TUTORIAL

Morris Riedel (FZJ)
m.riedel@fz-juelich.de

August 2006
Outline

- Motivation & History
- Production UNICORE 5
  - Features
  - Architecture
  - Example deployments within DEISA and T-Systems
- UNICORE 6
  - Improved Features
  - Architecture & Higher Level Services
  - Future Developments
- UNICORE as Open Source
- UNICORE Forum
- Summary
Motivation: Scientists & Parallel Computing

- Scientists need computational and storage related resources
Motivation: Scientists & Parallel Computing

- Supercomputers are managed by Resource Management Systems (RMSs) that handle the scheduling

- But: There are many RMSs available

- All proprietary way of job submit
  - IBM Loadleveler → `llsubmit`…
  - Torque Resource Manager → `qsub`…
  - Different job description languages… (# of nodes, memory requirements,…)

Resource Management System
Motivation: Scientists & Parallel Computing

- Solution: Grid System UNICORE
- Define job workflows in abstract manner
- Immediate portability of job definitions for other systems with other architectures
- No ‘learn overhead’ if a new RMS is used
- Applications across multiple supercomputers/clusters → ‘going meta’
Initial UNICORE developments

- UNICORE 08/1997-12/1999
- UNICORE Plus 01/2000-12/2002
- GRIP 01/2002-02/2004
- OpenMolGRID 09/2002-02/2005
From Testbed to Production

Success factor: VERTICAL INTEGRATION

2002
- Different communities
- Different computing resources (super computers, clusters, …)
- Know-how in Grid middleware

2006

Different communities
Different computing resources (super computers, clusters, …)
Know-how in Grid middleware
UNICORE in (mostly) European Projects
Production UNICORE Features

- A vertically integrated Grid middleware system
- Provides seamless, secure, and intuitive access to distributed resources and data
- Used in production and projects worldwide

Features

- Intuitive GUI with single sign-on
- X.509 certificates for AA and job/data signing
- Only one opened port in firewall required
- Workflow engine for complex multi-site/multi-step workflows
- Extensible application support with plug-ins
- Matured job monitoring
- Interactive access with UNICORE-SSH
- Integrated secure data transfer
- Resource management
- Full control of resources remains
- Production quality, ...
Recent Developments

- OGSA-based and WSRF-compliant UNICORE 6 alpha
- Interactive access → UNICORE-SSH
- High-level API for programming Grids
- DRMAA-based access to RMSs
- Comfortable configuration tool
UNICORE Architecture Overview

- Workflow-Engine
- Resource Management
- Job-Monitoring
- File Transfer
- User Management
- Application Support

- similar to /etc/grid-security/grid-mapfile
- similar to Globus jobmanager
  - fork
  - LoadLeveler, (Open)PBS(Pro), CCS, LSF, NQE/NQS, ...
- CONDOR, GT 2.4
UNICORE Client

- Job Preparation
- Usites
- Workflow Management
- Vsites

Job Monitoring
UNICORE 5 Architecture (1)

- Definition of abstract Jobs (not machine-specific)
- Creation of complex workflows (e.g. multi-site jobs for geographically dispersed supercomputers)
UNICORE 5 Architecture (2)

- Client extensions via application-specific plugins
  - Car-Parrinello Molecular Dynamics Plugin (speed up quantum chemical computations)
  - Fluent, Gaussian, MSC Nastran Plugins
  - OpenMolGrid – Plugins (Molecular Science)
- UNIX-Style Command-Line Interface (CLI)

```
[roger@zam439]:~$ uls
ssl://zam461.zam.kfa-juelich.de:9100/AVANT
ssl://zam031.zam.kfa-juelich.de:4014/Linux-zam461
ssl://zam031.zam.kfa-juelich.de:4016/zam777
ssl://zam177.zam.kfa-juelich.de:4005/JUMP
ssl://zam439:4004/u2
ssl://zam439:4004/u1
[roger@zam439]:~$ ucd ssl://zam439:4004/u1/home
[roger@zam439]:~$ uexec sleep.xml
submitted
[roger@zam439]:~$ ups -l
ssl://zam439:4004/u1 1750424334 SUCCESSFUL sleep
ssl://zam439:4004/u1 1750424391 EXECUTING sleep
```
Various plugins have been developed the last years
Base upon the Arcon Client API (Java - API)
UNICORE 5 Architecture (4)

- **Gateway**
  - Single entry-point for UNICORE Sites (Usites)
  - Contains n Virtual Sites (Vsites) (e.g. provides access to n supercomputers)
  - Only **ONE OPEN PORT** in firewall
  - Authentication of users via X.509 certificates
  - Perfect base to group resources and access within Virtual Organizations (VOs)
UNICORE 5 Architecture (5)

