

University of Groningen

Astro-WISE and Grid

Begeman, Kor; Belikov, Andrey N.; Dijkstra, Fokke; Meyer-Zhao, Zheng; Valentijn, Edwin A.; Vriend, Willem-Jan

Published in:
Experimental Astronomy

DOI:
[10.1007/s10686-012-9293-2](https://doi.org/10.1007/s10686-012-9293-2)

IMPORTANT NOTE: You are advised to consult the publisher's version (publisher's PDF) if you wish to cite from it. Please check the document version below.

Document Version
Publisher's PDF, also known as Version of record

Publication date:
2013

[Link to publication in University of Groningen/UMCG research database](#)

Citation for published version (APA):

Begeman, K., Belikov, A. N., Dijkstra, F., Meyer-Zhao, Z., Valentijn, E. A., & Vriend, W.-J. (2013). Astro-WISE and Grid. *Experimental Astronomy*, 35(1-2), 355-366. <https://doi.org/10.1007/s10686-012-9293-2>

Copyright

Other than for strictly personal use, it is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license (like Creative Commons).

Take-down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Downloaded from the University of Groningen/UMCG research database (Pure): <http://www.rug.nl/research/portal>. For technical reasons the number of authors shown on this cover page is limited to 10 maximum.

Astro-WISE and Grid

**Kor Begeman · Andrey N. Belikov ·
Fokke Dijkstra · Zheng Meyer-Zhao ·
Edwin A. Valentijn · Willem-Jan Vriend**

Received: 20 June 2011 / Accepted: 16 February 2012 / Published online: 8 March 2012
© The Author(s) 2012. This article is published with open access at Springerlink.com

Abstract The paper reviews the Astro-WISE infrastructure and demonstrates that the Astro-WISE Information System provides a Grid itself. We describe the integration of Astro-WISE with an external Grid infrastructure (BiGGrid). The integration is performed on all infrastructural layers (data storage, meta-data and processing layers) with Astro-WISE as a “master” infrastructure. We report the use of the integrated infrastructure for the processing of Astro-WISE hosted data and for the future development of Astro-WISE and Target projects.

Keywords Data grid · Grid computing · Information system · Middleware

K. Begeman · A. N. Belikov (✉) · E. A. Valentijn · W.-J. Vriend
Kapteyn Astronomical Institute, University of Groningen, Landleven 12, 9747AB,
Groningen, The Netherlands
e-mail: belikov@astro.rug.nl

K. Begeman
e-mail: kgb@astro.rug.nl

E. A. Valentijn
e-mail: valentyn@astro.rug.nl

W.-J. Vriend
e-mail: wjvriend@astro.rug.nl

F. Dijkstra · Z. Meyer-Zhao
Donald Smits Center for Information Technology, University of Groningen, Nettelbosje 1,
9747AJ, Groningen, The Netherlands

F. Dijkstra
e-mail: f.dijkstra@rug.nl

Z. Meyer-Zhao
e-mail: Z.Zhao@rug.nl

1 Introduction

In the last decade the use of Grid became popular for astronomical data processing. SETI@home project¹ was one of the first infrastructures developed for an astronomical application providing a computational Grid. It is one of the first examples of a distributed data processing grid, using cpu's all over the globe, allowing to combine computing power of “nodes” as light as a laptop.

Astro-WISE is an astronomical information system which was created to process the data of ESO's VLT Survey Telescope (VST) and particularly one of ESO's public surveys—the Kilo Degree Survey (KiDS). Astro-WISE is much more than just a pipeline and data storage for one particular survey. It is an integrated data storage and data processing system, which allows the user to produce science-ready results from the raw images of a wide range of astronomical instruments maintaining all dependencies and processing parameters within the system. The data processing in Astro-WISE is optimized to prevent unnecessary multiplication of the data. Astro-WISE employs a system of Authorization and Authentication (allows users to form groups and share the data within groups) and maintains full data lineage (avoid reprocessing of the same data entity with the same processing parameters) to optimize the growth of the processed data in the system. Nevertheless, users are allowed to create their own specific surveys and combine them with a number of other surveys and catalogs keeping all the data in private data storage space.

