GreenDays@Lille

Energy efficient mapping of virtual machines

Thursday 28th November 2013

Violaine Villebonnet

Supervisor :
Georges DA COSTA

Current approaches for energy savings in cloud

Several actions at different levels :

Infrastructure	*Localization, Architecture, Cooling System*
Machines	*Power off / Suspend unused machines*
Processor	*DVFS Dynamic Voltage and Frequency Scaling*
Middleware	*Virtual machines mapping*
Applications	*Optimizations inside the code*

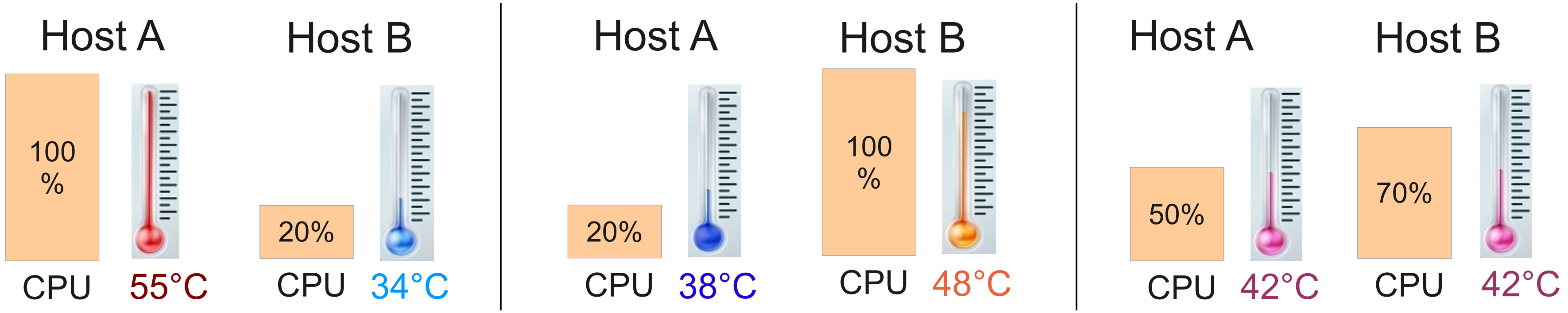
Different approach

- Current approaches are limited

  Decisions only made with data from the system : CPU, Network, Memory utilizations

→ New approach

Taking into account the hardware by monitoring the temperatures



Host A
100 %
CPU 55°C

Host B
20%
CPU 34°C

Host A
20%
CPU 38°C

Host B
100 %
CPU 48°C

Host A
50%
CPU 42°C

Host B
70%
CPU 42°C

Basic consolidation

Consolidation + some thermal awareness

Thermal management: temperature threshold

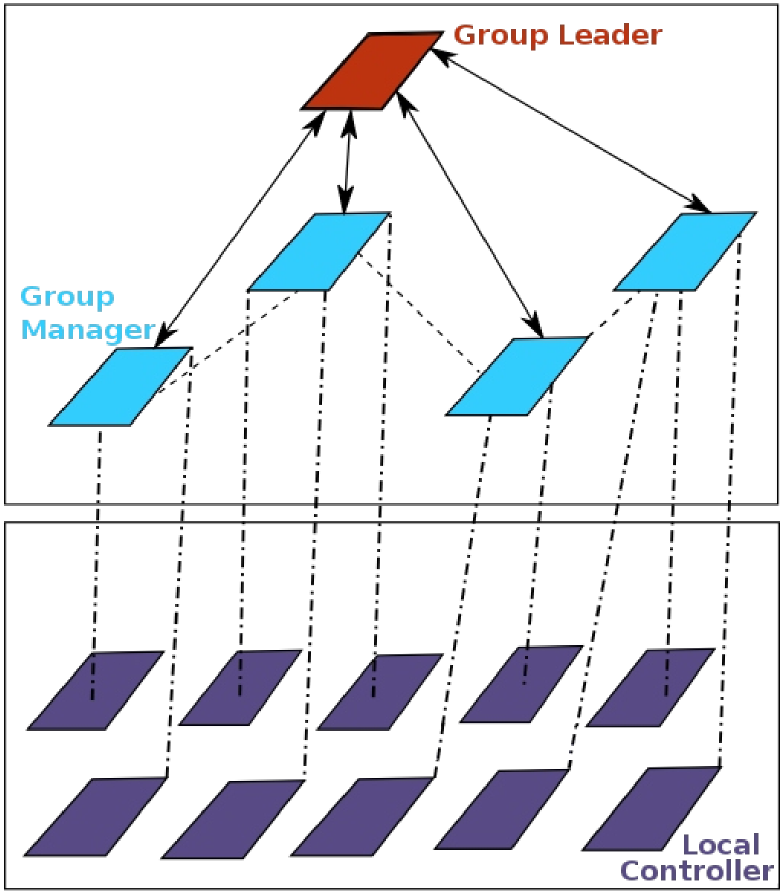
Energy efficient mapping of virtual machines
+ Thermal aware

