**Ishfaq Ahmad**
Hong Kong University of Science and Technology
Clear Water Bay, Kowloon
Hong Kong
iahmad@cs.ust.hk

# Network computers: The changing face of computing

Computer system architectures have always evolved over time, but we are currently witnessing an especially dramatic paradigm shift in computer system design.

The first computing paradigm, the mainframe paradigm, provided a centralized control to a set of terminals. While this model offered efficient system and data control and proper resource management, users could not easily modify the system or their applications to meet new requirements. Because the costs of maintaining and administering the mainframe computer were very high, this paradigm also suffered from a poor price-to-performance ratio.

Next came powerful PCs and workstations. These stand-alone machines were heavily loaded with their own computation and storage units, so they could work independently. However, they were not very scalable.

With the advent of fast networking technologies, PC-LAN became more popular, providing a wide array of new functionalities, including e-mail, ftp, and the sharing of devices such as common disks and printers. By giving end users a rich set of distributed resources and considerable control over application behavior, this paradigm offered better scalability and enhanced effectiveness. However, it suffered from two major drawbacks:

- lack of effective security and integrity mechanisms, and
- inefficient communication software to coordinate different devices to complete a complicated job.

Distributed systems then evolved to the three-tier or client–server paradigm, which lets users integrate applications from the network and add unique features on top. In the three-tier approach proposed by the Gartner Research Group, the functionality is separated at the server and client sides, and functions are divided into three modules: data-access logic, application logic, and presentation logic. There are different combinations for dividing these functions on the server and client sides. The *fat client* is like a PC and carries the bulk of all three functions. The *hybrid client* carries some functions and relies on the server, especially for data access. The *thin client* pushes most functionalities back to servers, leaving only the presentation function on the user's desktop. Better centralized control and reduced demand on the client results. However, this approach poses limitations on functionality and might still involve administration and maintenance on the client side.

**The Internet has shifted the distributed computing paradigm from the traditional closely coupled form to a loosely coupled one.**

Concurrent to this evolution came the Internet in its present form, which changed our application-development and content-delivery modes. The Internet has shifted the distributed computing paradigm from the traditional closely coupled form to a loosely coupled one. Closely coupled distributed object models such as Microsoft's Distributed Component Object Model (DCOM), the Object Management Group's Common Request Broker Architecture (CORBA), and Sun's Remote Method Invocation (RMI) let users rely on remote services but maintain control with their local machines. While the loosely coupled model provides a number of attractive features, it is a direct contrast to the Internet, which is intrinsically loosely coupled. A loosely coupled platform lets developers change the implementation at either side of the connection.

The Internet also must be scalable to allow expansion and heterogeneous so that an application on one side does not depend on the implementation on the other side.

The key factor in building the new networked computing infrastructure will be communication, not computing, says C.C. Jay Kuo, (http://sipi.usc.edu/~cckuo/index.html), who leads a major research effort at the University of Southern California to address fundamental system issues of network computing. According to Kuo, the issues that need to be addressed are

- *Scalability*: the system architecture must be fundamentally scalable.
- *Heterogeneity*: support for a heterogeneous environment needs language tools for the back end, front end, and user interface.
- *Real-time support*: existing and future multimedia applications need real-time traffic control for interactivity and quality-of-service requirements.
- *Adaptability*: network I/O bottlenecks must deal with caching, job batching, and adaptive piggybacking techniques.
- *Load balancing*: the system must be load balanced, with both static and dynamic adaptive algorithms.
- *Security*: security issues are highly critical and call for protecting the integrity and privacy of information.
- *Reliability*: the system must be able to detect and mask failures through good redundancy, replication, and recovery techniques.
- *Simplicity*: reduced complexity and limited overhead are required to make the system a manageable commodity.
- *Ease of maintenance and implementation*: the system must be low cost for maintenance and administration to make it a commercially viable solution.

Microsoft is launching a major research effort, known as Microsoft.Net (www.microsoft.com/net/default.asp). This computing infrastructure takes the advantages of closely and loosely coupled systems, combining multiple and incompatible interfaces. It works as an application-specific programming model by assuming that various Web services such as PCs, handheld computers, mobile phones, and Internet TV are its components. Scalability and application reliability are the two key considerations. To enable a wide variety of Web services on a heterogeneous infrastructure, Microsoft is adapting the Extensible Markup Language as a data-description format. According to Microsoft, XML software is not like something installed from CD, but rather like caller ID or pay-per-view television that users can subscribe to through a communication medium.

Sun Microsystems is working on a computing paradigm called Sun Ray Hot Desk Architecture (www.sun.com/products/sunray1/hotdesk.html). In it, all services and resources reside centrally on a server, but the architecture provides a level of performance and access to applications and resources beyond the capabilities of today's desktops. All computing executes on one or more centralized shared machines. The client's terminal contains only the human interface, including input from the keyboard, mouse, voice, or output for display and audio. The objective is to reduce acquisition costs, administration, and desktop maintenance without limiting the desktop's resource capacity. According to Sun, the partitioning of functionalities under the Sun Ray Hot Desk architecture model is analogous to the partitioning between a desktop telephone set and a PBX or PSTN. The appliances connected to the architecture are stateless fixed-function devices like telephone handsets, which enables easy session mobility. The central Solaris-based server, like a PBX, also provides services for incorporating new applications. IBM (Networking Computing Software), Oracle (Oracle Application Server), and Hewlett-Packard are also embracing this new computing concept.

