Performance of a Multiprocessor Multidisk CD-ROM Image Server

Rolf Muralt, Benoit A. Gennart, Bernard Krummenacher, Roger D. Hersch
Ecole Polytechnique Fédérale de Lausanne (EPFL), Switzerland

Abstract

Professionals in various fields such as medical imaging, biology, and civil engineering require rapid access to huge amounts of image data. Multimedia interfaces further increase the demand for high-performance image and media servers. We consider a parallel image server architecture that relies on arrays of intelligent disk nodes, each disk node consisting of one processor and one or more disks. The multiprocessor multidisk design is applied in an original way to provide both high-performance real-time access and unlimited, inexpensive mass storage capacities through the use of removable CD-ROM medium. Reading uncompressed or lossless compressed image data striped across multiple CD-ROM disks yields sustained performance of up to 2.5 Mbytes/s. Unlike RAID storage devices, the proposed CD-ROM architecture offers information distribution control and local processing capabilities. Two application fields that benefit from such features are presented.

Keywords: CD-ROM; disk arrays; multiprocessing; parallel file system.

1 Introduction

In the fields of scientific modeling, medical imaging, biology, civil engineering, cartography, and graphic arts, there is an urgent need for huge storage capacities, fast access, and real-time interactive visualization of pixmap images. While processing power and memory capacity double every two years, data access bandwidth—both disk and memory—increases at a much slower rate. Parallel input/output devices are required in order to access and manipulate image data at high speed.

A high-performance high-capacity image server must provide users located on local or public networks with a set of adequate services for immediate access to images stored on disk arrays. Basic services include real-time extraction of image parts, zooming in and out, browsing through 3-D image cuts, and accessing image sequences at the required resolution and speed. The RAID [1] concept focuses on increasing transfer rates between CPU and high-bandwidth disk arrays by parallelizing access to disk blocks. However, block and file management continue to be handled by a single CPU with limited processing power and memory bandwidth.

The multiprocessor multidisk (MPMD) approach aims at associating disks and processors so as to form an array of intelligent disk nodes capable of applying parallel local preprocessing operations. Previous research has led to the implementation of such an MPMD image server, called the *GigaView*, that relied on an array of winchester disks [2].

While offering high performance levels, hard-disk based storage servers are hindered by limited, and comparatively expensive non-removable mass storage capacities in the form of fixed-disk arrays. Over the past years, CD-ROM disks have emerged as the leading technology when it comes to settling for a low-cost, high-capacity removable medium. Thanks to readily available recording hardware, writing CD-ROM disks has become remarkably affordable. Such

In Proc. Int. Conf. Parallel and Distributed Processing Techniques and Applications, Sunnyvale, USA, Aug. 9-11, 1996, 902-913

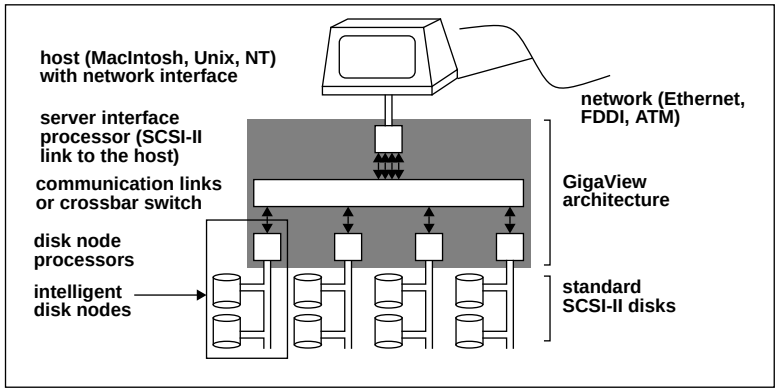

Figure 1: *GigaView 8-disk architecture.*

master disks may further be applied for industrial mass-production of low-cost CD-ROM distributions [3]. However, real-time access performance remains rather modest—in terms of high random-access times (around 220 ms) and low transfer rates (612 Kbytes/s at quad-speed)—when compared to state-of-the-art winchester disks.

The originality of this contribution consists in integrating run-of-the-mill CD-ROM hardware into a multiprocessor multidisk design, in an attempt to achieve a high-performance architecture that makes up for the low performance at the drive level by parallelizing accesses to CD-ROM disks.

This contribution focuses on the design of the *GigaView CD* image server: the hardware architecture, the multidimensional file system (MDFS), and issues related to the underlying CD-ROM technology. Single-request behavior and parallelization efficiency are analyzed to identify the GigaView CD parallel architecture's performances and bottlenecks. At this time, we consider two application fields for the CD-ROM image server: geographical information systems, and medical imaging.

Section 2 describes the hardware architecture and the multidimensional file system (MDFS). In Sec. 3, we evaluate the performance of the GigaView CD parallel image server. Typical fields of application are reviewed in Sec. 4.

