Active Disks for Large-Scale Data Processing

Active disk systems leverage the aggregate processing power of networked disks to offer greatly increased processing throughput for large-scale data mining tasks.

Erik Riedel
Hewlett-Packard Laboratories

Christos Faloutsos

Garth A. Gibson

David Nagle
Carnegie Mellon University

As processor performance increases and memory cost decreases, system intelligence continues to move away from the CPU and into peripherals. Storage system designers use this trend toward excess computing power to perform more complex processing and optimizations inside storage devices. To date, such optimizations take place at relatively low levels of the storage protocol. Trends in storage density, mechanics, and electronics eliminate the hardware bottleneck and put pressure on interconnects and hosts to move data more efficiently.

We propose using an *active disk* storage device that combines on-drive processing and memory with software downloadability to allow disks to execute application-level functions directly at the device. Moving portions of an application's processing to a storage device significantly reduces data traffic and leverages the parallelism already present in large systems, dramatically reducing the execution time for many basic data mining tasks.

TECHNOLOGY

As Figure 1 shows, current disk drives include all the components of a simple computer: a microprocessor, RAM, and a communications subsystem (SCSI), in addition to the specialized servo and signal processing hardware to handle drive control. Until 1999, this collection of chips formed the electronic backplane of a standard 3.5-inch disk drive. Most current-generation drives fit all these core drive-control and communications functions into a single application-specific integrated circuit (ASIC). If we extrapolate to the next generation of silicon process technology in .35 or .25 micron feature sizes, the specialized drive circuitry occupies approximately one-quarter of the chip, leaving sufficient area to include a 200-MHz ARM core or similar embedded microprocessor.

Disk drive and chip manufacturers are already pursuing this processor-in-ASIC technology. Infineon (formerly Siemens Microelectronics) markets a chip called the TriCore that includes a 100-MHz 32-bit microcontroller, up to 2 Mbytes of on-chip RAM, and customer-specific logic—such as the disk functions of Figure 1, upper right—in a .35 micron process. Cirrus Logic offers an integrated system-on-chip hard disk drive controller called 3Ci that includes a 25-MHz ARM core in the first generation, with promise of 200 MHz in the next generation.

Taking a larger system view, Table 1 shows details of several large database systems that manage transaction and data mining workloads. These trends and ratios in CPU versus aggregate processing power have remained roughly steady since we compiled this data in 1998 using information from the Transaction Processing Performance Council, Microsoft's TerraServer project, and vendor data sheets. Assuming a conservative 25 MHz of host-equivalent computing power available at the individual drives, a collection of 50 or 100 disk drives contains two to three times more aggregate computing power than even a powerful SMP server. Perhaps even more important for data-intensive applications, the aggregate server I/O subsystem transfer rates fall far below the maximum data bandwidth this number of drives can provide.

Processing power and memory inside disk drives currently optimize functions behind standardized interfaces

