

**git\_comments:**

**git\_commits:**

1. **summary:** TAJO-691: HashJoin or HashAggregation is too slow if there is many unique keys.  
(hyoungjunkim via hyunsik)  
**message:** TAJO-691: HashJoin or HashAggregation is too slow if there is many unique keys.  
(hyoungjunkim via hyunsik)

**github\_issues:**

**github\_issues\_comments:**

**github\_pulls:**

**github\_pulls\_comments:**

**github\_pulls\_reviews:**

**jira\_issues:**

1. **summary:** HashJoin or HashAggregation is too slow if there is many unique keys  
**description:** HashJoin or HashAggregation is too slow if there is many unique keys. Java's native Map is inefficient to handle many items. In case more than 1 million items in HashMap, Adding 10000 items takes more than 7 ~ 10 seconds. This should be improved.

**jira\_issues\_comments:**

1. I suggest MapDB's LongHashMap. Please check the bellow site. [http://www.mapdb.org/http://kote.net/blog/3G\\_map](http://www.mapdb.org/http://kote.net/blog/3G_map)
2. +1. It looks a good suggestion.
3. +1 for this idea. I've heard some experimental result from him in offline. There will be significant performance gain.
4. Created a review request against branch master in reviewboard
5. HashMap is not the cause of poor performance. VTuple.hashCode() returns a same hash value in case of following. {code} VTuple v1 = new VTuple(new Datum[]{new Int4Datum(1), new Int4Datum(2)}); VTuple v2 = new VTuple(new Datum[]{new Int4Datum(2), new Int4Datum(1)}); System.out.println(v1.hashCode()); System.out.println(v2.hashCode()); {code} This code prints same hashcode. {noformat} 94 94 {noformat}
6. {color:red}\* -1 overall. {color} Here are the results of testing the latest attachment <http://issues.apache.org/jira/secure/attachment/12636085/TAJO-691.patch> against master revision 7283c58. {color:green} +1 @author. {color} The patch does not contain any @author tags. {color:green} +1 tests included. {color} The patch appears to include 6 new or modified test files. {color:green} +1 javac. {color} The applied patch does not increase the total number of javac compiler warnings. {color:green} +1 javadoc. {color} The applied patch does not increase the total number of javadoc warnings. {color:green} +1 checkstyle. {color} The patch generated 0 code style errors. {color:red} -1 findbugs. {color} The patch appears to introduce 183 new Findbugs (version 1.3.9) warnings. {color:green} +1 release audit. {color} The applied patch does not increase the total number of release audit warnings. {color:green} +1 core tests. {color} The patch passed unit tests in tajo-core/tajo-core-backend tajo-rpc tajo-storage. Test results: <https://builds.apache.org/job/PreCommit-TAJO-Build/245/testReport/> Findbugs warnings: <https://builds.apache.org/job/PreCommit-TAJO-Build/245/artifact/incubator-tajo/patchprocess/newPatchFindbugsWarningstajo-core-backend.html> Console output: <https://builds.apache.org/job/PreCommit-TAJO-Build/245/console> This message is automatically generated.
7. +1 for latest patch. Thank you for your contribution. The patch looks good to me. After hashCode of both VTuple and LazyTuple are changed, some non-determined query statements seem to result in different results. From your patch, I'll try to find more unit tests which potentially can cause the same problem. This patch contains more fixes of the cases that I found.

