NorduGrid and ARC

Andrej Filipčič
Jožef Stefan Institute, Ljubljana
Outline

- NorduGrid and associated projects
- ARC middleware
- Applications
- Development status
About

- **NorduGrid:**
  - A GRID research and development collaboration
  - Development, support and maintenance of ARC grid middleware

- **ARC – Advanced Resource Connector**
  - Started in 2002 as a HEP project
  - ARC 0.6.5 (ARC0) in production

- **KnowARC – EU funded project 2006-9**
  - To extend and re-design the middleware (ARC1)

- **NDGF – Nordic DataGrid Facility**
  - Data storage, transfers for LHC experiments and other projects
  - 24/7 operations, monitoring
Usage

- 17 countries
- 50 clusters
- 35k CPU cores
- 1.5 PBytes of storage
Design principles

- Open source (Apache 2.0 license)
  - Open standards
  - Uses only open components (as few as possible)

- Portability:
  - Runs on most Linux distributions (RedHat, Fedora, SuSE, Debian, Ubuntu, Gentoo, …)
  - On 64-bit systems since 2004
  - OS X, Windows, Solaris support for ARC1 client
Design principles (2)

- Clear separation of local batch system/cluster and grid
  - Plug-ins (perl modules) for batch systems: pbs, SGE, slurm, easy, LoadLeveler, LSF, Condor
    - Resource reporting
    - Grid → local batch job translation
    - Job status
  - ARC frontend → ALL grid-related operations
    - Authorization
    - input/output file handling with automatic caching
  - NO grid middleware on nodes (unless required by users)
    - Jobs are completely local
Design principles (3)

- Resource discovery and brokering encapsulated in the client
  - No single point of failure
  - Redundancy, mobility, scalability
  - Uniform job distribution among available clusters

- Client API (C++, python)
  - Facilitates development of project specific clients (Dulcinea, aCT, ganga, CRAB), portals, etc
Advantages

- Non-intrusive and easy to setup
  - Installation only on the frontend
  - Can co-exist with any other middleware
  - Single configuration file /etc/arc.conf
    - A couple of hours for a beginner with clear instructions

- Efficient Information system:
  - Cluster resources (cores, memory, disk space, authorization, available software...)
  - Top level IS → all available resources
  - Project IS → faster resource discovery
Advantages (2)

- **Runtime Environment (RTE)**
  - Shell scripts for software setup
    - APPS/HEP/ATLAS-15.3.1.1
  - Registered in information system

- **Brokering:**
  - Jobs submitted to fastest clusters with free cpus where the submitter is authorized
  - Only clusters with required RTE and required resources (cputime, walltime, memory, disk space) are selected
Ease of Operation

- Job retries for input preparation, output post-processing failures
- Transparent downtime handling
  - Scheduled (submission blocked)
  - Unscheduled (not registered in IS)
- Low failure rate, mostly due to
  - Cluster failures
  - Network downtimes, project services downtimes
- Operations
  - Little additional effort for system administrators
- Shared, local job space
Ease of Use

- Standalone client
  - 14MB tarball
  - Does not require additional packages
  - Does not require privileges to install

- Job description file (xrsl)
  - Executable (purely local shell script, same as for usual batch jobs)
  - Resource usage
  - input, output files description

- Submission
  - `ngsub -f job.xrsl`
    - Job id: gsiftp://cluster/jid

- Retrieval:
  - `ngget jobid`

- Monitoring:
  - `http://www.nordugrid.org/monitor`
Job Workflow
User Communities

- Middleware of choice:
  - M-GRID (material science project Finland)
  - SwiNG (Swiss Science GRID)
  - SweGRID (Swedish national computational resources)
  - BalticGrid-II
  - SiGNET (Slovenian NGI)

- Bioinformatics
  - Lubeck University

- Healthcare – medical imaging tool
  - University of Geneva

- High Energy Physics – distributed Tier-1 (cloud)

- Many others...
Applications (Oxana Smirnova slide)

Brief applications overview

Disclaimer: information shown here is incomplete and was collected in half an hour by asking people around and googling

- Biophysics
- Biochemistry
- Computational chemistry
- Quantum chemistry
  - GAMESS
- Molecular dynamics
  - GAUSSIAN, DALTON, MOLDEN
- Bioinformatics
  - Taverna
  - BLAST, HMMER
  - eQTL
- Language studies
- Solid state physics
- Computational physics
- Mathematical crystallography
- Informatics, mathematical logic clause solving
- Automatic malware comparison
- Medical imaging
- Simulation of avalanche dynamics
- HEP
  - ATLAS, IceCube, CMS, ALICE, LHCb tested
- CO2 sequestration
- Other materials sciences
Bioinformatics - Taverna

- Polygenic autoimmune disease
- ARC plugin for Taverna → 3-5 times better usage for uniform accessibility and result retrieval
MedGIFT

- Medical images processing tool → used for diagnostics
- 70k images/day in 2007
- Geneva University Hospitals
LHC

- Atlas, CMS, Alice, LHCb
- Distributed Tier-1 (“equivalent” clusters)
- Distributed disk pool servers within NDGF dCache

