DRE at Purdue: A Distributed Rendering Environment for Teaching Animation and Scientific Visualization

Purdue University

What is it?

Purdue University’s Distributed Rendering Environment (DRE) is a forward-looking application of grid computing for teaching and learning. Generating images from data—a process known as “rendering”—requires a great deal of computer processing power. As part of the Purdue High Performance Classroom initiative, the DRE allows students to render 3D models and animations using unused processing capacity of multiple computers located in various facilities on campus. While colleges and universities acknowledge the need to prepare students for a work environment that may involve advanced visualization and animation, many institutions may find it impractical to build an expensive infrastructure exclusively for rendering animations. Working toward a low-cost alternative, Purdue designed its DRE to harvest computational resources from large computer laboratories that remain unused for part of each day.

Purdue’s Teaching and Learning Technologies division supports and maintains more than 60 instructional computing labs on campus. The solution designed and deployed by Purdue addresses the needs of students and instructors by splitting their animation jobs into individual frames and distributing those frames to any of the 1,500 computers (or “render nodes”) currently on the system. As the number of computers in the Purdue “render farm” grows, students will be able to undertake increasingly complex projects approaching the caliber of those undertaken in industry. The long-term goal of the Purdue DRE project is to lower the threshold of entry into grid computing for students and faculty in the arts, humanities, and social sciences and link the system to the national cyberinfrastructure, making this rendering resource available to an even wider community.

What problem does it solve?

In fields such as computer science, industrial design, biomedicine, nanotechnology, and an ever-increasing number of other disciplines, the rendering of images and animations at high resolution is becoming central to the scientific enterprise. While upper-division courses stress the use of high-resolution renders, resource limitations often make it impossible for students to apply the techniques they are learning to produce high-quality results. A two-minute animation rendered on a single machine can tie up that computer’s processor for upwards of 7,000 hours, making the machine unusable for any other activity. Worse still, it is difficult even for an experienced animator to diagnose all errors in shadowing, lighting, animation, and texture before rendering a full animation. It is not unusual for a student to go through the time-consuming process of rendering an animation several times to get the desired results. Students, loath to give up their machines and have a rendering job interrupted, “baby-sit” them for hours at a time. The quality of student submissions inevitably suffers because there is no time to re-visit mistakes and render the animation again. Multiply this scenario by hundreds of students, and the magnitude of the rendering problem in the educational setting becomes all too apparent.

To solve this bottleneck, Purdue’s DRE makes it possible for students to use free rendering software installed on machines in the various instructional labs and to submit their jobs for rendering. Usually the heavy-duty rendering is done on computers that would otherwise sit idle after the labs are closed for the evening. Purdue’s low-cost grid computing solution makes it possible for instructors to assign course work that more accurately reflects the difficulty and scale of real-world industrial and scientific projects.

How did they do it?

Grid computing techniques were first used by animation production units, film studios, and national labs to solve the problem of rendering animations in a timely manner. Because large animations could be broken up into district frames that could be rendered independently, the entertainment industry invested in tightly controlled clusters of computers known as “render farms,” limiting access to a small number of people. By contrast, Purdue’s approach to render farms stresses

- re-purposing existing resources distributed over a broad area;
- unfettered access to resources for authenticated students and faculty; and
- no additional investment in hardware, heating, ventilating, and cooling.

In its current implementation, the Purdue DRE distributes rendering work among up to 1,500 “recycled” instructional lab machines. Since these machines are in instructional labs and available to students during the day, the actual number of machines doing rendering at any given moment varies. The newest instructional lab machines are equipped with large memory capacity, multiple processors, Gigabit Ethernet connections,
and short-term storage (with more than 2 TB backup), making them fast enough to serve as render nodes in the Purdue DRE.

With Purdue’s DRE solution, students simply visit an instructional computing laboratory on campus and use an animation package already installed on the machines there to submit their animations for rendering with a click of the mouse. Purdue has engineered a server-based method of job storage and distribution that receives job submissions, splits the submitted animation into individual frames, and sends them to any available machine for processing. E-mail notifications at various points in the rendering process keep students informed of the job’s progress. Because completed renders are placed on network storage, authenticated users can easily retrieve them from any lab machine on the DRE grid.

Essentially, Purdue adapted a grid computing application already common within the computer graphics and entertainment industries to the needs of academic institutions. Those interested in replicating Purdue’s project at their home institution should consider the unique set of technical challenges posed by the harvesting of computer cycles within a university setting. These include:

- **Storage:** The institution must maintain a considerably large network storage space. Purdue has dedicated a portion of the central network storage to the DRE for holding completed renders and providing a shared space for rendering source files, front-end software extensions, unique plug-ins, and DRE access scripts.

- **Management:** Purdue currently uses a tool called EasyCondor to manage the submission, distribution, and processing of rendering jobs. The submitted job will be handled from start to finish by this job-management server acting as a resource broker, farming out the job to available computers (render nodes). Because the job details have been placed in central storage, the user can log off the submission machine, making it available for other users or render jobs. Students who walk into an instructional lab during normal hours can use any of the machines at any time because their activities take precedence over rendering jobs. With a move of the mouse, the student kicks the current render job off the machine and back into the job queue. The job management system makes Purdue’s distributed approach to high-performance computing cost-effective and flexible.

- **Security:** All federal data-security requirements must be taken into account when designing the networking system, the medium through which rendering tasks and assets are distributed to the computers. Purdue connects the DRE submission nodes, render nodes, job-management server, and central storage through a Gigabit Ethernet pipeline. Users are authenticated against a centralized directory database before they are authorized to submit jobs to the DRE.

**Why is it noteworthy?**

- **Transformational:** The DRE approach to supporting undergraduate access to high-performance computing resources provides students with the kinds of learning experiences that will prepare them for a work environment increasingly reliant on the creation of high-resolution visualizations and simulations.

- **Cost-effective:** By developing a mechanism for distributing time-consuming computational chores across hundreds of recycled instructional lab machines already in service, Purdue has developed a way of supporting innovative educational experiences without having to invest in expensive new infrastructure.

- **Replicable:** The lessons learned from Purdue’s pioneering experimentation with the EasyCondor job-management system for distributed rendering environments will benefit other institutions pursuing this approach to innovative teaching and learning. The Purdue support team welcomes inquiries, will provide assistance, and is particularly interested in forming multi-institutional partnerships to increase the number of applications that can be supported through this distributed architecture (for example, MatLab, Ansys engineering design software, and so on).

To learn more

To share your innovation
If your institution has a practice that you believe would be of interest to the EDUCAUSE Learning Initiative, please share it with us. To submit your innovation for review, please use the ELI Innovations Contribution Form on our Community Exchange page [http://www.educause.edu/ELICommunityExchange/6797](http://www.educause.edu/ELICommunityExchange/6797). A panel will review your submission and make a recommendation to the ELI staff.

**About the EDUCAUSE Learning Initiative**

The EDUCAUSE Learning Initiative (ELI) is a community of higher education institutions and organizations committed to advancing learning through IT innovation. To achieve this mission, ELI focuses on learners, learning principles and practices, and learning technologies. We believe that using IT to improve learning requires a solid understanding of learners and how they learn. It also requires effective practices enabled by learning technologies. We encourage institutions to use this report to broaden awareness and improve effective teaching and learning practice.