Porting a Complex Workflow on EGEE Infrastructure: The case of Wien2k

Alex Villazon, Malik Junaid, Mumtaz Sidiqqui and Thomas Fahringer

Distributed and Parallel Systems Group (DPS)

University of Innsbruck

2th EGEE User Forum – Manchester 9-11 May 2007

Information Society and Media

www.eu-egee.org

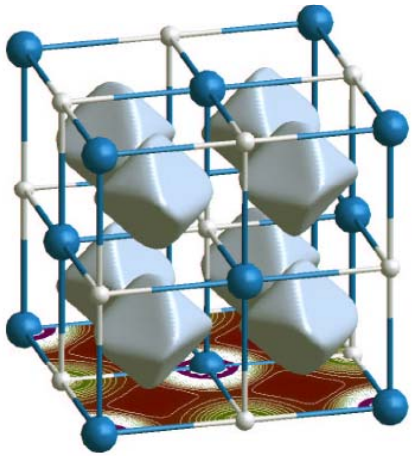
EGEE and gLite are registered trademarks

Outline

- Introduction
- Wien2k application
- Porting workflow on the Grid
- Licensing issues
- Experiments
- Lessons learned and conclusion

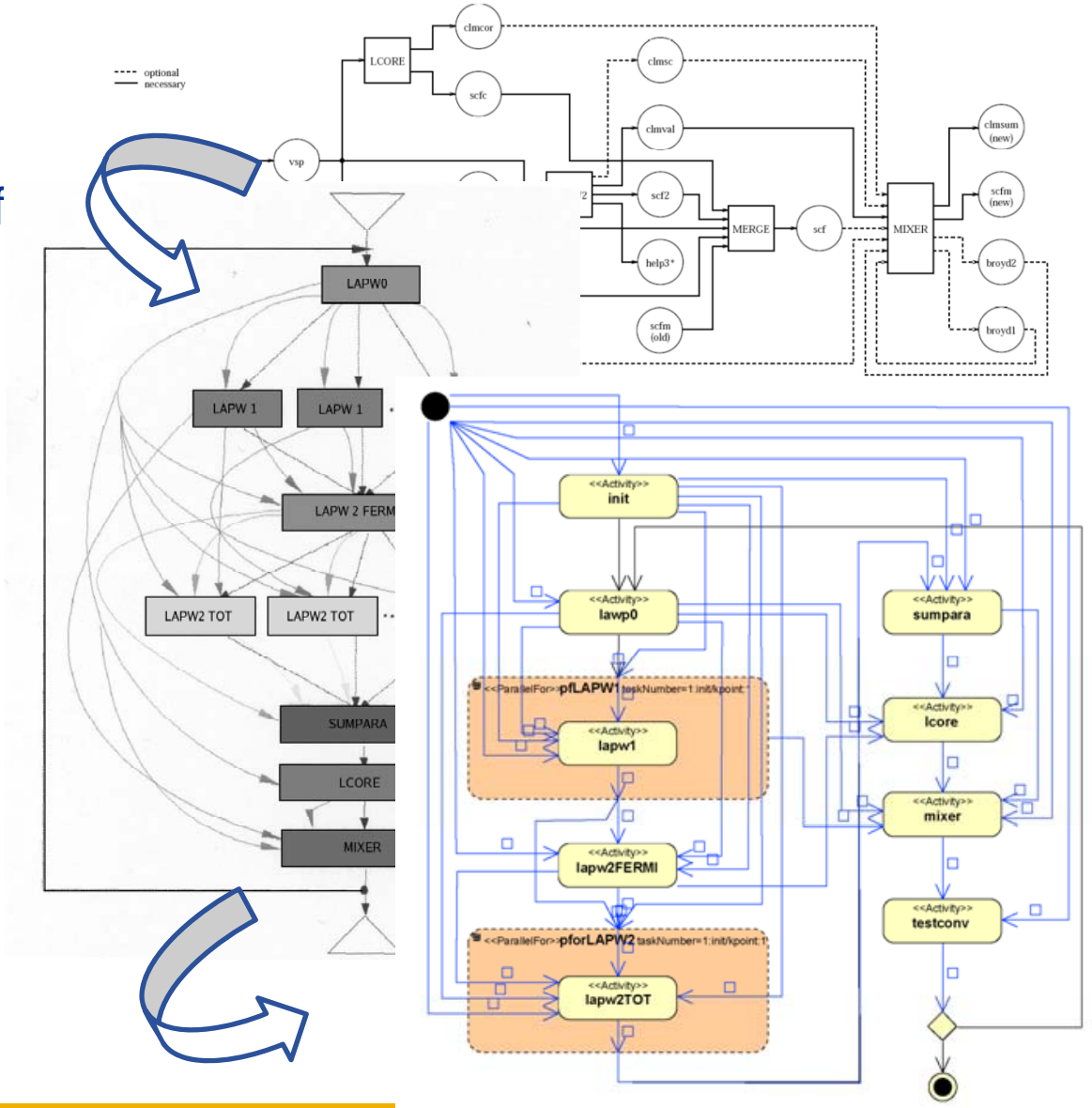
Wien2k application http://www.wien2k.at

- **Performs electronic structure calculation of solids (crystals)**
