Editors: Paul F. Dubois, paul@pfdubois.com
George K. Thiruvathukal, gkt@cs.luc.edu

GLOBAL E-SCIENCE COLLABORATION

By Gigi Karmous-Edwards

TODAY'S E-SCIENCE, WITH ITS EXTREME-SCALE SCIENTIFIC APPLICATIONS, MARKS A TURNING POINT FOR HIGH-END REQUIREMENTS ON THE COMPUTE INFRASTRUCTURE AND, IN PARTICULAR, ON OPTICAL NETWORKING RESOURCES. ALTHOUGH ongoing research efforts are aimed at exploiting the vast bandwidth of fiber-optic networks to both interconnect resources and enable high-performance applications, challenges continue to arise in the area of the *optical control plane*. The ultimate goal in this area is to extend the concept of application-driven networking into the optical space, providing unique features that couldn't be achieved otherwise.

Many researchers in the e-science community are adopting Grid computing to meet their ever-increasing computational and bandwidth needs as well as help them with their globally distributed collaborative efforts. This recent awareness of the network as a prime resource has led to a sharper focus on interactions with the optical control plane, Grid middleware, and other applications. This article attempts to explain the rationale for why high-end e-science applications consider optical network resources to be as essential and dynamic as CPU and storage resources in a Grid infrastructure and why rethinking the role of the optical control plane is essential for next-generation optical networks.

The Optical Control Plane

The optical control plane is an infrastructure and distributed intelligence that controls the establishment and maintenance of connections in a network (including protocols and mechanisms to disseminate this information) as well as algorithms for engineering an optimal path between end points. Support of Grid computing and high-performance applications by the optical control plane emerged from three converging trends:

- *Advances in optical networking technologies*. Widespread deployment of the fiber-optic infrastructure has led to low-cost, high-capacity optical connections.
- *Affordability of the required computational resources through sharing*. New e-science applications' increasing demand of computational power and bandwidth is proving to be costly unless resources are shared across research institutions.
- *The need for interdisciplinary research*. The growing complexity of scientific problems is driving increasing numbers of scientists from diverse disciplines and locations to work together to achieve breakthrough results.

Some high-end Grid applications put unique and challenging demands on the optical network infrastructure. These applications assume a dynamic on-demand use of end-to-end optical networking resources, global transfers of very large data sets across great distances, coordination of network resources with other vital Grid resources (such as CPUs and storage servers), advanced reservations of networking resources, deterministic end-to-end connections (with low jitter and low latency), connection timescales of a few microseconds to long-lived lightpaths, and near-real-time feedback of network performance measurements and resource availability to both the applications and middleware.

To meet these challenges, the optical networking community, in conjunction with the Grid community, must rethink the role of intelligent optical control planes. Existing control plane protocols and architectures are motivated by service provider requirements rather than end-user requirements. Today's networks don't have end-user applications provisioning end-to-end optical connections. Instead, provisioning is a manual function performed from a centralized management application. The connections' endpoints are part of some form of edge-network device (such as an edge router), and the connection's duration is in terms of weeks, months, or years. In contrast, future applications will make on-demand requests for end-to-end optical connections that regard endpoints as workstations, PCs, clusters, sensors, and instruments

Copublished by the IEEE CS and the AIP 1521-9615/05/$20.00 © 2005 IEEE

Collaborative Community

One recent example of a collaborative Grid community is the North Carolina Statewide Grid, currently being built by MCNC in partnership with North Carolina's public and private universities. MCNC Grid Computing and Networking Services is an independent, nonprofit, advanced technology research and service center that develops, tests, and deploys Grid computing and advanced networking solutions in test-bed environments to serve education, research, government, and commercial organizations. MCNC helped create one of the nation's first bioinformatics Grids in 2001, the North Carolina Bioinformatics Grid test bed. As it continues to develop over the next two years, the NC Grid will be one of the nation's first statewide production Grid services networks.

