GenApp Integration with Apache Airavata

1.INTRODUCTION

1.1 About GenApp

GenApp is a modular framework for multi-scale science computations. It enables one to generate applications for different platforms from the same source.

Basic Framework Structure of GenApp

Features

• GenApp provides user supplied executable modules, Wrapped with JSON definition of their input and output.
• The target languages for the user interface side are totally independent of the modules
• GenApp tools generate complete application code for an extensible set of target languages.
• Currently it provides application for HTML5/PHP, QT/C++.
• It can generate multiple targets simultaneously

Benefits
• can add features, create interfaces etc. with minimal effort
• changes to target code can be done without effecting the "base" logic, i.e. if we generate Qt4 GUI code and want to move to Qt6, our underlying modules don't need change.

1.2 About Apache Airavata

Apache Airavata is a software framework for executing and managing computational jobs and workflows on distributed computing resources including local clusters, supercomputers, national grids, academic and commercial clouds.

Components

• **XBaya Workflow Suite** - includes a GUI for workflow composition and monitoring.
• **Airavata Workflow Engine** - the de-facto workflow enacting engine. In future, it'll be Apache ODE.
• **GFac** - an application wrapper service that can be used to wrap command line-driven science applications.
• **WS-Messenger** - a publish-subscribe based message broker implemented on top of Apache Axis2 web services stack.
• **Registry-API** – A thick client registry API for Airavata to put and get documents.

2. BENEFITS FOR AIRAVATA AND GENAPP

2.1 Benefits to Airavata

Airavata will have the following benefits from this project:

• The new Airavata will be compatible with the older version and will get included in the source tree for GenApp. The codes written for HTML5/PHP will be fixed to make it functional with the new Airavata.
• Airavata submission will be possible from JAVA.

2.2 Benefits to GenApp

GenApp will have the following benefits from this project:
• The diversity of GenApp languages framework will increase with the addition of JAVA.
• Integration of JAVA framework of GenApp with Airavata will enable it to compose and execute workflows using Airavata's workflow engine.
• Submission from the new Airavata will be possible using HTML5/PHP which will provide GenApp the ability to run long running, non-interactive jobs on XSEDE.
• GenApp can then harness distributed computing resources including supercomputers and commercial clouds and more complex modules by chaining them.

3. DETAILED DESCRIPTION AND APPROACH

3.1. Providing interface for GenApp Java target

• The JAVA framework will be generated for GenApp to enable creation of JAVA-based applications.
• The integration will be reflected as **GenApp languages java**

3.2. Integration of GenApp with new Airavata to allow execution of long running jobs on clusters

• The following interfaces of GenApp will be fixed to properly integrated with new Airavata:
  • PHP/HTML5
  • JAVA
• Integration will be achieved using **Airavata Thrift API**, which has language interfaces in PHP and java. This will enable integration of GenApp interfaces with Airavata for executing long running computational jobs on clusters.
• This will involve creating calls in the wrapper based on certain variable settings in input JSON for targeting execution to Airavata
• PHP integration can be achieved by using the Airavata PHP sdk, which allows connection with Airavata to get a job launched.

3.3. Messaging Integration with new Airavata

• Apache Airavata includes WS Messenger, a web services based messaging system which guarantees scalable, reliable and efficient message delivery
- Airavata messaging provides a different layer of messages from jobs submitted to cluster resources (i.e. there is information available even when the executable is not running - such as 'failed to start' or 'waiting in queue to run' or others).
- GenApp messaging needs to fix / include getting updates back on .GenApp, messages sent by the executable to the users for live status updates, job submission messaging, among others.
- GenApp messaging systems will be integrated with the Airavata Messaging system to address the above mentioned needs.
- An implementation-Independent Component Programming Interface will be defined and this will be exposed through the API to GenApp.

