iPlant Atmosphere: Gateway into the cloud infrastructure for the Plant Sciences

Edwin Skidmore
edwin@iplantcollaborative.org
What’s in a name?

“A cloud vendor... provides hardware, a software infrastructure (platform), or an application as a service to its customers. In the simplest scenario, a cloud vendor allows its customers to gain the capabilities of a simple server – albeit a virtual one – in which processing, network, and storage resources are controlled dynamically.”

-- “Cloud computing: A new business paradigm for biomedical information sharing”
Journal of Biomedical Informatics, 26 August 2009
Yet another cloud?

Technically savvy users have many options. But, existing clouds were limited in two ways:

1. No compelling platform for plant scientists’ needs, either tool developers or tool users
2. No compelling platform for non-technically savvy individuals
Example Integrative Use Case
Common use cases for Atmosphere

• Developing algorithms and tools and need a convenient (linux) environment
• Performing an analysis using existing tools provided by Atmosphere
• Performing an analysis using your own collection of tools
• Tool and analysis reproducibility
• Performing interactive pre and post processing of data for iPlant HPC systems
• Many, many more use cases
Why an iPlant Cloud, Tool Developers

- Ephemeral, on-demand environments
- Command-line access
- Some tools require user interactivity or a GUI
- Some tools don’t (always) require high performance environments
- Need to share their tools individually or as part of larger software stack (aka “Application”)
- Replica environments as DE and compute environments
Why an iPlant Cloud, Tool Users

• Ephemeral, on-demand environments (on-demand analyses)
• Command-line access
• Some analyses use tools that have user interactivity or GUI
• Some analyses don’t (always) require high performance environments
• Installing and sharing of larger, sometimes complicated software stacks
Why an iPlant Cloud, EOT

Education Outreach and Training (EOT) is a first-class citizen to the iPlant Collaborative.

Cloud-based VMs provide a way to provide preconfigured environments for workshops, classes, and other educational activities.
What is Atmosphere?

An open source cloud management interface and middleware built on top of common cloud engines that provides additional **security**, **scalability**, **usability**, and **integration** with iPlant cyberinfrastructure and allows users to create virtual machines in an on-demand fashion.

In addition to the technology, it is also the **support services** iPlant provides to help you use the cloud and perform science.
Atmosphere by the numbers

• Development began spring 2010 (technical evaluations and project plans much earlier)
• Public Beta: January 2011
• 325 users
• 8 “applications”, 91 images
• 2400+ virtual machines launched since beta
• 1 day 16:36 – mean lifetime of a vm
• 21 days – longest time for a running vm!
User Growth
Architecture

- Clients and other apps interface at the top (e.g. Web Frontend)
- Middleware
  - Authentication
  - Monitoring/auditing
  - Scheduling (Tasks, quota management & notifications)
  - Metadata Management
- Cloud engines and the VMs
  - integrated with iPlant’s compute environments and data services
  - Toolkits that assist with bundling and communication with higher-level components
Cloud Engine - Eucalyptus

• Considered “mature” at the time
  – Had recently went commercial (from an academic project)
  – Ubuntu adoption
  – Had many adopters

• Safe
  – AWS API compatible

• Support for OpenStack in the future
Middleware

• Python + Django
  • Fits nicely with python euca2ools
  • Django is a mature web framework
• Django
  • Not using the UI and CMS-related functions
  • ORM
  • Authentication
  • Django-celery for the scheduling and asynchronous tasks
Middleware

• Completely open web services APIs
• Integrates with common cloud engines, e.g. Eucalyptus and others soon
• Integrates with iPlant’s authentication services
• Notifications
• Resource Management (CPU, memory, and storage)
Middleware - Authentication

- Pluggable architecture for Authentication
- Currently, supports internal database and LDAP mechanisms
- Simplified token-based service, loosely based from OAuth
- Shibboleth in the near horizon
Middleware – Scheduling

Leverages Python Celery to perform asynchronous scheduling of tasks.

Uses include:

• Monitoring of the health of the cloud
• Compensating for deficiencies in the Eucalyptus, such as request throttling, failure states, etc. (boo!)
Very basic facilities store metadata in the internal database via tags and implicit attributes of the vm, such as user, image types, etc.
Web Front End

The web front end is a separate service component that makes calls to the Atmosphere middleware APIs.

