NorduGrid ARC Tutorial

Arto Teräs <arto.teras@csc.fi> and Juha Lento <juha.lento@csc.fi>

1st Nordic Grid Neigbourhood Conference

Oslo, August 17th 2005

Contents

- **Part 1: Introduction to Grids and NorduGrid**
  - Grid basics
  - NorduGrid collaboration and ARC middleware
  - Authentication and authorization in Grids
  - Grid environment compared to local computing resources
  - Job workflow and ARC user interface
- **Part 2: Hands-on exercises**
  - Submitting jobs in NorduGrid, writing job description files
  - Simple file transfers
  - Monitoring jobs using the Grid Monitor graphical interface

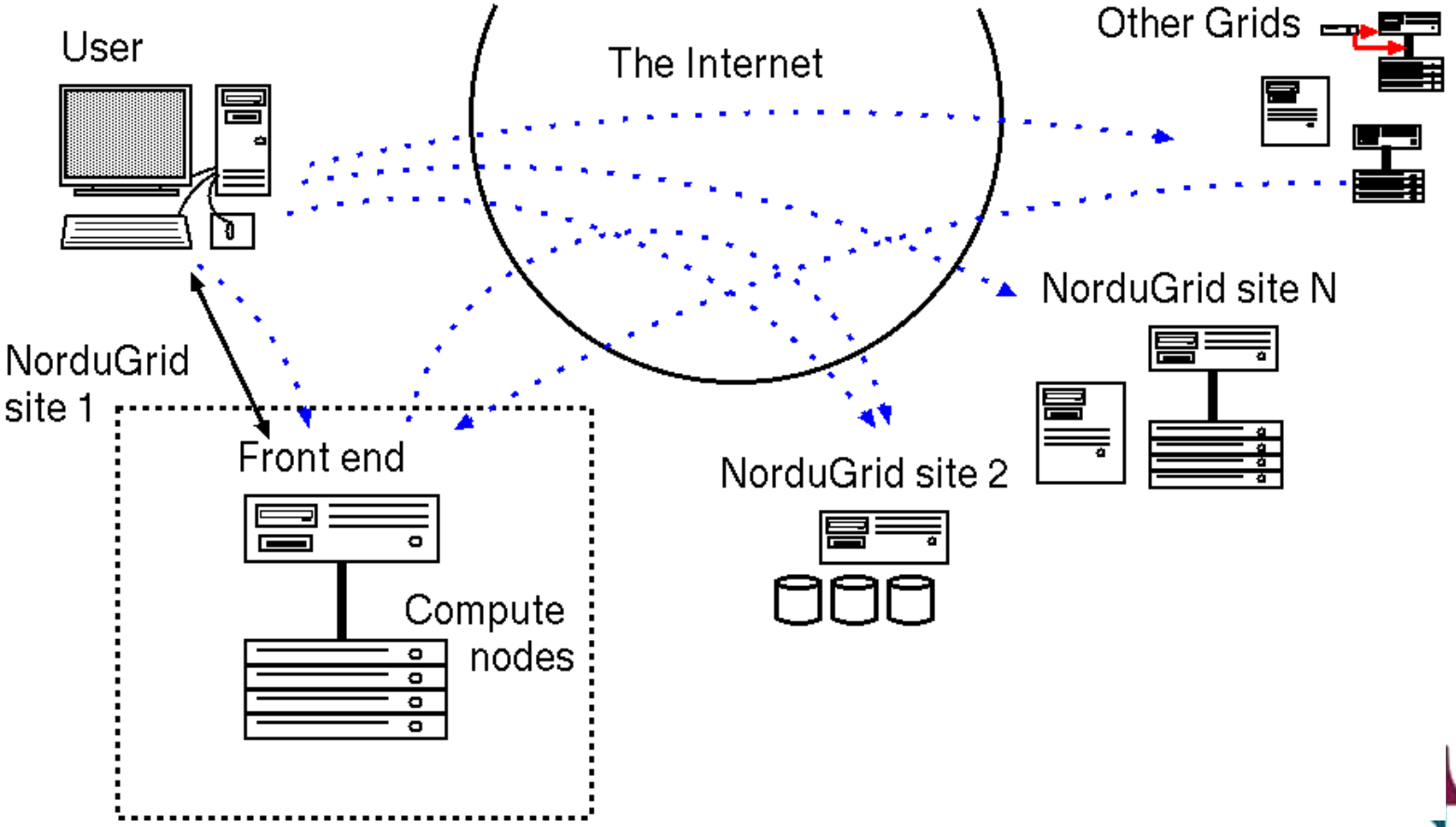
What is Grid?

