End-To-End Solution for Integrated Workload and Data Management using GlideinWMS and Globus Online

2012 J. Phys.: Conf. Ser. 396 032076

(http://iopscience.iop.org/1742-6596/396/3/032076)

Journal of Physics: Conference Series **396** (2012) 032076 doi:10.1088/1742-6596/396/3/032076

End-To-End Solution for Integrated Workload and Data Management using GlideinWMS and Globus Online

Parag Mhashilkar,[1 a] Zachary Miller,[d] Rajkumar Kettimuthu,[b,c] Gabriele Garzoglio,[a] Burt Holzman,[a] Cathrin Weiss,[d] Xi Duan,[e] Lukasz Lacinski[c]

[a] Scientific Computing Division, Fermi National Accelerator Laboratory, Batavia. IL 60563, USA
[b] Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL 60637, USA
[c] Computation Institute, The University of Chicago and Argonne National Laboratory, Chicago, IL 60637, USA
[d] Department of Computer Sciences, University of Wisconsin-Madison, Madison, WI 53706, USA
[e] IIT College of Science and Letters, Illinois Institute of Technology, Chicago, IL 60616, USA

E-mail: parag@fnal.gov, zmiller@cs.wisc.edu, kettimut@mcs.anl.gov, garzogli@fnal.gov, burt@fnal.gov, cweiss@cs.wisc.edu, xduan@iit.edu, lukasz@ci.uchicago.edu

Abstract. Grid computing has enabled scientific communities to effectively share computing resources distributed over many independent sites. Several such communities, or Virtual Organizations (VO), in the Open Science Grid and the European Grid Infrastructure use the GlideinWMS system to run complex application work-flows. GlideinWMS is a pilot-based workload management system (WMS) that creates an on-demand, dynamically-sized overlay Condor batch system on Grid resources. While the WMS addresses the management of compute resources, however, data management in the Grid is still the responsibility of the VO. In general, large VOs have resources to develop complex custom solutions, while small VOs would rather push this responsibility to the infrastructure. The latter requires a tight integration of the WMS and the data management layers, an approach still not common in modern Grids. In this paper we describe a solution developed to address this shortcoming in the context of Center for Enabling Distributed Peta-scale Science (CEDPS) by integrating GlideinWMS with Globus Online (GO). Globus Online is a fast, reliable file transfer service that makes it easy for any user to move data. The solution eliminates the need for the users to provide custom data transfer solutions in the application by making this functionality part of the GlideinWMS infrastructure. To achieve this, GlideinWMS uses the file transfer plug-in architecture of Condor. The paper describes the system architecture and how this solution can be extended to support data transfer services other than Globus Online when used with Condor or GlideinWMS.

[1] To whom any correspondence should be addressed.

1. Introduction

Grid computing has been widely deployed and used by big and widespread scientific communities with high computing demands, such as high-energy physics (HEP). While Grids enable pooling of resources across different administrative domains with relative ease, they also bring several challenges for the users in the management of these resources. Several communities, or Virtual Organizations (VO), in the Open Science Grid (OSG) [1] and the European Grid Infrastructure (EGI) have effectively used GlideinWMS to circumvent these challenges. Section 2 gives an overview of GlideinWMS, a pilot-based workload management system.

While GlideinWMS can effectively tackle the challenges of managing compute resources, data management in the Grid is still the responsibility of the VO. Large VOs have the resources to develop complex custom solutions for managing their data. Smaller VOs and individual researchers, however, would rather push the responsibility of data movement over the Grid to the underlying infrastructure. The CEDPS project [7] worked towards the goal of enabling large-scale computation by producing technical innovations designed to allow rapid and dependable data placement within a distributed high-performance infrastructure. Section 3 describes Globus Online [9], a fast, reliable, file transfer solution developed in the context of the CEDPS project.