Additional user interfaces can simply call the middleware directly.
Usability is key

Application market model: convenient access to preconfigured VMs

Privacy settings for “applications” (private, group, public)

An advanced management dashboard for the expert users
VM Toolkits (examples)

- **atmoCL** – provides command-line access within a vm to perform Atmosphere functions, such as managing EBS
- **atmo_init** – tool developers can perform post-boot configuration
- **image_request** – tool to initiate an imaging request
- **iPlant Data Store**
Data

• Integration with iPlant’s Data Store (iRODS)
  – FUSE-mounted within VM as a file system
  – Using command-line tools
  – GUI tools for the VM desktop (iDrop)

• EBS storage (ala Amazon)

• Traditional tools (Linux)
  – scp
  – rsync
Integration with iPlant’s Ecosystem

- Unified authentication
- VM images modeled after iPlant’s compute environments, include packages and dependencies
- Integration with iPlant’s unified data services
  - Discovery Environment
  - PhytoBISQUE
  - CoGe
## Web Services API

<table>
<thead>
<tr>
<th>Example Request</th>
<th>Example Response</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="http://atmosphere-url/launchInstance">http://atmosphere-url/launchInstance</a></td>
<td>{</td>
</tr>
<tr>
<td>params: instance_name=test&amp;image_id=emi-4F291974&amp;instance_size=m1.small&amp;auth_key=username_sample&amp;num_of_instances=1&amp;instance_description=testinstance&amp;instance_tags=testing</td>
<td></td>
</tr>
<tr>
<td>headers = {</td>
<td></td>
</tr>
<tr>
<td>&quot;Content-type&quot; : &quot;application/x-www-form-urlencoded&quot; ,</td>
<td></td>
</tr>
<tr>
<td>&quot;Accept&quot; : &quot;text/plain&quot; ,</td>
<td></td>
</tr>
<tr>
<td>&quot;X-Auth-User&quot; : yourusername ,</td>
<td></td>
</tr>
<tr>
<td>&quot;X-Auth-Token&quot; : auth_token ,</td>
<td></td>
</tr>
<tr>
<td>&quot;X-Api-Server&quot; : request_url ,</td>
<td></td>
</tr>
<tr>
<td>&quot;X-Api-Version&quot; : &quot;v1&quot;</td>
<td></td>
</tr>
<tr>
<td>}</td>
<td>}</td>
</tr>
<tr>
<td>result : {</td>
<td></td>
</tr>
<tr>
<td>&quot;code&quot; : &quot;success&quot; ,</td>
<td></td>
</tr>
<tr>
<td>&quot;meta&quot; : &quot;&quot; ,</td>
<td></td>
</tr>
<tr>
<td>&quot;value&quot; : &quot;i-3A720665&quot;</td>
<td></td>
</tr>
<tr>
<td>}</td>
<td></td>
</tr>
</tbody>
</table>


Mobile Companion

AtmoDroid here today

IOS/Apple app planned in the future
Future Work

• Features to facilitate analysis reproducibility and algorithm development (snapshotting, software-level reproducibility)
• Automated VM bundling and publishing
• On-demand application installation and configuration
• Integration with Apache Libcloud and/or OpenStack directly
• Multi-cloud support
Future Work (cont.)

- Better scheduling of resources (limitation of Eucalyptus)
- More high performing to data stored in iPlant Data Services
- Tighter integration with iPlant’s portfolio
- Shibboleth integration; federating authentication, such as InCommon
- Better metadata management
- Better sharing facilities
Learning Nuggets

• While usability is key, it is very difficult to make something simple. Unfortunately, simplicity is expected.
• Accept the fact that a few technologies are “robust”. Set expectations and be honest with your users.
• A little hand holding is a good thing. Too much is a dangerous one.
• Provide multiple ways to get work done, especially if a technology isn’t “robust”. Users learn to quickly adjust to the solution that suits their needs.
• Figure out the right combination of technology, (support) service, and documentation. Especially teaching users what cloud means...
We recognized that our analysis pipeline and workflow were far too complex!

So we put them into the cloud.

Let the clouds make your life easier.
# The Atmosphere Team

<table>
<thead>
<tr>
<th>Role</th>
<th>Members</th>
</tr>
</thead>
<tbody>
<tr>
<td>Core Services Team Lead</td>
<td>Edwin Skidmore</td>
</tr>
<tr>
<td>Developers</td>
<td>Seung-jin Kim, JMatt Peterson</td>
</tr>
<tr>
<td>Operations and Integration</td>
<td>Sangeeta Kuchimanchi, Andy Edmonds, Jose Salcedo</td>
</tr>
<tr>
<td>UI Graphic Designer</td>
<td>Monica Lent</td>
</tr>
<tr>
<td>Mobile Development</td>
<td>Steve Gregory</td>
</tr>
<tr>
<td>Documentation</td>
<td>Matthew Helmke, Shannon Oliver</td>
</tr>
</tbody>
</table>