Rapid Deployment of VS Workflows on PHOSPHORUS using Meta Scheduling Service

M. Shahid, Bjoern Hagemeier
Fraunhofer Institute SCAI, Research Center Juelich.
(TNC 2009)
Introduction and Motivation
  - Bioinformatics Applications on the Grid
  - PHOSPHORUS Testbed
  - Objectives within PHOSPHORUS
Issues related to Grid VS deployment
Development of a VS Framework in PHOSPHORUS
  - Virtual Screening Applications
  - vHTS Framework Design
High throughput Virtual Screening Workflows
Conclusions
Example VS Deployment using UNICORE/MSS
High computational and data storage demands
Data security/privacy restrictions
High Throughput data management & deployment on large-scale Grids
Need of high level tools to enable e-Scientists to use Grids in an easy and transparent way
PHOSPHORUS Testbed

- Large capacity optical networks
- Satisfying e-Science HPC applications’ (high computational & networking) demands…
- Needed: Advanced Grid-aware tools
Objectives within PHOSPHORUS

- Rapid deployment of VS applications to PHOSPHORUS environment
- Using user-friendly workflows based on UNICORE6 / MSS integration
- Efficient Stage-in and stage-out: easy distribution of input and output data
- Data distribution and collection without any data loss
- Support for post-processing and analyses
Issues in using Large-scale Grids

- Management and deployment of large number of jobs
- Scheduling policy: maximum resource utilization in the available time and space
- Pre/Post processing, management of huge amount of output data
- The production environment should provide automated and fault tolerant jobs and files management
The amount of transferred data impacts on the overall Grid performance
- Efficient Data distribution on the Grid storage
- Speedy transfer between computing elements and Grid storage.
- Automated post processing: data mining
Support large-scale deployment of virtual screening services in PHOSPHORUS

Addressing the issues involved in deployment of complex VS workflows

An extensible framework making the Grid more accessible to end users
Virtual Screening by Molecular Docking

- CPU and Data intensive applications:
  - **FlexX (BioSolveIT)**
    - Predicts geometry/binding free energy of protein ligand complex
  - **AutoDock (Scripps Research Institute)**
    - Comparatively time-consuming, a single docking job takes ~ 30-60 minutes on a standard CPU.
  - **Amber (MD Simulations)**
    - Package for simulations of biomolecules.

- **INPUT/OUTPUT data:**
  - Several GB to TB.
High Throughput VS Framework

- Using UNICORE-6 Middleware Technology
  - Seamless, secure and transparent access to Grid
  - Ease of use, reduced complexity, increased security
- Using UNICORE Client extensions/plugins
  - Simplifying interactions with applications on the Grid
- Using Meta Scheduling Service
  - Further support to utilize Grid resources
UNICORE Client Extension

- UNICORE Expert Client uses eclipse RCP mechanism
- RCP is based on plug-in architecture that provides development of extendible components
- UNICORE Client: communication to UNICORE services
- The client extension provides interface to the Meta Scheduling Service
- Exploits job/data management mechanism of UNICORE client
Meta Scheduling Service

- Orchestration & co-ordination of Grid resources, SLA
- Supporting users to utilize maximum available resources
- Workload distribution...
High Throughput Virtual Screening Workflow

- VS Applications: FlexX, AutoDock...
- Fully automated workflow

Diagram:
- Client Plug-in
- Meta Scheduling Service
- Autodock Docking
- FlexX Docking
- Application Wrappers
- Process output
- Format as Input
- Docking Database Management
- Molecular Dynamics Simulations
- Target preparation
- Compound DB (ZINC)
- Select subset (~1 million) MOE2
- Select files (~1-5) PDB
- Preprocess (create RDF)
- Run Flex
- Output: Scores, match info, Conformations (MOE2)
- Get license from license server (flexin)
- FlexX script
- Select ligands for further work
- DB
Example VS Workflow:
One single job submission
- Setting up Application specific parameters
- Input/output specification
Conclusions

- **Rapid deployment**
  - High bandwidth input/output (PHOSPHORUS network layer)
- **Better deployment efficiency with MSS than using UNICORE client plug-ins alone**
- **Simplifies, speeds up the complex virtual screening process in the Grid environment**
- **Flexibility and extensibility**
- **Allow easy integration of other tools & services**
Load MSS Client Plugin

Provide MSS URL, Global Storage URL

Input/Output
Example VS Deployment: UNICORE/MSS (3/9)

Create Job
Example VS Deployment: UNICORE/MSS (4/9)

Application parameters

- Job Name: AutoDock
- Receptor pdbqt File (pdbqt): 1zxb.a
- Multi-mol2 File (mol2): apligands.mol2
- Number of Energy Evaluations (ga_numeval): 25000
- Number of GA Population (ga_pop): 50
- Number of GA Runs (ga_run): 5
- RMS Tolerance (rmstol): 2

Flexible receptor docking
Example VS Deployment: UNICORE/MSS (5/9)

Stage in

Stage out
Example VS Deployment: UNICORE/MSS (8/9)
Example VS Deployment: UNICORE/MSS (9/9)
Thank You for your attention