0018-9162/01/$10.00 © 2001 IEEE

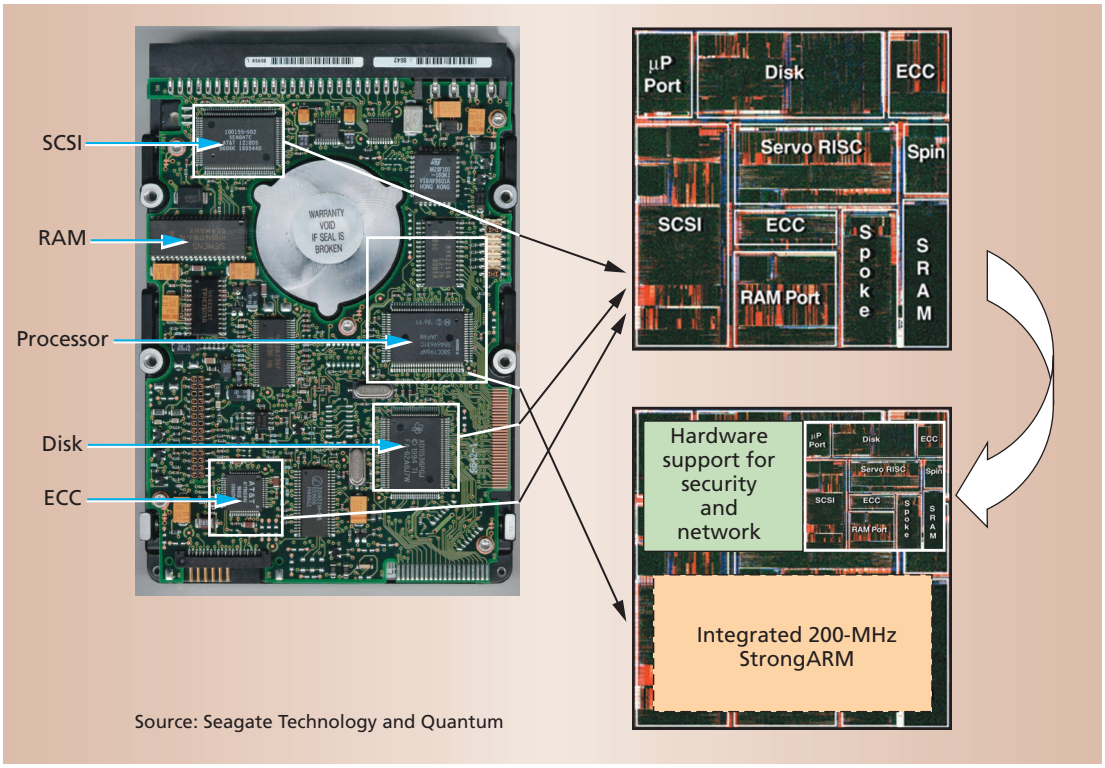


Figure 1. Hard drive processing architectures: Left: 1997 (3.5 × 6.5 inches)—All the control chips are separate. Upper right: 1999 (74 mm²)—The architecture combines many individual specialized chips into a single ASIC. Lower right: 2000 (74 mm²)—Advances in silicon process technology shrink the ASIC to a fraction of the original size, leaving room for a general-purpose RISC core and additional specialized functions.

such as SCSI or ATA, but limiting the interface to low-level, general-purpose tasks also limits the possible benefits of that processing power. With active disks, the rigid interface is broken and the excess computation power in drives is directly available for application-specific functions. The most compelling use of such processing power that scales with data size is large parallel scans.

APPLICATIONS

An increasing number of data-intensive applications require exactly this type of processing. Richer database structures, new content, new data sources, and novel applications for collected data have resulted in the development of a new class of data-intensive algorithms that require large amounts of disk space and high data-transfer rates for a variety of processing tasks. For example, an hour of video requires approximately 1 Gbyte of storage.

Carnegie Mellon's Informedia project uses a video database that holds more than 1 Tbyte of video from broadcast news sources, searchable by video, text, or audio content. In a search-by-content application, the user provides an image, text fragment, or audio segment and requests a set of similar images, pages, or sounds. The system then extracts feature vectors such as keywords, image edges, or color histograms from every image, then searches these feature vectors for *nearest neighbors*.[1]

The number of feature vectors for multimedia data types often exceeds several dozen properties of a particular image. In this case, two scan-intensive applications—the extraction of particular sets of features and the searching itself—often require a full scan.

Experience-on-demand applications collect sensor data from video cameras, microphones, and GPS transmitters to register experiences such as a firefighter

Table 1. Representative data mining servers and their aggregate processor and disk processing power.

System	Processors	On-disk processing	System bus	Storage throughput
Compaq ProLiant TPC-C 4 × 400-MHz Pentiums, 1 PCI, 141 disks	1,600 MHz	3,525 MHz	133 Mbyte/s	1,410 Mbyte/s
Microsoft TerraServer 8 × 440-MHz Alphas, 2 × 64-bit PCI, 324 disks	3,520 MHz	8,100 MHz	532 Mbyte/s	3,240 Mbyte/s
Digital AlphaServer TPC-C 500-MHz Alpha, 2 × 64-bit PCI, 61 disks	500 MHz	1,525 MHz	266 Mbyte/s	610 Mbyte/s
Digital AlphaServer TPC-D 12 612-MHz Alphas, 2 × 64-bit PCI, 521 disks	7,344 MHz	13,025 MHz	532 Mbyte/s	5,210 Mbyte/s

