United States Patent [19]

Gelb et al.

[11] **Patent Number: 5,018,060**

[45] **Date of Patent: May 21, 1991**

[54] **ALLOCATING DATA STORAGE SPACE OF PERIPHERAL DATA STORAGE DEVICES USING IMPLIED ALLOCATION BASED ON USER PARAMETERS**

[75] Inventors: **Jack P. Gelb; John C. Tyrrell,** both of San Jose, Calif.

[73] Assignee: **IBM Corporation,** Armonk, N.Y.

[21] Appl. No.: **301,970**

[22] Filed: **Jan. 26, 1989**

[51] **Int. Cl.**[5] **G06F 13/00;** G06F 12/00

[52] **U.S. Cl.** **364/200;** 364/970; 364/970.2; 364/968; 364/967; 364/967.2; 364/964.6; 364/253; 364/253.2; 364/253.1; 364/243; 364/245.8; 364/245.5

[58] **Field of Search** ... 364/200 MS File, 900 MS File

[56] **References Cited**

U.S. PATENT DOCUMENTS

4,024,507	5/1977	Berkling	364/200
4,121,286	10/1978	Venton	364/200
4,310,883	1/1982	Clifton	364/200
4,313,161	1/1982	Hardin	364/200
4,429,363	1/1984	Duke	364/200
4,463,424	7/1984	Mattson	364/200
4,466,059	8/1984	Bastian	364/200
4,574,346	3/1986	Hartung	364/200
4,653,021	3/1987	Takagi	364/900
4,660,130	4/1987	Bartley	364/200
4,727,487	2/1988	Masui et al.	364/300
4,769,782	9/1988	Iwanaga	364/900
4,771,375	9/1988	Belgin	364/200
4,805,093	2/1989	Ward	364/200

OTHER PUBLICATIONS

Garbage Collection in a Large Lisp System, David A. Moon 1984, ACM. pp. 235–246.

Garbage Collection in a Very Large Address Space, by Peter B. Bishop, MIT, Cambridge, Mass. 1975.

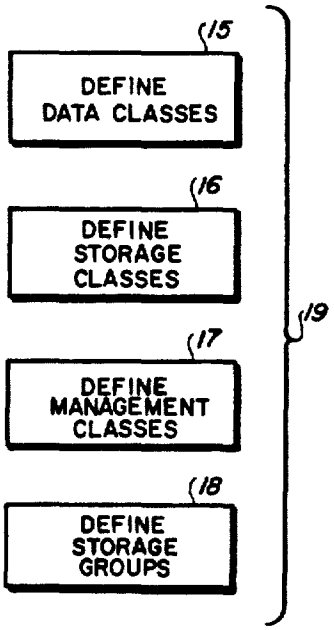
Generation Scavenging: A Non-Disruptive High Performance Storage Reclamation Algorithm, 1984 ACM, by David Ungar, pp. 157–167.

Primary Examiner—Thomas C. Lee
Assistant Examiner—Eric Coleman
Attorney, Agent, or Firm—Herb Somermeyer

[57] **ABSTRACT**

Units of data (data sets, data bases, etc.) are allocated data storage space in a data storage system based on implicit analysis of the unit of data. A plurality of data classes, each defining predetermined characterisitics of diverse units of data, are established for the data storage system. A plurality of storage classes, each defining predetermined sets of storage performance and availability requirements are established for the data storage system. A plurality of management classes, each defining respective diverse sets of life cycle attributes for units of data are established for the data storage system. A plurality of storage groups, each defining diverse predetermined performance device and management available in the data storage system but independently of the individual storage devices of the data storage system are established. The devices are selectively assigned to different ones of the established storage groups. Each received spaced allocation requested has its parameters (source, type of data, etc.) matched with the data, storage and management classes for assignment of one each of those classes to the unit of data related to the allocation request. A storage group is also assigned. The matching of the different classes and group are independent of one another. Allocation is based upon the resulting class and group selection.

