

Re: Trying to optimize a query with a bunch of INNER JOINS

Source:

<http://www.tech-archive.net/Archive/SQL-Server/microsoft.public.sqlserver.programming/2004-10/1559.html>

From: Louis Davidson (*dr_dontspamme_sql_at_hotmail.com*)

Date: 10/07/04

Date: Wed, 6 Oct 2004 19:08:14 -0500

A derived table is something like this:

```
select sum(derivedTable.column) as summedColumn
from ( select column
      from table) as derivedTable
```

It can help to put the meat of the query in the derived table and then do some calculations outside. Not sure it will in your case, just a possibility.

A right outer join is just a left outer join turned around, so that may be what you are seeing.

Can you take out the call to the other database and see if that improves things? It "shouldn't" be a big deal, but every situation is different.

If you can post the query, the plan, approximate table data counts, statistics IO, etc in text, it would be easier to help you. Even table structures would not hurt.

--

Louis Davidson - drsql@hotmail.com
SQL Server MVP
Compass Technology Management - www.compass.net
Pro SQL Server 2000 Database Design -
<http://www.apress.com/book/bookDisplay.html?bID=266>

Note: Please reply to the newsgroups only unless you are interested in consulting services. All other replies may be ignored :)

"Scott Lyon" <scott.RED.lyon.WHITE@rapistan.BLUE.com> wrote in message news:eKM3jq7qEHA.4008@TK2MSFTNGP14.phx.gbl...

> The data isn't that large, but it potentially could be.

>

> Just as a test, I changed the SUM(column) to just return the column itself
> (so rather than summarizing, it's giving every record), while leaving the
> GROUP BY intact, and it's taking just as long as with the sum. In fact, it
> took almost 2 minutes for the query to come back, which is right in-line
> with what it took with the SUM in there. That way, it returned just over
350

microsoft.public.sqlserver.programming: Re: Trying to optimize a query with a bunch of INNER JOINS

```
> rows. With the SUM in place, it returns just under 100.
>
> If I take out the GROUP BY entirely (and still not using the SUM(), it
came
> back in just under a minute (faster than before, but still too slow).
>
>
> What do you mean by a "derived" table?
>
>
> I did an analysis against that query (without the GROUP or SUM), and that
> showed the highest cost on a few "Hash Match/Right Outer Join" operations,
> although my query doesn't have any RIGHT OUTER joins, just a couple of
LEFT
> (OUTER) joins, and some INNER joins.
>
>
> One other thing, is part of the query is LEFT joins to tables that are in
> another database on the server. Not sure how much slowdown is involved
> crossing databases, though.
>
>
> Any ideas?
>
>
> Thanks!
> -Scott
>
>
> "Louis Davidson" <dr_dontspamme_sql@hotmail.com> wrote in message
> news:eZmGce7qEHA.1204@TK2MSFTNGP12.phx.gbl...
> > How large is your set? Also, how fast is it if you drop out the SUM and
> the
> > GROUP BY? You might consider using a derived table, or even a temp
table
> > for this if there are not too many rows in the final output. Is the SUM
> the
> > last step before aggregation? Or is it done early in the query plan to
> > also facilitate a MERGE join?
> >
> > --
>
> -----
> --
> > Louis Davidson - drsql@hotmail.com
> > SQL Server MVP
> >
> > Compass Technology Management - www.compass.net
> > Pro SQL Server 2000 Database Design -
> > http://www.apress.com/book/bookDisplay.html?bID=266
> > Note: Please reply to the newsgroups only unless you are interested in
> > consulting services. All other replies may be ignored :)
> >
> > "Scott Lyon" <scott.RED.lyon.WHITE@rapistan.BLUE.com> wrote in message
> > news:%23XkN1rvqEHA.3700@TK2MSFTNGP15.phx.gbl...
> > > Yes, I do have foreign keys on the main table, as well as primary keys
> > on
> > > everything.
> > >
> > >
> > > One thing I do have (that I'm sure slows things down, but cannot be
> > helped),
```

microsoft.public.sqlserver.programming: Re: Trying to optimize a query with a bunch of INNER JOINS

```
> > > is that my source "table" is actually a handful of tables connected
with
> > > INNER and LEFT JOINS (but everything has primary keys, as well as
> foreign
> > > keys where applicable).
> > >
> > > One other unavoidable thing is that I'm doing a SUM() on one of the
> > fields,
> > > so I've therefore got a GROUP BY statement of everything else too.
> > >
> > >
> > > Looking at the plan, it's saying that the SORT step is taking up 97%
of
> > the
> > > cost. I can only assume this is because of the GROUP BY. But since I
> have
> > > the SUM() in there, I can't exactly leave that part off the query.
> > >
> > >
> > > Any ideas?
> > >
> > >
> > > Thanks!
> > > -Scott
> > >
> > > "Louis Davidson" <dr_dontspamme_sql@hotmail.com> wrote in message
> > > news:uZnbNTvqEHA.348@TK2MSFTNGP15.phx.gbl...
> > > > As Mike says look to the plan. Look at the percentages of time it
> takes
> > > to
> > > > do each of the steps. Also, is there a WHERE clause on you main
> > > statement?
> > > > And do you have primary keys and foreign keys on all of the tables?
> You
> > > may
> > > > need foreign key indexes on your primary table, thought that totally
> > > depends
> > > > on what your where looks like.
> > > >
> > > > Post the plan if you can.
> > > > use northwind
> > > > go
> > > > set showplan_text on
> > > > go
> > > > select * from customers
> > > >
> > > > returns:
> > > >
> > > > StmtText
> > > >
> > > >
> > > >
> > > > -----
> > > > --
> > > > -----
> > > > |--Clustered Index
> > > > Scan(OBJECT:([Northwind].[dbo].[Customers].[PK_Customers]))
> > > >
> > > > showplan_all gives more information but is a pain to post :)
> > > >
> > > > --
> > > >
```

