int.eu.grid

Interactive Workflows

Branislav Šimo, Ondrej Habala, Ladislav Hluchý

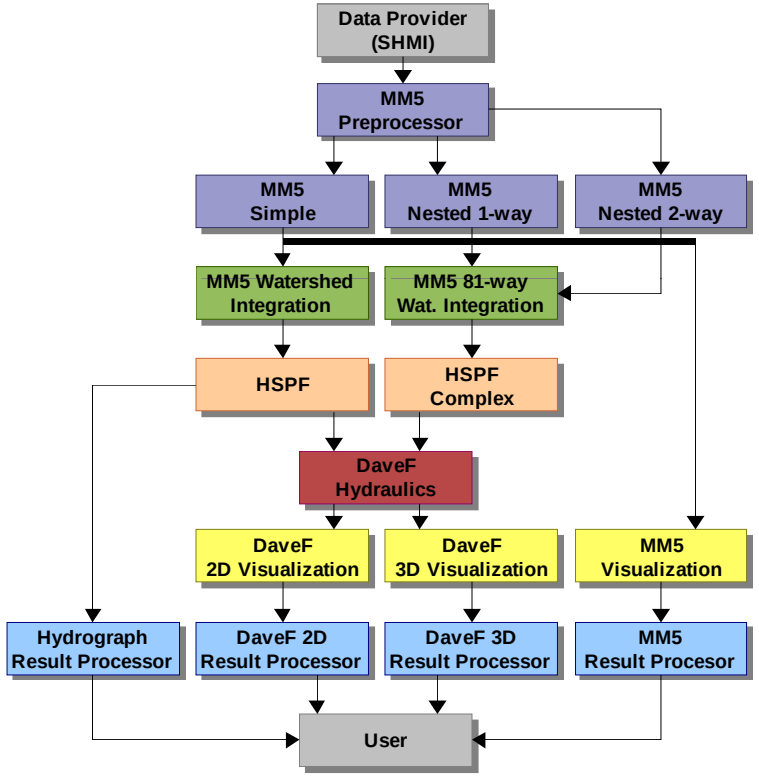
Institute of Informatics,
Slovak Academy of Sciences