- **Uniform and secure access to geographically distributed heterogeneous systems**
- **Both the set of users and connected resources vary dynamically**
- **Grids go across multiple administrative domains!**

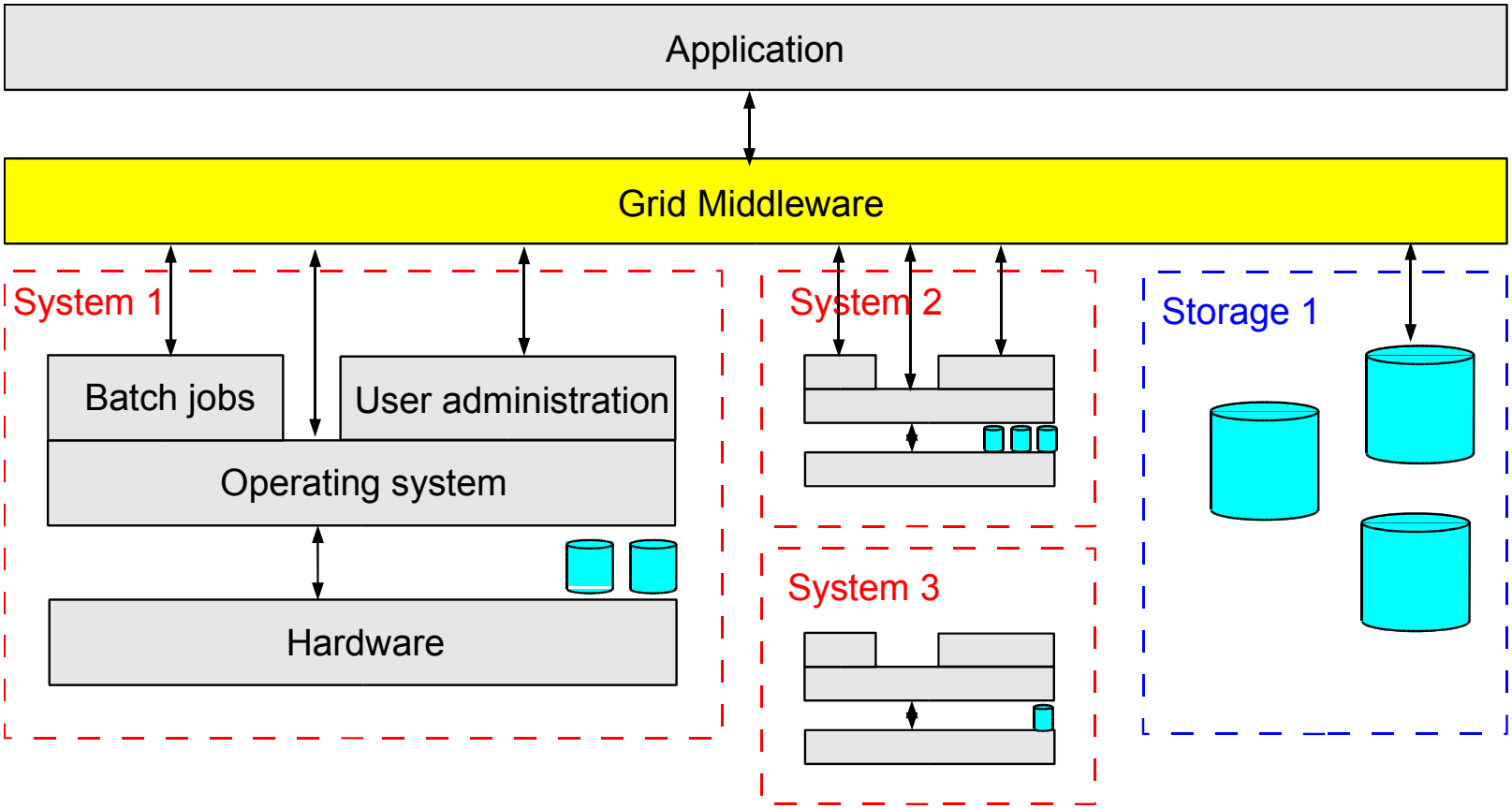
Common Misconceptions

- **Grid increases resources**
  - Popular comparison with the World Wide Web is misleading:
    - One web server may serve a thousand users, but one grid user wants to use a thousand servers...
