# The Worldwide Computer

By David P. Anderson
and John Kubiatowicz

An operating system spanning the Internet would bring the power of millions of the world's Internet-connected PCs to everyone's fingertips

## When Mary gets home

from work and goes to her PC to check e-mail, the PC isn't just sitting there. It's working for a biotech company, matching gene sequences to a library of protein molecules. Its DSL connection is busy downloading a block of radio telescope data to be analyzed later. Its disk contains, in addition to Mary's own files, encrypted fragments of thousands of other files. Occasionally one of these fragments is read and transmitted; it's part of a movie that someone is watching in Helsinki. Then Mary moves the mouse, and this activity abruptly stops. Now the PC and its network connection are all hers.

This sharing of resources doesn't stop at her desktop computer. The laptop computer in her satchel is turned off, but its disk is filled with bits and pieces of other people's files, as part of a distributed backup system. Mary's critical files are backed up in the same way, saved on dozens of disks around the world.

Later, Mary watches an independent film on her Internet-connected digital television, using a pay-per-view system. The movie is assembled on the fly from fragments on several hundred computers belonging to people like her.

Mary's computers are moonlighting for other people. But they're not giving anything away for free. As her PC works, pennies trickle into her virtual bank account. The payments come from the biotech company, the movie system and the backup service. Instead of buying expensive "server farms," these companies are renting time and space, not just on Mary's two computers but on millions of others as well. It's a win-win situation. The companies save money on hardware, which enables, for instance, the movie-viewing service to offer obscure movies. Mary earns a little cash, her files are backed up, and she gets to watch an indie film. All this could happen with an Internet-scale operating system (ISOS) to provide the necessary "glue" to link the processing and storage capabilities of millions of independent computers.

### Internet-Scale Applications

ALTHOUGH MARY'S WORLD is fictional—and an Internet-scale operating system does not yet exist—developers have already produced a number of Internet-scale, or peer-to-peer, applications that attempt to tap the vast array of underutilized machines available through the Internet [*see box on page 42*].

PHILIP HOWE

These applications accomplish goals that would be difficult, unaffordable or impossible to attain using dedicated computers. Further, today's systems are just the beginning: we can easily conceive of archival services that could be relied on for hundreds of years and intelligent search engines for tomorrow's Semantic Web [see "The Semantic Web," by Tim Berners-Lee, James Hendler and Ora Lassila; SCIENTIFIC AMERICAN, May 2001].

Unfortunately, the creation of Internet-scale applications remains an imposing challenge. Developers must build each new application from the ground up, with much effort spent on technical matters, such as maintaining a database of users, that have little to do with the application itself. If Internet-scale applications are to become mainstream, these infrastructure issues must be dealt with once and for all.

We can gain inspiration for eliminating this duplicate effort from operating systems such as Unix and Microsoft Windows. An operating system provides a virtual computing environment in which programs operate as if they were in sole possession of the computer. It shields programmers from the painful details of memory and disk allocation, communication protocols, scheduling of myriad processes, and interfaces to devices for data input and output. An operating system greatly simplifies the development of new computer programs. Similarly, an Internet-scale operating system would simplify the development of new distributed applications.

**More than 150 MILLION hosts are connected to the Internet, and the number is GROWING exponentially.**

## Existing Distributed Systems

### COMPUTING

**GIMPS (Great Internet Mersenne Prime Search): www.mersenne.org/**
Searches for large prime numbers. About 130,000 people are signed up, and five new primes have been found, including the largest prime known, which has four million digits.

**distributed.net: www.distributed.net/**
Has decrypted several messages by using brute-force searches through the space of possible encryption keys. More than 100 billion keys are tried each second on its current decryption project. Also searches for sets of numbers called optimal Golomb rulers, which have applications in coding and communications.