Outline

- Motivation
- Approach
- Implementation
- Conclusion

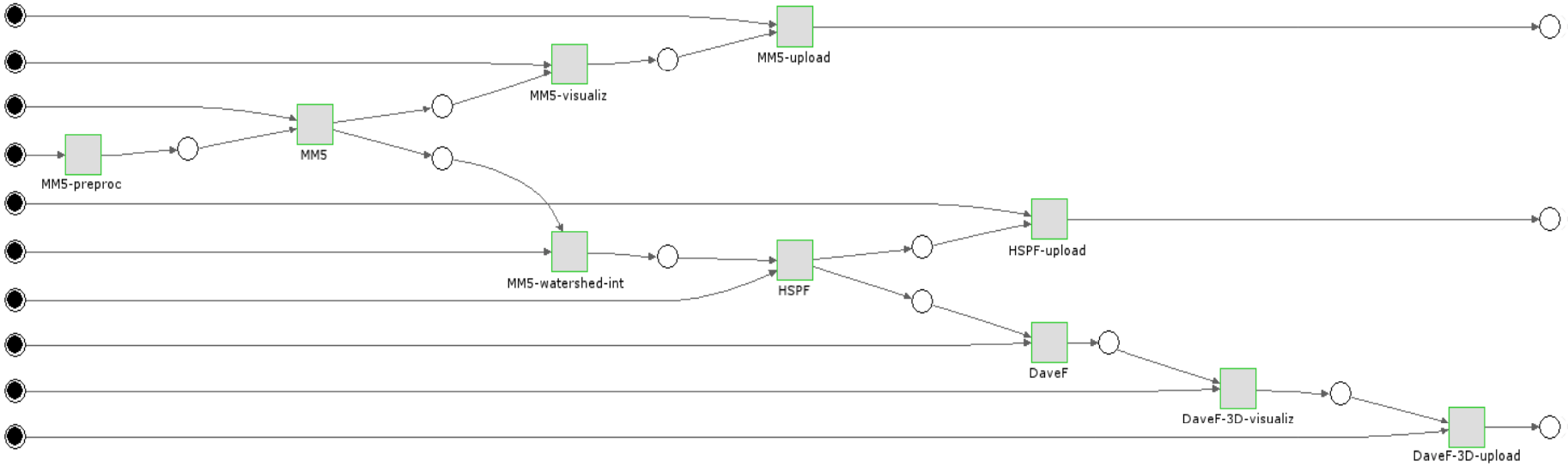
Motivation

- Application
  - Flood forecasting – a workflow of different tasks
  - Set of job **services** orchestrated by a Grid Workflow Execution Service (GWES), controlled and monitored using Java GUI
    - Uses Petri nets
    - GWES and GUI originally developed in the project K-Wf Grid by Fraunhofer FIRST
    - http://www.gridworkflow.org/kwfgrid/gwes/docs
- Using current grid tools, there is no finer control of the workflow
  - Once the workflow is submitted, it cannot be altered



