

Re: VB-ADO-SQL Server : SQL Server performs logins after some queries

Source: <http://www.tech-archive.net/Archive/VB/microsoft.public.vb.database.ado/2004-03/0501.html>

From: Oscar (*oku_at_xs4all.nl*)

Date: 03/23/04

Date: Tue, 23 Mar 2004 23:40:18 +0100

Val, Thanks for your advise!

I'll implement these points and see whether it will improve the performance.

*>.Also try to minimize selection of the records using WHERE clause
> and select only the records which you really need.*

I don't know another method than the WHERE clause to select records when there are some fields involved. Is there an alternative ?

The only thing that prevents me from using SP is that I am not sure whether SP's can be added to and maintained within SQL Server 2K by a VB front-end. In case that this would not be possible this would force to do the maintenance by hand. What's your suggestion about this point ?

Oscar

"Val Mazur" <group51a@hotmail.com> schreef in bericht
news:%231jDznIEEHA.1376@TK2MSFTNGP10.phx.gbl...

> Hi,

>

*> 1. Using ADO-specific type will not increase performance significantly,
but*

*> you could get some small improvement, since your application will avoid
late*

*> binding. If your query works very slow, then you could face an issue with
> the indexes. If you have some joins or WHERE clause in your statement,
then*

*> check if you have indexes in a database for the fields, involved into
> operation. Also try to minimize selection of the records using WHERE
clause*

> and select only the records which you really need

>

*> 2. Use list of the fields. Even if your query will look 10 times longer,
but*

*> list of the selected fields will include only third part of the all number
> of fields, performance could increase in several times. When you select*

all

> *the fields, then you waste bandwidth with the transferring of the data,*
> *which you do not need.*

>

> *3. In case of Update, ADO requires to maintain connection and locking*
> *through network (Internet), which is pretty expensive and requires*
multiple

> *roundtrips between server and client. In case of actual action query, like*
> *INSERT, it requires less resources and should work faster*

>

> *4. Yes, using SPs will definitely improve performance. What happens in*
case

> *of SP is the SQL Server will prepare query execution plan when application*
> *call SP first time. Then when you call same SP next time, it will take*
> *already prepared execution plan and will insert record much faster*

>

>

> *But, I think, main issue in your case is that YOU SELECT TOO MUCH DATA.*

Try

> *to minimize selection using WHERE clause to select ONLY the records which*
> *you REALLY need*