  - Effective use of resources can bring some savings, but new services and easy access much more important
- **Grid magically binds software together**
  - Vision: Computing power as electricity from the plug
    Reality: still quite far from it
  - If data formats or APIs are incompatible Grid doesn't help
  - Possibility to monitor job execution is important — trying to make a black box easily results in a black hole

Grid Overview



Role of Grid Middleware



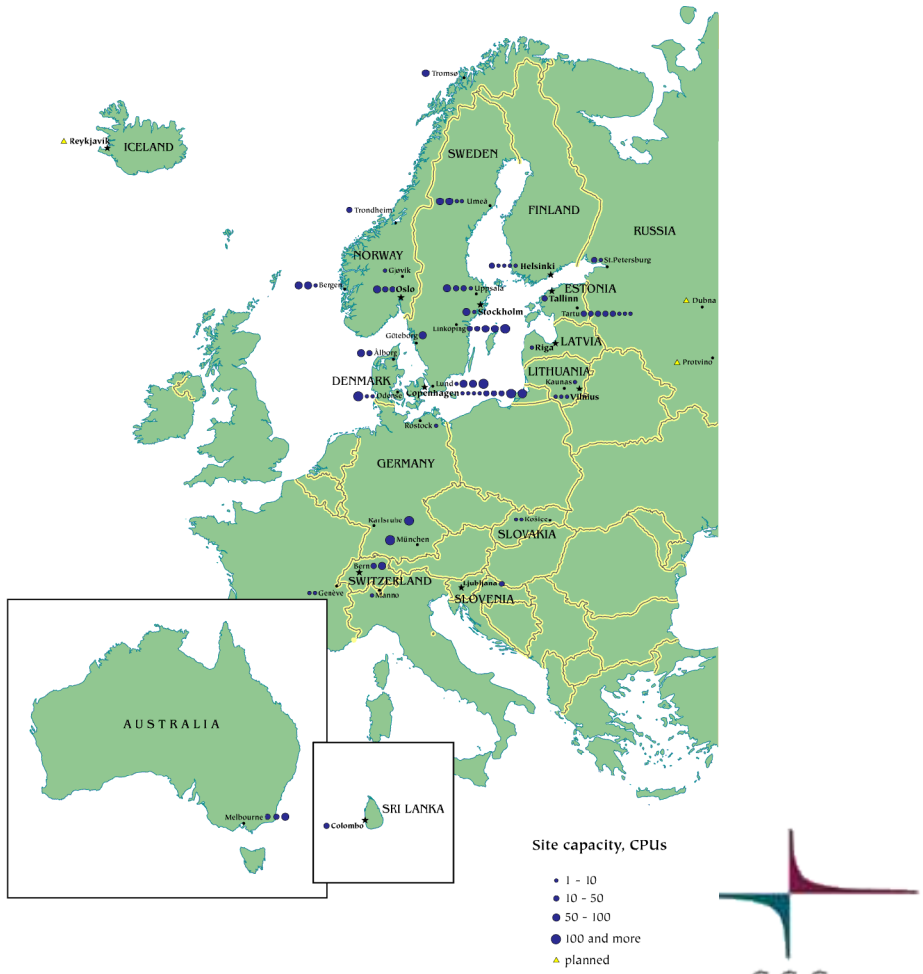
Does One Need to Change Existing Applications?