**SETI@home (Search for Extraterrestrial Intelligence): http://setiathome.berkeley.edu/**
Analyzes radio telescope data, searching for signals of extraterrestrial origin. A total of 3.4 million users have devoted more than 800,000 years of processor time to the task.

**folding@home: http://folding.stanford.edu/**
Run by Vijay Pande's group in the chemistry department at Stanford University, this project has about 20,000 computers performing molecular-dynamics simulations of how proteins fold, including the folding of Alzheimer amyloid-beta protein.

**Intel/United Devices cancer research project: http://members.ud.com/projects/cancer/**
Searches for possible cancer drugs by testing which of 3.5 billion molecules are best shaped to bind to any one of eight proteins that cancers need to grow.

### STORAGE

**Napster: www.napster.com/**
Allowed users to share digital music. A central database stored the locations of all files, but data were transferred directly between user systems. Songwriters and music publishers brought a class-action lawsuit against Napster. The parties reached an agreement whereby rights to the music would be licensed to Napster and artists would be paid, but the new fee-based service had not started as of January 2002.

**Gnutella: www.gnutella.com/**
Provides a private, secure shared file system. There is no central server; instead a request for a file is passed from each computer to all its neighbors.

**Freenet: http://freenetproject.org/**
Offers a similar service to Gnutella but uses a better file-location protocol. Designed to keep file requesters and suppliers anonymous and to make it difficult for a host owner to determine or be held responsible for the Freenet files stored on his computer.

**Mojo Nation: www.mojonation.net/**
Also similar to Gnutella, but files are broken into small pieces that are stored on different computers to improve the rate at which data can be uploaded to the network. A virtual payment system encourages users to provide resources.

**Fasttrack P2P Stack: www.fasttrack.nu/**
A peer-to-peer system in which more powerful computers become search hubs as needed. This software underlies the Grokster, MusicCity ("Morpheus") and KaZaA file-sharing services.

An ISOS consists of a thin layer of software (an ISOS agent) that runs on each "host" computer (such as Mary's) and a central coordinating system that runs on one or more ISOS server complexes. This veneer of software would provide only the core functions of allocating and scheduling resources for each task, handling communication among host computers and determining the reimbursement required for each machine. This type of operating system, called a microkernel, relegates higher-level functions to programs that make use of the operating system but are not a part of it. For instance, Mary would not use the ISOS directly to save her files as pieces distributed across the Internet. She might run a backup application that used ISOS functions to do that for her. The ISOS would use principles borrowed from economics to apportion computing resources to different users efficiently and fairly and to compensate the owners of the resources.

Two broad types of applications might benefit from an ISOS. The first is distributed data processing, such as physical simulations, radio signal analysis, genetic analysis, computer graphics rendering and financial modeling. The second is distributed online services, such as file storage systems, databases, hosting of Web sites, streaming media (such as online video) and advanced Web search engines.