Astro-WISE should function as an open system in the sense that the system should be able to integrate the resources of a new party or project. For example, the LOFAR Long-Term Archive employs the Astro-WISE metadata systems while the storage and processing resources are provided by the LOFAR consortium [3, 18].

The infrastructure for the processing of data of the Large Hadron Collider (LHC) and experiments connected with the LHC involved a newly invented common middleware solution, gLite. gLite middleware is a standard way to access Grid infrastructure for a number of applications (see, for example, [12] for a detailed description). The gLite-based Grid is widely used for many scientific applications, from earth science [8] to stellar evolution and stellar dynamic computations [14, 16], as well as for the data processing of space missions [9, 15].

The difference between these examples and the present work is that the classical approach to use Grid resources for an application is to port the application to the Grid, or to code of a new application which is able to use Grid resources. In our case we provide an integration between two existing Grids—Astro-WISE [19] and the Dutch national Grid infrastructure BigGrid.² BigGrid is the National Grid Initiative of the Netherlands which continues the Dutch

¹<http://setiathome.ssl.berkeley.edu>

²<http://www.biggrid.nl>

participation in the European effort to build a transnational Grid infrastructure, which started with EGEE and is now continued in EGI.

In this paper we will review Astro-WISE as a Grid, describe a general approach to the integration of Grids, and test the implemented solution.

Following a preceding study [4], this paper presents more practical results, along with the results of tests of the developed system.

2 Astro-WISE as a Grid

For the definition of Grid we will follow Ian Foster³ [6]. According to him the grid is an infrastructure which should have three main features: *coordinates resources that are not subject to centralized control, uses standard, open, general-purpose protocols and interfaces, delivers nontrivial quality of service*. Let us review the Astro-WISE information system from these points of view.

Coordination of resources that are not subject to centralized control Astro-WISE is a fully federated system both on the level of data administration, storage and processing. Each node is independent and is managed by the local authorities only, storage resources and data processing facilities can be shared with other nodes. The metadata layer (the metadata database) can be both centralized and distributed, in the latter case each node's administration can select a level of mirroring. The bulk of the data is stored as files on Astro-WISE dataservers, which form a network of storage nodes. The user can access any of the deployed dataservers to retrieve the data. The same approach is used for processing facilities. Each Astro-WISE node can include a number of Distributed Processing Units (DPU) which serve as front-end to the processing cluster. Jobs can be submitted to any DPU in the system.

Uses standard, open, general-purpose protocols and interfaces The data storage layer part of Astro-WISE is close to the "classic" Grid regarding its functions and services. The dataservers are using standard http and https protocols. The metadata part of Astro-WISE is using Oracle RAC. Recently, Astro-WISE was also ported to MySQL DBMS. The Astro-WISE DPU can be installed on top of any queuing system.

Delivers non-trivial quality of service Astro-WISE was build to perform a whole chain of data processing for wide-field astronomical imaging, and was initially created to support the data processing for an optical imaging pipeline. Nevertheless, Astro-WISE as an infrastructure is not restricted to this type of processing only. In recent years Astro-WISE was extended to suit radio astronomical data processing and even handwritten recognition. In general,

³<http://www-fp.mcs.anl.gov/~foster/Articles/WhatIsTheGrid.pdf>

Astro-WISE provides a service for storing data (files) together with their metadata description, all combined with an advanced access right policy.

