PCT

WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 :
G06F 13/14, 13/00, 13/10, 13/12

A1

(11) International Publication Number: WO 99/26150

(43) International Publication Date: 27 May 1999 (27.05.99)

(21) International Application Number: PCT/US98/21203

(22) International Filing Date: 8 October 1998 (08.10.98)

(30) Priority Data:

60/065,848	14 November 1997 (14.11.97)	US
09/034,247	4 March 1998 (04.03.98)	US
09/034,248	4 March 1998 (04.03.98)	US
09/034,812	4 March 1998 (04.03.98)	US

(81) Designated States: AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, DE, DK, EE, ES, FI, GB, GE, GH, GM, HR, HU, ID, IL, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, UA, UG, UZ, VN, YU, ZW, ARIPO patent (GH, GM, KE, LS, MW, SD, SZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published
With international search report.
Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.

(71) Applicant: 3WARE, INC. [US/US]; 420 Waverly Street, Palo Alto, CA 94301 (US).

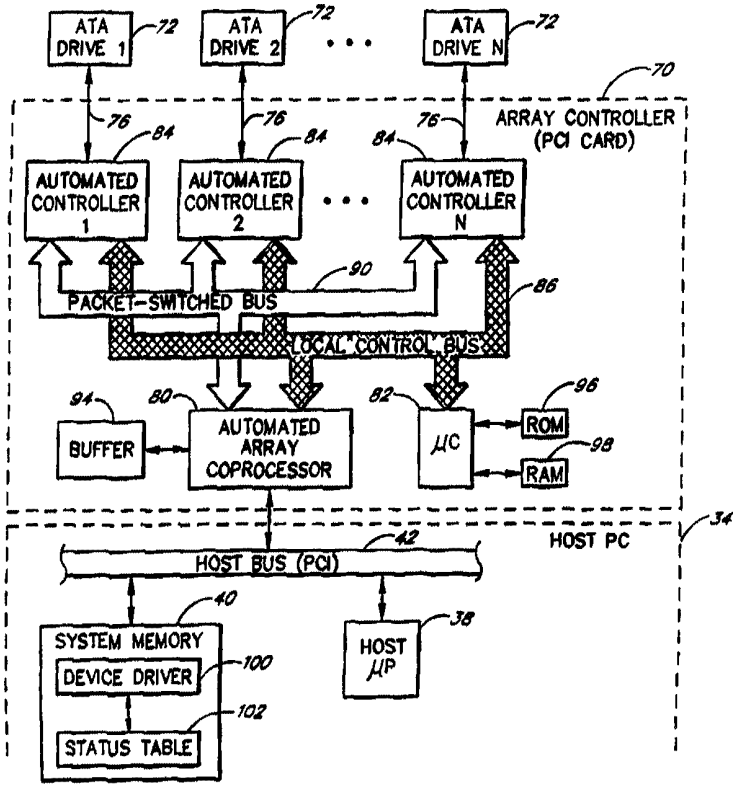
(72) Inventors: MCDONALD, James, A.; 940 Colonial Lane, Palo Alto, CA 94301 (US). HERZ, John, Peter; 36 Pine Lane, Los Altos, CA 94022 (US). ALTMAN, Mitchell, A.; 572 Hill Street, #Penthouse, San Francisco, CA 94114 (US). SMITH, William, Edward, III; 23797 Thurston Court, Hayward, CA 94568 (US).

(74) Agent: SIMPSON, Andrew, H.; Knobbe, Martens, Olson and Bear, LLP, 16th floor, 620 Newport Center Drive, Neuport Beach, CA 92660 (US).

(54) Title: HIGH–PERFORMANCE ARCHITECTURE FOR DISK ARRAY CONTROLLER

(57) Abstract

A high–performance RAID system for a PC comprises a controller card (70) which controls an array of ATA disk drives (72). The controller card (70) includes an array of automated disk drive controllers (84), each of which controls one respective disk drive (72). The disk drive controllers (84) are connected to a microcontroller (82) by a control bus (86) and are connected to an automated coprocessor (80) by a packet–switched bus (90). The coprocessor (80) accesses system memory (40) and a local buffer (94). In operation, the disk drive controllers (84) respond to controller commands from the microcontroller (82) by accessing their respective disk drives (72), and by sending packets to the coprocessor (80) over the packet–switched bus (90). The packets carry I/O data (in both directions, with the coprocessor filling–in packet payloads on I/O writes), and carry transfer commands and target addresses that are used by the coprocessor (80) to access the buffer (94) and system memory (40). The packets also carry special completion values (generated by the microcontroller) and I/O request identifiers that are processed by a logic circuit (144) of the coprocessor (80) to detect the completion of processing of each I/O request. The coprocessor (80) grants the packet–switched bus (90) to the desk drive controllers (84) using a round robin arbitration protocol which guarantees a minimum I/O bandwidth to each disk drive (72). This minimum I/O bandwidth is preferably greater than the sustained transfer rate of each disk drive (72), so that all drives of the array can operate at the sustained transfer rate without the formation of a bottleneck.



FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	**ES**	Spain	**LS**	Lesotho	**SI**	Slovenia
AM	Armenia	**FI**	Finland	**LT**	Lithuania	**SK**	Slovakia
AT	Austria	**FR**	France	**LU**	Luxembourg	**SN**	Senegal
AU	Australia	**GA**	Gabon	**LV**	Latvia	**SZ**	Swaziland
AZ	Azerbaijan	**GB**	United Kingdom	**MC**	Monaco	**TD**	Chad
BA	Bosnia and Herzegovina	**GE**	Georgia	**MD**	Republic of Moldova	**TG**	Togo
BB	Barbados	**GH**	Ghana	**MG**	Madagascar	**TJ**	Tajikistan
BE	Belgium	**GN**	Guinea	**MK**	The former Yugoslav Republic of Macedonia	**TM**	Turkmenistan
BF	Burkina Faso	**GR**	Greece			**TR**	Turkey
BG	Bulgaria	**HU**	Hungary	**ML**	Mali	**TT**	Trinidad and Tobago
BJ	Benin	**IE**	Ireland	**MN**	Mongolia	**UA**	Ukraine
BR	Brazil	**IL**	Israel	**MR**	Mauritania	**UG**	Uganda
BY	Belarus	**IS**	Iceland	**MW**	Malawi	**US**	United States of America
CA	Canada	**IT**	Italy	**MX**	Mexico	**UZ**	Uzbekistan
CF	Central African Republic	**JP**	Japan	**NE**	Niger	**VN**	Viet Nam
CG	Congo	**KE**	Kenya	**NL**	Netherlands	**YU**	Yugoslavia
CH	Switzerland	**KG**	Kyrgyzstan	**NO**	Norway	**ZW**	Zimbabwe
CI	Côte d'Ivoire	**KP**	Democratic People's Republic of Korea	**NZ**	New Zealand		
CM	Cameroon			**PL**	Poland		
CN	China	**KR**	Republic of Korea	**PT**	Portugal		
CU	Cuba	**KZ**	Kazakstan	**RO**	Romania		
CZ	Czech Republic	**LC**	Saint Lucia	**RU**	Russian Federation		
DE	Germany	**LI**	Liechtenstein	**SD**	Sudan		
DK	Denmark	**LK**	Sri Lanka	**SE**	Sweden		
EE	Estonia	**LR**	Liberia	**SG**	Singapore		

HIGH-PERFORMANCE ARCHITECTURE FOR DISK ARRAY CONTROLLER

FIELD OF THE INVENTION

The present invention relates to disk arrays, and more particularly, relates to hardware and software architectures for hardware-implemented RAID (Redundant Array of Inexpensive Disks) and other disk array systems.

BACKGROUND OF THE INVENTION

A RAID system is a computer data storage system in which data is spread or "striped" across multiple disk drives. In many implementations, the data is stored in conjunction with parity information such that any data lost as the result of a single disk drive failure can be automatically reconstructed.

One simple type of RAID implementation is known as "software RAID." With software RAID, software (typically part of the operating system) which runs on the host computer is used to implement the various RAID control functions. These control functions include, for example, generating drive-specific read/write requests according to a striping algorithm, reconstructing lost data when drive failures occur, and generating and checking parity. Because these tasks occupy CPU bandwidth, and because the transfer of parity information occupies bandwidth on the system bus, software RAID frequently produces a degradation in performance over single disk drive systems.

Where performance is a concern, a "hardware-implemented RAID" system may be used. With hardware-implemented RAID, the RAID control functions are handled by a dedicated array controller (typically a card) which presents the array to the host computer as a single, composite disk drive. Because little or no host CPU bandwidth is used to perform the RAID control functions, and because no RAID parity traffic flows across the system bus, little or no degradation in performance occurs.

One potential benefit of RAID systems is that the input/output ("I/O") data can be transferred to and from multiple disk drives in parallel. By exploiting this parallelism (particularly within a hardware-implemented RAID system), it is possible to achieve a higher degree of performance than is possible with a single disk drive. The two basic types of performance that can potentially be increased are the number of I/O requests processed per second ("transactional performance") and the number of megabytes of I/O data transferred per second ("streaming performance").

Unfortunately, few hardware-implemented RAID systems provide an appreciable increase in performance. In many cases, this failure to provide a performance improvement is the result of limitations in the array controller's bus architecture. Performance can also be adversely affected by frequent interrupts of the host computer's processor.

In addition, attempts to increase performance have often relied on the use of expensive hardware components. For example, some RAID array controllers rely on the use of a relatively expensive microcontroller that can process I/O data at a high transfer rate. Other designs rely on complex disk drive interfaces, and thus require the use of expensive disk drives.

The present invention addresses these and other limitations in existing RAID architectures.

SUMMARY OF THE INVENTION

The present invention provides a high-performance architecture for a hardware-implemented RAID or other disk array system. An important benefit of the architecture is that it provides a high degree of performance (both transactional and streaming) without the need for disk drives that are based on expensive or complex disk drive interfaces.

