Hands-on tutorial on usage the Kepler Scientific Workflow System

(including INDIGO-DataCloud extension)

Michał Konrad Owsiak (@mkowsiak)
Poznan Supercomputing and Networking Center
michal.owsiak@man.poznan.pl
e-Research Summer Hackfest
Agenda

• Kepler overview
• Main Kepler’s concepts
• Examples of Kepler application into science
• Hands on tutorials
After this workshop

You will understand:
• general idea of Kepler
• how to build simple workflow
• how to build control statements
• how to access Python from Kepler
• how Kepler contributes to INDIGO-DataCloud
Essential info about Kepler

• Scientific Workflow System
• Built on top of Ptolemy II framework
• Actor based computations
• Provides different models of execution (via Directors)
• Supported by NSF-funded team: UC Davis, UC Santa Barbara, UC San Diego
• Used across different fields of science: ecology, engineering, geology, physics, bioinformatics, biology, nuclear fusion, astrophysics, nanotechnology, climate change, etc.
Kepler application (GUI)

Looks nice, isn't it?!
Kepler - built-in components

- Around 500 components ready to use (~1,000,000 LOC)
  - mathematics: functions, operators, actors dedicated for R/Matlab
  - visualization: plotters, graphs, image displays
  - data manipulation: primitive types, strings, tables, records, GeoData, GIS, Data Access Protocol (DAP) 2.0
  - data access: support for Oracle, MySQL, MS Access, MS SQL, PostgreSQL

http://kepler-project.org
Kepler - built-in components

- **Web-Services:** SOAP, REST, XML processing
- **executing external code:** SSH, Cloud, Hadoop, UNICORE, gLite, Nimrod, Serpens
- **Kepler as batch job**
- **field specific modules:** BioKepler, GAMESS Input generator, Sensor Processing and Acquisition Network
- **provenance module:** you can track workflow's execution
Support for various languages

• main language: Java
  Write once, run anywhere!
• scripting: Groovy, JavaScript, Jython
• Python: external process, called via JNI
• C/C++: external process, can be turned into actor
• Fortran: can be called via JNI (or via JNA)

These guys help you call native code from Java
Various execution models

Execution models:
- Sequential Data Flow
- Dynamic Data Flow
- Process Network
- Continuous Time

These two are our main concern today!
Building workflows

Kepler provides a set of building blocks you can use:

- workflow
- director
- actor
- port
- relation
Building workflow

- workflow allows to compose data flow between elements
- Drag'n'Drop to put elements on canvas
- workflow can be started in GUI or from CLI
Building workflow
Building workflow

• Essential components are available from tool bar
Building components
Directors

Directors orchestrate the execution of workflow

- **SDF** - Sequential Data Flow
- **DDF** - Dynamic Data Flow
- **PN** - Process Network
- **CT** - Continuous Time
Actor - basic building block
Actor - basic building block

You can:
• use existing actors
• build your own actors (Java, Python, Groovy)
• import actors from other people
• install whole modules
Composite actors
Use-cases
Nuclear Fusion - EFDA ITM-TF/EURoFusion

- Integration of simulation tools for ITER and DEMO plasmas
- Kepler as workflow platform
- Numerous user codes: Fortran, C++, C, Python, Java, Matlab
- Common database for information exchange
- 90+ users, workflow developers and support team
- Custom build system: we extend default version of Kepler 2.5 with custom components
- Version tracking: workflows, actors, Kepler, results
Nuclear Fusion - EFDA ITM-TF/EUROfusion
## European Transport Simulator

### Workflow parameters

<table>
<thead>
<tr>
<th>General parameters:</th>
<th>Times:</th>
<th>ETS dimensions:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Transport:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Equilibrium:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Numerics:</td>
</tr>
</tbody>
</table>

### Start up
- Initializes the run, sets new shot and run numbers.
- Reads profiles of the input file from the database.
- Sets up the ETS grid.
- Interpolates starting profiles on the ETS grid.
- Initializes Equilibrium-Transport coupling.

### Time loop
- **RUN THE TIME EVOLUTION:**
  - Internally calls:
    - equilibrium
    - transport
    - source
    - impurities
    - solves transport equations
    - checks convergence
    - shoots pellet
- **Advance time**
  - **Stop time evolution**

### Finalizing the run
- Necessary post-processing, after the time evolution is finished, before closing the UAL connection.
- Close UAL connections and stop the run.
Nanotechnology

• A software suite - ANELLI has been designed with the aim of performing ring analysis

• Requirements:
  • easy way to add new actors
  • easy way to parametrise and trigger components
  • possibility to update codes live
  • running in distributed environment (parametric scans)
  • workflows should be transparent for users
ANELLI workflow
Astronomy

• Several workflows available (DIAPL, BF)
• Based on IVOA (International Virtual Observatory Alliance) standards
• The goal is to create modules that can be used by multiple groups of astronomers
  • support for VO services (SAMP, ds9, Topcat)
  • workflows should provide some level of interaction
  • integration with iRods (Integrated Rule-Oriented Data System)
  • integration of numerous codes
  • user friendly GUI
Astronomy - DIAPL workflow
INDIGO-DataCloud

• H2020 project - approved in January 2015
• 26 European partners, 11 European countries
• development of open source cloud platform for computing and data storage - DataCloud
• targets multi-disciplinary scientific communities
• deployable on hybrid Cloud infrastructures (public or private)
• answer to call raised by increasing demand for easy access to distributed Cloud/Grid resources
INDIGO-DC module for Kepler

- indigokepler - module that uses INDIGO-DC PaaS services
- First version of module *is available*; gradually extended
- Based on *FutureGateway API*
- Available via *Docker* image
- Sources are available in *GitHub*

indigoclient  - https://github.com/indigo-dc/indigoclient
indigokepler  - https://github.com/indigo-dc/indigokepler
docker-kepler  - https://github.com/indigo-dc/docker-kepler
Indigo - DataCloud

https://youtu.be/sEDBZFZjrvE

INDIGO-DataCloud RIA-653549
Questions?
Hands on part - Docker Kepler

"For the things we have to learn before we can do them, we learn by doing them."

- Aristotle

http://goo.gl/3Rurx7