3 EGI—BigGrid

The currently running European Grid Infrastructure (EGI) started development in the year 1999 when the Grid was perceived for the storage and processing of the data of the LHC experiments. In 2004 the Enabling Grid for e-science (EGEE, [10]) project was initiated. In 2010 after completing the last phase of EGEE, EGI took over the operations of the European infrastructure for scientific computing and storage. Currently, more than 200 computing centers are participating in the project with a combined processing power of 160,805 logical CPUs and a storage capacity of more than 120 PB.⁴ The branch of EGI in the Netherlands is called BiGGrid. It combines resources of Stichting Academisch Rekencentrum Amsterdam (SARA), the National Institute for Nuclear and High energy physics (Nikhef), Philips Research at the High Tech Campus and the Center for Information Technology of the University of Groningen (CIT).

The Grid infrastructures which are used by Astro-WISE are supported by EGEE, EGI and BiGGrid. They are built on the same middleware—gLite⁵ [12] which also makes use of Globus⁶ [7] and Condor⁷ [17] software. EGI supports data management, workload management and a Global File System—LHC Computing Grid File Catalog⁸ (LFC).

4 Astro-WISE—BigGrid integration

There are a number of reasons for integrating two Grid infrastructures. The most important in our case are:

- to join resources of each of infrastructures to achieve challenging data storage and processing tasks;
- to integrate the data stored on one of the infrastructures with the data storage and data processing facilities of other infrastructures without actually copying the data. This is important for users who deal with Petabyte-scale data volumes;
- to provide the users of one of the infrastructures with the capabilities of the second one. For example, users of the LOFAR Long-Term Archive

⁴<http://gstat-prod.cern.ch/gstat/summary/>

⁵<http://www.glite.org>

⁶<http://www.globus.org>

⁷<http://www.cs.wisc.edu/condor>

⁸<https://twiki.cern.ch/twiki/bin/view/LCG/LfcGeneralDescription>

are employing functionalities of the Astro-WISE metadata database, while also keeping the data on the BiGGrid infrastructure.

In practice we have two groups of users: LOFAR LTA users who would like to have access to a system providing an advanced data model for metadata of the data stored on BiGGrid, and Astro-WISE users who would like to have additional computational resources at their disposal but wish to keep all the data within the Astro-WISE system. More groups within the Target⁹ project are going to use the same approach.

We emphasize that in the present solution the Astro-WISE Grid is a “master” one in the sense that all the metadata are kept within Astro-WISE, while the data files described by the metadata can be shared between Astro-WISE and BiGGrid. Furthermore the user submits jobs to the grid via the Astro-WISE DPU, which is an analog of the BiGGrid Workload Management System.

Astro-WISE has three layers of data storage and processing infrastructure: a metadata layer implemented by a relational database, a data layer which consists of a grid of dataservers providing access to stored files, and finally a data processing layer with all available processing clusters. Each of this layers can be used independently. The integration of two Grids (Astro-WISE and gLite-based) should be considered for each of these three layers.

The metadata layer of the Astro-WISE system allows to implement a complicated data model which contains a full description of the data stored in the files, i.e. all data beyond the measurement data such as a World Coordinate System matrix, photometric system, observing block, quality parameters, processing parameters etc. In this sense the gLite-based grid is lacking a metadata layer. Despite the introduction of AMGA¹⁰ [11], which, among other functions, extends gLite with a metadata description of files stored on the Grid, gLite itself does not allow to implement a complicated data model but only stores a path to the file, the checksum, the size of the file and access rights for this file. The complicated object-oriented data model is realized in the metadata layer of Astro-WISE. The metadata layer is used by a number of web services of Astro-WISE, data processing pipelines and workflows. In fact all interaction of the user with the Astro-WISE system goes through the metadata layer (see [2] and [5] for details). Consequently, in order to keep all the features of the Astro-WISE information system an integrated system should have access to the metadata layer of Astro-WISE adopted for the data storage and data processing on BiGGrid.

The data layer of BiGGrid and Astro-WISE has an essential difference: the LFC stores a full path to the file, whereas Astro-WISE stores the unique filename only, the location of the file is found dynamically by a request to one of the Astro-WISE dataservers.