- Add thermal aware features to a cloud manager
- Implement new reconfiguration algorithms
- Tests in real conditions on Grid'5000 platform
- Study the behaviour and then be able to propose algorithms that best fit with reality

Snooze

Cloud manager developped by Eugen Feller during his PhD at INRIA Rennes

Hierarchical architecture

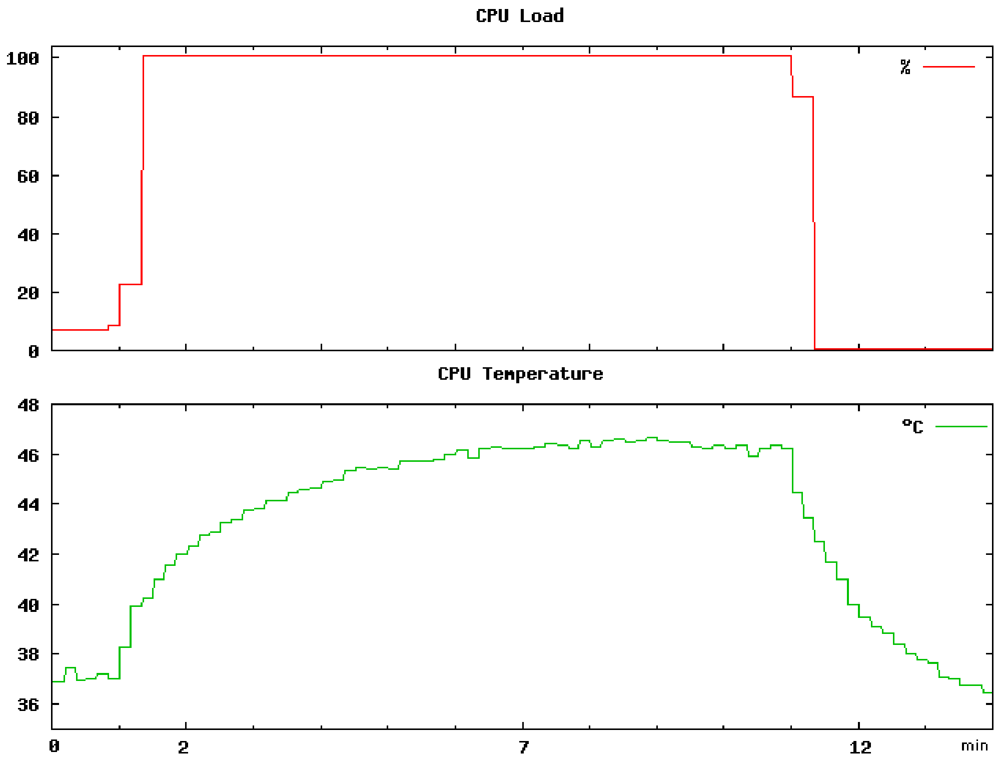


Particular features

- Self Configuration
- Self Healing
- Overload and Underload Detection
- Relocation and Consolidation
- Power Management
- Live Migration
- Generic Scheduler
- Open Source