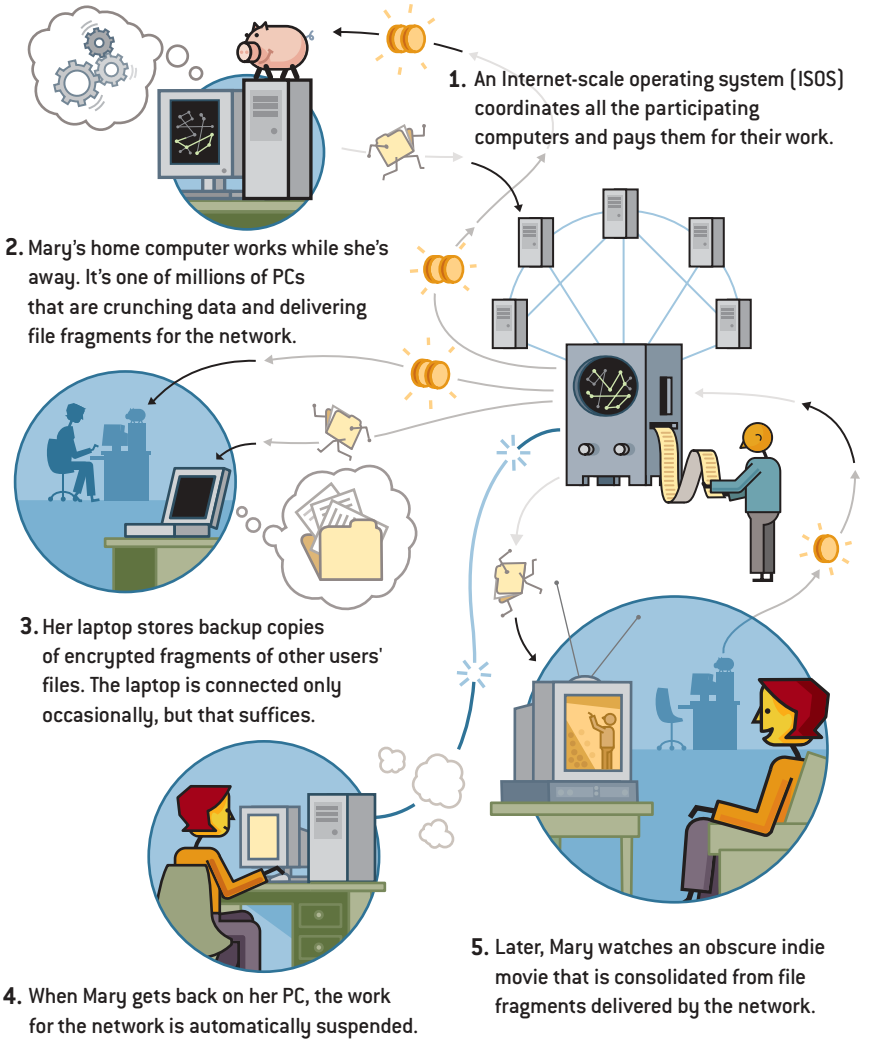
## What's Mine Is Yours

COMPUTING TODAY operates predominantly as a private resource; organizations and individuals own the systems that they use. An ISOS would facilitate a new paradigm in which it would be routine to make use of resources all across the Internet. The resource pool—hosts able to compute or store data and networks able to transfer data between hosts—would still be individually owned, but they could work for anyone. Hosts would include desktops, laptops, server computers, network-attached storage devices and maybe handheld devices.

The Internet resource pool differs from private resource pools in several important ways. More than 150 million hosts are connected to the Internet, and the number is growing exponentially. Consequently, an ISOS could provide a virtual computer with potentially 150 million times the processing speed and storage capacity of a typical single computer. Even when this virtual computer is divided up among many users, and after one allows for the overhead of running the network, the result is a bigger, faster and cheaper computer than the users could own privately. Continual upgrading of the resource pool's hardware causes the total speed and capacity of this über-computer to increase even faster than the number of connected hosts. Also, the pool is self-maintaining: when

## MOONLIGHTING COMPUTERS

With Internet-scale applications, PCs around the world can work during times when they would otherwise sit idle. Here's how it works:



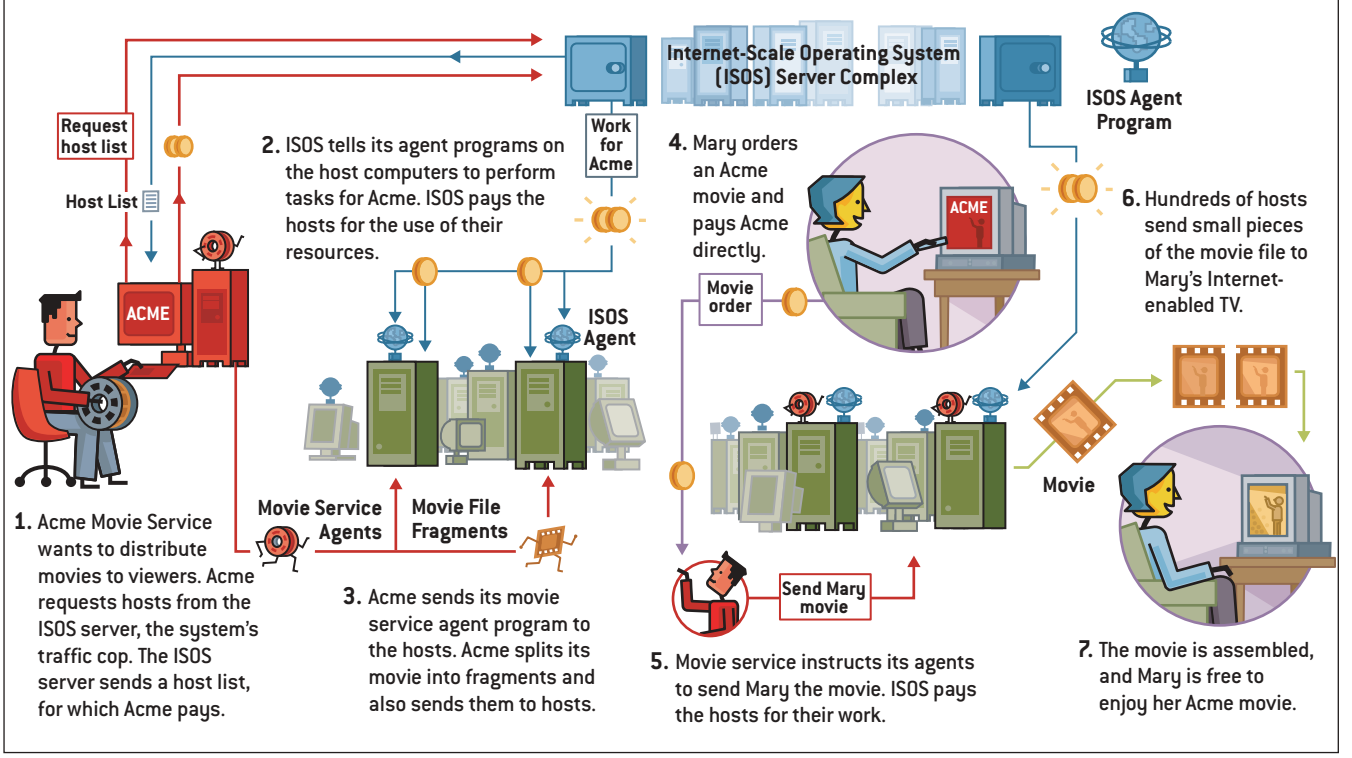


THE AUTHORS

*DAVID P. ANDERSON* and *JOHN KUBIATOWICZ* are both associated with the University of California, Berkeley. Anderson was on the faculty of the computer science department from 1985 to 1991. He is now director of the SETI@home project and chief science officer of United Devices, a provider of distributed computing software that is allied with the distributed.net project. Kubiatowicz is an assistant professor of computer science at Berkeley and is chief architect of OceanStore, a distributed storage system under development with many of the properties required for an ISOS.


XPLANE

## HOW A DISTRIBUTED SERVICE WOULD OPERATE




a computer breaks down, its owner eventually fixes or replaces it.

Extraordinary parallel data transmission is possible with the Internet resource pool. Consider Mary's movie, being uploaded in fragments from perhaps 200 hosts. Each host may be a PC connected to the Internet by an antiquated 56k modem—far too slow to show a high-quality video—but combined they could deliver 10 megabits a second, better than a cable modem. Data stored in a distributed system are available from any location (with appropriate security safeguards) and can survive disasters that knock out sections of the resource pool. Great security is also possible, with systems that could not be compromised without breaking into, say, 10,000 computers.