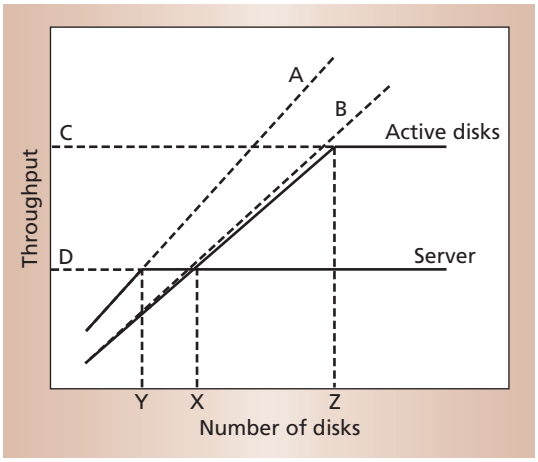


Figure 2. A model of an application's throughput running in an active disk system compared to running in a traditional single server system (line A: disk bandwidth; line B: processor performance). To the left of point Y, the traditional system is disk-bound. Below crossover point X, the active disk system is slower than the server due to its less powerful CPUs. Above point Z, even active disks saturate their interconnects. The traditional server system exhibits either interconnect or server bottlenecks above line D and can scale no further.

on a rescue mission. Users can apply this information later for training, planning, or data mining.

In medical image databases, a typical 3D brain image can consume between one and 100 Mbytes depending on the spatial resolutions and image depths. A medium-size hospital typically performs about 120,000 radiological imaging studies each year, including x-rays, producing more than 2 Tbytes of imaging data per year—with both the resolution and quantity of images increasing each year.

Data mining applications harvest massive amounts of customer data. For example, a large Pennsylvania retailer uses a data mining application that generates 5 Gbytes of point-of-sale data per week. Large telecommunication companies maintain tens of terabytes of historical call data. These databases must support ad hoc queries, and algorithms such as association discovery and classification require repeated scans of this data.[2]

ACTIVE DISK APPROACH

Successful active disk functions possess several basic characteristics. They

- leverage the parallelism available in systems with many disks;
- operate with a small amount of state, processing data as it streams off the disk; and
- execute relatively few instructions per byte of data.

These traits help develop an intuition about active disk system behavior relative to a traditional server with dumb disks. Figure 2 shows the basic trade-offs for active disk systems if we assume, for simplicity of analysis, that we can pipeline and overlap disk transfer, disk computation, interconnect transfer, and host computation with negligible start-up and postprocessing costs and that interconnect transfer rates always exceed single-disk rates. The slope of line A represents the raw disk limitation in both systems. The slope of line B represents the active disk CPU limitation, which may be less than the raw disk transfer rate for some applications.

The ratio of active disk to server processor speed is the most important system parameter. Near term, we expect 100- and 200-MHz microprocessors in drives and individual server CPUs of 500 to 1,000 MHz, a ratio of about 1 to 5. In this case, the aggregate active disk processing power exceeds the server processing power once more than five disks are working in parallel.

Prevailing technology trends predict processor performance (line B) continuing to improve by 60 percent per year and disk bandwidth (line A) improving by 40 percent per year. This will improve the ratio of processing power to disk bandwidth by 15 percent per year, narrowing the gap between lines A and B, and bringing active disks closer to the ideal of running at total storage bandwidth.

PROTOTYPE AND EXPERIMENTS

Our experimental testbed contained 10 prototype active disks, each emulated with a six-year-old 133-MHz DEC Alpha 3000/400 with 64 Mbytes of RAM, two 2.0-Gbyte Seagate Medalist disks, and the Digital Unix 3.2g operating system. For the server case, we used a single 500-MHz DEC AlphaStation 500/500 with 256 Mbytes of RAM, four 4.5-Gbyte Seagate Cheetah disks on two ultrawide SCSI buses, and the Digital Unix 3.2g operating system. An Ethernet switch and a 155-Mbps OC-3 ATM switch connected all these machines. Our experiments compared the performance of a single server with fast, directly attached SCSI disks against the same machine with network-attached active disks, each of which consisted of a workstation with two slower, directly attached SCSI disks.

Nearest-neighbor search in high-dimensionality data

First, we attempted a search variation that determines the k items in a database that are closest to a particular input item. Many classification and memory-based learning tasks require this kind of search. We used synthetic data that contains individual loan-applicant records with several independent attributes such as age, education, salary, make of car, and cost of house. Our test used a single target record as input and processed records from the database, maintaining a list of the k closest matches so far and adding the current record to the list if it was closer to the target than any record already in the list.