Since the mid 1980s, MCNC has operated the state's North Carolina Research & Education Network (NCREN), a production-level Internet Protocol network that interconnects all of North Carolina's public universities and many of its private universities and colleges, as well as other research, education, government, and commercial organizations. This high-performance, high-speed communications and computing network serves as the backbone for North Carolina's future technology growth and is the foundation for its statewide Grid.

In 2001, MCNC and North Carolina universities, in partnership with Cisco Systems, IBM, and Sun Microsystems, launched the North Carolina BioGrid—one of the nation's first Grid test beds for life sciences research. This Grid offers a reference platform for developing the high-performance computing, data storage, and networking resources needed for bioinformatics and cheminformatics applications. The test bed currently involves resources from the University of North Carolina at Chapel Hill, North Carolina State University, Duke University, and MCNC.

MCNC launched its Enterprise Grid in 2003 to address the needs of a broader range of scientific disciplines and to provide resources for the NC BioGrid and statewide Grid. In addition, MCNC is formalizing its early work in Grid computing with the establishment of the Grid Technology and Evaluation Center. GTEC is a collaborative Grid deployment test bed for applications, infrastructure, and systems architecture that supports interoperability, integration, experimentation, development, and training.

rather than network elements. Moreover, the duration of these connections will be based on the particular application's requirements, from microseconds to hours/days, and the application's request for network resource will have to be coordinated with other required resources such as CPU and storage. A control plane's interactions with applications and Grid middleware represents a paradigm shift for both the optical control plane and application development, and only through the combined efforts of the two communities in the form of vertical integration will such an infrastructure (composed of both hardware and software) be developed.

E-Science and Grid Computing

E-science research and engineering involves a variety of disciplines, including high-energy physics, biology, environmental science, and engineering, most of which use different scientific modes such as remote sensor data collection, simulation and modeling analysis, and remote instrumentation, just to name a few, for advancing scientific knowledge. The simultaneous rapid growth of scientific data along with its wide geographic distribution has rapidly overwhelmed the ability to manage, move, store, and analyze it. Moreover, wide-area networks of instruments, data archives, and simulation facilities lack the bandwidth and performance needed to enable next-generation scientific discovery and mission-critical, on-demand simulations.

Future scientific research will undoubtedly entail scientists, researchers, and technologists working together and interconnecting with a common powerful compute infrastructure, which means Grid computing will probably be one of the most significant enablers to future e-science applications.[1–6] The Global Grid Forum (GGF; www.ggf.org), for example, is already expending a great deal of effort to standardize software and technologies associated with Grid computing. Until recently, the Grid community didn't consider networking resources to be as vital as CPU and storage resources. One of the working groups within the GGF—Grid High Performance Networking (GHPN)—is not only raising the awareness of these resources' importance, but also outlining related network challenges and solutions. Documents produced by this group describe the need for these applications to have access to on-demand end-to-end high-capacity optical connections to meet their high-end requirements.[7]

Today, the e-science community is driving these paradigm shifts. Many global research teams collaborate via government-funded Grid infrastructure for advancing scientific discovery. A prime example is the high-energy physics (HEP) community's experiment at CERN's Large Hadron Collider, to begin data collection in 2007. This community consists of thousands of scientists globally distributed around the world (called a *collaboratory*), sharing and analyzing petabytes of complex data (see Figure 1). The large amount of data to be exchanged among these different regions requires the capacity available today with optical networks using 10-Gbyte links between clusters. Many more such collaboratories exist in all other areas of scientific research; later in this article, we'll look at an example of a very dynamic e-science application used by the astrophysics community.

