



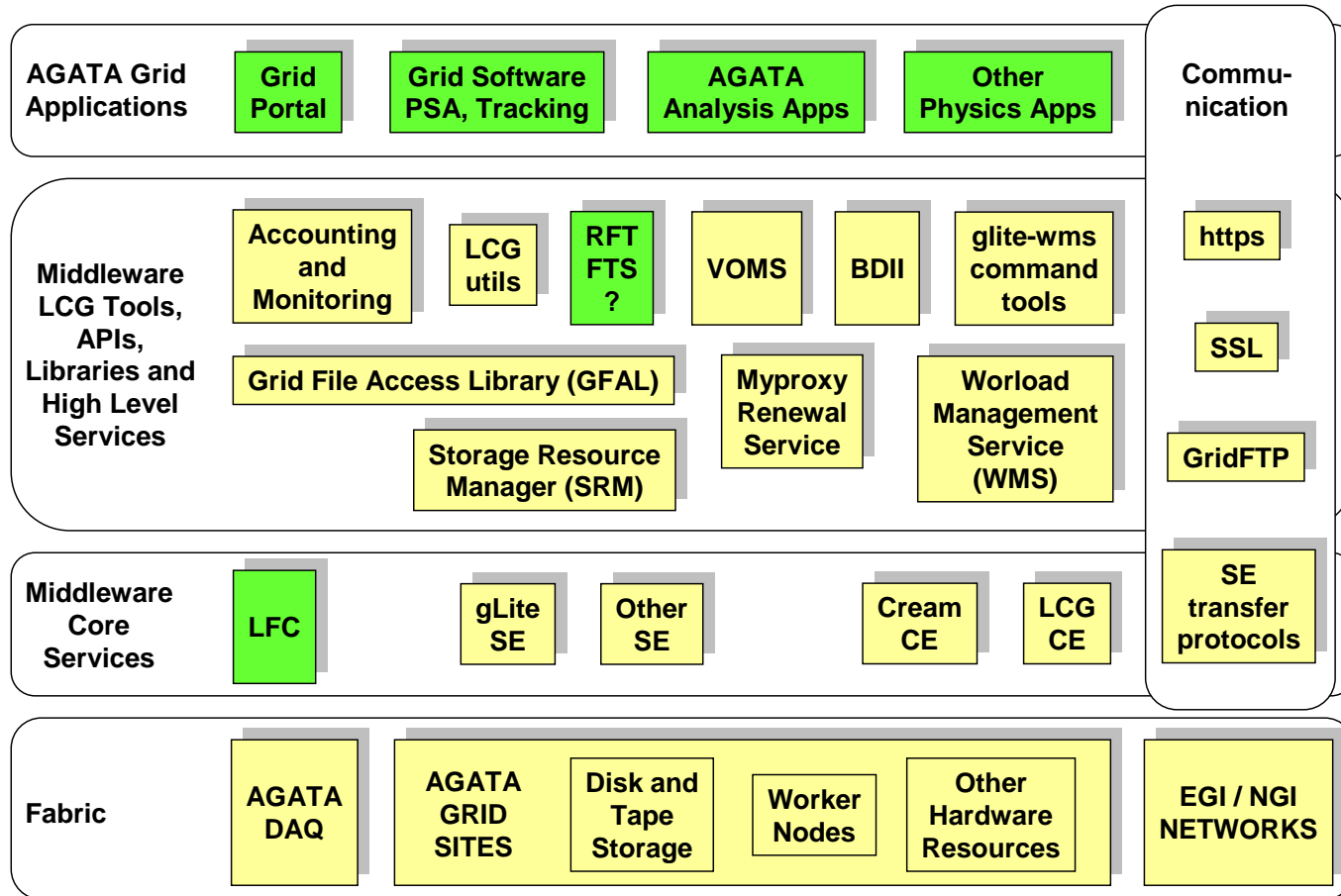
*STATUS OF THE
AGATA DATA PROCESSING
ON THE GRID*

M. KACI

Mohammed.kaci@ific.uv.es

12th AGATA Week, 11-13 June 2012, GSI Darmstadt, Germany

THE AGATA GRID ARCHITECTURE



LEGEND:

General EGI / NGI components

Adapted components

HOW TO RUN A JOB ON THE GRID

I. Login into a User Interface machine and create a valid proxy

```
$> ssh kaci@lcgui.ific.uv.es  
$> voms-proxy-init -voms vo.agata.org
```