⁹<http://www.rug.nl/target>

¹⁰<http://amga.web.cern.ch>

4.1 Data layer integration

The data storage layer of Astro-WISE employs dataservers. The Dataserver is a Python-coded server which accepts a string as input (an unique filename) and returns the file content. The unique file name is generated by Astro-WISE when the file is inserted into the system and can consist of the name of the owner, the type of the data stored in the file (for example, the name of the image frame or/and the name of the instrument) and a hash value (the MD5 sum of the file's content) or timestamp. The name generating algorithm assures that the name is unique within the system. The metadata layer of Astro-WISE does not store the path to the file but the unique name of the file only, i.e., `WFI.2000-09-28T02:22:37.466_8.fits`. The Python class `DataObject` must be used to store the file. Each data item is an instance of the `DataObject` class, which links the file containing the data with the metadata record for these data.

Adding new storage (in this case, BiGGrid storage) requires changes to the data file handling by the `DataObject` class. Previously, the Astro-WISE dataserver provided a single way to store a data file. Now Astro-WISE can operate on a number of additional storage systems and, hence, protocols to store and retrieve files. The protocol and the location can be stored as a URI (Uniform Resource Identifier), in the form `protocol://server/path/file`, where `protocol` can be *http*, *srm* or *gpfs*. SRM¹¹ (Storage Resource Manager) is a protocol for data access on Grid storage systems, and GPFS is IBM's Global Parallel File System. The interface to a new data storage system, like BiGGrid, should not only add a new method to retrieve or store data but it should check first if the user has access to the storage space, according to the access policy of BiGGrid. At the same time, interfaces should allow the user to select which of the stored files to retrieve in case of multiple copies on different storage types (dataserver, BiGGrid, GPFS).

The separation of the data files and the metadata is also important for security reasons and must be supported by the new interface as well. Indeed, the user can have rights to browse and access metadata, but may have no rights to retrieve the file with the data itself.

These issues are solved by adding a new element to the Astro-WISE approach for the data storage: the Storage Table. The Storage Table contains the URI of a file; each file can have multiple copies stored on a number of different storage spaces of different types. The Storage Table is synchronized with any changes of a data item in the system, as any operation on a file goes through the Storage Table. In addition the Storage Table adds an extra layer on the security, separating the data from the metadata layer.

¹¹<https://sdm.lbl.gov/srm-wg/doc/srm.methods.v2.0.doc>

BiGGrid is using a LFC to create a Global File System over the grid; the LFC has a number of ways to identify stored files. The Grid Unique Identifier (GUID) is an unique key to the file, the logical file name (LFN) is a human readable name which is connected with the GUID while the *srm* is the storage URL of a physical location of a copy of the file. To avoid unnecessary browsing of the LFC and to provide a counting mechanism for the number of copies of the file we save the *srm* URL in the Storage Table for each copy of the file.

The DataObject instance stores the reference to the data file and an unique object identifier. Meanwhile the filename, filesize, hash value, and URI are stored in the instance of another Astro-WISE class—FileObject. The abstract Storage class defines an interface to protocols, as well as abstract retrieve and storage methods implemented by classes inherited from the Storage class. Figure 1 shows the dependencies between DataObject-, FileObject- and Storage classes.

The implementation of the hierarchy of classes for LOFAR storage allows to use both Astro-WISE and BiGGrid storage. The method of integration chosen by Astro-WISE allows to extend the data storage to other storage systems by adding the corresponding protocol interface in the scheme described above, without any changes in the parent classes.

4.2 Processing layer integration

The integration of the processing layer is approached like the integration of the data layer: the data processing facilities of BiGGrid should be coupled to the existing data processing facilities of Astro-WISE so that a user of Astro-WISE or its derivative systems will be able to execute Astro-WISE pipelines on BiGGrid resources without changing any code.