2 The GigaView CD architecture

In this section, relevant hardware and software topics related to the GigaView CD are reviewed. The multiprocessor multidisk architecture is presented in Sec. 2.1. The Multidimensional File System is introduced in Sec. 2.2, and the integration of CD-ROM technology into the parallel architecture is dealt with in Sec. 2.3.

2.1 Hardware Architecture

The parallel image server consists of a server interface processor connected through communication links to an array of intelligent disk nodes (Fig. 1). The server interface processor provides the network interface, namely SCSI-2, ATM, or FDDI. Each disk node consists of one or more standard disks connected through a SCSI-2 bus to a local processing unit.

The local processors are T800 transputers, and provide both processing power and four communication links. Data transfers through the links and data processing do not interfere, since data packets sent over the links are written by DMA (direct memory access) into the processor's memory. The T800 transputers operate at a 20-MHz clock frequency and are rated at 10 MIPS by Inmos [4]. Communication link bandwidth is 1.6 Mbytes/s over each of the four DMA channels, and local memory bandwidth is 18 Mbytes/s, and the units lack on-chip

cache memory. Since the transputer handles context switches in hardware, they do not add noticeable overhead to the main computations.

The disk nodes support processes for disk access, extent caching, image part extraction, and image (de)compression. Disk-node processors can use their processing power to extract image windows while at the same time, thanks to direct memory accesses, executing SCSI transfers and communicating with the server interface processor (Fig. 1). Consequently, when disk accesses are combined with local processing operations, parallelization becomes very efficient and high speedups can be achieved.

The design of the proposed MPMD architecture is highly scalable since it can be extended both at the processor and at the disk level to accommodate specific performance and capacity requirements (Fig. 1). Additional disk nodes may be connected to the server interface processor (SIP); scaling beyond four disk nodes further requires a crossbar switch to share access to the four SIP communication links. At the disk level, up to seven standard SCSI-2 disks may physically be connected to a single disk node processor using its local SCSI-2 bus interface. There is however a practical upper bound to the number of disks that can effectively be accommodated by individual disk nodes with respect to the local T800 processing power and of the 1.6-Mbytes/s link communication throughput between disk-node and SIP. Therefore, effectively balancing the architecture is a key design target, and this issue is further addressed in Sec. 3.

Apart from performance concerns, the physical layout of the SCSI-2 disks—whether they be added by means of independent disk nodes or by chaining them to a unique disk node—is irrelevant and cannot be distinguished from a functional point of view. Furthermore, we introduce at the file system level the higher notion of *disk pool* which consists of a subset of the disks in the cluster. Several disk pools may coexist as long as a given disk is part of at most one pool at a time.

2.2 Multidimensional File System

Single image access characteristics are known: client workstations require rectangular portions—referred to as visualization windows—of large pixmap image files. In order to access disks in parallel, images are partitioned into rectangular *extents* (Fig. 2). The Multidimensional File System (MDFS) stores 1-dimensional (1-D), 2-D, and 3-D images divided into 1-D, 2-D, and 3-D extents respectively, and provides excellent access performance, regardless of the image file size and hardware architecture.

The server interface processor runs the image server master process receiving client access requests, and issuing access calls to the parallel image file server. The parallel file server includes a file system master process responsible for maintaining overall parallel file system coherence (directories, file index tables, file extent allocation tables), and extent serving processes running on local disk-node processing units. Extent serving processes are responsible for serving extent access requests, maintaining the disks' free-block lists, and managing local extent caches. Local image processing tasks required for image presentation such as zooming are located on disk-node processing units.

Access performance depends on the way extents are distributed onto the disk array. Extents are mapped sequentially on the k disks in the pool (modulo k). In order to achieve a uniform load on all disk-nodes, one needs to determine the most suitable extent row offset, that is the difference in disk index between two extents of the same image extent column. A suitable extent row offset ensures that extents covered by the same visualization window are distributed uniformly on the disk pool so as to achieve high parallelization efficiency. For 2 and 3 disk nodes (resp. 4 and 5, 6, 7 and 8), the most suitable row offset is 1 (resp. 3, 1, 3).

Lossless compression algorithms tuned to the display of scanned maps stored on the disks are integrated into the file system. The compression algorithms are variations of the run-length coding scheme [5] and are optimized to provide high-speed software decompression, at the expense of compression efficiency. The MDFS file system provides zooming facilities

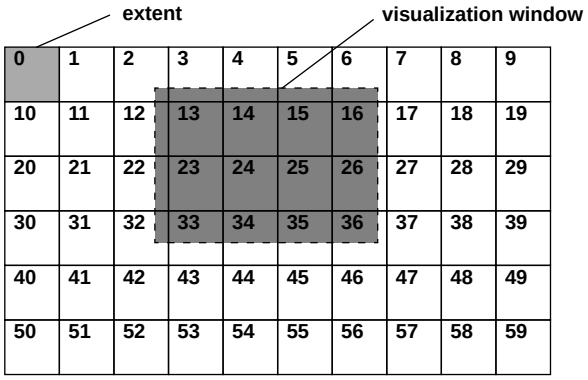

Figure 2: *Division of an image into extents.*

that support the reduction of large images down to the visualization window size. Both compression and zooming operations are processed in parallel on local disk nodes to ensure high parallelization efficiency.