- Network Job Supervisor (NJS)
  - Analyse defined Workflows (workflow engine)
    - Multi-site, multi-step Jobs
    - Directed Acyclic Graphs

- Submit of sub-jobs to other sites over Gateway

- Job Incarnation via Incarnation Database (IDB)
  - turn abstract-jobs into machine-specific jobs
  - E.g. location of preinstalled software packages

- Authorization via Unicore User Database (UUDB)
  - turn X.509 user certs to users xlogin of machine
  - The complete certificate is checked (not only DN)
UNICORE 5 Architecture (6)

- **Target System Interface**
  - Represents a real Target System (e.g. a supercomputer or storage server)

- **Interaction with Resource Management Systems**
  - Torque, PBS Pro, Load Leveler, (UNIX fork), …
  - Job submit to Resource Management System
  - Scheduling via Resource Management System
  - Retrieve job results and job output
  - Manage workspace

- **Access to pre-defined Applications**
  - e.g. a massive parallel simulation for collaborative visualization & steering via UNICORE
TSI & Resource Management

- **Target System Tier**
  - Consists of one Target System Interface
  - Interactions with underlying RMSs

- **Target System Interface**
  - Each supported RMS is used in conjunction with a target system-specific TSI impl. in perl
  - Perl TSI implementations are stateless daemons
  - Statically configured before startup (RMS admin)
  - Various TSI implementations
    - PBS TSI (Linux), LSF TSI (SGI), PBS-Pro (HP-UX), NQS (IRIX), LoadLeveler TSI (AIX), …
TSI Framework using DRMAA

- Base upon Open Source SUN Grid Engine DRMAA impl.
  - Java Bindings of DRMAA specification are used
- Java-Based Target System Interface Core
  - Only one layer of UNICORE is exchanged → protocols still the same
- Platform & RMS command independent
  - DRMAA - API “only” provides Interfaces for job submission & management
    - Additional DRMAA Extensions for file transfer
    - Transfer over Unicore Protocol Layer (UPL)
    - Optional transfer over GridFTP

UNICORE 5/6
Consortium of leading national HPC centers in EU

- IDRIS – CNRS, France
- FZJ, Jülich, Germany
- RZG, Garching, Germany
- CINECA, Bologna, Italy
- EPCC, Edinburgh, UK
- CSC, Helsinki, Finland
- SARA, Amsterdam, NL
- HLRS, Stuttgart, Germany
- BSC, Barcelona, Spain
- LRZ, Munich, Germany
- ECMWF, Reading, UK
Deploy and operate a persistent, production quality, distributed, heterogeneous supercomputing environment
UNICORE in DEISA (3)

- Fully-meshed UNICORE infrastructure among partners
- Complex multi-site workflows easily possible
- Heavily used by DECI projects
UNICORE based Access to Computing-Resources. Delivery-Model for DWD, GRS and Team Shosholoza

UNICORE in Business scenarios
UNICORE - LIFE - CD

- Complete “out-of-the-box” usage of UNICORE 5
- Bootable Linux OS with UNICORE 5 pre-installed
- Does not harm your system → Sandbox scenario
- For testing, evaluating, …

AVAILABLE HERE AND AT THE GRID VILLAGE UNICORE BOOTH
Initial Interoperability
UNICORE & Globus 2.4

UNICORE Gateway
VO=DGI

UNICORE
Network Job Supervisor (NJS)
IDB
UUDB

DRMAA-based Java
Target System Interface (TSI) Core
DRMAA-compliant Globus TSI

GRAM Client
GridFTP Client

UNICORE-Client
Standardization necessary
Proprietary Protocols

UNICORE AJO/UPL, Globus RSL

Internet

Compute Resources
Resource Management System

Firewall

Globus 2
GRAM Gatekeeper
MDS
GRAM Job-Manager
GridFTP Server

Uspace

UNICORE-Client

UNICORE

Globus 2.4
UNICORE-Client

Standardization necessary
Proprietary Protocols

UNICORE AJO/UPL, Globus RSL
Emerging Grid Standards (1)

- **OGSA – Open Grid Services Architecture**
  - “The Physiology of the Grid”, GGF - GFD30
  - Resources shared via services in Grids
  - Lifecycle management for Web Services necessary

- **OGSI – Open Grid Services Infrastructure**
  - Globus Toolkit 3 provides early implementation
  - (US Projects using Globus are better funded by US – Gov)

- **WS-RF - Web Services Resource Framework**
  - Re-factors and evolves OGSI to exploit WS-* technologies
  - Same functionality as OGSI, but separated Specifications
Emerging Grid Standards (2)

