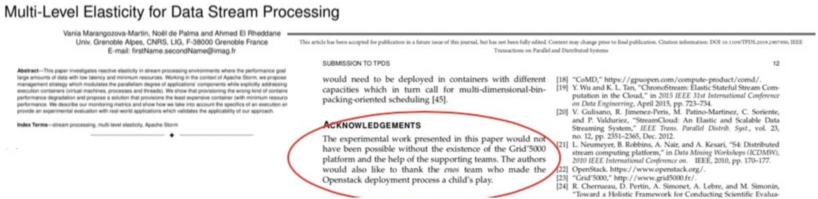
```
{"requirements":{"src":
[{"op":"node_attr_match",
"kwargs":{"index":
["node_name",
{"name":"runtime-
0"}]},"type":"+"}]}}
```

Toward a resource management system for Fog/Edge infras.

- **Inria Project Lab: Discovery**
  - Design a resource management system (a.k.a. a cloudkit) for Fog/Edge infrastructures
  - A four year project started in 2015 with Inria, Orange (and initially Renater)
  - Designing from scratch such a system cannot be envisioned (OpenStack 13 Millions of LOCs)
- **Contributions**
  - Implementation of a complete workflow to evaluate OpenStack WANWide scenarios
  - Evaluate OpenStack up to 1000 compute nodes (Grid'5000, oct 2016)
  - Evaluate OpenStack WANWide (impact of latency and throughput constraints) (oct 2017)
  - Evaluation of communication bus for Fog/Edge scenarios (May 2018)
  - Evaluation of database backends (NewSQL, NoSQL, etc. (May 2018)



Deploy a micro DC on each Network Point of Presence



Multi-Level Elasticity for Data Stream Processing

Vania Marangozova-Martin, Noël de Palma and Ahmed El Rheddane
Univ. Grenoble Alpes, CNRS, LIG, F-38000 Grenoble France
E-mail: firstName.secondName@imag.fr

Abstract—This paper investigates reactive elasticity in stream processing environments where the performance goal
large amounts of data with low latency and minimum resources. Working in the context of Apache Storm, we propose
management strategy which modulates the parallelism degree of applications' components while explicitly addressing
execution containers (virtual machines, processes and threads). We show that provisioning the wrong kind of containe
performance degradation and propose a solution that provisions the least expensive container (with minimum resourc
performance. We describe our monitoring metrics and show how we take into account the specifics of an execution en
provide an experimental evaluation with real-world applications which validates the applicability of our approach.

Index Terms—stream processing, multi-level elasticity, Apache Storm

◆

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPDS.2019.2907950, IEEE Transactions on Parallel and Distributed Systems

SUBMISSION TO TPDS 12

would need to be deployed in containers with different capacities which in turn call for multi-dimensional-bin-packing-oriented scheduling [45].

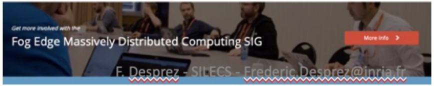
ACKNOWLEDGEMENTS

The experimental work presented in this paper would not have been possible without the existence of the Grid'5000 platform and the help of the supporting teams. The authors would also like to thank the *enos* team who made the Openstack deployment process a child's play.

[18] "CoMD," https://gpuopen.com/compute-product/comd/.
[19] Y. Wu and K. L. Tan, "ChronoStream: Elastic Stateful Stream Computation in the Cloud," in *2015 IEEE 31st International Conference on Data Engineering*, April 2015, pp. 723–734.
[20] V. Gulisano, R. Jimenez-Peris, M. Patino-Martinez, C. Soriente, and P. Valduriez, "StreamCloud: An Elastic and Scalable Data Streaming System," *IEEE Trans. Parallel Distrib. Syst.*, vol. 23, no. 12, pp. 2351–2365, Dec. 2012.
[21] L. Neumeyer, B. Robbins, A. Nair, and A. Kesari, "S4: Distributed stream computing platform," in *Data Mining Workshops (ICDMW), 2010 IEEE International Conference on*. IEEE, 2010, pp. 170–177.
[22] OpenStack. https://www.openstack.org/.
[23] "Grid'5000," http://www.grid5000.fr/.
[24] R. Cherrueau, D. Pertin, A. Simonet, A. Lebre, and M. Simonin, "Toward a Holistic Framework for Conducting Scientific Evalua-

Toward a resource management system for Fog/Edge infras.