For the active disks system, we assigned each disk

an integral number of records and performed the comparisons directly at the drives. The server sends the target record to each of the disks to determine the k closest records in their portions of the database. The system returns these lists to the server and combines them to determine the overall k closest records. This application reduces the records in a database of arbitrary size to a constant-sized list of k records (k = 10 in our experiments), resulting in an arbitrarily large selectivity (data reduction). The state required at each disk equals the storage for the list of k closest records.

Figure 3 compares the performance of the traditional server against a system with active disks as the number of disks increases. For a small number of disks, the server performs better. The server is four times as powerful as a single active disk processor and can perform the computation at full disk rate. However, the server CPU saturates at 25.7 Mbytes/s with two disks, while the active disks system continues to scale linearly to 58 Mbytes/s using 10 disks (the maximum size of our experimental testbed). Extrapolating the data from the prototype to a larger system with 60 disks—the smallest system in Table 1—would provide a throughput of nearly 360 MbyteS/s.

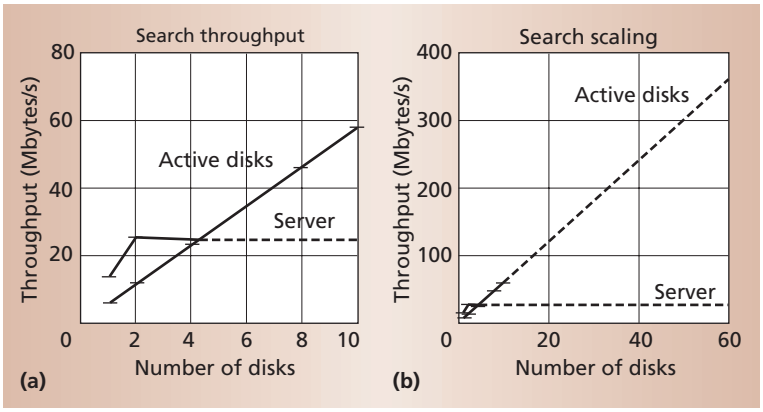


Figure 3. Performance of a search application running on a traditional server compared with an active-disks system: (a) compares systems with a few disks, while (b) compares systems with many disks.

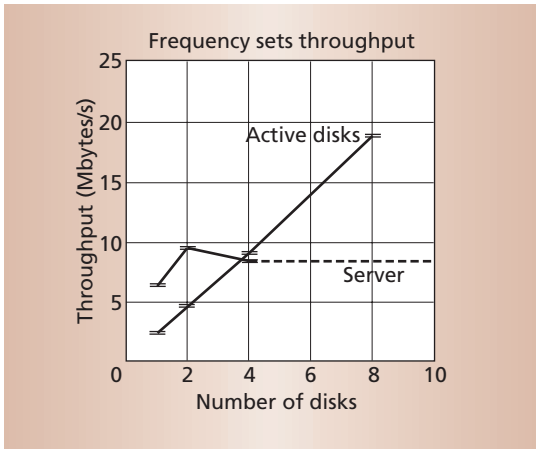


Figure 4. Throughput results for traditional and active disk systems running frequent sets applications: With more disks, the active disk system achieves a much higher throughput than the traditional server system.

Association rule discovery in retail data

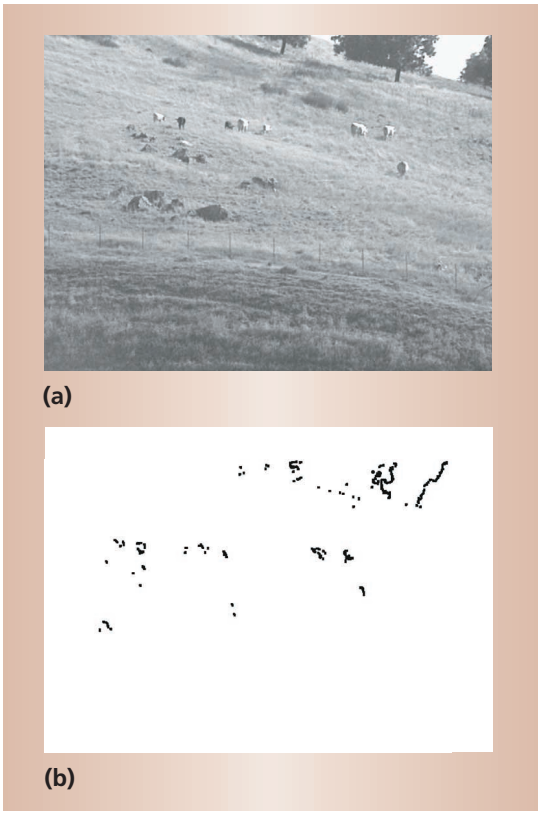
For our second application, we implemented an algorithm to discover association rules in sales transactions.[2] We used a database containing hypothetical point-of-sale data in which a record contains a transaction identifier, a customer identifier, and a list of items purchased on a particular shopping trip. Our use of frequent sets extracts rules in the form of "if a customer purchases items A and B, they are also likely to purchase item X", which merchants can use for inventory or store layout decisions. Our computations required several passes, first determining the items that occur most often (the 1-itemsets), then using this information to generate pairs of items that occur often (2-itemsets) and larger groupings (k-itemsets). We determined itemsets by making successive scans over the data in which each phase uses the k-itemset counts to create a list of candidate (k + 1)-itemsets until no k-itemsets met the desired support.

