PRAGMA-

Experience in developing MDO Framework using Globus 3.2

by

Rajeev Wankar, Neelakanta P.

{wankarcs, mc04mt07}@uohyd.ernet.in

Department of Computer and Information Sciences

University of Hyderabad, India

Outline

- Introduction to MDO Framework
- Work at University of Hyderabad
- Snapshots
- Demonstration

Multi **D**isciplinary **O**ptimization (**MDO**)

- Is a methodology for design and analysis of complex engineering systems and subsystems that work coherently to solve Grand challenging problem.
- MDO is used in design of complex engineering systems by performing system level analysis.
- For performing such a system level analysis a software integration system is needed which is termed as MDO Framework.

MDO Framework Requirements

- Architectural design
- Problem formulation construction
- Problem execution
- Information access

Architectural Design

- Provide intuitive GUI
- Support for developing interfaces for adding new programs
- Handle large size problems
- Support collaborative design

Problem Formulation Construction

- Configure complex branching and iterative problem formulations easily
- Support legacy and proprietary codes
- Support multiple optimization methods including multilevel schemes
- Provide debugging support for multiple processes executing across computers on the network

Problem Execution

- Automate execution of processes and movement of data
- Execute multiple processes in parallel
- Support execution distributed across network of heterogeneous computers

Information Access

- Provide database management features
- Capability to visualize intermediate and final results of the analysis
- Monitoring capability for viewing the status of an execution and system status
- Mechanisms for fault tolerance

Activation Graph

- In a Distributed Computing environment various services work in collaboration with each other.
- **The output of one service acts as an input for another service, which after processing produces an output file.**
- An Activation Graph can be created specifying the order in which various services are to be executed.
- The condition on execution and the input/output files of various services could also be specified.