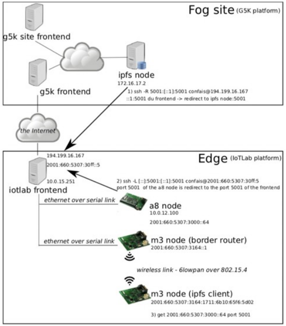
- **Inria Project Lab: Discovery (contd)**
  - The creation of a dedicated working group within the OpenStack community that deals with Fog/Edge challenges (now managed by the foundation with key actors such as ATT, Verizon, CISCO, China mobile etc.)
  - Several presentations / publications (see the DISCOVERY website)
  - France has the main academic actor in the worldwide community (Inria/IMT Atlantique) thanks to the G5K testbed in particular.
    - A leadership position
    - A strong expertise for experiments related to performance, scalability of OpenStack components (concrete actions with RedHat, ongoing actions with Huawei, etc.)



Open Infrastructure summit
Vancouver May 2018
(3000 participants)

Distributed Storage for Fog/Edge infrastructures

- **Objective**
  - Design of a storage system taking locality of edge resources into account
  - “Must-have”” Properties: data locality, network containment, mobility support, disconnected mode, scalability
- **Contributions**
  - Improving data locality by interconnecting Fog Scale-Out NAS systems with IPFS
  - Improving meta-data locality thanks to a tree based approach inspired by the DNS
- **Grid’5000 based experiments**
  - Evaluate the gain of using IPFS and scale out NAS systems for a 10 fog site infrastructure emulated on Grid’5000 (clients are deployed within Grid’5000).
  - Evaluate the gain of a tree-based meta-data management
  - ICFEC’2017 and GLOBECOM 2018
- **Grid’5000-FIT experiments**
  - Evaluate the penalties/side effects of using representative Fog clients (Fog servers are deployed on Grid’5000 and clients on the IoTLab platform)
  - Enabled us to identify several limitations (experiments using IoTlab and Grid’5000 are (currently) not easy to perform)

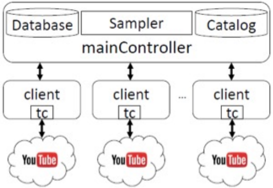


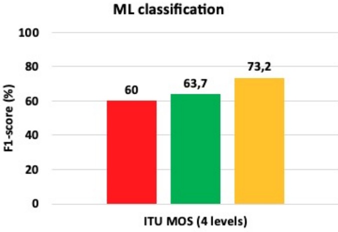
```
1) ssh -R 5001:[::1]:5001 confais@194.199.16.167
::1:5001 du frontend -> redirect to ipfs node:5001
2) ssh -L [::]:5001:[::1]:5001 confais@2001:660:5307:30ff:5
port 5001  of the a8 node is redirect to the port 5001 of the frontend
3) get 2001:660:5307:3000::64 port 5001
```

From Network Traffic Measurements to QoE for Internet Video

Problem solved: Estimation of QoE from encrypted video traces using network level measurements only)

- Play out a wide range of videos under realistic network conditions to build ML models (classification and regression) that predict the subjective MOS (Mean Opinion Score) based on the ITU P.1203 model along with the QoE metrics of startup delay, quality (spatial resolution) of playout and quality variations using only the underlying network Quality of Service (QoS) features
- **A diverse QoS-QoE dataset**
  - Around 100k unique video playouts from geographically distributed locations (Sophia Antipolis, Grenoble, Rennes, Nancy, Nantes) using compute resources from AWS, Grid5000, and R2lab platforms
- **Input features for ML:**
  - Network QoS (outband,inband, inband+chunks)
- **Output labels:**
  - App QoS (startup delay, resolution, quality switches) and ITU P.1203 MOS)





