In the first article of this series (see http://doi.ieeecomputersociety.org/10.1109/MDSO.2008.16), we identified the need for teaching environments that provide infrastructure to support education and training in distributed computing. Training infrastructure, or t-infrastructure, is analogous to the teaching laboratory in biology and is a vital tool for educators and students. In practice, t-infrastructure includes the computing equipment, digital communications, software, data, and support staff necessary to teach a course. The International Summer Schools in Grid Computing (ISSGC) series and the first International Winter School on Grid Computing (IWSGC 08) used the Grid INFN Laboratory of Dissemination Activities (GILDA) infrastructure so students could gain hands-on experience with middleware. Here, we describe GILDA, related summer and winter school experiences, multimiddleware integration, t-infrastructure, and academic courses, concluding with an analysis and recommendations.

GILDA

GILDA began in 2004 as part of the INFN (Istituto Nazionale di Fisica Nucleare) Grid Project and the EU FP6 EGEE (Enabling Grids for e-Science) and ICEAGE (International Collaboration to Extend and Advance Grid Education) projects. GILDA has served as a test bed for training and dissemination, using EGEE middleware services. GILDA delivers services such as a dedicated certification authority, a virtual organization, and monitoring and support systems for users. GILDA has supported more than 200 training and dissemination events during the four years it has been functional.

GILDA operates continuously so that students anywhere in the world, at any time of day, can use the infrastructure for their studies. Teachers can also freely interact using GILDA to plan courses and book resources without system time constraints. This constant service results in challenges relating to support responses, the rollout of new versions, and workload management, but the benefits of continuous operation outweigh the disadvantages.

GENIUS and P-GRADE portals for GILDA

In 2002, INFN and NICE, an Italian company delivering grid solutions, developed the GENIUS (Grid Enabled Web Environment for Site Independent User Job Submission) Web portal (https://genius.ct.infn.it) with the goal of creating a simple, powerful, and customizable instrument for teaching grid computing. The complexity of the standard command-line-based interface (CLI) offered by the grid middleware on a UNIX-like environment is discouraging to many grid beginners. GENIUS, usually installed on top of a user interface, offers a graphical, simple, and intuitive interface to the grid services accessible from a common Web browser without additional requirements. When used during training events, GENIUS is very effective in introducing grid concepts because it hides CLI details from users who don’t have to check command syntax and can easily abstract their meaning.

MTA SZTAKI (Computer and Automation Research Institute, Hungarian Academy of Sciences) and INFN set up a P-GRADE portal installation for the international GILDA training infrastructure in December 2006. The portal serves as a demonstration, dissemination, and learning environment for everybody interested in the usage and capabilities of GILDA, the EGEE grid middleware, and P-GRADE
itself. Over the course of approximately one year, the GILDA P-GRADE portal was used during every major EGEE induction, EGEE application developer event, and ICEAGE event.

The GILDA P-GRADE portal is a P-GRADE portal 2.5 installation connected to the GILDA training infrastructure. It provides a graphical environment to perform certificate management, job submission, file transfer, information system browsing, and application monitoring on GILDA, eliminating the sometimes cumbersome and hard-to-memorize commands from the learning curve. This tool can significantly shorten the learning time required for grids. Besides providing graphical interfaces for GILDA middleware services, the GILDA P-GRADE portal also contains high-level tools that extend gLite's capabilities. The graphical editor and integrated workflow manager components can define and manage workflows and parameter studies.

Summer school and winter school experiences

Summer schools have been among the most important activities to promote grid technology around the world. Students who come to these events are generally experiencing grid for the first time. In a few weeks, they learn how it works and how to exploit this new technology in their daily lives. The second article in this series described ISSGC 2007 in detail (see http://doi.ieeecomputersociety.org/10.1109/MDSO.2008.20).

