2189

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Application No.: 10/109,186
Filed: March 28, 2002
Inventor(s):
Hans F. van Rietschote

Title: Disaster Recovery and Backup Using Virtual Machines

Examiner: Chace, Christian P.
Group/Art Unit: 2189
Atty. Dkt. No: 5760-00400/VRTS 0064

I hereby certify that this correspondence is being deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450, on the date indicated below.

Lawrence J. Merkel
Printed Name

Signature 11/14/05
Date

RESPONSE TO OFFICE ACTION OF September 26, 2005

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

This paper is submitted in response to the Office Action of September 26, 2005, to further highlight why the application is in condition for allowance.

Please amend the case as set forth below:

IN THE SPECIFICATION:

Please amend the paragraph beginning on page 24, line 21, as set forth below:

It is noted that, in various embodiments shown above, the backup program 42, the checkpoint program 76, the recovery program 78, and/or the image 40 of the virtual machine 16A are shown stored on various storage devices. Generally, any one or more of the above (and/or the VM kernel 18A, the O/S 30, the application 28, etc.) may be carried on a carrier medium. Generally speaking, a carrier medium may include storage media such as magnetic or optical media, e.g., disk or CD-ROM, volatile or non-volatile memory media such as RAM (e.g. SDRAM, RDRAM, SRAM, etc.), ROM, etc. Any of the previous media and/or any other physical media readable by a computer may comprise computer readable media. A carrier medium may further comprise ~~, as well as~~ transmission media or signals such as electrical, electromagnetic, or digital signals, conveyed via a communication medium such as a network and/or a wireless link.

REMARKS

Claims 1-30 remain pending. In the present Office Action, claims 1-30 were rejected under 35 U.S.C. § 103(a) as being unpatentable over Holiday, U.S. Patent No. 6,421,739 ("Holiday") in view of Oyamada et al., U.S. Patent No. 6,802,062 ("Oyamada"). Claims 1, 12, and 23 were also provisionally rejected under the judicially created doctrine of obviousness type double patenting over three co-pending applications. Applicant respectfully traverses these rejections and requests reconsideration.

Section 103 Rejection

Applicant respectfully submits that each of claims 1-30 recites a combination of features not taught or suggested in Holiday and Oyamada. For example, claim 1 recites a combination of features including: "capture a state of a first virtual machine executing on a first computer system… wherein the first virtual machine comprises at least one virtual disk storing at least one file used by at least one application executing in the first virtual machine, and wherein the state of the first virtual machine comprises the at least one file."

Holiday's Data Objects Do Not Teach or Suggest a File

The present Office Action alleges that Holiday's data objects (e.g. Holiday, col. 3, lines 52-62) teach the at least one file described above (see Office Action, page 5, lines 1-4). Applicant respectfully disagrees.

Holiday's objects are data stored in the heap memory 32 allocated to the JVM (see, e.g., Fig. 1). The objects are deleted via a garbage collection function separate from the application if they are no longer referenced by the running application (see, e.g., Holiday, col. 3, lines 10-13). Additionally, Holiday teaches "The software application program residing in the memories 32a and 42a preferably uses event-driven 'run-to-completion' models of processing, wherein, once an event is received, it is processed to completion without interruption from other threads or processes in the JVM, and a response is generated as appropriate. The point of receipt of such an event and the point of completion of processing of such an event define two points in time *between which*

points the application program *adds, modifies, and/or discards data objects* stored in the heap memories 32 and 42, and thereby changes the state of the program. (Holiday, col. 3, lines 39-49). Furthermore, Holiday teaches that all data objects related to a transaction are discarded at the end of the transaction (Holiday, Fig. 2, element 220 and col. 5, lines 39-42). Thus, data objects are created in heap memory temporarily for processing a transaction, and then discarded when the transaction is complete. On the other hand, a file is often stored in a non-volatile memory and may exist after terminating execution of the application that uses the file. These data objects of Holiday's do not teach or suggest "at least one virtual disk storing at least one file used by at least one application executing in the first virtual machine" as recited in claim 1.

Previous Office Actions have further alleged that "the fact remains that [Holiday's data objects] are blocks of data, which is exactly what files or disks are". Applicant does not disagree that both Holiday's data objects and a file may comprise blocks of data. However, while files and data objects may both comprise blocks of data, they also have other attributes and/or characteristics which are not the same. For example, the manner in which an application in Holiday's disclosure adds, modifies, and discards data objects in the heap memory during the life of a transaction is not the same as the manner in which an application interacts with a file. A file is typically opened, closed, read, and written using a predefined API provided by an operating system. It appears that the Office Actions in the present application accord no meaning to the term "file" other than "block of data", which is not a reasonable interpretation of the term "file". Applicant notes that the broadest reasonable interpretation of a claim term must be consistent with the interpretation that one of skill in the art would reach (see MPEP 2111). In effect, the Office Action appears to identify two species of blocks of data (a file and Holiday's data object), and attempts to anticipate one species with the other. **Anticipation requires fairly strict identity (see MPEP 2131).** While different terminology may be used, it must be clear that the terms have identical meaning. As explained above, files and data objects in memory do not have identical meanings. Data objects in memory do not teach or suggest files.

Applicant notes that the present Office Action cites patent 6,542,909 (Tamer et al., herein "Tamer") as describing objects as files. Tamer is not used in a rejection, but Applicant notes that Tamer, like any other patentee, is free to define a term in anyway he/she likes. Tamer's definition cannot be used to overcome Holiday's clear description of an object as temporary, heap-resident data and not a file, as explained above. Furthermore, Tamer is concerned with file systems and describes objects in the file system, which has nothing to do with Holiday's JVM.

Holiday and Oyamada Do Not Teach or Suggest Copying Features

Furthermore, claim 1 recites "copy at least a portion of the state to a destination separate from a storage device to which the first virtual machine is suspendable, wherein suspending the first virtual machine is performed responsive to a suspend command". Holiday does not teach or suggest the above highlighted features. The present Office Action suggests that the suspend command is inherent because a computer must be told what to do. However, given Holiday's repeated discussions of a failure of the JVM as causing an application to move to another machine, Applicant respectfully traverses this assertion. A computer is not told to fail. Failure occurs due to error, either in the software or in the hardware on which the software is executing. Previous Office Actions have also noted the phrase "or otherwise becomes unavailable" in col. 6, line 62 of Holiday as allegedly supporting the inherency of a command. Applicant respectfully disagrees. Holiday's focus is on fault-tolerance (see, e.g., the title) and recovering from the failure of a JVM (see, e.g., the abstract). There is no evidence that the phrase "or otherwise becomes unavailable" is intended to indicate that a command is somehow involved. A JVM may become unavailable, e.g., due to the failure of a network to which the computer executing the JVM is coupled, but that would still not indicate a command telling the computer to cause the JVM to fail.

Additionally, the present Office Action alleges that Oyamada teaches a suspend command in col. 8, lines 28-45. Applicant respectfully disagrees. Oyamada teaches "In the case where the judgment is that a VM of an identical configuration cannot be generated, in contrast, the virtual machine system 22 gives a response notifying the VM

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.