A typical MDFS operation consists of reading a visualization window obtained by assembling several extents (Fig. 2). The *data access pipeline*, i.e. the path from the (un)compressed data stored on disk to a visualization window on the GigaView interface processor consists of four steps: (1) reading the required (un)compressed image extents from their respective disks to their disk-node processor's local memory; (2) if necessary, decompressing the extents on the disk-node processors ; (3) transferring the uncompressed extents from the disk nodes to the server interface processor by means of DMA communication links; (4) and merging the uncompressed extents into the visualization window buffer. In the case of extents retrieved from the same disk, the four steps are pipelined.

2.3 CD-ROM Integration

The integration of CD-ROM technology is motivated by the challenge to combine high-performance real-time image access with the use of removable, inexpensive mass storage medium. In terms of performance, we expect parallelization efficiency to be higher with CD-ROM pools than with hard-disk arrays, enabling the architecture to be extended beyond four disks.

Accessing data from CD-ROM disk pools. As illustrated in Fig. 1, mass storage devices are attached to local disk-node processors by SCSI-2 interfaces. Consequently, hardware integration of standard SCSI-compliant CD-ROM drives is a simple matter of connecting them to available disk nodes. The present study is based on using NEC *MultiSpin 4Xe* CD-ROM hardware, rated at a peak throughput of 612 Kbytes/s and an average access time of 220 ms (combined seek time and rotational delay).

The MDFS file system achieves high performance levels by placing data on disks as contiguous rectangular extents, thereby minimizing costly disk-level access times. Clearly, such a data allocation scheme is highly desirable in the context of striping image data across several CD-ROM disks. The formats written on CD-ROM disks however require special interfacing considerations.

The Yellow Book standardization by Philips and Sony extends the original Red Book definition to CD-ROM disks [6]. Each block is composed of 98 audio frames, or 2352 bytes. In CD-ROM Data Mode 1 format, every block further consists of 12 synchronization bytes for block identification, 4 bytes for block addressing, 2048 bytes for actual user data, 4 bytes for error detection, 8 empty (zero) bytes, and 276 bytes for layered error correction codes. Through SCSI-2 normalization [7], CD-ROM Data Mode 1 disks appear as a logical sequence

of 333,225 randomly-addressable read-only 2048-byte blocks.

However, the MDFS file system organization relies on accessing data formatted as 512-byte blocks. It therefore becomes necessary to introduce low-level mapping capabilities between logical 512-byte blocks and actual physical blocks, granting efficient medium-independent access. Whereas [7] suggests that logical-block mapping capabilities be provided by CD-ROM hardware, we integrate block translation at the SCSI-2 driver level in the MDFS file system, to prevent any hardware dependences. The enhanced SCSI-2 driver automatically recognizes the physical characteristics of attached storage devices upon startup, making it possible to access both winchester and CD-ROM disk pools at a time. In fact, the design is flexible enough to accomodate combined winchester and CD-ROM disks as part of the same pool. Low-level evaluations have shown that access performance is improved by reducing the number of individual SCSI-2 requests. High disk-access efficiency is ensured by the fact that whole extents are read issuing a single SCSI-2 command.

An essential feature of the GigaView CD file system is the ability to read data from a CD-ROM pool that comprises any number of disks less or equal to the actual number of physical CD-ROM drives in the array. This enables to share disk pools across a wide range of GigaView CD architectures, and to determine CD-ROM pool size upon specific performance and storage capacity requirements.

Creating CD-ROM disk pools. In the case of an array of fixed winchester disks, images are striped among the array of disks at writing time, and assembled from the same array at reading time. In the case of removable CD-ROM disk pools, the question of concern is to determine the number of disks to stripe data across. On the one hand, storage capacity requirements impose a minimal CD-ROM pool size, based on the total amount of image data to be stored (up to 650 Mbytes on a CD-ROM disk). On the other hand, applications may require that specific read-access performance levels be attained, making it necessary to stripe image data across more CD-ROM disks. Section 3 reports the way performance increases with the number of disks in a CD-ROM pool.