  - Based on full-potential (linearized) augmented plane-wave ((L)APW method
  - One of the most accurate schemes for band structure calculations
- **Developed by Computational Quantum Chemistry Group at Tech. Uni. of Vienna (K. Schwarz, P. Blaha)**
- **Over 1000 licenses world-wide**
- **Sequential and MPI versions**



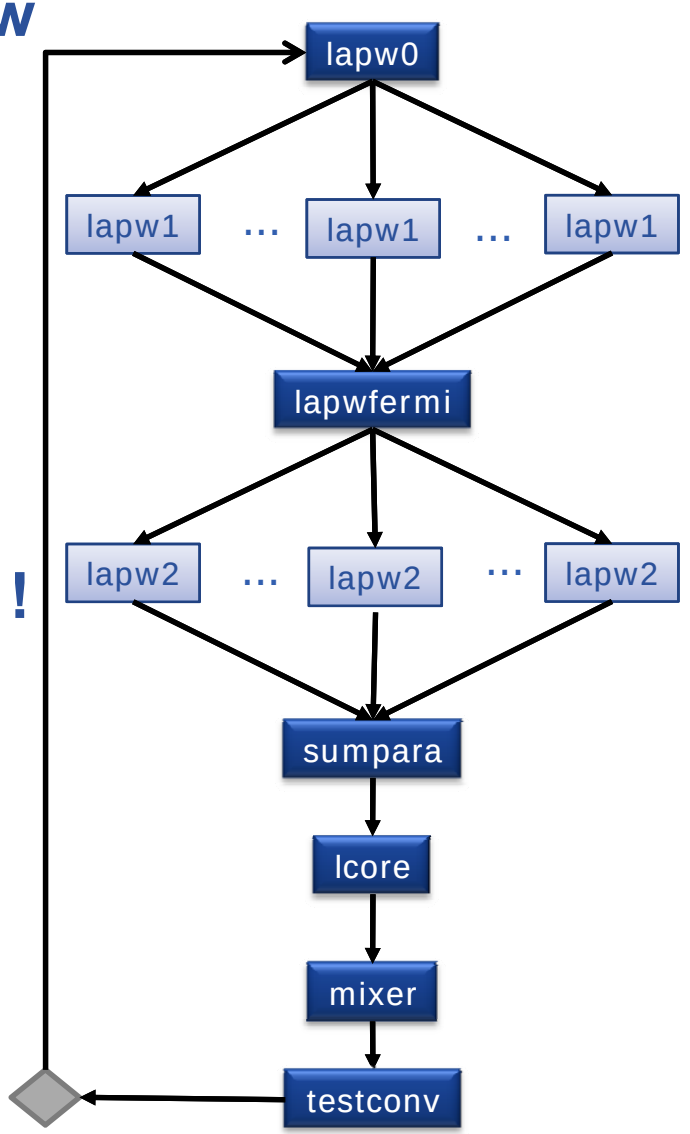
Workflow representations

- **Original workflow**
  - Graphical description
  - Textual description of data dependencies
- **"Translation" to several workflow representations**
  - Detailed control and dataflow
- **Implementation of Grid workflow**
  - Quite different to the original one



