

The next-generation ARC middleware

O. Appleton · D. Cameron · J. Cernak · P. Dóbbé · M. Ellert · T. Frågát ·
M. Grønager · D. Johansson · J. Jönemo · J. Kleist · M. Kočan · A. Konstantinov ·
B. Kónya · I. Márton · B. Mohn · S. Möller · H. Müller · Zs. Nagy · J. K. Nilsen ·
F. Ould Saada · Katarina Pajchel · W. Qiang · A. Read · P. Rosendahl · G. Róczy ·
M. Savko · M. Skou Andersen · O. Smirnova · P. Stefán · F. Szalai · A. Taga ·
S. Z. Toor · A. Wäänänen · X. Zhou

Received: 15 September 2009 / Accepted: 15 September 2010 / Published online: 2 October 2010
© The Author(s) 2010. This article is published with open access at Springerlink.com

Abstract The Advanced Resource Connector (ARC) is a light-weight, non-intrusive, simple yet powerful Grid middleware capable of connecting highly heterogeneous computing and storage resources. ARC aims at providing general purpose, flexible, collaborative computing environments suitable for a range of uses, both in science and business. The server side offers the fundamental job execution management, information and data capabilities required for a Grid. Users are provided with an easy to install and use client which provides a basic toolbox for job- and data management. The KnowARC project developed the next-generation ARC middleware, implemented as Web Services with the aim of standard-compliant interoperability.

Keywords Grid · Distributed computing · Middleware · Standardization · Interoperability · Web service

1 Introduction

Many collaborative projects with common data and computing needs require a system to facilitate the sharing of resources and knowledge in a simple and secure way. Some 10 years ago, the high-energy physics community involved in the new detectors of the Large Hadron Collider (LHC) at CERN, the European high-energy physics laboratory outside Geneva, faced very challenging computing demands combined with the need for world wide distribution of data and processing. The emerging vision of Grid computing [1] as an easily accessible, distributed, pervasive resource was embraced by the physicists and answered both technical and political requirements for data processing to support the LHC. In 2002 NorduGrid [2], a collaboration of leading Nordic academic institutions, introduced the

O. Appleton · D. Cameron · T. Frågát · A. Konstantinov ·
J. K. Nilsen · F. Ould Saada (✉) · K. Pajchel · W. Qiang ·
A. Read · A. Taga
University of Oslo, Oslo, Norway
e-mail: farid.ould-saada@fys.uio.no

K. Pajchel
e-mail: katarina.pajchel@fys.uio.no

J. Cernak · M. Kočan · M. Savko
Pavol Jozef Šafárik, Košice, Slovak Republic

P. Dóbbé · I. Márton · Zs. Nagy · G. Róczy · P. Stefán · F. Szalai
NIIF/HUNGARNET, Budapest, Hungary

M. Ellert · B. Mohn · P. Rosendahl · S. Z. Toor
Uppsala University, Uppsala, Sweden

M. Grønager
NDGF, Kastrup, Denmark

D. Johansson
Linköping University, Linköping,
Sweden

J. Jönemo · B. Kónya · O. Smirnova
Lund University, Lund, Sweden

J. Kleist
Aalborg University, Aalborg, Denmark

S. Möller
University of Lübeck, Lübeck, Germany

H. Müller · X. Zhou
University of Geneva, Geneva, Switzerland

M. Skou Andersen · A. Wäänänen
University of Copenhagen, NBI,
Copenhagen, Denmark

Advanced Resource Connector (ARC) middleware as a complete grid solution for the Nordic region and beyond.

As NorduGrid represented stakeholders with highly heterogeneous hardware, it was necessary for its software to run natively on a range of different platforms. They developed a light-weight, non-intrusive solution that respects local policies and assures security, both for the resource providers and the users. ARC [3] was and still is developed following a “bottom-up” approach: to start with something simple that works for users and add functionality gradually. The decentralized services of ARC make the system stable and reliable, and have facilitated the creation of a highly efficient distributed computing resource accessed by numerous users via an easy to use client package.