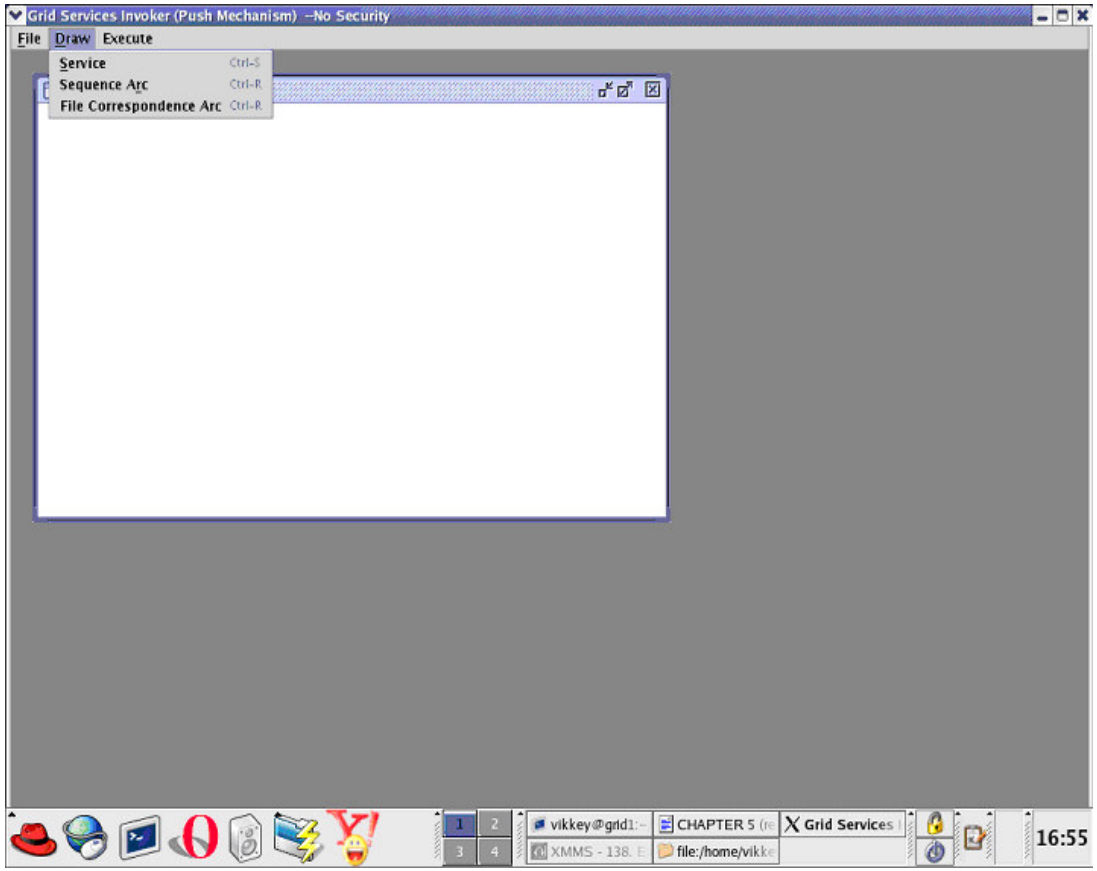
The Grid Service Invoker

It is a Java based GUI Tool to invoke Grid Services-

Main Functionalities

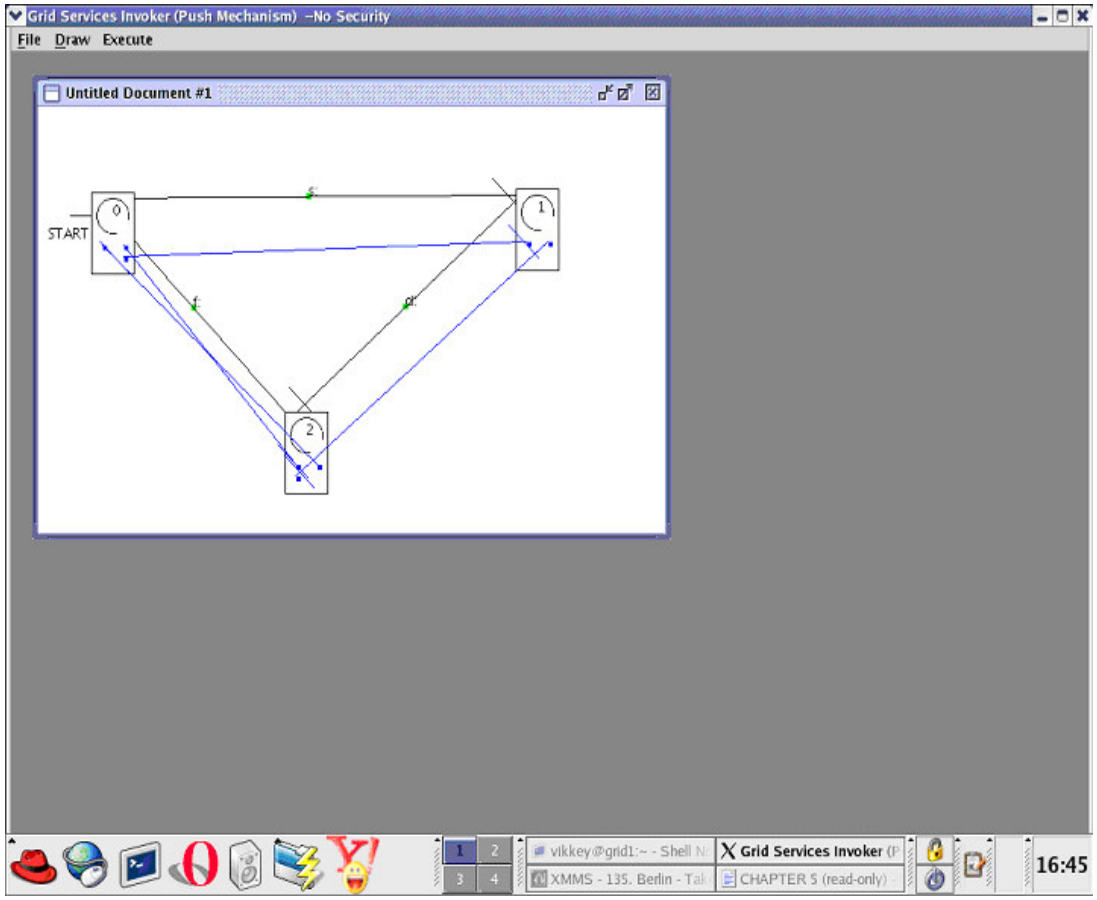
- Choosing a Service
- Specify input/output files
- Specify the order of execution (Activation Graph)
- Specify file correspondence
- Service invocation

GUI of the Grid Service Invoker



In the drawing area of Grid Service Invoker, the user can create an Activation Graph

Activation Graph



Components

- ***Circle*:** Specifies a Service.
- ***Rectangle*:** Specifies the machine in which the service is located.
- ***Small filled rectangles*:** Specifies Input/Output files.
- ***Sequence Arcs*:** Specifies order of execution of services.
- ***File Correspondence Arc*:** Specifies output file of which service will act as an input file of other.

