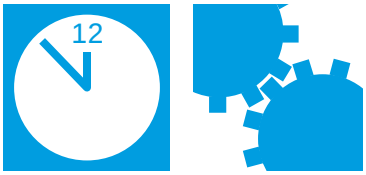

The MAGIC Project: From Vision to Reality

Barbara Fuller, Mitretek Systems
Ira Richer, Corporation for National Research Initiatives

Abstract

In the MAGIC project, three major components — an ATM internetwork, a distributed, network-based storage system, and a terrain visualization application — were designed, implemented, and integrated to create a testbed for demonstrating real-time, interactive exchange of data at high speeds among distributed resources. The testbed was developed as a system, with special consideration to how performance was affected by interactions among the components. This article presents an overview of the project, with emphasis on the challenges associated with implementing a complex distributed system, and with coordinating a multi-organization collaborative project that relied on distributed development. System-level design issues and performance measurements are described, as is a tool that was developed for analyzing performance and diagnosing problems in a distributed system. The management challenges that were encountered and some of the lessons learned during the course of the three-year project are discussed, and a brief summary of MAGIC-II, a recently initiated follow-on project, is given.

Gigabit-per-second networks offer the promise of a major advance in computing and communications: high-speed access to remote resources, including archives, time-critical data sources, and processing power. Over the past six years, there have been several efforts to develop gigabit networks and to demonstrate their utility, the most notable being the five testbeds that were supported by ARPA and National Science Foundation (NSF) funding: Aurora, BLANCA, CASA, Nectar, and VISTAnet [1]. Each of these testbeds comprised a mix of applications and networking technology, with some focusing more heavily on applications and others on networking. The groundbreaking work done in these testbeds had a significant impact on the development of high-speed networking technology and on the rapid progress in this area in the 1990s.

It became clear, however, that a new paradigm for application development was needed in order to realize the full benefits of gigabit networks. Specifically, network-based applications and their supporting resources. such as data servers, must be designed explicitly to operate effectively in a high-speed networking environment. For example, an interactive application working with remote storage devices must compensate for network delays. The MAGIC project, which is the subject of this article, is the first high-speed networking testbed that was implemented according to this paradigm. The major components of the testbed were considered to be interdependent parts of a system, and wherever possible they were designed to optimize end-to-end system performance rather than individual component performance.

The objective of the MAGIC (which stands for "Multidimensional Applications and Gigabit Internetwork Consortium") project was to build a testbed that could demonstrate real-time, interactive exchange of data at gigabit-per-second rates among multiple distributed resources. This objective was pursued through a multidisciplinary effort involving concurrent development and subsequent integration of three testbed components:

- An innovative terrain visualization application that requires massive amounts of remotely stored data
- A distributed image server system with performance sufficient to support the terrain visualization application
- A standards-based high-speed internetwork to link the computing resources required for real-time rendering of the terrain

The three-year project began in mid-1992 and involved the participation, support, and close cooperation of many diverse organizations from government, industry, and academia. These organizations had complementary skills and had the foresight to recognize the benefits of collaboration. The principal MAGIC research participants were:

- Earth Resources Observation System Data Center, U.S. Geological Survey (EDC)[1]
- Lawrence Berkeley National Laboratory, U.S. Department of Energy (LBNL)[1]
- Minnesota Supercomputer Center, Inc. (MSCI)[1]
- MITRE Corporation[1]
- Sprint
- SRI International (SRI)[1]

The work reported here was performed while the authors were with the MITRE Corp. in Bedford, MA, and was supported by the Advanced Research Project Agency (ARPA) under contract F19628-94-D-001.

