How is built
a mOSAIC of Clouds

Dana PETCU

West University of Timisoara, Romania
Research Institute e–Austria Timisoara
http://web.info.uvt.ro/~petcu
Content

- About
- Problem
  - Use case scenario
  - Problem definition
- mOSAIC solutions
- Follow up
HPC @ UVT/IeAT

- IBM Blue Gene/P with 1024 cores, 13 TFlops
- Parallel computing since ’94, in:
  - Diff. Eqs & CFD
  - Non-linear eqs.
  - Evolutionary computing
  - Image processing
  - Expert systems
  - …
Grid @ UVT/IeAT

- Cluster 400 cores, 13 TFlops connected to EGEE/EGI, SEE-Grid/HP-SEE
- Grid computing since ’02
- Grid services for:
  - Earth observation: ESA – GiSHEO, EGEE- ESIP
  - Symbolic computing: SymGrid
  - Multi-objective optimization: DEMO-G
- Web/Grid/Cloud service management
  - Scheduling: OSyRiS
  - Workflow: SiLK
  - Composition: VISP
UVT/leAT @ FP6/FP7

- **HPC**
  - HP-SEE: SEE PRACE
  - ComplexHPC (COST)
  - HOST: HPC in Cloud (starts in Jan ’12)

- **Grid**
  - EGEE-II/-III/EGI
  - SEE-Grid-2/-SCI
  - SCIEnce (SymGrid)
  - RoGrid

- **Software services**
  - SPRERS – workshops e.g. Cloud (prj) 2nd WoSS
  - AVANTSSAR – security
  - VISP – composition

- **Cloud**
  - mOSAIC
mOSAIC motivation

Application Developer
Invented a new application. Want to run or share it? Go to the Cloud!
So many options!

No idea what to choose!
Pick “a” Cloud!
Write application for Cloud A

Deploy the application in Cloud A
After a while, a new feature is needed ... and only Cloud B has it...

Now you’ll need to refactor the application! :(
mOSAIC promise

Flying through the Clouds
WHAT IS THE PROBLEM?
The Sky
What is Sky computing?

Interconnection & provisioning of Cloud services from multiple domains

Federation of Clouds (InterClouds)

Multi-Cloud
The Sky problem: different APIs
Vendor agnostic code?

mOSAIC

Write once, deploy anywhere!
mOSAIC broker acquires resources
mOSAIC platform deploy resources
Application is shared
Application is scaling
What is mOSAIC?

**Open-source API and Platform for multiple Clouds**
Keywords

- Vendor agnostic API
- Platform as a Service
- Multi-Cloud and Cloud broker
- Multi-agent technologies
- Semantic processing
- Component-based long time running apps
- Auto-scaling [and self-adaptation]
- Event driven, asynchronous
Applications are built from scalable components
SOTA of portability

- At PaaS? NO!
  - *Use services from different Clouds*

- At IaaS? Ongoing task!
  - OCCI /OGF
  - UniCluster, OpenStack, jClouds, DeltaClouds...
  - Migration of VMs between Cloud providers: OpenNebula
  - Agreements between Cloud providers (federation)
  - Communications between Clouds
mOSAIC’s APIs

- **Application components**
- **Support for components**
- **For different languages**
- **mOSAIC middleware**
- **For same service type**

- **Component**
- **Cloudlet API**
- **Connector API**
- **Interoperability API**
- **Driver**
- **API**
Provisioning by a Cloud Agency
Use a common language

Use a Cloud Ontology
### Proof of the concept apps

<table>
<thead>
<tr>
<th>Type</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data intensive</td>
<td>Storage and data distribution in Earth Observation</td>
</tr>
<tr>
<td></td>
<td>Earth Observation mission reprocessing</td>
</tr>
<tr>
<td></td>
<td>Routine production of Earth Observation products</td>
</tr>
<tr>
<td></td>
<td>Fast data access for crisis situations</td>
</tr>
<tr>
<td></td>
<td>Distributed intelligent maintenance</td>
</tr>
<tr>
<td>Compute</td>
<td>Cloud-distributed parameter sweep</td>
</tr>
</tbody>
</table>
mOSAIC promises

- September 2011: API available
- September 2012: Platform available
- March 2013: Full software package
Current status

- Architecture & concepts & use cases
  - See project site – deliverables from last Feb
- API
  - Specifications to be released in September
  - First implementation in Java (Sept), next in Python (March)
  - Based on the “Cloudlet” notion
- Platform
  - Currently (non-integrated) components for agent system/provisioning, deploying, semantic processing
  - mOS – small OS with mOSAIC basic components
- Applications
  - Build from scratch: a Twitter watcher
  - Rebuilding legacy apps: GiSHEO training platform
Twitter watcher

- Components: Http gateway, servlet, fetcher, indexer, scavenger, message queues
- Use the drivers for open-source Cloud techs: Riak, RabittMQ, Eucalyptus
- Use classical appl development tools: Jetty
- Running example: mOS installed in 2 VMs, several components in these VMs
- Studies on the response time of the appl according to the no. users (appl scalability)
- Interested? Ask me for the video in breaks
Technical details in papers

- **Overview:** Building a Mosaic of Clouds, EuroPar 2010 - workshops, Springer, LNCS 6586, 529-536
- **API design – layers:** Towards a cross-platform Cloud API, CLOSER 2011, May 2011.
- **API design – interop:** Building an Interoperability API for Sky Computing, InterCloud, July 2011
- **SLA management:** A Cloud Agency for SLA Negotiation and Management, EuroPar ’10, LNCS 6586, 547-554
- **Patterns:** Identifying Cloud Computing Usage Patterns, 2010 IEEE Cluster,
- **Platform services:** Architecturing a Sky Computing Platform, ServiceWave 2010 - workshops, LNCS
- **Cloud ontology:** An Ontology for the Cloud in mOSAIC Cloud. In Cloud computing: methodology, system, and applications. CRC, Taylor & Francis group, 2011,
- **Cloud agency:** Agent based Cloud provisioning and management, CLOSER 2011, May 2011.
- **Use case:** From Grid Computing Towards Sky Computing. Case Study for Earth Observation, 10th CGW 2010,
- **Scheduling agents:** Self-Healing Distributed Scheduling Platform, CCGrid 2011, May 2011
mOSAIC partners

Second University of Naples, Italy
Institute e-Austria Timisoara, Romania
European Space Agency, France
Terradue SRL, Italy
AITIA International Informatics, Hungary
Tecnalia, Spain

www.mosaic-cloud.eu