10 Claims, 3 Drawing Sheets



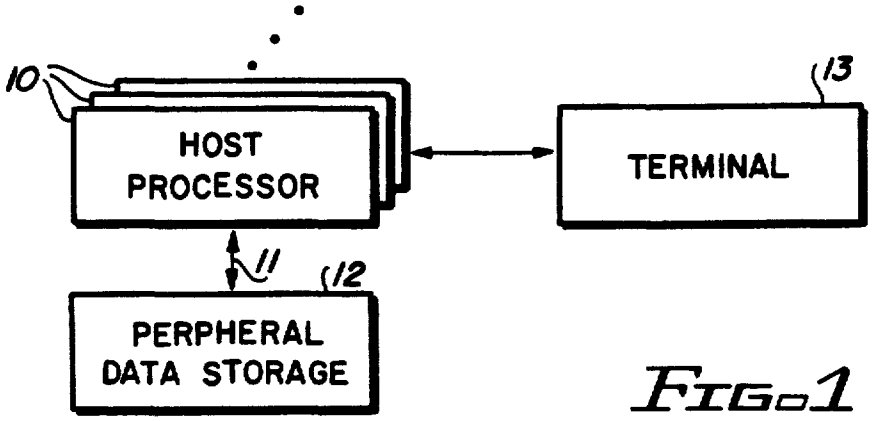


FIG. 1

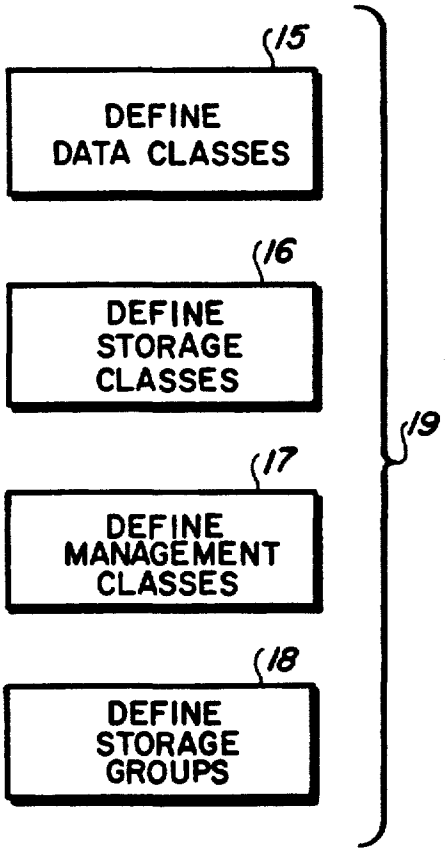


FIG. 2