Temperature monitoring

Motivations



Link between temperature and CPU load

- Avoid heat points
- Avoid peak consumption due to the cooling system : Fans and Air Conditioning

→ Reduce the energy consumption of the whole datacenter

Temperature monitoring

Integration in Snooze

- Package HARDWARE MONITORING BY lm-sensors to get temperatures from sensors
- Distributed monitoring system Ganglia
- Module python which gets temperature each second
- Each Local Controller in Snooze gets its own values, and send them to the Group Manager
- Group Manager receives monitoring data from all the Local Controller it supervises

Relocation algorithm modifications

- Java project, many packages, 450 classes
- Deployment, testing and bug corrections in collaboration with Matthieu Simonin, Research engineer at INRIA Rennes
- Features useful for my project :
  - Anomaly detection
  - Live migrations
  - Power On / Off of the nodes

Relocation algorithm

Goals

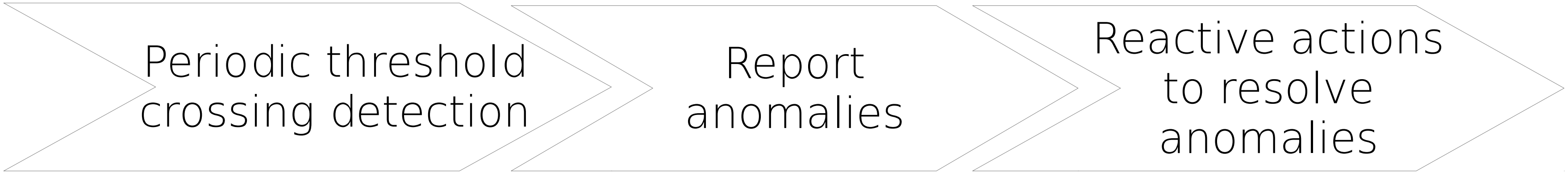
- Reduced energy consumption
- No crossing of a maximal temperature threshold
- Minimal number of used hosts
- Minimal number of migrations

→ Problem: Paradoxal goals

Dispatching not to overheat

≠ Consolidate to optimize the use of resources

Relocation algorithm



Periodic threshold crossing detection

Report anomalies

Reactive actions to resolve anomalies

2 states of anomaly :

- OVERHEAT (temperature > max threshold)

VM migration to the coldest node

- UNDERLOAD (CPU load < min threshold)

VM migrations to the coldest node, that already hosts VM, and not in OVERHEAT state

⚠ Priority in anomaly resolution : OVERHEAT first !

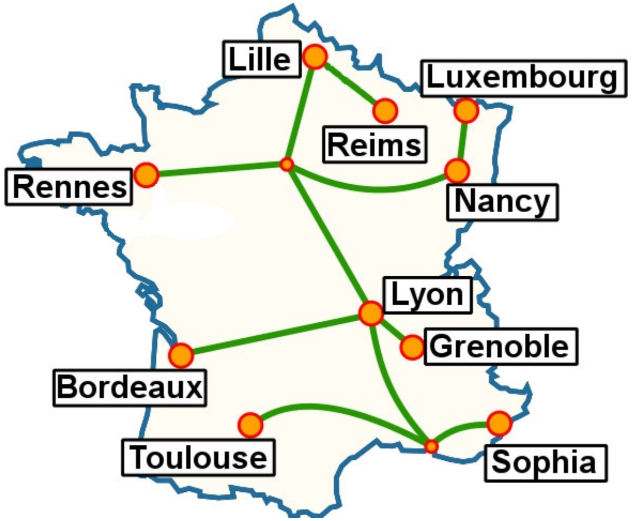
Variants of the algorithm

- Different possibilities for :
  - Threshold crossing detection
  - Choose the destination host of a migration
- Historic list of the last temperature and CPU load values
  - Decision on the average of the last values
  - Decision only on the last value