Naming Conventions

- The naming conventions of the various files:

  Example: MathService (with service data)

  Service name: MathService

  (Every service name should end with word 'Service')

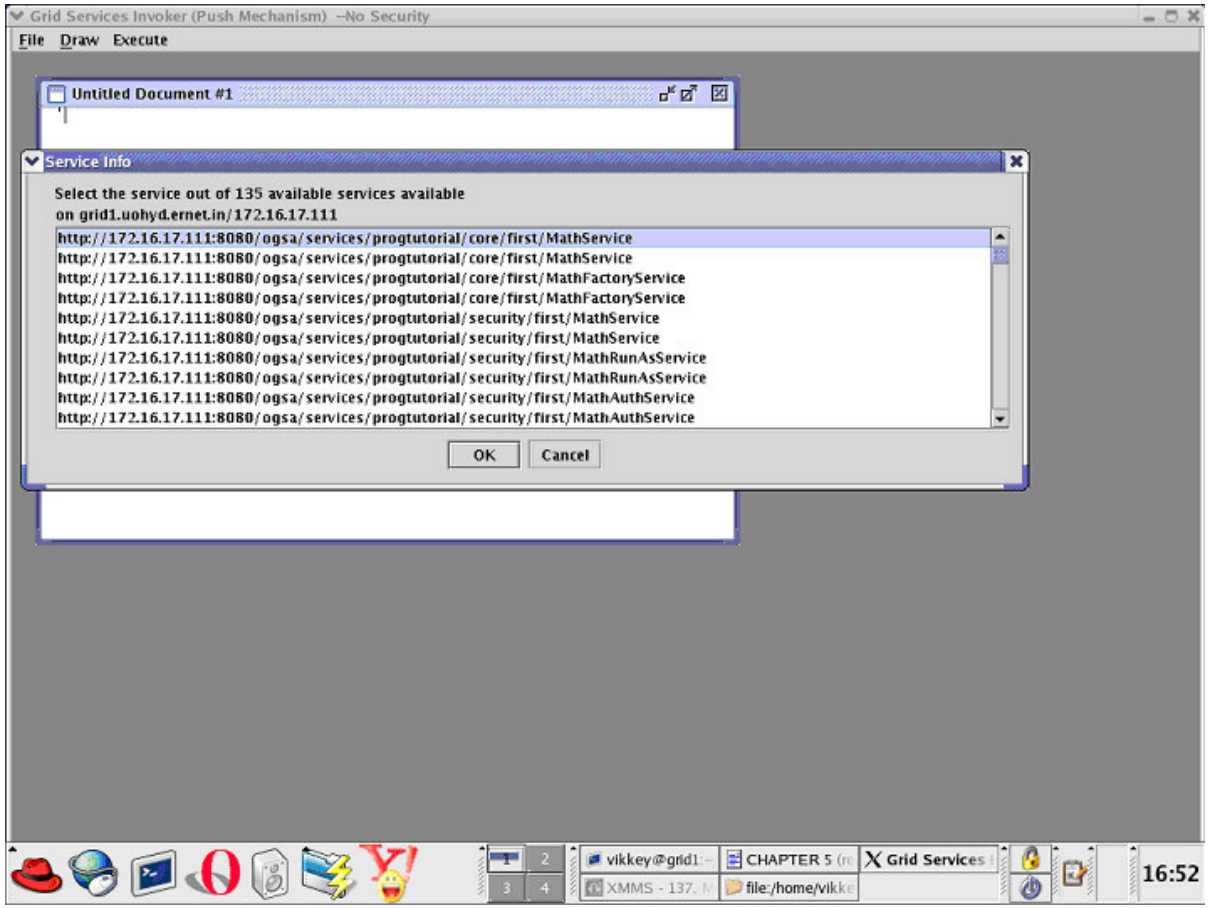
  Port type: MathPortType

  Service Data name: MathData

  Service Data type name: MathDataType

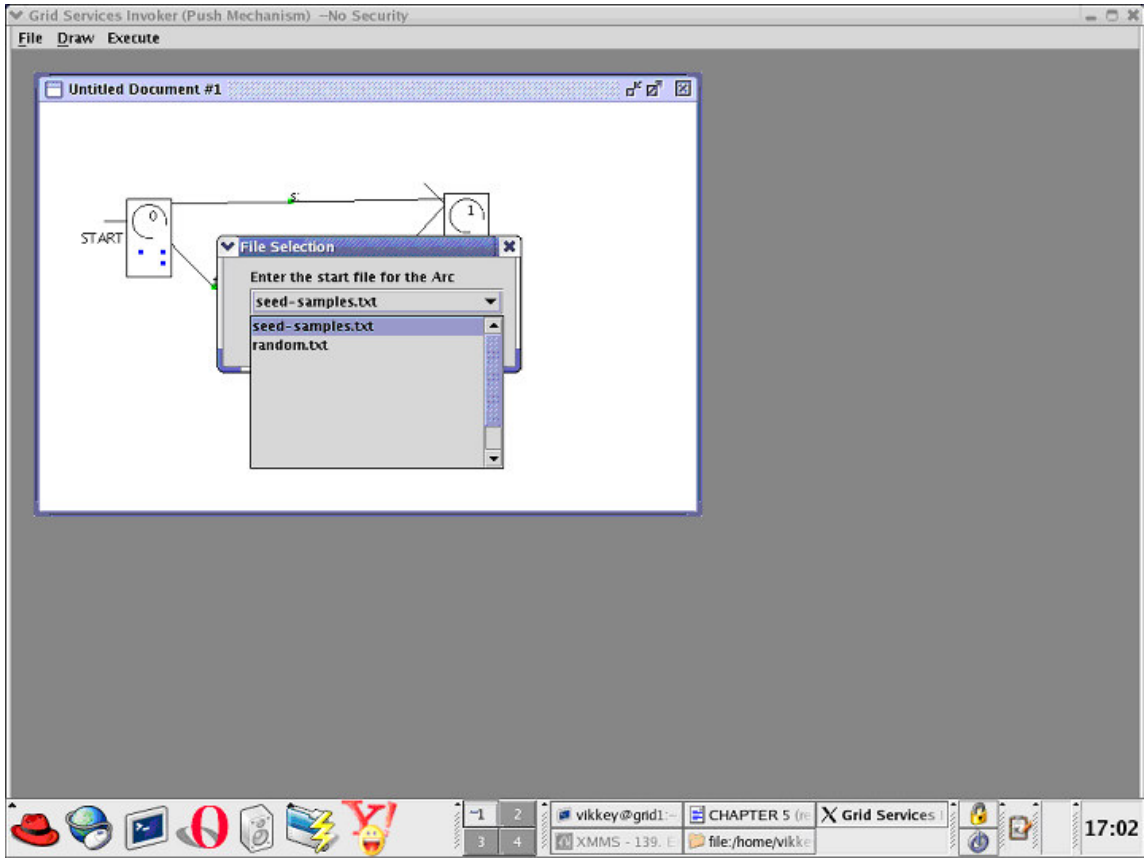
- The service should be deployed in directory pointed by **$MTECHPROJ_DIR**

