DYNAMIC LOAD-BASED MERGING

Related Application

[0001] Under 35 U.S.C. 119, this application claims the benefit and right of priority of Chinese Patent Application No. 201410429616.0, filed 27 August 2014.

BACKGROUND

Field

[0002] The present application relates to the field of computer technologies, and relates generally to a method and system for data management.

Related Art

[0003] With the advancement of the computer and network technologies, various operations performed by the users from different applications lead to extensive storage allocation. For example, multiple users may concurrently shop on an e-commerce website via different applications, such as mobile applications running on different platforms as well as web-interfaces running on different browsers in different operating systems. The corresponding shopping records are stored in storage devices (e.g., in a database) of a system facilitating the e-commerce service.

[0004] However, different applications concurrently accessing the system can cause an increasing number of user activities, consequently, an increasing amount of user records are being stored in the system. A large number these records can be accessed by users while they are being updated. Hence, the stored data in the system needs to be continuously managed based on the access and update frequencies of the users. This management of stored data includes allocation of write operations to the data storage such a way that its impact is reduced to users' data access.

[0005] Although a number of methods are available for data management in a system, some problems still remain in efficient storage management for concurrent applications.

SUMMARY

[0005a] The invention is defined in the independent claims. Further embodiments of the invention are defined in the dependent claims.

[0006] One embodiment disclosed herein may provide a system for facilitating dynamic load-based merging. During operation, the system identifies incremental data in

BRIEF DESCRIPTION OF THE FIGURES

[0016] The accompanying drawings herein, which are incorporated herein and constitute a part of the specification, illustrate several exemplary embodiments of the present application and together with the description, serve to illustrate the present application, construing no limitation to the present application. In the drawings:

[0017] FIG. 1A illustrates exemplary data management based on incremental and reference data, in accordance with an embodiment of the present application.

[0018] FIG. 1B illustrates an exemplary data management system based on update and storage servers, in accordance with an embodiment of the present application.

[0019] FIG. 2 illustrates exemplary tablets of the reference data, in accordance with an embodiment of the present application.

[0020] FIG. 3A presents a flowchart illustrating a data management system initiating the merging process, in accordance with an embodiment of the present application.

[0021] FIG. 3B presents a flowchart illustrating the off-peak merging process of a data management system, in accordance with an embodiment of the present application.

[0022] FIG. 3C presents a flowchart illustrating the dynamically-controlled merging process of a data management system, in accordance with an embodiment of the present application.

[0023] FIG. 4 is a schematic structural diagram of an exemplary data management system, in accordance with an embodiment of the present application.

[0024] In the figures, like reference numerals refer to the same figure elements.

DETAILED DESCRIPTION

[0025] Embodiments disclosed herein may efficiently manage data access by diverting data updates from the storage servers being accessed. If data access is unavoidable from those storage servers, data from the storage server is updated when the resources at the storage servers are available.

[0026] With existing technologies, data is typically stored in a Log-Structured Merge-Tree (an LSM Tree). In an LSM tree, data are generally divided into multiple levels in memory and storage disk of a system. If data are updated (e.g., due to a user operation), that data are written into the memory of the system. However, data in the memory is volatile. Hence, the system writes the updated data of the memory to a disk file via a merging process. This merging process is usually performed in parallel with data access (i.e., data reading). The merging

process is batch sequential disk writes. However, such merging process typically consumes a large quantity of input/output (I/O) resources, and directly affects the throughput and response time of the regular operations (e.g., online services) of the storage system. An online service is a service a system keeps operational. For example, availability and accessibility of a user's purchase records.

[0027] Embodiments disclosed herein may provide a system for dynamic load-based merging process by separating data access and update operations, and automatically diverting data access traffic from storage servers that are being currently updated. The system divides data into two sets of data, reference data and incremental data, for data update (e.g., the write operations) and data access (e.g., the read operations), respectively. Incremental data reside in the memory of an update server, and update operations are performed on the incremental data. On the other hand, reference data are stored in disks of storage servers. Reference data usually have multiple copies in multiple storage servers to facilitate high availability. When the amount of data written into the memory of the update server reaches a threshold, the merging process is triggered. The incremental data in the memory is merged to the reference data on the disk of a corresponding storage server.

[0028] Furthermore, to ensure separation of data access from the merging process, the system maintains at least two server groups of storage servers, each set maintaining the entire reference data. When the merging process is performed for a first group, the data access requests are served from a second group. Upon completion of the merging process for the first group, the data access requests are diverted to the first group and the merging process is performed for the second group. As a result, during the merging process, the performance of the data access from the storage servers is not hindered. In some circumstances, due to high volume of data access traffic or a timeout event, some data access requests may be served from a group currently being merged. Under such circumstances, to reduce impact of the merging process, the system monitors the load of the storage servers, and allows the merging process to continue only when the load is low. Otherwise, the system initiates an overload prevention mode and pauses the merging process to recover the resources (e.g., network, processing, and I/O resources) allocated for the merging process.

[0029] FIG. 1A illustrates exemplary data management based on incremental and reference data, in accordance with an embodiment of the present application. In this example, a data management system 100 for dynamic load-based data merging is deployed to facilitate separation of data access and update operations. Suppose that the data includes incremental data 122, which resides in memory 112 of an update server 102. In some embodiments, incremental data 122 can be organized in a B-tree. On the other hand, reference data 124 reside on disks 114

What Is Claimed Is:

1. A computer-implemented method for facilitating dynamic load-based merging, the method comprising:

identifying incremental data in a memory of a first computing device, wherein the incremental data comprises updated data associated with reference data in a storage device in a second computing device;

in response to a size of the incremental data being greater than a threshold, selecting a first server group from a plurality of server groups, wherein the second computing device belongs to the first server group;

migrating data access requests for the reference data from the first server group to a second server group in the plurality of server groups; and

merging the incremental data from the memory of the first computing device with the reference data in the storage device in the second computing device, wherein the merging includes batch sequential disk writes to the storage device of the second computing device.