- **Three different approaches:**

1) Using the application as is: grid middleware will move it and data to target system
   - Link statically, pack all libraries to go with the application or hope for the best...
2) Installing the application on the target system and using it via the Grid interface
   - Batch processing type applications normally work without changes, interactive applications are more difficult
3) Modifying the application to fully exploit a distributed environment
   - Distributing over a large geographical area is not practical unless the computation can be split to independent subtasks

NorduGrid Collaboration

- **Past: Grid pilot project by Nordunet in 2001**
  - Implemented a production Grid system working non stop since May 2002
- **Present: A community around open source Grid middleware: NorduGrid ARC**
  - 13 countries, over 50 sites, 5000+ CPUs
  - Real users, real applications
- **Open for anyone to participate**



NorduGrid ARC

- **ARC (Advanced Resource Connector) is the middleware used and developed by the NorduGrid collaboration**
- **Based on Globus Toolkit™ 2 libraries, can also be compiled against Globus 3.2 or 4.0**
  - Adds services not provided by Globus such as scheduling
  - Extends or completely replaces some Globus components
- **Initial development principles: simple, stable, non-invasive**
  - Resources don't need to be dedicated to the Grid
- **GPL licence**
- **http://www.nordugrid.org**

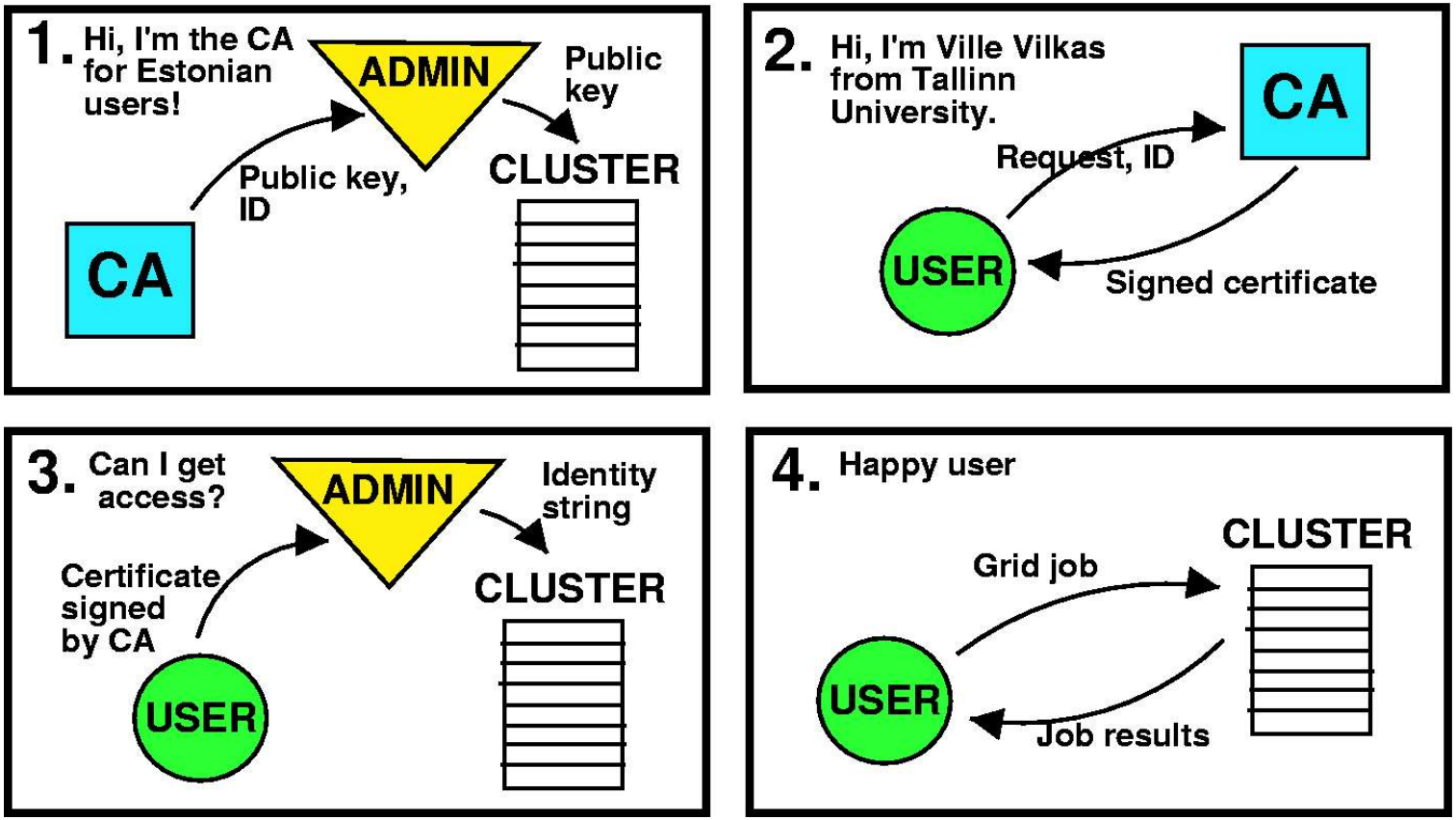
Grid Security

- **”It's like when the PC came...” (Urpo Kaila, manager, information security, CSC)**
  - Grid account is a pass to computers beyond organizational domains!
  - New risks, IT staff often afraid
  - Great power => great damage
- **Security aspects have been considered in the technology**