Application workflow

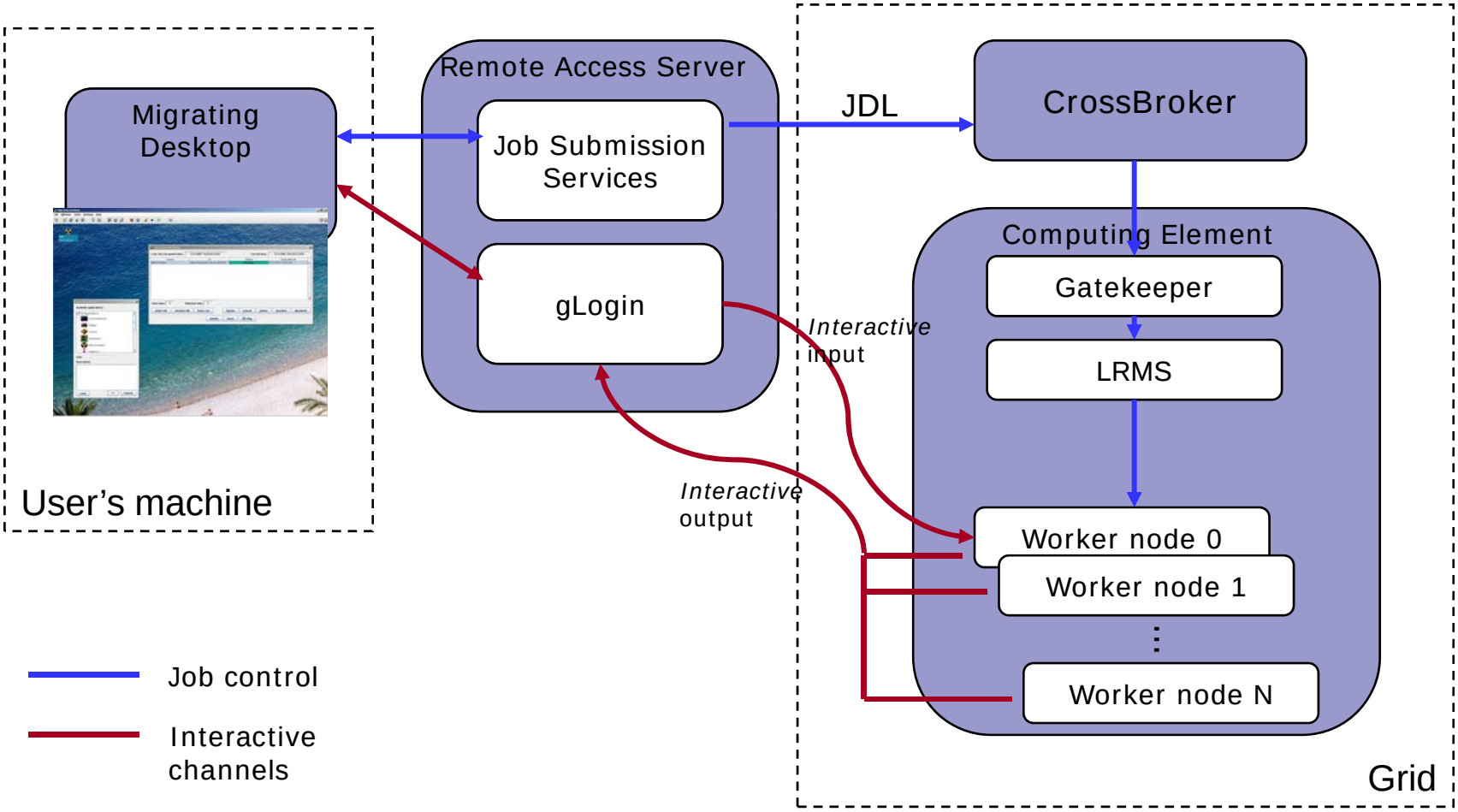
- Petri nets



Motivation

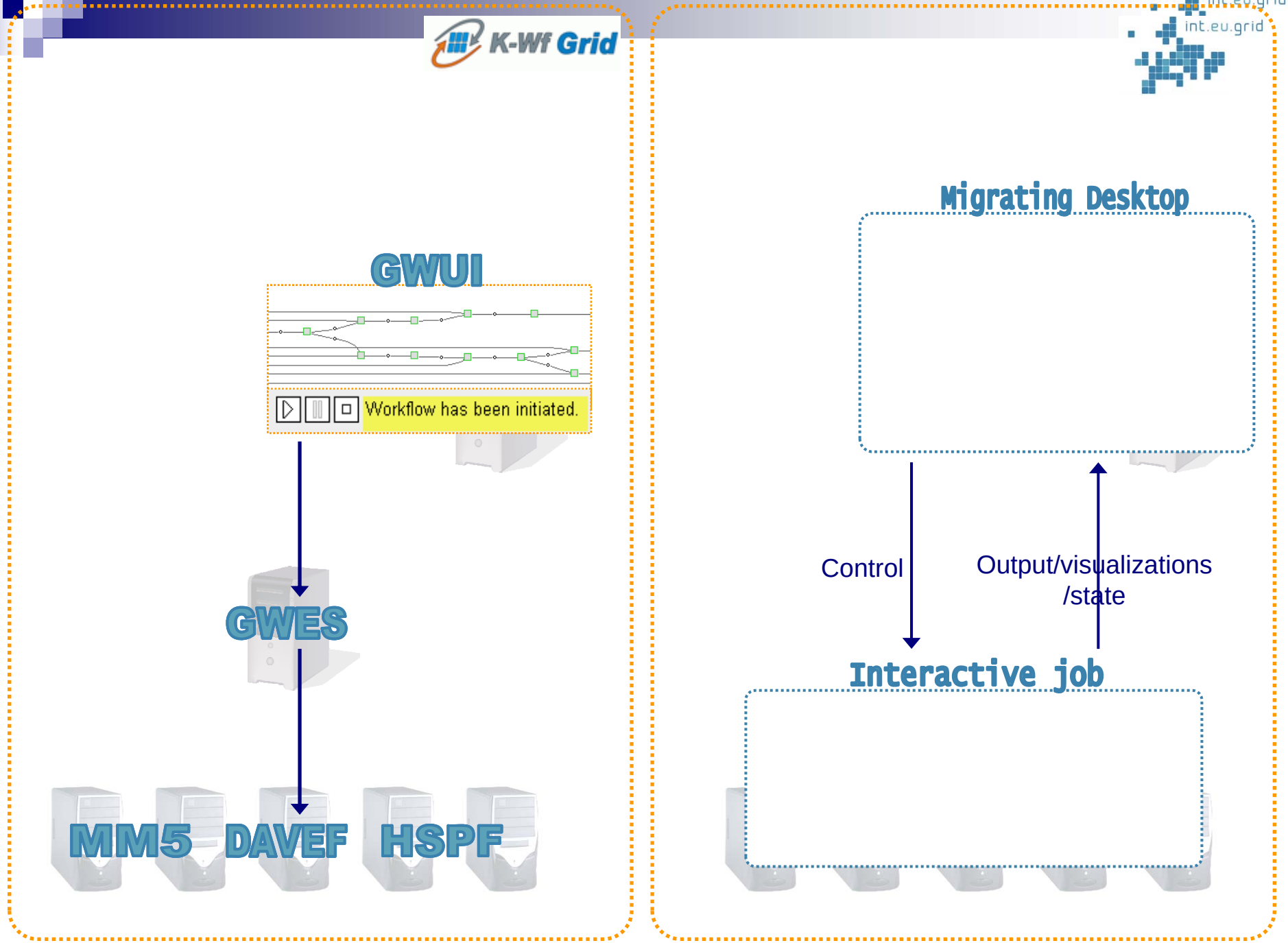
- Int.eu.grid project provides tools for interactive control of applications running on remote grid resources
- Let's implement the application as an interactive job
  - Allow users to manage interactively and comfortably complex jobs composed of multiple program executions
  - Real-time access to job state and results
  - Negligible task startup time