From experiences at ISSGC and other schools around the world, we’ve noticed that infrastructures that let students carry out exercises themselves are important to the learning process. (For more details, see a list of schools and tutorials supported by GILDA infrastructure at https://gilda.ct.infn.it/tutorials.html and the official 2008 ISSGC website at www.iceage-eu.org/issgc08/index.cfm.) In fact, the infrastructure’s quality in terms of performance, availability, and reliability is a key factor in students’ decisions to use the grid in the future. A frustrating experience at the school can negatively affect grid adoption.

Generally, the school’s organizer, in cooperation with the teaching staff, implements an ad hoc training infrastructure for the school. This infrastructure must contain all the components of a production grid infrastructure but on a smaller scale, typically three to four sites plus several required additional services such as centralized brokers, information systems, and file catalogs. The infrastructure’s implementation requires a precise analysis of the location, number of students, and nature of the school curriculum to ensure quality resources. Organizers must evaluate the location in terms of networks, space, power, and cooling. Also, components such as servers, network switches, and cables must be transported to and accommodated at the school venue, so organizers must include these factors in their analysis. Moreover, some lessons or exercises could require external sites, so the necessary bandwidth should be established for the duration of the school.

The number of students directly affects the infrastructure because each student produces a workload. Each practical session and team exercise exacerbates the challenge of supporting this workload—nearly all students attempt the same action at approximately the same time. Failure to meet these challenges produces exaggerated effects. First, if the load builds up because of these contemporaneous requests, queues extend in the scheduling software and the overall performance degenerates—a phenomenon well-predicted by queuing theory. Second, if some students are disheartened by the poor response, their negative views might rapidly propagate through the group.

Furthermore, the nature of the school’s curriculum affects the training infrastructure. If the school is designed for computer scientists willing to learn how the grid middleware works, then few elements are required, but students must access them to understand how each component functions. If planners design the school for application developers willing to develop and try their applications on the grid, then a substantial amount of power is required; students might submit complex applications many times on the infrastructure to understand how the grid can improve their execution.

These evaluations of the required infrastructure can be complex, and this analysis takes up a significant proportion of the grid school preparation time. For example, the organizers of ISSGC 08,
held at Hungary’s Lake Balaton in July 2008, began discussing t-infrastructure in December 2007 when they selected the school’s location. Local staff prepared the t-infrastructure, and a discussion with middleware experts was organized to decide how to configure the middleware and test exercises before assigning them to the students.

2008 was the first year at ISSGC in which all the technologies could be presented on a single infrastructure. The technologies were also expanded to include UNICORE, gLite, Condor, Globus, OGSA-DAI, and Microsoft HPC. In 2008, planners reduced the time given to the students to complete the integrating practical from two days to one day and roughly doubled the number of pillars. Figure 1 shows the details of the acquisition of pillars by individual student groups during this year’s summer school.

Figure 1. Time to acquisition of pillars by students at ISSGC 08.

Part 3 of this series discussed IWSCG 08, held from 6 February to 12 March 2008. As was the case with the summer schools, the winter school used the GILDA t-infrastructure. School planners explored several different t-infrastructure distributions at different instances of ISSGCs and IWSCG 08, varying the number of servers at the school, the location of services, and the bandwidth to external services and server banks. These distributions have met with various degrees of success. Having a large resource offsite and relying on a continuously available service from the IP network proved successful at the Mariefred summer school but caused network failures and overload in previous years. Even when a distribution of resources has been carefully planned, we’ve encountered problems—for instance, when all the students downloaded one file from the same network file system server. The current state of the art requires careful testing with an emulation of student load before the school starts, but there’s no shortcut for getting a group together to behave as if it were the student cohort. It’s possible to build a student-workload emulator if the t-infrastructure provides appropriate means of “plugging it in.” As far as we know, this idea hasn’t yet been explored.

Multimiddleware integration
At ISSGC 05, organizers and trainers articulated the need for a single environment under which all of the summer school’s practical work could take place and that could be made available to students after the school ended. This led to the development of multimiddleware integration, which became part of the EU FP6 ICEAGE project. The GILDA infrastructure, which already supplied a teaching environment for EGEE and gLite middleware, was selected as the basis for this multimiddleware environment. Other middleware, such as Globus, OMII-UK, and Condor, were added progressively.