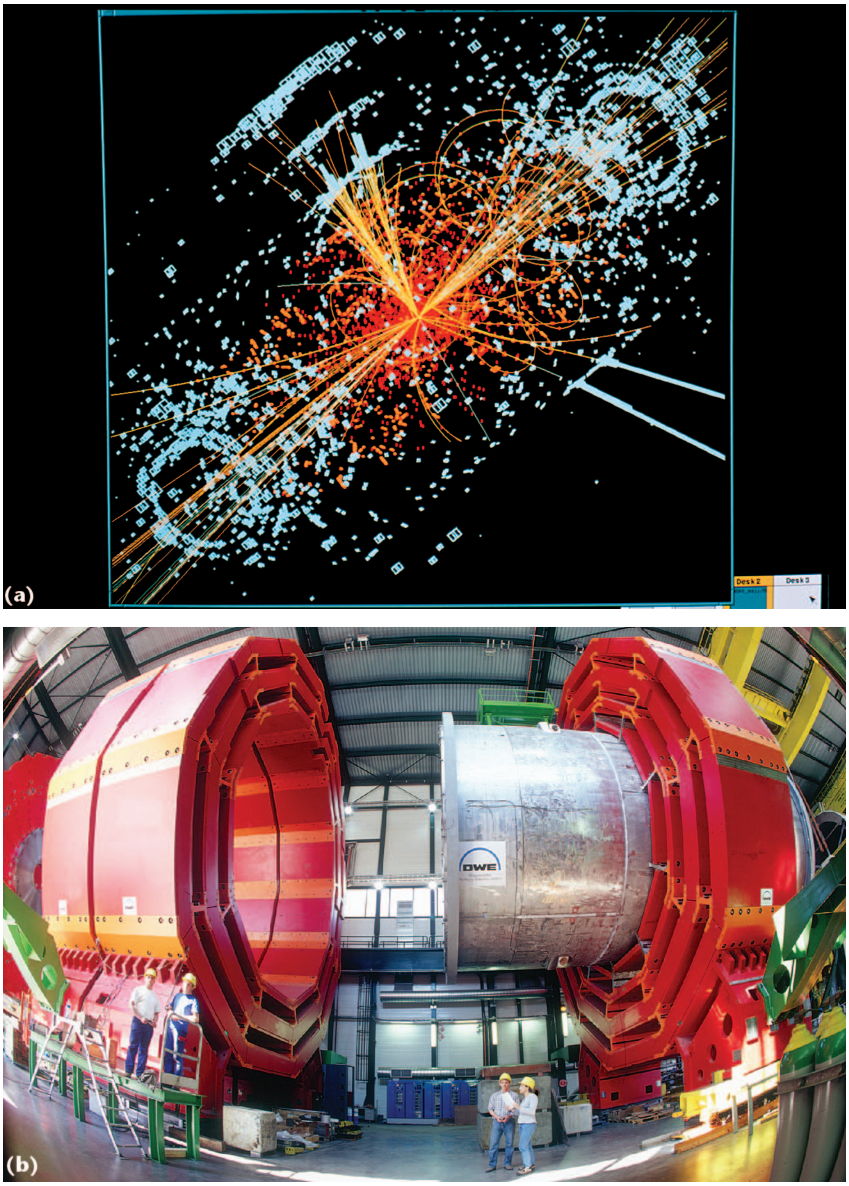

Figure 1. Collaboratories. Grid technologies are essential for distributing data from collaborative research efforts such as those taking place at CERN, the world's largest particle physics research center. Simulation of (a) Higgs decay in (b) the Compact Muon Solenoid detector at CERN's Large Hadron Collider. (Photos courtesy of CERN.)

Next-Generation Optical Networks

World governments have started to realize their role in supporting Grid infrastructures, including the purchase of dark fiber (fiber deployed but not yet lit up using telecommunications equipment) for high-capacity optical networks on a national and international scale. The scale of the infrastructure necessary to advance scientific discovery is very large and similar in scope to other national-scale infrastructures such as power Grids.[6]

The hardware and software to provide this large-scale dynamic infrastructure isn't found with off-the-shelf parts; rather, it's usually in the form of concepts and prototypes. The high priority that nations place on advancing science and engineering for achieving their own goals (as well as humanity's) is enough of an incentive for most governments to provide funding to support such infrastructures. Federal agencies in the US—including the National Science Foundation (NSF), the National Institutes of Health, the Department of Defense, and the Department of Energy—have several programs for researchers seeking funds for studying, developing, and experimenting on next-generation network-centric Grid infrastructures. Other nations have similar activities and programs, with most involving both interdisciplinary and international components.

