A leading edge Cloud editor since 2007
Company

➢ Created from INRIA in October 2007

➢ Distribution, Parallelism, HPC, Grids, Cloud

➢ Develop solutions for many application domains:

   - Engineering, Bio Technologies, Finance, IT

➢ Our labels: DGA (French Dept of Defense), Pacte PME,

   - Initiative Achats de l’État pour PME innovantes

➢ Growth in France and worldwide
Mission

➢ Distribution, Parallelism, HPC, Grids, Cloud for DSI Infrastructure and Critical Business Applications

➢ Anticipate technological challenges
➢ Innovation as a service to our clients

➢ Provide strong expertise in key areas such as Distributed Systems and Cloud Computing

➢ Always be at the forefront of innovation through big investment in R&D
History

1999: INRIA research team creation
First developments on ProActive

2003: ProActive gets into the OW2 Open Source community

2007: Creation of Activeeon as an INRIA spin-off. Strong software IP

2010: R&D for Product Development

2012: First Production Customers and Partnerships

2014: Fund raising for commercial development
Innovative Technology

HPC Workflow & Parallelization

Scheduling & Orchestration

Cloud & Grid IaaS

GPU nodes
Products
ProActive Cloud Automation

- Automatic Deployment of Many Vms Applications
- Synchronization of Composite Services
- Monitoring (Physical, Virtual, Functional)
- Auto-Scaling
- Multi-Cloud Management with Bursting
- Reversibility (e.g. Vmware ⇔ OpenStack)
ProActive Cloud Automation

- elasticsearch
- hadoop
- nuxeo
- CLIF
- openstack
- Microsoft Hyper-V
- cloudstack
- VMware
- ERP
- NUMERGY
- Amazon Web Services
- Windows Azure
Cloud Automation
Cloud Automation

Scalable
Deployment
Updates
BackUp
Control and Monitoring
Life Cycle Management

Services
Brokering
Elasticity
Fault Management
Reversibility
Cloud Automation

Catalogue

Service Catalogue
- Software
- Platform
- Infrastructure

Management Console
- LifeCycle
- Supervision
- (Re) configuration

Added Value Services:
- Elasticity
- Fault-Tolerance
- Reversibility
- Contextualization of generic VMs
Workflows & Scheduling

Easily build New workflows
- Web applications
- Java SDK
- XML files

Create any type of task
- Scripts
- Executables
- Java

Advanced workflows
- Loops
- Conditions
- Variables
- Execution control
Workflows Orchestration

Generate reports
- Users
- Groups
- Usage

Plan your actions
- Unique
- Recurrent

Manage workflows execution
- Logs
- History
- Priority
- Pause, stop, etc.
Resource Management

All your resources in one view
- Physical machines
- Virtual machines
- Multi-OS
- Private clouds
- Public clouds

Secure your resources
- User rights
- Group rights
Monitor all your resources
- Physical machines
- Virtual machines
- Storage
- Network
- Applications

Expose your own metrics
- JMX
- API REST
- Flat file
So looping back … from Bottom to Top
Live Demonstration
1. Deployment of a Composite Service
2. Update of the Application
3. Applying a Security Patch
4. Life Cycle Management
Thanks for your attention!
Integration: Scilab, Matlab & R Applications

Dedicated resources

Static Policy
Clusters

Timing Policy
12/24
Desktops

Dynamic Workload Policy
EC2, Azure, HP Cloud, …

Amazon EC2

Scilab
Matlab
R Applications
With High Availability

- Availability
- Recovery
- Performance
- Backup
High Availability

What if?

- A node is lost
  - Failure is detected and the task running (if any) is restarted on another free node

- A crash happens while task are running
  - Using the database, running tasks are recovered from the database

The unit of work is the task
High Availability

On **node side**, failure is expected and failure management built in

- How to maintain computational capacity?
  - The node can reconnect to the Scheduler automatically

- The Scheduler can redeploy new nodes
  - Restart Down Nodes policy
On the **server side**, state is stored in:

- Database
- Real time replication
- MySQL Master-Master replication
- MySQL+DRBD Failover

- File system
- Real time replication
  - DRBD
  - SAN
High Availability

Via Active-Passive architecture

Primary installation

Failover installation
Tools involved

**Linux only**

- Failover
  - Corosync to communicate between primary/failover site
    - Pacemaker to coordinate service states
      - Also does IP failover via virtual IP
      - Stonith to avoid brain split issues
  - SSH Infrastructure & Restart Down Nodes policy for nodes deployment

- State Redundancy
  - DRBD
  - MySQL
Disaster Recovery

- To Backup
  - Database
    - Task/Job queue (running, pending, finished)
      - File system
      - Task logs
      - Monitoring history

- Recovery Point Objective
  - all running tasks are lost and will be restarted from scratch
  - all active user sessions are lost

- Recovery Time Objective
  - time to start ProActive Scheduler