

# Re: Poor performance after upgrading to sql server 2005

---

*Source:*

<http://www.tech-archive.net/Archive/SQL-Server/microsoft.public.sqlserver.setup/2006-03/msg00057.html>

---

- *From:* "Andrew J. Kelly" <[sqlmvpnoospam@xxxxxxxxxxxxx](mailto:sqlmvpnoospam@xxxxxxxxxxxxx)>
  - *Date:* Fri, 3 Mar 2006 17:35:25 -0500
- 

Didn't mean to imply you just threw it together. I have no way to know your knowledge of such things and wanted to point them out. Most people are not aware of such things especially the table variables. Sorry if I offended you as it was not the intent.

—

Andrew J. Kelly SQL MVP

"Rich Wood" <[RichWood@xxxxxxxxxxxxx](mailto:RichWood@xxxxxxxxxxxxx)> wrote in message [news:2CDB845E-8115-4D9E-8395-354A8A9515BF@xxxxxxxxxxxxx](mailto:news:2CDB845E-8115-4D9E-8395-354A8A9515BF@xxxxxxxxxxxxx)

I do happen to be familiar with the pros and cons of using temp tables vs table variables, when to use variables and when not to, and a few other database concepts as well. We didn't just "throw the code at the optimizer", we spent a lot of time testing our code using alternative methods and eventually we arrived at the fastest solution given our environment. Now we have to do the same thing for SQL 2005?

Rich Wood

"Andrew J. Kelly" wrote:

I know we can change our code to work with SQL 2005, but I don't think there is anything unusual about our database code — subqueries are not uncommon or even discouraged by Microsoft

Re: Poor performance after upgrading to sql server 2005

I don't think it is as simple as a subquery or not. That would be a major problem if the optimizer could not handle subqueries but I know for a fact

it can because of all the other SQL2005 sites that are using them effectively. You have lots of things involved such as UDF's, table variables, joins etc. and they need to determine the ultimate cause.

and table variables are actually encouraged over temporary tables

In some cases they are. They have pros & cons just like temp tables do but they should in no way be a total replacement for temptables. This is especially true the more rows you have in the table var and when you need the updated stats. Of course the stats can also cause recompiles. You should always test both ways to see which works best for that particular scenario.

<http://www.support.microsoft.com/?id=305977>

not to mention using simple data type variables in sql queries.

Again they have their place but there are always exceptions. You need to understand where and when they can affect the resulting plan to ensure yo get the optimal results.

The bottom line is that you can't simply throw anything at hte optimizer and expect it to do a perfect job. Sometimes it needs some help to get it right. Luckily this is the exception not the rule.

—  
Andrew J. Kelly SQL MVP

"Rich Wood" <RichWood@xxxxxxxxxxxxxxxxxx> wrote in message [news:CBC7395F-1C09-46A5-B3E5-EA6852EB7D01@xxxxxxxxxxxxxxxxxx](mailto:news:CBC7395F-1C09-46A5-B3E5-EA6852EB7D01@xxxxxxxxxxxxxxxxxx)

I know we can change our code to work with SQL 2005, but I don't think there is anything unusual about our database code — subqueries

Re: Poor performance after upgrading to sql server 2005

are not  
uncommon  
or  
even discouraged by Microsoft and table variables are  
actually  
encouraged  
over temporary tables -- not to mention using simple data  
type  
variables  
in  
sql queries. I'm sure a lot of people will run into performance  
problems  
like  
ours. This could be a real problem for Microsoft.

Rich Wood

"Andrew J. Kelly" wrote:

<http://www.support.microsoft.com/default.aspx?scid=fh:EN-US:sql>  
SQL

Support

<http://www.mssqlserver.com/faq/general-pss.asp>

MS PSS

They will require a credit card for a charge  
of around \$225.00 US but  
it  
will be refunded if this is a bug.