**Security**

The first problem planners faced involved the security model. gLite and Globus have a common basis for security because gLite initially adopted the Grid Security Infrastructure model that Globus developed for user authentication. This model foresees the use of digitally signed certificates and proxies for the mutual authentication of users and hosts. Still based on X.509 certificates and proxies, this model has been extended in gLite with the adoption of the Virtual Organization Membership Service (VOMS), which issues fully X.509-compatible digitally signed extensions to proxies. These extensions bring additional information requirements for the users, needed to map users on different levels of authorization. The fact that a VOMS proxy is fully compatible with a X.509 proxy ensures that a VOMS proxy can be accepted as the authentication credential, even on resources deploying GT4. A VOMS server contains information about users authorized to access the grid infrastructure, the virtual organizations they belong to, the roles they can perform inside the VO, and the VO subgroups they can be part of. When users create a proxy with the command `voms-proxy-init`, they must specify the VO and, if they wish, the group and role. If the user-assigned privileges match those requested, a proxy is created and extended with the information returned by the VOMS server. This is called an *attribute certificate* (AC). The responsibility for authentication and user authorization is largely based on the AC extensions returned by the VOMS server and included in the proxy used for authentication.

In the proposed architecture, work has been done to harmonize between systems to limit the user problems related to the management of multiple certificates and tools. However, because the goal was coexistence of standard middlewares “out of the box,” harmonization activity was limited to installation and configuration without modification of the source code. For Globus, the target was to use the same certificates, the same local users, and the same authorization policy defined for gLite. The certificate model follows the X.509 standard so certificates can be exchanged between Globus and gLite without any problems. VOMS extensions were only used for gLite. At the end, the solution was to set up scripts that periodically contact the VOMS and download the list of all users authorized to access the site. This list is then used to update the grid-mapfile routinely as needed by Globus. In this way, gLite and Globus share the same policy. It’s worth noting that the same X.509 credentials are used to access both the gLite and Globus interface on the local batch scheduler, and even the same set of local accounts where remote users are mapped is shared by the two middleware.

The OMII-UK model is different from the gLite model, and it was impossible to share the same policy with gLite, so the integration was limited to reusing a single certificate for authentication. Also, in contrast to the Globus/gLite approach, OMII-UK resource administrators must accept users manually, and even manually apply single changes in the set of supported users.

**Sharing the infrastructure**

A second focus of multimiddleware integration activity was sharing the infrastructure to let users use the same resources with all the middleware. This means that the cluster manager handling the local resources must be configured so it can accept jobs submitted with different middleware. Our infrastructure’s local scheduler is Torque/MAUI, and jobs can be submitted via several queues. Torque/MAUI has the advantage of being well-supported by gLite, Globus, and OMII-UK, so defining new queues for additional middleware to use the same resources has sufficed. It wasn’t necessary to define new queues, but it simplified debugging and allowed for a better definition of resource scheduling and usage accounting.

**Academic courses**
In the last few years, grid computing (or high-throughput computing, desktop grids, campus grids, or clusters) has been introduced to undergraduate and master's curricula in many universities. Initially grid computing was taught as part of consolidated courses (such as distributed systems, advanced computing, or complex systems), but more recently it has become the topic of dedicated courses (see a list of MSc Grid courses at www.iceage-eu.org/v2/msc%20courses.cfm). Although university courses shouldn't be related to specific projects and middleware, they commonly let students gain experience with examples of current middleware on operational t-infrastructure. Consequently, the courses are influenced by the specific middleware provision they have used.

To make the courses more independent, standardizing t-infrastructure is crucial. This will let teachers and students choose the middleware they want to teach or learn without putting considerable effort into setting up the training infrastructure. The primary purpose of this practical experience is to reinforce concepts expected to persist as e-infrastructure and e-science evolve. Details that students must learn in order to use the specific exemplar t-infrastructure are a distraction and must be kept to a minimum.