Government-funded test beds are also available at regional, national, and international levels for e-science and Grid experimentation, and the high-capacity optical network is a key component in most of these test beds. Experimentation conducted on existing carrier-supported networks has shown that the Internet doesn't provide certain high-end e-science applications with the required determinism.[8] However, running those same applications using dedicated end-to-end optical connections resulted in timely, failure-free operations. Research scientists have concluded that today's networks and protocols will have to be rethought and reengineered for tomorrow's scientific community. Meeting this need through the use of on-demand, end-to-end optical connections over dark fiber pro-

Optical Control Plane Workshops

The advent of high-capacity optical networking will soon provide the raw capacity to carry vast amounts of data generated by collaborative e-science Grid communities. However, network control plane challenges will need to be addressed to achieve true interoperability among international network research test beds.

In recognizing this need, MCNC helped organize a series of workshops focused on optical control plane challenges faced by these Grid communities. Workshop participants include a strong concentration of international leaders from optical research networks, such as Canada's CANARIE, the Netherlands' NetherLight, the United Kingdom's UKLight, and the US-based StarLight.

Research areas discussed and presented during these workshops include

- optical connection signaling and provisioning,
- optical recovery (protection and restoration),
- layer interactions,
- optical network performance monitoring, metrics, and analysis,
- security,
- resource discovery,
- topology state information dissemination,
- intra- and interdomain routing,
- integrating advanced optical technology architectures,
- OGSA integration and Web services,
- interaction and coordination with other Grid resources, and
- advanced resource reservation.

Workshop outcomes encompass the strengthening and unification of this global research network community to develop new standards for this emerging technology area. Another important goal is to coordinate a streamlined effort to pursue further research funding in support of designing control planes for Grid infrastructure that can benefit researchers globally.

An optical control plane mailing list has been created for this international research community to exchange ideas in furthering progress on this work; see www.mcnc.org/mcncopticalworkshop/nov04/ for more information regarding these workshops.

vides the required determinism (or quality of service [QoS]). Many examples of such test beds exist around the world, including Canada's CANARIE (see www.canarie.ca/canet4) and the NSF-funded Optiputer (see www.optiputer.org as well as the "Optical Control Plane Workshops" sidebar). Key to the success of these test beds is the interdisciplinary nature of the teams using them and the process of vertical integration from the driving e-science application to the physical layer resources.

An Astrophysics Example

Researchers in the astrophysics community generate highly demanding applications that push the limits of current computation, storage, and network technologies for simulating exotic events in the universe, such as colliding black holes. They run simulations of realistic events that occur in our universe to better understand the dynamics of gravitational waveforms, hoping that this understanding will lead to enhanced identification and interpretation of existing waveforms as well as the capability to predict future events. Driven by their requirements, this community is prototyping a new mode of working that will rely heavily on dynamic, intelligent, network-centric simulations.

Ed Seidel and his team in the Numerical Relativity Group at Louisiana State University's (LSU's) Center for Computation and Technology and astrophysicists at the Albert Einstein Institute of Potsdam engaged in black hole research are experimenting with several highly dynamic e-science applications using the Grid Application Toolkit (GAT; www.gridlab.org). The simulations they run typically require well over 100 Gbytes of memory, generate terabytes of data, and require days of compute time on thousands of processors. Such research is highly computational and data-intensive, and its needs can't be satisfied in a typical cluster environment. Rather, this research could take advantage of several clusters distributed via high-capacity optical networks because the scientists involved need advanced interactive and Grid tools and middleware for managing and visualizing these large amounts of data.

As the amount of data collected or generated for simulations continues to explode, data-mining techniques increasingly require the use of high-end visualization tools as an alternative to analysis methods based on statistical or AI techniques. Visualization exploits the human perception to identify structures, patterns, and anomalies by presenting abstract data visually, letting the user explore the complex information and draw his or her own conclusions. Figure 2, for example, shows the final stages of black hole simulations run on machines at NCSA and NERSC.