--

Andrew J. Kelly SQL MVP

"Rich Wood"

<RichWood@xxxxxxxxxxxxxxxxxxx> wrote in  
message

[news:8E66D323-C73B-4440-8272-3317F74F6DF3@xxxxxxxxxxxxxxxxxxx](mailto:news:8E66D323-C73B-4440-8272-3317F74F6DF3@xxxxxxxxxxxxxxxxxxx)

That would be great -- how  
do I open a case with MS  
PSS? Thanks  
again  
for  
your suggestions.

Rich Wood

"Andrew J. Kelly" wrote:

Re: Poor performance after upgrading to sql server 2005

Are  
you  
saying  
that  
I'm  
not  
supposed  
to  
use  
table  
variables  
in  
join  
statements  
in  
SQL  
2005  
code?  
If  
so,  
that  
would  
be  
a  
major  
limitation.

I am not  
saying that  
at all. What  
I was  
stating is  
that by  
using a  
table  
variable you  
limit to a  
degree how  
much the  
optimizer  
knows  
about  
the  
data  
in that table  
variable and  
thus it may  
not always

Re: Poor performance after upgrading to sql server 2005

make the  
optimal  
decision  
vs. if it did  
have the  
missing  
stats.

Anyway,  
all  
of  
this  
is  
beside  
the  
point,  
which  
is  
that  
the  
exact  
same  
code  
in  
the  
exact  
same  
database  
with  
the  
exact  
same  
indexes  
runs  
a  
lot  
slower  
in  
SQL  
2005  
on  
a  
machine  
which  
is  
much,  
much  
faster  
in

Re: Poor performance after upgrading to sql server 2005

every  
way  
than  
the  
SQL  
2000  
machine.  
I'm  
starting  
to  
get  
the  
feeling  
we're  
going  
to  
have  
to  
re-write  
our  
database  
code  
for  
SQL  
2005.

As I stated  
the  
optimizer  
has changed  
between the  
two  
versions  
and  
in  
most  
ways is  
much  
smarter on  
how it  
handles  
things. As  
such most  
things  
should  
work as  
good or  
better than  
in 2000 but  
there will be

Re: Poor performance after upgrading to sql server 2005

times when  
that  
is  
the  
opposite.  
You may  
have simply  
been lucky  
that the  
optimizer  
guessed  
correctly in  
2000 where  
as in 2005 it  
guessed  
wrong.

I highly  
suggest you  
open a case  
with MS  
PSS so they  
can work  
directly  
with  
you to see if  
there is  
anything  
they can do.  
If this is a  
case in  
which  
the  
optimizer is  
not smart  
enough for  
a particular  
case they  
will want  
to  
know  
about it so  
they can  
make it  
better. We  
can't solve  
this in a  
newsgroup  
post  
when we  
don't have

Re: Poor performance after upgrading to sql server 2005

all the  
information  
and  
especially if  
we are  
dealing  
with  
pseudo  
code. This  
sounds like  
an  
optimizer  
issue and if  
so your  
best  
bet  
is  
direct  
contact so  
the support  
engineers  
can give the  
dev team all  
the  
right  
data to get  
to the  
bottom of  
this.

--

Andrew J.  
Kelly SQL  
MVP

"Rich  
Wood"

<RichWood@xxxxxxxxxxxxxxxxxx>

wrote in  
message

[news:0EDB38CF-BB34-47B3-B41D-20B02579BE7F@xxxxxxxxxx](mailto:news:0EDB38CF-BB34-47B3-B41D-20B02579BE7F@xxxxxxxxxx)

Andrew,

This  
is  
an  
extract  
from

Re: Poor performance after upgrading to sql server 2005

a  
function  
which  
returns  
a  
table  
variable.  
I  
hard-coded  
the  
'D'  
for  
purposes  
of  
the  
example.  
Normally  
it  
is  
an  
argument  
to  
the  
function,  
as  
are  
the  
other  
variables  
I  
"hard-coded".