ATLAS Grid Monitor

2009-06-20 CEST 10:58:53

<table>
<thead>
<tr>
<th>Country</th>
<th>Site Description</th>
<th>CPUs</th>
<th>Load (processes: Grid + Local)</th>
<th>Queueing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Norway</td>
<td>EPF (UIO/FI)</td>
<td>12</td>
<td>0+1</td>
<td>0+0</td>
</tr>
<tr>
<td></td>
<td>Tier1 (BCCS/UIB)</td>
<td>442</td>
<td>0+71</td>
<td>0+19</td>
</tr>
<tr>
<td></td>
<td>Titan A (UIO/USIT)</td>
<td>3840</td>
<td>1596+41477</td>
<td>4143+264</td>
</tr>
<tr>
<td>Slovenia</td>
<td>SIGNET</td>
<td>584</td>
<td>381+4</td>
<td>56+0</td>
</tr>
<tr>
<td>Sweden</td>
<td>Grad (SweGrid, Uppmax)</td>
<td>512</td>
<td>258+176</td>
<td>75+0</td>
</tr>
<tr>
<td></td>
<td>Ritsem (SweGrid, HPC2)</td>
<td>432</td>
<td>432+0</td>
<td>76+0</td>
</tr>
<tr>
<td></td>
<td>Ruth (SweGrid, PDC)</td>
<td>377</td>
<td>388+9</td>
<td>74+1</td>
</tr>
<tr>
<td></td>
<td>Siri (SweGrid, Lunarc)</td>
<td>512</td>
<td>294+118</td>
<td>95+20</td>
</tr>
<tr>
<td></td>
<td>Smokerings (NSC)</td>
<td>504</td>
<td>288+288</td>
<td>58+0</td>
</tr>
<tr>
<td>Switzerland</td>
<td>Bern (UDELIX T3 Cluster)</td>
<td>1120</td>
<td>208+152</td>
<td>420+0</td>
</tr>
<tr>
<td></td>
<td>Geneva ATLAS T3</td>
<td>152</td>
<td>128+0</td>
<td>14+0</td>
</tr>
<tr>
<td></td>
<td>Manno PHOENIX T2</td>
<td>660</td>
<td>487+83</td>
<td>165+10</td>
</tr>
<tr>
<td>TOTAL</td>
<td>12 sites</td>
<td>9447</td>
<td>4622 + 2272</td>
<td>1486 + 294</td>
</tr>
</tbody>
</table>
Atlas production in 2009

- 10% of total production on ARC
- Highest job efficiency (90%)
- Walltime efficiency 90% (was 98.5% in 2008)
  - ND (NDGF) cloud ran MANY test/validation tasks
  - Many bugs in Athena 15.*.* → 2009 performance does not distinguish between software and grid failures

<table>
<thead>
<tr>
<th>cloud</th>
<th>success</th>
<th>failure</th>
<th>success (walltime)</th>
<th>failure (walltime)</th>
<th>efficiency</th>
<th>efficiency (walltime)</th>
</tr>
</thead>
<tbody>
<tr>
<td>BNL</td>
<td>4695178</td>
<td>719392</td>
<td>89154902547</td>
<td>8587522582</td>
<td>86.7%</td>
<td>91.2%</td>
</tr>
<tr>
<td>FZK</td>
<td>2501494</td>
<td>499355</td>
<td>45022820318</td>
<td>5525722159</td>
<td>83.4%</td>
<td>89.1%</td>
</tr>
<tr>
<td>LYON</td>
<td>2482319</td>
<td>359316</td>
<td>43628213861</td>
<td>6829799111</td>
<td>87.4%</td>
<td>86.5%</td>
</tr>
<tr>
<td>RAL</td>
<td>2058529</td>
<td>397741</td>
<td>39666331830</td>
<td>3415684328</td>
<td>83.8%</td>
<td>92.1%</td>
</tr>
<tr>
<td>NDGF</td>
<td>1738060</td>
<td>195420</td>
<td>30985509131</td>
<td>3272778040</td>
<td>89.9%</td>
<td>90.4%</td>
</tr>
<tr>
<td>SARA</td>
<td>925572</td>
<td>342971</td>
<td>15566419196</td>
<td>2404828561</td>
<td>73%</td>
<td>86.6%</td>
</tr>
<tr>
<td>TRIUMF</td>
<td>925999</td>
<td>125224</td>
<td>14088619010</td>
<td>1830032099</td>
<td>88.1%</td>
<td>88.5%</td>
</tr>
<tr>
<td>CNAF</td>
<td>801763</td>
<td>238144</td>
<td>18124515428</td>
<td>208440707</td>
<td>77.1%</td>
<td>89.7%</td>
</tr>
<tr>
<td>PIC</td>
<td>704883</td>
<td>143205</td>
<td>10734418354</td>
<td>1765292584</td>
<td>83.1%</td>
<td>85.9%</td>
</tr>
<tr>
<td>None</td>
<td>283978</td>
<td>412328</td>
<td>7593762255</td>
<td>388006748</td>
<td>40.8%</td>
<td>95.1%</td>
</tr>
<tr>
<td>ASGC</td>
<td>140738</td>
<td>140367</td>
<td>3565818625</td>
<td>297924658</td>
<td>50.1%</td>
<td>92.3%</td>
</tr>
<tr>
<td>CERN</td>
<td>62877</td>
<td>52016</td>
<td>757280306</td>
<td>506230023</td>
<td>54.7%</td>
<td>59.9%</td>
</tr>
<tr>
<td>total</td>
<td>17321090</td>
<td>3625480</td>
<td>3.1888610616e+11</td>
<td>3.69082616e+10</td>
<td>82.7%</td>
<td>89.6%</td>
</tr>
</tbody>
</table>
When production is stable
ND cloud in Atlas