The proposed CD-ROM disk-pool mastering procedure is straightforward and flexible. Basically, the idea consists in writing an original set of the image data to be stored on CD-ROM onto a hard-disk based GigaView system. The master hard-disk pool must be formatted to feature the same number of disks as the planned CD-ROM pool. The physical data blocks of the master disks are then mapped into local *mirror* files on the station connected to the CD-R recording unit. Applying appropriate CD-ROM mastering software, the pool of CD-ROM disks is finally recorded in CD-ROM Data Mode 1 format, based on the former mirror files. Since the information is written as CD-ROM Data Mode 1 blocks, the resulting GigaView CD disk pool is not compatible with the ISO 9660 normalized file system [8].

3 Performance Analysis

This section analyzes the single-request performance behavior of the multiprocessor multidisk CD-ROM image server. Section 3.1 introduces a model that describes single-request performance in terms of latency and throughput. In Sec. 3.2, parallelization efficiency is optimized by analyzing throughput and disk utilization.

3.1 Single-request Behavior

For the GigaView CD performance measurements, it is assumed that transferring a visualization window from disks to host is a two-stage pipeline. The first stage of the pipeline transfers extents from the CD-ROM disks to the server interface processor (SIP) memory. The second stage transfers window segments consisting of rows of extents from the SIP to the host memory.

We plot the delay of the parallel CD-ROM image server at increasing visualization window sizes. Extent caching performed by the disk nodes is undesirable in this context, and the caches

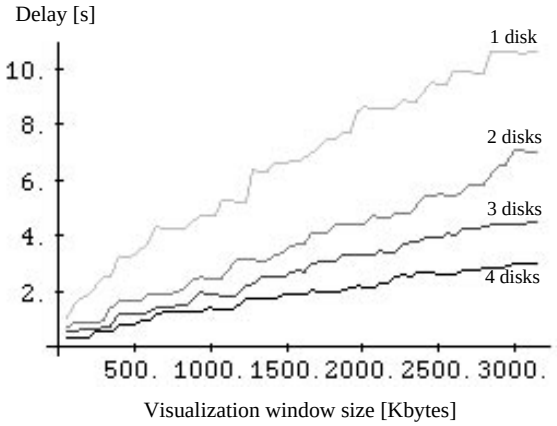

Figure 3: *GigaView CD single-request delay as a function of the number of CD-ROM drives (single CD-ROM drive per disk node; uncompressed data).*

are therefore flushed between window accesses. Furthermore, access times are sensitive to the position of the visualization window. To take this effect into account, the visualization window is accessed four times for each size at random positions. The resulting delays are averaged to obtain the access time for a given visualization window size.

Experimental measurements performed on various MPMD architectures reveal that access times increase linearly with the size of the requested visualization window. Therefore, the performance behavior of the parallel storage server may be described using two numbers, latency and throughput. This is similar to the way secondary storage devices are described by seek time and throughput. The approach is to measure the delay of the parallel storage server for increasing visualization window sizes, to linearize the access time, and get a formula of the type :

$$AccessTime = Latency + \frac{RequestSize}{Throughput} \quad (1)$$

The linearization approach has proved particularly effective, regardless of the data allocation scheme and the architecture of the system.

The GigaView CD architecture performance is sensitive to the extent allocation scheme. In particular, the extent size and the row offset have to be chosen carefully to achieve the highest performance. Oversized extents waste disk and processor bandwidth, as large amounts of image data lying outside the visualization window have to be processed. Furthermore, parallelization can only be efficient if visualization windows consist of at least as many extents as there are disks in the pool. Changing the data allocation scheme to improve parallelization efficiency by radically decreasing extent size does not improve performance either: the overhead due to the larger number of extents negates the effect of the improved data allocation. In practice, extent sizes of about 64 Kbytes (e.g. 256-by-256 1-byte pixels) maximize sustained throughput, in the case of both uncompressed and compressed data stored on CD-ROM disk pools.

Figure 3 shows that latency decreases and throughput increases as the number of CD-ROM disk nodes increases. Latency is closely related to the delay required to access a single extent from a CD-ROM disk pool. Whether extents are stored in uncompressed or compressed mode does not affect latency. Latency decreases when striping image data across larger disk pools, as less extents are stored on each disk, thereby improving extent access locality. Latency decreases from 348 ms, in the case of a single disk, to 214 ms, in the case of a four-disk CD-ROM pool.

3.2 Parallelization Efficiency

The objective is to optimize parallelization efficiency and to identify balanced architectures by analyzing throughput and disk utilization when increasing the number of CD-ROM units in the cluster.

Throughput. The most restrictive stage in the data access pipeline, in terms of throughput, defines the bottleneck of the entire architecture. Based on measurements [2], it is assumed that the SIP is capable of handling up to 5 Mbytes/s. At the disk-node level, pipeline efficiency may be limited by either CD-ROM disk access, local processing, or link communication. Furthermore, pipeline *stalls* may occur due to memory buffer allocation contention, when a limited number of on-board memory buffers have to be shared among multiple extent serving processes.