Experiments on Grid'5000

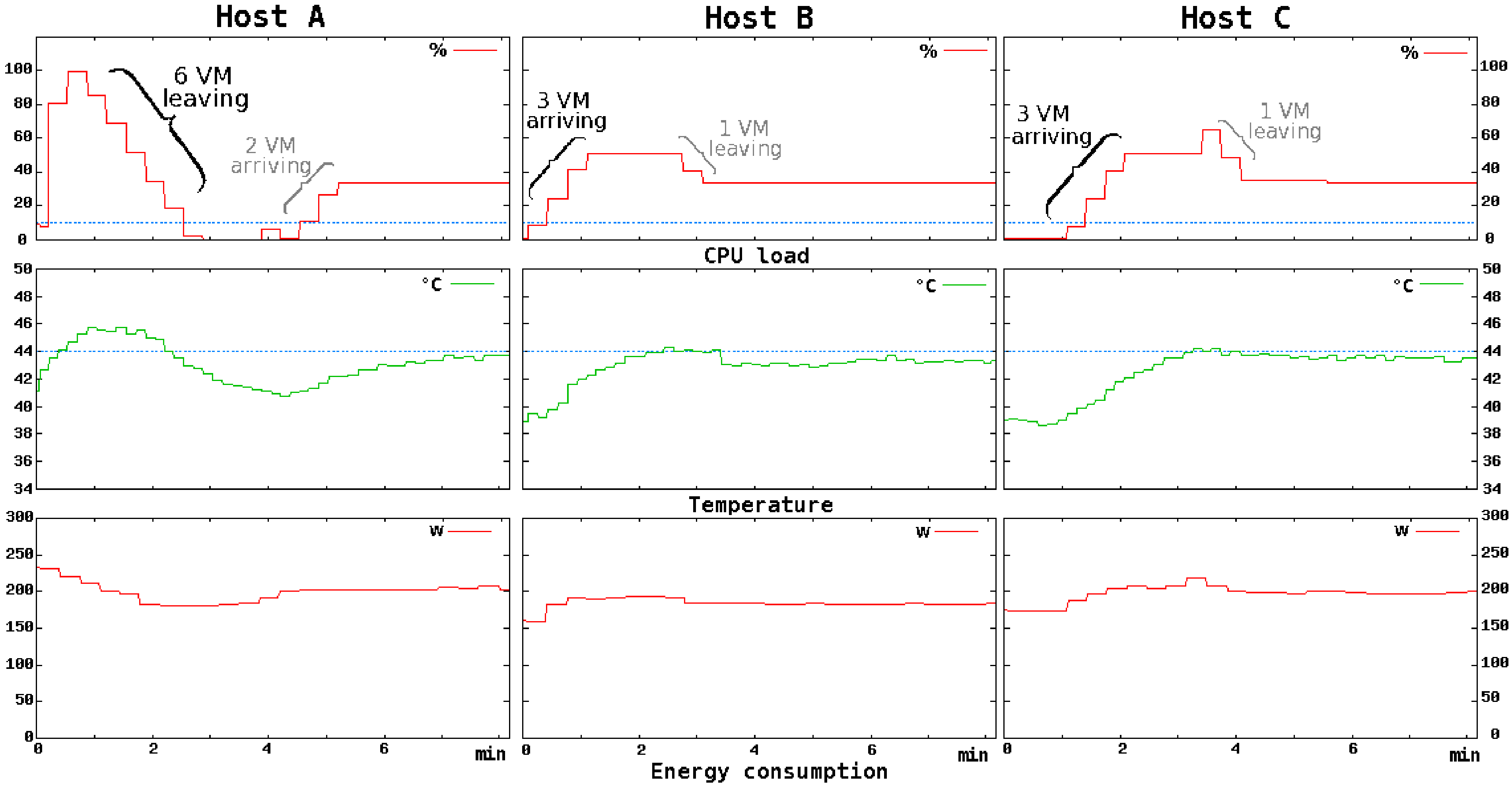
- Deployment on Rennes site – Cluster Parapluie
- Scripts to run experiments, extract data from log files, and then calculate the metrics :