M. Khokhar, T. Ehlinger, C. Barakat. From Network Traffic Measurements to QoE for Internet Video. IFIP Networking Conference 2019,

An experiment's outline

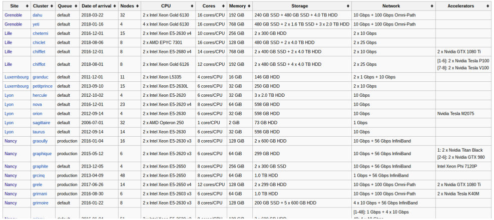
1. Discovering resources and selecting resources
2. Reconfiguring the resources to meet experimental needs
3. Monitoring experiments, extracting and analyzing data
4. Controlling experiments $\leadsto$ automation, reproducible research

Discovering and selecting resources

```
"processor": {
  "cache_l2": 8388608,
  "cache_l1": null,
  "model": "Intel Xeon",
  "instruction_set": "",
  "other_description": "",
  "version": "X3440",
  "vendor": "Intel",
  "cache_l1i": null,
  "cache_l1d": null,
  "clock_speed": 2530000000.0
},
"uid": "graphene-1",
"type": "node",
"architecture": {
  "platform_type": "x86_64",
  "smt_size": 4,
  "smp_size": 1
},
"main_memory": {
  "ram_size": 17179869184,
  "virtual_size": null
},
"storage_devices": [
  {
    "model": "Hitachi HDS72103",
    "size": 298023223876.953,
    "driver": "ahci",
    "interface": "SATA II",
    "rev": "JPFO",
    "device": "sda"
  }
],
```

- Describing resources ⇝ understand results
  - Covering nodes and network infrastructure
  - Machine-parsable format ⇝ scripts
  - Human-readable description on the web[5]
  - Archived (*State of testbed 6 months ago?*)
  - Verified
    - Avoid inaccuracies/errors ⇝ wrong results
    - Self-checking by nodes before each reservation
- Selecting resources
  - Complex queries using resource manager

```
oarsub -p "wattmeter='YES' and gpu='YES'"
oarsub -l "{cluster='a'}/nodes=1+
 {cluster='b' and eth10g='Y'}/nodes=2"
```