[1]*These organizations were funded by ARPA.*

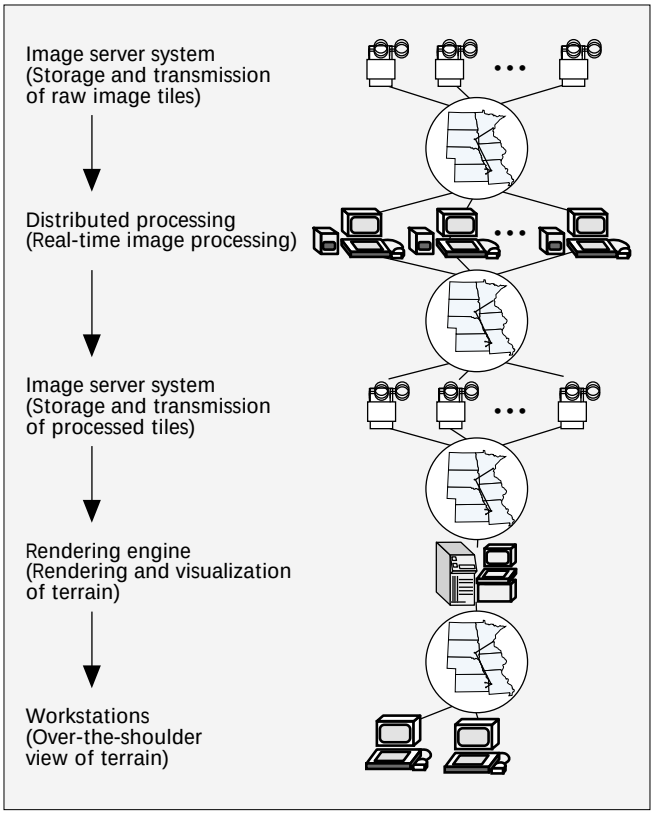

Figure 1. *Planned functionality of the MAGIC testbed.*

- University of Kansas (KU)[1]
- U S WEST Communications, Inc.

Other MAGIC participants that contributed equipment, facilities, and/or personnel to the effort were:

- Army High-Performance Computing Research Center (AHPCRC)
- Battle Command Battle Laboratory, U.S. Army Combined Arms Command (BCBL)
- Digital Equipment Corporation (DEC)
- Nortel, Inc./Bell Northern Research
- Southwestern Bell Telephone
- Splitrock Telecom

This article presents an overview of the MAGIC project with emphasis on the challenges associated with implementing a complex distributed system. Companion articles [2, 3] focus on a LAN/WAN gateway and a performance analysis tool that were developed for the MAGIC testbed. The article is organized as follows. The following section briefly describes the three major testbed components: the internetwork, the image server system, and the application. The third section discusses some of the system-level considerations that were addressed in designing these components, and the fourth section presents some high-level performance measurements. The fifth (affectionately entitled "Herding Cats") and sixth sections describe how this multi-organizational collaborative project was coordinated, and the technical and managerial lessons learned. Finally, the last section provides a brief summary of MAGIC-II, a follow-on project begun in early 1996.

Overview of the MAGIC Testbed

One of the primary goals of the MAGIC project was to create a testbed to demonstrate advanced capabilities that would not be possible without a very high-speed internetwork. MAGIC accomplished this goal by implementing an interactive terrain visualization application, TerraVision, that relies on a distributed image server system (ISS) to provide it with massive amounts of data in real time. The planned functionality of the MAGIC testbed is depicted in Fig. 1. Currently, TerraVision uses data processed off-line and stored on the ISS. In the future the application will be redesigned to enable real-time image processing as well as real-time terrain visualization (see the last section). Note that the workstations which house the application, the servers of the ISS, and the "over-the-shoulder" tool (see subsection entitled "The Terrain Visualization Application"), as well as those that will perform the on-line image processing, can reside anywhere on the network.

The MAGIC Internetwork

The MAGIC internetwork, depicted in Fig. 2, includes six high-speed local area networks (LANs) interconnected by a wide area network (WAN) backbone. The backbone, which spans a distance of approximately 600 miles, is based on synchronous optical network (SONET) technology and provides OC-48 (2.4 Gb/s) trunks, and OC-3 (155 Mb/s) and OC-12 (622 Mb/s) access ports. The LANs are based on asynchronous transfer mode (ATM) technology. Five of the LANs — those at BCBL in Fort Leavenworth, Kansas, EDC in Sioux Falls, South Dakota, MSCI in Minneapolis, Minnesota, Sprint in Overland Park, Kansas, and U S WEST in Minneapolis, Minnesota — use FORE Systems models ASX-100 and ASX-200 switches with OC-3c and 100 Mb/s TAXI interfaces. The ATM LAN at KU in Lawrence, Kansas, uses a DEC AN2 switch, a precursor to the DEC GigaSwitch/ATM, with OC-3c interfaces. The network uses permanent virtual circuits (PVCs) as well as switched virtual circuits (SVCs) based on both SPANS, a FORE Systems signaling protocol, and the ATM Forum User-Network Interface (UNI) 3.0 Q.2931 signaling stan-