4.DELIVERABLES

The deliverables to the community from this project include the following:

- GenApp Java target to generate java GUI from GenApp.
- Fix GenApp HTML5 interface integrated with new Airavata for executing long computational jobs.
- GenApp Java interface integrated with Airavata for executing long computational jobs.
- Documentation of the model for GenApp Java framework and also for integration with new Airavata.

5.TIMELINE

*Community bonding period*

**April 21 - April 27 (1 Week)**: Understanding and playing around with the GenApp code.

**April 28 - May 4 (1 Week)**: Understanding and playing around with the Airavata.

**May 5 – May 11 (1 Week)**: Deciding the implementation model for GenApp Java Framework.

**May 12 – May 18 (1 Week)**: Deciding the implementation model to integrate GenApp Java Framework with Airavata to execute long running jobs on various clusters.
Coding period

May 19 – June 1 (2 Weeks) : Fix code to get GenApp html5 working with the new version of Airavata.


June 24 – July 7 (2 Weeks) : Integration of GenApp Java Framework Interface with Airavata

June 8 – July 28 (3 Weeks) : Fix and Integrate GenApp with Airavata’s Messaging System

July 29 – August 18 (3 Weeks) : Documentation, Testing and fixing GENAPP code bugs.

6. MILESTONES

- **Sprint 1**: Bonding with community, learning more about GenApp and Airavata.
- **Sprint 2**: Finalizing implementation models for GenApp Java target.
- **Sprint 3**: Implementation model to integrate GenApp Java with Airavata to execute long running jobs.
- **Sprint 4**: Fix code to get GenApp html5 working with the newer version of Airavata.
- **Sprint 5**: Code for GenApp Java Target.
- **Sprint 6**: Integration of GenApp’s Java Interface with Airavata.
- **Sprint 7**: Fixing and Integration of GenApp with Airavata’s Messaging System.
- **Sprint 8**: Testing, Documentation and Evaluation.

7. PERSONAL INFORMATION AND BACKGROUND

**Name**: Abhishek Kapoor
**University**: Indian Institute of Technology (IIT), Kharagpur
**Major**: Pursuing B.Tech in Electrical Engineering (Fourth Year Student)
**Expected Graduation**: May 2015

**Technical Skills**: C, JAVA, HTML, CSS, JavaScript, jQuery, Ajax, php, Python, MySQL,
Experience: I have been involved in the following projects:

- **Aakash Development Laboratories (Guided By: Prof. Anupam Basu, IIT Kharagpur):** Developed a Java based search engine on Android platform for securing search results concerning authors publications and journals of any field by parsing HTML files.
- **Power Network Simulation (Guided By: Prof. A.K. Sinha, IIT Kharagpur):** Developed Java application for simulating power networks including simulation of generators, bus bars, Solar panels, auto transformers. Developed algorithms for feedback networks using interactive computing environment. Developed GUI for managing grid-lines.
- **Home Energy Management System (Guided By: Prof. A.K. Sinha, IIT Kharagpur):** Developed a java based android application for detecting FTDI chip and sending & receiving data from microcontroller.
- **Smart Device User Interface (Guided By: Mr. Ryuichi Inagaki, Works Applications):** Fixed around 10 bugs in company’s final product (groupware browser Application). Build Sorting feature for the browser application and enhanced Searching feature of the browser application.
- **Kshitij 2014 (Web Team Head):** Development of website for Kshitij 2014 and ACM_ICPC 2014. Also, Developed web based Android application for Kshitij 2014.
- **PolicyWonks:** Developing website for startup registered under the name policywonks.

Contact Information:
- **Email:** abhishek.kapoor52@gmail.com
- **Skype ID:** abhishek.kapoor52@gmail.com

8. AVAILABILITY

I am available full time during the entire GSoC 2015 period. I have no other commitments during this period

**Typical Working Hours:**

9 am to 12 pm, 2pm to 8 pm IST
(6 days a week, amounting to 54 hours per week.).

**Other Commitments:** None

**Any time in Summer where you will be away for more than 2 days:** None