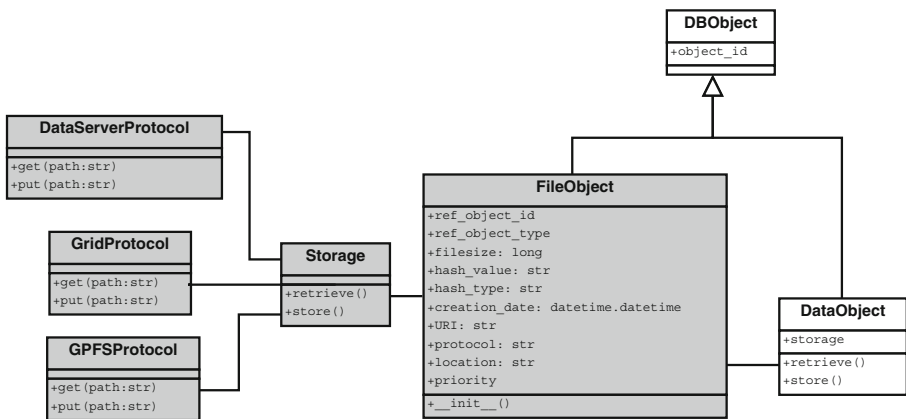


Fig. 1 Class hierarchy which allows to use Grid storage along with Astro-WISE storage. Original Astro-WISE classes are highlighted white, classes which were modified to implement the integration with BiGGrid are highlighted in gray

The job submission in Astro-WISE goes through the DPU—a front-end to the Workload Management System of the processing cluster. Usually the job is sent to a cluster where Astro-WISE is already installed, so the compute element has all necessary packages. In the case there is no pre-installed Astro-WISE on the compute element the system can be downloaded and installed prior to the execution of the task. Currently installations are available for SuSE, Fedora and RedHat linux for the x86_64 architecture.

The processing layer of Astro-WISE (DPU) consists of three subsystems: DPU server, DPU client and DPU runner (all are Python-coded). The DPU server handles requests from users for job submission and serves as a mediator between the user and the Workload Management System of the compute cluster, allowing to check the status of the job, query jobs submitted by the user or cancel them. The DPU client is a package which allows the user to connect to the DPU server from the user's program or Astro-WISE prompt. The DPU runner is the part of the DPU which runs on the compute element, checks the status of the Astro-WISE installation and installs Astro-WISE if necessary.

The DPU server has a web service front-end which can be used to monitor the status of the job, both for Astro-WISE HPC cluster¹² and BiGGrid compute elements.¹³

4.3 Authentication and authorization

The integration efforts described above would not be successful without taking care of an important component of the Grid: its system of authentication and authorization. EGEE, EGI and BiGGrid have very strict authorization rules which cannot be bypassed. Astro-WISE user has a single username and password. The authorization in all Astro-WISE services is coupled to the authorization in the Oracle DBMS, the same DBMS which is used to store the metadata. Grid authorization on the other hand is based on X.509 certificates.

An Astro-WISE user who wishes to store data on the Grid or to submit jobs to the Grid (non-Astro-WISE) infrastructure must first obtain a Grid certificate. This certificate is then used to create a proxy certificate on a MyProxy server¹⁴ which will, in turn, be used by the DPU to submit a job to the Grid or to store data. To create a proxy on the MyProxy server, Astro-WISE ports proxy creation software¹⁵ to the local environment. A command-line version of this java tool was made. As soon as the proxy has been created the user can execute Astro-WISE commands to operate on Grid. For the data storage we wrapped an external package jLite¹⁶ which allows to execute gLite