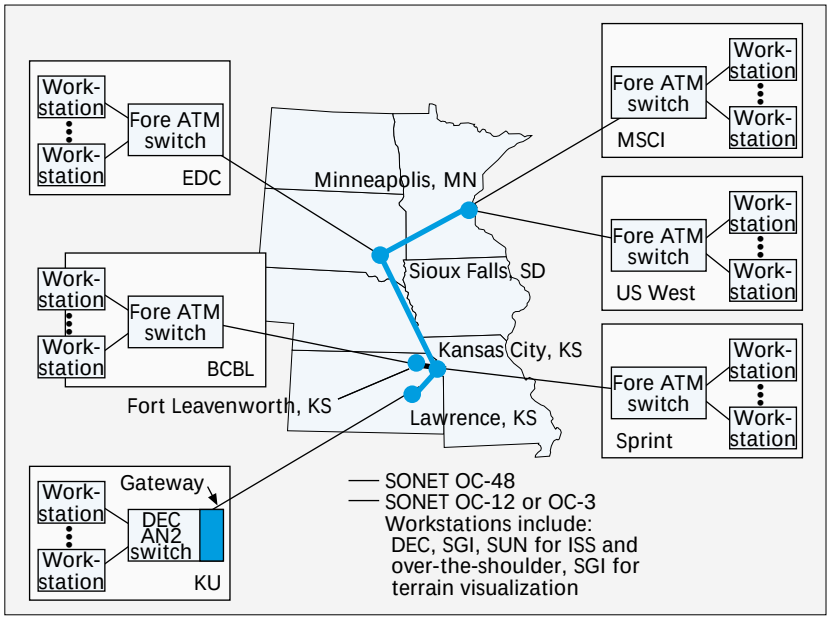

Figure 2. *Configuration of the MAGIC ATM internetwork.*

dard. The workstations at the MAGIC sites include models from DEC, SGI, and Sun. As part of MAGIC, an AN2/SONET gateway with an OC-12c interface was developed to link the AN2 LAN at KU to the MAGIC backbone [2].

In addition to implementing the internetwork, a variety of advanced networking technologies were developed and studied under MAGIC. A high-performance parallel interface (HIPPI)/ATM gateway was developed to interface an existing HIPPI network at MSCI to the MAGIC backbone. The gateway is an IP router rather than a network-layer device such as a broadband integrated services digital network (B-ISDN) terminal adapter, and was implemented in software on a high-performance workstation (an SGI Challenge). This architecture provides a programmable platform that can be modified for network research, and in the future can readily take advantage of more powerful workstation hardware. In addition, the platform is general-purpose; that is, it is capable of supporting multiple HIPPI interfaces as well as other interfaces such as fiber distributed data interface (FDDI).

Software was developed to enable UNIX hosts to communicate using Internet Protocol (IP) over an ATM network. This IP/ATM software currently runs on SPARCstations under Sun OS 4.1 and includes a device driver for the FORE SBA series of ATM adapters. It supports PVCs, SPANS, and UNI 3.0 signaling, as well as the "classical" IP and Address Resolution Protocol (ARP) over ATM model [4]. The software should be extensible to other UNIX operating systems, ATM interfaces, and IP/ATM address-resolution and routing strategies, and will facilitate research on issues associated with the integration of ATM networks into IP internets.

In order to enhance network throughput, flow-control schemes were evaluated and applied, and IP/ATM host parameters were tuned. Experiments showed that throughput close to the maximum theoretically possible could be attained on OC-3 links over long distances. To achieve high throughput, both the maximum transmission unit (MTU) and the Transmission Control Protocol (TCP) window must be large, and flow control must be used to ensure fairness and to avoid cell loss if there are interacting traffic patterns [5, 6].