Irrespectively of the data management solution selected, its effective use requires that it work in concert with the workload management system, an approach still not common in modern Grids. Section 4 describes the approach used to integrate GlideinWMS with Globus Online to provide an End-to-end solution for integrated WMS and data management system. Section 5 describes the results of our scalability tests for this approach. We discuss future work in section 6 and we conclude in section 7.

2. GlideinWMS

A pilot-based WMS is a pull-based WMS that creates an overlay pool of compute resources on top of the Grid. Several pilot-based WMS infrastructures [5] are used by different VOs in OSG and EGI. GlideinWMS is a pilot-based WMS that creates on demand a dynamically sized overlay Condor batch system [2] on demand on Grid resources to address the complex needs of VOs in running application workflows. This section describes Condor and GlideinWMS.

2.1. Condor

Condor started as a batch system for harnessing idle cycles on personal workstations [8]. Since then it has matured to become a major player in the compute resource management area.

A Condor pool is composed of several logical entities, as shown in figure 1:

- The central manager acts as a repository of the queues and resources. A process called the *"collector"* acts as an information dashboard.
- A process called the *"startd"* manages the compute resources provided by the execution machines (worker nodes in the diagram). The *startd* gathers the characteristics of compute resources such as CPU, memory, system load, etc. and publishes it to the collector.
- A process called the *"schedd"* maintains a persistent job queue for jobs submitted by the users.
- A process called the *"negotiator"* is responsible for matching the compute resources to user jobs.

The communication flow in Condor is fully asynchronous. Each startd and each schedd advertise the information to the collector asynchronously. Similarly, the negotiator starts the matchmaking cycle using its own timing. The negotiator periodically queries the schedd to get the characteristics of the queued jobs and matches them to available resources. All the matches are then ordered based on user priority and communicated back to the schedds, that in turn transfer the matched user jobs to the selected startds for execution. To fairly distribute the resources among users, the negotiator tracks resource consumption by users and calculates user priorities accordingly. With these characteristics,

Condor is an excellent choice for a pilot-based WMS; all a pilot jobs needs to do is configure and start a startd that reports as an idle compute resource to the collector.

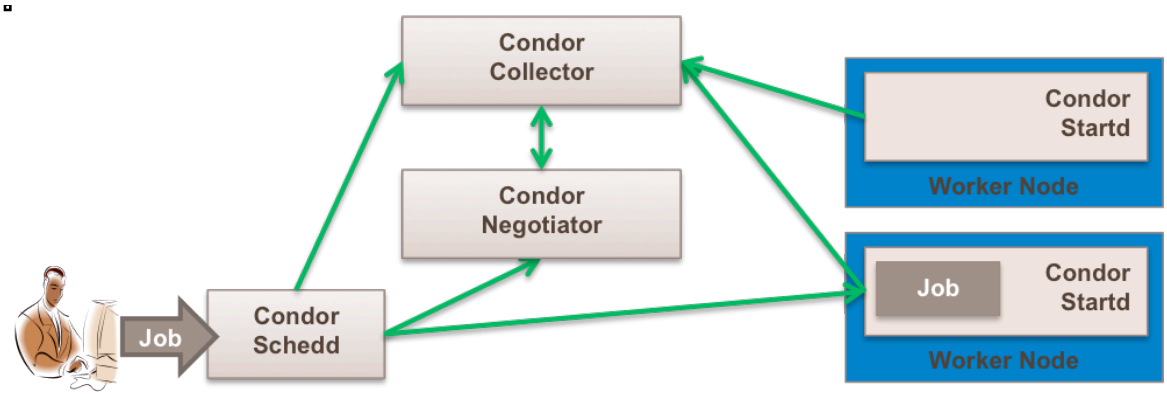


Figure 1: Condor architecture overview

2.2. Condor file transfer plugin architecture

Condor supports the transferring of input files to a worker node (startd) before a job is launched and of output files to the submit node (schedd) after the job is finished. When launching large amounts of jobs, however, this can cause a significant I/O load on the submit machine.