Grid workflow

- **We took the simplest sub-workflow**
- **Identify atomic and compound activities**
  - Atomic: single activities
  - Compound: Can be splitted and parallelized
- **Different control- and data- flow**
- **Application activity ≠ Grid activity !**
- **Grid activity**
  - wraps application activity (or activities)
  - can run independently of the others
  - performs data flow management
  - sets environment
  - cleanup environment

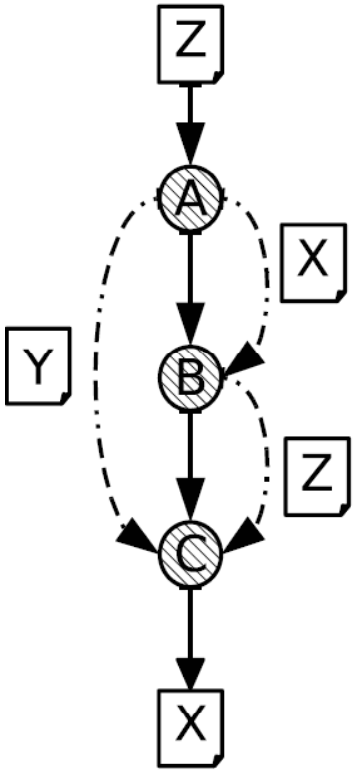


Grid activities

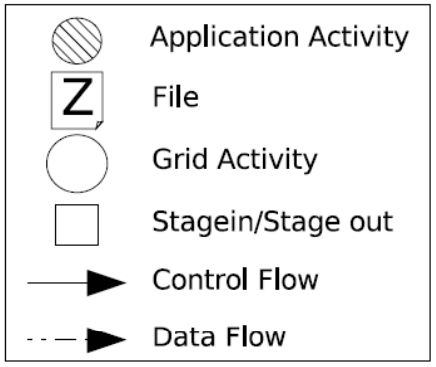
- **Common error when porting applications**
  - Mapping one-to-one application activities and grid activities
- **We experienced several redesigns of the tools !!!**
- **Important aspects related to Grid Middleware**
  - Execution models
    - Centralized workflow enactor
    - Delegation using execution agents (e.g. DIANE)
    - Using resource broker or manual submission
    - Shared file system on worker nodes
    - Application deployment
    - Workflow support of the middleware (DAG)
  - Data management models
    - Direct access to file transfer mechanisms
    - File staging
    - Transfer to intermediate data repositories

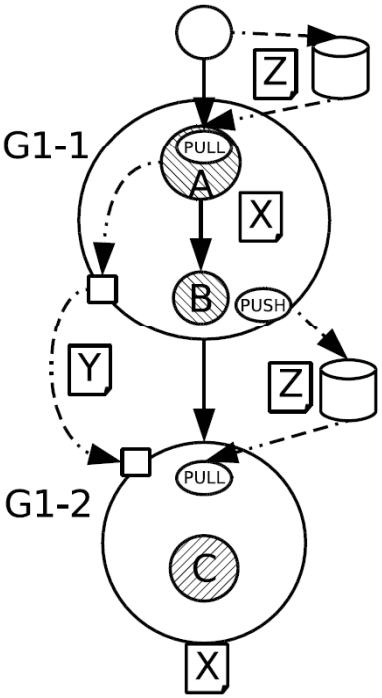
How to implement Grid activities

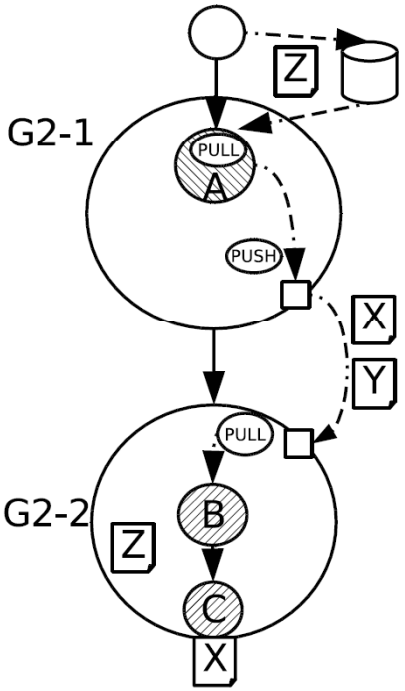
- **Activity Attraction Pattern**
  - Using approximated execution times and file sizes
    - Known by the application developer or scientist
  - "Bigger" activities attract "smaller" ones



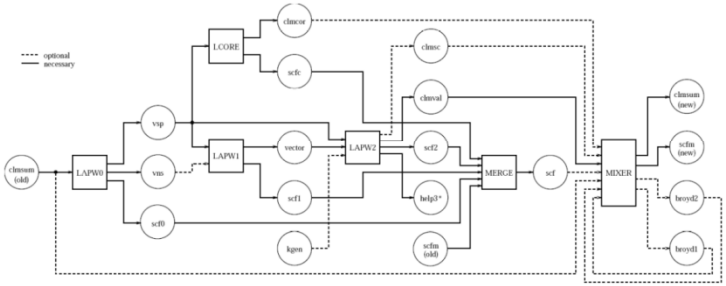
Activity or Data	Case 1	Case 2
A	1 min	2 Hours
B	1 Hour	1min
C	2 Hours	2 Hours
X	1 MB	1 MB
Y	50 MB	5 MB
Z	100 MB	100 MB

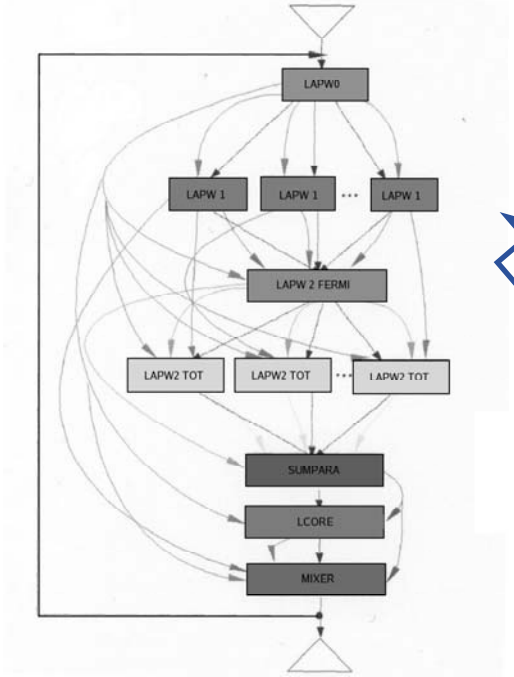


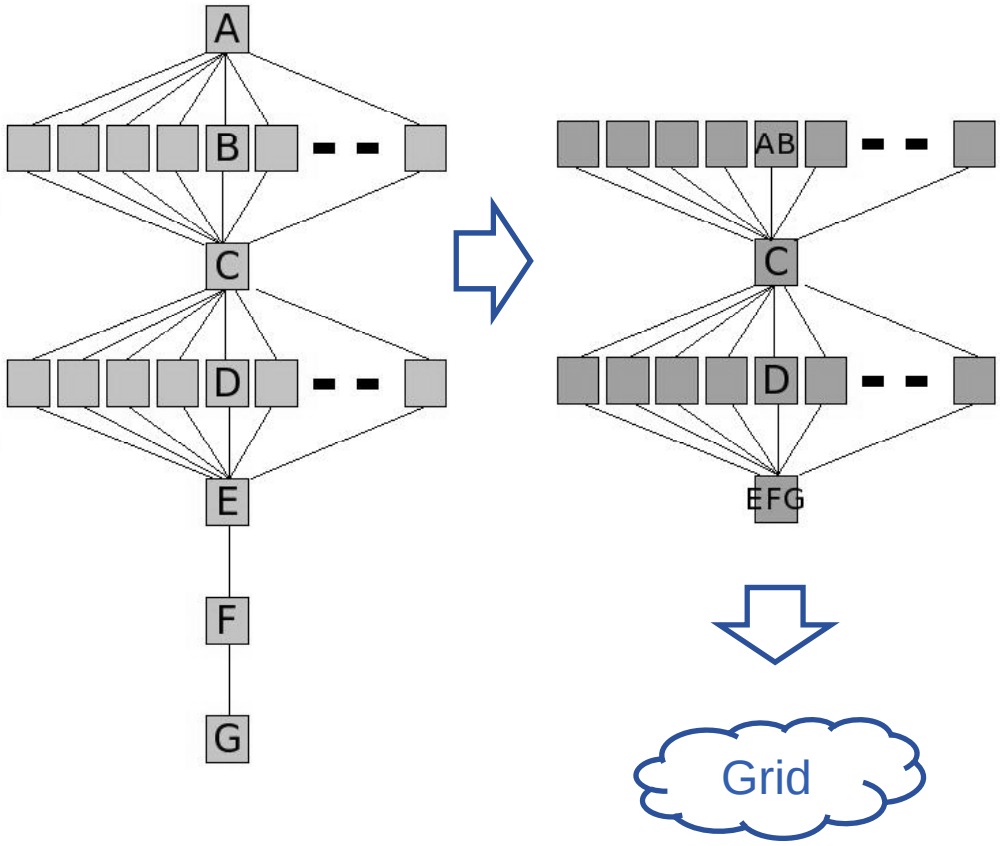




Workflow evolution







Step-by-step

- **First try: GILDA**
  - Too unstable for development... Most of test jobs failed
- **Second try: VOCE**
  - More stable, but not the proper infrastructure to run “tests”
- **Final decision: Local gLite testbed (together with SA1)**
  - Deploy a “mini Grid” using Virtual Machine technology
    - UI, WMS, LB, CE, R-GMA, WN, LFC, SE, VOMS
  - Full control, **but hard to configure and maintain**
    - Bad experiences with VMWare
    - Current XEN-based mini Grid (a full Grid in a Box)
  - Very helpful to refine workflow engine, grid activities, data management... ➔ until a successful “Grid run”
- **Now, running on VOCE and CompChemVO**

Workflow execution

- **gLite workflow support is not sufficient (JDL DAG)**
  - Not expressive enough for more complex control on data dependence
  - No support for loops
- **We built our own workflow engine based on Ganga**
  - Very flexible to generate code on-the-fly
    - Changing workflow
    - Add/change activities
    - Make local tests with local backend
    - Support loops
  - Too “application-specific” (no general workflow description)
- **Complex part (how to make grid activities) is understood**
  - Try other workflow engines (DIANE...)