The Terrain Visualization Application

TerraVision allows a user to view and navigate through (i.e., "fly over") a representation of a landscape created from aerial or satellite imagery [7]. The data used by TerraVision are derived from raw imagery and elevation information which have been preprocessed by a companion application known as TerraForm. TerraVision requires very large amounts of data in real time, transferred at both very bursty and high steady rates. Steady traffic occurs when a user moves smoothly through the terrain, whereas bursty traffic occurs when the user jumps ("teleports") to a new position. TerraVision is designed to use imagery data that are located remotely and supplied to the application as needed by means of a high-speed network. This design enables TerraVision to provide high-quality, interactive visualization of very large data sets in real time. TerraVision is of direct interest to a variety of organizations, including the Department of Defense. For example, the ability of a military officer to see a battlefield and to share a common view with others can be very effective for command and control.

Terrain visualization with TerraVision involves two activities: generating the digital data set required by the application, and rendering the image. MAGIC's approach to accomplishing these activities is described below. Enhancements to the application that provide additional features and capabilities are also described.

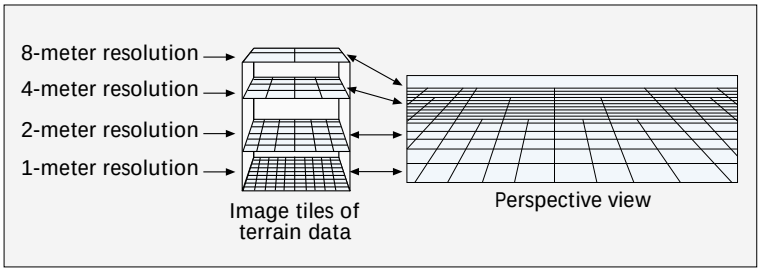

Figure 3. *Relationship between tile resolutions and perspective view. (Source: SRI International)*

Data Preparation — In order to render an image, TerraVision requires a digital description of the shape and appearance of the subject terrain. The shape of the terrain is represented by a two-dimensional grid of elevation values known as a *digital elevation model* (DEM). The appearance of the terrain is represented by a set of aerial images, known as *orthographic projection images* (ortho-images), that have been specially processed (i.e., ortho-rectified) to eliminate the effects of perspective distortion, and are in precise alignment with the DEM. To facilitate processing, distributed storage, and high-speed retrieval over a network, the DEM and images are divided into small fixed-size units known as *tiles*.

Low-resolution tiles are required for terrain that is distant from the viewpoint, whereas high-resolution tiles are required for close-in terrain. In addition, multiple resolutions are required to achieve perspective. These requirements are addressed by preparing a hierarchy of increasingly lower-resolution representations of the DEM and ortho-image tiles in which each level is at half the resolution of the previous level. The tiled, multiresolution hierarchy and the use of multiple resolutions to achieve perspective are shown in Fig. 3.

Rendering of the terrain on the screen is accomplished by combining the DEM and ortho-image tiles for the selected area at the appropriate resolution. As the user travels over the terrain, the DEM tiles and their corresponding ortho-image tiles are projected onto the screen using a perspective transform whose parameters are determined by factors such as the user's viewpoint and field of view. The mapping of a transformed ortho-image to its DEM and the rendering of that image are shown in Fig. 4.

The data set currently used in MAGIC covers a 1200 km^2 exercise area of the National Training Center at Fort Irwin, California, and is about 1 Gpixel in size. It is derived from aerial photographs obtained from the National Aerial Photography Program archives and DEM data obtained from the U.S. Geological Survey. The images are at approximately 1 m resolution (i.e., the spacing between pixels in the image corresponds to 1 m on the ground). The DEM data are at approximately 30 m resolution (i.e., elevation values in meters are at 30 m intervals).

Software for producing the ortho-images and creating the multiresolution hierarchy of DEM and ortho-image tiles was developed as part of the MAGIC effort. These processes were performed "off-line" on a Thinking Machines Corporation Connection Machine (CM-5) supercomputer owned by the AHPCRC and located at MSCI. The tiles were then stored on the distributed servers of the ISS and used by terrain visualization software residing on rendering engines at several locations.

