

Re: Generating one table with a terabyte of data

Source:

<http://www.tech-archive.net/Archive/SQL-Server/microsoft.public.sqlserver.dts/2005-05/msg00214.html>

- *From:* "Jim Vierra" <jvierra@xxxxxxx>
 - *Date:* Mon, 9 May 2005 16:24:22 -0400
-

Consider un-raiding the disks for the bulk insert tables. It will give you a 15% or greater push. Ad the full 4Gb.

Run perfmon counters to a log and analyze them after a bulk load. Try to see if memory or processor is in need. I suspect that with two 3Ghz processors that you are having an IO problem which can be solved by partitioning the table.onto multiple disks depending on the SCSI card. I hope it is one the dual channel 320 Proliant cards.

Be sure to run the Bulk loader on the SQLServer and not over the network unless you have an optimized 1Gb net. The UI slowdown seems to indicate that the bottleneck is either with the processor or memory. As you clear up each bottle neck you will probably find a new one further down the line. You will have to do some analysis to chase this chain of events.

Remember – you can use multiple copies of BCP to load different sections of the file by filtering the file. Be sure that each copy of BCP is targeting a different set of output files for maximum load speed.

Run perfmon to analyze IO channel usage on you raid. If the card is to slow it will not keep up with a 15K disk set. If write buffering is turned on the buffer may fill up to often but this usually shows as an active UI due to IO stalling.

Watch the page file usage. Ideally it should not show any major activity. If it does something is wrong. Be sure to not allow any transactional relations to occur on the instance during loading as they can block the load, intermittently, but enough to degenerate performance. Monitor locks (you should have only a table lock) for interference.

—

Jim Vierra

"Jim Vierra" <jvierra@xxxxxxx> wrote in message
news:OSFJuJNVFHA.1508@xxxxxxxxxxxxxxxxxxxxxxxxxxxx

- > On fast bulk loads you should make sure that the database is not logging.
- > If you keep track of an identity or some key field you can undo the load
- > if necessary. Another thing is to split the updates across file sets.

Re: Generating one table with a terabyte of data

- > Each file can get it's own thread where, I believe, only one thread can be
- > release per file for IO. This can be done by portioning the data.
- >
- > You are running up against the classic issues of Data Warehousing. Fast
- > loading of daily data in the fastest time. The rules for a data warehouse
- > fir perfectly here. No transactions and many small file sets to allow
- > parallel loading of a table.
- >
- > The more memory you can add to the box the better. More processors – at
- > least two. Use /3GB to give SQLServer a larger address space but only if
- > you have 4GB of memory or more on the Datacenter Editions.
- >
- > The slowness of the UI says that the database is working. On SQS this is
- > what should happen. The UI is low priority on a server.
- >
- > You failed to say which OS and hardware specs. This is very important to
- > tuning.
- > Consider that you can set the database and OS up for loading and then
- > reboot with an operational configuration.
- >
- > If you use multiple files for partitioned data be sure the files sets are
- > on different drives. The drives should NOT be raided for maximum
- > performance. If the bulk load is fast enough you can always reload (like
- > we do in a DW) instead of doing backups. The raw data is archived to a
- > fast tape and a loader can reload it faster than you can restore a
- > database from tape.
- >
- > In the past I have split the databases up. Fast Loader.Read only
- > datawarehouse tables and transactional tables in different databases. The
- > read–only DW databases are not backed up but the others are.
- >
- > On a 3 Tb DW we could loaded close to over 1.5 G records per day with an
- > average record greater than 700 bytes. The loading had to be complete by
- > 7 AM and did not start until after 2 AM. 4 1.2Ghz (64bit) Processors and
- > a RAID farm. We didn't have to tear down the RAID 5 farm because we had
- > hundreds of file sets and could optimize parallel loading. We also had 8
- > Channels of Raid and 4Gb of memory. Building this beast we learned a lot
- > about optimizing loading.
- >
- > I applied the lessons to a SQLServer 7.0 instance on a 4 Processor
- > Proliant with 4Gb and one RAID channel. We were able to blast a few
- > hundred million records in daily within a four hour window something
- > everyone said would be impossible. Like you we needed to do repeated
- > passes on these records to find subtle matches in the data. (Data Mining).
- > Applying the DW rules worked for this application.
- >
- > ---
- > Jim Vierra
- >
- > "Kevin3NF" <KHill@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx> wrote in message
- > news:%23i42snMVFHA.628@xxxxxxxxxxxxxxxxxxxxxxxxxxxx

