A Lightweight Framework for Implementing Grid Portals
Jonas Lindeman and Göran Sandberg
LUNARC

- Lund Center for Scientific and Technical Computation
- 3 Production level clusters
  - Milleotto 1008 cores (x86_64)
  - Docenten 210 cores (x86_64)
  - Sigrid 100 cores (32-bit) part of SweGrid
- ~50-80 active users
- All machines have GRID-frontends
What are people running

- "Home grown" code compiles before each run 70% of the usersbase
- Packaged application 30%
  - MATLAB
  - MOLCAS – Quantum Chemistry code
  - ABAQUS – Finite Element Code
  - NASTRAN – Finite Element Code
  - FLUENT
  - …
“Packaged applications” on the Grid

- Good candidate for the Grid
- No need to transfer sources/binaries for execution, only input files
- Used by many (new/novice) users at Lunarc
- Published on the grid-infosystem using runtime-environments
- Main target of the Lunarc Application Portal
Providing user interfaces for Grid applications

• No “standard” way
• Must provide user interfaces for
  – The application
  – job submission, job control and monitoring
• For web-based portals
  – Libraries for web user interfaces
• For Window-based applications
  – Libraries for user interfaces
• … A lot of work …
The Lunarc Application Portal

- Provide an easy to use interface for a number of commonly used applications
  - Add new users of Lunarc resources
  - Simplify for existing users
- Provide a lightweight developer interface and installer for modifying, extending and setting up portals
  - Enable larger groups of users to create custom portals, without a lot of coding knowledge
The two faces of the Lunarc Application portal

User

Developer
The user side of LAP

- Provides the user with an easy to use interface for
  - Certificate management
  - Authentication
  - Job creation and management
  - Storage (Under development)
Welcome to the LUNARC application portal

LAP Version 0.8.0 (Currently undergoing revision... Please have patience.)
Copyright © 2004-2006 LUNARC, Lund University
Distributed under the GNU Public License version 2 or later
Written by: Jonas Lindemann

Credits:
Web application developed in WebWare for Python
Grid access through NorduGrid/ARC middleware (arcLib)
HyperText HTML code generation library by John A. (Andy) Dustman
JSDOMuBar by Toh Zhiqiang

User: /O=Grid/O=NorduGrid/OU=byggned/botten/CN=Jonas Lindemann
Proxy valid for: 23 hours, 58 minutes, 42 seconds
Welcome! - Mozilla Firefox

Welcome to the LUNARC application portal.

LAP Version 0.8.0 (Currently undergoing revision)
Copyright © 2004-2006 LUNARC, Lund University.
Distributed under the GNU Public License version 2.
Written by: Jonas Lindemann

Credits:

Web application developed in WebWare for PyGrid access through NorduGrid/ARC middleware.
HyperText HTML code generation library by John.
jeDOMenuBar by Toh Zhiqiang.

User: /=Grid/O=NorduGrid/OU=byggmek.Ith.se/CN=Jonas Lindemann
Proxy valid for: 23 hours, 58 minutes, 42 seconds
Create an ABAQUS job

Job name: AnAbaqusJob

Create
Edit Abaqus job

Abaqus settings
- Input file: C:\Documents and Settings\Browse...
- Current file: sample.inp
- License server: parkon.byggmek

Job settings
- CPU time (s): 15
- Job name: abaqus Job
- Email notification:

Modify | Back
Submitting job(s)...
ABAQUS VERSION 6.6-1
DATE 20-Oct-2006  TIME 15:13
For use at Lund University of Technology under academic license from ABAQUS, Inc.

BARREL VAULT (100 X 204 MESH)
SELF-WEIGHT
STEP 1  INCREMENT 1  STEP TIME 0.00

STEP 1  STATIC ANALYSIS

SELF-WEIGHT

FIXED TIME INCREMENTS
TIME INCREMENT IS 2.220E-16
TIME PERIOD IS 2.220E-16

LINEAR EQUATION SOLVER TYPE DIRECT SPARSE

THIS IS A LINEAR PERTURBATION STEP.
ALL LOADS ARE DEFINED AS CHANGE IN LOAD TO THE REFERENCE STATE
EXTRAPOLATION WILL NOT BE USED
CHARACTERISTIC ELEMENT LENGTH 2.01
DETAILS REGARDING ACTUAL SOLUTION WAVEFRONT REQUESTED
Implementation