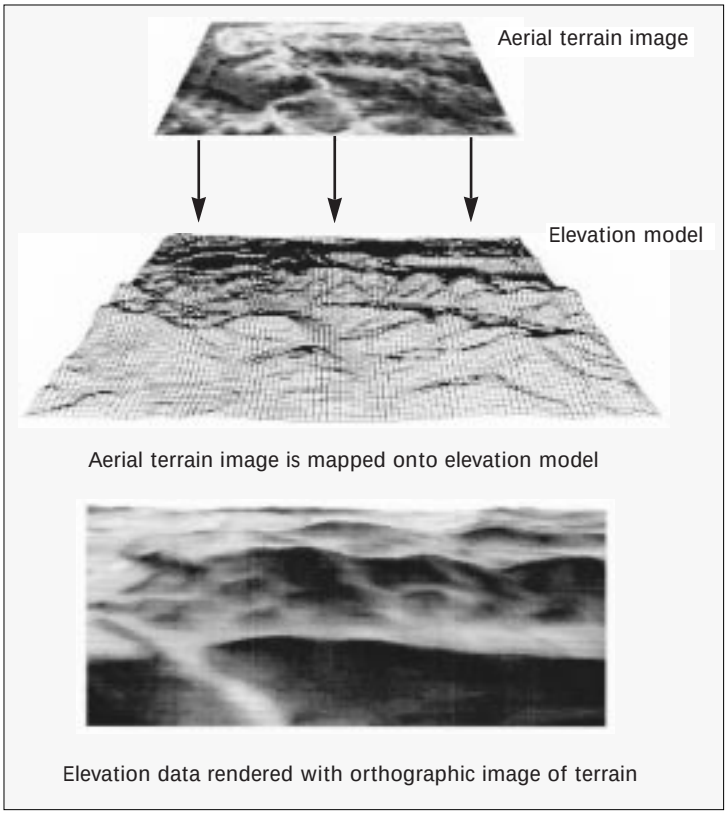

Figure 4. *Mapping an ortho-image onto its digital elevation model. (Source: SRI International)*

Image Rendering — TerraVision provides for two modes of visualization: two-dimensional (2-D) and three-dimensional (3-D). The 2-D mode allows the user to fly over the terrain, looking only straight down. The user controls the view by means of a 2-D input device such as a mouse. Since virtually no processing is required, the speed at which images are generated is limited by the throughput of the system comprising the ISS, the network, and the rendering engine.

In the 3-D mode, the user controls the visualization by means of an input device that allows six degrees of freedom in movement. The 3-D mode is computationally intensive, and satisfactory visualization requires both high frame rates (i.e., 15–30 frames/s) and low latencies (i.e., no more than 0.1 s between the time the user moves an input device and the time the new frame appears on the screen).

High frame rates are achieved by using a local very-high-speed rendering engine, an SGI Onyx, with a cache of tiles covering not only the area currently visible to the user, but also adjacent areas that are likely to be visible in the near future. A high-speed search algorithm is used to identify the tiles required to render a given view. For example, as noted above, perspective (i.e., 3-D) views require higher-resolution tiles in the foreground and lower-resolution tiles in the background. TerraVision requests the tiles from the ISS, places them in memory, and renders the view. Latency is minimized by separating image rendering from data input/output (I/O) so that the two activities can proceed simultaneously rather than sequentially (see the section entitled "Design Considerations").

Additional Features and Capabilities — TerraVision includes two additional features: superposition of fixed and mobile objects on the terrain, and registration of the user's viewpoint to a map. Both of these features are made possible by precisely aligning the DEM and imagery data with a world coordinate system as well as with each other.

A number of buildings and vehicles have been created and stored on the rendering engine for display as an overlay on the terrain. The locations of vehicles can be updated periodically by transferring vehicle location data, acquired with a global positioning system receiver, to the rendering engine for integration into the terrain visualization displays. Registration of the user's viewpoint to a map enables the user to specify the area he wishes to explore by pointing to it, and it aids the user in orienting himself.