Let's look at a typical scenario of a new-generation Grid application, exhibiting intelligent and adaptive behavior within a Grid environment. Suppose the astrophysics researchers at LSU submit a black hole simulation via a portal somewhere on a global Grid. Using an abstract, application-oriented API, such as that provided by a GAT, the researcher's application will use queries made to a Grid information service to determine where to launch the requested simulation. Once the initial target is decided, the application migrates the

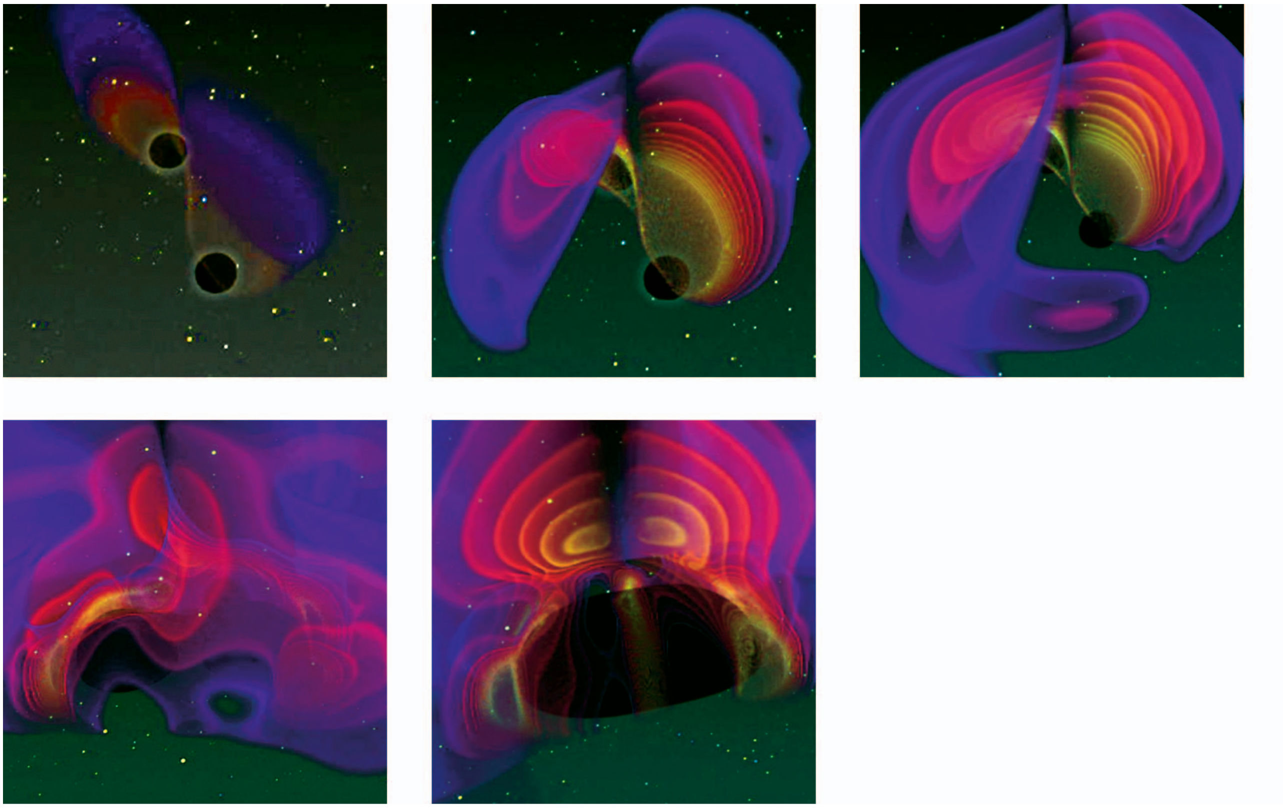

Figure 2. Black hole simulations. The final stages of black hole simulations run on machines at NCSA and NERSC show (from left to right, top to bottom) a pair of orbiting black holes that have spiraled together due to energy carried by gravitational waves. Accurately simulating such collisions, and observing the properties of the emitted gravitational waves, is important for understanding the experimental data generated by gravitational wave detectors.