License issues

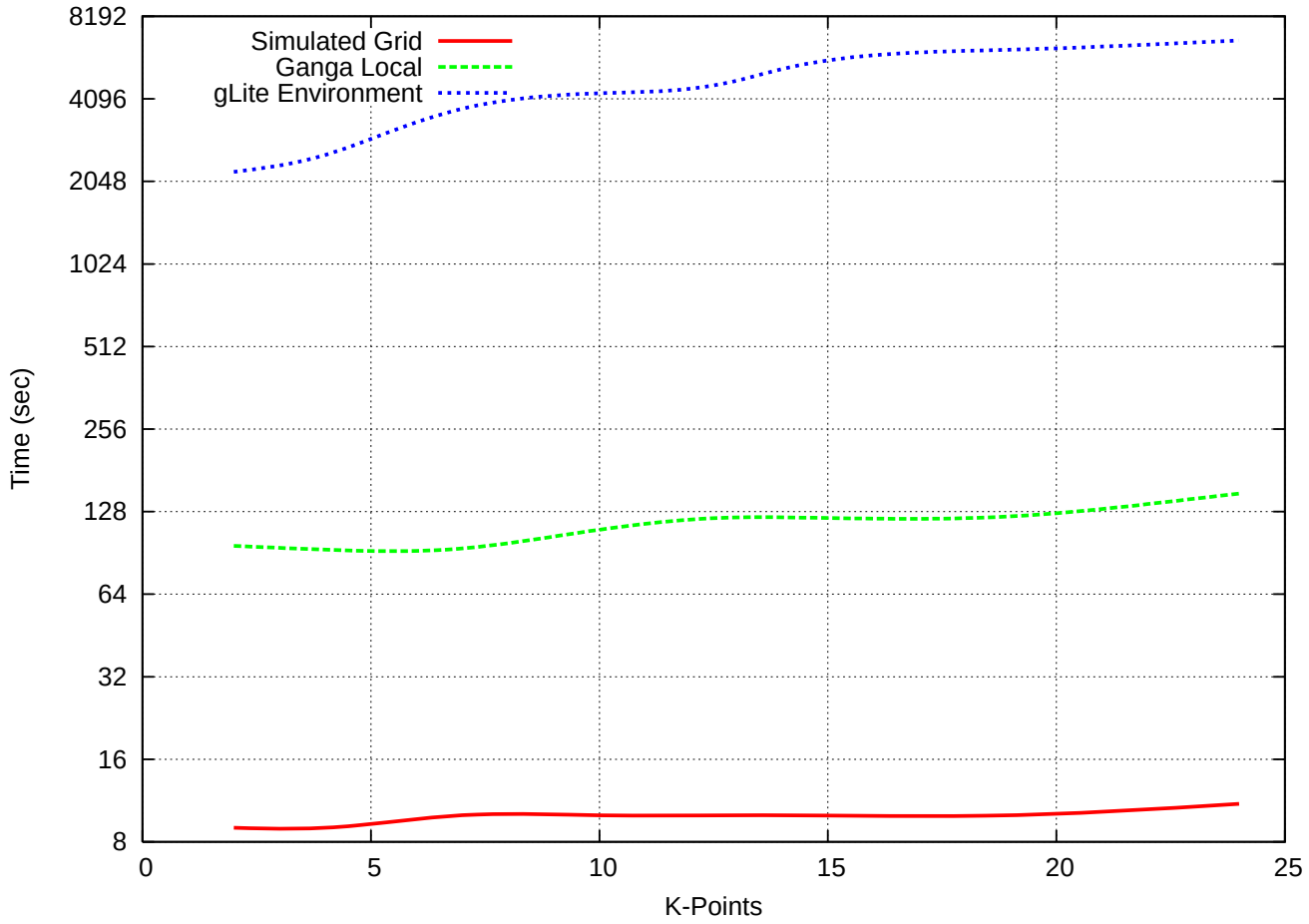
- **VO per licensed software too restrictive**
- **Our proposal: Use proxy roles**
  - Use similar solution as software deployment
    - Licensed users mapped to other accounts
    - VO_<VO_NAME>_SW_DIR access rights not enough
    - May require more sophisticated access control (ACL)
  - Grant licensed role via VOMS
    - VO manager can grant licensed users
    - Automate this process with external licenses database
- **Temporary solution**
  - Stripped Wien2k version deployed “on-demand”
  - Removing executables after execution