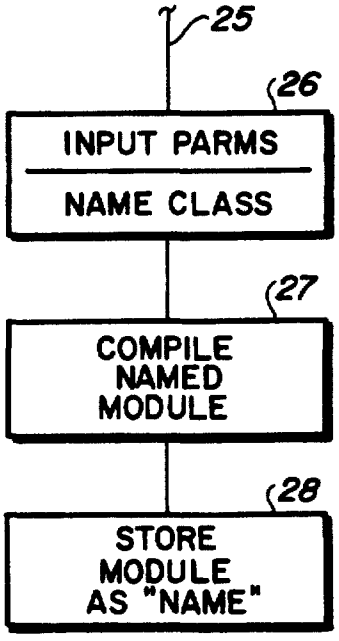


FIG. 3

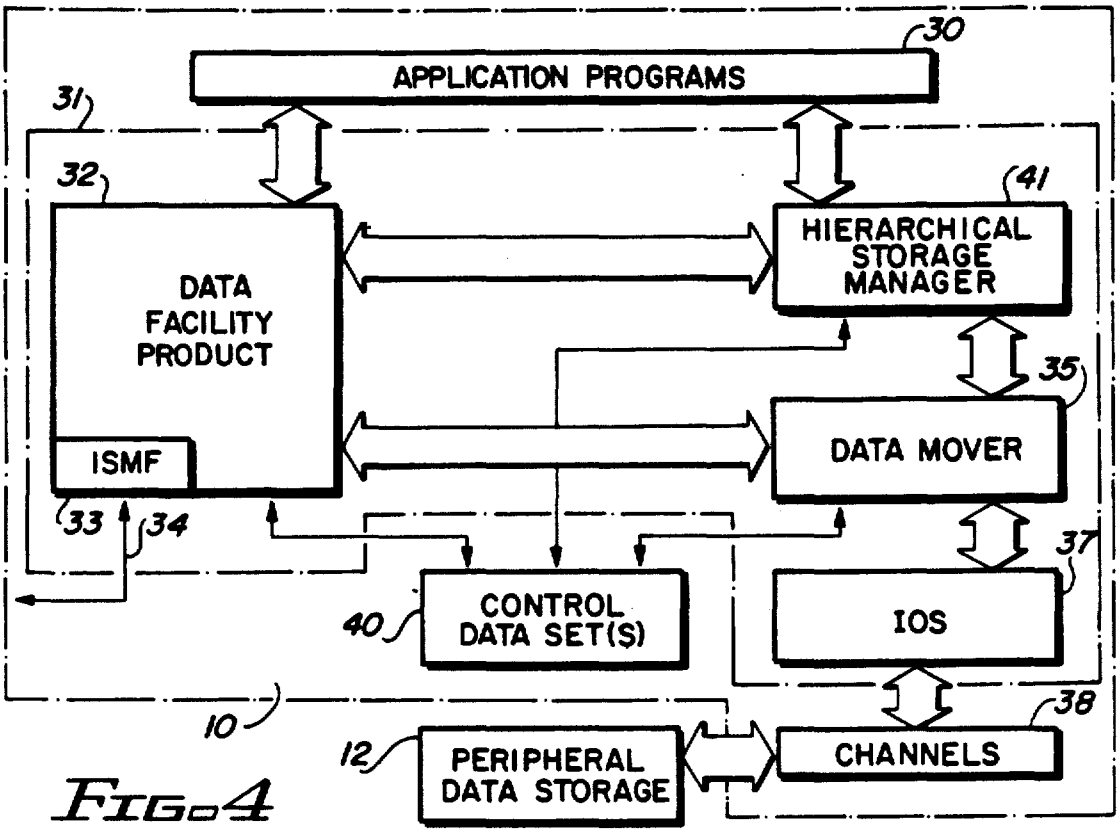
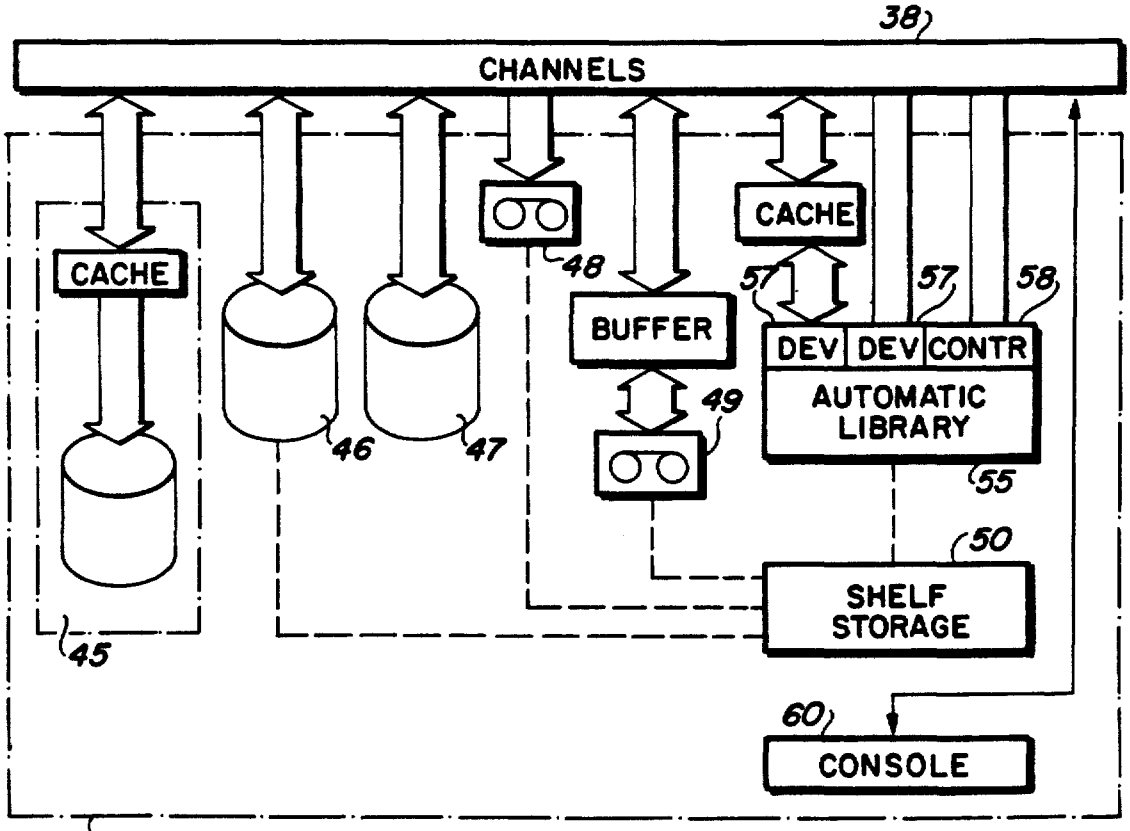


FIG-4



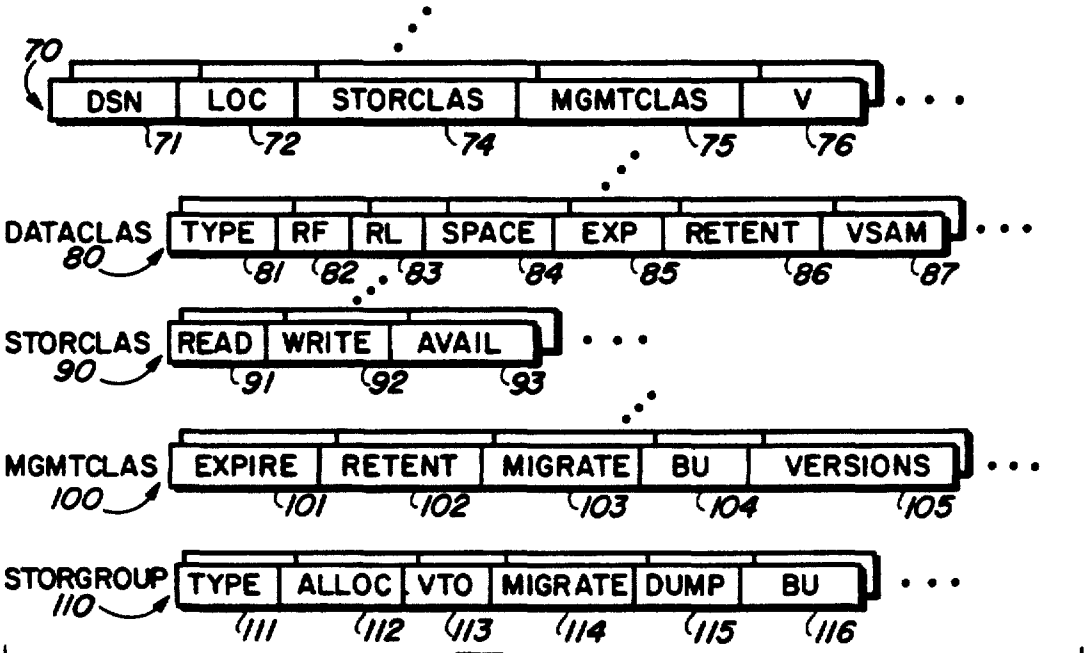


FIG. 6

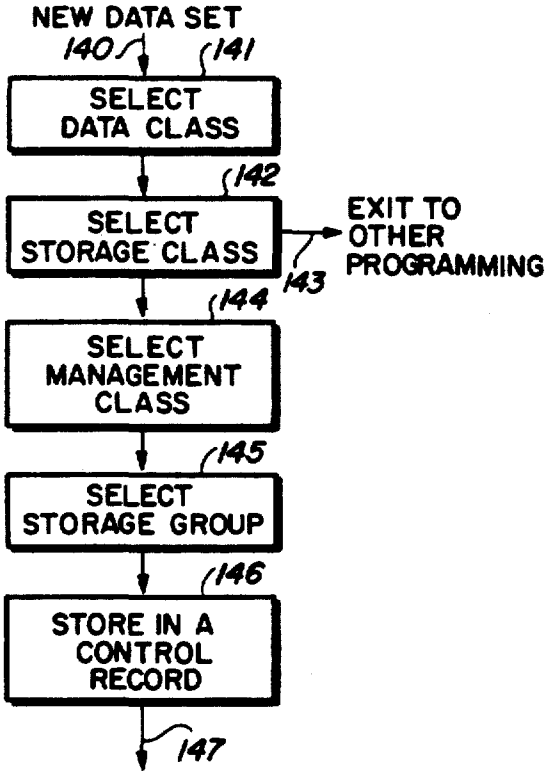


FIG. 7

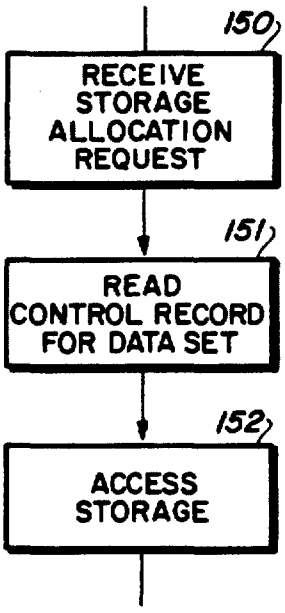


FIG. 8

ALLOCATING DATA STORAGE SPACE OF PERIPHERAL DATA STORAGE DEVICES USING IMPLIED ALLOCATION BASED ON USER PARAMETERS

FIELD OF THE INVENTION

This invention relates to computer-controlled peripheral data storage and more particularly to the global management of data storage for provided transparent isolation of the peripheral data storage characteristics from application programs using the peripheral data storage.