code to the target and spawns the simulation; the newly spawned code then registers the new location with the Grid information service. As the initial simulation runs, the application might perform one or more of the following scenarios:

- Huge amounts of data result from the simulation, which require storage either local to the simulation or at geographically dispersed locations. If remote storage is necessary, then the application itself creates an on-demand network connection and streams data to the site.
- The simulation application might do near-real-time analysis of the output data and detect a black hole event horizon, which suggests that the parameters and conditions of the simulation are closer to a detection of a black hole. This detection could spawn a new finer-grained simulation at a remote cluster available on the global Grid.
- A slow part of the simulation runs asynchronously, so the application might spawn that part separately.
- New, more powerful resources could become available (the application becomes aware of newly available resources from a feedback loop with Grid resource management middleware), so the simulation might migrate to a faster cluster.
- An end user could interact with the simulation and perform computational steering (interactive control over computational process during runtime).

For the black hole simulation application to perform any of these scenarios, the application must have access to changing resources within the Grid infrastructure as well as reservation access to those resources (such as the rapid creation and deletion of end-to-end optical connections). Grid middleware provides a near-real-time feedback loop of information about various resources so that applications can decide how best to exploit them. Under these conditions, the applications are no longer limited to local resources or resources available only at the time initiated; rather, the application can dynamically adapt to changing resources within a geographically distributed Grid infrastructure. The network control plane plays a key role in providing this vertical integration.

Paradigm Shifts in the Optical Control Plane

As the networking and Grid communities work through the various challenges involved in vertically integrat-

Café Dubois

By Paul Dubois

Parallel Testing

The thrill of victory, the agony of parallelism: when you can use a lot of resources on a cluster to get the job done, the speed improvements can make a big improvement in your life, but the pain involved in getting it to work can be intense.

Recently, my colleague Nu Ai Tang and I set out to improve our testing system. We have a little Python-based testing system that's been a really big success. Our project developers now run more than 200 tests before committing their changes to our main line. Most of these tests take less than one minute, but some run to four or five. (We have some much longer tests that we run once a week.) All of this testing really shows in the stability of the product and the productivity of the developers. Well, do the math: 200 tests at a good fraction of a minute each puts you in the three-hour range. Ouch.

Our computers are clusters of nodes, and each has a certain number of processors, ranging from two per node on some Linux boxes to 16 on some IBM machines. Many of our tests use more than one processor. Jobs that use a lot of processors—or more processors than are on a single node—are shipped off to the batch system for execution. For purposes of this discussion, assume that each test "fits" on one node.

A surprising number of tricky issues arise. The first set concerns the tests themselves, and a second set concerns the use of multiple nodes.

First, some tests depend on other tests. You must be sure that the parent is finished and that the child is only started when the parent is done. The most typical scenario is a test of a restart capability. Test number one runs and writes a dump halfway through: it gets to the stop time, checks the answer, and if it's good, exits with a success status. The next test restarts the code at the midway point and runs to the end, making sure it still gets the same answer.

This wasn't so hard when we ran tests serially. Our test input is in Python, and we used to say,

ing e-science applications down to the optical layer, the optical control plane will have to undergo some major paradigm shifts. Traditional network management functionality is often referred to as FCAPS, an acronym for *f*ault, *c*onfiguration, *p*erformance, and *s*ecurity management; traditional optical networks also use a centralized-operator-controlled method for creating optical connections from one end to the other. In contrast, Grid users and applications initiate end-to-end optical connections via signaling to link connections to end stations rather than network aggregators (or edge nodes). Standards bodies such as the Internet Engineering Task Force, the Optical Internetworking Forum, and the International Telecommunication Union are discussing ways to formalize the optical control plane's protocols and functionality, but they still have yet to consider some of the unique requirements that Grid computing places on the control plane.