To ease this I/O load, Condor also supports third-party transfers for both input and output sandboxes. The input can be specified as a URL in the job description file. When the job is launched, the submit node does not send the file itself, but instead instructs the execute node to retrieve this URL directly. The output can also be specified as a URL. In this case, the execute node invokes a plugin to move the output to destination, instead of moving all files back to the submit node.

This is accomplished by a flexible plugin architecture, which allows Condor to support a wide variety of mechanisms to handle any URL type. By default, Condor supports basic URL types, such as ftp and http. Using the plugin architecture, though, Condor can easily be extended to support domain-specific protocols, such as GridFTP and Globus Online. Plugins are registered statically with the system and report success or failure of a transfer via an exit code.

Input files may be a mix of regular transfers and different URL types. In order for the system to improve transfer reliability, Condor automatically adds the requirement for the job that the necessary URL plugins exist. These requirements are then enforced by the Condor matchmaking [11] mechanism. The output sandbox can also be transferred by a plugin, although some additional limitations exist. In practice, all output must be transferred using the same plugin with the same destination. Hence, individual files cannot go to separately specified URLs. In addition, the plugin for a URL is invoked for each output file in a list. The Condor team is planning to address these issues in subsequent releases of Condor.

2.3. GlideinWMS overview

The GlideinWMS [4] is built on top of Condor. Functionally, it can be organized into three logical architectural entities as shown in figure 2

- A dashboard used for message exchange. A Condor collector is used for this task. This collector, along with a Condor negotiator, also acts as a central manager for the pilot jobs.
- Glidein factories that create and submit pilot jobs to the Grid using Condor-G [3] as a Grid submission tool.
- VO frontends that monitor the status of the Condor WMS and regulate the number of pilot jobs sent by the glidein factories.

The GlideinWMS architecture supports multiple glidein factories and frontends working together through a common WMS collector. This separation of tasks allows for better scalability.

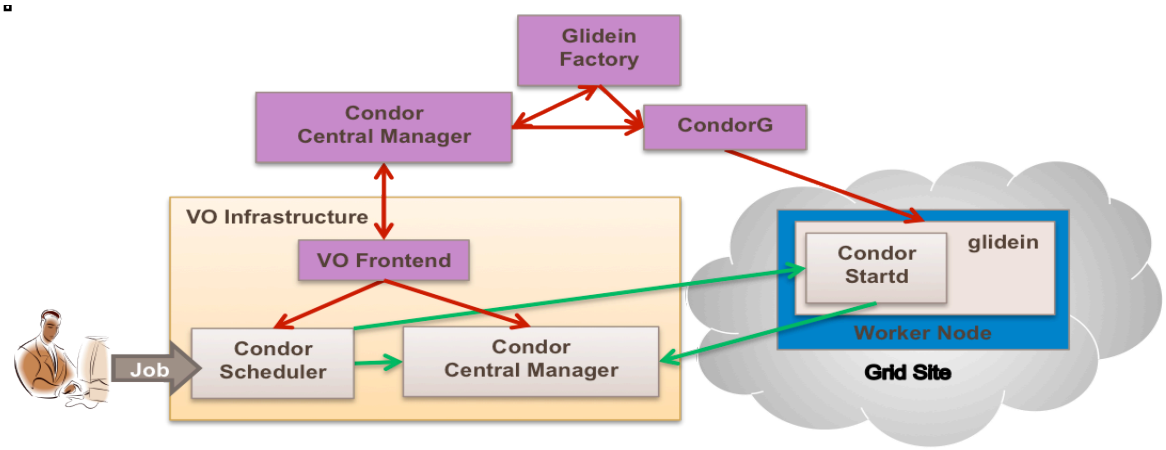


Figure 2: GlideinWMS architecture overview

3. Globus Online

Globus Online uses a Software-as-a-Service (SaaS) infrastructure to deliver high-performance, reliable and secure data movement capabilities. It allows users to request the movement or synchronization of datasets between remote storage systems that are commonly located in different administrative domains. Globus Online provides an easy-to-use web, REST, and command line interfaces. Figure 3 provides an overview of Globus Online.