In addition, an over-the-shoulder (OTS) tool was developed to allow a user at a remote workstation to view the terrain as it is rendered. The OTS tool is based on a client/server design and uses XWindow system calls. The user can view the entire image on the SGI screen at low resolution, and can also select a portion of the screen to view at higher resolution. The frame rate varies with the size and resolution of the viewed image, and with the throughput of the workstation.

The Image Server System

The ISS stores, organizes, and retrieves the processed imagery and elevation data required by TerraVision for interactive rendering of the terrain. The ISS consists of multiple coordinated workstation-based data servers that operate in parallel and are designed to be distributed around a WAN. This architecture compensates for the performance limitations of current disk technology. A single disk can deliver data at a rate that is about an order of magnitude slower than that needed to support a high-performance application such as TerraVision. By using multiple workstations with multiple disks and a high-speed network, the ISS can deliver data at an aggregate rate sufficient to enable real-time rendering of the terrain. In addition, this architecture permits location-independent access to databases, allows for system scalability, and is low in cost. Although redundant arrays of inexpensive disks (RAID) systems can deliver higher throughput than traditional disks, unlike the ISS they are implemented in hardware and, as such, do not support multiple data layout strategies; furthermore, they are relatively expensive. Such systems are therefore not appropriate for distributed environments with numerous data repositories serving a variety of applications.

The ISS, as currently used in MAGIC, comprises four or five UNIX workstations (including Sun SPARCstations, DEC Alphas, and SGI Indigos), each with four to six fast SCSI disks on two to three SCSI host adapters. Each server is also equipped with either a SONET or a TAXI network interface. The servers, operating in parallel, access the tiles and send them over the network, which delivers the aggregate stream to the host. This process is illustrated in Fig. 5. More details about the design and operation of the ISS can be found in [8].

Design Considerations

In MAGIC, the single most perspicuous criterion of successful operation is that the end user observes satisfactory performance of the interactive TerraVision application. When the user flies over the terrain, the displayed scene must flow smoothly, and when he teleports to an entirely

different location, the new scene must appear promptly. Obtaining such performance might be relatively straightforward if the terrain data were collocated with the rendering engine. However, one of the original premises underlying the MAGIC project is that the data set and the application are not collocated. There are several reasons for this, the most important being that the data set could be extremely large, so it might not be feasible to transfer it to the user's site. Moreover, experience has shown that in many cases the "owner" of a data set is also its "curator" and may be reluctant to distribute it, preferring instead to keep the data locally to simplify maintenance and updates. Finally, it was anticipated that future versions of the application might work with a mobile user and with fused data from multiple sources, and neither of these capabilities would be practical with local data. Therefore, since the data will not be local, the MAGIC components must be designed to compensate for possible delays and other degradations in the end-to-end operation of the system.