Provisioning dynamic high-speed optical interconnects under the complete control of the application is a fundamental step toward a true Grid virtual computer. Most applications used in Grid environments are IP-based, but the TCP protocol as defined today is often regarded as inefficient for transferring large data files over very long distances due to the large round-trip time (RTT). Moreover, TCP's congestion control mechanism regards any loss as potential congestion and reacts by dramatically reducing the rate at which data is sent, which is particularly problematic when sending large data files across long distances. To combat this inefficiency, researchers have developed several variants of IP and non-IP transport protocols.

Although dedicated end-to-end optical connections provide the performance and QoS that demanding applications want, most researchers find this to be an expensive proposition because "end to end" usually refers to end workstations, not aggregated data. Having a single TCP flow over the dedicated optical connection means a dataflow (TCP or other) isn't contending for resources among other flows like in aggregated traffic (such as IP routed traffic). This means that there aren't issues with sharing and fairness, which can sometimes result in long queues and packet loss due to congestion. By reducing queue size and loss, latency and jitter also drop, giving the

```
if test('test1', ...):   test('test2', ...)
```

Here, `test` is a command to the testing system to run a certain test, and the test function will return true if the test passed. (We use the exit status as a pass/fail flag, with the developers putting logic inside the test to detect failure and exiting with a nonzero status if they do fail.)

Now, we have to have a new command, `testif`, which creates a test dependency, and make the test command itself return an object that can run the test on command:

```
test1 = test('test1', ...)
testif(test1, 'test2', ...)
```

The testing system runs the first test and holds the second one in a queue until it knows that the first one passed.

The developers assumed a serial execution for the tests—for example, two tests might create the same files. In ordering the tests within a testing system input file, they might have relied on one being run before the other. Fortunately, since we run the tests simultaneously on two architectures, the developers were already conscious of creating unique filenames, and our little utility library encodes both the architecture and the time of day in generating filenames for the user. But even this might not be safe. We also found that on one of our systems, the nodes have large local disks and the tests were using those disks for restart dumps. This meant we had to be sure a child would run on the same node as its parent, or else it would look at a *different* local disk.

We solved most of these problems by partitioning the test set into families of tests and their direct and indirect dependents, each headed by the "chief" test on which all the others depend. We then made sure that no two members of one family could run at once, that each family runs on the same node, and that no two tests run at once if they involve starting the same script as the family chief of another family, because some tests run the same script with different physics options. Although they don't depend on one another necessarily, we can avoid a lot of easy-to-create conflicts with this rule.

Now we're ready to use all the processors on a single node and dispatch different families to different nodes. In doing so, we encountered even more pain. The starting up of a parallel job involves a considerable amount of machinery, which differs from one architecture to the next. At the very least, you're running a job that's running your job. Add in the `ssh` to start the job on another node, and there can be varied numbers of actors between you and the test running on the other node. It is, as it turns out, quite tricky to get back the correct exit status over all architectures, and it's even harder to *kill* the job (which happens when the user notices a lot of early failures and decides to call the whole thing off). And for a final insult, the `ssh` key to that node has to be preestablished because no human is around when it comes time to accept a new key.

We give each test a priority (which can be overridden in the input) equal to the number of processors it uses plus the number used by all its direct and indirect dependents. At any given time, we start the highest priority remaining

continued on p. 74

dataflow much better QoS and performance. To utilize the dedicated optical connection more efficiently, it would be useful to allow a controlled set of flows over the connection. This type of control requires the optical control plane to interact with the transport layer in a feedback loop to assure each flow is provided the requested QoS.

Obviously, research in Grid computing will continue to be supported and funded, and increased deployment of Grid test beds will provide the networking community with an opportunity to rethink, research, and experiment with the role of an intelligent optical control plane in the emerging network-centric compute environment.

Following through with the research and development of a new optical control plane marked by major paradigm shifts from what currently exists today could lead to a next-generation optical network that vastly differs from today's mode of operation. Research in this vital area shouldn't be constrained by current models or existing infrastructure. Grid computing architectures will continue to evolve and could even change nomenclature, but they're unlikely to disappear.

Acknowledgments

The author acknowledges Admela Jukan, Ed Siedel, Tom Goodale, Gabriele Allen, Wolfgang Gentzsch, Steve Thorpe, Lavanya Ramakrishnan, Mark Johnson, Barbara Capone, and Bonnie Hurst for their insights.