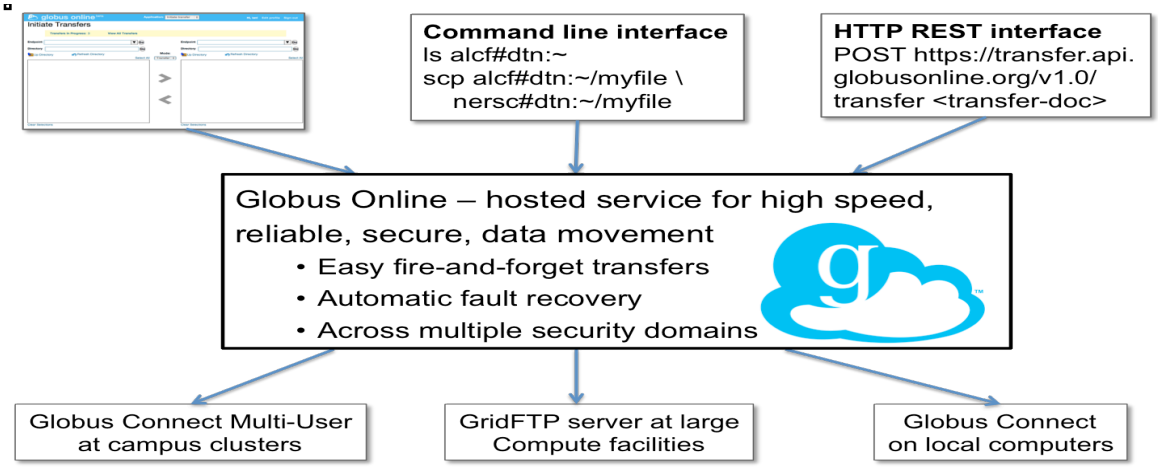


Figure 3: Globus Online architectural overview

Globus Connect Multi User (GCMU) [12], a multiuser version of Globus Connect described in Section 3.1, provides the functionality of a Globus Online endpoint in multi-user environments such as campus clusters. Researchers with minimal IT expertise can use Globus Online to move large scientific data sets reliably and quickly amongst national cyber infrastructures, super computing facilities, campus systems, and personal computers.

3.1. Globus Connect

Globus Connect solves the "last mile problem" of transferring data to and from the user's desktop or laptop. Globus Connect is a special packaging and wrapping of the GridFTP server binaries for

Windows, Mac OS X, and Linux. Since Globus Connect makes only outbound connections, it can be used to transfer files to and from a machine behind a firewall or Network Address Translation device. Figure 4 shows the steps involved in performing a transfer to/from a Globus Connect endpoint via Globus Online.

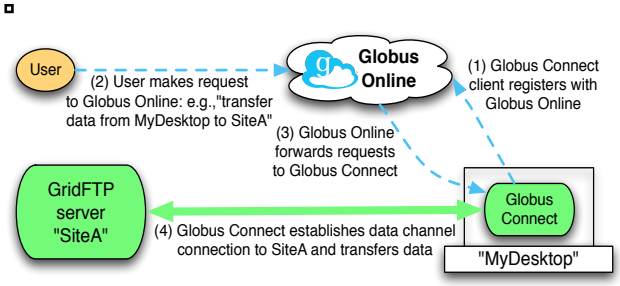


Figure 4: Globus Connect

4. Integrating GlideinWMS with the Globus Online

Modern Grid user communities put significant effort in identifying WMS and data management solutions that cater to their needs. Typical data movement happens within the user job; however, we argue that moving the data to the job or a job to the data should be the responsibility of the infrastructure. A similar argument holds for the transferring of the output sandbox produced by the job. Integrating the WMS and data management pushes this responsibility to the infrastructure, thus allowing the jobs to focus exclusively on data processing.

