

Introduction to NorduGrid ARC

Arto Teräs <arto.teras@csc.fi>

Free and Open Source Software Developers' Meeting

Brussels, February 27th 2005



Contents

- **What is Grid?**
- **NorduGrid collaboration**
- **NorduGrid ARC middleware**
- **Live demo and examples**
- **User perspective**
- **System administrator perspective**
- **Developer perspective**

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!**
 - Previously internal operations such as user account management need to be made interoperable with partners
 - Collaboration skills needed to build trust

Common Misconceptions

- **Grid multiplies available resources**
 - Popular comparison with the World Wide Web doesn't work:
 - One web server serves 1000 users => price 0.001x
 - One grid user wants to use 1000 servers => price 1000x
 - Load balancing 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

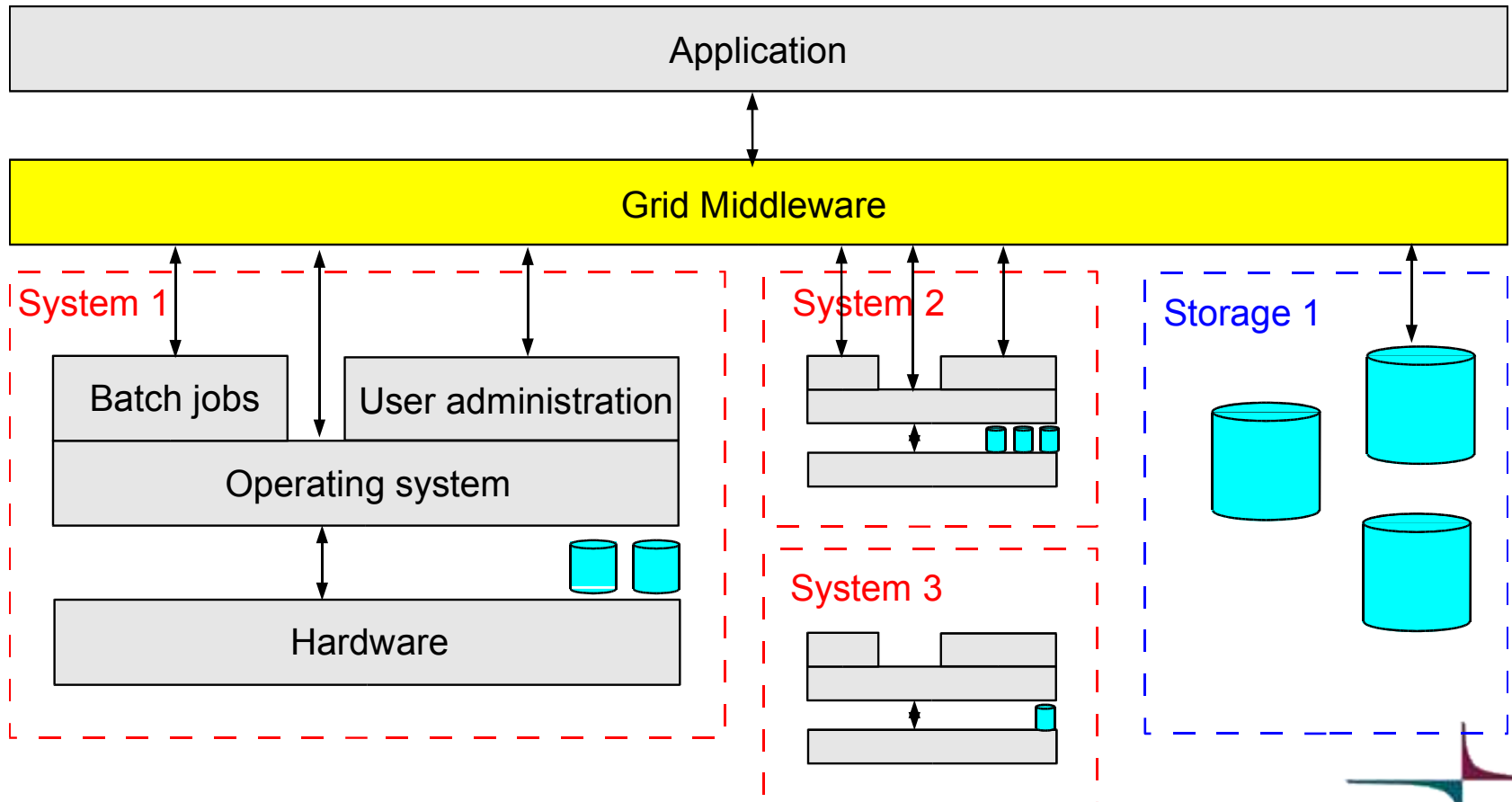


NorduGrid Collaboration

- **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**
 - 10 countries, about 40 sites, 4000+ CPUs
 - Real users, real applications
- **Open for anyone to participate**