- First week after STEP09 $\rightarrow$ uniform usage
Atlas computing

- EGEE, OSG, NorduGrid
- Panda: central job submission system for production and user jobs
  - Cloud brokering
  - Job distribution
    - Pilot jobs (grid job asks for a real job from Panda server)

- arcControlTower:
  - Pilot job → ARC job
  - Keeps ARC brokering, input file caching
  - Job priorities, production/user job fair-share
  - Job retries for trivial failures, rescheduling
Ganga: cross-grid tool
ARC → gLite WMS

NorduGrid gLite-WMS

JobAdapter.cpp

if(is_condor_resource) {
    jdl::set_grid_type("condor");
    ...
} else if(is_arc_resource) {
    jdl::set_grid_type("nordagrid");
    ...
} else {
    jdl::set_grid_type("globus");
    ...
}
Unique features for Atlas

- Maximal job throughput, no idle CPUs, nodes do not spend any time for transfers
  - Up to 40k jobs per day
- Extremely flexible cpu allocation
  - High usage of opportunistic resources (Titan)
  - Atlas resources available for other projects when not in use
- Custom tasks:
  - NorduGrid used for validation, test, custom tasks on several platforms (SLC4,5... / 32,64-bit / Intel, AMD)
  - Large memory jobs (16GB pile-up for SLHC)
  - Not possible with gLite clouds
- Selectable OS with chroot-ed systems on the same cluster
Atlas impact on ARC

- Fast bug fixes
- Rapid implementation of additional functionality
  - Support for LFC file transfers
  - Dynamic transfer lists
  - Faster job processing/throughput
  - grid-manager scalability → job processing slaves
  - Fair-share for job processing
  - Distributed cache (Nordugrid wide) → Virtual T2
  - Cache-aware job brokering
ARC0 Components
ARC0 Architecture

- GSI based, pre-WS
- Services:
  - gridftpd – job submission, file transfer, gsiftp v2
  - Information System: globus MDS based, will switch to BDII/native LDAP with ARC 0.8
  - Grid-manager:
    - input/output file processing (downloader, uploader)
    - grid job → batch job
What is wrong with ARC0?

- Nothing (robust, efficient, easy to operate and use)
- Need for
  - Standardization
  - Interoperability
  - Extensions and custom services
  - Project oriented plug-ins/services
ARC1 The next generation

• Developed by KnowARC
• Completely new architecture
• The standards (OGF), interoperability
  - GLUE2, JSDL, BES, ...
• Universal Middleware Distribution (UMD) of EGI:
  - gLite
  - Unicore
  - ARC
ARC1 architecture

- Standard interfaces
- WS based
- Modular design
- No third party dependencies
- Portable
- User friendly
- Extensible
- Developer friendly
ARC1 Client

- Dedicated library
  - C++
  - Python wrapper through SWIG

- Implements
  - Credential handling
  - Resource discovery, brokering, retrieval, matchmaking, submission
  - Data transfers

- Plug-in based:
  - Adaptors for different CE types
  - Brokering algorithms
  - Data transfers

- GUI interface
Hosting Environment Daemon

- Container for all server-side functionality
  - Message router
  - Communication between services
- Web Services design
- Pluggable Modules
- Light-weight and fast
- Basic security infrastructure
ARC1 core services

- Information System
- A-REX: next generation grid-manager
  - gsiftp compatibility layer
  - BES, JSDL, GLUE2 support
  - Already available in ARC 0.8 as an alternative for grid-manager
- New ARC storage:
  - Distributed
  - Metadata, collections, replication
  - High level user interface
  - Low level physical storage interface
Prospects

- ARC0 still the production version:
  - 0.8, next stable release, must provide a robust platform for first year of LHC data

- ARC1
  - 0.9 expected within few months
  - Will run IN PARALEL with ARC0 on the same clusters → short path to stabilization and production readiness
  - Atlas can transparently use both at the same time thus providing necessary information to optimize for the stability/throughput and make the switch to the new service on-the-fly
Conclusions

- ARC has proven to be the most reliable and performant grid middleware
- Portability, non-intrusive design and simple setup are welcomed by many clusters which would otherwise refuse to provide grid resources under strict terms of other solutions
- The new ARC architecture will significantly ease the development of custom project-oriented clients as well as dedicated HED based services for project needs
- UMD will help to spread usage of ARC (client, server, mixed grid installations)
- The ARC future seems bright