- Change of Terminology
  - From *Grid Services* …
  - …to *stateful Web Services*
Development of next generation of UNICORE called UNICORE 6 based on OGSA and compliant with WS-RF

- Broader vision of interoperability between different Grid middleware
- Interoperable infrastructures through standards
Using WSDL description for XML tags of doGoogleSearch()

```xml
<soap>
  <soap:header>
    http://api.google.com/search/beta2
  </soap:header>
  <soap:body>
    doGoogleSearch( key = 'Grid' )
  </soap:body>
</soap>
```

- Data – Layer“: SOAP
- Application – Layer: HTTP
- Transport – Layer: TCP
- Internet – Layer: IPv4
- Host-To-Network – Layer : PPP

E.g. HTTP POST REQUEST with SOAP-compliant doGoogleSearch() call.
Understanding SOAP & WS-RF (2)

Host-To-Network – Layer : PPP
Internet – Layer: IPv4
Transport – Layer: TCP
Application – Layer: HTTP
"Data – Layer": SOAP

Client

HTTP POST Request: doGoogleSearch (XML)

Google Server
(http://api.google.com/search/beta2)
doGoogleSearch()

HTTP POST Response: doGoogleSearchResult (XML)

```
Registry / URL
WSDL of Service
```

```
Google Search
```

```
Functionality
```

```
SEARCH
```

```
UNICORE
```

```
"Data – Layer": SOAP
```

```
Application – Layer: HTTP
```

```
Transport – Layer: TCP
```

```
Internet – Layer: IPv4
```

```
Host-To-Network – Layer : PPP
```

```
Network
```

```
```
Understanding SOAP & WS-RF (3)

Client → Registry / URL → UNICORE Server

1. UNICORE Server
   (http://jump.kfa-juelich.de/TargetSystemService)
   submit()

2. Registry / URL
   WSDL of Service

3. HTTP POST Request:
   submit (XML → JSDL)

4. HTTP POST Response:
   submitResult (XML)

5. TargetSystemPort

Host-To-Network – Layer : PPP

Internet – Layer: IPv4

Transport – Layer: TCP

Application – Layer: HTTP

‟Data – Layer“: SOAP

Job Execution

HTTP POST Request: submit (XML → JSDL)

HTTP POST Response: submitResult (XML)
Standardization for interoperability

- Web Services Resource Framework (WS-RF) protocol
  - Add semantics & syntax to operations (GetResourceProperties)
    - Get a list of properties that the service is offering
  - A Web Service itself is typically Stateless → WS-RF stateful
  - Grids need access to stateful resources
    - jobs, supercomputers, telescope, collider,…

- Autonomic behaviour
  - Services know how they can interact
  - Standardized operations & properties
WS-RF Specifications

- Base for others WS-Notifications, WS-Agreement, …
- Five Specifications (public comment - Version 1.2)
  - WS - Resource
    - Relationship Web service and resource
  - WS – Resource Lifetime
    - Lifecycle of a WS-Resource, TerminationTime, etc.
  - WS – Resource Properties
    - Getters/Setter/Queries of Properties
  - WS - BaseFaults
    - Base set of information that appear in fault messages
  - WS – Service Groups
    - Collections of WS or WS-Resources
A WS-RF message via SOAP
(use of WS-Addressing&Security)

Resize
On Demand...
Stateful Web Services & Grids with UNICORE

- 'Grid Services' – Web Services that implement Grid patterns
- Open Grid Services Architecture (OGSA)
  - WS-RF is one implementation of OGSA concepts
  - Access and management of Grid resources of interest
  - Web Services with standardized state interactions
- State of a supercomputer (memory, applications, etc.)
- State of submitted jobs on resources

Size raise a demand for autonomic behaviour
Transforming UNICORE

- Developments of UNICORE 6 alpha

![Diagram of UNICORE architecture]

JSDL, etc.
Job Submission Description Language Example

```xml
<?xml version="1.0" encoding="UTF-8"?>
<JobDefinition xmlns="http://schemas.ggf.org/jsdl/2005/11/jsdl">
  <JobDescription>
    <Application>
        <Executable>/bin/echo</Executable>
        <Argument>hello world</Argument>
      </POSIXApplication>
    </Application>
  </JobDescription>
</JobDefinition>
```

- JSDL Specification ➔ GFD.56
  - November 2005 - Published as GGF Recommendation
New Gateway of UNICORE 6

- Multiple Protocol Support
- Supports UPL
  - For Production UNICORE Installations
- Supports WS-* technologies
  - WS-Addressing
  - SOAP messages / HTTPS
- Retains Unicore Security (!)
  - Single point of entry
  - Client Authentication
  - SSL Connections
UniGrids Atomic Services