In this way, the Internet-resource paradigm can increase the bounds of what is possible (such as higher speeds or larger data sets) for some applications, whereas for others it can lower the cost. For certain applications it may do neither—it's a paradigm, not a panacea. And designing an ISOS also presents a number of obstacles.

Some characteristics of the resource pool create difficulties that an ISOS must deal with. The resource pool is heterogeneous: Hosts have different processor types and operating systems. They have varying amounts of memory and disk space and a wide range of Internet connection speeds. Some hosts are behind firewalls or other similar layers of software that prohibit or hinder incoming connections. Many hosts in the pool are available only sporadically; desktop PCs are turned off at night, and laptops and systems using modems are frequently not connected. Hosts disappear unpredictably—sometimes permanently—and new hosts appear.

The ISOS must also take care not to antagonize the owners of hosts. It must have a minimal impact on the non-ISOS uses of the hosts, and it must respect limitations that owners may impose, such as allowing a host to be used only at night or only for specific types of applications. Yet the ISOS cannot trust every host to play by the rules in return for its own good behavior. Owners can inspect and modify the activities of their hosts. Curious and malicious users may attempt to disrupt, cheat or spoof the system. All these problems have a major influence on the design of an ISOS.

## Who Gets What?

AN INTERNET-SCALE operating system must address two fundamental issues—how to allocate resources and how to compensate resource suppliers. A model based on economic principles in which suppliers lease resources to consumers can deal with both issues at once. In the 1980s researchers at Xerox PARC proposed and analyzed economic approaches to apportioning computer resources. More recently, Mojo Nation developed a file-sharing system in which users are paid in a virtual currency ("mojo") for use of their resources and they in turn must pay mojo to use the system. Such economic models encourage owners to allow their resources to be used by other organizations, and theory shows that they lead to optimal allocation of resources.

Even with 150 million hosts at its disposal, the ISOS will be dealing in "scarce"

XPLANE

resources, because some tasks will request and be capable of using essentially unlimited resources. As it constantly decides where to run data-processing jobs and how to allocate storage space, the ISOS must try to perform tasks as cheaply as possible. It must also be fair, not allowing one task to run efficiently at the expense of another. Making these criteria precise—and devising scheduling algorithms to achieve them, even approximately—are areas of active research.

The economic system for a shared network must define the basic units of a resource, such as the use of a megabyte of disk space for a day, and assign values that take into account properties such as the rate, or bandwidth, at which the storage can be accessed and how frequently it is available to the network. The system must also define how resources are bought and sold (whether they are paid for in advance, for instance) and how prices are determined (by auction or by a price-setting middleman).

Within this framework, the ISOS must accurately and securely keep track of resource usage. The ISOS would have an internal bank with accounts for suppliers and consumers that it must credit or debit according to resource usage. Participants can convert between ISOS currency and real money. The ISOS must also ensure that any guarantees of resource availability can be met: Mary doesn't want her movie to grind to a halt partway through. The economic system lets resource suppliers control how their resources are used. For example, a PC owner might specify that her computer's processor can't be used between 9 A.M. and 5 P.M. unless a very high price is paid.

Money, of course, encourages fraud, and ISOS participants have many ways to try to defraud one another. For instance, resource sellers, by modifying or fooling the ISOS agent program running on their computer, may return fictitious results without doing any computation. Researchers have explored statistical methods for detecting malicious or malfunctioning hosts. A recent idea for preventing unearned computation credit is to ensure that each work unit has a number of intermediate results that the server can quickly check and that can be obtained only by performing the entire computation. Other approaches are needed to prevent fraud in data storage and service provision.

The cost of ISOS resources to end users will converge to a fraction of the cost of owning the hardware. Ideally, this fraction will be large enough to encourage owners to participate and small enough to make many Internet-scale applications economically feasible. A typical PC owner might see the system as a barter economy in which he gets free services, such as file backup and Web hosting, in exchange for the use of his otherwise idle processor time and disk space.

