



Dziedzinowo zorientowane
usługi i zasoby infrastruktury
PL-Grid dla wspomagania
Polskiej Nauki w Europejskiej
Przestrzeni Badawczej

Tools, methods and services enhancing the usage of the Kepler-based scientific workflow framework

Presenter: Marcin Płóciennik

Poznan Supercomputing and Networking Center, Poland

ICCS 2014, Crains, 10.06.2014

- Use cases: Nuclear Fusion(EFDA ITM), Nanotechnology, Astronomy,
- Tools:
 - Actor generators
 - C
 - Binding physics code: FC2K, WS2K, HPC2K (ITM)
 - Central Kepler installation (ITM)
 - Provenance enabled monitoring and profiling (ITM)
 - Sharing workflows
 - GUI for running workflows in distributed environment

Use cases



- EFDA ITM-TF work on validated suite of simulation tools for ITER and DEMO plasmas
- Whole simulation platform that allow cross validation between different fusion devices (Kepler as w-f engine)
- Individual codes in Fortran, C++, C, Python, Java and also Matlab
- Many very complex multi level workflows has been developed in Kepler: European Transport Solver (ETS), Turbulence-transport, Equilibrium reconstruction and MHD
 - ETS: the workflow integrates more than 1000 entries among which there are more than 150 actors and has multiple layer structure, which goes up to 15 levels deep, a number of convergence and time loops implemented at different levels of the workflow
- the data structure based on standardized generic objects, called Consistent Physical Objects (CPO), and the Universal Access Layer (UAL) for these data objects
- 90+ users- workflow developers and support team

European Transport Simulator

Workflow parameters



General parameters:

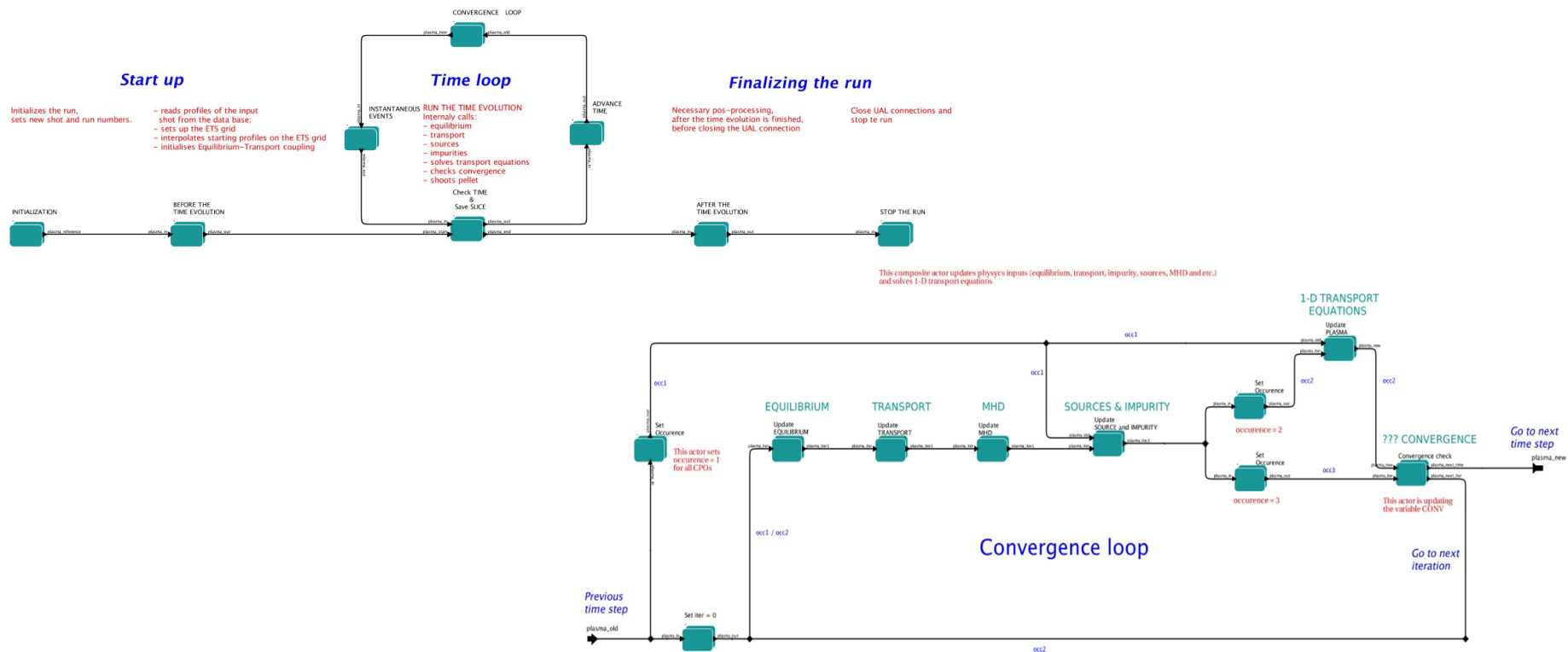
- USER: kalupin
- machine: test
- shot_in: 77922
- run_in: 2
- shot_out: 77922
- run_out: 4