Limitations in the number of available memory buffers introduce pipeline interstage dependences, in the sense that extent processing may stall, waiting for the preceding stage to complete and release memory buffers. Eventually, pipeline efficiency will degrade to the point where extents are processed sequentially.

Uncompressed data. Figure 4 illustrates the way throughput increases when accessing visualization windows inside uncompressed images.

When connecting up to four disk nodes consisting each of one local processor and a single CD-ROM drive, linearly-increasing performance confirms the assumption that SIP processing power is not a potential bottleneck. Sustained disk-level throughput appears to be about half the manufacturer-rated peak CD-ROM throughput (312 Kbytes/s per CD-ROM disk).

Throughput increases as multiple CD-ROM drives are connected to a single local processor. However, a single disk node is not able to sustain linear increases in performance (Fig. 4). To investigate this behavior, let us consider what components may become bottlenecks in the data access pipeline. Local SCSI-bus contention is not likely to be an issue at the considered throughput levels. Raw communication link bandwidth is rated at 1.6 Mbytes/s, and sustained link throughput has been measured at 1263 Kbytes/s, as a result of the message-passing overhead. Parallelization efficiency proves to be limited by processor memory buffer contention. In order to perform disk access and extent transfer concurrently, extent serving processes require the allocation of two extent buffers in memory: one for the extent being read from disk, and the other for the extent being transferred to the SIP. If memory is in short supply, the extent reading process will not be able to allocate memory until the previous extent has been completely transferred down the transputer link, thereby stalling the data access pipeline.

Compressed data. In the case of compressed images, performance will obviously depend on the actual image data accessed. Higher compression rates imply less data to be retrieved at the disk level, and less decompression processing at the disk-node level, resulting in improved overall performance. It is therefore desirable to evaluate performance in the following cases : highly-variable data making compression ineffective (compression rate 1.02), typical topographic map data (compression rate 5.08), and uniform data achieving the highest possible compression (compression rate 57.14).

Figure 5 presents the expected linear behavior as throughput increases in function of disk nodes consisting of one processor and a single CD-ROM drive. Interestingly, reading compressed images outperforms reading uncompressed images by about 10 % in the average case (topographic map). Therefore, data compression ought to be applied whenever single CD-ROM disks are connected to disk nodes, in order to achieve higher performance and to reduce storage capacity requirements. Incidentally, Fig. 5 confirms that the SIP is capable of handling throughputs of up to 5 Mbytes/s.

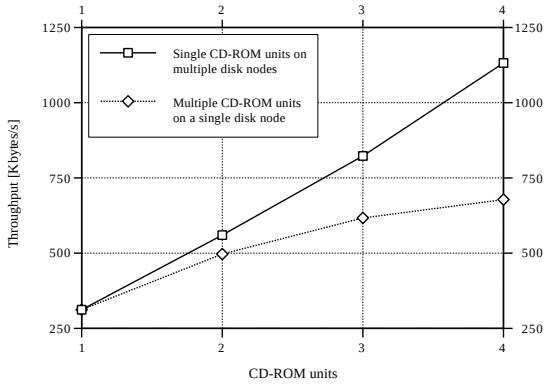

Figure 4: *Performance reading uncompressed image data.*

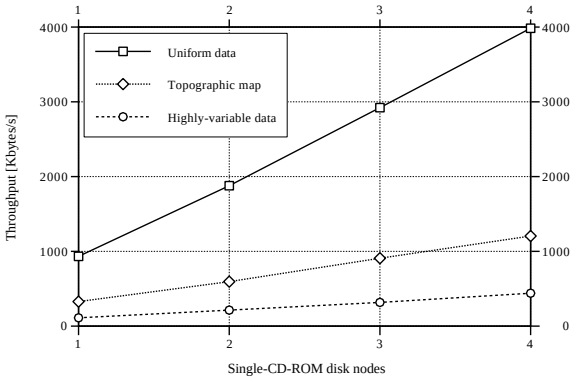

Figure 5: *Performance reading compressed image data. Each disk node consists of one processor and one CD-ROM drive.*

Figure 6 illustrates the way throughput evolves when multiple CD-ROM units are connected to a single disk node. Adding a fourth CD-ROM drive fails to significantly improve performance, pipeline efficiency being severely compromised by memory buffer contention. As a matter of fact, throughput *decreases* in all cases, except when reading highly-variable data. For reference, the local T800 transputers are capable of decompressing highly-variable data (respectively typical map data, and uniform data) at the sustained rate of 150 Kbytes/s (resp. 470 Kbytes/s, and 1.3 Mbytes/s). Accessing highly-variable data, local T800 computing power becomes the bottleneck in the data access pipeline, allowing for only one extent to be decompressed during a single extent disk access. In the case of reading topographic map data, disk-access and decompression times appear as quite effectively balanced, parallelization efficiency being solely limited by memory-bound pipeline stalls. When decompressing uniform data, extents expand from 1035 bytes in compressed state to 64 Kbytes, achieving high reading speed (single CD-ROM block) and decompression performance (T800 *memset* instruction). The throughput peaks at 1.3 Mbytes/s, as a result of limited communication link bandwidth. Performance decreases when adding a fourth CD-ROM unit due to memory buffer contention, stalling the data access pipeline between decompression processing and link DMA transmission.