A standard permanent training infrastructure (such as the one we've discussed here) can provide this freedom in grid courses, especially those outside of computer science where the goal is to set students on a trajectory where they will become adept at exploiting e-infrastructure. In fact, a permanently available t-infrastructure will let students develop those skills whenever they choose to constantly improve their use of e-infrastructure in their chosen scientific disciplines.

Establishing such a t-infrastructure would require an economic model for its maintenance and operation. Today, universities are prepared to invest in Internet connections and mail, diary, and document services, many of which are wholly or partially shared. T-infrastructure could be similarly shared, possibly in conjunction with Web 2.0 providers. However, this need and the economic and quality benefits of sharing haven't yet been recognized. The community should investigate the academic and business case for such sharing.

In the interim, university staff often must make do with hardware they already have and establish the running system themselves. Although this isn’t satisfactory, it’s much helped by downloadable systems, such as “grid in a box” and “grid in a room.”

**Analysis and recommendations**

Experiences with t-infrastructure systems such as GILDA have provided us with an understanding of what is required for, or facilitates, education and training in grid computing. We learned several lessons that can serve as a guide to best practice when it comes to t-infrastructure.

- Convenient provision of authentication and authorization is crucial. This can be achieved by using automatic provision via Web interface to users, batch provision to educators for courses and training events, and provision of certificates via prerequisite self-paced exercises (summer schools).
- The provision of directly accessible training material should be prioritized. We provided training material via trainer-maintained wiki pages on GILDA and through the ICEAGE and EGEE digital libraries. Trainer materials should be strictly related to and tested on the t-infrastructure (limiting the problem of adapting a generic procedure to a specific context).
- Developers of training infrastructure should use virtual machines for support systems administration training and for provision of reconfigured middleware as a downloadable, installable context for users to experience.
- Web portals ameliorate interface complexity, as we identified specifically in the cases of GENIUS and P-GRADE. We also provided continuous access to training platforms such as GILDA.

The proliferation of grid computing outside its original scope (scientific research) is limited, although there are extensive but largely hidden HTC (high-throughput computing) applications based on a
variety of customized or commercial e-infrastructures in finance and pharmaceutical companies. To overcome this limitation, it’s important that the main grid stakeholders promote new and more efficient training activities. These should involve not only all established grid communities but also universities, which are important places for knowledge creation.

Students learning grid computing need to perform exercises to understand how grid infrastructures work and how to use them. Exercises shouldn’t be limited to a specific period; students should be able to perform tests and exercises whenever they think that using the grid can help solve their problems. Therefore, public authority financing of grid projects should be encouraged so that training and education are considered important to any and every project. There should also be a strong move toward financing the creation of a permanent multimiddleware training infrastructure.

As we’ve mentioned earlier in this series, adequate t-infrastructure is only one requirement for distributed computing education and training. The collaboration and sharing that distributed computing enables ensures greater excellence in teaching but brings with it concerns about respect for intellectual property rights (IPR). The next article will explore IPR in the context of education and training repositories such as digital libraries to highlight legal problems and solutions in this area.

David Fergusson is deputy director of training, outreach, and education at the National e-Science Centre, Edinburgh. Contact him at dfmac@nesc.ac.uk.

Roberto Barbera is an associate professor in the Department of Physics and Astronomy at Catania University. Contact him at roberto.barbera@ct.infn.it.

Marco Fargetta is an external collaborator for Consorzio COMETA, Sicily. Contact him at marco.fargetta@ct.infn.it.

Emidio Giorgio is a researcher at INFN, Catania. Contact him at emidio.giorgio@ct.infn.it.

Gergely Sipos is a research fellow at MTA SZTAKI, Hungary. Contact him at sipos@sztaki.hu.

Diego Romano is a researcher in Computational and Computer Sciences at the University of Naples Federico II. Contact him at diego.romano@dma.unina.it.

Malcolm Atkinson is director of the e-Science Institute and e-science envoy at the National e-Science Centre, Edinburgh. Contact him at mpa@nesc.ac.uk.

Elizabeth Vander Meer is the education and training policy officer at the National e-Science Centre, Edinburgh. Contact her at evmeer@nesc.ac.uk.

Cite this article: