WEAVING COMPUTATIONAL GRIDS: HOW ANALOGOUS ARE THEY WITH ELECTRICAL GRIDS?

Can computational grids make as great an impact in the 21st century as electrical grids did in the 20th? A comparison of the two technologies could provide clues about how to make computational grids pervasive, dependable, and convenient.

Following Alessandro Volta's invention of the electrical battery in 1800, Thomas Edison paved the way for electricity's widespread use by inventing the electric bulb. Figure 1 shows Volta demonstrating the battery for Napoleon I in 1801 at the French National Institute, Paris. Whether or not Volta envisioned it, his invention evolved into a worldwide electrical power grid that provides dependable, consistent, and pervasive access to utility power and has become an integral part of modern society.

We are now witnessing rapid developments in computer networks and distributed and high-performance computing. Inspired by the electrical power grid's pervasiveness and reliability, computer scientists in the mid-1990s began exploring the design and development of a new infrastructure, *computational power grids* for network computing.[1] Emerging computational grids currently serve scientists working on large-scale, data- and resource-intensive applications that require more computing power than a computer, a supercomputer, or a cluster can provide in a single domain. This need for greater computing power has driven advances in scalable computing—from distributed parallel computing on local-area networks of PCs and workstations (cluster computing)[2,3] to distributed computing on high-end computers connected by wide-area networks across multiple domains. Computational grids are an extension of the scalable computing concept: Internet-based networks of geographically distributed computing resources that scientists can share, select from, and aggregate to solve large-scale problems. The research and developmental work for implementing such grids is proceeding at a very brisk pace; their performance and ease of use could reach the level of the electrical power grid within a few years.

In this article, we describe how computational grids developed, their layered structure, and their emerging operational model, which we envisage as providing seamless, utility-like access to computational resources. We also attempt to show the similarities and dissimilarities between this system, still in its infancy, and the mature electrical power grid. By identifying quantities and parameters that are analogous between the two grids, we hope that we

1521-9615/02/$17.00 © 2002 IEEE

MADHU CHETTY
Monash University
RAJKUMAR BUYYA
University of Melbourne, Australia

Figure 1. Volta demonstrates the battery for Napoleon I at the French National Institute, Paris, in 1801. The painting is from the Zoological Section of "La Specula" (N. Cianfanelli, 1841), at the National History Museum, Florence University, Italy.

can bring to light areas in computational grid development that need more focus.

Computational grids

Computational grids are already being used to solve large-scale problems in science, engineering, and commerce—the "Applications of Grid Computing" sidebar lists some of the more prominent applications and projects. The advantages of this approach to computing are many.[4] For example, grids

- Enable resource sharing
- Provide transparent access to remote resources
- Allow on-demand aggregation of resources at multiple sites
- Reduce execution time for large-scale, data-processing applications
- Provide access to remote databases and software
- Take advantage of time zone and random diversity (in peak hours, users can access resources in off-peak zones)
- Provide the flexibility to meet unforeseen emergency demands by renting external resources for a required period instead of owning them

The enabling factors in the creation of computational grids have been the proliferation of the Internet and the Web and the availability of low-cost, high-performance computers.[1]

Technological milestones

Compared to the history of the electrical power grid, which spans more than two centuries, the computational grid—rather, the entire computer communication infrastructure, the Internet—has a history of less than half a cen-

Applications of Grid Computing

Many application domains in which large processing problems can easily be divided into subproblems and solved independently are already taking great advantage of grid computing. These include Monte Carlo simulations and parameter sweep applications, such as ionization chamber calibration,[1] drug design,[2] operations research, electronic CAD, and ecological modeling.

On other fronts, projects such as Distributed.net, launched in 1997, and SETI@home, launched in 1999, attracted worldwide attention to peer-to-peer computing (P2P).[3] Millions of participants contributed their PCs' idle CPU cycles: for Distributed.net, they processed RSA Labs RC5-32/12/7 (56-bit) secret key challenge; participants in SETI@home processed a database of large pulsar signals in a search for extraterrestrial intelligence. Emerging from these successes are the notions of virtual organizations[4] and virtual enterprises,[5] which could develop a computational economy for sharing and aggregating resources to solve problems.

References

1. D. Abramson, J. Giddy, and L. Kotler, "High-Performance Parametric Modeling with Nimrod-G: Killer Application for the Global Grid?" *Proc. Int'l Parallel and Distributed Processing Symp.*, IEEE CS Press, Los Alamitos, Calif., 2000.
2. R. Buyya et al., "The Virtual Laboratory: A Toolset to Enable Distributed Molecular Modelling for Drug Design on the World-Wide Grid," *J. Concurrency and Computation: Practice and Experience*, to be published, 2002; currently available as a tech. report from Monash University, Melbourne, Australia.
3. A. Oram, ed., *Peer-to-Peer: Harnessing the Power of Disruptive Technologies*, O'Reilly Press, Sebastopol, Calif., 2001.
4. I. Foster, C. Kesselman, and S. Tuecke, "The Anatomy of the Grid: Enabling Scalable Virtual Organizations," *Int'l J. Supercomputer Applications*, vol. 15, no. 3, 2001.
5. R. Buyya et al., "Economic Models for Management of Resources in Peer-to-Peer and Grid Computing," *Proc. SPIE Int'l Conf. on Commercial Applications for High-Performance Computing*, SPIE, Bellingham, Wash., 2001.

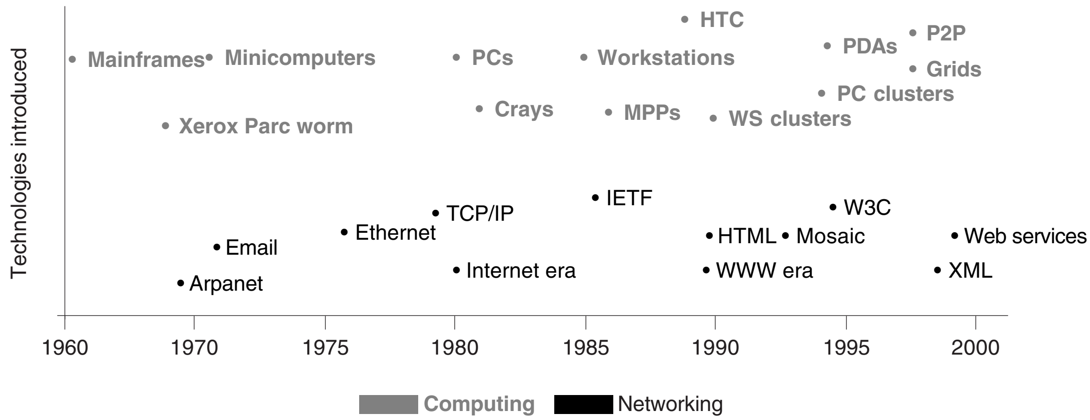

Figure 2. Major milestones in networking and computing technologies from 1960 to the present. Along with technological advances have come the rise and fall of various systems. In the 1960s, mainframes (mainly from IBM) served the needs of computing users, but a decade later DEC's less-expensive minicomputers absorbed the mainframe's market share. During the 1980s, vector computers such as Crays and, later, parallel computers such as massively parallel processors became the systems of choice for grand-challenge applications.

tury. Figure 2 outlines the major technological advances in networking and computing leading to the emergence of peer-to-peer networks[5] and computational grids.[1]

Communication. The computational grid's communication infrastructure is the Internet, which began as a modest research network supported by the US Department of Defense's Advanced Research Projects Agency. DARPA's effort began as a response to the USSR's launch of Sputnik, the first artificial earth satellite, in 1957. From September to December of 1969, DARPA launched Arpanet's original four nodes—at the University of California, Los Angeles; Stanford Research Institute; University of California, Santa Barbara; and the University of Utah. By the mid-1970s, Arpanet's Internet work embraced more than 30 universities, military sites, and government contractors, and its user base had expanded to include the greater computer science research community.

In 1973, Bob Metcalfe outlined the idea for Ethernet, a local-area network to interconnect computers and peripherals, in his doctoral dissertation at Harvard; Ethernet came into existence in 1976.[6] In 1974, Vint Cerf and Bob Kahn proposed the transmission control protocol, which split into TCP/IP in 1978.

In 1983, Arpanet still consisted of several hundred computers on a few local area networks. In 1985, the National Science Foundation arranged with DARPA to support a collaboration of supercomputing centers and computer science researchers across Arpanet. In 1986, the Internet Engineering Task Force (IETF) formed as a loosely self-organized group of people who contributed to the engineering and evolution of Internet technologies.[7] In 1989, responsibility for and management of Arpanet officially passed from military interests to the academically oriented NSF. Much of the Internet's etiquette and rules of behavior evolved during this period.

The Web—invented in 1989 by Tim Berners-Lee of CERN, Switzerland, as a way to easily share information—fueled a major revolution in computing.[8] Its language, HTML, provided a standard means of creating and organizing documents; HTTP protocols, browsers, and servers provided ways to link these documents and access them online transparently, regardless of their location. The World Wide Web Consortium (www.w3c.org), formed in 1994, is developing new standards for information interchange. For example, work on XML (Extensible Markup Language) aims to provide a framework for developing software that can be delivered as a utility service via the Internet.

Computation. The idea of harnessing unused CPU cycles emerged in the early 1970s, when computers were first linked by networks. (See the History of Distributed Computing and other sites listed in the "Distributed and Grid Computing Web Sites" sidebar.) Arpanet ran a few early experiments with distributed computing, and in 1973, the Xerox Palo Alto Research Center installed the first Ethernet network. Scien-

Distributed and Grid Computing Web Sites

Distributed.net, Project RC5	www.distributed.net/rc5
Global Grid Forum	www.gridforum.org
Grid Computing Info Centre	www.gridcomputing.com
IEEE Distributed Systems Online	http://dsonline.computer.org
History of Distributed Computing	www.ud.com/company/dc/history.htm
Hobbes' Internet Timeline	www.zakon.org/robert/internet/timeline
Peer-to-Peer Working Group	www.p2pwg.org
SETI@home	http://setiathome.ssl.berkeley.edu

tists at PARC developed a worm program that roamed about 100 Ethernet-connected computers, replicating itself in each machine's memory. Each worm used idle resources to perform a computation and could reproduce and transmit clones to other nodes of the network. With the worms, developers distributed graphic images and shared computations for rendering realistic computer graphics.