8. {color:red}\* -1 overall.\*{color} Here are the results of testing the latest attachment [http://issues.apache.org/jira/secure/attachment/12637406/TAJO-691\\_2.patch](http://issues.apache.org/jira/secure/attachment/12637406/TAJO-691_2.patch) against master revision 5d94b03. {color:green}+1 @author.{color} The patch does not contain any @author tags. {color:green}+1 tests included.{color} The patch appears to include 15 new or modified test files. {color:green}+1 javac.{color} The applied patch does not increase the total number of javac compiler warnings. {color:green}+1 javadoc.{color} The applied patch does not increase the total number of javadoc warnings. {color:green}+1 checkstyle.{color} The patch generated 0 code style errors. {color:red}-1 findbugs.{color} The patch appears to introduce 197 new Findbugs (version 1.3.9) warnings. {color:green}+1 release audit.{color} The applied patch does not increase the total number of release audit warnings. {color:green}+1 core tests.{color} The patch passed unit tests in tajo-core/tajo-core-backend tajo-rpc tajo-storage. Test results: <https://builds.apache.org/job/PreCommit-TAJO-Build/280//testReport/> Findbugs warnings: <https://builds.apache.org/job/PreCommit-TAJO-Build/280//artifact/incubator-tajo/patchprocess/newPatchFindbugsWarningstajo-core-backend.html> Console output: <https://builds.apache.org/job/PreCommit-TAJO-Build/280//console> This message is automatically generated.
9. committed the latest patch to master and branch-0.8.0. Thanks!
10. SUCCESS: Integrated in Tajo-master-build #145 (See [<https://builds.apache.org/job/Tajo-master-build/145/>]) TAJO-691: HashJoin or HashAggregation is too slow if there is many unique keys. (hyoungjunkim via hyunsik) (hyunsik: rev 36007d779142b48194ff8eab5610db29390ab9d2) \* tajo-core/tajo-core-backend/src/test/resources/queries/TestCaseByCases/testTAJO415Case.sql \* tajo-core/tajo-core-backend/src/test/resources/results/TestNetTypes/testGroupby.result \* tajo-core/tajo-core-backend/src/test/resources/results/TestJoinQuery/testJoinCoReferredEvalsWithSameExprs1.result \* tajo-core/tajo-core-backend/src/test/resources/results/TestCaseByCases/testTAJO415Case.result \* tajo-core/tajo-core-backend/src/test/resources/results/TestNetTypes/testGroupby2.result \* tajo-core/tajo-core-backend/src/test/resources/queries/TestJoinQuery/testJoinCoReferredEvalsWithSameExprs2.sql \* tajo-core/tajo-core-backend/src/test/resources/results/TestBuiltinFunctions/testAvgDouble.result \* tajo-core/tajo-core-backend/src/test/resources/queries/TestBuiltinFunctions/testAvgDouble.sql \* tajo-core/tajo-core-backend/src/test/resources/queries/TestNetTypes/testGroupby2.sql \* tajo-core/tajo-core-backend/src/test/resources/results/TestGroupByQuery/testHavingWithNamedTarget.result \* tajo-storage/src/main/java/org/apache/tajo/storage/LazyTuple.java \* tajo-rpc/src/main/java/org/apache/tajo/rpc/NettyClientBase.java \* tajo-core/tajo-core-backend/src/test/resources/queries/TestJoinQuery/testJoinCoReferredEvalsWithSameExprs1.sql \* tajo-core/tajo-core-backend/src/test/resources/results/TestGroupByQuery/testGroupBy4.result \* tajo-core/tajo-core-backend/src/test/resources/queries/TestGroupByQuery/testGroupBy4.sql \* tajo-core/tajo-core-backend/src/test/resources/queries/TestNetTypes/testGroupby.sql \* CHANGES.txt \* tajo-core/tajo-core-backend/src/test/resources/results/TestBuiltinFunctions/testRandom.result \* tajo-storage/src/main/java/org/apache/tajo/storage/VTuple.java \* tajo-core/tajo-core-backend/src/test/resources/results/TestJoinQuery/testJoinCoReferredEvalsWithSameExprs2.result
11. SUCCESS: Integrated in Tajo-0.8.0-build #48 (See [<https://builds.apache.org/job/Tajo-0.8.0-build/48/>]) TAJO-691: HashJoin or HashAggregation is too slow if there is many unique keys. (hyoungjunkim via hyunsik) (hyunsik: rev ebc60c51e819432fda8a19618cb4ff9323168ddd) \* tajo-core/tajo-core-backend/src/test/resources/results/TestBuiltinFunctions/testRandom.result \* tajo-storage/src/main/java/org/apache/tajo/storage/VTuple.java \* tajo-core/tajo-core-backend/src/test/resources/results/TestBuiltinFunctions/testAvgDouble.result \* tajo-rpc/src/main/java/org/apache/tajo/rpc/NettyClientBase.java \* tajo-core/tajo-core-backend/src/test/resources/queries/TestGroupByQuery/testGroupBy4.sql \* tajo-core/tajo-core-backend/src/test/resources/queries/TestJoinQuery/testJoinCoReferredEvalsWithSameExprs2.sql \* tajo-core/tajo-core-backend/src/test/resources/queries/TestBuiltinFunctions/testAvgDouble.sql \* tajo-core/tajo-core-backend/src/test/resources/results/TestCaseByCases/testTAJO415Case.result \* tajo-core/tajo-core-backend/src/test/resources/results/TestJoinQuery/testJoinCoReferredEvalsWithSameExprs2.result \* tajo-core/tajo-core-backend/src/test/resources/queries/TestNetTypes/testGroupby.sql \* tajo-core/tajo-core-backend/src/test/resources/results/TestGroupByQuery/testHavingWithNamedTarget.result \* tajo-core/tajo-core-backend/src/test/resources/results/TestNetTypes/testGroupby.result \* tajo-core/tajo-core-backend/src/test/resources/results/TestJoinQuery/testJoinCoReferredEvalsWithSameExprs1.result \* tajo-core/tajo-core-backend/src/test/resources/results/TestNetTypes/testGroupby2.result \* tajo-core/tajo-core-backend/src/test/resources/queries/TestJoinQuery/testJoinCoReferredEvalsWithSameExprs1.sql \* tajo-core/tajo-core-backend/src/test/resources/queries/TestNetTypes/testGroupby2.sql \* tajo-storage/src/main/java/org/apache/tajo/storage/LazyTuple.java \* CHANGES.txt \* tajo-core/tajo-core-

backend/src/test/resources/queries/TestCaseByCases/testTAJO415Case.sql \* tajo-core/tajo-core-backend/src/test/resources/results/TestGroupByQuery/testGroupBy4.result