In this section we describe the mechanism used to integrate GlideinWMS with Globus Online. The mechanism described uses the "custom scripts" feature in GlideinWMS (section 4.1) and the file transfer plugin architecture of Condor (section 2.2). This mechanism can be easily extended by VOs to support custom file transfer protocols. This decouples the responsibilities of the user jobs and workflows from those of the data delivery layer.

4.1. GlideinWMS custom scripts

GlideinWMS supports running validation scripts on the grid worker nodes. The pilot, commonly called as a *glidein* in the GlideinWMS, runs these scripts after it starts up on the worker node and before it runs the Condor daemons that register the node to the VO pool. These validation scripts perform necessary checks to validate the ability of the grid worker node to run user jobs. If any of the validation checks fail, the *glidein* exits, thus shielding the user job from potential failures.

The glideinWMS factory comes with the validation scripts that perform common validation checks for OSG and EGI. GlideinWMS also empowers the VO to configure their frontend with custom scripts specific to their VO.

4.2. Globus Online transfer plugin

The Globus Online (GO) plugin [6] developed for Condor uses Globus Connect as an endpoint to transfer files using GO. The Globus Online plugin only supports X509 authentication and requires a valid proxy to successfully transfer files. It also expects that the location to the file be specified in the format *"globusonline://<GOusername>:<GOendpoint>:<GOfilepath>"*

The Globus Online plugin is independent of the version of GlideinWMS, but it depends on the Globus Connect version. The plugin registers the support for the *globusonline://* file transfer protocol with the Condor daemons through the plugin registration. On encountering an input or output file starting with *globusonline://*, Condor invokes the Globus Online plugin. The plugin creates a Globus Connect endpoint on the fly that registers itself with the central Globus Online Services. The plugin then uses this endpoint to transfer the file using Globus Online. After the transfer completes, the plugin shuts down the Globus Connect endpoint and performs the required cleanup.

4.3. Integrating GlideinWMS with Globus Online

Figure 5 shows the steps involved in transferring an input file using the plugin. The flow for transferring output files is very similar, except that the starter executes the job before invoking the plugin to transfer the file.