  - Introduces additional overhead

Experiments

- **Simple wien2k subworkflow**
  - Small toy crystal calculation
  - Varying the number of parallel activities
    - Varying “k-points” to have up to 150 grid activities

Grid testbed	User Interface	Available CPU	Virtual Organization
DPS UIBK gLite testbed	gliteui.dps.uibk.ac.at	2	dpsgltb
HEPHY UIBK gLite testbed	atlas.uibk.ac.at	5	uibktest
Central Europe VOCE	skurut4.cesnet.cz	1278	voce
Computational Chemistry	ui.grid.unipg.it	3990	compchem

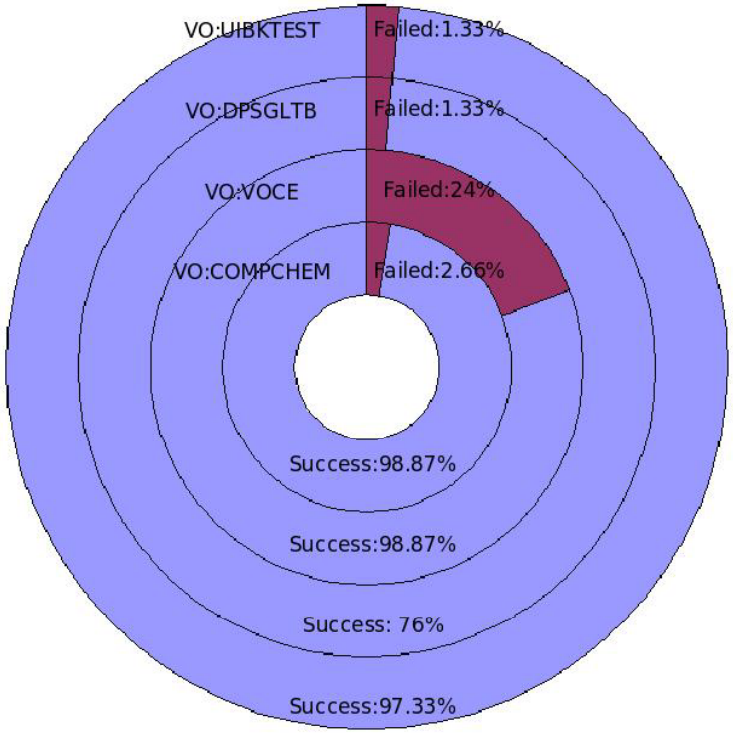
Experiments (cont'd)