Utilization. A key concept in identifying balanced architectures is that of CD-ROM disk *utilization*, defined as the ratio between sustained disk-level throughput and CD-ROM peak throughput (612 Kbytes/s).

The approach is to plot the CD-ROM disk utilization ratio for increasing zoom rates (Fig. 7). A zoom factor of n is achieved by selecting one in n^2 pixels in a decompressed pixmap image. As the zoom factor increases, the visualization window size is not changed, and consequently the amount of data fetched from the disks is increased. Sustained disk-level throughput increases together with zoom rates, due to higher data access locality when reading huge amounts of contiguous extents. Furthermore, the amount of processing required for each extent at the disk-node level decreases with higher zoom factors (one-in-n^2 pixel selection). The odd behavior between zoom rates 1 and 2 is caused by the transition from no local processing to one-in-four pixel selection. Processing the zooming operations at the disk-node level proves to be highly valuable in the case of reading visualization windows from huge scanned topographic maps. Typically, large bytemap images can be reduced to displayable size at near CD-ROM disk reading speed.

Layout	Zoom rate	Un-cmpr.	Compressed		
			Worst	Avg.	Best
4 / 2	1	994	281	810	2383
	2	226	70	147	271
	4	103	25	66	164
12 / 4	1	2459	578	1626	4214
	2	481	156	328	607
	4	289	57	164	383

Table 1: *Performance of balanced architectures at the application level [Kbytes/s].*

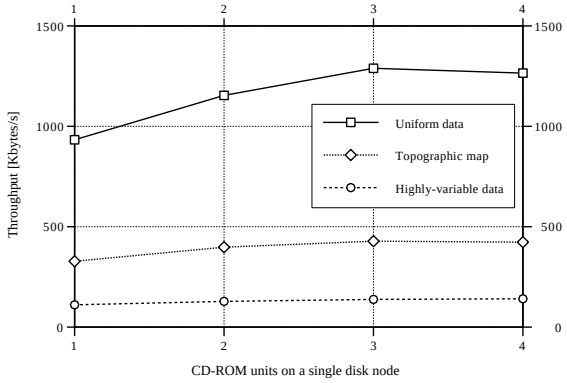

Figure 6: *Performance reading compressed image data. Multiple CD-ROM units connected to a single disk node.*

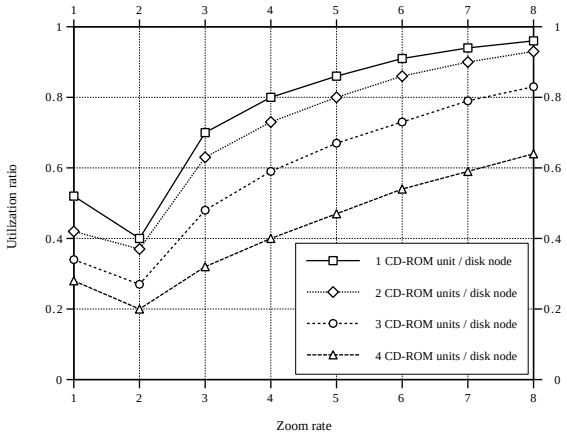

Figure 7: *CD-ROM disk utilization ratio.*

Balanced Architectures. The bottomline of the above analysis is that the performance of the GigaView CD increases linearly when adding up to four disk nodes to the SIP. Furthermore, the disk-node throughput does improve when connecting up to three CD-ROM drives per node. High parallelization efficiency can be achieved when zooming operations are applied at the disk-node level. Overall component utilization is maximized when compression is applied on an architecture featuring two CD-ROM units per disk node (Fig. 6). Adding a third CD-ROM drive yields higher performance, however at the expense of disk utilization. Due to memory buffer contention, connecting any further CD-ROM units proves to be inefficient.

A GigaView CD d/n architecture consists of d CD-ROM drives connected to n disk nodes (i.e. d/n disks per node). The following two balanced architectures will be considered hereafter (Table 1). The entry-level GigaView CD 4/2 consists of two disk nodes, each node being connected to two CD-ROM drives. This layout optimizes component utilization when data compression is applied, making effective use of the disk-pool storage capacity (2.6 Gbytes). The high-end GigaView CD 12/4 includes four disk nodes with three CD-ROM units each, in order to achieve high performance (up to 2.5 Mbytes/s at the application level) and huge disk-pool storage capacity (7.8 Gbytes in uncompressed mode).