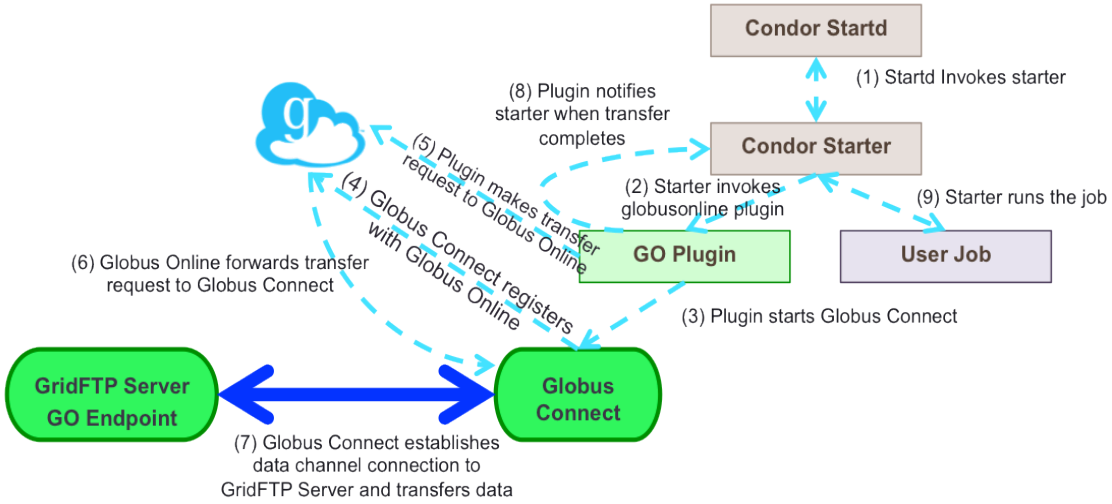


Figure 5: Transferring input files using Globus Online plugin

4.4. Asynchronous sandbox management

One potential way to achieve better long-term throughput over the life span of several jobs is to overlap job execution and I/O. The I/O phase, in fact, often blocks waiting for the availability of data transfer servers. In this case, the CPU can be used to run computations from another *independent* job, rather than staying idle. Specifically, if the input, execution, and output of Job A are independent of the input, execution, and output of Job B, overlapping is possible. When Job A finishes, it will begin transferring its output while Condor can begin the transfer of input and the execution of Job B. When considered over a workload of many jobs, this technique has the potential to benefit overall computational throughput. Figure 6 shows a diagram that compares the synchronous and asynchronous sandbox transfer in Condor.

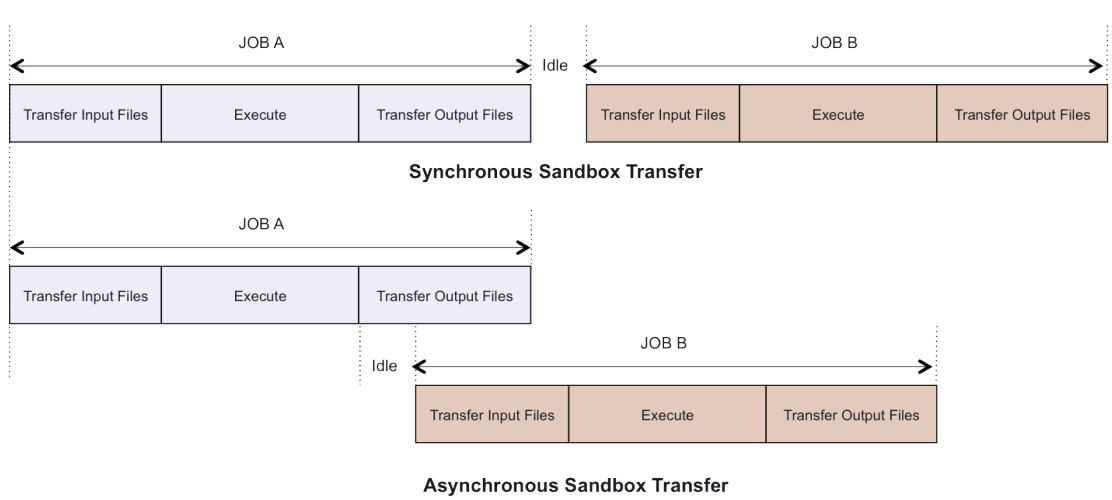


Figure 6: Synchronous vs. Asynchronous sandbox transfer in Condor

Condor was extended to allow persistent registration of job sandboxes, so that job execution and the management of the job's sandbox are decoupled. The sandbox management component is responsible for creating, registering, tracking, and cleaning up job sandboxes as requested. The job's execution was explicitly broken into three distinct phases: transferring input, execution, and transferring output; these were made into separate Condor job states. Using those mechanisms, Condor was then modified to explicitly begin the transfer and execution of another job while simultaneously transferring the output of the previous job. When the output has been successfully transferred, Condor unregisters that sandbox and that job is marked as complete.

5. Scalability Tests

The integration approach described in the previous sections involves several systems, namely GlideinWMS as the WMS interface to Grid and the Globus Online as the data management system. Both of these systems have different scalability limits. For practical purposes, one is interested in how the integration of the WMS and data management systems scales, as well as identifying potential bottlenecks.

