

TAJO-87

hyunsik

2013-07-18

Contents

Chapter 1

Root issue TAJO-87

1.1 Summary

Integration of tajo algebra module and SQL parser.

1.2 Description

In the current implementation, QueryAnalyzer transforms a sql statement into a data structure that represents a parser tree and a query block. They has the following limitations: * They can't support multiple block queries including table and scalar subqueries. * They are tightly coupled to a certain grammar.

Before ASF incubation, I have developed tajo-algebra and tajo-frontend-sql. Tajo-algebra is a kind of intermediate layer. It represents a just relational algebraic expression. With this, we can do more interesting things. For example, we can enable users to describe a logical plan for a certain query. It also very helpful to support an another DSL.

For this work, tajo-algebra should be improved to support full specifications, and we should rewrite a SQL parser to use tajo-algebra.

1.3 Attachments

1. [TAJO-87.patch](#)

1.4 Commits

1. Commit **2e9d788** by **Hyunsik Choi** (2013-07-29): TAJO-87: Integration of tajo algebra module and SQL parser. (fixed tajo-dist/pom.xml)
2. Commit **9e98200** by **Hyunsik Choi** (2013-07-29): TAJO-87: Integration of tajo algebra module and SQL parser. (fixed tajo-project/pom.xml)
3. Commit **00c3ee2** by **Hyunsik Choi** (2013-07-26): TAJO-87: Integration of tajo algebra module and SQL parser

1.5 Comments

1. **hyunsik:** The patch does as follows:
 - * implement some missed relational algebraic expression classes, such as InPredicate, Case-WhenPredicate, Having, and NotExpr
 - * remove tajo-front module
 - * completely rewrite SQLAnalyzer
 - * change SQLParser.g and SQLLexer.g using antlr3 into ones using Antlr4
 - * rewrite LogicalPlanner
 - * This change is the initial work for cost-based optimization.
 - * rewrite LogicalOptimizer
 - * remove some unused classes

I've also uploaded the patch to the reviewboard.
<https://reviews.apache.org/r/12964/>

I verified 'mvn clean install'.

2. **jhkim:** +1 Looks great.

Hyunsik,
The error handling code in the new parser is omitted. If you are ok, I'll move the error handling code from the old parser to new parser.

3. **hyunsik:** Jinho,

I know your elegant error handling code in the old parser. But, the API of Antlr4 is incompatible to that of Antlr3. Actually, I need more time in order to implement the same feature because I have to survey Antlr4's error handling. If you are willing to port the error handling code from the old parser to the new parser, I really appreciate you.

4. **hyunsik:** I've committed it. Thank you for the review.

1.6 Pull requests

No pull requests

Chapter 2

Connected issue TAJO-24

2.1 Summary

Cost-based Optimizer for Tajo

2.2 Description

The current query optimizer is too simple and is not extensible. In addition, it does not support scalar and table subqueries.

So, I would like to propose the implementation of new optimizer which is extensible and is based on some cost model.

In my opinion, this work is not trivial and will take a very long time. Also, we will have many discussions on this issue. This is the first draft, and I would like to continue to improve the requirements and details.

2.3 Attachments

1. [Cost-based_Optimizer_130401.pdf](#)
2. [Cost-based_Optimizer_130402.pdf](#)

2.4 Commits

No related commits

2.5 Comments

1. **burakkk:** Hi,
I think if you could split this task to subtasks, it can be done faster in parallel. If so, can you assign me please?

Thanks
Best Regards...

```
--
*BURAK ISIKLI** *| *http://burakisikli.wordpress.com*
*
*
```

2. **hyunsik:** I attach the revised design document that fixes some typo errors and adds some references.

3. **hyunsik:** Burak,

Of course, that's what I want. In this issue, I have a plan to design and develop the interfaces to fully support the proposal. Then, we will be able to create a number of subtasks for rewrite rules and join enumeration algorithms.

However, the design of the interfaces may take a long time. I'll continue to share the progress here.

4. **initialcontext:** Great idea. Lets solidfy the plan here in the thread, and we can post sub-JIRAs here for the sub tasks as the work progresses. I agree this can (and should) take a while so we can do it right.

5. **hyunsik:** I'm fixing this issue because we resolved all sub tasks.

2.6 Pull requests

No pull requests