Active disk systems perform the counting portion of each phase directly at the drives. The server produces the list of candidate k-itemsets and provides this list to each disk. Each disk counts its portion of the transactions locally and returns these counts to the server. The server combines the counts, produces a list of candidate (k + 1) itemsets, and sends the list back to the disks. The application reduces an arbitrarily large number of transactions into a single, variably sized set of summary statistics. The state the disks require is the storage for the candidate k-itemsets and their counts at each stage. Figure 4 shows the results for the first two passes of the frequent sets application—those for the 1- and 2-itemsets. Again, the crossover point is at four drives, where the server system bottlenecks at 8.4 Mbytes/s and performance no longer improves, while the active disks system continues to scale linearly to 18.9 Mbytes/s.

Preprocessing for mixed-media mining with image data

For our first image-processing application, we looked at an application that detects edges and corners in a set of gray-scale images. Using real images, we attempted to detect cows in the landscape near San Jose, California.

The application processes a set of 256-Kbyte images and returns only the edges it finds in the data using a fixed 37-pixel mask. The tracking, feature extraction, and positioning applications operate on only a small subset of the original image data, focusing on a particular set of features such as the edges of objects rather than the entire image.



(a)

(b)

Figure 5. Edge detection scan using landscape image: (a) the raw image, and (b) the edges detected with a brightness threshold of 75.

The active disk system performs edge detection for each image directly at the drives, and the search returns only the edges to the server. A request for the raw image returns only the identified edges in Figure 5b, which the scan can represent much more compactly, reducing the amount of data transferred to the server for this particular image from 256 Kbytes to 9 Kbytes. Edge detection bottlenecks the server CPU at 1.4 Mbytes/s, while the active disk system scales to 3.2 Mbytes/s with 10 disks.

Image registration in medical data

Our second image processing application—the most computationally intensive we have studied—analyzed the image-processing portion of a magnetic resonance imaging brain scan. This analysis determined the set of parameters necessary to register—rotate and translate—an image with respect to a reference image to compensate for subject movement during the scan. The application processes 384-Kbyte images and returns a set of registration parameters for each image. The algorithm performs a fast Fourier transform, determines the parameters in Fourier space, and computes an inverse-FFT on the resulting parameters.

For the active disk system, this application is similar to edge detection. The system provides the reference image to all the drives and registers each image directly at the drives, returning only the final parameters—1.5 Kbytes for each image—to the server.

Figure 6 shows the results for the two image processing applications. Both required more CPU time than the simple comparisons of the search, counting, and frequent sets applications, leading to much lower throughput. Image registration achieves only 225 Kbps on the server system and 650 Kbps with 10 active disks.

Active disks or PC clusters?

A collection of commodity PCs with directly attached storage that are connected by high-speed networks offers performance improvements similar to active disks. The University of Tokyo[3] prototyped such an architecture, and variants of it exist in many commercial clustered database servers. PC clusters differ from active disks primarily in the level of integration. In a PC cluster, each individual disk contains a processor, an I/O bus to connect the disk or disks to a PC, the PC processor and memory, and a second interconnect that connects the PC to the rest of the cluster. An active disk system has only a single disk processor that places data directly on the cluster interconnect.

The active disk approach eliminates the need for the PC processor, PC memory subsystem, and I/O backplane, making active disks inherently less expensive than a PC solution. Active disks leverage the processing power that already exists on commodity disk drives by extending their interface and capabilities.

