

CLAIMS

1. A system, comprising:
one or more computing devices configured to:
generate, at a metadata subsystem of a distributed multi-tenant file storage service, a directed acyclic graph (DAG) to represent a namespace of a directory, wherein each node of a plurality of nodes of the DAG comprises one of: (a) an array of node identifiers (NIArray) of a lower level of the DAG or (b) a list of namespace entries, and wherein at least one node of the DAG is stored at a different storage device than another node of the DAG;
in response to a lookup request for a file store object,
apply a hash function to a name of the file store object to obtain a particular hash value expressible as a first bit sequence of a selected length;
navigate, using successive subsequences of the first bit sequence to identify DAG nodes to be examined at each level, a plurality of levels of the DAG starting from a root level of the DAG, to identify a target node of the DAG that includes a namespace entry of the file store object; and
utilize contents of the namespace entry to determine one or more attributes of the file store object; and
in response to a request to create a new file store object with a specified name,
apply the hash function to the specified name to obtain another hash value expressible as a second bit sequence of the selected length;
navigate, using successive subsequences of the second bit sequence, a plurality of levels of the DAG to identify a candidate node of the DAG for storing a new entry

corresponding to the new file store object, wherein the candidate node comprises a particular list of entries; and
in response to a determination that the candidate node meets a split criterion, distribute, using respective bit sequences obtained by applying the hash function for each entry, the new entry and at least a selected subset of entries of the particular list of entries among a plurality of DAG nodes including at least one new DAG node.

2. The system as recited in claim 1, wherein the one or more computing devices are further configured to:

in response to a determination that the candidate node identified for the new entry meets a first split criterion of a plurality of split criteria,
compute respective bit sequences representing hash values for each entry of the particular list of the candidate node;
assign, using a selected subsequence of each of the respective bit sequences, corresponding entries of the particular list to a plurality of new lists;
assign, using a selected subsequence of the second bit sequence, the new entry to one list of the plurality of new lists;
store each list of the plurality of new lists in a respective new node of the DAG; and
replace the candidate node's entry list with an NSArray comprising node identifiers of the respective new nodes.

3. The system as recited in claim 2, wherein the one or more computing devices are further configured to:

in response to a request to create another file store object with a different specified name,
determine that a second candidate node identified for a second new entry corresponding to the different specified name meets a different

split criterion of the plurality of split criteria, wherein the second candidate node comprises a second list of entries,
identify a sub-list of the second list of entries to be redistributed among a new list and the second list;
obtain respective bit sequences of hash values for each entry of the sub-list;
based at least in part on the respective bit sequences of the entries of the sub-list, redistribute the entries of the sub-list and the different entry among the new list and the second list;
store the new list at a new node of the DAG; and
store a node identifier of the new node in a selected subset of NIArray elements of a parent node of the second candidate node.

4. The system as recited in claim 2, wherein the one or more computing devices are further configured to:

determine, based at least in part on a respective configuration parameter, one or more of: (a) the hash function, (b) a minimum length of a bit sequence obtained from the hash function, (c) a length of a particular subsequence of the first bit sequence to be used for traversing a particular level of the DAG, (d) a number of lists to which the entries of the particular list are to be assigned, (e) a number of NIArray elements in which an identifier of a particular new node is to be stored, (f) a threshold population level, or (g) a maximum number of levels of the DAG.

5. The system as recited in claim 1, wherein the DAG does not store a hash value corresponding to at least a subset of entries of the namespace.

6. A method, comprising:

performing, by one or more computing devices:

generating a directed acyclic graph (DAG) to represent a namespace of a directory, wherein each node of a plurality of nodes of the DAG

comprises one of: (a) an array of node identifiers (NIArray) of a lower level of the DAG or (b) a list of namespace entries;

in response to a lookup request for a file store object,

applying a hash function to a name of the file store object to obtain a particular hash value expressible as a first bit sequence;

navigating, using successive subsequences of the first bit sequence to identify DAG nodes to be examined at each level, a plurality of levels of the DAG starting from a root level of the DAG, to identify a target node of the DAG that includes a namespace entry of the file store object; and

utilizing contents of the namespace entry to determine one or more attributes of the file store object; and

in response to a request to create a new file store object with a specified name,

applying the hash function to the specified name to obtain another hash value expressible as a second bit sequence;

navigating, using successive subsequences of the second bit sequence, a plurality of levels of the DAG to identify a candidate node of the DAG for storing a new entry corresponding to the new file store object, wherein the candidate node comprises a particular list of entries; and

in response to determining that the candidate node meets a split criterion, distributing the new entry and at least a selected subset of entries of the particular list of entries among a plurality of DAG nodes including at least one new DAG node, wherein said distributing comprises using respective bit sequences obtained by applying the hash function for each distributed entry.

7. The method as recited in claim 6, further comprising performing, by the one or more computing devices:

in response to determining that the candidate node identified for the new entry meets a first split criterion of a plurality of split criteria,
computing respective bit sequences representing hash values for each entry of the particular list of the candidate node;
assigning, using a selected subsequence of each of the respective bit sequences, corresponding entries of the particular list to a plurality of new lists;
assigning, using a selected subsequence of the second bit sequence, the new entry to one list of the plurality of new lists;
storing each list of the plurality of new lists in a respective new node of the DAG; and
replacing the candidate node's entry list with an NIArray comprising node identifiers of the respective new nodes.

8. The method as recited in claim 7, wherein said determining that the candidate node meets the first split criterion comprises determining that (a) the population of particular list exceeds a threshold population level and (b) a parent node of the candidate node comprises exactly one NIArray element with a node identifier of the candidate node.

9. The method as recited in claim 7, further comprising performing, by the one or more computing devices:

determining, based at least in part on a respective configuration parameter, one or more of: (a) the hash function, (b) a minimum length of a bit sequence obtained from the hash function, (c) a length of a particular subsequence of the first bit sequence to be used for traversing a particular level of the DAG, (d) a number of lists to which the entries of the particular list are to be assigned, (e) a number of NIArray elements in which an identifier of a particular new node is to be stored, (f) the threshold population level, or (g) a maximum number of levels of the DAG.

10. The method as recited in claim 7, further comprising performing, by the one or more computing devices:

in response to a request to create another file store object with a different specified name,

determining that a second candidate node identified for a second new entry corresponding to the different specified name meets a different split criterion of the plurality of split criteria, wherein the second candidate node comprises a second list of entries,

identifying a sub-list of the second list of entries to be redistributed among a new list and the second list;

obtaining respective bit sequences of hash values for each entry of the sub-list;

based at least in part on the respective bit sequences of the entries of the sub-list, redistributing the entries of the sub-list, and the different entry, among the new list and the second list;

storing the new list at a new node of the DAG; and

storing a node identifier of the new node in a selected subset of NIArray elements of a parent node of the second candidate node.

11. The method as recited in claim 10, wherein said determining that the second candidate node meets the different split criterion comprises determining that (a) the population of second list exceeds a threshold population level and (b) a parent node of the second candidate node comprises more than one NIArray element with a node identifier of the second candidate node.

12. The method as recited in claim 10, wherein the selected subset of NIArray elements of the parent node of the second candidate comprises a subset of NIArray elements that previously included a node identifier of the second candidate.

13. The method as recited in claim 10, further comprising performing, by the one or more computing devices:

combining, within an atomic write operation, (a) said storing the new list at the new node and (b) said storing the node identifier of the new node in the selected subset of NIArray elements of the parent node of the second candidate.

14. The method as recited in claim 6, further comprising performing, by the one or more computing devices:

- receiving a request to delete a particular entry from the namespace;
- determining that the particular entry is the sole entry within an entry list of a different node of the DAG;
- determining that the parent node of the different node comprises a particular NIArray element storing an identifier of the different node, and an adjacent second NIArray element storing an identifier of another node;
- storing the identifier of the other node in the particular NIArray element; and
- deleting the different node from the DAG.

15. The method as recited in claim 6, further comprising performing, by the one or more computing devices:

- receiving a request to delete a particular entry from the namespace;
- determining that the particular entry is the sole entry within an entry list of a different node of the DAG;
- determining that the parent node of the different node does not comprise an NIArray element storing an identifier of any other node than the different node;
- initializing an empty entry list in the parent node of the different node; and
- deleting the different node from the DAG.

16. The method as recited in claim 6, further comprising performing, by the one or more computing devices:

in response to determining that (a) a number of levels in the DAG has reached a maximum permissible level and (b) an entry list of a different candidate node identified for insertion of a different entry is full,
initializing an additional DAG node to be added to an overflow chain of the DAG, wherein a sequential scan is to be used to identify entries inserted into the overflow chain;
inserting a pointer to the additional DAG node within the different candidate node; and
inserting the different entry into the additional DAG node.

17. A non-transitory computer-accessible storage medium storing program instructions that when executed on one or more processors:

generate a directed acyclic graph (DAG) to represent a namespace of a directory, wherein each node of a plurality of nodes of the DAG comprises one of:
(a) an array of node identifiers (NIArray) of a lower level of the DAG or
(b) a list of namespace entries;

in response to a request to create a new file store object with a specified name,

apply a hash function to the specified name to obtain a hash value expressible as a first bit sequence;

navigate, a plurality of levels of the DAG to identify a candidate node of the DAG for storing a new entry corresponding to the new file store object, wherein the candidate node comprises a particular list of entries, and wherein, to navigate a particular level of the plurality of levels, a selected subsequence of the bit sequence is used as an index; and

in response to a determination that the candidate node meets a split criterion, distribute the new entry and at least a selected subset of entries of the particular list of entries among a plurality of DAG nodes including at least one new DAG node using respective bit sequences obtained by applying the hash function for each distributed entry.

18. The non-transitory computer-accessible storage medium as recited in claim 17, wherein the instructions when executed on the one or more processors:

in response to a determination that the candidate node identified for the new entry meets a first split criterion of a plurality of split criteria,
compute respective bit sequences representing hash values for each entry of the particular list of the candidate node;
assign, using a selected subsequence of each of the respective bit sequences, corresponding entries of the particular list to a plurality of new lists;
assign, using a selected subsequence of the second bit sequence, the new entry to one list of the plurality of new lists;
store each list of the plurality of new lists in a respective new node of the DAG; and
replace the candidate node's entry list with an NIArray comprising node identifiers of the respective new nodes.

19. The non-transitory computer-accessible storage medium as recited in claim 18, wherein to determining that the candidate node meets the first split criterion, the instructions when executed on the one or more processors:

determine that (a) the population of particular list exceeds a threshold population level and (b) a parent node of the candidate node comprises exactly one NIArray element with a node identifier of the candidate node.

20. The non-transitory computer-accessible storage medium as recited in claim 18, wherein the instructions when executed on the one or more processors:

determine, based at least in part on a respective configuration parameter, one or more of: (a) the hash function, (b) a minimum length of a bit sequence obtained from the hash function, (c) a length of a particular subsequence of the first bit sequence to be used for traversing a particular level of the DAG, (d) a number of lists to which the entries of the particular list are to

be assigned, (e) a number of NIArray elements in which an identifier of a particular new node is to be stored, (f) the threshold population level, or (g) a maximum number of levels of the DAG.

21. The non-transitory computer-accessible storage medium as recited in claim 18, wherein the instructions when executed on the one or more processors:

in response to a request to create another file store object with a different specified name,

determine that a second candidate node identified for a second new entry corresponding to the different specified name meets a different split criterion of the plurality of split criteria, wherein the second candidate node comprises a second list of entries,

identify a sub-list of the second list of entries to be redistributed among a new list and the second list;

obtain respective bit sequences of hash values for each entry of the sub-list;

based at least in part on the respective bit sequences of the entries of the sub-list, redistribute the entries of the sub-list, and the different entry, among the new list and the second list;

store the new list at a new node of the DAG; and

store a node identifier of the new node in a selected subset of NIArray elements of a parent node of the second candidate node.

22. The non-transitory computer-accessible storage medium as recited in claim 21, wherein to determine that the second candidate node meets the different split criterion, the instructions when executed on the one or more processors:

determine that (a) the population of second list exceeds a threshold population level and (b) a parent node of the second candidate node comprises more than one NIArray element with a node identifier of the second candidate node.