BACKGROUND OF THE INVENTION

Storing computerized data (coded or uncoded—uncoded is also termed image, graphics or raster data) in peripheral devices and subsystems has become an important cost consideration in modern day information processing systems. Further, the immensity of the storage capacity available in information processing systems has reached staggering levels. The current way of managing the peripheral storage of data is fast becoming unsatisfactory because of the costs involved and difficulty in managing the data storage space, providing performance (rapid access to data storage spaces or areas) as well as placing the burden on application programmers to effectively utilize such immense peripheral data storage capacity. It is desired to insulate the application programmers from the current requirement that their respective programs must have some information about the physical parameters of the peripheral data storage. It is desired to provide a completely logical control wherein the application programmers need only concern their programming with data sets, data bases, and the like as such items best serve the application program. Such data set considerations can include a broad indication of desired performance, indications of size, need for availability and reliability levels, and the like. All factors concerning space management, performance management, availability management and device install management of the peripheral subsystems should not be noticeable to the user programmer involved with application programs. To date, this has not been the situation.

DISCUSSION OF THE PRIOR ART

Initially job control language (JCL) required the computer user to specify many parameters of program execution including specifications of peripheral data storage. Then each time the peripheral data storage was changed, a corresponding change was required in the JCL statements for the programs to be executed. An example of JCL control of a data processor is shown by Poublan et al in U.S. Pat. No. 4,104,718. It is this interaction between operation of the peripheral data storage and the user application programs that is to be minimized, if not eliminated.

It is desired to let the user or application programmer issue specifications, preferably implicitly, of program execution and data storage requirements which are at a relatively high or logical level. The term "logical" herein refers to the specifications for desired program execution, particularly for peripheral data storage. It is desired that all allocation and controls of the peripheral data storage be removed from direct intervention by the application programmer.

The term "logical" has been used for years to describe program denominated units, such as "logical devices". Such logical descriptions did not remove the interaction of the application programmer with peripheral data storage. An example of such logical device specifications is found in the Christian et al patent U.S. Pat. No. 4,403,288 wherein one physical device (DASD—direct access storage device) was addressable as four logical devices. The proliferation of DASD in peripheral data storage subsystems resulted in a centralized DASD storage space allocation program called DADSM (direct access device space management). Still the application programmer had to request space in terms of device parameters and types. See the article by Duvall et al, "Asynchronous Allocation Requests", IBM TECHNICAL DISCLOSURE BULLETIN, Vol. 25 No. 8; January 1983, pp 4148–4150. It is desired to remove device parameter specifications from the application programmer; let the application programmer only be concerned with high or logical levels of specification independent of device parameters. This desire requires implicit translation of the logical parameter specification into data storage device selections and space allocation on selected devices.

March in the article "Data Base Control and Processing System", IBM TECHNICAL DISCLOSURE BULLETIN, Vol. 25, No. 7A, December 1982, pp 357814 3582 illustrates some of the complexity involved with data base management. It is desired to remove the complexity from adversely affecting application programmer efforts.

White in U.S. Pat. No. 4,467,421 describes a virtual data storage system which employs a memory control processor external to a host processor which divides user-defined data sets into blocks of a size convenient for storage in peripheral data storage devices. The memory processor assigns such blocks to storage locations on peripheral data storage devices. This memory processor intercepts the device oriented input-output commands from the host processor and adjusts operation of the peripheral data storage subsystem to make it more efficient. From this brief description, it is readily seen that the application programmer still is required to have JCL statements that are device dependent even if an external memory processor interprets the device dependencies differently than intended by the application programming. While this arrangement may ease the effects of changing device types and the like, it still seems to saddle the application programmer with device parameter considerations. It is desired to avoid the continuance of such a requirement.

Hartung in U.S. Pat. No. 4,638,425 shows a cached-DASD data storage subsystem in which the application programmer may insert a storage parameter indication of whether certain data records can be stored primarily in volatile (cache) peripheral storage or is primarily retentively stored (DASD) in the peripheral subsystem. When primarily stored in the cache, performance is enhanced but the application program is responsible for any lost data due to power problems or equipment failures. The selection is based on "use" status, i.e. is the use interactive with terminals (to be cached for performance) or is the use post-terminal activity (to be primarily retentively stored). While the use indication appears logical, it is seen that the application programmer has to understand important operational characteristics of the peripheral data storage subsystem. It is desired to make such determinations implicit rather than explicit.

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.