II. Write a JDL file and a Script file to be executed on the Grid

Content of a Script file:

- Uncompress the femul software with the right configuration
- Compile the software
- Download the Data files to be processed if any
- Run femul
- Upload the output and log files if proceeds

III. Grid commands for Job management

Submit the Job to the Grid :

```
$> glite-wms-job-submit -a -o jobsID myJDLfile.jdl
```

Follow up the status of the Job:

```
$> glite-wms-job-status -i jobsID
```

Retrieve the outputs:

```
$> glite-wms-job-output -i jobsID
```



ALL THE ACTIONS REPEATED BY USER FOR EACH JOB... OR WRITE SCRIPT FILES

HOW TO RUN A JOB ON THE GRID

I. Login into a User Interface machine and create a valid proxy

```
$> ssh kaci@lcgui.ific.uv.es  
$> voms-proxy-init -voms vo.agata.org
```

II. Write a JDL file and a Script file to be executed on the Grid

Content of a Script file:

- Uncompress the femul software with the right configuration
- Compile the software
- Download the Data files to be processed if any
- Run femul
- Upload the output and log files if proceeds

III. Grid commands for Job management

Submit the Job to the Grid :

```
$> glite-wms-job-submit -a -o jobsID myJDLfile.jdl
```

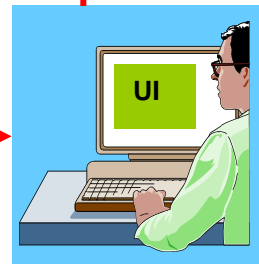
Follow up the status of the Job:

```
$> glite-wms-job-status -i jobsID
```

Retrieve the outputs:

```
$> glite-wms-job-output -i jobsID
```

A
P
P
L
I
C
A
T
I
O
N



ALL THE ACTIONS HANDLED AUTOMATICALLY BY THE APPLICATION

APPLICATION :

NAWAT-AGATA: A USER-ORIENTED GRID APPLICATION DESIGNED FOR AGATA SPECIFIC DATA PROCESSING ON THE GRID

GOAL:

HIDE THE COMPLEXITY OF THE GRID TO THE USER
(THE EXECUTION OF THE TASKS ON THE GRID IS PERFORMED TRANSPARENTLY)

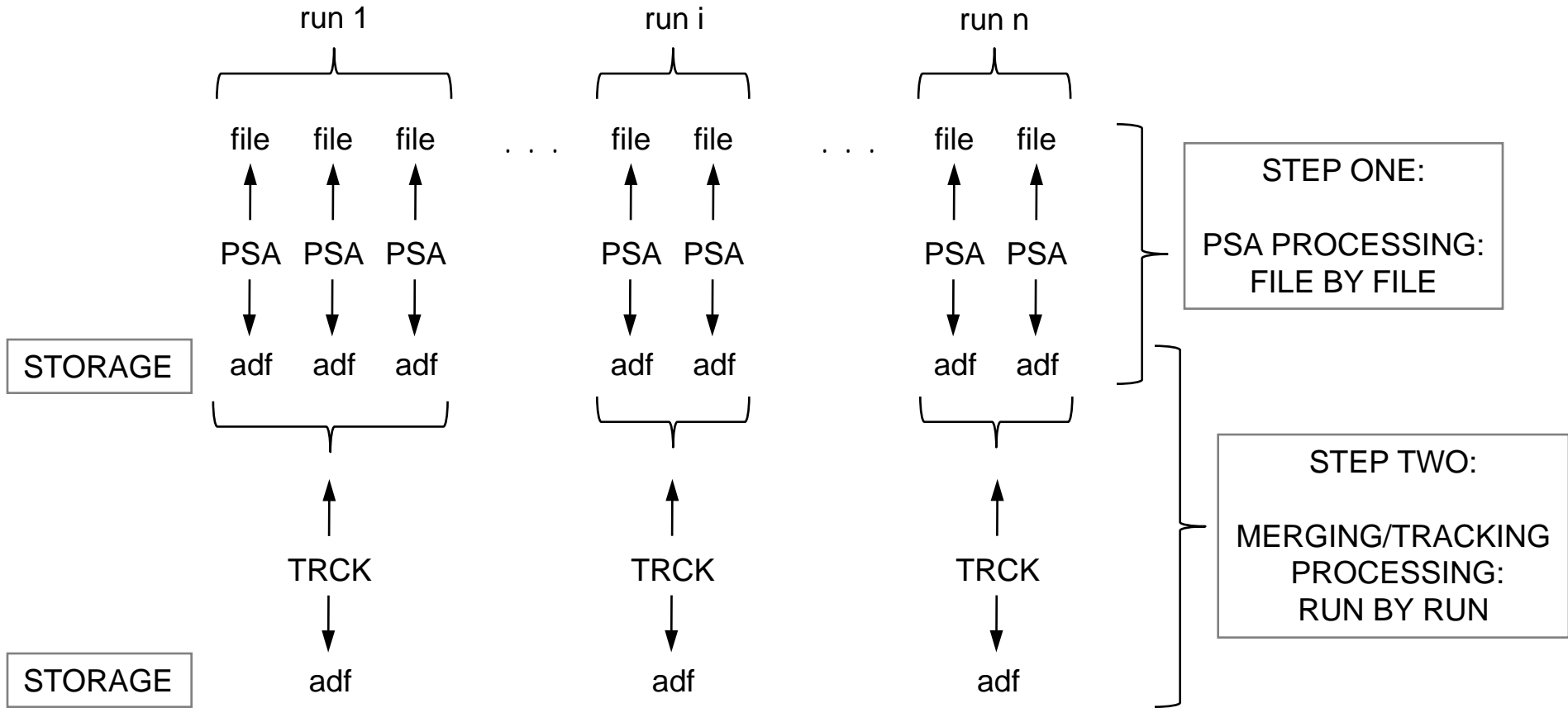
MAKE AUTOMATIC THE EXECUTION OF THE DATA PROCESSING ON THE GRID

FEATURES:

LIGHT-WEIGHT APPLICATION
(COPY FILES AND RUN)

USER FRIENDLY GRAPHICAL USER INTERFACE
(AVOID TYPING LONG COMMAND LINES)

COMPLETELY AUTOMATIC
(DESIGNED AS A ROBOT –AUTOMATE-
HANDLES THE TASKS UP TO THE END WITHOUT INTERVENTION OF THE USER)



THE USER PROVIDES:

In any case:

A Task Configuration File which contains the information about the task to be run on the Grid

InputStorage = srm://<storage-where-the-input-data-are-located>

OutputStorage = srm://<storage-where-to-upload-the-output-adf-files>

NumberOfJobs = <number-of-jobs-to-run-for-this-task>

ProcessType = <PSA-or-TR-or-ANC+TR>

DataAccess = <NONE-or-LUSTRE-or-GFAL>

A ConfExp/ directory that contains the configuration files for each run: Conf-run_xx/

A compressed copy of the femul software (to be installed on the Grid)

In case of PSA processing:

A file that contains the list of mappings Ge/BaseFile

A file that contains the list of the input data filenames (event_mezzdata) to be processed

In case of MERGING/TRACKING processing:

A file that contains the list of the PSA_xx.adf filenames and Ancillary files (vmedata) if proceeds

THEN:

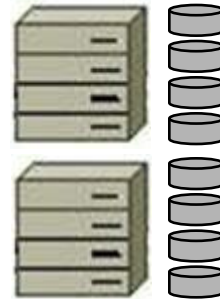
- Launch the nawat-agata application
- Select the Grid CPU resources to be used (Computing Element)
- Click the Execute button
- Go for a coffe or do some other work...

User Interface (UI)



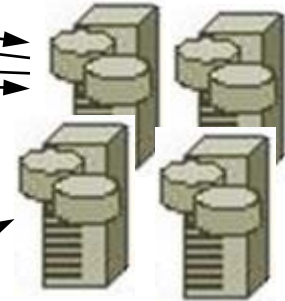
jobs

Worker Nodes (WN)



data

Disk Storage



Nawat-agata runs on the UI:

- Generate the Script file
- Generate the JDL file
- Generate and compress the Config/
- Submit jobs to the Grid
- Follow up their execution
- Retrieve outputs when jobs done

Jobs run on the Grid:

- Uncompress software
- Uncompress Config/
- Download data files, if proceeds
- Compile software (femul)
- Generate Topology
- Update BasicAFP/C
- Run femul
- Upload obtained adf files

Various Instances of femul running simultaneously on the Grid , each instance processing part of the data

ALL RUNS AUTOMATICALLY UNTILL THE TASK IS DONE

Nawat-agata has been run in all AGATA Grid sites:

Using run_0047 of 2011_week29 experiment:

Number of data files processed: 30 (2 to 4 GB per file, total size around 93 GB)

ProcessType: PSA

DataAccess: NONE

Sites:

IPNL :	01:02:21	success	25/30
IPHC :	01:06:25	success	30/30
IPNO :	01:13:41	success	30/30
IFIC :	01:03:17	success	30/30
	00:52:15	success	30/30

with DataAccess = Lustre

Using the whole data of 2011_week29 experiment:

For the following Tasks, around 200 cores was available

PSA data processing of the whole experiment:

373 data files (near 0.5 TB); 01:21:36

MERGING+TRACKING processing:

25 runs; 00:37:43

PSA data processing for calibration files:

600 data files (2.3 TB); 02:55:40; DataAccess = Lustre

947 data files (3.6 TB); 03:57:45; 43 jobs failed.