Since 1990, distributed computing has reached a new, global level. The availability of powerful PCs and workstations and high-speed networks (such as gigabit Ethernet) as commodity components has led to the emergence of clusters for high-performance computing.[9] The availability of such clusters within many organizations has fostered a growing interest in aggregating distributed resources to solve large-scale problems of multi-institutional interest.

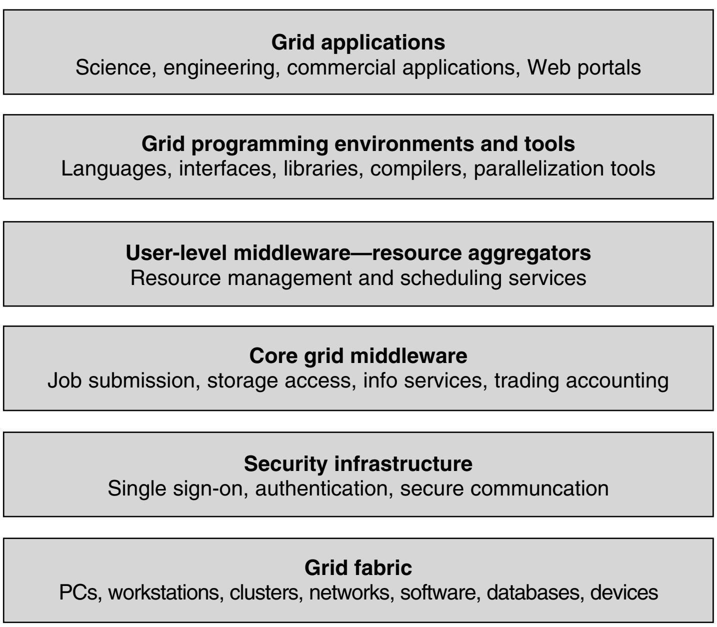

Figure 3. A layered architecture for the computational grid and related technologies.

Computational grids and peer-to-peer computing are the results of this interest. The grid community generally focuses on aggregating distributed high-end machines such as clusters, whereas the P2P community concentrates on sharing low-end systems such as PCs connected to the Internet. P2P networks can amass computing power, as does the SETI@home project, or share contents, as do Napster and Gnutella. Given the number of grid and P2P projects and forums that began worldwide in early 2000, it is clear that interest in the research, development, and commercial deployment of these technologies is burgeoning.[10]

Layered structure

A computational grid consists of several components—from enabling resources to end-user applications. Figure 3 shows a computational grid's layered architecture. (Ian Foster, Carl Kesselman, and Steven Tuecke discuss another comprehensive architecture for the grid.[11])

At the bottom of the grid stack, we have distributed resources managed by a local resource manager with a local policy and interconnected through local- or wide-area networks. Thus, the bottom layer serves as *grid fabric*. This fabric incorporates

- Computers such as PCs, workstations, or SMPs (symmetric multiprocessors) running operating systems such as Unix or Windows
- Clusters running various operating systems
- Resource management systems such as Load Sharing Facility, Condor, Portable Batch System, and Sun Grid Engine
- Storage devices
- Databases
- Special scientific instruments such as radio telescopes and sensors

The next layer, *security infrastructure*, provides secure and authorized access to grid resources. Above that, *core grid middleware* offers uniform, secure access to resources (it can also implement a security layer). The next two layers are *user-level middleware*, consisting of resource brokers or schedulers responsible for aggregating resources; and *grid programming environments and tools*. Resource brokers manage execution of applications on distributed resources using appropriate scheduling strategies. Developers use the grid development tools to grid-enable applications. The top layer consists of *grid applications*, which range from collaborative computing to remote access, scientific instruments, and simulations.

Operational model

For the operation of a computational grid, the broker discovers resources that the user can access through grid information servers, negotiates with grid-enabled resources or their agents using middleware services, maps tasks to resources (scheduling), stages the application and data for processing (deployment), and finally gathers results.[12] The broker also monitors application execution progress and manages changes in the grid infrastructure and resource failures (see Figure 4). Several projects worldwide are actively exploring the development of various grid-computing system components, services, and applications.

The grid environments comprise heterogeneous resources, fabric management systems (single-system image OSs, queuing systems, and so on) and policies, and scientific, engineering, and commercial applications with varied requirements (they can be CPU-, I/O-, memory-, or network-intensive). The *producers* (also called resource owners) and *consumers* (the grid's users) have different goals, objectives, strategies, and demand patterns.[13] More importantly, both resources and end users are geographically distributed, inhabiting multiple time zones.

Researchers have proposed several approaches for resource management architectures; the prominent ones are centralized, decentralized, and hierarchical. Traditional approaches use centralized policies that need complete state information and a common fabric management policy, or a decentralized consensus-based policy. These approaches attempt to optimize a systemwide performance measure. However, because of the complexity of constructing successful grid environments, it is impossible to define either an acceptable systemwide performance matrix or a common fabric management policy, so the traditional approaches are not suitable. Therefore, hierarchical and decentralized approaches are better suited to grid resource and operational management.[13]

Within these approaches, there exist different economic models for managing and regulating resource supply and demand.[14] The grid resource broker mediates between producers and consumers. Producers and consumers can both grid-enable resources by deploying low-level middleware systems on them. On producers' grid resources, the core middleware handles resource access authorization, letting producers give resource access only to authorized users. On consumers' machines, the user-level middleware lets them grid-enable applications or produce the necessary coupling technology for executing legacy applications on the grid.