- Web-application is implemented in Python using WebWare for Python
  - Integrated in the Apache Webserver
- Grid middleware
  - Advanced Resource Connector (ARC)
  - Arclib python interface to ARC
- Additional applications can be added using a Plugin-based architecture
The developer side of LAP

- Lightweight framework for implementing user interfaces for grid-enabled applications
  - Implemented in Python
  - Web user interface library
  - Base classes for job definitions and job user interfaces ➔ Plugin
  - Framework handles submission, management and monitoring
Implementing Plugins

**Task**
- `xrslAttributes`
- `attributes`
+ `setup()`
+ `clean()`

**JobPage**
+ `onCreateTask()` : <unspecified>
+ `onCreateJobEditForm()` : <unspecified>
+ `onValidateValues()`
+ `onAssignValues()`
+ `onHandleUploadedFile()`

---

**Plugin**

**CustomTask**
- Maintains

**CustomJobPage**
- Creates
- Displays

- Updates

**Job description**
- XRSL/JSDL

**Input files**
- Scriptfiles

---

**Graphical Representation**

Image depicting the structure and interactions of the components.
 Implementing a Plugin

• Implement the task class
  – Responsible for generating job definitions, input files and scripts
  – Job described in a neutral format from which XRSL/JDSL is generated automatically

• Implement the page class
  – Responsible for generating the user interface for the task class.

```python
def setup(self):
    # Get directory and attributes
    self.clearInputFiles()
    self.clearOutputFiles()
    taskDir = self.getDir()
    attribs = ...  # CUT ----
    xrslFile = Lap.Job.XRSLFile(self)
    xrslFile.setFilename(taskDir+"/job.xrsl")
    xrslFile.write()

def onCreateEditJobForm(self, task):
    form = Web.Ui.Form("editJobForm", ...)
    attribs = task.getAttributes()
    xrslAttribs = task.getXRSLAttributes()
    form.beginFieldSet("PovRay settings")
    form.addText("Initial frame", "initialFrame", attribs["initialFrame"], fieldType="int")
    form.addBreak()
    form.addText("End frame", "finalFrame", attribs["finalFrame"], fieldType="int")
    form.addBreak()
    form.addText("Image width", "imageWidth", attribs["imageWidth"], fieldType="int")
    form.addBreak()
    form.addText("Image height", "imageHeight", attribs["imageHeight"], fieldType="int")
    form.addBreak()
```
LAP installer

- Web application setup can be complicated
  - Many packages
  - Dependencies
  - Configuration files
  - ...
LAP installer

• Python based install script
  – Single file download
  – GUI/Command line interface
• Checks for prerequisites
• Downloads all necessary packages
• Configures and installs packages
• Installs and configures the portal
LAP installer

[bmjl@docenten lap-installer]$ ./setup-lap.py

---------------------------------------
Checking for prerequisites.
---------------------------------------

Checking for wget. Found.
Checking for tar. Found.
Checking for uname. Found.
Checking for gunzip. Found.
Checking for ARC client tools. Found.
Checking for Apache Extension Tool (APXS). Found.
Checking for ARC installation dir. Found.
Checking for Python based ARCLib. Found.

---------------------------------------
Lunarc Application Portal - Setup 0.1
---------------------------------------

Please set the installation options:

1. Portal application directory name : lap
2. Portal version : 0.8.0-20061026
3. Target installation directory : ./opt
4. WebWare version : 0.9.2

Enter option to change. (Enter = accept, 0 = quit) :
Ongoing projects

- **Lund**
  - User interface to the StarSim simulation package (Parallel) – Simulating large telescopes
  - User interface for the MOLCAS chemistry package
- **Uppsala**
  - User interface to a bioinformatics code
  - QTL analysis
StarSim
StarSim implementation process

- Structuring of existing code
- Parallelised version of code
- Setting up the code as a runtime-environment (STARSIM-X.Y)
- Implementation of LAP-plugin, generating input for the STARSIM-X.Y runtime-environment
What is currently being developed

- Job Submission Service (JSS) - WS
  - Middleware neutral job submission
  - Automatic job recovery and resubmission
- Simplified plugin definition (No coding)
- More interactive user interfaces
  - AJAX based using the ExtJS library
Concluding remarks

- The Lunarc Application Portal
  - Provides a common user interfaces for job submission, management, monitoring etc…
  - Is a lightweight framework (Python), enabling quick grid user interface development
  - Provides a plugin architecture enabling easy extensibility
  - Available to download
    - http://sourceforge.net/projects/laportal