Times:

- tbegin: 47.7
- tend: 48

ETS dimensions:

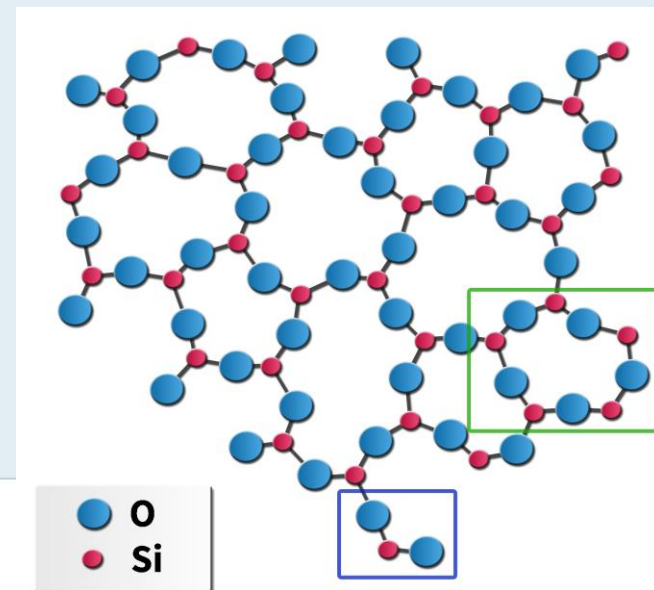
- TRANSPORT:
 - NRHO: 100
- EQUILIBRIUM:
 - NPSI: 100
 - NEQ_DIM1: 100
 - NEQ_DIM2: 100
 - NEQ_MAX_NPOINTS: 150
- NUMERICS:
 - NUMERICAL_SOLVER: 4