Like many other middlewares, ARC development started by building on the Globus Toolkit [4]. This first generation of middlewares was quite diverse as there were few standards in the field, making interaction between the Grids difficult. In recent years, there has been a growing awareness and need for interoperability which has resulted in a number initiatives working towards Grid standards that can drive Grid development and allow for interoperability.

The EU-funded KnowARC [5] project developed the next-generation ARC middleware, re-engineered the software into a modular, Web Services (WS)-based, standard-compliant basis for future Grid infrastructures. Although the Grid has been somewhat overshadowed by the emerging remain a highly relevant solution for scientific and business users. ARC in particular is a more ‘cloudy’ solution than other Grid solutions, partly due to the absence of the strict coupling between grid storage and compute element.

2 The ARC middleware design

From the beginning of ARC, major effort has been put into making the ARC Grid system stable by design. This is achieved by avoiding centralized services which could represent single points of failure, and building the system around only three mandatory components:

The computing service, implemented as a GridFTP [6]—Grid Manager [7] (GM) pair of services. The GM which runs on each computing resource’s front-end, is the “heart” of the ARC-Grid. It serves as a gateway to the computing resource by providing an interface between the outside world and a Local Resource Management System (LRMS). Its primary tasks are job manipulation and job related data management which

includes download and upload of input and output data as well as caching of popular data. It may also handle accounting and other essential functions.

The information system serves as the “nervous system” of an ARC-Grid. It is implemented as a distributed database, a setup which gives this important service considerable redundancy and stability. The information is stored locally for each service in a local information system, while the hierarchically connected indexing service maintain the list of known resources. ARC also provides a powerful and detailed monitoring web page [8] showing up-to-date status and workload related to resources and users.

The brokering client is the “brain” of the Grid. It has powerful resource discovery and brokering capabilities, and is able to distribute a workload across the Grid. This interaction between the distributed information system and the user clients makes centralized workload management unnecessary and thus avoids typical single points of failure. The client toolbox provides functionalities for all the Grid services covering job management, data management as well as user and host credentials handling. In addition to the monitoring web page, users have access to detailed job information and real-time access to log files. In order to make the Grid resources easily accessible, the client has been kept very light-weight and is distributed as an easy to install stand-alone package available for most popular platforms.

The ARC middleware is often used in environments where the resources are owned by different institutions and organizations. A key requirement in such an environment is that the system is non-intrusive and respects local policies, especially those related to the choice of platform and security. The Computing Service is deployed as a thin and firewall-friendly Grid front-end layer, so there is no middleware on the compute nodes, and it can coexist with other middlewares. The job management is handed over to the local batch system and ARC supports a long list of the most popular LRMSs (PBS, LSF, Torque, LoadLeveler, Condor, Sun Grid Engine, SLURM). Data management is handled by the GM on the front-end, so no CPU time is wasted on staging files in or out. A job is allowed to start only if all input files are staged in and in case of failure during stage out, it is possible to resume the job from the stage out step at a later point, for example when the problem causing the failure is fixed. This architecture and workflow principles lead to a highly scalable, stable and efficient system showing particularly good results for high throughput and data-intensive jobs.

As a general purpose middleware, ARC strives to be portable and easy to install, configure and operate. One of the explicit goals for the upcoming releases is to reduce external dependencies, like for example on the Globus Toolkit and be included in the repositories of the most popular Linux distributions like Debian and Fedora. By porting both Globus Toolkit and the ARC software to Windows, there is a hope to open ARC to a broader range of user groups, both on the client and the server side.

3 Next-generation ARC

The EU-funded project KnowARC and the NorduGrid collaboration has developed the next generation of ARC. Building on the successful design and the well established components, the software has been re-engineered, with the implementation of the Service Oriented Architecture (SOA) concept and addition of standard-compliant Web Service (WS) interfaces to existing services as well as new components [9]. The next-generation ARC consists of flexible pluggable modules that may be used to build a distributed system that delivers functionality through loosely coupled services. While faithful to the architecture described in Section 2, ARC will at the same time provide most of the capabilities identified by Open Grid Service Architecture (OGSA) road-map of which the execution management, information and data capabilities are the most central. OGSA defines a core set of WS-standards and describes how they are used together to form the basic building blocks of Grids. Some of the capabilities are provided by the novel ARC hosting environment Hosting Environment Daemon (HED) which will be described in the next Section. Figure 1 is an overview of the next-generation ARC architecture which shows the internal structure both of the client and the server side.