  - Strong authentication and encryption: no plain-text passwords
  - Identity tied to a certificate: revocation blocks access in the whole Grid
- **Implementation details vary from site to site, be careful if your data is sensitive!**

Authentication

- **Local resources**
  - User name and password
  - “Login” authenticates and usually also authorises to use local resources
- **Grid environment**
  - Authentication based on X.509 certificates granted by a third trusted party, Certificate Authority (CA)
  - Each user has his/her own personal certificate
  - Authentication is separate from authorization => having a valid certificate does not automatically give access to resources

Certificate Trust Chain



Authorization in Grid

- **Users form user groups called Virtual Organizations (VO)**
  - Based on common research area, nationality, funding agency or project
- **Resource providers grant access to VOs**
  - Scales better than managing individual users at every resource
  - Implies trust towards the organization managing the VO

Resource Sharing

- **Many different models are in use**
  - Anarchy: for example local resources in laboratories — relies on solidarity and personal relations
  - Centralized allocation within an organization, organization level agreements
  - Giving away free cycles while local jobs have higher priority: a model used in several NorduGrid clusters
- **Challenges in resource allocation and sharing**
  - User friendliness
  - Maximal resource utilization rate
  - Technical implementation, lack of standards

Grid Environment Compared to Local Resources

- **NorduGrid can in some respects be viewed as an extended batch queue system**
  - Of course it's much more, but we have to start from somewhere...