On authenticating to the grid, consumers interact with resource brokers to execute their application on remote resources. The resource broker takes care of resource discovery, selection, aggregation, and data and program trans-

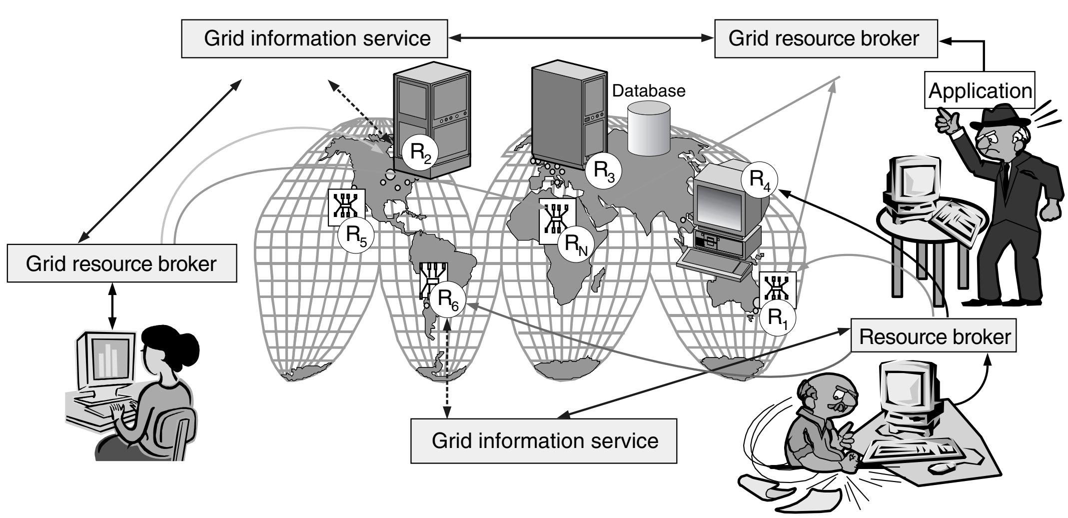

Figure 4. A generic view of a computational power grid.

Table 1. Electrical and computational power grids: A comparison.

Parameter	Electrical power grid	Computational power grid
Resources	Heterogeneous: thermal, hydro, wind, solar, nuclear, others	Heterogeneous: PCs, workstations, clusters, and others; driven by different operating and management systems
Network	Transmission lines, underground cables. Various sophisticated schemes for line protection.	Internet is the carrier for connecting distributed resorces, load, and so on.
Analogous quantities	Bus Energy transmission Voltage Bulk transmission system (230 kV to 760 kV) Subtransmission (25 kV to 150 kV) Distribution (120/240V, 25 kV) Cable Energy (MW-hour) Only small storage capacity in the form of DC batteries	Node Computational transmission Bandwidth Bulk transmission by fiberoptic-OC48, ATM (2.4 Gbps) Ethernet, T-3 (45 Mbps) Modem, ISDN, and so on (56 to 128 Kbps) Cable Computational power (Mflops) Any magnitude of storage (Mbytes)
Power source	Power station (turbogenerators, hydrogenerators), windmill	Grid resource (computers, data sources, Web services, databases)
Load type (based on use type)	Heterogeneous application devices: for example, mechanical energy for fans, electricity for TVs, heat for irons	Heterogeneous applications: for example, graphics for multimedia applications, problem solving for scientific or engineering applications
Operating frequency	Uniform: 50 or 60 Hz DC systems also exist Analog quantity, sinusoidal	Nonuniform: Depends on computer processing power and clock speed. Digital, square wave
Access interface	Direct: Wall socket for small consumers, transformer for industrial consumers	Uniform interface to heterogeneous resources: for example, Globus GRAM interface for submitting jobs to resources[16]
Ease of use	Very simple: Plug and play	Very complex: Expected to change as computing portals and network-enabled solvers[17] emerge
Matching device to varying power levels (voltage, bandwidth, CPU speed)	Transformer changes voltage levels to match, for example, a 25 V device with a 220 V supply.	Resource brokers select resources to meet user requirements such as quality and cost. Applications can run on machines with different capabilities, so devices like transformers aren't required.
Aggregation of resources	When a load requires more power than can be provided locally, the grid provides additional power. Economic dispatch center uses sophisticated scheduling algorithms and load-flow studies that provide the mechanisms to carry this out.	When an application needs more computational power than a single resource can provide, or for faster execution, computational grids allow resource aggregation for executing application components in parallel. Grid resource brokers such as Nimrod-G[12] provide resource aggregation capability.
Reliability	Important lines are duplicated. Sophisticated protection schemes exist for power stations, transmission lines, equipment, and so on.	Resources in a grid may fail without notice. Resource brokers must handle such failure issues at runtime.

portation; it initiates execution on a remote machine, and it gathers the results.