- Interoperability between OGSA based Grid systems
- Cross–authentication for UNICORE and Globus

- Enable cross–Grid resource brokering
- Workflows over different Grid Systems
**Interoperability & Standards**

**UniGrids**
- Target System Factory
- Target System Service

**Atomic Services**
- Job Management Service
- Storage Management Service

**Services**
- File Transfer Service

UNICORE and Globus developers

- Feed in standardisation process of GGF via OGSA Basic Execution Services (BES) Working Group
Job Submission Standards

- OGSA - Basic Execution Services (BES)
- Unigrids (Uniform Interface to Grid Services)
  - Developed WS-RF- based UniGrids atomic services (UAS)
  - UAS: basic interfaces for job/file management in UNICORE 6 alpha

<table>
<thead>
<tr>
<th>Target System Factory</th>
<th>Target System Service</th>
<th>Job Management Service</th>
<th>Storage Management Service</th>
<th>File Transfer Service</th>
</tr>
</thead>
</table>

- Execution Services Interface
  - Draft by Argonne (Globus) and Fujitsu (UNICORE)
  - Consists of UAS input and Globus GRAM demands
  - Input to the OGSA - BES GGF working group
- Refactoring of UAS when OGSA-BES is revised
Grid Programming Environment (GPE) Clients

- GPE consists of a set of Grid tools
- e.g. GPE Client Framework for UNICORE 6
  - Client-side for UNICORE 6 and other Grid systems
  - Three different clients for three different usages
  - GridBeans as scientific-area specific Plugins
GPE Clients and GridBean Deployments

UniGrids Atomic Services / ESI / OGSA-BES Client API

Unicorn 6
Globus Toolkit4
China Grid Support Platform
Portable Clients and GridBeans

- GPE Clients work on top of Unicore/GS, GTK4 or ChinaGrid
  - Only requirement: Atomic Service interfaces
- GridBeans are portable without modifications
  - Client API hides underlying infrastructure details
Database Access: OGSA - DAI

- OGSA – Database Access and Integration Services
  - Initially implemented within UniGrids
  - Re-engineering of OGSA-DAI alpha implementation within EU Project OMII - Europe
- OGSA – DAI services hosted in UNICORE 6
Database Access: OGSA-DAI GridBean

GPE Client - OGSA-DAI GridBean

Target Systems
OGSA-DAI GridBean

OGSA-DAI Input
Available data resources:

Data Resource

Refresh

Resource Info
Service address:
https://plenism.tudelft.nl:8889/v1/axs/services/ConfigurableDataService

Version:
OGSA-DAI WSRF-1UnicoreGS 2.1 (SQL)

Name:
DataResource

SQL query:
SELECT * FROM littleblackbook LIMIT 5

Result:

Query invocation was successful

UNICORE
Other Grid Beans for Applications

- BLAST GridBean used by molecular biology scientists to determine the structure and sequence of DNA
- KTA GridBeans provide solutions for an industrial application
  - Application named as the PreStack Depth Migration (PSDM)
  - Application belongs to the Kirchhoff True Amplitude (KTA) methods class for seismic signal processing
  - Runs on machines with MPI
GridFTP File Transfer Service Architecture

- Client – Server transfers
- Server – Server transfers
  - Including SRB
- GFTS GridBean available for clients

Data Transfer of huge files via GridFTP
Collaborative Visualization & Steering (1)

- VISualization Interface Toolkit (VISIT)
  - Light-weight library that supports bi-directional data exchange between visualizations and parallel applications
  - Visualisation application acts as a server: All operations have to be initiated by the simulation
  - SSH Tunnel for using the VISIT protocol for secure communications with short latency (UGSF optionally)

- VISIT / GS Family
  - A higher level service family for collaborative visualizations
  - Hosted within the Unicore/GS WS-RF hosting environment
  - Parallel to the atomic services
Collaborative Visualization & Steering (2)

- **VISIT GridBean**
  - Manage participants of visualization sessions via VISIT / GS family

- **Framework works with all VISIT-enabled applications**
  - Unique Unicore/GS feature for interactive scenarios
  - Not naturally provided by other Grid middleware, e.g. Globus, gLite
  - Used in Astrophysics (nBody), Laser-Plasma Physics (PEPC),…

Life Demo at Grid Village
Initial Portal Clients & Services (1)
Initial Portal Clients & Services (2)

```plaintext
File Edit

texture {
    Bright_Bronze
}

// Other objects: Box, Cone, Cylinder, with more textures
box {
    <-15, -1, 25>, // One corner of the box.
    <-6, 2, 30>   // The opposite corner.
    texture {
        DMFWood1 scale 5
    }
}
cone {

File: //ala3.pov

SECURITY OFF!
```

UNICORE
Initial Portal Clients & Services (3)
Roadmap to UNICORE 6 Production Quality

- New infrastructure based on Web services
  - OGSA-based and WS-RF-compliant
  - Heavily committed to Open Standards from GGF & OASIS
  - Various useful and modern higher level services (e.g. streaming)

- Alpha was released at the end of July 2006
  - [http://unicore.sourceforge.net](http://unicore.sourceforge.net)
  - TBD: Name des Bundles!