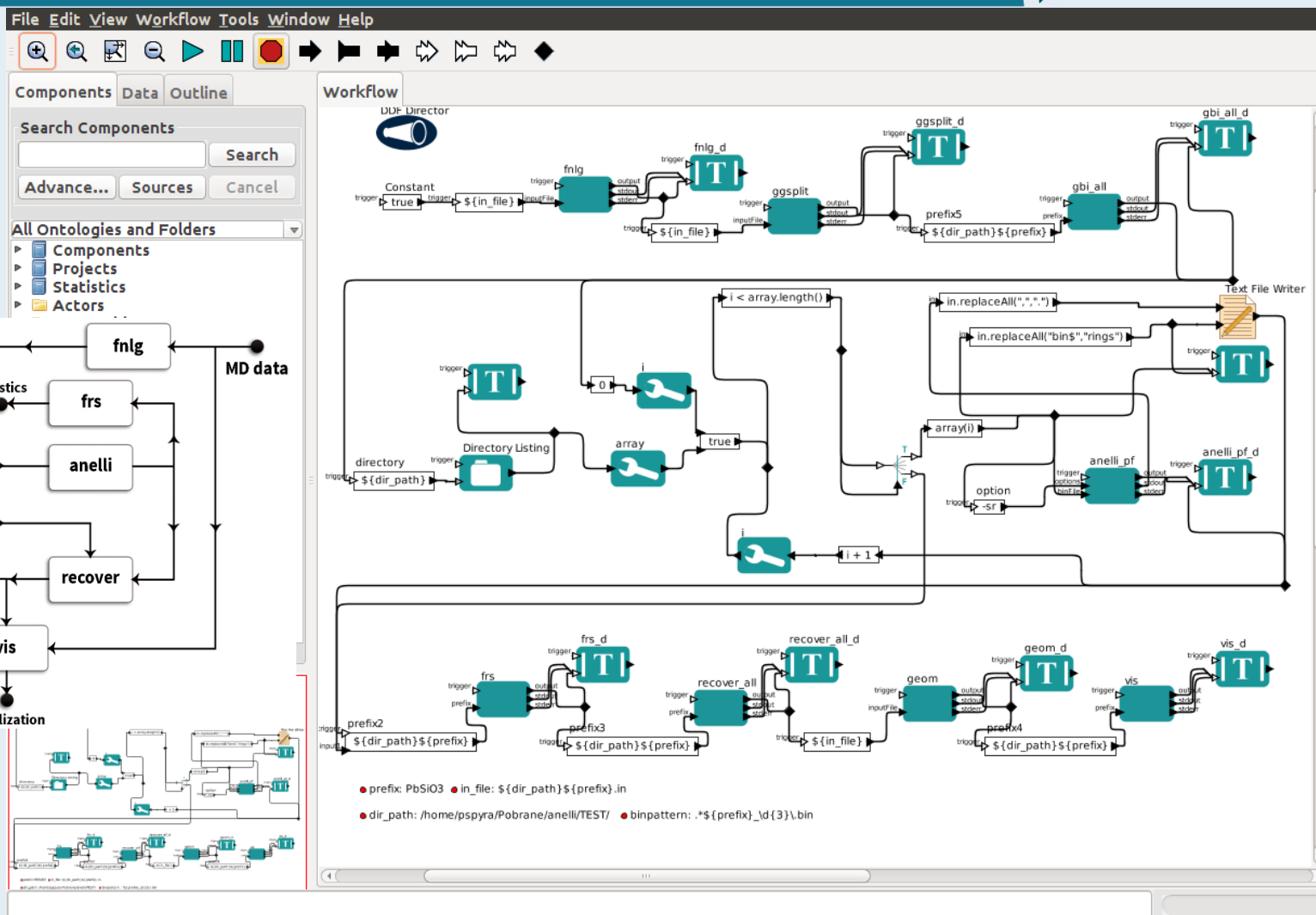
- Automatically include the physics codes written in different languages
- Run codes in the context of different computational resources,
- Management of the platform releases (including customised Kepler),
- Maintenance and sharing of the physics actors
- User friendly interface for run configuration
- Easy exchange of modules of the same kind, in order to optimize the physics complexity versus the performance (CPU time)

- Many issues solved, but still some challenges are ahead
 - How to provide user friendly interface for a large sets of Kepler workflow parameters?
 - Often the exchange of actors are done at the level of the composite actors, which already includes a number of individual physics actors and substructures. number of repetitive parts in different places of the workflow. User preferences would be to synchronize the settings for repetitive components. What would be the best strategy?
 - It would be great to have the possibility (standard) of exporting of the actors but not from GUI mode, like save as:
 - function of API of kepler.jar library?
 - as an argument of the Kepler in non-gui?
 - User friendly GUI to steer the workflow execution/provide interaction (standalone)

- Atomistic simulation methods such as molecular dynamics are used to study materials. Output from the simulation (atoms within the simulation box), needs further data processing e.g., the calculation of radial and angular distribution functions. Significant difficulties arise when disordered systems (e.g. glasses) are investigated. In such cases, in order to exhaustively characterise the structure of the system, one needs to take into account the medium-range order.
- A software suite -ANELLI has been designed with the aim of performing ring analysis (medium-range order in computer-simulated solid state)
- Requirements:
 - Creation phase:
 - Easy addition of new C- actors
 - Easy parameterisation and triggering the components
 - Possibility of easy update of the codes on the fly
 - Execution phase:
 - Running in distributed computing environment
 - Workflow should be transparent to user



