The Advanced Resource Connector (ARC) and the NorduGrid

Balázs Kónya
Lund University, Sweden
NorduGrid Collaboration
GGF16, 14th February 2006, Athens
NorduGrid Collaboration

NorduGrid middleware:

Advanced Resource Connector (ARC)

“NorduGrid”: the ARC-enabled Grid(s)

ARC in view of the emerging standards and its relation with other pre-standard middlewares
2001-2002: a research project of the NORDUNET2 program aimed to enable Grid in the Nordic countries. Since end-2002 is a research collaboration between Nordic academic institutes:

- Open to anybody, non-binding

Since end-2003 focuses on middleware:

- Develops own Grid middleware: the Advanced Resource Connector (ARC)
- Provides middleware to research groups and national Grid projects

ARC is now installed on ~50 sites (~5000 CPUs) in 13 countries all over the World.
The NorduGrid Collaboration

From ... ... To

- EDG
- Tesbed
- HEP
- 4 Nordic
- 20 cpu’s
- 2001

>ARC
>50 sites
>+Bio,Chem,...
>13 countries
>5000 cpu’s
>2003

...from a research project to a research collaboration
...from a Grid testbed to a major middleware provider
How did ARC appear?

Back in 2001...High Energy Physics Institutes from Scandinavia wanted to share their computing resources and jointly contribute to CERN/LHC computing

- They needed a Grid!
- The Grid hype just begun
- Globus was regarded as the “de facto standard” middleware

NO production ready middleware was available or seen on the horizon as of November 2001:

- Very alpha Globus GT-2.0 (GRAM-1.5, MDS-2.0) nevertheless Globus & IBM already started to work on OGSA/I, i.e. GT v.3 (which was announced in February, 2002)
- EDG middleware was in an extremely embryotic phase

Since May 2002 ARC has been used in production Data Challenges
A short ARC history

The beginnings: “Grid Book”, Globus jumps into the middle of interests
2000 September: GT-1.1.4 released
2001 January: EDG started up
2001 November: GT-2.0 was announced (very much alpha-quality software)
2002 February: OGSA Initiative
2002 April: long-awaited GT-2.0 was delivered with GRAM-1.5, MDS-2.0 (missing critical functionality, stability problems)

2001 June: First meeting of the NorduGrid developers
2001 September: Grid Testbed with GT-1.1.4, switching to GT-2.0 pre-alpha, evaluating EDG
2002 February: decision to develop an alternative middleware by making use of Globus libraries. NorduGrid design, architecture, philosophy
2002 May: 3rd NorduGrid Workshop, Helsinki demonstration of the first release of the middleware on the Testbed
Design philosophy

1. The system must be:
   a) Light-weight
   b) Portable & modular
   c) Non-intrusive on the resource side:
      • Resource owners retain full control
      • No requirements w.r.t. OS, resource configuration, etc.
      • Clusters need not be dedicated
      • Runs independently of other existing Grid installation
   d) Special attention to functionality & performance

“Traditionally, Scandinavian design has been associated with simple, uncomplicated designs, functionality and a democratic approach”

www.scandesign.org
e) Flexible & powerful on the client part

- must be easily installable by a novice user
- trivial tasks must be trivial to perform
- no dependency on central services
- No central client(s), create a real distributed system

1. Strategy: start with something simple that **works for users** and add functionality gradually

Source of design illustrations: “Scandinavian Design beyond the Myth”
www.scandesign.org
Components overview

NorduGrid ARC
Middleware Components

Goal: no single point of failure
Architecture key points

Each resource has a front-end
- Authenticates users, interprets tasks, interacts with LRMS, publishes information, moves data
- Resources are Grid-enabled by the ARC layer deployed on the front-end, no middleware components behind the front-end!

Each user can have an independent lightweight brokering client (or many)
- Resource discovery, matchmaking, job submission and manipulation, monitoring

Grid topology is achieved by an hierarchical, multi-rooted set of indexing services
Monitoring relies entirely on the information system
Ad-hoc data management, for the beginning
Components: Computing service

Computing resources: Grid-enabled via ARC layer on head node (front-end):
- Custom GridFTP server for all the communications
- Grid Manager handles job management upon client request, interfaces to LRMS
- Performs most data movement (stage in and out), cache management, manages user work area
- Publishes resource and job information via LDAP
Components: Clients

**Client**: a lightweight *User Interface* with the built-in Resource Broker

- A set of command line utilities
- Minimal and simple
- Under the hood: resource discovery, matchmaking, optimization, job submission
- Complete support for single job management
- Basic functionality for multiple job management
- Support for single file manipulations
- Built upon ARCLIB

Portals and GUI clients are being developed
Components: Infosystem

Information System: based on Globus-patched OpenLDAP: it uses GRIS and GIIS back-ends
- Keeps strict registration hierarchy
- Multi-rooted
- Effectively provides a pseudo-mesh architecture, similar to file sharing networks
- Information is only kept on the resource; never older than 30 seconds
- Own schema and providers
**Components: Storages**

**Storage**: any kind of storage system with a disk front-end

- **Conventional Storage:**
  - Own GridFTP server implementation with pluggable back-ends
  - Ordinary file system access
  - Grid Access Control Lists (GACL) based access
- **“Smart” Storage Element**: WS based data service with direct support for Indexing Services (Globus’ RC, RLS)
- no tape storage systems in use so far
ARC: functionality overview

Provides reliable implementation of fundamental Grid services:

- The usual grid security: single sign on, Grid ACLs (GACL), VOs (VOMS)
- Job submission: direct or via matchmaking and brokering
- Information services: resource aggregation, representation, discovery and monitoring
- Implements core data management functionality
  - Automated seamless input/output data movement
  - Data Indexing (RLS, Fireman), client-side data movement
- Job monitoring & management
- Logging service

Builds upon standard open source solutions and protocols

- Globus Toolkit® pre-WS API and libraries (no services!)
- OpenLDAP, OpenSSL, SASL, SOAP, GridFTP, GSI
Few important facts about ARC

- General purpose Open Source European Grid middleware
  - Being developed & maintained by the NorduGrid Collaboration
  - Deployment support
- Lightweight architecture for a dynamic heterogeneous system
- User & performance driven development
  - Production quality software since May 2002
  - First middleware ever to contribute to HEP data challenge
- Middleware of choice by many national academic projects due to its technical merits
  - SWISS Grid(s), Finnish M-Grid, etc...
  - Majority of ARC users are NOT from the HEP community
- Involvement in Interoperability initiatives
  - LCG <-> ARC gateway
- Strong commitment to standards:
  - JSDL, GGF Usage Record support with the next release
At ftp.nordugrid.org:

- **Stable releases, including:**
  - Binary RPMs and tar-balls are available for most Linux platforms
  - Source RPMs and tar-balls
  - **Standalone client tar-ball** for installation by a non-privileged user
    - Only 13 MB when unpacked
    - Pre-configured, out-of-the-box client
    - Contains all the EU Grid PMA approved CA keys
    - Includes all the external dependencies (e.g. Globus):
      comes together with all the basic Globus client tools
  - **Weekly development builds**
  - **Nightly builds**

**CVS at cvs.nordugrid.org**

**License:** GPL

**More info, complete documentation, support (!), contacts at** www.nordugrid.org
NorduGrid: ARC-enabled Grid(s)

Loosely coupled collection of Grid(s) based on ARC middleware
- Initially driven by the needs of the LHC experiments
- Multidisciplinary Grids/VOs: Swiss Bio, Finnish Material Science Grids
- Very low operational costs!!!

Grid of National Grids
- SWEGRID, DCGC, NorGrid, M-Grid, SwissGrid, etc..
- National operational procedures

Easy to Join: assistance in Grid deployment outside the Nordic area
No central operational centre, no centrally forced VOs, large autonomy

Cooperation with other Grid infrastructure projects: interoperability efforts with EGEE
On interoperability

For simplicity, let’s “forget” most of the major middleware (UNICORE, Avaki, SRB, GT4, Condor, etc) and take a look at only on the LCG/Glite vs ARC interoperability issue:

- both are production level middleware
- both are pre-standard implementations (non WS-based systems)

<table>
<thead>
<tr>
<th>Service/component</th>
<th>LCG-2, gLite</th>
<th>ARC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Basic software stack</td>
<td>GT2 solutions from VDT</td>
<td>pre-WS GT libraries, own patches</td>
</tr>
<tr>
<td>Data transfer</td>
<td>GridFTP, SRM v? (DPM)</td>
<td>GridFTP, SRM v1.1 client</td>
</tr>
<tr>
<td>Data management</td>
<td>EDG RLS, Fireman &amp; Co, LFC</td>
<td>RC, RLS, Fireman</td>
</tr>
<tr>
<td>Information</td>
<td>LDAP, GLUE1.1, BDII-GIIS, R-GMA, involved in “Glue2”</td>
<td>LDAP, ARC schema, ARC-GIIS, involved in “Glue2”</td>
</tr>
<tr>
<td>Job description</td>
<td>JDL (based on classAds)</td>
<td>RSL, soon JSDL support</td>
</tr>
<tr>
<td>Job submission</td>
<td>mixture of Condor &amp; GRAM</td>
<td>ARC protocol via GridFTP</td>
</tr>
<tr>
<td>VO management</td>
<td>GSI, VOMS, MyProxy, CAS (?)</td>
<td>GSI, VOMS, GACL</td>
</tr>
</tbody>
</table>
ARC vs. the “de facto middleware”

ARC is built upon the GT2 (pre-WS) libraries and partially makes use of the GT2 framework, BUT

- ARC implements its own set of core Grid services, original GT2 solutions are replaced!
  - No GRAM!, no Globus-Gatekeeper, no Globus-jobmanager, no GT2 MDS schema, no Globus Gridftp-server, no GT2 user tools

- Innovative ARC solutions:
  - Grid-manager, ARC Gridftpd, SSE, ARC Client with Broker, Information model and providers, Monitoring, Logging, XRSL

- ARC is a Globus library-based middleware therefore it heavily depends on pre-WS Globus libraries as an external software
  - Actually, this limits our portability
  - Nordugrid contributed numerous fixes to pre-WS GT
Standardization from ARC's perspective

Areas where standardization & implementations are urgently needed

• Job description language (JSDL?)
• Description of Grid-related objects (Glue?, CIM?)
  – Computing & storage resources
  – Grid jobs (e.g. any standard for job status info?)
• Standard interface to computing resources (OGSA-BES?)
  – Job submission, Job management, Job monitoring
• Standard interface to Storages (SRM?)
• Pre-deployed Grid Application frameworks
• Standards related to Grid economy (e.g. Usage info)
NorduGrid Collaboration develops, maintains & supports an open source Grid middleware
ARC is a reliable, robust, easy-to-use Grid middleware, enabling distributed production facilities already for almost 3 years, non-stop
ARC is a choice of middleware for numerous national Grid projects. These Grids are connected to form a decentralized autonomous Grid
ARC developers are involved in global Grid standardization and interoperability efforts
  – Currently, only site and user certification is standardized, and to some extent – data transfer