University of Westminster MTA SZTAKI

Hands-on with the P-GRADE/GEMLCA Portal

NGS Application Developer Course
Edinburgh, 23. February, 2007.

1. **Login to the Portal**

1.1 Go to the URL:
https://grid-portal.cpc.wmin.ac.uk:8080/gridsphere/gridsphere
1.2 Login with the account and password given to you by the tutor

2. **Download a short-term proxy credential from the NGS MyProxy server into the Portal server. The proxy will be used by the workflow manager to access resources of the NGS VO.**

2.1 Hit the “Certificates” tab
2.2 Click on the “Download” button
2.3 Submit the download form with the following data:

Hostname:	*<default>*
Port:	7512
Login:	*<given by the tutor>*
Password:	*<given by the tutor>*
Lifetime:	10
Description:	*<optional>*

2.4 Set your certificate to be used with the NGS Grid.

3. **Create a traffic simulation workflow combining standard and GEMLCA jobs**

The aim of this exercise is to demonstrate the difference between standard and GEMLCA jobs. The first component of your workflow will be a standard job where you define the executable and I/O parameter format. The rest of the workflow will be composed of GEMLCA jobs selected from the central repository of the portal.

The “manhattan” executable file required for the first job can be downloaded from https://gngs-portal.cpc.wmin.ac.uk:8080/gridsphere/html/.

A description of the different components of the traffic simulation workflow can be found here:
http://portal.p-grade.hu/tutorials/induction/Traffic_simulation_workflow_description.htm

3.1. Set your NGS grid environment

Hit the “Settings” tab, then the “Resources” button and load the default resources of the NGS Grid. These resources will be accessible by components of your workflow.

3.2. Create a standard job using the “Manhattan” executable

Hit the “Workflow” tab, then the “Workflow manager” button. Open the workflow editor.

3.2.1 Create a new standard job in the workflow editor. Go into properties by right-clicking on the job and set job type as “Standard job”. Define the job with the following parameters:

Name:	manhattan
Job type:	SEQ
Job executable:	*<Path of the previously downloaded “manhattan” executable>*
Attributes:	-r 10 -c 10 -w 150 -h 150 -C 2 -R 5 -n file.net -t file.trn
Grid:	NGS
Resource:	*<Any NGS resource>*

3.2.2 Define a port for the job with the following parameters:

Port Name: 0
Type: out
Internal File Name: file.net

3.2.3 Define another port to the job with the following parameters:

Port Name: 1
Type: out
Internal File Name: file.trn

3.2.4 Save and submit your workflow. Monitor and visualize the progress of execution.

3.2.5 Download and unzip the result file.

3.3 Extend your workflow with 3 additional GEMLCA jobs

In this exercise you will extend your workflow with already published legacy codes contained in the GEMLCA repository. The executables of these programs are already on the portal server, the I/O parameter sets are already defined. You only have to specify the actual input parameter values.

3.3.1 Open your previous workflow in the workflow editor and create a new GEMLCA job. Go into properties by right-clicking on the job and set job type as “GEMLCA job”. Double click on the job and go into job properties. Define a job with the following parameters:

Name: Sim1
Grid: NGS
Resource: *<Any GEMLCA resource>*
Legacy code: MadCity traffic simulator

Note that parameters with default values and ports are created automatically. Keep the default parameters for this job.

3.3.2 Define another job with the following parameters:

Name: Sim2
Grid: NGS
Resource: *<Any GEMLCA resource>*
Legacy code: MadCity traffic simulator