In a preferred embodiment, the architecture is embodied within a PC-based disk array system which comprises an array controller card which controls an array of ATA disk drives. The controller card includes an array of automated ATA disk drive controllers, each of which controls a single, respective ATA drive.

The controller card also includes an automated coprocessor which is connected to each disk drive controller by a packet-switched bus, and which connects as a busmaster to the host PC bus. The coprocessor is also connected to a local I/O data buffer of the card. As described below, a primary function of the coprocessor is to transfer I/O data between the disk drive controllers, the system memory, and the buffer in response to commands received from the disk drive controllers. Another function of the coprocesor is to control all accesses by the disk drive controllers to the packet-switched bus, to thereby control the flow of I/O data.

The controller card further includes a microcontroller which connects to the disk drive controllers and to the coprocessor by a local control bus. The microcontroller runs a control program which implements a RAID storage configuration. Because the microcontroller does not process or directly monitor the flow of I/O data (as described below), a low-cost, low-performance microcontroller can advantageously be used.

In operation, the controller card processes multiple I/O requests in at-a-time, and can process multiple I/O requests without interrupting the host computer. As I/O requests are received from the host computer, the microcontroller generates drive-specific sequences of controller commands (based on the particular RAID configuration), and dispatches these controller commands over the local control bus to the disk drive controllers. In addition to containing disk drive commands, these controller commands include transfer commands and target addresses that are (subsequently) used by the coprocessor to transfer I/O data to and from system memory and the local buffer.

Some of the controller commands also include disk completion values and tokens (I/O request identifiers) that are used by the coprocessor to monitor the completion status of pending I/O requests. The disk completion values are generated by the microcontroller such that the application of a specific logic function to all of the disk completion values for a given I/O request produces a final completion value that is known *a priori* to the coprocessor. As described below, this enables the coprocessor to detect the completion of processing of an I/O request without prior knowledge of the details (number of invoked disk drives, etc.) of the I/O request.

In response to the controller commands, the disk drive controllers access their respective disk drives and send packets to the coprocessor over the packet-switched bus. These packets carry I/O data (in both directions, with the coprocessor filling-in packet payloads on I/O writes), and carry transfer commands and target addresses that are used by the coprocessor to access the buffer and system memory. During this process, the coprocessor grants the packet-switched bus to the disk drive controllers (for the transmission of a single packet) using a round robin arbitration protocol which guarantees a minimum I/O bandwidth to each disk drive. The minimum bandwidth is equal

to 1/N of total I/O bandwidth of the packet-switched bus, where N is the number of disk drive controllers (and disk drives) in the array.

Because this minimum I/O bandwidth is greater than or equal to the sustained transfer rate of each disk drive, all N drives can operate concurrently at the sustained transfer rate indefinitely without the formation of a bottleneck. When the packet-switched bus is not being used by all of the disk drive controllers (i.e., one or more disk drive controllers has no packets to transmit), the arbitration protocol allows other disk drive controllers to use more than the guaranteed minimum I/O bandwidth. This additional I/O bandwidth may be used, for example, to transfer I/O data at rate higher than the sustained transfer rate when the requested I/O data resides in the disk drive's cache.

The disk drive controllers process their respective sequences of controller commands asynchronously to one another; thus, the disk drive controllers that are invoked by a given I/O request can finish processing the I/O request in any order. When a given disk drive controller finishes processing an I/O request, the controller sends a special completion packet to the coprocessor. This completion packet contains the completion value that was assigned to the disk drive controller, and contains an identifier (token) of the I/O request.

Upon receiving the completion packet, the coprocessor cumulatively applies the logic function to the completion value and all other completion values (if any) that have been received for the same I/O request, and compares the result to the final completion value. If a match occurs, indicating that all disk drives invoked by the I/O request have finished processing the I/O request, the coprocessor uses the token to inform the host computer and the microcontroller of the identity of the completed I/O request. Thus, the microcontroller monitors the completion status of pending I/O requests without directly monitoring the flow of I/O data.

BRIEF DESCRIPTION OF THE DRAWINGS

There and other features of the architecture will now be described in further detail with reference to the drawings of the preferred embodiment, in which:

Figure 1 illustrates a prior art disk array architecture.

Figure 2 illustrates a disk array system in accordance with a preferred embodiment of the present invention.

Figure 3 illustrates the general flow of information between the primary components of the Figure 2 system.

Figure 4 illustrates the types of information included within the controller commands.

Figure 5 illustrates a format used for the transmission of packets.

Figure 6 illustrates the architecture of the system in further detail.

Figure 7 is a flow diagram which illustrates a round robin arbitration protocol which is used to control access to the packet-switched bus of Figure 2.

Figure 8 illustrates the completion logic circuit of Figure 6 in further detail.

Figure 9 illustrates the transfer/command control circuit of Figure 6 in further detail.

Figure 10 illustrates the operation of the command engine of Figure 9.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

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.