## A Basic Architecture

WE ADVOCATE two basic principles in our ISOS design: a minimal core operating system and control by central servers.

**Curious and malicious USERS may attempt to DISRUPT, CHEAT or spoof the system.**

## Primes and Crimes

By Graham P. Collins

NO ONE HAS SEEN signs of extraterrestrials using a distributed computation project (yet), but people have found the largest-known prime numbers, five-figure reward money—and legal trouble.

The Great Internet Mersenne Prime Search (GIMPS), operating since 1996, has turned up five extremely large prime numbers so far. The fifth and largest was discovered in November 2001 by 20-year-old Michael Cameron of Owen Sound, Ontario. Mersenne primes can be expressed as $2^P - 1$, where $P$ is itself a prime number. Cameron's is $2^{13,466,917} - 1$, which would take four million digits to write out. His computer spent 45 days discovering that his number is a prime; altogether the GIMPS network expended 13,000 years of computer time eliminating other numbers that could have been the 39th Mersenne.

The 38th Mersenne prime, a mere two million digits long, earned its discoverer (Nayan Hajratwala of Plymouth, Mich.) a $50,000 reward for being the first prime with more than a million digits. A prime with 10 million digits will win someone $100,000.

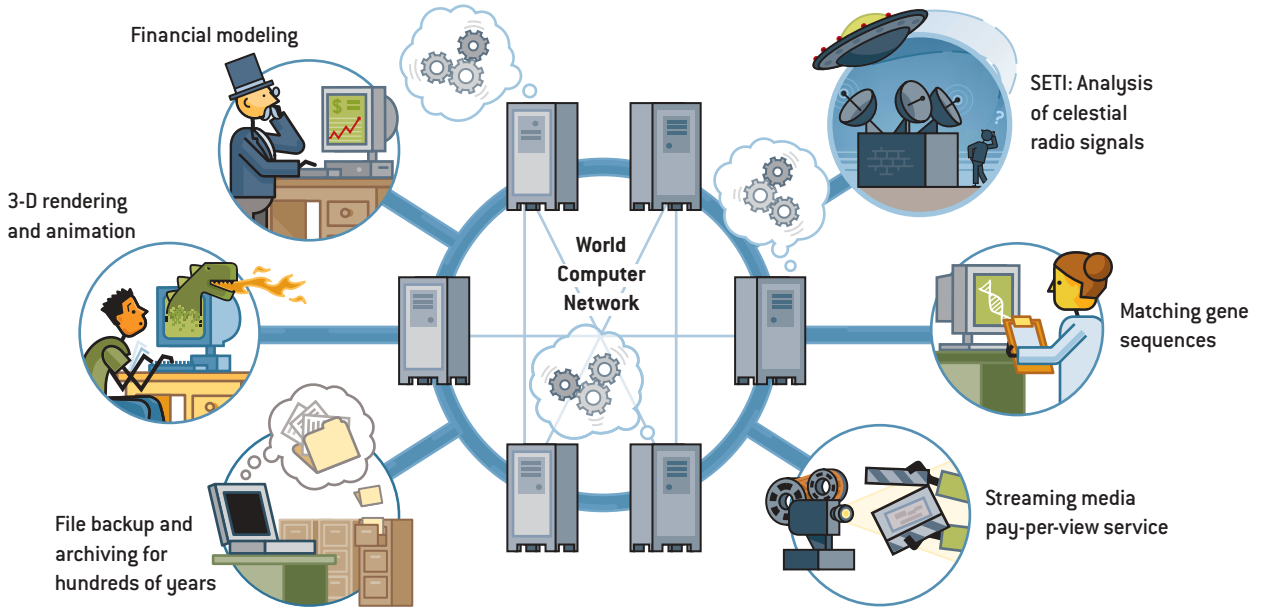
A Georgia computer technician, on the other hand, has found nothing but trouble through distributed computation. In 1999 David McOwen installed the client program for the "distributed.net" decryption project on computers in seven offices of the DeKalb Technical Institute, along with Y2K upgrades. During the Christmas holidays, the computers' activity was noticed, including small data uploads and downloads each day. In January 2000 McOwen was suspended, and he resigned soon thereafter.