>

>

> --

> *Val Mazur*

> *Microsoft MVP*

>

>

> *"Oscar" <oku@xs4all.nl> wrote in message*

> *news:405f7569\$0\$85959\$e4fe514c@dreader8.news.xs4all.nl...*

>> *Hi Val,*

>>

>> *No I haven't. The reason that I use Object types is that my code should*

>> *work*

>> *with a SQL Server 2K database (at a internet server) as well as MS*

Access

>> *DAO (in LAN situation). Therefore I've coded a subroutine which sets the*

>> *recordset to a ADODB.recordset or a DAO recordset prior to creating the*

>> *recordset object.*

>>

>> *Although the application has a very good performance in a LAN situation,*

>> *in*

>> *case that the SQL Server is contacted from the internet (DSL*
connection),

>> *it*

>> *features a very poor performance. There are some routines which call a*
lot

>> *of queries and I've concluded that in case of an internet connection the*

>> *subroutines take as much as 40- 50 times the duration of a LAN*

situation

>> *and this is even the case for one single user. In particular I saw that*

> > *during the worst cases (longest duration) the upstream and downstream*
> > *bitrate to the client PC was almost equal while in better cases the*
> > *upstream*
> > *bitrate is far more less than the downstream. Therefore I would like the*
> > *following questions :*
> >
> > *1. Can using of ADODB recordsets instead of object give a far more*
better
> > *performance ?*
> > *2. While there are some queries which use about 20 fields of a table*
that
> > *has 80 fields should I use the wildcard (SELECT * FROM ..) in the*
> > *selection*
> > *string or should I use each necessary fieldname (that's the case now)*
> > *3. When updating with the .update method rather than the explicit INSERT*
> > *action, there was a huge loss of performance. Updating took about 4*
> > *minutes*
> > *for one record, while in case of a LAN it took about 1 second. Is this*
due
> > *to the .update method or any other suggestion ?*
> > *4. I don't use stored procedures yet as I want to optimize the code*
> > *without stored procedures first. Do you think that applying stored*
> > *procedures in my case with many recordsets and also wide (many fields)*
> > *recordsets and based on internet server/client environment could give a*
> > *huge*
> > *performance gain ?*
> >
> > *To my opinion, as I concluded after monitoring the network traffic,*
there
> > *was too much traffic generated. One retrieve query for retrieving*
employee
> > *data for about 50 employees with 14 queries for each employee generated*
> > *4MB*
> > *traffic data upstream and 4 MB downstream, totalising 8 MB within 7*
> > *minutes.*
> > *The test server operates on 1024 kbPs upstream and 512 kbPs downstream.*
> > *While the performance is far too slow, I don't dare to do additional*
tests
> > *with more users. I am sure I have to do recode (optimize) my source to*
> > *get*
> > *a better performance for the internet situation. Maybe you could provide*
> > *me*
> > *with any 'golden' tips ..*
> >
> > *Oscar*
> >
> >
> >
> >
> >
> >
> >
> >
> >

> >
> > *"Val Mazur" <group51a@hotmail.com> schreef in bericht*
> > *news:e6PgNm3DEHA.2576@TK2MSFTNGP11.phx.gbl...*
> >> *Hi Oscar,*
> >>
> >> *Do you have any bound controls in your application? Based on your*
another
> >> *posting, try to avoid using Object types and use ADO-specific types*
> >>
> >> *--*
> >> *Val Mazur*
> >> *Microsoft MVP*
> >>
> >>
> >> *"Oscar" <oku@xs4all.nl> wrote in message*
> >> *news:405d9ee3\$0\$8927\$e4fe514c@dreader9.news.xs4all.nl...*
> >>> *Hi Val,*
> >>>
> >>> *Setting the Activeconnection was a good approach. The problems almost*
> > *were*
> >>> *solved. However, sometimes it appeared again. After this I have set*
the
> >>> *Cursorlocation to Client side and also didn't set the*
Activeconnection
> >>> *property and this looks like the solution, since I didn't see the*
> > *problem*
> >>> *again up to now.*
> >>>
> >>> *thanks, so far for your help.*
> >>> *Oscar*
> >>>
> >>>
> >>>
> >>>
> >>>
> >>> *"Val Mazur" <group51a@hotmail.com> schreef in bericht*
> >>> *news:uyub0orDEHA.3980@TK2MSFTNGP09.phx.gbl...*
> >>>> *Hi Oscar,*
> >>>>
> >>>> *It should not open new connection, based on your code. Looks like*
your
> >>>> *recordsets pickup connection string from your opened connection*
> >>>> *instead*
> >>>> *of*
> >>>> *the actual connection. Try to set ActiveConnection property of your*
> >>>> *recordset before calling Open and remove connection from the method*
> >>>> *itself*
> >>>>
> >>>> *Set rst = New ADODB.Recordset*
> >>>> *Set rst.ActiveConnection= DB*
> >>>> *rst.Open strSQL, , adOpenForwardOnly, adLockReadOnly*

```
>>>>
>>>> --
>>>> Val Mazur
>>>> Microsoft MVP
>>>>
>>>>
>>>> "Oscar" <oku@xs4all.nl> wrote in message
>>>> news:405a480b$0$32701$e4fe514c@dreader17.news.xs4all.nl...
>>>>> Hi Val,
>>>>>
>>>>> At the start of the application the connection object is created
>>>>>
>>>>> Set db = New ADODB.Connection
>>>>> DB.Provider = "SQLOLEDB"
>>>>> DB.Properties("Data Source").Value = "80.xxx.yy.aaa,1433"
>>>>> DB.Properties("Network Library").Value = "DBMSSOCN"
>>>>> DB.Properties("Initial Catalog").Value = "CompanyDB"
>>>>> DB.Properties("User ID").Value = "sa"
>>>>> DB.Properties("Password").Value = pw
>>>>> DB.Open
>>>>>
>>>>>
>>>>> All recordsets in all procedure refer to this specific DB
connection
>>>>> object
>>>>>
>>>>>
>>>>> for opening not updatable recordset :
>>>>> Set rst = New ADODB.Recordset
>>>>> rst.Open strSQL, DB, adOpenForwardOnly, adLockReadOnly
>>>>>
>>>>> for adding/updating records
>>>>> Set rst = New ADODB.Recordset
>>>>> rst.Open strSQL, DB, adOpenForwardOnly, adLockOptimistic
>>>>> rst.Add
>>>>> rst!field1=value1
>>>>> rst.Update
>>>>> rst.Close
>>>>>
>>>>> for deleting records
>>>>> Set rst = New ADODB.Recordset
>>>>> rst.Open strSQL, DB, adOpenForwardOnly, adLockOptimistic
>>>>> rst.Delete
>>>>> rst.Close
>>>>>
>>>>>
>>>>>
>>>>> Meanwhile I've found the origin of all my SQL Server errors that I
>>>>> faced
>>>>> :
>>>>> lifetime of connection object
```

> > > > *I am porting a VB- Access DAO application to a VB-ADO SQL Server
2K
> > > > application. While the translated coding didn't give problems for
> > small
> > > > routines there was one large Subroutine that processed an employee
> > list
> > > > one
> > > > for one in order to report a table. During the processing of each
> > > employee
> > > > there were exactly 15 recordsets created while sometimes about 8
> > > > recordsets
> > > > were 'live' at the same time. All these recordsets were closed
> > > correctly.
> > > > While all these recordsets referred to the same connection object
DB
> > it
> > > > probably caused capacity problems, SQL Server appeared quite
> > > > instable
> > > and
> > > > every time reported that 'SQL was not there or that there was no
> > > > access
> > > > granted nearby the completion of the subroutine. At such a point,
> > > > SQL
> > > > Server
> > > > hanged and I had to waite about 2-3 minutes before the VB debugger
> > was
> > > > able
> > > > to continue. One of your responses to another thread in this group
> > gave
> > > me
> > > > the idea to think about more connection objects. In the case of
DAO
> > I
> > > was
> > > > not used to open one or more connection objects for different
> > > > recordsets
> > > > and
> > > > the application was able to have all recordsets refer to the one
> > > > and
> > > only
> > > > Database object. I decided to create 15 ADODB connection objects
in
> > > order
> > > > to
> > > > have each recordset refer to its own connection object and see :
all
> > > > errors
> > > > disappeared and the subroutine was performed almost 3 times as
fast
> > as
> > > > I*


```
> > > > How do you open your recordsets?
> > > >
> > > > --
> > > > Val Mazur
> > > > Microsoft MVP
> > > >
> > > >
> > > > "Oscar" <oku@xs4all.nl> wrote in message
> > > > news:405a08a8$0$141$e4fe514c@dreader4.news.xs4all.nl...
> > > > >
> > > > > I am testing a VB application that connects through ADO 2.7 and
> > > > > OLEDB
> > > > > with
> > > > > A
> > > > > SQL Server 2K database.
> > > > > For a reason that I don't know SQL Server seems to perform a
> > > > > login
> > > > each
> > > > > time
> > > > > that some queries are executed.
> > > > >
> > > > > This is what was logged in the tracer :
> > > > >
> > > > > SQLBatchCompleted SELECT * FROM TabEmployee ORDER BY kind
> > > > > errorlog (date) (time) logon
> > > > > login
> > > > > succeeded
> > > > > for user 'sa' Connection non-trusted
> > > > > eventlog 18454 login succeeded for
> > > > > user
> > > > 'sa'
> > > > > Connection non-trusted
> > > > > .
> > > > > .
> > > > > .
> > > > > SQLBatchCompleted SELECT * FROM TabClient
> > > > > exception error 2809 , severity 18 ,
state
> > 1
> > > > > SQLBatchCompleted exec tblContracting
> > > > > exception error 2809 , severity 18 ,
state
> > 1
> > > > > >
> > > > > >
> > > > > >
> > > > > > this is my connectionstring :
> > > > > >
> > > > > > Set db = New ADODB.Connection
> > > > > > db.Provider = "SQLOLEDB"
> > > > > > db.Properties("Data Source").Value = "80.xxx.yy.aaa,1433"
```