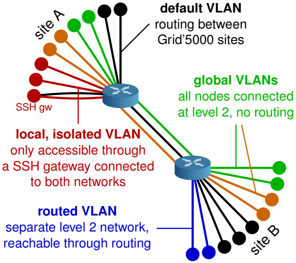
Site	Cluster	Queue	Date of arrival	Nodes	CPU	Cores	Memory	Storage	Network	Accelerators
Grenoble	dahu	default	2018-03-22	32	2 x Intel Xeon Gold 6130	16 cores/CPU	192 GiB	240 GB SSD + 480 GB SSD + 4.0 TB HDD	10 Gbps + 100 Gbps Omni-Path	
Grenoble	yeti	default	2018-01-16	4	4 x Intel Xeon Gold 6130	16 cores/CPU	768 GiB	480 GB SSD + 2 x 1.6 TB SSD + 3 x 2.0 TB HDD	10 Gbps + 100 Gbps Omni-Path	
Lille	chetemi	default	2016-12-01	15	2 x Intel Xeon E5-2630 v4	10 cores/CPU	256 GiB	2 x 300 GB HDD	2 x 10 Gbps	
Lille	chiclet	default	2018-08-06	8	2 x AMD EPYC 7301	16 cores/CPU	128 GiB	480 GB SSD + 2 x 4.0 TB HDD	2 x 25 Gbps	
Lille	chifflet	default	2016-12-01	8	2 x Intel Xeon E5-2680 v4	14 cores/CPU	768 GiB	2 x 400 GB SSD + 2 x 4.0 TB HDD	2 x 10 Gbps	2 x Nvidia GTX 1080 Ti
Lille	chifflot	default	2018-08-01	8	2 x Intel Xeon Gold 6126	12 cores/CPU	192 GiB	2 x 480 GB SSD + 4 x 4.0 TB HDD	2 x 25 Gbps	[1-6]: 2 x Nvidia Tesla P100 [7-8]: 2 x Nvidia Tesla V100
Luxembourg	granduc	default	2011-12-01	11	2 x Intel Xeon L5335	4 cores/CPU	16 GiB	146 GB HDD	2 x 1 Gbps + 10 Gbps	
Luxembourg	petitprince	default	2013-09-10	15	2 x Intel Xeon E5-2630L	6 cores/CPU	32 GiB	250 GB HDD	2 x 10 Gbps	
Lyon	hercule	default	2012-10-02	4	2 x Intel Xeon E5-2620	6 cores/CPU	32 GiB	3 x 2.0 TB HDD	10 Gbps	
Lyon	nova	default	2016-12-01	23	2 x Intel Xeon E5-2620 v4	8 cores/CPU	64 GiB	598 GB HDD	10 Gbps	
Lyon	orion	default	2012-09-14	4	2 x Intel Xeon E5-2630	6 cores/CPU	32 GiB	598 GB HDD	10 Gbps	Nvidia Tesla M2075
Lyon	sagittaire	default	2006-07-01	32	2 x AMD Opteron 250	1 core/CPU	2 GiB	73 GB HDD	1 Gbps	
Lyon	taurus	default	2012-09-14	14	2 x Intel Xeon E5-2630	6 cores/CPU	32 GiB	598 GB HDD	10 Gbps	
Nancy	graoully	production	2016-01-04	16	2 x Intel Xeon E5-2630 v3	8 cores/CPU	128 GiB	2 x 600 GB HDD	10 Gbps + 56 Gbps InfiniBand	
Nancy	graphique	production	2015-05-12	6	2 x Intel Xeon E5-2620 v3	6 cores/CPU	64 GiB	299 GB HDD	10 Gbps + 56 Gbps InfiniBand	1: 2 x Nvidia Titan Black [2-6]: 2 x Nvidia GTX 980
Nancy	graphite	default	2013-12-05	4	2 x Intel Xeon E5-2650	8 cores/CPU	256 GiB	2 x 300 GB SSD	10 Gbps + 56 Gbps InfiniBand	Intel Xeon Phi 7120P
Nancy	grcinq	production	2013-04-09	48	2 x Intel Xeon E5-2650	8 cores/CPU	64 GiB	1.0 TB HDD	1 Gbps + 56 Gbps InfiniBand	
Nancy	grele	production	2017-06-26	14	2 x Intel Xeon E5-2650 v4	12 cores/CPU	128 GiB	2 x 299 GB HDD	10 Gbps + 100 Gbps Omni-Path	2 x Nvidia GTX 1080 Ti
Nancy	grimani	production	2016-08-30	6	2 x Intel Xeon E5-2603 v3	6 cores/CPU	64 GiB	1.0 TB HDD	10 Gbps + 100 Gbps Omni-Path	2 x Nvidia Tesla K40M
Nancy	grimoire	default	2016-01-22	8	2 x Intel Xeon E5-2630 v3	8 cores/CPU	128 GiB	200 GB SSD + 5 x 600 GB HDD	4 x 10 Gbps + 56 Gbps InfiniBand	
									[1-48]: 1 Gbps + 4 x 10 Gbps	