Int.eu.grid - Interactive channel

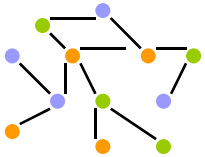


Approach

- Modification of the Grid Workflow Execution Service (GWES)
  - Tool for management of applications composed of web and grid services
- Use the interactive channel of the Int.eu.grid project architecture
  - Forward commands from a GUI to the on-site workflow manager to control the job during execution
  - Use the channel for transfer of debug/info messages and simulation results to the client GUI


GWUI
Workflow has been initiated.
GWES
MM5
DAVEF
HSPF
Migrating Desktop
Control
Output/visualizations
/state
Interactive job

GWES



~~Ontology of modules and data~~

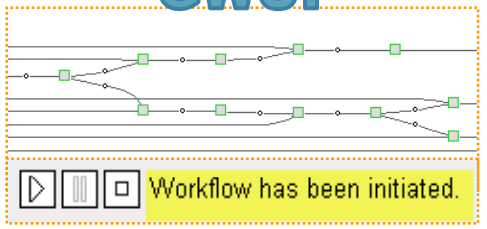
Construction of workflows

Execution of workflows

Monitoring of workflows

Migrating Desktop

GWUI



Control

Output/visualizations

Interactive job

GWES

MM5 DAVEF HSPF

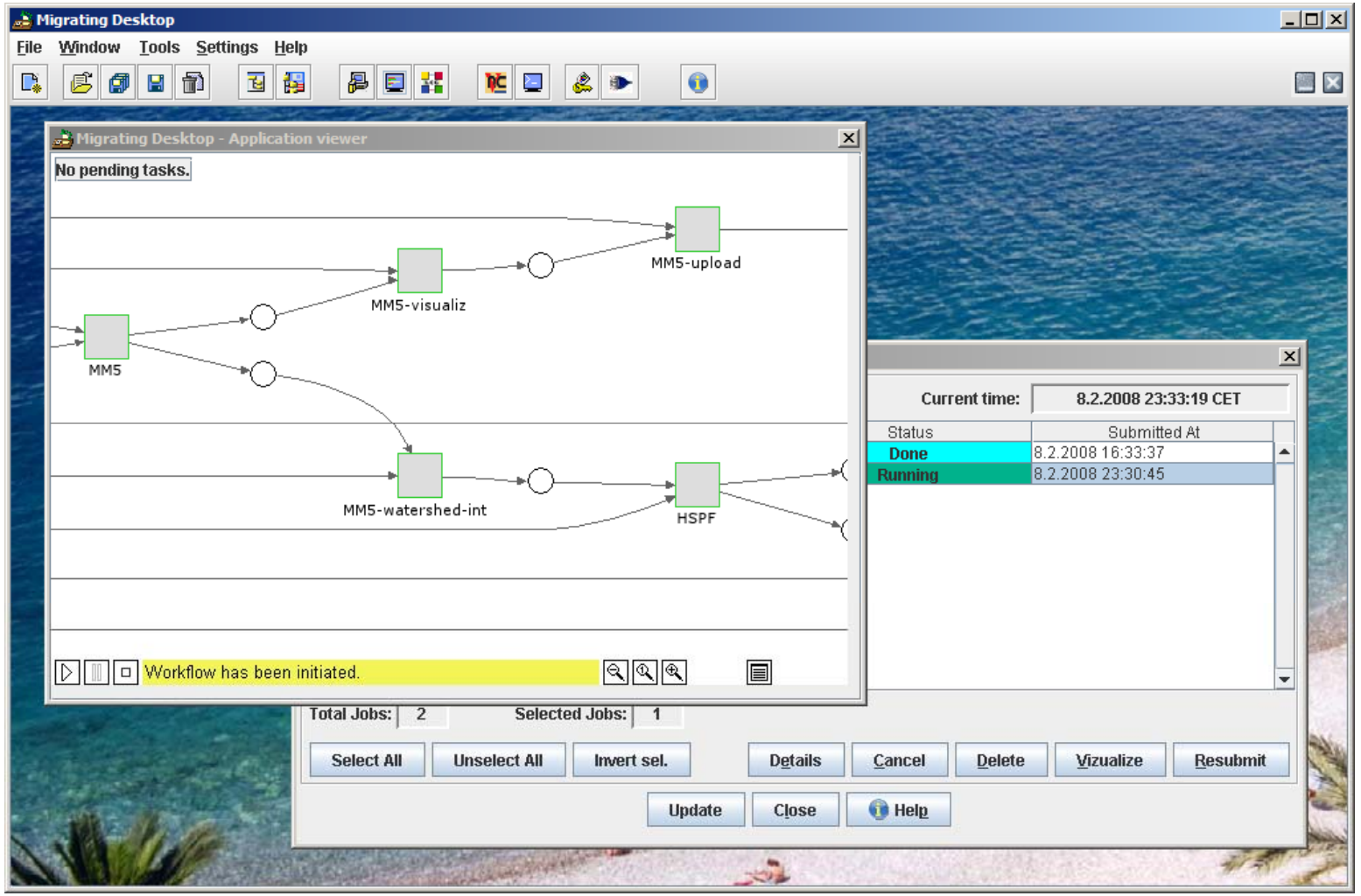
Implementation

- Reimplement GWES⇔UI protocol
  - Was using web service calls (SOAP)
  - Now, we intercept the function calls and serialize the call using a simple “custom” wire protocol
- Reimplement the GUI as an Migrating Desktop plug-in
  - Was an applet before
  - Needed to change some start-up and window code
  - Use the new wire protocol

Implementation

- Internal GWES changes
  - Use the custom wire protocol
  - Node allocation (Scheduler)
    - Need to manage the allocation of nodes belonging to grid job to individual tasks
  - Disabled the semantic part of GWES
  - New activity type
    - Create local job starters instead of WS calls
  - Change GWES from web service to standalone Java process that can be run on the grid
- Grid job submitted contains/downloads the code for the tasks that can be used

MD with workflow window



Flood application interactivity

- Grid job is internally controlled by a workflow manager
  - Everything in one package, submitted as an MPI job ⇨ allocation of necessary nodes
  - Job tasks can be added/removed during runtime
  - Can be paused/restarted/modified and saved/loaded
- Job outputs/visualized data are available through the MD interface
- Job may be cloned
  - State = workflow state + intermediate data, may be easily transferred
    - XML file
  - Good for parameter studies

Relevance of interactive workflows

- Ability to interactively manage a workflow of tasks running as a grid job on remote grid resource by direct communication with the workflow engine running inside of the job
- Tasks executed as part of the workflow start immediately compared to the ones going through resource broker and queues
- Suitable for applications for which the user may want to adapt his execution during runtime to partial results.
  - Instead of repeatedly trying to run, tune, debug, and change a master script of the application, the user can modify the application workflow at runtime.
  - If the need arises, another analysis to process any interesting partial results that were computed may be added.
  - A subtree of the workflow may be cancelled, if a simulation provides uninteresting data, and resources shifted to other parts of the job.
- Any application that currently uses a shell script calling several components (binary modules or other scripts) can be easily converted to a visually controlled workflow.
- The workflow can then be saved, exported to an XML file, and later reused.
  - Simple even for non-experts.

Relevance to EGEE

- The infrastructure of int.eu.grid is compatible with EGEE
  - It is possible to submit jobs to EGEE
  - Another way of exploiting the EGEE infrastructure
- Allowing the user to use state-of-the-art, comfortable and powerful interactive workflow management tools developed for SOA environment in the non-SOA EGEE infrastructure.

Conclusion

- A demonstration and one of the use cases for the interactivity features developed in the int.eu.grid project.
- Similar to “pilot jobs” approach
  - Pre-allocated resources that are used for arbitrary computation

Thank you

branislav.simo@savba.sk