Few Snapshots



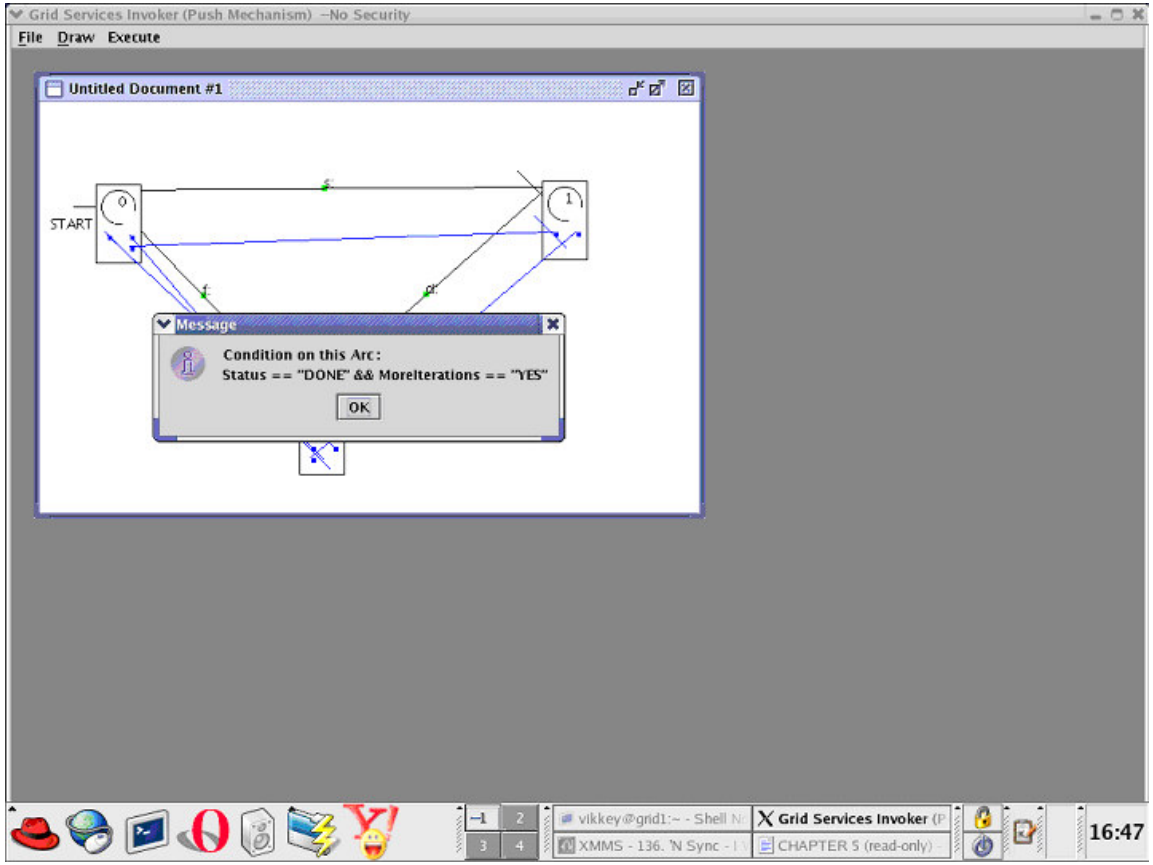
The tool asks user to choose a service from the list of available services on the specified machine.

Snapshots Continued…



The tool asks user for the name of the output file on the specified machine which will be copied as the input file on some other machine

Snapshots Continued…



The tool displaying the condition on the sequence arc clicked

Calculation of PI: A Case Study

1. Generation of random numbers given a **seed** and **number of samples** to be generated
2. Calculation of the value of **PI** using **Monte Carlo method**
3. Checking whether the error in the calculated value of **PI** is within some specified threshold by comparing it with a standard value of **PI**.

If the calculated value is not within the threshold, the whole process is repeated.

Calculation of PI in distributed environment

The three steps above were implemented as **grid services** on three different machines

- RandomNumService (grid1.uohyd.ernet.in)
- PiCalculateService (grid2.uohyd.ernet.in)
- CheckPiService (grid3.uohyd.ernet.in)

RandomNumService

Input files

seed-samples.txt

Output files

random.txt

seed-samples.txt

Description

The **RandomNumService** calculates random numbers based on initial value of seed in its input file. It writes the calculated random numbers in random.txt file and also copies the seed-samples.txt input file to its output directory.

PiCalculateService

Input files

random.txt

Output files

piVal.txt

Description

The service uses Monte Carlo method to calculate the value of PI from the random numbers in its input file random.txt.