Re: Generating one table with a terabyte of data

>> I've locked SQL server to 1/3 physical memory usage. DB growth is 512Meg
>> chunks
>>
>> You need to free it up to fly, and pre-allocate the db size. In-process
>> file growth will kill performance
>>
>> --
>> Kevin Hill
>> President
>> 3NF Consulting
>>
>> www.3nf-inc.com/NewsGroups.htm
>>
>> www.DallasDBAs.com/forum – new DB forum for Dallas/Ft. Worth area DBAs.
>>
>> www.experts-exchange.com – experts compete for points to answer your
>> questions
>>
>>
>> "lee" <lee@xxxxxxxxxxxxxxxxxxxxxxxxxxxx> wrote in message
>> news:B772A8C2-F009-4E29-816C-5B6C2F45151E@xxxxxxxxxxxxxxxxxxxx
>>> Hi All,
>>>
>>> I need ideas on how to improve a process that can generate a terabyte of
>>> data in a week. Our company produces software that analyzes code and one
>>> of
>>> the things we do is Code Duplication. We're processing hundreds of
>>> millions
>>> of lines against each other and generate gigabytes of results. In
>>> reality we
>>> generate over 20 gigabytes of data a day. We are noticing major slow
>>> downs in
>>> the system. Here's how we're going about it
>>>
>>> 1. Non-Indexed table residing on it's own FileGroup (which is not on the
>>> same drive as the SQL server).
>>> 2. A process that generates Text files that we use BULK INSERT to load
>>> the
>>> data (any other way of doing inserts is just to slow).
>>>
>>> This process should generate 10 million rows in the table per hour. Only
>>> problem is that we're noticing a considerable slow down as we pummel the
>>> Database. We're making sure our Bulk Insert has exclusive table locks
>>> and
>>> nobody else accesses the system while this is happening. We're also
>>> noticing
>>> a strange behavior at the OS level. The system just slows down to a
>>> crawl.
>>> Even mouse movement becomes jagged (Win2K Server).
>>>
>>> I should also mention that once we're done inserting we create indexes

Re: Generating one table with a terabyte of data

>>> at
>>> the end. Average time to create 1 indexed field (on a different
>>> filegroup) on
>>> 100 million rows is about 3 to 4 hours. Seems a little excessive to me,
>>> must
>>> be doing something wrong.
>>>
>>> I've gone over this 100 times and I can't see anything else to improve
>>> performance. I have a friend that is an ORACLE DBA and he keeps telling
>>> me
>>> that SQL server just can't handle this type of volume and that I should
>>> move
>>> to ORACLE. I refuse to believe ORACLE can do more then SQL Server. He
>>> actually took one of our 500 meg text files and loaded it in 1/10 the
>>> time it
>>> took SQL Server. My answer to that is "I must be configuring it
>>> incorrectly".
>>>
>>> I've locked SQL server to 1/3 physical memory usage. DB growth is 512Meg
>>> chunks. We don't need any transaction logs, so we're using RECOVERY
>>> MODEL =
>>> SIMPLE (any other mode kills SQL server anyway, LOG fills to fast with
>>> our
>>> data creation speed). I've thrown everything at this except the kitchen
>>> sink.
>>>
>>> Any suggestions on how I can improve my Bulk Inserts and how to improve
>>> my
>>> Index creations? Should I consider SQL Server 2005? Will it handle my
>>> needs
>>> better?
>>>
>>
>>
>
>

• *Follow-Ups:*

- ◆ **Re: Generating one table with a terabyte of data**
◇ From: lee

• *References:*

- ◆ **Generating one table with a terabyte of data**
◇ From: lee
- ◆ **Re: Generating one table with a terabyte of data**
◇ From: Kevin3NF

Re: Generating one table with a terabyte of data

◆ **Re: Generating one table with a terabyte of data**

◇ From: Jim Vierra

- Prev by Date: **can't see my activex code in the activex script task**
- Next by Date: **Re: Unzip file using DTS**
- Previous by thread: **Re: Generating one table with a terabyte of data**
- Next by thread: **Re: Generating one table with a terabyte of data**
- Index(es):
 - ◆ **Date**
 - ◆ **Thread**