microsoft.public.sqlserver.programming: Re: Trying to optimize a query with a bunch of INNER JOINS

```
> >
>
> -----
> > > --
> > > Louis Davidson - drsql@hotmail.com
> > > SQL Server MVP
> > >
> > > Compass Technology Management - www.compass.net
> > > Pro SQL Server 2000 Database Design -
> > > http://www.apress.com/book/bookDisplay.html?bID=266
> > > Note: Please reply to the newsgroups only unless you are interested
in
> > > consulting services. All other replies may be ignored :)
> > >
> > > "Scott Lyon" <scott.RED.lyon.WHITE@rapistan.BLUE.com> wrote in messa
ge
> > > news:edr%23DMuqEHA.592@TK2MSFTNGP11.phx.gbl...
> > > > Considering my look-up tables are small, would it improve
efficiency
> > > at
> > > > all
> > > > to load those into variables (defined as tables), and join to
those
> > > > variables rather than joins to the actual tables themselves?
> > > >
> > > > What I'm thinking (and I admit I don't know how stored procedures
> work
> > > > "behind the scenes", so this may be a moot point), is that if I
have
> a
> > > > lookup table consisting of only two rows/values, when I join to
it,
> > I'm
> > > > retrieving those two values a BUNCH of times. Wouldn't it be
faster
> if
> > > > those
> > > > two values were already in memory?
> > > >
> > > >
> > > > Thanks!
> > > > -Scott
> > > >
> > > >
> > > > "Dan Guzman" <guzmanda@nospam-online.sbcglobal.net> wrote in
message
> > > > news:%23DkOZ%23tqEHA.1452@TK2MSFTNGP10.phx.gbl...
> > > > > Since you are joining on the primary key of the lookup tables, I
> > would
> > > > > expect your query to perform reasonably well. You might try
> > updating
> > > > > statistics and creating foreign key constraints if you haven't
> > already
> > > > > done
> > > > > so. A non-clustered index on DataStr, Code1_ID, Code2_ID,
> Code3_ID
> > > will
> > > > > cover this particular query but I wouldn't think this would be
> > needed
> > > on
> > > > a
> > > > > table with only a few thousand rows.
```

microsoft.public.sqlserver.programming: Re: Trying to optimize a query with a bunch of INNER JOINS

```
> > > > >
> > > > > --
> > > > > Hope this helps.
> > > > >
> > > > > Dan Guzman
> > > > > SQL Server MVP
> > > > >
> > > > > "Scott Lyon" <scott.RED.lyon.WHITE@rapistan.BLUE.com> wrote in
> > > > > message
> > > > > news:eR6I32tqEHA.1964@TK2MSFTNGP12.phx.gbl...
> > > > > > I've got a table of data that also contains an ID column for a
> > > > > bunch
> > > > > of
> > > > > > look-up tables.
> > > > > >
> > > > > > For example, let's assume I've got 3 look-up tables (each of
> > > > > > which
> > > > > > will
> > > > > > > have
> > > > > > > from 2 to 20 possible values):
> > > > > >
> > > > > > > Code1:
> > > > > > > Code1_ID int (PK)
> > > > > > > Code1_value char(1)
> > > > > > >
> > > > > > > Code2:
> > > > > > > Code2_ID int (PK)
> > > > > > > Code2_value char(1)
> > > > > > >
> > > > > > > Code3:
> > > > > > > Code3_ID int (PK)
> > > > > > > Code3_value char(1)
> > > > > > >
> > > > > > >
> > > > > > > My data table (just a portion of it) is like this:
> > > > > > >
> > > > > > > Data1:
> > > > > > > DataStr char(50)
> > > > > > > Code1_ID int
> > > > > > > Code2_ID int
> > > > > > > Code3_ID int
> > > > > > >
> > > > > > >
> > > > > > > Now, I want to have a query (in a stored procedure) to get all
> > > > > > > records
> > > > > > > in
> > > > > > > > Data1, with the value (not the ID) for each of the lookup
tables
> > > > > > > > associated
> > > > > > > > with it.
> > > > > > > >
> > > > > > > >
> > > > > > > > I know that the most direct way to do this is to join all the
> > > > > > > > tables,
> > > > > > > > like
> > > > > > > > this:
> > > > > > > >
> > > > > > > > SELECT Data1.DataStr, Code1.Code1_value, Code2.Code2_value,
> > > > > > > > Code3.Code3_value
> > > > > > > > FROM Data1
> > > > > > > > INNER JOIN Code1
> > > > > > > > ON Code1.Code1_ID = Data1.Code1_ID
```


microsoft.public.sqlserver.programming: Re: Trying to optimize a query with a bunch of INNER JOINS

> > >
> >
> >
>
>