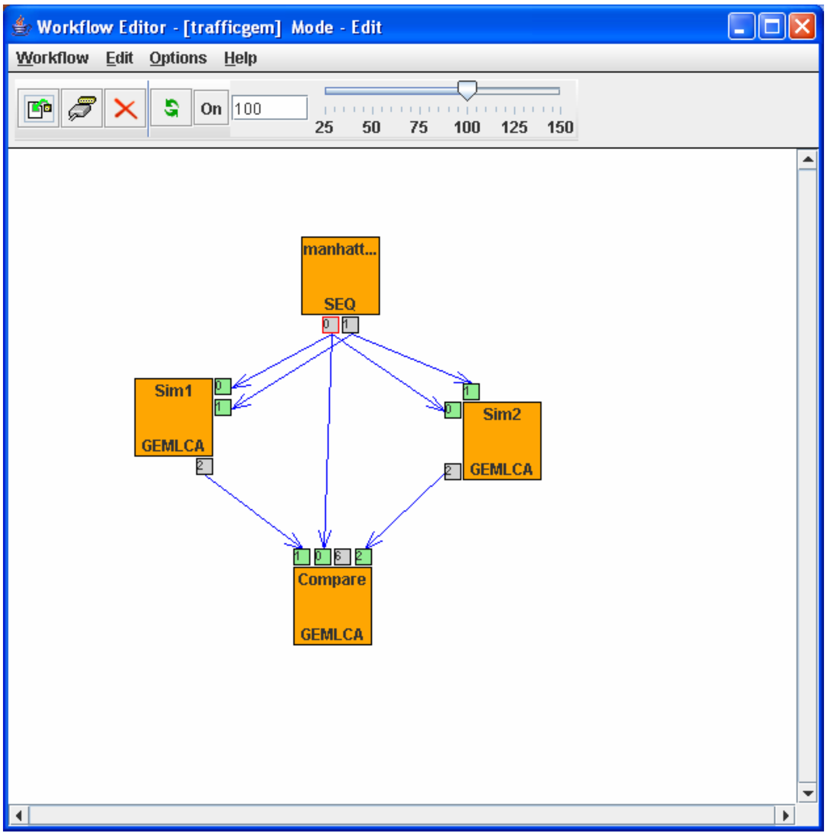
Change the parameter value “number of cars per lane” from 14 to another value.

3.3.3 Define your last job with the following parameters:

Name: Compare
Grid: NGS
Resource: *<Any GEMLCA resource>*
Legacy code: Trace file compare

Keep the default parameter values.

3.3.4 Connect the components of the workflow graph as follows:



3.3.5 Save the workflow, submit it, visualize its execution, and download the results.

4. **Download another short-term proxy credential, this time from the GILDA MyProxy server. The proxy will be used by the workflow manager to access resources of the GILDA VO of the EGEE Grid.**

2.1 Hit the “Certificates” tab
2.2 Click on the “Download” button
2.3 Submit the download form with the following data:

Hostname:	grid001.ct.infn.it
Port:	7512
Login:	*<given by the tutor>*
Password:	*<given by the tutor>*
Lifetime:	10
Description:	*<optional>*

2.4 Set your certificate to be used with the GILDA broker.

5. **Set your GILDA grid environment**

Note that no resources are defined in the GILDA broker grid. This is because the broker keeps track of the all time resource list of the VO.

6. **Create a one job “Matrix multiplication” workflow**

Note: The “Matrix operations” binary C program will be used during the rest of the hands-on. The tutor introduces the program at this point. Further information can be found at: http://portal.p-grade.hu/tutorials/induction/Matrix_operations_program_description.htm

Please download the binary program with two sample input matrixes from this page.

6.1. Create a standard job using the "matrix_operations" executable

Hit the "Workflow" tab, then the "Workflow manager" button. Open the workflow editor.

6.1.1 Create a new standard job in the workflow editor. Go into properties by right-clicking on the job and set job type as "Standard job". Define the job with the following parameters:

Name:	multiply
Job type:	SEQ
Job executable:	*<Path of the downloaded "matrix_operations" file>*
Attributes:	M V
Grid:	*<gilda_LCG_broker or NGS>*
Resource:	*<default>*

6.1.2 Define a port for the job with the following parameters:

Port Name:	0
Type:	in
File type:	local
File:	*<Path of the downloaded "INPUT1" file>*
Internal File Name:	INPUT1 *(case sensitive)*

6.1.3 Define a second port to the job with the following parameters:

Port Name:	1
Type:	in
File type:	local
File:	*<Path of the downloaded "INPUT2" file>*
Internal File Name:	INPUT2 *(case sensitive)*