¹²<http://dpu.hpc.rug.astro-wise.org:9000>

¹³<http://dpu.grid.target.astro-wise.org:9000/>

¹⁴<http://grid.ncsa.illinois.edu/myproxy/portals.html>

¹⁵<http://www.ngs.ac.uk/tools/certwizard>

¹⁶<http://code.google.com/p/jlite>

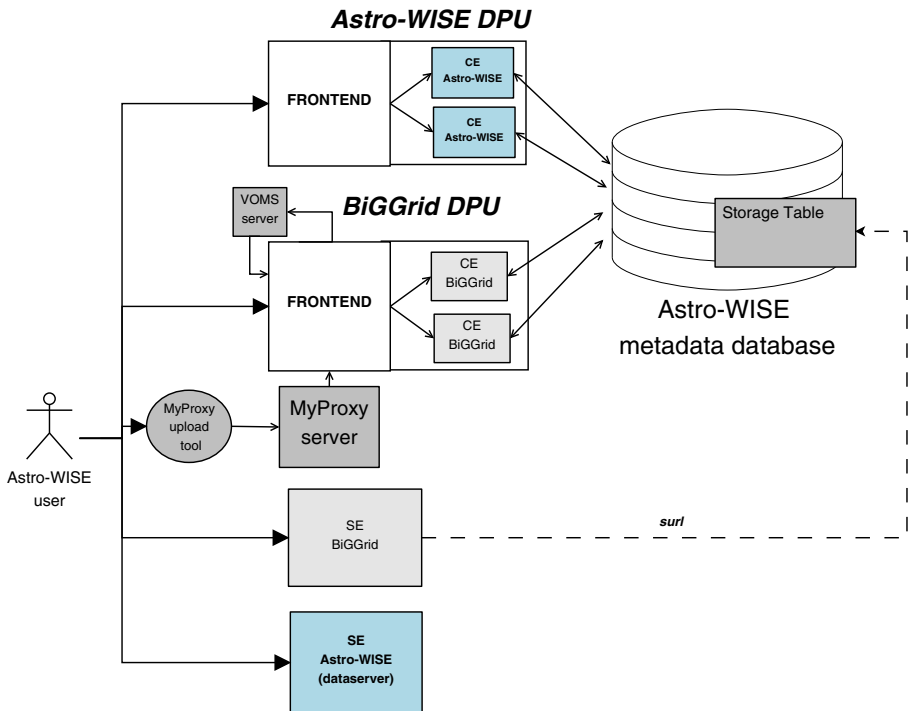


Fig. 2 The general overview of the user access to Astro-WISE and BiGGrid resources, including computing elements (CE) and storage elements (SE)

commands without installing gLite itself, together with SRM client software from dCache.¹⁷ For the processing, the DPU server was modified to use a proxy from the MyProxy server during submission of a job to the Grid.

Figure 2 shows the job submission and data storage possibilities for an integrated system. The user can select which DPU (and, hence, which computing elements) to load with the job and which storage elements to use. In the case of a Grid storage element the user's certificate will be checked for the verification of the user and the user's affiliation with the Virtual Organization involved. Virtual Organization Membership Service (VOMS, [1]) is used to assign roles to the user within Virtual Organization (*omgac* for Astro-WISE, *lofar* for LOFAR). The DPU will take care of adding the required VOMS role to the users Grid proxy.

¹⁷<http://www.dcache.org>

5 Tests: Astro-WISE data processing

The testing of Astro-WISE-Grid integration was done with typical Astro-WISE data processing: the data were processed from a raw image to the final catalog. The processing chain involves *Reduce*, *Regrid*, *Coadd* and *SourceList* tasks along with *Astrometry* task.¹⁸ The test data consists of two chunks: an observation of NGC 6383 made with WFI at 2.2 m telescope at La Silla during the night of 13 June 2004 (raw data and raw calibration files are publicly available via the ESO archive) and an observation made with the OmegaCAM camera at VST at Paranal observatory during the commissioning phase on the night of 13 August 2011.

The data processing consists of subsequent steps which reduce the raw image using all necessary calibration files, interpolate the reduced image into the standard coordinate system, combine all regridded images in one single image and extract sources from it. The pipeline is described in [13]. The same pipeline implemented in Astro-WISE is used to process the data for both instruments.