KnowARC had a strong focus on the Open Grid Forum [10] (OGF) standardization efforts and both implemented and participated in the development of a range of standards. The project navigated through the fragmented landscape of standards [11], adhering to the usable and complete ones, while in case of incomplete or partly matching standards, still applied them and propagated feedback to relevant Standard Development Organizations (SDO). In case of non-existing standards, ARC developers provided proposals supported by implementation to appropriate SDOs and took an active role in the standard development process. As a result of this commitment, ARC developers are currently major contributors to the GLUE2.0 [12] information specification schema and other OGF working groups.

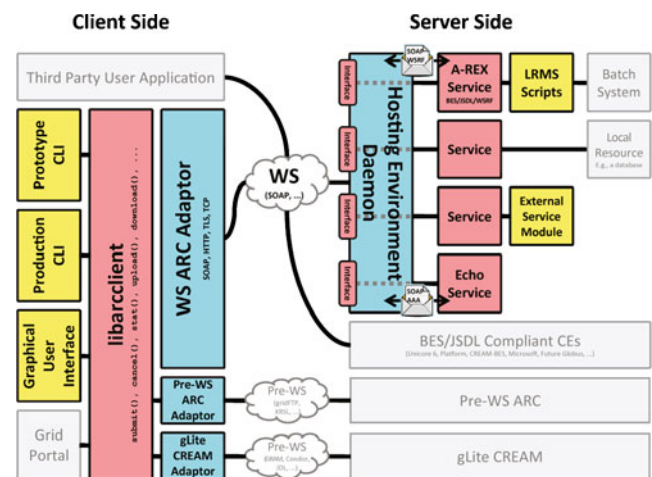


Fig. 1 Overview of the ARC architecture showing the internal structure both of the client and the server side. The client based on the libarcclient is available via a number of interfaces. Plugin adaptors for other target CEs can be easily added. On the server side is structured around the HED container hosting all functional components. The communication is WS based, but there are also mechanisms for pre-WS backwards compatibility

The goal of this effort is to obtain interoperability with other middleware solutions which follow the standards.

3.1 Hosting environment daemon

The next-generation server side ARC software is centered on the Hosting environment daemon (HED) web-service container which is designed to providing a light-weight, flexible, modular and interoperable basis. HED differs from other WS hosting environments in that it is designed to provide a framework for gluing together functionalities and not to re-implement various standards. One of the main functionalities is to be a connection to the outside world and provide efficient inter-service communication. HED supports via the crucial message chain component different levels and forms of communication, from simple UNIX sockets to HTTPS/SOAP messages. This implementation separates the communication related functionalities from the service logic itself. As HED is handling the communication, it also implements the different security policies. Also in this area, ARC has focused on using standards, applying the SSL, TLS, and GSI protocols, authentication mechanisms based on X.509, while VO management is supported using VOMS [13].

3.2 Execution capability

The ARC Resource-coupled EXecution service [14] (A-REX) provides the computing element functionalities, offers a standard-compliant WS interface and

implements the widely accepted basic execution service [15]. In order to provide vital information about service states, capabilities and jobs, A-REX has implemented the GLUE2.0 information schema (to which NorduGrid and KnowARC have been major contributors).

Although KnowARC had a strong focus on novel methodologies and standards, the core of the A-REX service is the powerful, well tested and robust Grid Manager familiar from the pre-WS ARC. Thus, the new execution service implementation imposes the same non-intrusive policies of restricting the jobs to dedicated session directories and avoiding the middleware installation on the compute nodes. A-REX supports a long list of batch systems, offers logging capability and support for Runtime Environments. efficient workflow system where all input and output staging managed by the front-end is preserved. This distinction between the tasks done on the front-end and on the compute nodes has resulted in a highly efficient utilization of the computing resources.

3.3 Information system

The functioning of an ARC-enabled Grid strongly relies on an efficient and stable information system that allows the distributed services to find each other and co-operate in a coherent way, providing what looks to a user as a uniform resource. This special role requires a high level of redundancy in order to avoiding single points of failure. The basic building block is the Information System Indexing System (ISIS) container, implemented as a WS within HED, and in which every ARC service registers itself. Its functionality is twofold. On one hand they work as ordinary Web Services, and on the other hand, they maintain a peer-to-peer (P2P) self-replicating network—the ISIS cloud. Being implemented as a service in HED allows all WS related communication to be delegated the hosting environment and profit from the flexible and uniform configuration system, security framework and built in self-registration mechanism.

The user clients will then query any nearby ISIS service in order to perform the resource and service discovery necessary for the matchmaking and brokering that is a part of the job submission process.

3.4 Data management capability

One of the reasons why the high-energy physics community has embraced Grid technology is its ability to provide a world-wide distributed data storage system. Each of the four LHC experiments will produce sev-

eral petabytes of useful data per year. No institution is capable of hosting all the LHC data needed by a typical research group locally, rather one has to enforce the policy of “sending jobs to data” meaning that all analyses eventually will have to run, at least at some stage, on the Grid.

Impressive as the data volumes are, the advantages of the Grid data storage are not limited to its size. It also offers easy, transparent and secure access to data, which is just as important. In many projects, common data is often the very core of the collaborative work and knowledge sharing.

The ARC middleware has traditionally aimed at high throughput, data-intensive jobs and reliable data management. The next-generation ARC introduced the distributed, self-healing storage system—Chelonia [16]. storage It consists on a set of SOAP-based services residing within HED. The Bartender service provides the high-level user interface and a possibility to access third-party storage systems. The Shepherd is the front-end of the physical storage, while the Librarian manages the entire storage namespace in the A-Hash, a distributed metadata database, thus avoiding often problematic centralized services. The services provide a scalable, consistent, fault-tolerant and self-healing data storage system. Files are grouped in a hierarchy of collections and sub-collections, which conceptually can be thought of as a UNIX-like file system that can be accessed through a root collection functioning as a global namespace. The security, user access and permissions within this hierarchical structure are imposed by well controlled user- or VO-related authorization.

The Chelonia storage system is an independent Grid-enabled system that can be used in three ways. It can be integrated in a Grid job specification as the location of input or output data and handled in an appropriate way by the Computing Element. It can also be viewed as an extended shared file system accessed via two types of client tools. One possibility is the Command-Line Interface offering basic functions like copy, move, list or create a collection. Methods for modifying access and ownership are also available. The second interface is based on the high-level Filesystem in Userspace [17] or FUSE-module which allows users to mount the storage namespace into the local file system enabling the use of graphical browsers and simple drag-and-drop file management.

3.5 Interoperable client

Much of the success of a middleware depends on the user interface. Therefore, ARC strives to implement the principles behind the term Grid and associated

analogies to the uniformity and simplicity of the power grid. The main features of the client have already been described in Section 2. In addition, the focus in the next-generation client is on user friendliness, flexibility and interoperability. The plugin-based libraries facilitate simple integration of support for new Grid job execution services and data access. In order to be standard-compliant, ARC has moved from the extended Resource Specification Language [18] job description to Job Submission Description Language [19]. It has a built-in job description translator and is capable of submitting jobs to both the native ARC job execution services, as well as to the gLite's CREAM [20] and to UNICORE [21] execution services. These are the two main interoperability targets for ARC.

In order to make the Grid resources available for a wide range of users, the client is made available for all popular Linux distributions and a significant effort has been made to port it to Windows and Mac OS.

Developers of third-party applications or backends can easily build directly on the C++ libarcclient or on its Python or Java bindings. The light weight and standalone nature of the client makes it straightforward to include in a software package, e.g., the Ganga job management framework [22] used by CERN physicists in the quest for new physics at the LHC. The ARC job management functionalities are also available via a graphical interface (GUI) and via the web portal Lunarc Application Portal (LAP). Users may also choose between several optimized brokering algorithms, for example brokering based on data availability or fastest CPU.