6.1.4 Define a third port to the job with the following parameters:

Port Name:	2
Type:	out
File type:	local
Internal File Name:	OUTPUT *(case sensitive)*
File storage type:	permanent

6.1.5 Save and submit the workflow. Monitor and visualize the progress of execution.

6.1.6 Download and unzip the result file.

6.2 Define a matrix operations workflow

In this exercise you should define a workflow which computes the following expression:

AB[*, 0]T * AB[*, 1]

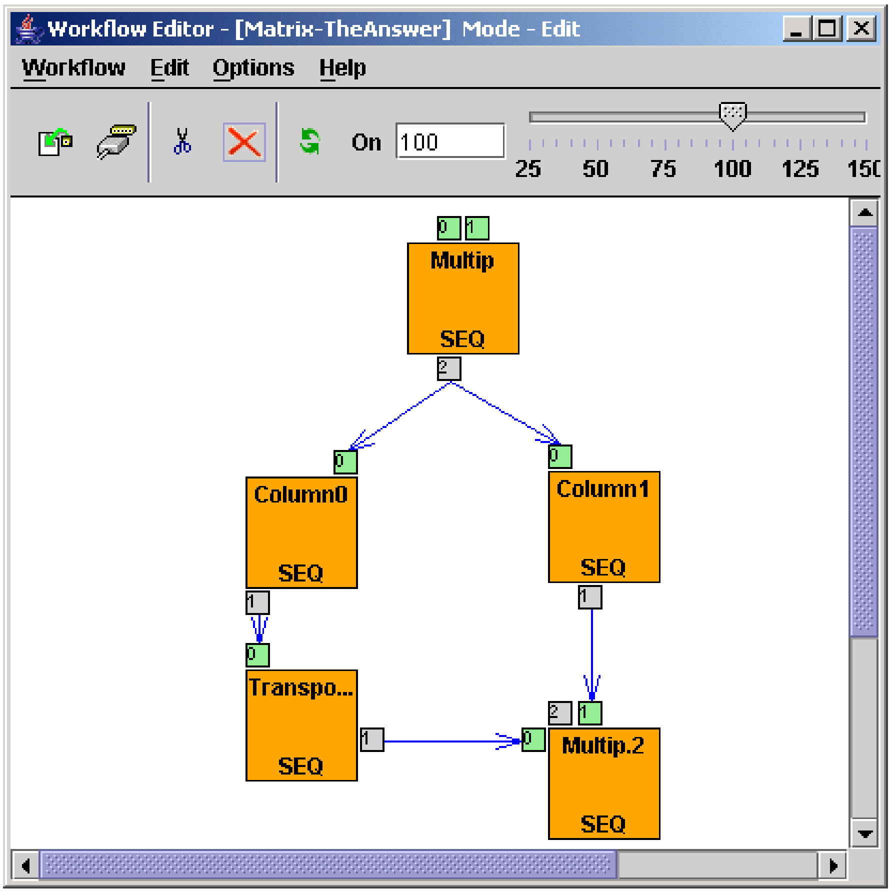
(A and B represent the previously downloaded INPUT1 and INPUT2 matrixes)

Hint: The "matrix_operations" binary reads and produces files in the same format. Add the matrix_operation program 4 times to the Multiply workflow as it is shown in the figure below.

The jobs should compute the following operations:

- Multip: A*B *(command line parameter: M)*
- Column0: A*B[*, 0] *(command line parameters: C 0)*
- Column1: A*B[*, 1] *(command line parameters: C 1)*
- Transpose: A*B[*, 0]T *(command line parameter: T)*
- Multip.2: A*B[*, 0]T * A*B[*, 1] *(command line parameter: M)*

Please use NGS and GILDA resources in a mixed way.



Workflow to compute AB[*, 0]T * AB[*, 1] matrix expression

The complete workflow can be downloaded from:
http://portal.p-grade.hu/tutorials/induction/workflows/Matrix-operations-workflow.tar.gz

Do not forget to reallocate the jobs in the editor to the NGS and/or GILDA VOs, after importing the workflow.