

git_comments:

1. * Licensed to the Apache Software Foundation (ASF) under one * or more contributor license agreements. See the NOTICE file * distributed with this work for additional information * regarding copyright ownership. The ASF licenses this file * to you under the Apache License, Version 2.0 (the * "License"); you may not use this file except in compliance * with the License. You may obtain a copy of the License at * * <http://www.apache.org/licenses/LICENSE-2.0> * * Unless required by applicable law or agreed to in writing, software * distributed under the License is distributed on an "AS IS" BASIS, * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. * See the License for the specific language governing permissions and * limitations under the License.
2. * * Generate the old string for the old test cases. * @return the expression as a string using the leaf ids
3. * * Get the id of the leaf. * The ids are assigned sequentially from 0. * @return the offset in the list returned from {@link SearchArgument#getLeaves}
4. * * Get the expression tree without the normalization to conjunctive normal * form. It will still have the negations pushed to the leaves, but it * won't have the potentially exponential expansion.
5. * * Generate the backwards compatible string for test cases * @return the sarg using the old string

git_commits:

1. **summary:** HIVE-24458: Add a method to provide a non-normalized ExpressionTree (#1726) (Owen O'Malley reviewed by Panos Garefalakis, Zoltan Haindrich)
message: HIVE-24458: Add a method to provide a non-normalized ExpressionTree (#1726) (Owen O'Malley reviewed by Panos Garefalakis, Zoltan Haindrich)

github_issues:

github_issues_comments:

github_pulls:

1. **title:** HIVE-24458: Add a method to provide a non-normalized ExpressionTree
body: This pull request provides a method to get a non-normalized copy of the SArg expression. This can be useful for applications where the smaller non-normalized expression is useful. I've changed the SArg ExpressionTree leaves to keep a pointer to the PredicateLeafImpl rather than just the id. That makes a lot of things easier, including dealing with keeping the normalized and non-normalized versions in sync with the leaves. It also changes the format of the toString to be easier to read, but to make the old tests easier to port I've provided a toOldString method that generates the old string.
2. **title:** HIVE-24458: Add a method to provide a non-normalized ExpressionTree
body: This pull request provides a method to get a non-normalized copy of the SArg expression. This can be useful for applications where the smaller non-normalized expression is useful. I've changed the SArg ExpressionTree leaves to keep a pointer to the PredicateLeafImpl rather than just the id. That makes a lot of things easier, including dealing with keeping the normalized and non-normalized versions in sync with the leaves. It also changes the format of the toString to be easier to read, but to make the old tests easier to port I've provided a toOldString method that generates the old string.
label: code-design
3. **title:** HIVE-24458: Add a method to provide a non-normalized ExpressionTree
body: This pull request provides a method to get a non-normalized copy of the SArg expression. This can be useful for applications where the smaller non-normalized expression is useful. I've changed the SArg ExpressionTree leaves to keep a pointer to the PredicateLeafImpl rather than just the id. That makes a lot of things easier, including dealing with keeping the normalized and non-normalized versions in sync with the leaves. It also changes the format of the toString to be easier to read, but to make the old tests easier to port I've provided a toOldString method that generates the old string.
4. **title:** HIVE-24458: Add a method to provide a non-normalized ExpressionTree
body: This pull request provides a method to get a non-normalized copy of the SArg expression. This can be useful for applications where the smaller non-normalized expression is useful. I've changed the SArg ExpressionTree leaves to keep a pointer to the PredicateLeafImpl rather than just the id. That makes a lot of things easier, including dealing with keeping the normalized and non-normalized versions in sync with

the leaves. It also changes the format of the toString to be easier to read, but to make the old tests easier to port I've provided a toOldString method that generates the old string.

13. **title:** HIVE-24458: Add a method to provide a non-normalized ExpressionTree

body: This pull request provides a method to get a non-normalized copy of the SArg expression. This can be useful for applications where the smaller non-normalized expression is useful. I've changed the SArg ExpressionTree leaves to keep a pointer to the PredicateLeafImpl rather than just the id. That makes a lot of things easier, including dealing with keeping the normalized and non-normalized versions in sync with the leaves. It also changes the format of the toString to be easier to read, but to make the old tests easier to port I've provided a toOldString method that generates the old string.

14. **title:** HIVE-24458: Add a method to provide a non-normalized ExpressionTree

body: This pull request provides a method to get a non-normalized copy of the SArg expression. This can be useful for applications where the smaller non-normalized expression is useful. I've changed the SArg ExpressionTree leaves to keep a pointer to the PredicateLeafImpl rather than just the id. That makes a lot of things easier, including dealing with keeping the normalized and non-normalized versions in sync with the leaves. It also changes the format of the toString to be easier to read, but to make the old tests easier to port I've provided a toOldString method that generates the old string.

15. **title:** HIVE-24458: Add a method to provide a non-normalized ExpressionTree

body: This pull request provides a method to get a non-normalized copy of the SArg expression. This can be useful for applications where the smaller non-normalized expression is useful. I've changed the SArg ExpressionTree leaves to keep a pointer to the PredicateLeafImpl rather than just the id. That makes a lot of things easier, including dealing with keeping the normalized and non-normalized versions in sync with the leaves. It also changes the format of the toString to be easier to read, but to make the old tests easier to port I've provided a toOldString method that generates the old string.

label: code-design

16. **title:** HIVE-24458: Add a method to provide a non-normalized ExpressionTree

body: This pull request provides a method to get a non-normalized copy of the SArg expression. This can be useful for applications where the smaller non-normalized expression is useful. I've changed the SArg ExpressionTree leaves to keep a pointer to the PredicateLeafImpl rather than just the id. That makes a lot of things easier, including dealing with keeping the normalized and non-normalized versions in sync with the leaves. It also changes the format of the toString to be easier to read, but to make the old tests easier to port I've provided a toOldString method that generates the old string.

github_pulls_comments:

1. @pgaref I think we are ok with two trees, because we were already keeping the larger tree in memory. Adding the compact one should be ok.

github_pulls_reviews:

1. unfinished senece
2. I find `unexpanded` a little strange - a different keyword might fit this thing better (what do you think about compact or original).
3. **body:** Unused variable
label: code-design
4. This could be changed to PredicateLeaf -- see below
5. **body:** Nit: maybe we should also explicitly mention in `**getExpression**` doc that it normalizes the expression
label: documentation
6. maybe rename vars to compact and normalized ExpressionTree to avoid confusion?
7. Apache License?
8. **body:** I believe `**Id**` setter and getter could be moved to the `**PredicateLeaf**` interface. It would be cleaner and as a result `**addLeaf**` would need less modifications.
label: code-design
9. This can now become PredicateLeaf
10. Argument may change to PredicateLeaf?
11. same
12. same on all alterations below..
13. **body:** I made getId more visible, but left setId as private. setId only makes sense in the context of the builder.

label: code-design

jira_issues:

1. **summary:** Allow access to SArgs without converting to disjunctive normal form
description: For some use cases, it is useful to have access to the SArg expression in a non-normalized form. Currently, the SArg only provides the fully normalized expression.
2. **summary:** Allow access to SArgs without converting to disjunctive normal form
description: For some use cases, it is useful to have access to the SArg expression in a non-normalized form. Currently, the SArg only provides the fully normalized expression.
3. **summary:** Allow access to SArgs without converting to disjunctive normal form
description: For some use cases, it is useful to have access to the SArg expression in a non-normalized form. Currently, the SArg only provides the fully normalized expression.

jira_issues_comments:

1. merged into master. Thank you Owen for extending this!