ANELLI Workflow



- Several workflow use cases implemented (DIAPL, BF) – more on next presentation
- Based on *I/VOA* standards
- The goal is to create universal modules that could be used by different astronomy groups
- Requirements:
 - Interoperability with VO services and applications (mainly SAMP, ds9, Topcat)
 - Interactivity in the workflows
 - Integration with iRods
 - Integration of different codes
 - User friendly GUI to run w-f

Example - DIAPL



11

Applications | Unnamed1 | /home/pci/w... | DIAPL_img_a... | SAMP Hub | SAOImage ke... | pci@pci-think... | pci@pci-think... | 2512 rpm | net | mem | 2013-10-21 15:15

/home/pci/work/kepler/astrogrid/DIAPL_img_avg/Kepler/DIAPL_img_avg-samp-new6.xml

File Edit View Workflow Tools Window Help

Components Data Outline

Search Components

Advanced S... Sources Cancel

All Ontologies and Folders

- Components
- Projects
- Statistics
- Demos
- Actors
- Dataturbine
- Directors
- Job
- Opendap
- Outreach
- Plgrid
- R

SAMP Hub

File Clients Profiles

Clients Messages

Clients

- Hub
- SAOImage D59

Registration

Public ID: hub

Profile: Internal

Metadata

samp.name: Hub

samp.description.tex: org.astrogrid.samp.

samp.icon.url: http://127.0.0.1:384

author.mail: m.b.taylor@bristol.ac.uk

author.name:

Subscriptions

samp.app.ping

samp.query.by-meta

x-samp.query.by-meta

Workflow

Image averaging with DIAPL

Workflow parameters

- parameters: /home/pci/work/kepler/astrogrid/DIAPL_img_avg/Kepler/DIAPL_img_avg-samp-new6.xml
- ParameterSet

Utility functions

- basename: (function(path:string) path.split("/")[(path.split("/").length-1)])
- noext: (function(t:string) t.substring(0, t.lastIndexOf(".")))

Locations of files

- imagesDir: \$(imagesDir)
- reference: \$(reference)
- inlist: \$(inlist)
- outputDir: \$(outputDir)
- template: \$(template)

DIAPL parameter files

- diaparDir: \$(diaparDir)
- instrument: \$(diaparDir)/\$(instrument)
- parMask: \$(diaparDir)/\$(parMask)
- parFwhm: \$(diaparDir)/\$(parFwhm)
- parSfind: \$(diaparDir)/\$(parSfind)
- parXymatch: \$(diaparDir)/\$(parXymatch)
- parXygrid: \$(diaparDir)/\$(parXygrid)
- parResample: \$(diaparDir)/\$(parResample)
- parMstack: \$(diaparDir)/\$(parMstack)

START true

imagesDir + "/" + inlist

File Re...

Adjust mask met...

checkbox dialog

Refine automated image selection

File	fwhm	sky
im4404.fits	4.32	525.84
im4406.fits	4.39	538.18
im4409.fits	4.31	514.77
im4413.fits	4.42	493.72
im4414.fits	4.16	473.0
im4415.fits	4.41	449.11

OK

1. Prepare bad pixel masks and calculate image statistics

2. Choosing best images for averaging

6.3e+02 6.5e+02

status x bmask x fwhm x

Starting applications:

- > SAMP hub (running)
- > ds9 (running)