Role of Grid Middleware



NorduGrid ARC

- **Let's first see what it looks like (live demo)**
- **ARC (Advanced Resource Connector) is the middleware used and developed by the NorduGrid collaboration**
- **Based on Globus Toolkit™ 2 libraries and services**
 - Adds services not provided by Globus such as scheduling
 - Extends or completely replaces some Globus components
- **Initial development principles: simple, stable, non-invasive**
- **GPL licence**
- **<http://www.nordugrid.org>**



ARC Components

- **Job management**
 - Computing resource (typically cluster) frontend: ARC accepts jobs from clients and passes them to the local batch queue system
 - Client: simple command line user interface with integrated resource broker (**no central scheduler!**)
- **Data management**
 - Storage elements: Disk servers accessible through Grid, indexing services
- **Information system**
 - Information provider components at computing resources
 - Globus MDS (hacked LDAP server) with an extended schema, Grid Monitor web interface to browse the information



User Perspective

- **Unix login and password are replaced by a personal Grid certificate (X.509 public key infrastructure)**
- **Starting computing jobs is very similar to submitting them to a local batch queue system**
 - Some additional attributes in the job description language
 - Different system architectures and types of resources can be accessed through a uniform interface
- **Applications designed to run as batch jobs don't usually need any changes**
 - Compiling statically may help with library dependency problems, which Grid does not make any easier...
- **Hard reality: Grid jobs fail more often than local jobs**



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** - transfer files to and from clusters and storage elements
- **ngrequest** - third party transfers or data tasks
- **ngremove** - delete remote files



xRSL Job Description Example

```
& (executable=rspace-0.81_i386-linux_SERIAL)
  (JobName=CH4_LUCKY)
  (inputFiles=(INPUT ""
    (potentials/C "gsiftp://se.somewhere.com/data1")
    (potentials/H "gsiftp://se.somewhere.com/data2")))
  (outputFiles=(energies ""
    (forces ""
    (WAVES_1 ""
    (POTENTIAL "")))
  (stdout=stdout.txt)
  (stderr=stderr.txt)
  (gmlog=debugdir)
  (CpuTime=10)
  (memory=64)
  (disk=10)
  ( | (architecture=i386)
    (architecture=i686))
```



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
 - 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=povray-3.5)`



Sysadmin Perspective

- **Users are authenticated using X.509 certificates**
- **Authorization is completely separate from authentication**
 - Usually users are managed as groups called Virtual Organizations (VO) and access to a resource is granted to one whole VO at a time
- **Grid users are mapped to local unix accounts**
 - Simplest setup is to map all users on one account but it means no real local security between users (ARC creates random session directory to prevent accidentally using each others' files)
 - Dynamic creation and deletion of accounts for each job is the right solution but not implemented yet
- **Grid access is (normally) authorization to run any binary
=> practically same security risk than a unix shell account**



Sysadmin Wish List

- **It shouldn't take days to install that thing**
 - :-) Prebuilt binaries for all major Linux distributions, source tar.gz for other systems, reasonably simple configuration files
- **Please don't make me change my cluster configuration**
 - :-) ARC respects local setups: no need to dedicate the system to Grid use, installation only on front end (no changes to nodes)
- **Tell me I don't have to change my firewall**
 - :-) Client works well even behind NAT
 - :-) Server needs a number of ports and a port range open
- **Don't ruin my weekend by letting a cracker in**
 - :-) Strong authentication and encrypted connections
 - :-) Distributed systems with many users are always a security risk



Developer Perspective

- **Written in C/C++, Grid Monitor is PHP**
- **Standard GNU tools: autoconf, automake, cvs, Bugzilla**
- **Modular: can add plugins and other extensions, don't need to understand the whole codebase to contribute**
- **Playground: a real system spanning several countries**
 - Need to get a certificate from a reasonably well operated CA (e.g. EUGridPMA members), then can be included in the “guests” VO
- **Friendly development community open to contributions**
- **Recommendation: Get the latest development version**
- **Open source: Globus Toolkit has it's own (sort of BSD like) license, NorduGrid ARC is GPL**



More Information

- **Lots of documentation, presentations and tutorials on the NorduGrid web site <http://www.nordugrid.org>**
 - Try out the Grid Monitor!
- **Read the source: <http://cvs.nordugrid.org>**
- **Technical discussion mailing list nordugrid-discuss@nordugrid.org**
 - Main communication channel between developers
- **NorduGrid technical meetings: next one in Vilnius March 16-19, 2005**
- **Thank you! Questions?**