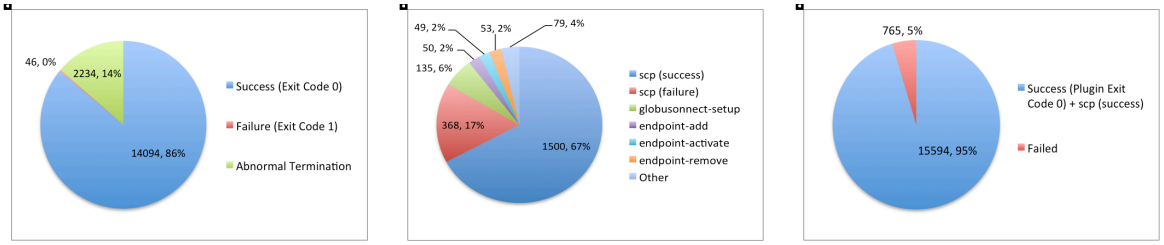
5.1. Test setup

To test the scalability of the system, we deployed a GlideinWMS factory and frontend on a Virtual Machine (VM). The testbed was hosted on FermiCloud [10] resources. FermiCloud provides an Infrastructure as a service (IaaS) private cloud for the Fermilab Scientific Program. A GridFTP server running on another VM in the FermiCloud served as the Globus Online endpoint.

For the purpose of scalability tests, 5000 user jobs were submitted. Because of the usage quota only 2636 jobs of these 5000 jobs were considered for analysis. 2636 jobs translated to 16374 files transferred using Globus Online.

5.2. Analysis

Figure 9 shows the graphical representation of the results of scale tests using a prototypical plugin. First, the exit code of the plugin was used to analyze the success of a transfer. As shown in figure 9(a), 86% of the plugin invocations resulted in successful transfers while a very small percentage resulted in failure. Almost 14% of the plugin invocations – 2234 in total – terminated abnormally. Figure 9(b) shows further analysis of the abnormal termination of the plugin. Out of 2234 abnormal terminations, 67% resulted in successful transmission the files, while the remaining 33% failed at various stages in the plugin. The total number of successful transfers was 95%, as determined from the two analyses above (figure 9(c)).



(a) Plugin exit status

(b) Plugin abnormal termination

(c) File transfer status

Figure 7: Scalability Tests

Several simultaneous invocation of the plugin resulted in multiple Globus Connects contacting the Globus Online relay servers. This exceeded the scalability limits and thus increased the load on the Globus Online relay servers. We think that this load, along with job preemption by the testbed batch system contributed to the abnormal termination of the plugin. Also, these results were from the initial

phase of the Globus Online infrastructure. Improved error handling and retries in the plugin can greatly reduce the abnormal terminations, making the plugin exit status a reliable measure of the file transfer status.

Since this plugin is Globus Online-specific, transfer plugins written for other protocols or services would see a different success rate based on the data transfer service/protocol used.

5.3. Plugin performance

In order to measure the performance of the plugin with respect to direct Globus Online transfers, we ran a series of transfers for files of varying sizes using the plugin and again directly through Globus Online. Figure 10 shows the comparison of the transfers. As expected, the plugin suffers from overhead involved in setting up the Globus Connect endpoint before the transfers can start. As the file size increases, the overhead becomes less significant compared to actual transfers. The overhead can also be reduced by if GCMU is pre-installed on the compute node.