As a guideline, up to the total disk-node throughput sustained by the SIP (5 Mbytes/s), the throughput of a GigaView CD d/n architecture can be predicted to be n times the throughput of d/n CD-ROM units attached to a single disk node (Fig. 4 and 6).

Figure 8: *GigaView CD geographical system.*

4 Applications

Two application fields for the GigaView CD image servers are being considered at this time: geographical information systems, and medical imaging. Both fields require large amounts of pixmap data, as well as the ability to define relationships between various pixmap layers.

4.1 Geographical Systems

The GigaView CD architecture can be applied to store any kind of image data—from 1-D to 3-D files, and from bitmaps to real-color 24-bit high-resolution images. Typically, in the field of geographical information systems, scanned multilayer topographic maps consist of 1-byte pixels, whereas scanned cadastral maps are predominantly white and very sparse bitmaps. In order to pack scanned (500 dpi) maps of a significant region onto a disk pool of reasonable size, lossless compression techniques are highly valuable. High-performance zooming capabilities are further required, in order to reduce large areas to displayable size in real time.

Figure 8 illustrates a working geographical information system consisting of a four-disk GigaView CD back-end server and a Macintosh personal computer for display purposes. Visualization window on a high-resolution real-color display is about 1 Mbyte in size.

In uncompressed mode, the current PowerMac 6100/60 SCSI-interface limits SIP-to-host sustained throughput to 2.08 Mbytes/s. At the disk-node level however, high zoom rates (8) enable extents to be accessed at CD-ROM disk reading speed, achieving a raw throughput of 6.92 Mbytes/s (12/4), respectively of 2.31 Mbytes/s (4/2). In compressed mode, the combined disk-access and data decompression throughput reaches in the average case (typical topographic map) 2.68 Mbytes/s (12/4), resp. 1.22 Mbytes/s (4/2). The lower throughputs in compressed mode result from the decompression bottleneck due to limited T800 disk-node processing power.

These results illustrate the benefits of integrating local processing facilities at the disk-node level, saving valuable host communication bandwidth. They suggest that CD-ROM based multiprocessor multidisk architectures overcome the slow throughput of CD-ROM drives, and still benefit from the advantages of the CD-ROM : large capacity, low storage cost, and low distribution cost.

4.2 Medical Imaging

A 3-D MRI scan (Magnetic Resonance Imaging) has been acquired and stored on the GigaView CD parallel image server. The image size is 100 Mbytes (512-by-512-by-384 1-byte pixels). Thanks

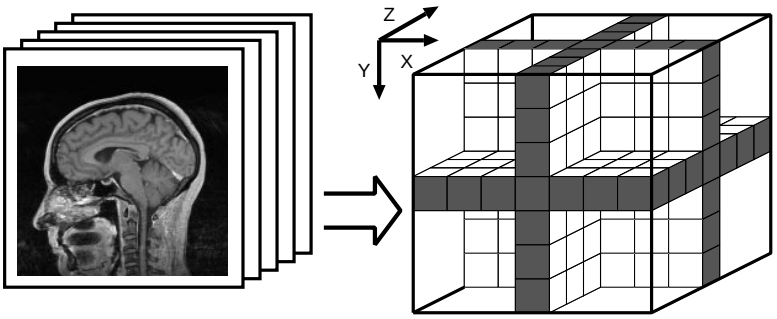

Figure 9: *MRI scan and 3-D extents.*

to the GigaView CD, image views orthogonal to the Euclidian axes can be interactively visualized, allowing to browse through image frames that come directly from CD-ROM disks, without the costly operation of preloading them into memory.

For this application, the 3-D images are divided in 3-D extents, which improve the locality of both disk and memory accesses. This feature is essential as access delays depend almost completely on access locality. Let us assume an image width (X-axis), height (Y-axis), and depth (Z-axis) of W, H, and D pixels; a visualization window width and height of w and h; and an extent size of ε pixels.

Consider first the case where the 3-D images are stored as a set of adjacent 2-D images (X-Y planes stacked along the Z-axis, top of Fig. 9). In this format, an extent has a width of W and a height of $\lceil \varepsilon / W \rceil$. Fetching a given visualization window along the X-Y plane requires accessing an extent for every $\lceil \varepsilon / W \rceil$ lines in the visualization window. Along the X-Z plane, it requires accessing an extent for every column in the visualization window. Along the Y-Z plane, it requires accessing for every visualization window column one extent per $\lceil \varepsilon / W \rceil$ pixels in the column. To give some numbers, assuming W at 2048 pixels and ε at 32768 pixels, we get a visualization window access time that is 64 times longer along the X-Z plane (resp. 2048 times longer along the Y-Z plane) than the access delay along the X-Y plane; the access anisotropy is important.