Comparing the grids

The trends in computational and network technologies that led to the emergence of computational grids is similar to the technological evolution that resulted in the electrical power grid. Historically, the notions of analogy, similarity, and generality of phenomena have frequently given researchers increased perspective and provided greater perceptual significance in their investigations. Indeed, advances in physics have confirmed and continue to confirm that many objective processes are subject to general

Parameter	Electrical power grid	Computational power grid
Stability	Stability is crucial for keeping the generators in sync. Sophisticated control algorithms ensure automated mechanism.	Stability depends on resource management policy. If resource is shared, available computing power for a user can vary.
Transmission capacity	Maximum upper limit for the lines depends on the lines' thermal limits.	Upper limit depends on carrier's bandwidth capability.
Security/safety	Fuses, circuit breakers, and so on	Firewalls, public-key infrastructure, and PKI-based grid security[18]
Cogeneration (consumers own power generators working seamlessly with global grids)	Optional	Optional
Storage	Only storage for low-power DC using batteries.	No storage of computational power is possible.
Automated accounting	Advanced metering and accounting mechanisms are in place.	Local resource management systems support accounting. Resource brokers can meter resource consumption (Nimrod-G agent does application-level metering); global-level service exchange and accounting mechanisms such as GridBank[19] are required.
Interconnection	Various regional power pools are interconnected by weak connections called tie-lines.	Internet provides connectivity service; tools such as JobQueue in Legion[20] and Condor-G[21] can provide federation resources with tight coupling.
Unregulated grid operation	Successful operation in countries with sufficient generation capacity.	Not yet. As this technology matures and businesses start taking advantage of it, we believe this will come into picture.
Regulated grid operation	Load dispatch center manages optimal system operation.	Greater potential exists for using market-based pricing mechanisms to help regulate resource supply and demand.[13,14]
Regulators	In general, managed by an autonomous body of vendors and government regulators—for example, NEMMCO in Australia (www.nemmco.com.au).	No regulator yet exists. However, the need for a watchdog will grow as the grid enters mainstream computing. Some national supercomputing centers (for example, in the UK[22]) have a facility management committee that decides on token allocation and value in CPU time per sec., which varies according to the resource. This resembles price regulation in a single administrative domain, which can be extended to the national level with appropriate cooperation and understanding among all such centers.
Standards body	Many standardization bodies exist for various components, devices, system operation, and so on. (For example, the IEEE publishes standards on transformers, harmonics, and so on.)	Forums such as Global Grid Forum and the P2P Working Group promote community practices. The IETF and W3C handle Internet and Web standardization issues.

laws and are therefore described by similar equations. For example, based on similarity relations, we can apply a unified mathematical approach to different branches of science—for example, we can use the same approach to oscillations as we use with different kinds of waves.

Inspired by the significance of similarities, therefore, we are investigating analogies and similarities between computational power grids and the electrical power grid. Such a comparison will let us establish that the progress toward developing a computational grid is analogous to the electrical grid's development.

Based on the structure and operating models of the two grids, we can easily identify several analogous elements, which we present in Table 1. Although most of the parameters are self-explanatory, we'll examine and discuss some of them in greater detail.

Resources

The modern power grid has a wide variety of power resources. It typically derives 70 percent of its electricity from coal, gas, and oil; 15 percent from hydropower; and 15 percent from nuclear generation.[15] Although only to a small degree, new prime-energy resources—solar, wind, wave geothermal, and tidal powers, and photovoltaic energy—also contribute to grid power. Most generating stations for fossil-fired power are mine-mouth stations—that is, they are located close to mines. Although the resources in the electrical power grid are heterogeneous, they produce an identical output: electricity that is thoroughly uniform—a sinusoidal signal (voltage or current) at 50 or 60 Hz.

Similar to the electrical grid, the computational grid draws on a wide variety of computational resources. Supercomputers, clusters, and SMPs that include low-end systems such as PCs and workstations are connected in a grid to give the user seamless computing power. In addition, devices for visualization, storage systems and databases, special classes of scientific instruments (such as radio telescopes), computational kernels, and other resources are also logically coupled and presented to the user as a single, integrated resource (see Figure 4).

Clearly, heterogeneity is inherent in nature. For centuries, it has been prevalent in the electrical grid. Therefore, computational grid technologies and applications should be designed to handle and take advantage of heterogeneity that is present in resources, systems, and management policies.

Network

An electric power system, even the smallest one, constitutes an electric network of vast complexity. However, in any of these systems, a transmission line's voltage level determines its energy transmission capacity. By increasing the voltage level and physical length of the transmission network, we can create a superhighway that can transmit large blocks of electric energy over large distances.

As shown in Figure 5, a typical power network is characterized by three transmission systems: transmission, subtransmission, and distribution. The *transmission system* handles the largest blocks of power and interconnects all the system's generator stations and major loading points. The energy can be routed, generally, in any desired direction on the transmission system's various links to achieve the best overall operating economy or to best serve a technical objective. The *subtransmission system* serves a larger geographical area and compared to the distribution system it distributes energy in large blocks at high voltage levels. The distribution system is very similar to the subtransmission system, but it constitutes the finest meshes (overhead and underground) in the overall network; it distributes power mainly to residential consumers.