Local Jobs and Grid Jobs

- **Local batch jobs**
  - Batch queue system options specifying job requirements are usually written to small scripts, defining also directory paths etc.
  - qsub, llsubmit, ...
- **Grid jobs**
  - Described using (extended) Resource Specification Language (xRSL)
  - ngsub
  - Runtime Environments
  - File transfers from the submitting machine or separate file servers on the Grid, Storage Elements (SE)
  - Grid middleware transforms the Grid job to a local batch job

Steps to Start Using NorduGrid

1) **Install the client software**
2) **Request a certificate from a Certificate Authority (CA)**
3) **Install the certificate**
4) **Log in to the Grid**
5) **Write a job description using xRSL language**
6) **Submit the job**
7) **Monitor the progress of the job**
8) **Fetch the results**

Installing the NorduGrid Client

- **Required to submit jobs to NorduGrid**
- **Download from http://ftp.nordugrid.org/download/**
  - Binaries for various Linux distributions, source code also available
- **Easiest way to get started is to install the standalone client**
  - Uncompress in a directory (no root privileges required):
    ```
    $ tar zxvf nordugrid-standalone-0.4.5-1.i386.tgz
    ```
  - Run the environment setup script:
    ```
    $ cd nordugrid-standalone-0.4.5
    $ . ./setup.sh
    ```
- **RPM packages are recommended for multi-user installations**

Requesting and Installing the Certificate

- **Create a certificate request**

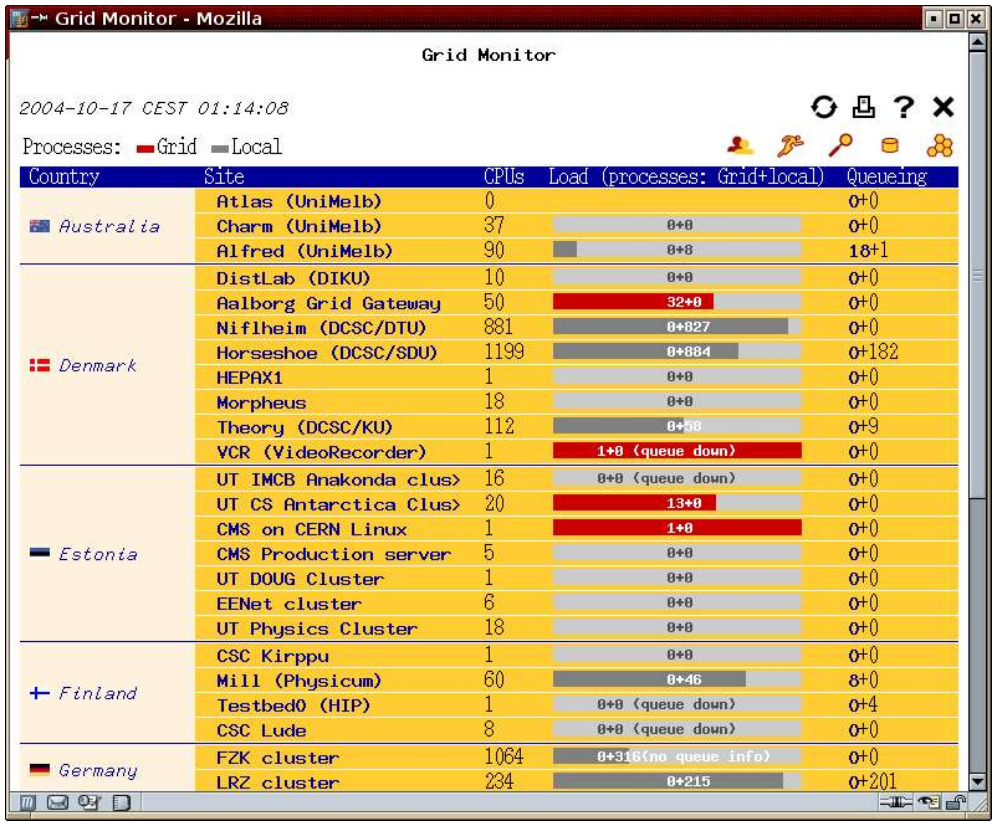
  ```
  $ grid-cert-request -int
  ```

  - Generates a .globus subdirectory with a key and the request
  - Identity string: e.g. /O=Grid/O=NorduGrid/OU=csc.fi/CN=Arto Teras
  - Remember to select a good passphrase and keep the key secret!
- **Send the file `~/.globus/usercert_request.pem` to a Certification Authority (CA)**
  - Check the instructions at your local site / country which CA to contact
- **Wait for an answer from the CA**
  - Signed certificate returned by the Certificate Authority should be saved as file `.globus/usercert.pem`

