This research is supported by the french Agence Nationale de la Recherche with the ANR-08-SEGI-017
Domain 1: Self-sizing and green PaaS

Goal: improve the platform efficiency
- Performance
- Energy cost

Challenges
- Cluster growth/shrink capabilities according to workload
- Data center multi-tenancy with limited capacities requires arbitration policies
- Data center placement for minimizing the energy consumption

Use case: Orange Infrastructure
- Large-scale data centers (+40,000 x86 servers)
- JavaEE JOnAS middleware stack: +250 applications, +1,000 application server instances
- Static server consolidation (12/1) through virtualization: Utilization level increased from 20% to 75%

Technologies
- OW2: JOnAS, Jasmine, ProActive, Clif, Entropy
- 3rd party: Xen, KVM + Collaboration in progress: OpenStack, OpenNebula, Ubuntu
- Technology to be developed:

Partners
- Bull, Ecole Mines Nantes, Inria, Orange, ActiveEon
- O-Engine, Iscas, Buaa, PKU, UCM, ...

Resources
- Collaborative Projects: SelfXL (ANR), 4caast (FP7), Easi-Clouds (ITEA2), Compatible One (FU10), Internetware (863)...
Self-sizing and green PaaS Architecture

PaaS
Linux

IaaS

Infrastructure

Cloud managers

Hypervisors

SelfXL AutoScaler

Apply Arbitration policies

ProActive

Scheduling

ProActive

Resourcing

Management of Physical / Resources

Scale Up / Down

Data provisioning

JASMINe

Autonomic Manager

Scaling

Apply Scaling policies

VM

Apache

JonAS

JavaEE/APP

1

VM

JonAS

VM

PSQL

VM

JonAS

Apply Scaling policies

VM

Apache

JonAS

VM

PSQL

VM

JonAS

VM

JonAS

JavaEE/APP

2

VM

Apache

JonAS

VM

PSQL

VM

JonAS

VM

JonAS

Apply Scaling policies

VM

Apache

JonAS

VM

PSQL

VM

JonAS

VM

JonAS

Apply Scaling policies

ProActive

Resourcing

PA VM

OpenNebula

Amazon EC2

Openstack

Infrastructure

KVM

vmware

Xen
SelfXL Automatic Scaler

**Goals**

- **Build elastic Java EE application**
  - Automatic scale-up
  - Automatic scale-down

- **Agnostic**
  - Not tightly coupled with a IaaS impl or a load-balancer impl
  - Extensible

- **Hybrid cloud support**
  - JavaEE cluster spanning multiple IaaS
  - Ex: dealing with private cloud overflow by acquiring resource on public cloud
Principles

Rules based decision module
- Nodes monitoring (maintain node state)
  - Heartbeat impl tru cpu load indicator
- Rate aggregation
  - Servlet indicator
- Scale-up triggering
  - PROVISIONNING_NEW_NODE
  - START_NEW_NODE
  - GET_INFO // @ip,...
  - ADD_PROBE
  - ADD_NEW_NODE_IN_CLUSTER
- Scale-down triggering
  - DISABLE_NODE_IN_CLUSTER
  - [WAITING_SESSION_ENDING]
  - STOP_NODE
  - REMOVE_NODE_FROM_CLUSTER
  - REMOVE_PROBE
  - DELETE_NODE
Architecture (control loop)

Service oriented solution
- IaaS interface
- Loadbalancer interface
Technologies

**Managed element**
- JOnAS

**Manager**
- JASMINe Monitoring infrastructure
  - JMX probe
  - EventSwitch
  - Drools
- ClusterScaler
  - OSGi/iPOJO
Monitoring

➔ JMX Interface (attributes)
- Pool nodes activity
- Total rate, average node rate
Error management

✿ Principle
  ▪ Put human in the loop

✿ JMX interface (operations)
  ▪ Void retry()
  ▪ Void ignore()
  ▪ Void cancel()
  ▪ Void setState()
Code, Status & Roadmap

**Source code available on OW2 SVN**
- `svn://svn.forge.objectweb.org/svnroot/jasmine/sandboxes/selfXL/selfxl-cluster-scaler`

**Status**
- IaaS
  - Cluster Daemon (start/stop JOnAS instances)
  - VMM (control JOnAS VM thru hypervisor)

**Roadmap**
- Add others IaaS drivers: jclouds, OCCI (ON),...
- Parallel nodes provisioning to cope with peak load
- Graphical console
Elastic JavaEE /clusterSample

ClusterScaler
1. Detect overload
2. scaleUp()
3. startNode()
4. rcfgLb()

Apache

Db

jonas1
jonas2
jonas3

JASMINe Agent

Cluster Daemon

SOAP
JMX

Local file update
Local control (start/stop)