In a computational grid, the resources (and loads) are connected by the Internet, using gateways and routers to form a LAN and give the client computers of that network services such as file transfer, email, and document printing. A LAN can connect to other LANs to form a WAN. The network's bandwidth (a measure of its data-handling capacity) is analogous to the electrical network's voltage levels (a measure of power-handling capacity). Analogous to the electrical grid's transmission system for bulk power transfers are the computational grid's optical network and ATM connections for large data transfers. Similarly, the computational grid's T1, E1, and Ethernet connections are analogous to the subtransmission system; modem connections correspond to the distribution system.

System load

The electric power grid can support various forms of load—electrical load for such things as televisions, mechanical load for fans and the like, heat for devices such as irons, and so on. Similarly, the computational grid's load can also be heterogeneous, varying with the scope of problem to be solved (the number of parameters involved, for example) and its nature (whether it is I/O- or computation-intensive, for example). However, a resource broker hides the complexities of aggregating a diverse set of resources. This technique for solving massively parallel problems is very much analogous to feeding a large electric load from several distributed generators in the electrical grid. However, unlike the power grid, where the user is unaware of which generators are delivering power to which load, the computational grid provides clear evidence of the resources carrying out the computations.

Operational model

While various mechanisms for the computational grid's operation are still in the research and development phase,[10,13] the electrical grid's operational model is established and ubiquitous. Traditionally, operation of the electrical grid has been monopolistic. Its load dispatch and operation center continually manages the system's generation requirements based on the load demand. However, since the 1980s, much effort has gone into restructuring the power industry's traditional monopoly to introduce fair competition and improve economic efficiency. We discuss both these modes of operation briefly in the hope of providing a goal or benchmark for a future operational model for the computational grid.

Under *regulated power system operation*, a common practice is determining the total generation required at any time and how it should be distributed among the various power stations and the generators within each of these plants. Output of each power station and each of the generating units within the power station is commonly computer controlled for stable power system operation. By continually monitoring all plant outputs and the power flowing in interconnections, the computer system also controls the interchange of power with other systems. The term *area* refers to that part of an interconnected system in which one or more companies control generation to meet all their own load requirements. If an area experiences insufficient generation, the monitoring computer system implements a prearranged net interchange of power with other areas for specified periods. Monitoring the flow of power on the *tie-lines* between areas determines whether a particular area is satisfactorily meeting the load requirements within its own boundaries. Thus, automatic system operation ensures that an area meets its own load requirements, provides the agreed net interchange with neighboring areas, determines the desired generation of each plant in the area for economic dispatch, and ensures that the area provides its share.

Since the 1980s, efforts to restructure the power industry have led to *unregulated power system operation*. At the core of the changes are the creation of mechanisms for power suppliers—and sometimes large consumers—to openly trade electricity. However, the emergent electricity market is more akin to an oligopoly than to perfect market competition.[23] This is due to special features of the electricity supply industry—for example, a limited number of producers, large investment size (creating barriers to entry), transmission constraints that isolate consumers from the effective reach of many generators, and transmission losses that discourage consumers from purchasing power from distant suppliers. Thus, electricity markets are not perfectly competitive.

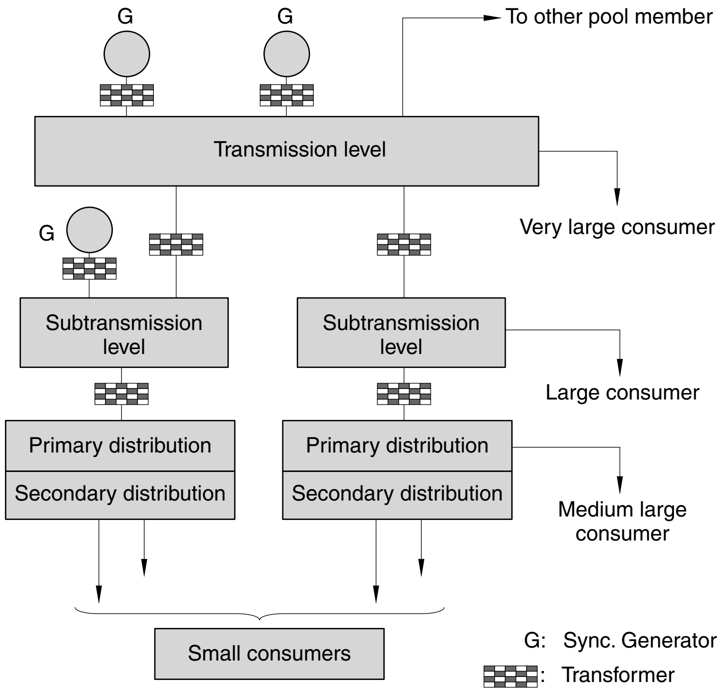

Figure 5. Power system diagram.