### New commercial applications

Historically, applications-driven computing requirements have always outpaced the available technology, so designers must forever seek faster and more cost-effective computer systems. Parallel and distributed computing takes one large step toward providing computing power that reaches beyond a single-processor system's technological limitations. While the concept of parallel and distributed computing has been with us for over three decades, the application areas have been changing. With the new multimedia-based information technology storm, the trend is changing again. Cluster computing is finding itself on the same popularity bandwagon with information technology generally and the Web in particular. A number of commercial applications have emerged that capitalize on the massive computing power afforded by clusters of commodity off-the-shelf components. (See the sidebar for an especially interesting scientific application.)

### Web servers

A cluster of loosely coupled servers connected by a fast network provides a scalable and fault-tolerant environment for Internet services. The server directs client network connection requests to the different servers and makes the cluster appear as a single virtual service with a single IP address.

One example is the Linux Virtual Server (www.LinuxVirtualServer.org), which has several prototypes installed at many sites to cope with heavy Internet loads. The server performs load-balancing techniques and uses sophisticated scheduling algorithms to direct client network connection requests in a user-transparent fashion. As an added advantage, the server nodes can be replicated for either scalability or high availability.

### Speech-to-e-mail services

As an example of a speech-to-mail system, also called *talking e-mail*, Evoke Talking E-mail (www.evoke.com) lets users record a free voice message and send it as an e-mail message. The service delivers the message to thousands of users simultaneously. Users can also post their voice messages as a link embedded on a Web page.

## Solving the mystery of life with sixfold speedup

To understand the evolutionary process from the beginning of life itself to present-day species, we must first determine the "tree of life." In this tree, called a *phylogeny*, known species reside at the tree's leaves, while conjectured ancestor (extinct) species reside where the tree's branches split. We follow this process down to the tree's base, normally represented by the three major limbs for plants, animals, and single-celled organisms.

In recent years, geneticists have made wondrous progress in determining genetic sequences from generation to generation; they have now mapped complete genomes for several species. Evolutionary models are approximations at best, but nevertheless provide the best guidance for determining the interrelation between species. Biologists are often concerned with a small portion of the phylogeny (a subtree) containing a family of related species. They believe that Nature follows a path of minimizing evolutionary processes, but even computing the minimum evolutionary tree (called *phylogeny reconstruction*) for a handful of species is intractable on several levels.

At the University of New Mexico's Albuquerque High Performance Computing Center (www.ahpcc.unm.edu), David Bader, Bernard Moret, and Tandy Warnow have achieved a nearly one-million-fold speedup on the UNM/Alliance LosLobos supercluster in solving the phylogeny reconstruction problem for the family of twelve Bluebell species. The problem size includes a thirteenth plant, tobacco, used as a distantly related outgroup. The LosLobos supercluster has 512 733-MHz Pentium III processors, interconnected with four 64-way Myrinet 2000 switches and running the Linux 2.2 operating system. The parallelization uses MPI and includes techniques for concurrently evaluating candidate trees and sharing improved upper and lower bounds by the processors.

The LosLobos supercluster executed the problem in about one hour and 40 minutes using the new phylogeny reconstruction code, Grappa (freely available as open source from www.cs.unm.edu/~moret/GRAPPA/). Run on a single processor, Grappa performs 2,500 times faster than previous methods but takes full advantage of parallel processing for additional speedups. Hence, the total speedup for the new solution is one million—equivalent to going from an estimated 200 years down to 100 minutes. Phylogenies derived from gene-order data might prove crucial in answering fundamental open questions in biomolecular evolution

To meet high user demand, the system employs a scalable cluster of Linux system currently comprising 200 CPUs, integrated with a very large disk storage array.

### *Video compression*

Digital video is an essential component of present multimedia applications, including HDTV, home television theatre, Photo-CD, CD-ROM video games, video-on-demand, medical imaging, scientific visualization, video conferencing, multimedia mailing, remote video surveillance, news gathering, and networked database services. The principal impediment to using digitized video is the enormous amount of data required to represent the visual information in digital format. An audio bit stream normally accompanies a video bit stream but is not a problem because of its relatively smaller data requirements.

Because video encoding is more complex and time-consuming than decoding, research efforts generally focus on the encoding process. The objectives of encoding include high visual quality, high compression ratios, and low complexity of the compression algorithm. Encoding can occur in real-time or non-real-time encoding environments. In real-time encoding, encoding must be achieved online to compress about 30 frames/sec, using data from a live source such as a video camera. In non-real-time schemes, encoding can occur offline without strict compression-rate requirements—for example, compressing and storing video sequences for production systems such as digital libraries or large video databases. Developers can use either hardware or software to implement the video encoder and decoder, each having its own advantages and disadvantages.

Video compression can also be done using a hardware- or software-based approach. A hardware approach involves using a special-purpose architecture. A hardware-based approach is easier to use and offers high compression speed. Low-cost hardware using ICs is now available.

In general, video quality depends on the bit rate allowed during the compression: the higher the bit rate, the better the picture quality. The software-based approach lets system developers incorporate new research ideas and algorithms into the encoding process to improve picture quality at a given bit rate or, alternatively, reduce the bit rate for a desired level of picture quality.

However, the typical video application's very high computation requirements often overwhelm single-processor sequential computers. Designers thus need to design efficient and fast algorithms, speeding up the computation by exploiting low-level machine primitives and using parallel processing. Video compression involves processing a large number of operations, and video can be easily decomposed along spatial temporal dimensions. Parallelism therefore offers the most logical method for handling the processing. Research at the Multimedia Technology Research center at Hong Kong University of Science and Technology (www.mtrec.ust.hk) is looking to build software-based video compressors. While the single processor can now yield real-time compression speeds, thanks to advancements in processor speed and development of fast algorithms, one research project at the center involves using cluster computing for production video compression. This approach is beneficial for offline compression of large videos at a very high speed.


### Acknowledgments

The author acknowledges C.C. Jay Kuo, David Bader, Hai Jin, Rajkumar Buyya, Mark Baker, and Amy Apon for supplying useful information.