References

1. F. Berman, G. Fox, and A. Hey, eds., *Grid Computing*, John Wiley & Sons, 2003.
2. I. Foster, C. Kesselman, and S. Tuecker, eds., *The Grid 2: Blueprint for a New Computing Infrastructure*, Morgan Kaufmann, 2004.
3. A. Abbas, *Grid Computing: A Practical Guide to Technology and Applications*, Charles River Media, 2003.
4. Wolfgang Gentzsch, "The Grid: Defining the Future of the Internet," *GRIDtoday*, vol. 3, no. 33, 16 Aug. 2004; www.gridtoday.com/04/0816/103659.html.
5. D. Reed et al., *HECRTF Workshop Report*, Nat'l Coordination Office for Information Technology Research and Development, June 2003; www.itrd.gov/hecrtf-outreach/20040112_cra_hecrtf_report.pdf.
6. D. Atkins et al., *Revolutionizing Science and Engineering through Cyberinfrastructure*, tech. report, Nat'l Science Foundation, 2003.
7. D. Simeonidou et al., *Optical Network Infrastructure for the Grid*, tech. report GFD.36, Global Grid Forum, 2004.
8. T. DeFanti et al., "TransLight, A Global-Scale

continued from p. 73

job that will fit (subject to the family-on-one-node restriction). This seems to keep the nodes pretty well packed until they run out of jobs toward the end.

The opiate that removes all this pain is seeing the test suite execute in about 10 minutes. If only we could build that fast. We've tried multinode builds, and they're even trickier.

Generator Expressions in Python 2.4

Python 2.4 is out (www.python.org). Take a moment as you download it to donate to the Python Software Foundation. We just finished awarding about US$40,000 in grants to applicants who proposed interesting projects that would benefit the community. One of the winning entries concerns preparation of educational materials for scientists using Python for science.

The most intriguing new addition to the language is a little hard to explain. Remember list comprehensions? If *s* is a sequence, a list comprehension creates a new list from it. For example,

```
s = [1.,2.,3.,4.,5.]
t = [x**2 for x in s]
w = [x**2 for x in s if x%2 == 0]
```

Here *t* is a list of the first five squares, and *w* is just the even ones. List comprehensions are faster than a loop that appends to an initially empty list.

Suppose, however, that what you really intend to do is add up these numbers:

```
total = sum([x**2 for x in s])
```

If *s* is large, this has a big space disadvantage compared to the loop:

```
total = 0.0
for x in s:
 total += x**2
```

It's certainly easier to read and write the first form, but it creates a list as long as *s* to do it.

Well, in Python 2.4 you can use a *generator expression*, which creates an iterator that will iterate through a sequence and return a value the iterator computes:

```
total = sum(x**2 for x in s)
```

Let's slow this down to understand it better. The parentheses around the generator expression are required, but the ones from a function operating on it will do:

```
>>> sum(x**2 for x in s)
30.0
>>> iter = (x**2 for x in s)
>>> type(iter)
<type 'generator'>
>>> sum(iter)
30.0
```

The sum function requests the next member of the sequence it thinks it's working on, and the iterator produces it on the fly. Pretty neat, eh?

Be sure to read the documentation on generator expressions, because they're less transparent than list comprehensions. In particular, *x* won't have a value after the total is computed. And, we'd better not reuse our iterator:

```
>>> sum(iter)
0.0
```

The iterator was consumed doing the first sum, and it believes there's nothing left in its sequence.

LambdaGrid for EScience," *Comm. ACM*, vol. 46, no. 11, 2003, pp. 35–41.

Gigi Karmous-Edwards is principal scientist in the Advanced Technology Group at MCNC Grid Computing and Network Services. Her research interests include novel optical technologies and their role in Grid computing. Karmous-Edwards received a BS in chemical engineering and an MS in electrical engineering, both from North Carolina State University. She is a member of the IEEE Communications Society. Contact her at gigi@mcnc.org.