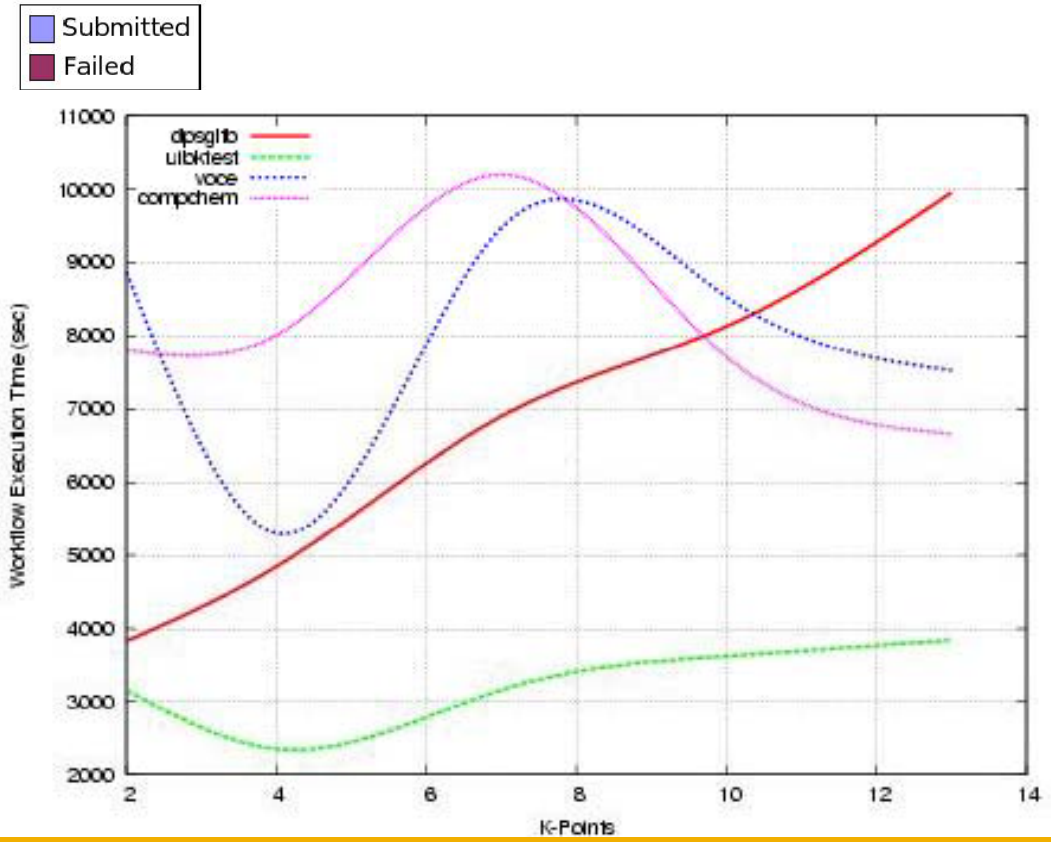


Very high Grid overhead

Experiments (cont'd)

Job Success/ Failure Rates





Lessons learned and conclusion

- **Using “mini Grid” good approach for porting**
  - Big effort to install but could prepare a pre-configured version
- **Important to consider execution time and file size**
  - Do not map “one-to-one” application activities to Grid activities
  - Activity Attraction pattern is a first step
- **Need for widely accepted Grid workflow description language**
- **Generic workflow execution engine needed**
  - Ganga good compromise, but engine from scratch...
- **Application workflow monitoring is needed**
- **Licensing on the Grid remains open issue**
- **Grid overhead**
  - Queuing time probably fine for parameter studies but sometimes inacceptable for Grid workflows

THANK YOU!

Questions?