4 Applications

Since 2002, ARC has been in continuous use delivering production-quality Grid infrastructures. It has been deployed by national Grid infrastructures like the Swiss SwiNG, the Swedish SweGrid, National Ukrainian GRID Infrastructure and by the M-Grid in Finland which also provides the Tier-2 center for the CMS experiment of the LHC.

Due to the excellent performance, ARC is the middleware chosen to power the Nordic Data Grid Facility [23] (NDGF). Using ARC, NDGF is capable of leveraging the existing national computational resources and Grid infrastructures, and create a Nordic large scale scientific Grid computing facility. One of the major projects of NDGF is to operate the Nordic Tier-1 for LHC experiments ATLAS and ALICE and thus be a partner in the world's largest scientific production Grid: the World-wide LHC Computing Grid. The Nordic site

has a unique organization as it is a distributed Tier-1 and its high efficiency and successful performance demonstrate the strength of the ARC middleware.

The ATLAS analysis covering for example searches for the Higgs particle and new physics phenomena is a very demanding computing task. It includes heavy processing of data and Monte Carlo simulations, distribution, and storage of large data volumes, as well as user analysis. In 2008, the NDGF managed system delivered the highest efficiency among the ten ATLAS Tier-1 sites.

Much of modern medical diagnosis is based on the analysis of images which are created in vast quantities at hospitals around the world. KnowARC partner University Hospital in Geneva (HUG) Switzerland, creates some 80,000 images per day alone. Content-based visual information retrieval is the basis for several applications that allow medical doctors and researchers to search through large collections of images, compare them, and obtain information about the particular cases. Image processing is a computing-intensive task which requires a significant amount of resources. In order to shorten the gap between the computing needs and the existing hardware, the multidisciplinary medGIFT team at HUG has developed an ARC-based Grid infrastructure using idle desktop machines which employs virtualization techniques (VMware). Three medical imaging applications (general content-based image retrieval, lung image retrieval, and fracture image retrieval) have been gridified so far. This use-case shows the capability of ARC to create volunteer or cycle scavenging computing infrastructures in the security-sensitive and challenging network environment of a hospital [24].

The Taverna [25] workbench gives biological scientists the means to rapidly assemble data analysis pipelines. KnowARC has provided an ARC client plugin for Taverna that gives its users seamless access to Grid resources, which often in the past have been inaccessible due to the complexity of the system.

5 Development outlook

The new WS-based ARC components are being gradually introduced into the production releases while maintaining backwards compatibility and assuring a smooth transition. At the end of 2009 the NOX release of the ARC, containing only the WS-based components and clients, was made available. In 2010, the 0.8.2 production release of ARC included, for the first time, some of the WS-components, alongside of the classic components, in order to ease deployment on

production facilities and pave the way for eventual migration. Further development and support is carried out by the NorduGrid collaboration and NDGF.

The ARC middleware has been selected by the European Grid Initiative [26] Design Study as one of the three sanctioned middleware solutions for future European-scale Grid computing, and is also a part of the emerging European Middleware Initiative, foreseen as the future common European middleware solution. The standard-based interoperability focus of ARC will play an important role in this context.

In order to prepare for ARC technology take-up, developers have successfully ported essential Grid tools like the Globus Toolkit, LHC File Catalogue [27] and VOMS to Debian and Fedora. These components are also being integrated into Ubuntu through their uptake of Debian components, and are available to distributions such as RedHat Enterprise Linux, CentOS, Solaris, and Scientific Linux via EPEL (Extra Packages for Enterprise Linux), and add-on repository maintained by Fedora. These are also planned to be ported to Windows, extending the Grid awareness of this important platform. In the future, ARC developers aim to make ARC itself an integral part of Linux distributions.

Building on the accumulated experience and expertise through a number of projects both in Grid applications and in middleware development, ARC will continue to provide simple and reliable Grid solutions and work towards standard-based interoperability.

Acknowledgements This work was supported in part by the Information Society and Technologies Activity of the European Commission through the work of the KnowARC project (Contract No.: 032691).

Open Access This article is distributed under the terms of the Creative Commons Attribution Noncommercial License which permits any noncommercial use, distribution, and reproduction in any medium, provided the original author(s) and source are credited.

References

- Foster I, Kesselman C (1999) The Grid: blueprint for a new computing infrastructure. Morgan Kaufmann
- The NorduGrid Collaboration. URL <http://www.nordugrid.org>. Web site. Accessed September 2010
- Ellert M et al (2007) Future Gener Comput Syst 23(2):219. doi:10.1016/j.cam.2006.05.008
- Foster I, Kesselman C (1997) Int J Supercomput Appl 11(2):115. Available at: <http://www.globus.org>. Accessed September 2010
- EU KnowARC project. URL <http://www.knowarc.eu>. Web site. Accessed September 2010
- Allcock W et al (2002) Parallel Comput 28(5):749
- Konstantinov A The NorduGrid Grid Manager And GridFTP Server: description and administrator's manual. The NorduGrid Collaboration. URL <http://www.nordugrid.org/documents/GM.pdf>. NORDUGRID-TECH-2. Accessed September 2010
- The NorduGrid Monitor. URL <http://www.nordugrid.org/monitor/>. Monitor web site. Accessed September 2010
- KnowARC Design Document, KnowARC Deliverable D1.1-1, 2007
- Openl Grid Forum. URL <http://www.ogf.org/>. Web site. Accessed September 2010
- KnowARC Standards Conformance Roadmap, KnowARC Deliverable D3.3-1, 2006
- Andreozzi S et al (2009) GLUE Specification v2.0. URL <http://www.ogf.org/documents/GFD.147.pdf>. GFD-R-P.147. Accessed September 2010
- Alfieri R et al (2005) Future Gener Comput Syst 21(4):549
- Konstantinov A The ARC computational job management module—A-REX. URL <http://www.nordugrid.org/documents/a-r-ex.pdf>. NORDUGRID-TECH-14. Accessed September 2010
- Foster I et al (2007) OGSA™ Basic execution service version 1.0. URL <http://www.ogf.org/documents/GFD.108.pdf>. GFD-R-P.108. Accessed September 2010
- Nagy Z, Nilsen J, Toor SZ (2009) Chelonia—self-healing distributed storage system. URL <http://www.nordugrid.org/documents/arc-storage-documentation.pdf>. NORDUGRID-TECH-17
- Filesystem in Userspace. URL <http://fuse.sourceforge.net/>. Accessed September 2010
- Smirnova O (2008) XRSL (Extended resource specification language). URL <http://www.nordugrid.org/documents/xrsl.pdf>. NORDUGRID-MANUAL-4. Accessed September 2010
- Anjomshoaa A et al (2008) Job Submission Description Language (JSDL) Specification, Version 1.0 (first errata update). URL <http://www.gridforum.org/documents/GFD.136.pdf>. GFD-R.136. Accessed September 2010
- Aiftimiei C et al (IOP, 2008) In Proc. of CHEP 2007, J Phys Conf Ser 119:062004, In: Sobie R, Tafirout R, Thomson J. (eds) URL <http://dx.doi.org/10.1088/1742-6596/119/6/062004>. Accessed September 2010
- UNICORE, Uniform Interface to Computing Resources. URL <http://www.unicore.eu>. Web site. Accessed September 2010
- Moscicki JT et al (2009) Comput Phys Commun 180(11):2303. URL <http://arxiv.org/abs/0902.2685v1>. Accessed September 2010
- Nordic DataGrid Facility. URL <http://www.ndgf.org>. Web site. Accessed September 2010
- Zhou X, Pitkanen MJ, Depeursinge A, Müller H (2009) A medical image retrieval application using grid technologies to speed up feature extraction in medical image retrieval. Philippine Journal of Information Technology. URL <http://publications.hevs.ch/index.php/publications/show/861>
- Hull D, Wolstencroft K, Stevens R, Goble C, Pocock MR, Li P, Oinn T (2006) Nucleic acids res 34 (Web Server issue). doi:10.1093/nar/gkl320. URL <http://dx.doi.org/10.1093/nar/gkl320>. Accessed September 2010
- European Grid Initiative. URL <http://www.eu-egi.eu/>. Web site. Accessed September 2010
- LHC File Catalog. URL <https://twiki.cern.ch/twiki/bin/view/EGEE/GlitéLFC>. Web site. Accessed September 2010