The processing was done on the HPC cluster of the University of Groningen (job submitted via Astro-WISE DPU) and on Grid resources (the *omegac* Virtual Organization is using processing clusters in Groningen and Amsterdam).

As a first step the user must get a Grid certificate, which is easy to do with the use of the Terena Certificate Service portal.¹⁹ This service is using the user's credentials from the institute with which the user is affiliated.

In the next step the user has to specify the DPU which will be used to submit a job—Grid DPU or Astro-WISE HPC DPU. This is done by setting the environment variable `dpu_name` or by setting it in the program `dpu=Processor("dpu.grid.target.astro-wise.org")`. This is the only change in the program which the user must do to switch from “native” Astro-WISE processing element to a Grid one.

The data processing is parallelized per CCD chip, i.e. the job is submitted to 8 cores of the cluster in the case of the WFI. Each core processes a single CCD chip. In the case of OmegaCAM a job is submitted to 32 cores of the cluster. The user can change this parameter to decrease the number of cores requested by the job. All processing tasks (*Reduce*, *Regrid*, *Coadd* and *SourceList*) have the same operational sequence: check the metadata database for the available data and dependencies for the target data item (the data item which will be created), retrieve necessary files to the local node, process the data, submit created files to dataservers, commit the newly created data entity to the metadata database.

The processing time depends on the configuration of the Grid system and Astro-WISE system, and many other parameters including load of the computing elements at the moment, state of the queuing system, network

¹⁸http://www.astro-wise.org/portal/aw_howtos.shtml

¹⁹<https://tcs-escience-portal.terena.org>

configuration etc. For this reason the results of the test do not identify the “right” processing system but should be taken as an indicator of the performance of the pipeline on different processing facilities. The execution time of each task is very close as the nodes of both the processing clusters (Grid and Astro-WISE) are similar in composition and performance. Nevertheless, the biggest difference is the load time—the time spent to collect all data files necessary to execute the task. For example, the elapsed time for the `Regrid` task for WFI 8.4 Megapixel data on the Grid computing element is 563 s, 274 s of this time was spent on loading and uploading data files. The same task on the “native” Astro-WISE cluster takes 474 s of which 237 s are spent on storage and retrieval operations. Thus, the overall performance depends mostly on the network connections between the processing element and storage elements.

6 Conclusions and further development

Integration of Astro-WISE with Grid resources extended the capabilities of the system and allows the user to involve Grid resources for the data processing. Despite successful use of the external Grid there are some problems that remain, the most important of them is the absence of a joint Workload Management System for both “native” and “external” Astro-WISE resources. Indeed, the user decides where to submit a job—to the Astro-WISE DPU or to the Grid DPU based on the visual monitoring of the state of already submitted jobs. In the case of Astro-WISE there is no system in place which would recommend the user which computing element to select for which job. Such a recommendation could be issued based on the status of computing elements, the expected job’s run time, required data and the storage location of these data.

There is currently no “automatic” reaction to a job failure. In the case of a terminated, aborted or “lost” job the user should decide the next action—resubmit the job to the same computing element or select another DPU for this job.

The queuing system of the “native” Astro-WISE computing elements and the external Grid have different waiting times due to the different number of users and overall load. Normally the waiting time for an Astro-WISE computing element is around 1 min, while the waiting time for the Grid queuing system varies between 200–400 s, due to different occupancy and policies on the system. For best performance the user should submit short-running jobs to the Astro-WISE DPU, while long-running ones can be send to Grid resources.

At the moment two WISE-based systems are using the external Grid (BiGGrid) infrastructure—Astro-WISE itself and the LOFAR Long-Term Archive. The ability to access the external Grid infrastructure became an integral part of Astro-WISE and WISE technology and in practice any information system developed from Astro-WISE can inherit this feature.