TO DO



Modify femul to include the direct access to the Storage Element using the GFAL (Grid File Access Library)

Provide Root trees as an output file at the end of the data processing chain



THANK YOU

BACKUP SLIDES

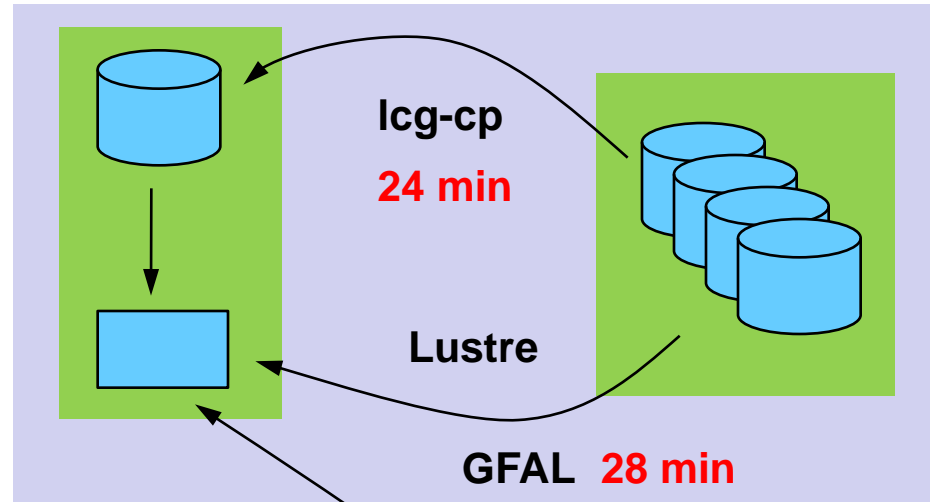
DATA ACCESS

Two triple clusters
1B, 1G, 1R, 2B, 2G, 2R

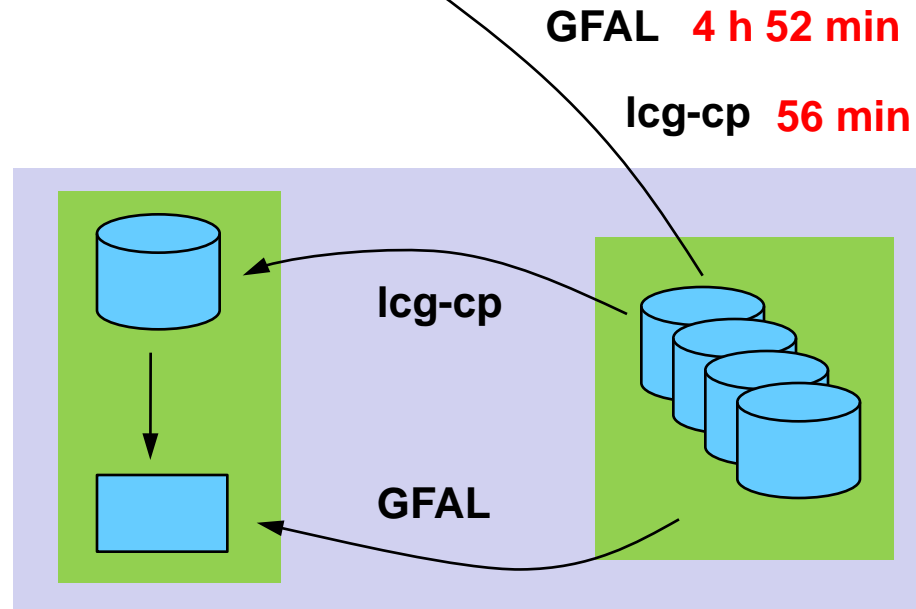
Each file is of 5 GB size

Total of 30 GB processed

PSA+TRACKING processed



SITE A
IFIC



SITE B
IPNL