Active disks provide a means for accelerating an existing database system by moving data-intensive processing to the disks and off-loading the server CPU. However, a centralized processor is still required for some data operations such as those that merge the results of parallel computation—which means that active disks will always pass data they process to a central server, just as disks do today. Centralized processing eliminates the need for potentially complex disk-to-disk communication. Active disks can accelerate all existing database servers, whether large multiprocessors or clustered solutions.

RELATED WORK

Active disks offer the parallelism available in large storage systems. Although processing power on disk drives remains less than top-of-the-line server CPUs, more aggregate CPU power often resides in the disks than in the server. Partitioning applications across server nodes to take advantage of this parallelism results in a much higher total computation power than running applications only on the server.

Active disks also dramatically reduce interconnect bandwidth by filtering at the disks. Interconnect bandwidth, often the most significant bottleneck, remains at a premium compared to processing cycles. Whether scanning large objects to select specific records or fields or gathering summary statistics, disk-based filtering discards a fraction of the data that would otherwise move across the interconnect, dramatically reducing the bottleneck and greatly increasing the apparent

storage data rate. These two advantages promise order-of-magnitude improvements.

Our work inspired the adoption of active disks for several data mining applications, where large data sets are scanned for patterns of varying complexity. Subsequent work has shown that active disks efficiently support the core operations of a traditional database system, often with improvements comparable to purely scan-based functions.[4] This allows a large class of data-intensive applications to take advantage of active disks with changes to only a few database primitives. VLSI technology has evolved to the point that significant additional computational power comes at negligible cost. The "Embedded Processing" sidebar provides more information about these developments.

We have also demonstrated novel optimizations that take advantage of the improved scheduling knowledge available when applications operate close to the disk drives.[5] As disk drive manufacturers rethink the interfaces to storage devices, we are already moving closer to the general-purpose programmability that active disks support.

Related work on Network-Attached Secure Disks at Carnegie Mellon addresses making disk drives first-class network citizens, removing the server bottleneck for data access, and providing the necessary security functions.[6] More recent development efforts include the Internet Engineering Task Force iSCSI working group, which is developing a standard to apply commodity Ethernet networking and greatly simplify the networking of storage devices. In addition, the ANSI T10 standards body is considering an object-based storage proposal that evolves storage interfaces to allow individual disk drives more control and knowledge over data organization. These advances are leading the way for deployment of active disks so that any host on the network can leverage both the computation resources and the bandwidth reduction promised by operating closer to the data.

Further work at Carnegie Mellon[4] and by research groups at Berkeley,[7] Maryland, and Santa Barbara[8] provides additional details on the use of active disks for data-intensive operations. Our work on programmable devices also has led to discussions and expressions of interest in the data-storage industry.[9,10]

Disk drives with expanded computational power are rapidly becoming a reality. Active disks will facilitate more efficient processing of large data sets, scaling well beyond terabyte and petabyte stores using commodity components. Efficient processing greatly broadens the scope of data mining and other data-intensive applications as it eliminates the need for expensive, specially optimized systems designed exclusively for such processing. ✷

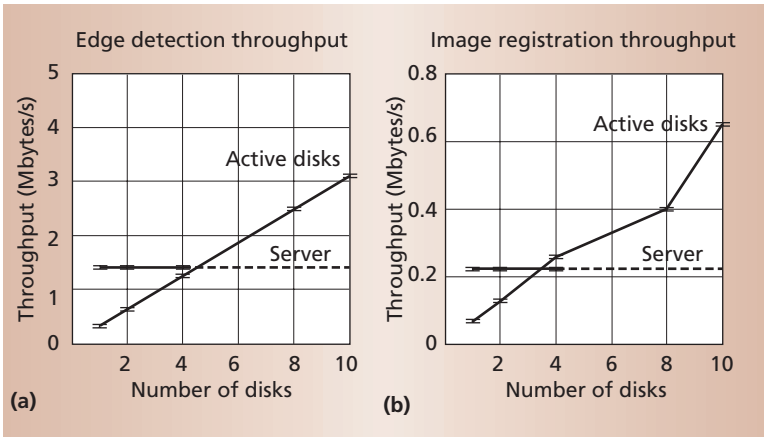


Figure 6. Results of (a) the edge detection test and (b) the image registration test. Even in CPU-bound tasks, active disks show linear or near linear improvement with increasing numbers of disks, whereas the traditional server's throughput flatlines in both tests.

