

Re: Recommended Fill Factor

Source:

<http://www.tech-archive.net/Archive/SQL-Server/microsoft.public.sqlserver.programming/2005-01/1220.html>

From: Andrew J. Kelly (*sqlmvpnoospam_at_shadhawk.com*)

Date: 01/07/05

Date: Thu, 6 Jan 2005 23:04:28 -0500

1204 is trace flag for Deadlocks. Blocking and Deadlocks are really 2 different things. Blocking occurs all the time in any app in moderation. A Deadlock is when two users are attempting to update two or more objects usually in opposite order. A Deadlock will automatically get killed right away. Long term blocking is what you are looking for.

--

```
Andrew J. Kelly  SQL MVP
"rerichards" <cbrichards@comcast.net> wrote in message
news:1105057147.222277.121140@z14g2000cwz.googlegroups.com...
> Okay, I'll buy into that. But is there not also some validity in the
> Fill Factor needing to be tweaked?
>
> Also, if DBCC TraceOn(3605,1204,-1) is set, could it be there is
> nothing being written to the error log because of the timeout and the
> application no longer talking to SQL Server?
>
> Andrew J. Kelly wrote:
>> The fact that trimming down the table adds to the idea that it is
> scanning
>> and locking. The fewer rows in the table the less time it takes to
> scan
>> them and the locks are not held as long. It could simply be just the
> time
>> it takes to scan the tables but you will need to run some traces to
> find
>> out.
>>
>> --
>> Andrew J. Kelly  SQL MVP
>>
>> "rerichards" <cbrichards@comcast.net> wrote in message
>> news:1105055287.095395.84650@f14g2000cwb.googlegroups.com...
>> > That is what I initially suspected too, was locking. But when we
> began
>> > archiving and keeping the rows in the table under 300,000 the
> problem
>> > disappears until we let the table grow back to around 300,000 rows.
> As
>> > long as we keep the table trimmed down and optimizing after
> archiving
>> > the timeout issue does not occur. I have a trace set up for locking
> and
```

microsoft.public.sqlserver.programming: Re: Recommended Fill Factor

```
>> > the results have been clean.
>> >
>> > In this table is a column of datatype "text" that stores XML as
> well as
>> > "varchar", "datetime", and "int" columns.
>> >
>> > Are you still convinced that locking may play a part?
>> >
>> >
>> > Andrew J. Kelly wrote:
>> >> The fill factor has virtually nothing to do with inserts on a
> table
>> > with a
>> >> clustered index on an identity column. Each new row is appended
> to
>> > the end
>> >> of the last page and fill factors are ignored on inserts anyway
> when
>> > it
>> >> comes to page fullness. When it will be a factor is if someone
> else
>> > is
>> >> updating these rows after they are inserted and making a variable
>> > length
>> >> datatype larger. This will cause page splitting and hinder
>> > performance.
>> >> 40K inserts a day is not that much and in and of itself should not
> be
>> > an
>> >> issue unless there is someone blocking. That is most likely what
> is
>> >> happening. You probably have table scans going on that are
> slowing
>> >> everything down and potentially escalating locks to the table
> level
>> > and
>> >> blocking others. The number of rows in the table should be of
> little
>> > factor
>> >> on how fast the web is if you have proper indexes on the tables.
> You
>> > need
>> >> to run some traces to see where the bad queries are and try
> looking
>> > for
>> >> blocking as well.
>> >>
>> >> --
>> >> Andrew J. Kelly SQL MVP
>> >>
>> >>
>> >> "rerichards" <cbrichards@comcast.net> wrote in message
>> >> news:1105046760.031499.241940@z14g2000cwz.googlegroups.com...
>> >> > I have inherited a table with a primary key Clustered Index on a
>> >> > identity based column with a Fill Factor of 90%. This table also
>> >> > contains a second composite index of two columns that also has a
>> > > Fill
>> >> > Factor of 90%.
>> >> >
>> >> > A web service inserts about 40,000 records per day into this
> table.
>> >> > Throughout the day the data (including the composite index) is
```

microsoft.public.sqlserver.programming: Re: Recommended Fill Factor

```
>> > updated
>> >> > almost continually and is frequently read.
>> >> >
>> >> > The table can only handle about 300,000 records before the web
>> > service
>> >> > connecting to the database times out, either updating or
> inserting.
>> > To
>> >> > rectify this problem we archive a subset of records to a
> separate
>> >> > table, and following this archive we rebuild the index on the
>> > table.
>> >> >
>> >> > I have been told that the query time out / ODBC time out has
> been
>> > set
>> >> > fairly high.
>> >> >
>> >> > In monitoring the server, CPU, Memory, IO, all seem to be
> running
>> > at
>> >> > excellent levels.
>> >> >
>> >> > I suspect that the high Fill Factor is largely contributing to
> this
>> >> > dilemma. Do you concur with this assessment? Any suggestions?
>> >> >
>> >
>
```