In recent years, some research has focused on optimal bidding strategies for competitive generators or large consumers, and also on a market in which sealed-bid auctions and uniform price rules are prevalent.[24] Broadly speaking, there are three ways to develop optimal bidding strategies. The first relies on estimations of the next trading period's market clearing price. The second uses techniques such as probability analysis and fuzzy sets to estimate rival participants' bidding behavior.[5] The third approach is to apply methods or techniques from game theory.[25,26] Further, there are a great variety of auction methods (for example, static and dynamic), as well as auction and bidding protocols, such as single-part bid, multipart bidding, iterative bidding, and demand-side bidding.[23]

Dissimilarities in the two grids

Obviously, the electrical and computational grids are not completely identical. Certain aspects of the two grids' dissimilarities are instructive. For example, the power system comprises several buses (junction points) or nodes, which are interconnected by transmission line

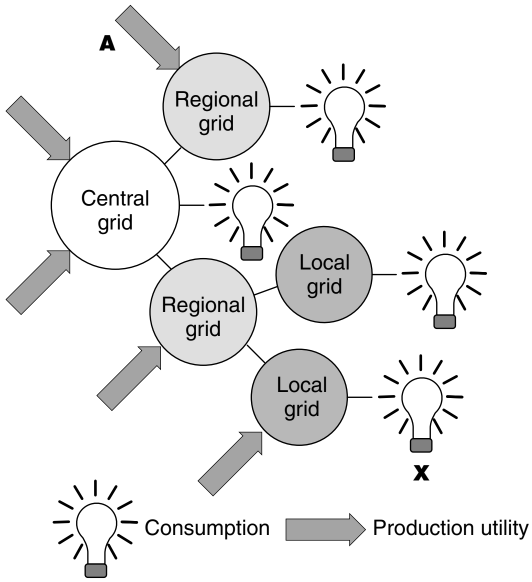

Figure 6. A schematic overview of the three levels of grid.[27]

networks. Power is injected into a bus from generators, and loads are tapped from it. At this stage, such an arrangement is not possible in the computational grid. Furthermore, besides conventional AC transmission, the electrical grid has implemented other transmission methods, such as high-voltage DC (HVDC) and underground transmission. The computational grid network does not have equivalent heterogeneous transmission for information and data.

Again, for economic and technological reasons, most electrical systems are interconnected into vast power grids, which are subdivided into regional operating groups called power pools,[27] as illustrated in Figure 6. Producer A can sell power to consumer X at a well-defined price in competition with all the other producers. Although each individual power system within such a pool usually has independent technical and economic operation, it is contractually tied to the other pool members in handling certain generation and scheduling features. Such an arrangement does not exist in the computational grid, but it could be implemented when there is greater cooperation among the participants, with resource-sharing policies that are globally acceptable. Some examples of such emerging grid tools are Condor-G[21] and Legion's JobQueue schedulers.[20]

Moreover, in computational grids, drawing power from the grid means pushing data or applications to a resource, processing it, and subsequently pulling results. This is not the situation in an electrical power grid, where the users can access (pull) the power as soon as they are connected. To make the computational grid work on that model, users' data and applications must be compatible with resource properties, or universalizing tehcnologies like Java must be used.

The electrical power grid is one of the most advanced and evolved grids in existence; the computational grid is a new and emerging field, now in a state in which the electrical power grid was almost a century ago. A true marketplace for the computational grid is yet to emerge. The use of computational grids for solving real-world problems is still limited to research labs and a highly specialized scientific community funded by government agencies. Pushing grids into mainstream computing will require major advances in grid programming, application development tools, application- and data-level security, and grid economy.[4]

Our comparison of the computational grid to the electric grid brings to light other deficiencies in computational grids as they are now. The need for an operational model (a regulated system or otherwise), proper division of the computational grid into regional pools, coordinated system operation to ensure network stability, and ease of use must all be priorities in further grid development. CiSE

Acknowledgments

We thank Domenico Laforenza, Ajith Abraham, Rob Gray, David Walker, Alexander Reinefeld, and Frank Karbarz for their constructive comments. We thank David Abramson for his encouragement and support. We extend special thanks to Domenico Laforenza for providing us a copy of the photograph included in Figure 1. Computational grid content is derived from Buyya's PhD thesis.[4]

References

1. I. Foster and C. Kesselman, eds., *The Grid: Blueprint for a Future Computing Infrastructure*, Morgan Kaufmann, San Francisco, 1999.
2. A. Raman et al., "PARDISC: A Cost-Effective Model for Parallel and Distributed Computing," *Proc. Third Int'l Conf. High-Perfor-*

mance Computing, IEEE CS Press, Los Alamitos, Calif., 1996.