Case closed? Case barely opened: The Georgia Bureau of Investigation spent 18 months investigating McOwen as a computer criminal, and in October 2001 he was charged with eight felonies under Georgia's computer crime law. The one count of computer theft and seven counts of computer trespass each carry a $50,000 fine and up to 15 years in prison. On January 17, a deal was announced whereby McOwen will serve one year of probation, pay $2,100 in restitution and perform 80 hours of community service unrelated to computers or technology.

*Graham P. Collins is a staff writer and editor.*

## WHAT AN INTERNET-SCALE OPERATING SYSTEM COULD DO

By harnessing the massive unused computing resources of the global network, an ISOS would make short work of daunting number-crunching tasks and data storage. Here are just a few of the possibilities:




A computer operating system that provides only core functions is called a microkernel. Higher-level functions are built on top of it as user programs, allowing them to be debugged and replaced more easily. This approach was pioneered in academic research systems and has influenced some commercial systems, such as Windows NT. Most well-known operating systems, however, are not microkernels.

The core facilities of an ISOS include resource allocation (long-term assignment of hosts' processing power and storage), scheduling (putting jobs into queues, both across the system and within individual hosts), accounting of resource usage, and the basic mechanisms for distributing and executing application programs. The ISOS should not duplicate features of local operating systems running on hosts.

The system should be coordinated by servers operated by the ISOS provider, which could be a government-funded organization or a consortium of companies that are major resource sellers and buyers. (One can imagine competing ISOS providers, but we will keep things simple and assume a unique provider.) Centralization runs against the egalitarian approach popular in some peer-to-peer systems, but central servers are needed to ensure privacy of sensitive data, such as accounting data and other information about the resource hosts. Centralization might seem to require a control system that will become excessively large and unwieldy as the number of ISOS-connected hosts increases, and it appears to introduce a bottleneck that will choke the system anytime it is unavailable. These fears are unfounded: a reasonable number of servers can easily store information about every Internet-connected host and communicate with them regularly. Napster, for example, handled almost 60 million clients using a central server. Redundancy can be built into the server complex, and most ISOS online services can continue operating even with the servers temporarily unavailable.

The ISOS server complex would maintain databases of resource descriptions, usage policies and task descriptions. The resource descriptions include, for example, the host's operating system, processor type and speed, total and free disk space, memory space, performance statistics of its network connections, and statistical descriptions of when it is powered on and connected to the network. Usage policies spell out the rules an owner has dictated for using her resources. Task descriptions include the resources assigned to an online service and the queued jobs of a data-processing task.

To make their computers available to the network, resource sellers contact the server complex (for instance, through a Web site) to download and install an ISOS agent program, to link resources to their ISOS account, and so on. The ISOS agent manages the host's resource usage. Periodically it obtains from the ISOS server complex a list of tasks to perform.

Resource buyers send the servers task requests and application agent programs (to be run on hosts). An online service provider can ask the ISOS for a set of hosts on which to run, specifying its resource requirements (for example, a distributed backup service could use sporadically connected resource hosts—Mary's laptop—which would cost less than constantly connected hosts). The ISOS supplies the service with addresses and descriptions of the granted hosts and allows the application agent program to communicate directly between hosts on which it is running. The service can re-

XPLANE

quest new hosts when some become unavailable. The ISOS does not dictate how clients make use of an online service, how the service responds or how clients are charged by the service (unlike the ISOS-controlled payments flowing from resource users to host owners).