1. Prepare bad pixel masks and calculate image statistics

2. Choosing best images for averaging

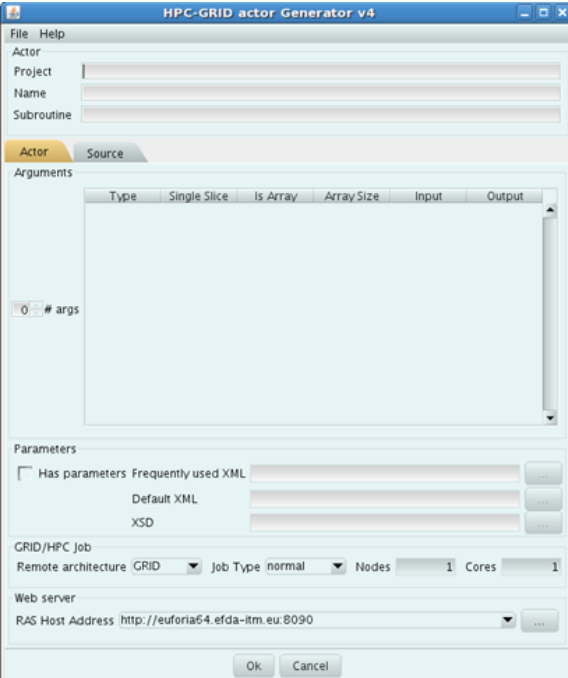


Tools



- wrapping the Fortran or C++ source code into Kepler actor:
 - FC2K(Fortran Code to Kepler),
 - HPC2K (HPC to Kepler),
 - WS2K (Webservices to Kepler)
- run in different computational context
- The precondition for using these tools is that the codes are using CPOs as input/output

- **WS2K:**
 - transforms the FORTRAN or C++ code into a web service,
 - adds data wrapper for accessing remote data structures,
 - installs web service inside a web server,
 - adds Java actor into Kepler
 - Using WS2K, the codes are stored separately and possibly remotely in a HTTP server
- **HPC2K**
 - automatically handles the jobs on the HPC or HTC resources.
 - the code (Fortran, C,C++) must initially be compiled as a library with a main, single function that corresponds to the execution of the code.
 - Uses Serpens suite
- **FC2K**
 - Addresses local and batch execution

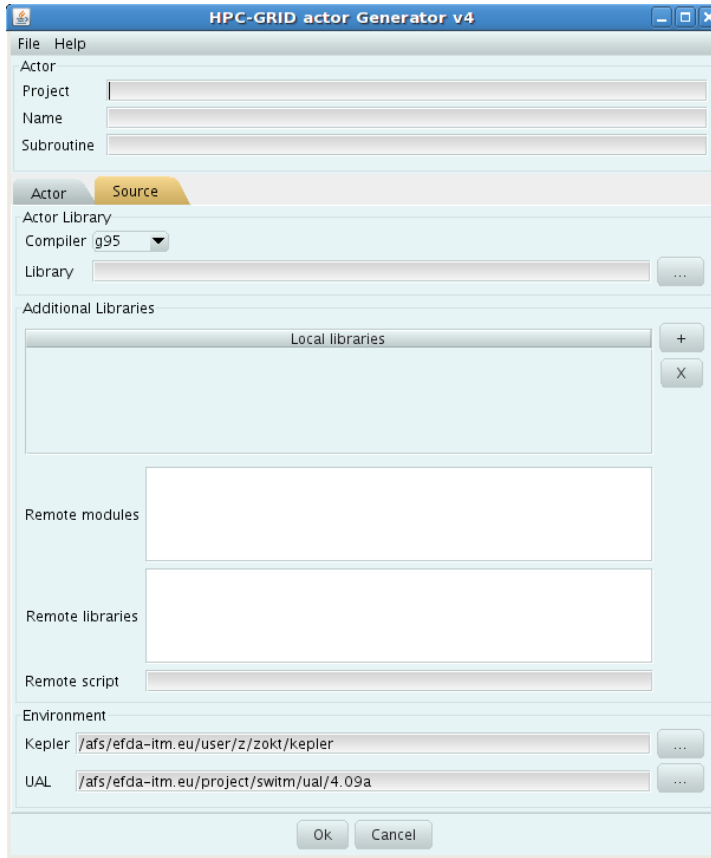


HPC-GRID actor Generator v4
 File Help
 Actor
 Project
 Name
 Subroutine
 Actor Source
 Arguments
 Type Single Slice Is Array Array Size Input Output
 0 # args
 Parameters
☐ Has parameters Frequently used XML
 Default XML
 XSD
 GRID/HPC Job
 Remote architecture GRID Job Type normal Nodes 1 Cores 1
 Web server
 RAS Host Address <http://euforia64.efda-itm.eu:8090>
 Ok Cancel

Project, actor name, subroutine

Arguments

Grid or HPC?