User Interface

- **ngsub - find suitable resource and start a job**
- **ngstat - check the status of jobs**
- **ngcat - display the stdout or stderr of a running job**
- **ngget - retrieve the results of a finished job**
- **ngkill - stop a job**
- **ngclean - delete a job from a computing resource**
- **ngsync - find users's jobs**
- **ngrenew - update remote credentials (authorization)**
- **ngls - list files on a storage element or in job's directory**
- **ngcopy / ngcp - transfer files to and from clusters and storage elements**
- **ngrequest - third party transfers or data tasks**
- **ngremove / ngrm - delete remote files**

Grid Monitor on NorduGrid Website



Country	Site	CPUs	Load (processes: Grid+local)	Queueing
Australia	Atlas (UniMelb)	0		0+0
	Charm (UniMelb)	37	0+0	0+0
	Alfred (UniMelb)	90	0+8	18+1
Denmark	DistLab (DIKU)	10	0+0	0+0
	Aalborg Grid Gateway	50	32+0	0+0
	Niflheim (DCSC/DTU)	881	0+827	0+0
	Horseshoe (DCSC/SDU)	1199	0+884	0+182
	HEPAX1	1	0+0	0+0
	Morpheus	18	0+0	0+0
	Theory (DCSC/KU)	112	0+58	0+9
	VCR (VideoRecorder)	1	1+0 (queue down)	0+0
Estonia	UT IMCB Anakonda clus>	16	0+0 (queue down)	0+0
	UT CS Antarctica Clus>	20	13+0	0+0
	CMS on CERN Linux	1	1+0	0+0
	CMS Production server	5	0+0	0+0
	UT DOUG Cluster	1	0+0	0+0
	EENet cluster	6	0+0	0+0
	UT Physics Cluster	18	0+0	0+0
Finland	CSC Kirppu	1	0+0	0+0
	Mill (Physicum)	60	0+46	8+0
	Testbed0 (HIP)	1	0+0 (queue down)	0+4
	CSC Lude	8	0+0 (queue down)	0+0
Germany	FZK cluster	1064	0+316(no queue info)	0+0
	LRZ cluster	234	0+215	0+201

- **Shows currently connected resources**
- **Almost all elements "clickable"**
  - browse queues and job states by cluster
  - list jobs belonging to a certain user
- **No authentication, anyone can browse the info**

Basic Job Workflow

- Logging in: `grid-proxy-init`
- Writing a job description file: `emacs job.xrsl`
- Submitting the job: `ngsub`
- Checking the job status: `ngstat / ngcat`
- Retrieving the result files: `ngget / ngcopy`
- Logging out: `grid-proxy-destroy`

Writing a Job Description File

- **Resource Specification Language (RSL) files are used to specify job requirements and parameters for submission**
  - NorduGrid uses an extended language (xRSL) based on the Globus RSL
- **Similar to scripts for local queueing systems, but include some additional attributes**
  - Job name
  - Executable location and parameters
  - Location of input and output files of the job
  - Architecture, memory, disk and CPU time requirements
  - Runtime environment requirements

xRSL Example

- **hellogrid.sh**

```
#!/bin/sh
echo “Hello Grid!”
```

- **hellogrid.xrsl**

```
& (executable=hellogrid.sh)
  (jobname=hellogrid)
  (stdout=hello.out)
  (stderr=hello.err)
  (gmlog=gridlog)
  (architecture=i686)
  (cputime=10)
  (memory=32)
  (disk=1)
```

Basic Operations

- **Submit the job**

```
$ ngsub -d 1 -f hellogrid.xrsl

=> Job submitted with jobid gsiftp://ametisti.grid.
   helsinki.fi:2811/jobs/455611239779372141331307
```

- **Query the status of the job**

```
$ ngstat hellogrid

=> Job gsiftp://ametisti.grid.helsinki.fi:2811/
      jobs/455611239779372141331307
   Jobname: hellogrid
   Status: INLRMS:Q
```

  - Most common status values are ACCEPTED, PREPARING, INLRMS:Q, INLRMS:R, FINISHING, FINISHED

Basic Operations (cont.)