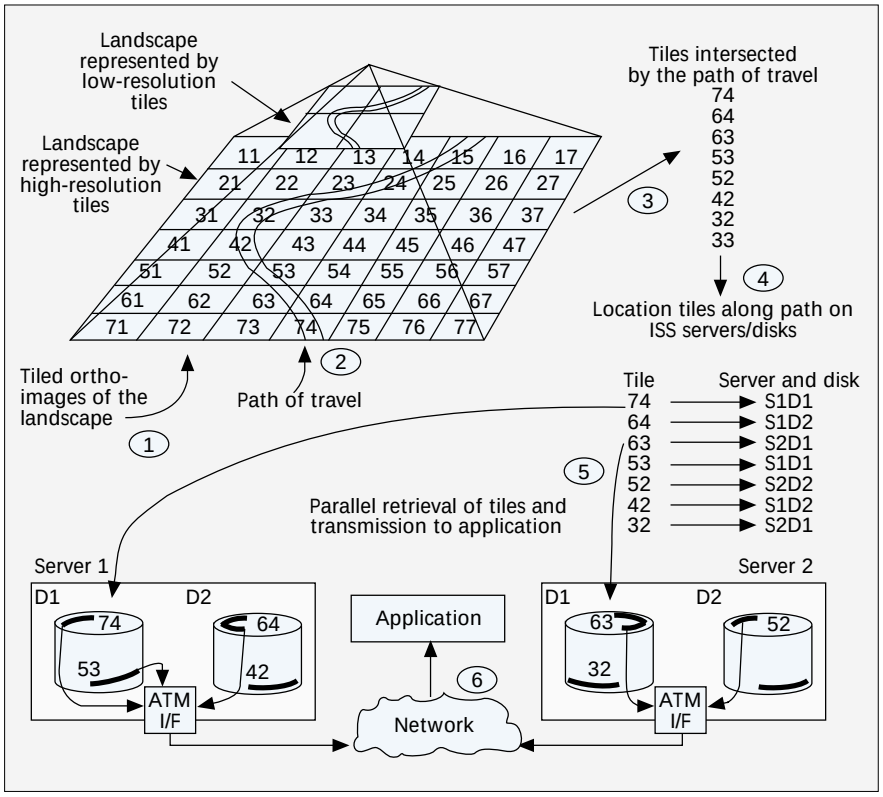

Figure 5. *Schematic representation of the operation of the ISS. (Source: Lawrence Berkeley National Laboratory)*

In order to understand system-level design issues, it is necessary to outline the sequence of events that occurs when the user moves the input device, causing a new scene to be generated. TerraVision first produces a list of new tiles required for the scene. This list is sent to an ISS master, which performs a name translation, mapping the logical address of each tile (the tile identifier) to its physical address (server/disk/location on disk). The master then sends each server an ordered list of the tiles it must retrieve. The server discards the previous list (even if it has not retrieved all the tiles on that list) and begins retrieving the tiles on the new list. Thus, the design for the system comprising TerraVision, the ISS, and the internetwork must address the following questions:

- How can TerraVision compensate for tiles it needs for the next image but have not yet been received?
- How often should TerraVision request tiles from the ISS?
- Where should the ISS master be located?
- How should tiles be distributed among the ISS disks?
- How can cell loss be minimized near the rendering site where the tile traffic becomes aggregated and congestion may occur?

Missing Tiles

Network congestion, an overload at an ISS server, or a component failure could result in the late arrival or loss of tiles that are requested by the application. Several mechanisms were implemented to deal with this problem. First, although the entire set of high-resolution tiles cannot be collocated with the application, it is certainly feasible to store a complete set of lower-resolution tiles. For example, if the entire data set comprises 1 Tbyte of high-resolution tiles, then all of the tiles that are five or more levels coarser would occupy less than 1.5 Mbyte, a readily affordable amount of local storage. If a tile with resolution at, say, level 3 is requested but not delivered in time for the image to be rendered, then, until the missing level-3 tile arrives, the locally available coarser tile from level 5 would be used in place of the 16 level-3 tiles. This substitution manifests itself by the affected portion of the rendered image appearing "fuzzy" for a brief period of time. Temporary substitution of low-resolution tiles for high-resolution tiles is particularly effective for teleporting because that operation requires a large number of new tiles, so it is more likely that one or more will be delayed.

Second, TerraVision attempts to predict the path the user will follow, requesting tiles that *might* soon be needed, and assigning one of three levels of priority to each tile requested. Priority-1 tiles are needed as soon as possible; the ISS retrieves and dispatches these first. This set of tiles is ordered by TerraVision, with the coarsest assigned the highest priority within the set. The reasons are:

- The rendering algorithm needs the coarse tiles before it needs the next-higher-resolution tiles.
- There are fewer tiles at the coarser resolutions, so it is less likely that they will be delayed.

The priority-2 tiles are those that the ISS should retrieve but should transmit only if there are no priority-1 tiles to be transmitted; that is, priority-2 tiles are put on a lower-priority transmit queue in the I/O buffer of each ISS server. (ATM switches would be allowed to drop the cells carrying these tiles.) Priority-3 tiles are those that should be retrieved and cached at the ISS server; these tiles are less likely to be needed by TerraVision. Note that there is a trade-off between "overpredicting" — requesting too many tiles — which would result in poor ISS performance and high network load, and "underpredicting," which would result in poor application performance.

Finally, a tile will continue to be included in TerraVision's request list if it is still needed and has not yet been delivered. Thus, tiles or tile requests that are dropped or otherwise "lost" in the network will likely be delivered in response to a subsequent request from the application.

DOCKET ALARM

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS
Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS
Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS
Sync your system to PACER to automate legal marketing.