---

[5] https://www.grid5000.fr/w/Hardware

Reconfiguring resources

- Operating System reconfiguration with Kadeploy:
  - Provides a *Hardware-as-a-Service* cloud infrastructure
  - Enable users to deploy their own software stack & get *root* access
  - **Scalable, efficient, reliable and flexible**: **200 nodes deployed in ~5 minutes**
- Customize networking environment with KaVLAN
  - Protect the testbed from experiments (Grid/Cloud middlewares)
  - Avoid network pollution
  - Create custom topologies
  - By reconfiguring VLANS $\leadsto$ almost no overhead



default VLAN
routing between
Grid'5000 sites

global VLANs
all nodes connected
at level 2, no routing

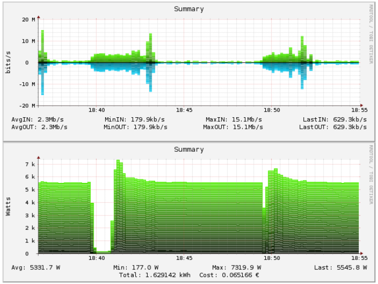
local, isolated VLAN
only accessible through
a SSH gateway connected
to both networks

routed VLAN
separate level 2 network,
reachable through routing

Monitoring experiments

Goal: enable users to understand what happens during their experiment

- System-level probes (usage of CPU, memory, disk, with Ganglia)
- Infrastructure-level probes: Kwapi
  - Network, power consumption
  - Captured at high frequency (≈1 Hz)
  - Live visualization
  - REST API
  - Long-term storage



Controlling experiments

- Legacy way of performing experiments: shell commands
  - ☹ time-consuming
  - ☹ error-prone
  - ☹ details tend to be forgotten over time
- Promising solution: **automation of experiments**
  $\leadsto$ Executable description of experiments
  $\leadsto$ Reproducible research
- Support from the testbed: Grid'5000 RESTful API
  (Resource selection, reservation, deployment, monitoring)
- Several higher-level tools to help automate experiments
  Execo, Python-Grid5000 (Python), Ruby-cute (Ruby)
  `https://www.grid5000.fr/w/Grid5000:Software`

Users and publications

Year	2010	2011	2012	2013	2014	2015	2016	2017	2018
Active users	564	553	592	514	528	458	573	600	564
Publications	154	141	101	134	106	143	122	151	127
PhD & HDR	14	20	9	27	24	30	27	23	22
Usage rate	50%	56%	58%	63%	63%	63%	55%	53%	70%

- 1313 active users over the last 3 years
- 3769 active users since 2003
- 2007 publications that benefited from Grid'5000 in our **HAL collection**[6]
  - Computer Science: 96%, Mathematics: 2.4%, Physics: 2.4%
  - Since 2015: LORIA: 23%, IRISA: 23%, LIG: 19%, LIP: 13%, LS2N: 13%, CRISTAL: 5%, LIRMM: 5%, LIP6: 3%

---

[6] `https://hal.archives-ouvertes.fr/GRID5000`

Organization and governance

- **Director** – Frédéric Desprez
- ***Bureau*** (6 members: FD, LN, Christian Perez, Adrien Lebre, Laurent Lefevre, David Margery)
- ***Comité des responsables de sites***
- **Technical Director** – Lucas Nussbaum
  - Technical team
- **Architects committee** (6 members)
- ***Conseil de groupement***
  - Inria, CNRS, RENATER, CEA, CPU, CDEFI, IMT (≈ Allistène + RENATER)
- ***Conseil scientifique***
  - 10 members

institutional and scientific steering

technical steering

advisory and evaluation bodies

Technical organization

- Distributed infrastructure, but managed by a single distributed team
  - Strong coherence and coordination between sites
- Current composition: 8.13 full-time engineers
  - Inria: 5.91 (perm: 0.86, CDD: 5.1), CNRS: 1.02 (perm: 1.02), U. Rennes: 0.6 (perm: 0.6), IMT Atlantique: 0.4 (CDD: 0.4), U. Lorraine: 0.2 (perm: 0.2)

Conclusions

- An advanced and established infrastructure for the *data-center* facets of Computer Science
  - Large-scale, distributed
  - Shared (many involved laboratories and institutions)
  - Designed for reconfigurability, observability, reproducible research
- Future: SILECS
  - SILECS Infrastructure for Large-scale Experimental Computer Science
  - On the fondations of Grid'5000 and FIT (IoT-Lab, CorteXLab, R2Lab, etc.)
  - Experiment on a single infrastructure, from edge to cloud