HPC-GRID actor Generator v4
 File Help
 Actor
 Project
 Name
 Subroutine
 Actor Source
 Actor Library
 Compiler g95
 Library ...
 Additional Libraries
 Local libraries + X
 Remote modules
 Remote libraries
 Remote script
 Environment
 Kepler /afs/efda-itm.eu/user/z/zokt/kepler ...
 UAL /afs/efda-itm.eu/project/switm/ual/4.09a ...
 Ok Cancel

Your compiled code

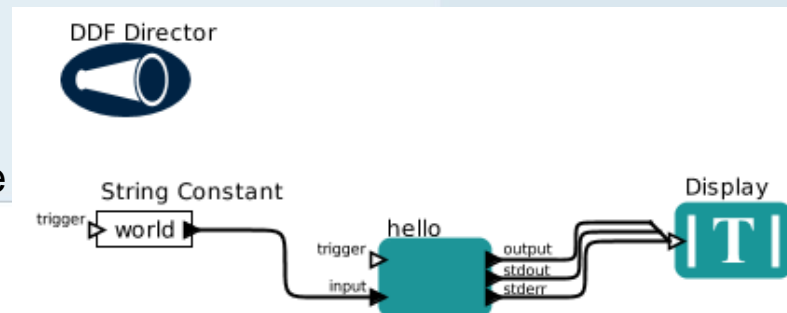
Additional libraries

Kepler and UAL

- The automatically generated actors make use of dynamically loaded libraries which remain in Kepler process space throughout the workflow execution
- The input arguments -> actor port inputs
- Automatic generator of 3 output ports:
 - Output (of c program); StdoutPort; StderrPort
- Actor generation procedure
 - Preparing actor port description in XML
 - Actor generation
 - Make depend
 - Make all
 - Make INSTALL=/path/to/kepler/plgrid install
 - Adding actors to Kepler
 - Tools->Instantiate Component
 - Class name: pl.psnk.kepler.jna.c-app-name

```
int main(int argc, char *argv[])
{
    switch (argc)
    {
        case 2: printf("hello %s", argv[1]);
                break;
        default: usage();
                return (1);
    }
    return(0);
}
```

```
<actor>
  <inputs>
    <in type="String" name="input"/>
  </inputs>
</actor>
```



- **Use case:**
 - Common core installation for all users (90+)
 - Personal set of actors/modules and specific settings
 - So far each user had copy of the Kepler in home directory
- **Solution**
 - scripts creating a shadow copy of central installation in the user area
 - only working directories specific for each user installed in the user area
- **Easy management and maintenance**

- Large workflows may trigger issues with proper understanding of data and execution flow. Having complex flows including multiple and nested composite actors (that wrap native code), makes it hard to determine how workflow is actually executed.
- ITM Workflow Profiler that is based on standard Provenance Recorder, however, it adds some modifications into it.
- Assumptions:
 - internal loop counters should be reported,
 - global parameters should be stored in the output,
 - each actor during its execution should store input and output value
 - execution of native codes should be reported in the output.
 - data transfer details and execution time included
 - Interactive analysis (during run)
- We have decided to use quasi XML format that can be analysed by external tool, however, it is not required to read the whole, complete XML output in order to analyse it.

- MWRKF browser visualises the output of the Provenance recorder (Workflow Profiler's result).
- person executing workflow can analyse the results live and determine elements that are most time consuming.
- An actor can be a composite actor. In this case the elapsed time pasted is the sum of elapsed time related to all of its children
- Clicking on a composite actor tab gives access to the content of this composite actor.