3. R. Buyya, ed., *High Performance Cluster Computing: Architectures and Systems, Volume 1*, Prentice Hall, Old Tappan, N.J., 1999.
4. R. Buyya, "Economic-based Distributed Resource Management and Scheduling for Grid Computing," PhD thesis, Monash Univ., Australia, April 2002; www.buyya.com/thesis.
5. A. Oram, ed., *Peer-to-Peer: Harnessing the Power of Disruptive Technologies*, O'Reilly Press, Sebastopol, Calif., 2001.
6. R. Metcalfe and D. Boggs, "Ethernet: Distributed Packet Switching for Local Computer Networks," *Proc. ACM Nat'l Computer Conf.*, vol. 19, no. 5, July 1976.
7. S. Harris, "The Tao of IETF—A Novice's Guide to the Internet Engineering Task Force," 2001; www.ietf.cnri.reston.va.us/rfc/rfc3160.txt, current March 2002.
8. T. Berners-Lee, *Weaving the Web: The Past, Present, and Future of the World Wide Web By Its Inventor*, Orion Publishing Group, London, 1999.
9. R. Buyya, ed., *High Performance Cluster Computing,* vols. 1 and 2, Prentice Hall, Old Tappan, N.J., 1999.
10. M. Baker, R. Buyya, and D. Laforenza, "The Grid: International Efforts in Global Computing," *Proc. Int'l Conf. Advances in Infrastructure for Electronic Business, Science, and Education on the Internet*, SSGRR, Rome, 2000.
11. I. Foster, C. Kesselman, and S. Tuecke, "The Anatomy of the Grid: Enabling Scalable Virtual Organizations," *Int'l J. Supercomputer Applications*, vol. 15, no. 3, 2001.
12. R. Buyya, D. Abramson, and J. Giddy, "Nimrod-G: An Architecture for a Resource Management and Scheduling System in a Global Computational Grid," *Proc. Fourth Int'l Conf. High-Performance Computing*, Asia-Pacific Region, IEEE CS Press, Los Alamitos, Calif., 2000.
13. R. Buyya, D. Abramson, and J. Giddy, "Economy-Driven Resource Management Architecture for Computational Power Grids," *Proc. Int'l Conf. Parallel and Distributed Processing Techniques and Applications*, CSREA Press, 2000.
14. R. Buyya et al., "Economic Models for Management of Resources in Peer-to-Peer and Grid Computing," *Proc. SPIE Int'l Conf. Commercial Applications for High-Performance Computing*, SPIE, Bellingham, Wash., 2001.
15. O. Elgerd, *Electric Energy Systems Theory: An Introduction*, 2nd ed., McGraw Hill, New York, 1982.
16. I. Foster and C. Kesselman, "Globus: A Metacomputing Infrastructure Toolkit," *Int'l J. Supercomputer Applications*, vol. 11, no. 2, 1997, pp. 115–128.
17. H. Casanova and J. Dongarra, "NetSolve: A Network Server for Solving Computational Science Problems," *Int'l J. Supercomputer Applications and High Performance Computing*, vol. 11, no. 3, Fall 1997.
18. I. Foster et al., "A Security Architecture for Computational Grids," *Proc. 5th ACM Conf. Computer and Communications Security*, ACM Press, New York, 1998.
19. R. Buyya, J. Giddy, and D. Abramson, "A Case for Economy Grid Architecture for Service-Oriented Grid Computing," *10th IEEE Int'l Heterogeneous Computing Workshop*, IEEE CS Press, Los Alamitos, Calif., 2001.
20. D. Katramatos et al., "JobQueue: A Computational Grid-Wide Queuing System," *Proc. 2nd Int'l Workshop Grid Computing*, Springer-Verlag, Berlin, 2001.
21. J. Frey et al., "Condor-G: A Computation Management Agent for Multi-Institutional Grids," *Proc. 10th Int'l Symp. High-Performance Distributed Computing*, IEEE CS Press, Los Alamitos, Calif., 2001.
22. J. Brooke et al., "Mini-Grids: Effective Test-beds for Grid Application," *Proc. 1st IEEE/ACM Int'l Workshop Grid Computing*, Springer-Verlag, Berlin, 2000.
23. A. David and F. Wen, "Strategic Bidding in Competitive Electricity Markets: A Literature Survey," *Proc. IEEE Power Engineering Society Summer Meeting*, IEEE Press, Piscataway, N.J., 2000.
24. J. Lamont and S. Rajan, "Strategic Bidding in an Energy Brokerage," *IEEE Trans. Power Systems*, vol. 12, no. 4, 1997.
25. R. Ferrero, S. Shahidehpur, and V. Ramesh, "Transaction Analysis in Deregulated Power Systems Using Game Theory," *IEEE Trans. Power Systems*, vol. 12, no. 3, 1997.
26. Z. Younes and M. Ilic, "Generation Strategies for Gaming Transmission Constraints: Will the Deregulated Electric Power Market Be an Oligopoly?" *Decision Support Systems*, vol. 24, nos.3–4, 1999, pp. 207–222.
27. P. Myrseth, "The Nordic Power Market and Its Use of Electronic Commerce," *Proc. OECD Workshop on Business-to-Business Electronic Commerce: Status, Economic Impact and Policy Implications*, www.nr.no/~pmyrseth/artikler/oecd_ie_wokshop_99; current March 2002.

Madhu Chetty is a lecturer in the Gippsland School of Computing and Information Technology, Monash University, Churchill, Australia. His current research interests are resource scheduling in computational grids, optimization, symbolic computation, and artificial intelligence applications. He has a PhD in electrical engineering from Nagpur University, India, and has more than 20 years of teaching and research experience. Contact him at madhu.chetty@infotech.monash.edu.au.

Rajkumar Buyya is leading the Grid Computing and Distributed Systems Laboratory in the School of Computer Science and Software Engineering, University of Melbourne, Australia. His research interests include operating systems and parallel and distributing computing. He contributed to the development of system software for the PARAM supercomputers produced by India's Centre for Development of Advanced Computing; He is the author of *Microprocessor x86 Programming, Mastering* C++, and *Design of PARAS Microkernel*, and the editor of *High Performance Cluster Computing* (Prentice Hall, 1999). He was awarded the Dharma Ratnakara Memorial Trust Gold Medal for academic excellence by Mysore and Kuvempu Universities. Contact him at rajkumar@buyya.com.