## An Application Toolkit

IN PRINCIPLE, the basic facilities of the ISOS—resource allocation, scheduling and communication—are sufficient to construct a wide variety of applications. Most applications, however, will have important subcomponents in common. It is useful, therefore, to have a software toolkit to further assist programmers in building new applications. Code for these facilities will be incorporated into applications on resource hosts. Examples of these facilities include:

**Location independent routing.** Applications running with the ISOS can spread copies of information and instances of computation among millions of resource hosts. They have to be able to access them again. To facilitate this, applications name objects under their purview with Globally Unique Identifiers (GUIDs). These names enable "location independent routing," which is the ability to send queries to objects without knowing their location. A simplistic approach to location independent routing could involve a database of GUIDs on a single machine, but that system is not amenable to handling queries from millions of hosts. Instead the ISOS toolkit distributes the database of GUIDs among resource hosts. This kind of distributed system is being explored in research projects such as the OceanStore persistent data storage project at the University of California at Berkeley.

**Persistent data storage.** Information stored by the ISOS must be able to survive a variety of mishaps. The persistent data facility aids in this task with mechanisms for encoding, reconstructing and repairing data. For maximum survivability, data are encoded with an "*m*-of-*n*" code. An *m*-of-*n* code is similar in principle to a hologram, from which a small piece suffices for reconstructing the whole image. The encoding spreads information over *n* fragments (on *n* resource hosts), any *m* of which are sufficient to reconstruct the data. For instance, the facility might encode a document into 64 fragments, any 16 of which suffice to reconstruct it. Continuous repair is also important. As fragments fail, the repair facility would regenerate them. If properly constructed, a persistent data facility could preserve information for hundreds of years.

**Secure update.** New problems arise when applications need to update stored information. For example, all copies of the information must be updated, and the object's GUID must point to its latest copy. An access control mechanism must prevent unauthorized persons from updating information. The secure update facility relies on Byzantine agreement protocols, in which a set of resource hosts come to a correct decision, even if a third of them are trying to lead the process astray.

**Other facilities.** The toolkit also assists by providing additional facilities, such as format conversion (to handle the heterogeneous nature of hosts) and synchronization libraries (to aid in cooperation among hosts).

An ISOS suffers from a familiar catch-22 that slows the adoption of many new technologies: Until a wide user base exists, only a limited set of applications will be feasible on the ISOS. Conversely, as long as the applications are few, the user base will remain small. But if a critical mass can be achieved by convincing enough developers and users of the intrinsic usefulness of an ISOS, the system should grow rapidly.

The Internet remains an immense untapped resource. The revolutionary rise in popularity of the World Wide Web has not changed that—it has made the resource pool all the larger. An Internet-scale operating system would free programmers to create applications that could run on this World Wide Computer without worrying about the underlying hardware. Who knows what will result? Mary and her computers will be doing things we haven't even imagined. SA

A typical PC owner might see the system as a BARTER ECONOMY that provides free services in exchange for PROCESSOR TIME and DISK SPACE.

### MORE TO EXPLORE


**The Ecology of Computation.** B. A. Huberman. North-Holland, 1988.

**The Grid: Blueprint for a New Computing Infrastructure.** Edited by Ian Foster and Carl Kesselman. Morgan Kaufmann Publishers, 1998.

**Peer-to-Peer: Harnessing the Power of Disruptive Technologies.** Edited by Andy Oram. O'Reilly & Associates, 2001.


Many research projects are working toward an Internet-scale operating system, including:

Chord: **www.pdos.lcs.mit.edu/chord/**

Cosm: **www.mithral.com/projects/cosm/**

Eurogrid: **www.eurogrid.org/**

Farsite: **http://research.microsoft.com/sn/farsite/**

Grid Physics Network (Griphyn): **www.griphyn.org/**

OceanStore: **http://oceanstore.cs.berkeley.edu/**

Particle Physics Data Grid: **www.ppdg.net/**

Pastry: **www.research.microsoft.com/~antr/pastry/**

Tapestry: **www.cs.berkeley.edu/~ravenben/tapestry/**