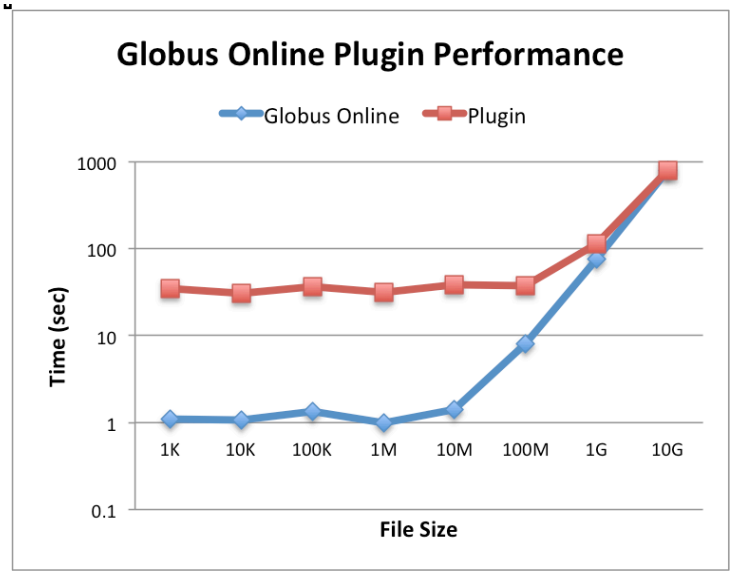


Figure 8: Globus Online plugin performance

6. Future Work

Based on the results of the scale testing, the Globus Online team is working on improving the scalability of the relevant components of their infrastructure. The Condor team is also working on enhancing the file transfer plugin architecture. The current architecture invokes the transfer plugin once per file. Grouping these invocations could greatly improve individual transfer performance. Globus Online plugin can take advantage of these improvements to increase the overall performance.

7. Conclusions

Several VOs in OSG and EGI use GlideinWMS as a workload management system to run complex scientific workflows. Large VOs have resources to develop complex data management solutions, while small VOs would rather cede this responsibility to the infrastructure. The Globus Online plugin developed in the context of the CEDPS project is one possible approach to integrate GlideinWMS with Globus Online using the Condor file transfer plugin architecture. This approach can be easily extended to integrate different data transfer protocols and services with GlideinWMS.

8. Acknowledgements

Fermilab is operated by Fermi Research Alliance, LLC under Contract number DE-AC02-07CH11359 with the United States Department of Energy.

The paper was partial sponsored by the Center for Enabling Distributed Petascale Science (CEDPS) Project.

The authors would like to thank the Intensity Frontier experiment and the REX group in the Fermilab for their help with the scale testing.

References

[1] R. Pordes, et. al., "The Open Science Grid", Journal of Physics: Conference Series 78, IoP Publishing, 2007 (15pp)
[2] Condor homepage (accessed on May 30, 2012): http://www.cs.wisc.edu/condor/
[3] J. Frey et. al., "Condor-G: A Computation Management Agent for Multi-Institutional Grids", Journal of Cluster Computing, 2002, vol 5, pp. 237-246.
[4] GlideinWMS homepage: (accessed on May 30, 2012): http://www.uscms.org/SoftwareComputing/Grid/WMS/glideinWMS
[5] I. Sfiligoi, et. al., "The Pilot Way to Grid Resources Using glideinWMS", CSIE '09 Proceedings of the 2009 WRI World Congress on Computer Science and Information Engineering - Volume 02
[6] Globus Online Plugin homepage (accessed on May 30, 2012): https://cdcvs.fnal.gov/redmine/projects/go-condor-plugin
[7] CEDPS homepage (accessed on May 30, 2012): http://www.cedps-scidac.org
[8] M. Litzkow, M. Livny, and M. Mutka, "Condor – A Hunter of Idle Workstations", Proc. of the 8th Int. Conf. of Dist. Comp. Sys., June, 1988, pp 104-111.
[9] Globus Online homepage (accessed on May 30, 2012): https://www.globusonline.org
[10] FermiCloud homepage (accessed on May 30, 2012): http://www-fermicloud.fnal.gov
[11] Rajesh Raman, Miron Livny, and Marvin Solomon, "Matchmaking: Distributed Resource Management for High Throughput Computing", Proceedings of the Seventh IEEE International Symposium on High Performance Distributed Computing, July 28-31, 1998, Chicago, IL
[12] Globus Connect Multi-User homepage (accessed on June 12, 2012): https://www.globusonline.org/gcmu