  - Number of migrations
  - Energy consumption
  - Time spent in each 3 states

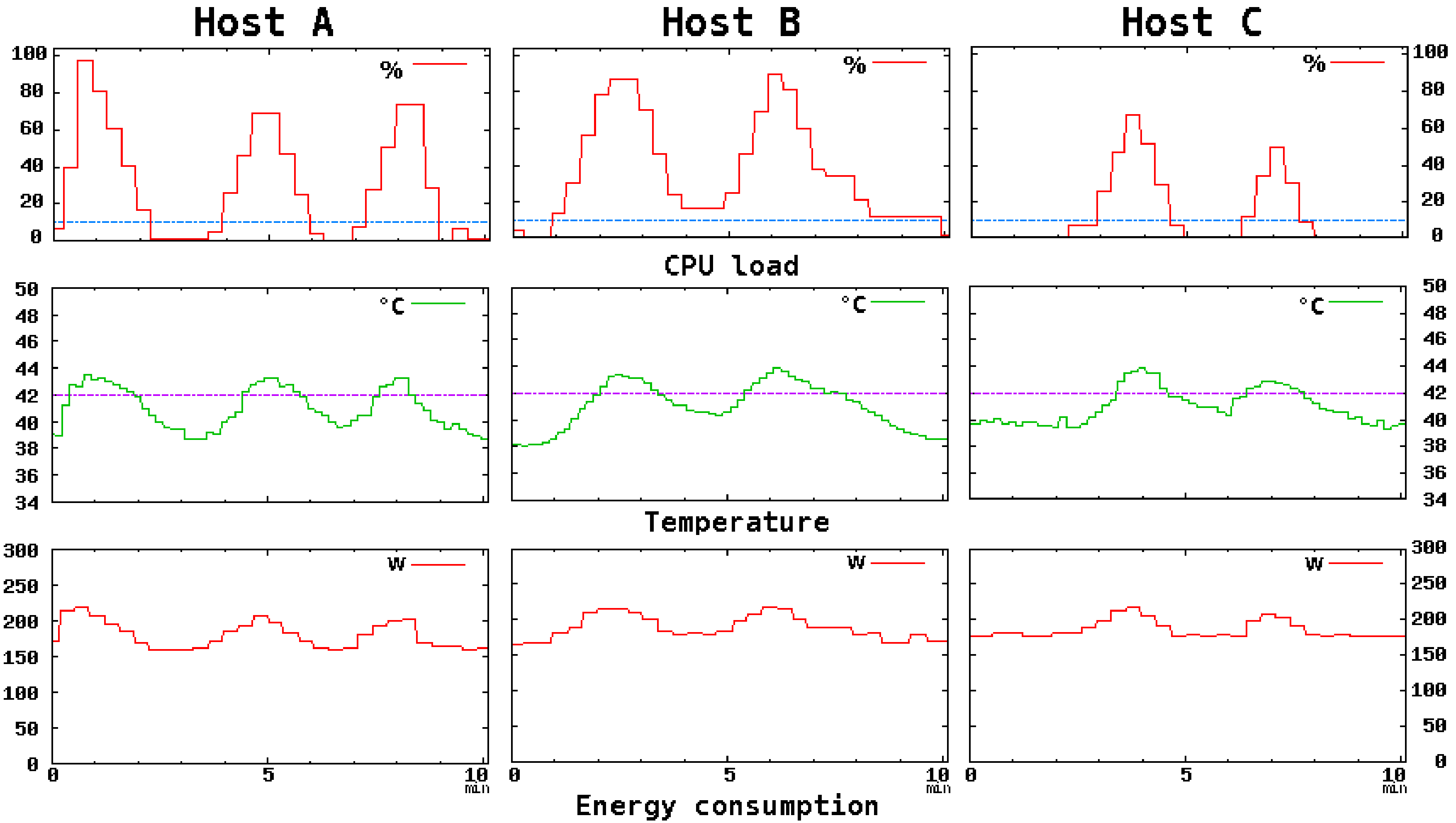


Results – Example Stable

3 hosts (1 processor 12 cores) – 6 virtual machines (2 CPU) cpuburn



Results – Example PingPong

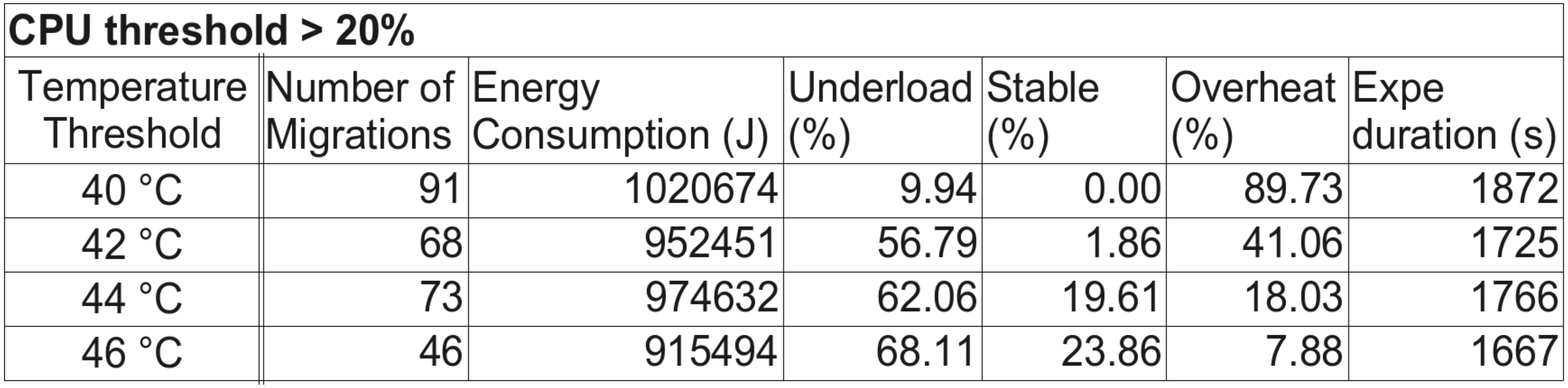


+ : Capping the temperatures and energy consumption

- : Unstable system, Degraded QoS, Overhead due to migrations

Results – Threshold variations

Variation of temperature max threshold



CPU threshold > 20%

Temperature Threshold	Number of Migrations	Energy Consumption (J)	Underload (%)	Stable (%)	Overheat (%)	Expe duration (s)
40 °C	91	1020674	9.94	0.00	89.73	1872
42 °C	68	952451	56.79	1.86	41.06	1725
44 °C	73	974632	62.06	19.61	18.03	1766
46 °C	46	915494	68.11	23.86	7.88	1667

Higher is the temperature threshold, lower is the energy consumption of the servers

⚠ But

Energy Consumption of the cooling system is not taken into account here !

Conclusion & Improvements

Conclusion

- Successful integration of new relocation algorithms in Snooze
- Snooze is now able to monitore temperature

  and even other measures with Ganglia
- Experimental platform is operational and possibility to compare algorithms by metrics calculation

Possible improvements

- Module to predict temperature
- List of last states of the nodes
- Variable thresholds
- Add a delay between migrations
- Energy consumption of the cooling system

Any further questions, contact me : Violaine.Villebonnet@irit.fr