CheckPiService

Input files

piVal.txt

seed-samples.txt

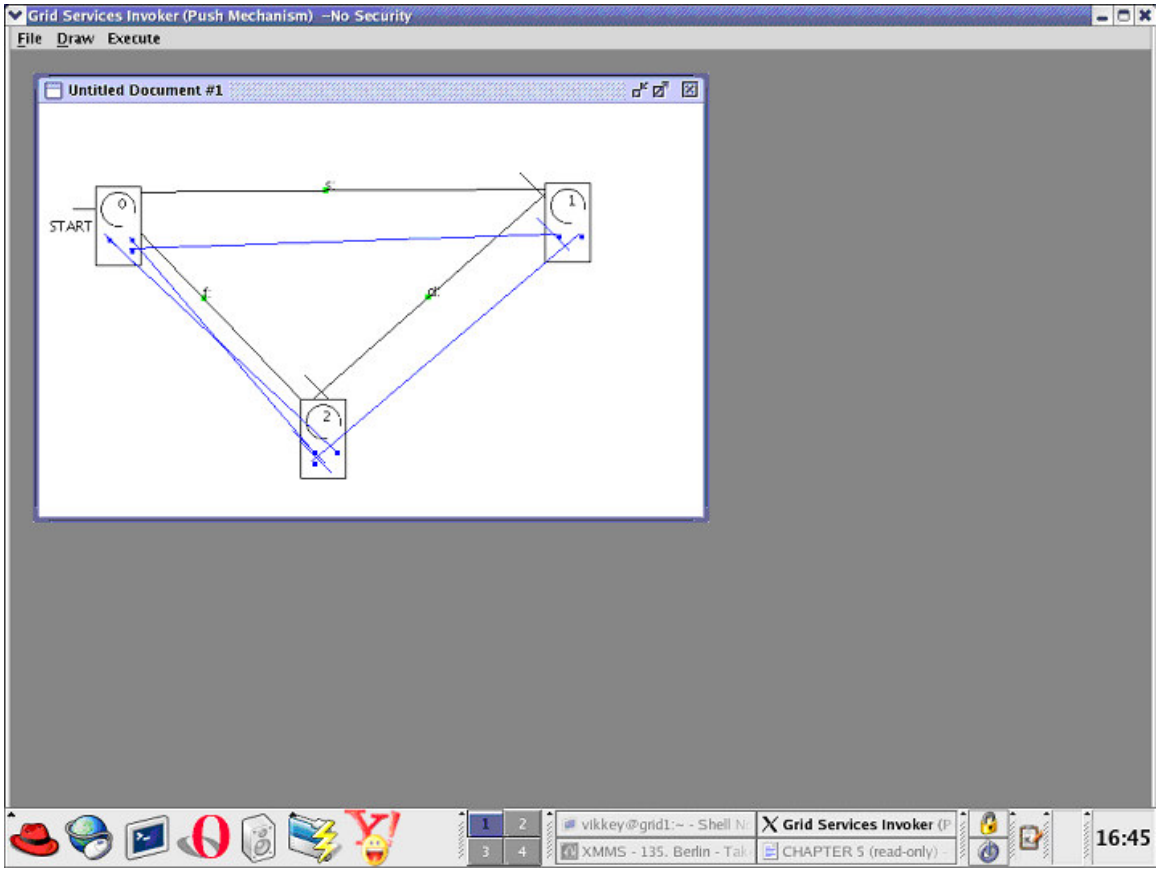
Output files

seed-samples.txt

Description

It checks the calculated value of PI. In case the error is more than threshold it updates the input seed-samples.txt and creates output seed-samples.txt. Also the value of *moreIterations* service data is set to *yes* in that case.

Activation Graph for the Calculation of PI



Work done so far…

- Configuration of **Amber**, the grid enabled cluster and making it a part of the PRAGMA grid
- Study of various technologies
- Design of MDO framework prototype on GT 3.2

Work in progress..

- Adding functionality to get information regarding the services-
  - Design of new tool for obtaining the current service execution status
  - Exploring the existing tools
- Checking integration of applications with third party software
- Investigations on secured service execution

The ultimate aim of our ongoing project is to design an Grid enabled Complete MDO test bed

Main Contributors

- Prof. Arun Agarwal
- Aditya Srivastava
- Saran G.
- Rajeev Wankar
- Neelakanta Reddy P.
- Niranjan T.

Thanks

Demo follows