Sort by	none	Filter	none	Time def	of fire
Variable			Value		
tstop			11.0		
dt			0.002		
tstart			10.0		
Actor	Fire		Elapsed time		
ualinit	1		00:00:01.722		
shot	2		00:00:00.000		

Sharing workflows



20

- Web based GUI for saving/sharing workflows (.xml/.kar) with the metadata and additional files (based on Liferay)
- Possibility of the workflow parameters modification
- Downloading and sharing of the workflows between users
- Searching through the workflows (name, metadata, etc)

W-f manager Groups WORKFLOW

Workflows

Upload workflow

Select group: pci@camk.edu.pl (private)

Uploaded workflows

Id	Name	Description	Tags	Kepler version	Owner
6	DIAPL_img_avg-samp-new8.xml				pci@camk.edu.pl

Save Download Edit metadata Share Delete

DIAPL_img_avg-samp-new8.xml Parameters

Name	Value	Edit
parameters	/home/pci/work/kepler/astrogrid/DIAPL_img_avg/Kepler/DIAPL_img_avg.properties	
basename	(function(path:string) path.split("/")((path.split("/").length()-1)))	
noext	(function(t:string) t.substring(0, t.lastIndexOf(".")))	
instrument	\${diaparDir}/\${instrument}	
parBmask	\${diaparDir}/\${parBmask}	
parFwhmm	\${diaparDir}/\${parFwhmm}	
parSfind	\${diaparDir}/\${parSfind}	
parXymatch	\${diaparDir}/\${parXymatch}	

Sharing workflows – group management



21

- Private group for each user
- Creation of the new groups
 - Rights management
 - Adding/removing users

W-f manager Groups WORKFLOW

Groups

Type the group name Create new group

Your groups

Group name	Your status in group
pci@camk.edu.pl	Admin
diapl	Admin
spectral analysis	Admin

Members of group diapl

Type user email Add member

Members

Email	Status
pci@camk.edu.pl	Admin
psych@camk.edu.pl	Manager

Change status Delete member from group

GUI for running workflows in distributed environment



22

- Liferay portlet developed, using VineToolkit/QCG Grid middleware
- Central installation concept used
- Kepler transparent from user perspective

The screenshot displays the 'Anelli - Vine Toolkit' web interface. The browser address bar shows 'https://vine.man.poznan.pl/web/guest/anelli'. The interface has a navigation bar with tabs: Welcome, NanoToolkit, NanoMech (selected), Chemistry, Acoustics, Visualisation, QCG-Services, File Manager, SSH Client, and Configuration. Below the navigation bar, there are three main sections: Nanotechnology, QCG-Broker, and Monitoring. The 'Nanotechnology' section is active, showing the 'Anelli' job configuration form. The form includes fields for Case name (PbSiO3), Configuration file (PbSiO3.xyz), Job name (prefix) (PbSiO3), Species list (Si O), Bounding box (A, B, C) (32.4016, 32.4016, 32.4016), Cutoff radius (2.1), Minimum cluster size (3), Create visualization files (checked), and Compute geometrical properties (checked). There are also input fields for min ring length (3), max ring length (40), RDF nbins (210), and ADF nbins (180). A 'Submit job' button is at the bottom of the form. To the right of the form, the 'Results' section shows a 'Download results' button and a 'Progress' bar that is green and labeled 'FINISHED'.



Summary



- Improvement of the provenance in order to report the firing of composite actors.
- The exposure of the composite actors as classes instead of instances in an automatic way.
- Address a need for automated tools could perform export/import functionality while using the API.
- User friendly GUI to steer the workflow execution/provide interaction

- Presented tools, methods and services enhancing the usage of Kepler-based scientific workflow framework.
- This research has been conducted by scientific communities that make intensive usage of Kepler framework.
- Users' requirements have been addressed by development of tools and services that could be mostly reused by different user communities.

■ Co-authors:

- Marcin Plóciennik, Michal Owskiak, Piotr Spyra, Bartek Palak, Tomasz Żok PSNC, Poland
- Szymon Winczewski, Jarosław Rybicki, TASK, Poland
- Pawel Ciecielag, Wojciech Pych, CAMK, Poland
- Frederic Imbeaux, Bernard Guillerminet, Philippe Huynh, Thierry Aniel, CEA France

- This research has been partially supported by the European Regional Development Fund program no. POIG.02.03.00-00-096/10 as part of the PL-Grid PLUS project.
- Work in the EFDA ITM-TF was funded by EURATOM through the European Fusion Development Agreement and the Participating Countries.