- Beta in July 2007

- Final in End 2007 (production quality)
Some Future Developments

- Improving the knowledge oriented scientific/industrial workflow capabilities & usability

- Improving the maturity of atomic and higher level services
  - Job Submission → OGSA - BI
  - Data Access → OGSA – DAI
  - Accounting → RUS and UR
  - VO Management → VOMS
  - Portals → GridSphere

- Improving Interoperability & Interoperation with other Grid systems
UNICORE OPEN SOURCE


- Open Source under BSD license

- Ready-to-use for research and industry

- Strong security

- Easy installation & configuration

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Install</th>
<th>Quit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Host</td>
<td>zam285.zam.kfa-juelich.de</td>
<td></td>
</tr>
<tr>
<td>VSite Name</td>
<td>Fermat</td>
<td></td>
</tr>
<tr>
<td>NJS-Gateway-Port</td>
<td>3826</td>
<td></td>
</tr>
<tr>
<td>Admin Port</td>
<td>3956</td>
<td></td>
</tr>
<tr>
<td>Trusted CAs</td>
<td>/cert/projects-ca-fz-juelich</td>
<td></td>
</tr>
<tr>
<td>Identity</td>
<td>/unicore/njs_identity.p12</td>
<td></td>
</tr>
<tr>
<td>Password</td>
<td>*****</td>
<td></td>
</tr>
<tr>
<td>Installation Dir</td>
<td>/unicore</td>
<td></td>
</tr>
<tr>
<td>java</td>
<td>/usr/bin/java</td>
<td></td>
</tr>
<tr>
<td>perl</td>
<td>/usr/bin/perl</td>
<td></td>
</tr>
<tr>
<td>Logging Level</td>
<td>Configuration</td>
<td></td>
</tr>
<tr>
<td>Keep Uspaces</td>
<td>false</td>
<td></td>
</tr>
<tr>
<td>Operation Mode</td>
<td>full</td>
<td></td>
</tr>
<tr>
<td>Gateway SSL</td>
<td>false</td>
<td></td>
</tr>
<tr>
<td>NJS SSL</td>
<td>true</td>
<td></td>
</tr>
<tr>
<td>Memory</td>
<td>remember</td>
<td></td>
</tr>
<tr>
<td>Save completed AJO</td>
<td>false</td>
<td></td>
</tr>
<tr>
<td>Change Log Files</td>
<td>24</td>
<td></td>
</tr>
<tr>
<td>TSI Worker Limit</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>TSI Update Interval</td>
<td>5000</td>
<td></td>
</tr>
<tr>
<td>Thread Incarnation</td>
<td>3</td>
<td></td>
</tr>
</tbody>
</table>

Configure UNICORE -- Press Alt + <key> for menu entries
UNICORE OPEN SOURCE


- Academic support by FZJ
  - Integration of own results and from other projects
  - Release management
  - Problem tracking, assistance
  - CVS, Subversion, mailing lists, docs

- Viable basis for many other projects
UNICORE FORUM

- Founded by developers, leading EU HPC centres, and supporting hardware vendors as a non-profit association.
- Foster the distribution and use of UNICORE, organize workshops, support presentations at conferences, publish and maintain the specifications, coordinate further development, certify implementations and extensions.
UNICORE is a Grid System from Europe, is open-source and is used in production worldwide (e.g. DEISA)

UNICORE 6 base upon OGSA concepts

Emerging standards such as WS-RF implement OGSA

UNICORE 6 is compliant with the WS-RF OASIS Standard

UniGrids Atomic Service as basic set for job management

UniGrids Atomic Service contribute to ESI & OGSA-BES

UNICORE 6 massively commits to emerging Grid standards

- OASIS WS-RF&WSN, GGF JSDL, W3C WS-Addressing

UNICORE OPEN SOURCE … under BSD license

UNICORE FORUM … supports activities after projects

UNICORE SUMMIT … to meet the UNICORE community
VISIT THE GRID VILLAGE UNICORE BOOTH