Acknowledgements Astro-WISE is an on-going project which started from a FP5 RTD programme funded by the EC Action “Enhancing Access to Research Infrastructures”. This work is supported by FP7 specific programme “Capacities—Optimising the use and development of research infrastructures” and has partly been done in the framework of EGEE-III NA4 Astronomy & Astrophysics Cluster. This work is part of the programme of BiG Grid, the Dutch e-Science Grid, which is financially supported by the Nederlandse Organisatie voor Wetenschappelijk Onderzoek (Netherlands Organisation for Scientific Research, NWO).

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

References

1. Alfieri, R., et al.: From gridmap-file to VOMS: managing authorization in a Grid environment. *Future Gener. Comput. Syst.* **21**, 549 (2005)
2. Begeman, K., Belikov, A.N., Boxhoorn, D., Valentijn, E.A.: The Astro-WISE datacentric information system. *Exp. Astron.* (2011, submitted)
3. Begeman, K., et al.: LOFAR information system. *Future Gener. Comput. Syst.* **27**(3), 319 (2011)
4. Begeman, K.G., et al.: Merging Grid technologies—Astro-WISE and EGEE. *J. Grid Comput.* **8**(2), 199 (2010)
5. Belikov, A.N., Vriend, W.-J., Sikkema, G.: Astro-WISE interfaces: scientific information system brought to the user. *Exp. Astron.* Published online <http://www.springerlink.com/content/14k92212142016/> (2011, accepted)
6. Foster, I.: What is the Grid? A Three Point Checklist, *GRIDToday*, 20 Jul 2002
7. Foster, I.: Globus toolkit version 4: software for service-oriented systems. In: *IFIP International Conference on Network and Parallel Computing*, LNCS, vol. 3779, p. 2. Springer (2006)
8. Fusco, L., Cossu, R.: Past and future of ESA earth observation Grid. *Mem. Soc. Astron. Ital.* **80**, 461 (2009)
9. Ibarra, A., et al.: On-the-fly XMM-Newton spacecraft data reduction on the Grid. *Sci. Program.* **14**, 141 (2006)
10. Jones, B.: An overview of the EGEE project. In: *Book Series Lecture Notes in Computer Science*, vol. 3664, p. 1 (2005)
11. Koblitz, B., Santos, N., Pose, V.: The AMGA metadata service. *J. Grid Comput.* **6**, 61 (2008)
12. Laure, E., et al.: Programming the Grid with gLite. *Comput. Methods Sci. Technol.* **12**, 33 (2006)
13. McFarland, J.P. et al.: The Astro-WISE optical image pipeline: development and implementation. *Exp. Astron.* [arXiv:1110.2509](https://arxiv.org/abs/1110.2509). Published online <http://www.springerlink.com/content/ng6w87476072u756/> (2012, accepted)
14. Taffoni, G., et al.: Grid and databases: BaSTI as a practical integration example. *J. Grid Comput.* **8**(2), 223 (2010)
15. Taffoni, G., et al.: Enabling Grid technologies for Planck space mission. *Future Gener. Comput. Syst.* **23**, 129 (2007)
16. Taisuke, B., et al.: HMC-G: Grid-enabled hybrid computing system for computational astrophysics. In: *Third IEEE International Symposium on Cluster Computing and the Grid (CCGrid'03)*, p. 558 (2003)
17. Thain, D., Tannenbaum, T., Livny, M.: Condor and the Grid. In: Berman, F., Hey, A.J.G., Fox, G. (eds.) *Grid Computing: Making The Global Infrastructure a Reality*. Condor High-Throughput Computing Software Framework. Wiley (2003)
18. Valentijn, E., Belikov, A.: LOFAR information system design. *Mem. Soc. Astron. Ital.* **80**, 509 (2009)
19. Valentijn, E., et al.: Astro-WISE: chaining to the Universe. In: Shaw, R.A., Hill, F., Bell, D.J. (eds.) *Proc. of ADASS XVI, ASP Conf. Ser.*, vol. 376, p. 491 (2007)