Are  
you  
saying  
that  
I'm  
not  
supposed  
to  
use  
table  
variables  
in  
join  
statements  
in  
SQL  
2005  
code?  
If

Re: Poor performance after upgrading to sql server 2005

so,  
that  
would  
be  
a  
major  
limitation.

Anyway,  
all  
of  
this  
is  
beside  
the  
point,  
which  
is  
that  
the  
exact  
same  
code  
in  
the  
exact  
same  
database  
with  
the  
exact  
same  
indexes  
runs  
a  
lot  
slower  
in  
SQL  
2005  
on  
a  
machine  
which  
is  
much,  
much  
faster  
in  
every  
way

Re: Poor performance after upgrading to sql server 2005

than  
the  
SQL  
2000  
machine.  
I'm  
starting  
to  
get  
the  
feeling  
we're  
going  
to  
have  
to  
re-write  
our  
database  
code  
for  
SQL  
2005.

Rich  
Wood

"Andrew  
J.  
Kelly"  
wrote:

Rich,

You  
have  
several  
areas  
in  
which  
it  
makes  
it  
hard  
to  
determine  
a  
proper  
query

Re: Poor performance after upgrading to sql server 2005

plan  
based  
on  
the  
way  
you  
are  
coding  
this.  
One  
is  
that  
you  
use  
table  
variables  
which  
do  
not  
keep  
statistics.  
Since  
you  
join  
on  
at  
least  
one  
of  
them  
the  
optimizer  
will  
not  
have  
a  
clue  
as  
to  
if  
you  
have  
1  
row  
or  
10K  
rows  
in  
that  
table.  
The

Re: Poor performance after upgrading to sql server 2005

other  
is  
what  
I  
mentioned  
yesterday  
about  
the  
@Frequency  
variable.  
Since  
it  
is  
a  
variable  
the  
optimizer  
does  
not  
know  
the  
real  
value  
and  
takes  
a  
guess.  
From  
what  
I  
can  
see  
in  
this  
code  
you  
are  
hard  
coding  
it  
to  
'D'  
so  
why  
bother  
with  
a  
variable  
in  
the  
first

Re: Poor performance after upgrading to sql server 2005

place?

--

Andrew

J.

Kelly

SQL

MVP

"Rich

Wood"

<RichWood@xxxxxxxxxxxxxxxxxx>

wrote

in

message

[news:928E91C0-0634-4046-8964-7D2612CBE37E](mailto:news:928E91C0-0634-4046-8964-7D2612CBE37E)

I'm

not

sure

how

helpful

this

code

will

be

without

the

database

itself,

but

here

is

a

sample.

I've

included

the

SQL

2005

execution

plan

but

I

do

not

know

of

a

way

to

Re: Poor performance after upgrading to sql server 2005

send  
the  
SQL  
2000  
execution  
plan.  
This  
query  
takes  
0  
milliseconds  
in  
SQL  
2000  
and  
233  
milliseconds  
in  
SQL  
2005,  
even  
after  
running  
it  
a  
few  
times.  
The  
lines  
of  
code  
that  
cause  
the  
slow-down  
in  
SQL  
2005  
are:

```
AND  
1  
=  
CASE  
WHEN  
@FREQUENCY  
=  
'W'  
THEN  
(  
SELECT
```

Re: Poor performance after upgrading to sql server 2005

```
1
WHERE
CURR.DATA_ID
IN
(SELECT
DATA_ID
FROM
fdINDDATA_DAILY
D
(NOLOCK)))
ELSE
(
SELECT
1)
END
```

SQL  
Code:

```
DECLARE
@SEC_IDS
VARCHAR(5000),
@DATE
DATETIME,
@FREQUENCY
CHAR(1),
@PREVIOUS_DAY
DATETIME
```

```
DECLARE
@SEC_LIST
TABLE(
SEC_ID
INT)
DECLARE
@IND_DATA
TABLE(
SEC_ID
INT,
DATE
DATETIME,
PRICE
FLOAT,
PRICE_PREV
FLOAT,
DIVIDEND
FLOAT)
```

```
SET
@SEC_IDS
```

Re: Poor performance after upgrading to sql server 2005

```
=
'2'
SET
@DATE
=
'2006-02-17
08:39:20'
SET
@FREQUENCY
=
'D'

IF
@SEC_IDS
=
"
INSERT
INTO
@SEC_LIST
SELECT
SEC_ID
FROM
fdSecuritiesMasterList
SML
(NOLOCK)
WHERE
TYPE_ID
=
1
AND
SEC_ID
IN
(SELECT
ID_DEP_VAR
FROM
mNAV_REGRESSION
WHERE
FREQUENCY
=
@FREQUENCY)
ELSE
INSERT
INTO
@SEC_LIST
SELECT
ItemValue
FROM
dbo.fn_SplitIn2Rows(
@SEC_IDS,
';')
```

Re: Poor performance after upgrading to sql server 2005

```
--  
**  
US  
Tickers  
**  
  
--  
SELECT  
TOP  
1  
@PREVIOUS_DAY  
=  
CDAY  
FROM  
calUSA  
(NOLOCK)  
WHERE  
WDC  
=  
(SELECT  
WDC  
-  
1  
FROM  
calUSA  
(NOLOCK)  
WHERE  
CDAY  
=  
CONVERT(VARCHAR(10),  
@DATE,  
101))  
ORDER  
BY  
CDAY  
--  
Get  
data  
for  
indepent  
index  
variables  
INSERT  
INTO  
@IND_DATA(  
SEC_ID,  
DATE,  
PRICE,  
PRICE_PREV,  
DIVIDEND)  
SELECT
```

Re: Poor performance after upgrading to sql server 2005

```
DISTINCT
R.ID_IND_VAR
AS
SEC_ID,
CURR.[DATE],
CURR.VALUE,
PREV.VALUE,
ISNULL(RD.AMOUNT,
0)
FROM
mvNAV_REGRESSION
R
(NOLOCK)
JOIN
@SEC_LIST
L
ON
R.ID_DEP_VAR
=
L.SEC_ID
JOIN
fdSecuritiesMasterList
SML
(NOLOCK)
ON
R.ID_IND_VAR
=
SML.SEC_ID
JOIN
fdEXCHANGES
EXCH
(NOLOCK)
ON
SML.EXCH_ID
=
EXCH.EXCH_ID
JOIN
fdINDDATA
CURR
(NOLOCK)
ON
SML.SEC_ID
=
CURR.SEC_ID
JOIN
fdINDDATA_DAILY
PREV
(NOLOCK)
ON
SML.SEC_ID
=
```

Re: Poor performance after upgrading to sql server 2005

```
PREV.SEC_ID
AND
PREV.[DATE]
=
@PREVIOUS_DAY
LEFT
JOIN
(fdSECINFO
SI
(NOLOCK)
JOIN
fdDIV
RD
(NOLOCK)
ON
SI.SEC_ID
=
RD.SEC_ID)
ON
SML.SEC_ID
=
SI.SEC_RELATED
AND
ISNULL(SI.SEC_RELATED,
-1)
!=
1
AND
RD.XDATE
=
CONVERT(VARCHAR(10),
@DATE,
101)
WHERE
EXCH.CALENDAR
=
'USA'
AND
CURR.[DATE]
=
@DATE
AND
CURR.VALUE
IS
NOT
NULL
AND
PREV.VALUE
IS
NOT
NULL
```

Re: Poor performance after upgrading to sql server 2005

```
AND
R.FREQUENCY
=
@FREQUENCY
AND
1
=
CASE
WHEN
@FREQUENCY
=
'W'
THEN
(
SELECT
1
WHERE
CURR.DATA_ID
IN
(SELECT
DATA_ID
FROM
fdINDDATA_DAILY
D
(NOLOCK)))
ELSE
(
SELECT
1)
END
```

Execution  
Plan:

```
<?xml
version="1.0"
encoding="utf-16"?>
<ShowPlanXML
```