Embedded Processing

The old lines between embedded and server processors are beginning to blur into a spectrum where power consumption and cost are the main differentiating factors, and raw performance is sufficient across the board.

The markets for embedded processors and desktop/server processors have traditionally been far apart. They represent different requirements, users, and research communities. Advances in technology have closed this gap considerably, and the differences between the two arenas are narrowing.

The first version of Cirrus Logic's 3Ci mass-storage processor contains a standard ARM7 RISC core. The second-generation product includes an ARM9 core that provides 220 MIPS at 200 MHz while consuming less than 500 mW of power. These contrast with the 25 W or more that a standard Pentium or the Alpha chip we used in our prototype consumes. To achieve these savings, the ARM core relies on a simpler design that lacks speculative execution, branch prediction, or floating-point processing. However, at 220 MIPS, the embedded processor remains more than sufficient for many core data mining application codes.

Embedded processors are typically strong in the digital signal processing that forms the basis of many multimedia functions, which are also increasingly important in many classes of data mining applications.

References

1. C. Faloutsos, *Searching Multimedia Databases by Content,* Kluwer Academic, Boston, 1996.
2. R. Agrawal and R. Srikant, "Fast Algorithms for Mining Association Rules," *VLDB J.,* Sept. 1994, pp. 487-499.
3. M. Oguchi and M. Kitsuregawa, "Parallel Data Mining on ATM-Connected PC Cluster and Optimization of Its Execution Environments," *Proc. Int'l Parallel Distributed Processing Symp.* (IPDPS 2000), Springer-Verlag, Berlin, LNCS 1911, 2000, pp. 366-373.
4. E. Riedel, "Active Disks—Remote Execution for Network-Attached Storage," doctoral dissertation, tech. report CMU-CS-99-177, Computer Engineering Dept., Carnegie Mellon Univ., Pittsburgh, 1999.
5. E. Riedel et al., "Data Mining on an OLTP System (Nearly) for Free," *Proc. ACM Sigmod Int'l Conf. Management of Data,* ACM Press, New York, pp. 13-21.

6. G. Gibson et al., "A Cost-Effective, High-Bandwidth Storage Architecture," *Proc. Conf. Architectural Support for Programming Languages and Operating Systems*, IEEE Press, Piscataway, N.J., Oct. 1998, pp. 92-103.
7. K. Keeton, D.A. Patterson, and J.M. Hellerstein, "A Case for Intelligent Disks (IDISKs)," *Sigmod Record*, Sept. 1998, pp. 42-52.
8. A. Acharya, M. Uysal, and J. Saltz, "Active Disks," *Proc. Conf. Architectural Support for Programming Languages and Operating Systems*, IEEE Press, Piscataway, N.J., Oct. 1998, pp. 81-91.
9. J. Gray, "What Happens When Processing, Storage, and Bandwidth Are Free and Infinite?" Keynote address, *Input/Output in Parallel and Distributed Computer Systems* (IOPADS 97), http://www.research.microsoft.com/~gray/talks/IOPADS.ppt.
10. G. Gibson, "What Do We Do with Excess Computational Power in Storage Devices?" *Network Attached Storage Devices Workshop*, June 1998, http://www.nsic.org/nasd/1998-jun/gibson.pdf.

Erik Riedel *is a researcher with Hewlett-Packard Laboratories, Palo Alto, Calif. His research interests include pervasive network storage, globally distributed systems, better storage interfaces, and security. He received a PhD in computer engineering from Carnegie Mellon University. Contact him at riedel@hpl.hp.com.*

Christos Faloutsos *is a professor in the School of Computer Science at Carnegie Mellon University. His research interests include data mining, indexing in relational and multimedia databases, and database performance. He received a PhD in computer science from the University of Toronto. Contact him at christos@cs.cmu.edu.*

Garth A. Gibson *is an associate professor in the School of Computer Science at Carnegie Mellon University and CTO of Panasas Inc.. His research interests include network-attached storage, object-based storage, RAID, and storage networking. He received a PhD in computer science from the University of California, Berkeley. Contact him at garth@cs.cmu.edu.*

David Nagle *is a senior research computer scientist in the School of Computer Science at Carnegie Mellon University. His research interests include network-attached storage, security for storage, MEMS-based storage systems, and storage networking. He received a PhD in computer engineering from the University of Michigan. Contact him at nagle@cs.cmu.edu.*