On the other hand, if we consider cubical extents, the number of extent accesses is identical along all three planes ($\lceil \frac{w}{\sqrt[3]{\varepsilon}} \rceil \cdot \lceil \frac{h}{\sqrt[3]{\varepsilon}} \rceil$). Furthermore, access to contiguous planes will be much faster, as the relevant extents can be maintained in disk-node extent cache memory.

Table 2 reports delays accessing an actual 3-D MRI scan stored on the GigaView CD 4/2 and 12/4 (512-by-512-by-384 1-byte pixels). Adequate image data division in 35-by-35-by-35 extents enables access times of the same order of magnitude along all three planes. Since extents are stored contiguously along the X-Y planes, higher extent access locality results in faster visualization window access along the X-Y planes than along the X-Z, resp. Y-Z planes.

Finally, thanks to 3-D extents, the amount of data read from the CD-ROM disks depends only on the visualization window size. This last feature becomes essential when considering that the 3-D scan of a complete human body represents about 24 Gbytes of data (2048-by-2048-by-2048 3-byte pixels).

5 Conclusion

The design, performance analysis, and applications of a multiprocessor multidisk CD-ROM image server have been reported. The GigaView CD is a transputer-based architecture connected through a standard SCSI-2 interface to a client host computer.

The originality of the design consists in combining high-performance real-time access with unlimited, inexpensive mass storage capacities through the use of removable CD-ROM medium. Unlike RAID storage devices, the proposed CD-ROM architecture provides control

GigaView CD layout	Planes		
	X-Y	X-Z	Y-Z
4 / 2	5.30 s	6.13 s	8.27 s
12 / 4	2.37 s	2.93 s	3.98 s

Table 2: *Delays accessing a 3-D MRI scan.*

over information distribution and features local processing facilities. The segmentation of image data into rectangular extents grants excellent data locality to random accesses of 2-D and 3-D pixmap images.The MDFS file system enables data access and processing to be pipelined, allowing decompression to be performed almost transparently or large images to be reduced to displayable size at CD-ROM disk reading speed.

Thanks to a highly scalable design, the proposed architecture can be tuned to achieve specific performance and storage capacity requirements. Assuming a high-performance client SCSI-bus interface, a GigaView CD architecture consisting of four disk nodes and twelve quad-speed CD-ROM drives achieves an application data transfer throughput of 2.5 Mbytes/s. Upgrading to readily available faster CD-ROM hardware will instantly yield sustained performance beyond the 3-Mbytes/s mark. Furthermore, combining CD-ROM disk access with local processing at the disk-node level allows for scaling large images to displayable size at near disk reading speed (e.g. a zoom rate of 4 yields a disk-level throughput of 4.6 Mbytes/s). Parallelization efficiency remains optimal when adding up to four disk nodes to the MPMD architecture, and proves to be higher with CD-ROM disk pools than with winchester-disk arrays [2]. Connecting multiple CD-ROM units to each disk node further increases performance, however, above two CD-ROM drives per node, at the expense of parallelization efficiency.

Current research aims at porting the GigaView CD concept to multiprocessor multidisk workstations and Windows NT PCs.

References

[1] D.A. Patterson, G.A. Gibson, and R.H. Katz. The case for RAID: Redundant arrays of inexpensive disks. In *Proceedings ACM SIGMOD Conference*, pages 106–113, Chicago, IL, May 1988.

[2] B.A. Gennart, B. Krummenacher, L. Landron, and R.D. Hersch. GigaView parallel image server performance analysis. In IOS Press, editor, *Transputer Applications and Systems, Proc. World Transputer Congress*, pages 120–135, Como, Sept. 1994.

[3] R. Harley. Mastering. In C. Sherman, editor, *CD-ROM Handbook*, chapter 15. McGraw-Hill, New York, NY, second edition, 1993.

[4] Inmos Databook Series. *The Transputer Databook*. 72 TRN 203 02. SGS-Thomson Microelectronics Group, third edition, 1992.

[5] C.W. Brown and B.J. Shepherd. *Graphics File Formats: Reference and Guide*, chapter 9. Manning Publications, Greenwich, CT, 1995.

[6] J.R. Fricks. Compact disc terminology. In C. Sherman, editor, *CD-ROM Handbook*, chapter 21. McGraw-Hill, New York, NY, second edition, 1993.

[7] American National Standards Institute (ANSI). *Small Computer System Interface - 2 (SCSI-2)*, chapter 13. X3.131-199X. Global Engineering Documents, Irvine, CA, Oct. 1991.

[8] International Organization for Standardization (ISO 9660). *Volume and File Structure of CD-ROM for Information Exchange*, first edition, 1988.