2. The method of claim 1, wherein the reference data is divided into a plurality of tablets, wherein a tablet is a non-overlapping segment of the reference data.

3. The method of claim 2, wherein a respective server group of the plurality of server groups includes a copy of a respective tablet of the reference data.

4. The method of claim 1, wherein selecting the first server group from the plurality of server groups includes identifying the first server group from a sequence of server groups.

5. The method of claim 4, further comprising:

determining whether the merging of the incremental data have been performed for a respective server group; and

selecting, based on the sequence of server groups, a server group for which the merging of the incremental data has not been performed.

6. The method of claim 1, further comprising:

maintaining a traffic control table comprising a data access request traffic ratio of a respective server group of the plurality of server groups;

determining whether the data access request traffic ratio for the first server group has reached a threshold value in the traffic control table; and

in response to the determination, determining the first server group to be ready for the merging.

7. The method of claim 1, further comprising:

determining whether the merging is complete for the first server group; and

in response to the determination, restoring the data access requests for the reference data from the second server group back to the first server group.

8. The method of claim 1, further comprising:

determining whether the first server group is needed to serve the data access requests for the reference data during the merging of the incremental data; and

determining whether resource consumption by computing devices in the first server group satisfies a condition.

9. The method of claim 8, further comprising:

determining performance level associated with the data access requests for the reference data served from the first server group; and

in response to the performance level being below a threshold, pausing the merging of the incremental data.

10. The method of claim 9, further comprising releasing resources occupied by the merging of the incremental data for serving the data access requests for the reference data.

11. A non-transitory storage medium storing instructions, which when executed by a processor cause the processor to perform a method for facilitating dynamic load-based merging, the method comprising:

identifying incremental data in a memory of a first computing device, wherein the incremental data comprises updated data associated with reference data in a storage device in a second computing device;

in response to a size of the incremental data being greater than a threshold, selecting a first server group from a plurality of server groups, wherein the second computing device belongs to the first server group;

migrating data access requests for the reference data from the first server group to a second server group in the plurality of server groups; and

merging the incremental data from the memory of the first computing device with the reference data in the storage device in the second computing device, wherein the merging includes batch sequential disk writes to the storage device of the second computing device.

12. The non-transitory storage medium of claim 11, wherein the reference data is divided into a plurality of tablets, wherein a tablet is a non-overlapping segment of the reference data.

13. The non-transitory storage medium of claim 12, wherein a respective server group of the plurality of server groups includes a copy of a respective tablet of the reference data.

14. The non-transitory storage medium of claim 11, wherein selecting the first server group from the plurality of server groups includes identifying the first server group from a sequence of server groups.

15. The non-transitory storage medium of claim 14, wherein the method further comprises:

determining whether the merging of the incremental data have been performed for a respective server group; and

selecting, based on the sequence of server groups, a server group for which the merging of the incremental data has not been performed.

16. The non-transitory storage medium of claim 11, wherein the method further comprises:

maintaining a traffic control table comprising a data access request traffic ratio of a respective server group of the plurality of server groups;

determining whether the data access request traffic ratio for the first server group has reached a threshold value in the traffic control table; and

in response to the determination, determining the first server group to be ready for the merging.

17. The non-transitory storage medium of claim 11, wherein the method further comprises:

determining whether the merging is complete for the first server group; and

in response to the determination, restoring the data access requests for the reference data from the second server group back to the first server group.

18. The non-transitory storage medium of claim 11, wherein the method further comprises:

determining whether the first server group is needed to serve the data access requests for the reference data during the merging of the incremental data; and

determining whether resource consumption by computing devices in the first server group satisfies a condition.

19. The non-transitory storage medium of claim 18, wherein the method further comprises:

determining performance level associated with the data access requests for the reference data served from the first server group; and

in response to the performance level being below a threshold, pausing the merging of the incremental data.

20. The non-transitory storage medium of claim 19, wherein the method further comprises releasing resources occupied by the merging of the incremental data for serving the data access requests for the reference data.

21. A computing system for facilitating dynamic load-based merging, the computing system comprising:

a processor;

a memory coupled to the processor and storing instructions, which when executed by the processor cause the processor to perform a method, the method comprising:

identifying incremental data in a memory of a first computing device, wherein the incremental data comprises updated data associated with reference data in a storage device in a second computing device;

in response to a size of the incremental data being greater than a threshold, selecting a first server group from a plurality of server groups, wherein the second computing device

belongs to the first server group;

migrating data access requests for the reference data from the first server group to a second server group in the plurality of server groups; and

merging the incremental data from the memory of the first computing device with the reference data in the storage device in the second computing device, wherein the merging includes batch sequential disk writes to the storage device of the second computing device.

22. A first computing device for facilitating dynamic load-based merging, the first computing device comprising:

a processor;

a memory coupled to the processor and storing instructions, which when executed by the processor, cause the first computing device:

to identify incremental data in the memory of the first computing device, wherein the incremental data comprises updated data associated with reference data in a storage device in a second computing device;

in response to a size of the incremental data being greater than a threshold, to select a first server group from a plurality of server groups, wherein the second computing device belongs to the first server group; and

in response to data access requests for the reference data migrating from the first server group to a second server group in the plurality of server groups, to send the incremental data from the memory of the first computing device to the second computing device for merging with the reference data in the storage device in the second computing device, for the merging to include batch sequential disk writes to the storage device of the second computing device.