- **Print the job output**

  ```
  $ ngcat hellogrid
  ```

  - shows the standard output of the job
  - this can be done also during the job is running

- **Fetch the results**

  ```
  $ ngget hellogrid

  => ngget: downloading files to
        /home/ajt/455611239779372141331307
     ngget: download successful - deleting job
        from gatekeeper.
  ```

Using a Storage Element

- **Storage Elements are disk servers accessible via the Grid**
  - Can be used to store job output while user is logged out and client machine disconnected from the Grid
- **Allows to store input files close to the cluster where the program is executed, on a high bandwith network**
- **Some files can be local and some remote:**

```
(inputFiles=
  (''input1''. ''/home/user/myexperiment''
  (''input2'', ''gsiftp://se.example.com/files/data''))

(outputFiles=
  (''output'', ''gsiftp://se.example.com/mydir/result1'')
  (''prog.out'', ''gsiftp://se.example.com/mydir/stdout''))

(stdout=''prog.out'')
```

xRSL Example Using a Storage Element

- **xRSL file for the hellogrid example, uploading the job results to a storage element:**

```
& (executable=hellogrid.sh)
(jobname=hellogrid-se)
(stdout=gsiftp://se1.ndgf.csc.fi/ndgf/tutorial/hello.out)
(stderr=gsiftp://se1.ndgf.csc.fi/ndgf/tutorial/hello.err)
(gmlog=gridlog)
(architecture=i686)
(cputime=10)
(memory=32)
(disk=1)
```

Gsincftp

- **Can be used to transfer files to and from storage elements**
  - Based on the popular **`ncftp`** ftp client, but uses certificate based authentication instead of standard ftp authentication
- **Example session:**

```
$ gsincftp se1.ndgf.csc.fi
   ...Logged in to se1.ngdf.csc.fi.
$ cd ndgf/tutorial
$ get hello.out
```

- Already deprecated by the Globus project, does not work with their newest GridFTP server
  - replacement: UberFTP (http://dims.ncsa.uiuc.edu/set/uberftp/)

Runtime Environments

- **Software packages which are preinstalled on a computing resource and made available through Grid**
  - Avoid the need of sending the binary at the start of executing a job
  - Useful if there are many users of the same software or if the same program is used frequently
  - Allow local platform specific optimizations
- **Implemented simply by shell scripts which initialize the environment and are placed in a specific directory**
- **Required runtime environments can be specified in the job description file, for example:**
  `(runtimeenvironment=APPS/GRAPH/POVRAY-3.6)`
- **Runtime Environment Registry: http://www.csc.fi/grid/rer/**

Real Jobs

- **Real jobs usually send several subjobs to the Grid to solve a larger problem**
- **Parallel MPI jobs to a single cluster are supported (if correct runtime environment installed), but no MPI between clusters**
- **Splitting the job to suitable parts and gathering the parts together is left to the user**
  - More error prone environment than traditional local systems => error checking and recovery important
  - Fault reporting and debugging has room for improvements
  - New ARClib API available in the development version

Information Resources

- **Lots of documentation, presentations and tutorials on the NorduGrid web site http://www.nordugrid.org**
  - User guide: http://www.nordugrid.org/documents/userguide.pdf
  - Try out the Grid Monitor!
- **User support mailing list nordugrid-support@nordugrid.org**
- **Technical discussion mailing list nordugrid-discuss@nordugrid.org**
  - Main communication channel between developers
- **Thank you! Questions?**