**[Erin Welker](http://sqlblog.com/blogs/erin_welker/default.aspx)**

SQL Server and Business Intelligence

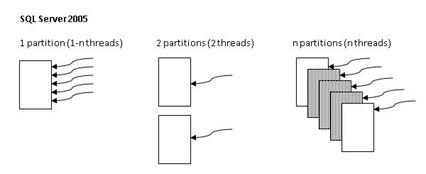
**Partitioning enhancements in SQL Server 2008**

Horizontal partitioning can be a great benefit to data warehouses, as it greatly facilitates the management of very large databases. Of course, querying on partitions can also have performance benefits, especially when only a small percentage of the database is queried and partition elimination can occur. SQL Server 2005, however, can have some unexpected behavior when querying few partitions.

In SQL Server 2005, one thread is allocated per partition when multiple partitions are queried. These threads then work in parallel to retrieve the rows to satisfy the query. If only one partition is queried, however, SQL Server 2005 allocates as many threads as are available to the one partition. As you can quickly see, there is a vulnerability when it comes to queries that query very few, but more than 1 partition.

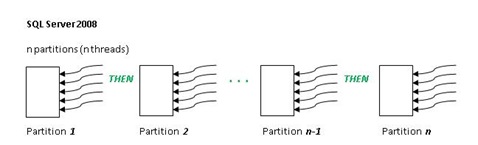
The common scenario is when only 2 partitions are queried. For example, let’s say you have a data warehouse database that is partitioned on a sales date by month. You write a query to compare last month’s sales to this month’s sales. Now, let’s say this runs on a 32-processor server. How many threads get allocated to this query? The answer is 2, one per queried partition.

This last scenario can become even more complicated when a window of time causes the query to fluctuate between 1 and multiple partitions. Let’s say we compare today’s sales to the same day of the week last week. If both weeks are in the same month, we get full thread allocation. If the weeks span two months (= 2 partitions), however, we only get 2 threads allocated and the query can appear to slow down considerably, for no apparent reason (at least, probably not to the issuer of the query).

[](http://sqlblog.com/blogs/erin_welker/WindowsLiveWriter/PartitioningenhancementsinSQLServer2008_F7B5/SQL2005Partitioning_2.jpg)

Note: All of these scenarios assume that the queries are able to eliminate partitions due to a direct search on the partitioning key. Be sure partition elimination is occurring in order to gain the best partitioned table query performance.

SQL Server 2008 behavior changes the way in which threads are allocated. It allocates all available threads to each partition relevant to the query in a round robin fashion. This results in fast and consistent query execution, no matter how many partitions are queried.

[](http://sqlblog.com/blogs/erin_welker/WindowsLiveWriter/PartitioningenhancementsinSQLServer2008_F7B5/SQL2008Partitioning_2.jpg)

Note that this makes it all the more important to allocate partitions to filegroups that are spread across many disk spindles, allowing the query on a given partition to be as efficient as possible.

But wait, there’s another improvement to partition query behavior in SQL Server 2008. SQL Server 2005 only allows lock escalation to the table level. Let’s say your query eliminates 75% of a table’s partitions, but scans all of the remaining partitions (1/4 of the table). If SQL Server decides to escalate the lock, all other queries will be locked out even if they are querying completely different partitions. SQL Server 2008 provides a table option to override this default behavior. Note that the default is still to escalate to table locks (at least for now) so this option will need to be changed to take advantage of partition-level lock escalation. Find out more about this, like I did, on über blogger Paul Randall’s [post](http://www.sqlskills.com/blogs/paul/2008/01/14/SQLServer2008PartitionlevelLockEscalationDetailsAndExamples.aspx).

Published Sunday, February 10, 2008 6:37 PM by [ErinW](http://sqlblog.com/user/Profile.aspx?UserID=2888)



**Comment Notification**

If you would like to receive an email when updates are made to this post, please register [here](http://sqlblog.com/user/CreateUser.aspx?ReturnUrl=/blogs/erin_welker/archive/2008/02/10/partitioning-enhancements-in-sql-server-2008.aspx)

Subscribe to this post's comments using [RSS](http://sqlblog.com/blogs/erin_welker/commentrss.aspx?PostID=5019)

**Comments**

|  |  |
| --- | --- |
| http://sqlblog.com/utility/anonymous.gif | [**Jason Haley**](http://sqlblog.com/utility/Redirect.aspx?U=http%3a%2f%2fjasonhaley.com%2fblog%2farchive%2f2008%2f02%2f11%2f141190.aspx) **said:**  February 11, 2008 10:06 AM |
| http://sqlblog.com/utility/anonymous.gif | [**Rob Mills**](http://sqlblog.com/utility/Redirect.aspx?U=http%3a%2f%2fwww.dotnetadvisor.com) **said:**  This looks like a good enhancement and one I could use on one of my current projects.  Now the question will be when my company will approve upgrading...  February 11, 2008 11:06 PM |
| http://sqlblog.com/utility/anonymous.gif | [**» Daily Bits - February 12, 2008 Alvin Ashcraft’s Daily Geek Bits: Daily links, development, gadgets and raising rugrats.**](http://sqlblog.com/utility/Redirect.aspx?U=http%3a%2f%2fwww.alvinashcraft.com%2f2008%2f02%2f12%2fdaily-bits-february-12-2008%2f) **said:**  PingBack from <http://www.alvinashcraft.com/2008/02/12/daily-bits-february-12-2008/>  February 12, 2008 9:55 AM |
| http://sqlblog.com/utility/anonymous.gif | **Bart Czernicki said:**  Another step in the right direction for partitioning.  SQL 2000 partitioning performance was horrible. SQL 2005 still not as good (because of the multi-core procs as mentioned in the article).  Looks like these 2008 features might finally give DW architects some more options.  February 14, 2008 2:10 AM |
| http://sqlblog.com/utility/anonymous.gif | **Tim T said:**  It would follow that with 2005, you would tend to make smaller partitions to minimize the size of a partition a single thread would have to process.  In 2008, larger partitions are fine, as there will be more threads working?  Any guidelines for deterimining partition size?  April 14, 2008 5:15 PM |
| http://sqlblog.com/utility/anonymous.gif | [**Denis Gobo**](http://sqlblog.com/utility/Redirect.aspx?U=http%3a%2f%2fsqlblog.com%2fblogs%2fdenis_gobo%2farchive%2f2008%2f12%2f31%2f10843.aspx) **said:**  Wow, it has been already a year since I wrote A year in review, The 21 + 1 best blog posts on SQLBlog  December 31, 2008 10:37 AM |
| http://sqlblog.com/utility/anonymous.gif | **Durgesh said:**  Question: Can we partition existing table? Unfortunately I haven't seen answer to this question anywhere.  Thanks  May 20, 2009 1:46 AM |
| http://sqlblog.com/utility/anonymous.gif | **Mike Schubert said:**  Existing tables can be partitioned with creating a "clustering Index ON PartitionFunction (Field)". If you don't need the Index you can drop this index. the partitioning would be not deleted :-)  so long.  Mike  July 22, 2009 2:38 AM |
| http://sqlblog.com/utility/anonymous.gif | [**Eralper**](http://sqlblog.com/utility/Redirect.aspx?U=http%3a%2f%2fwww.kodyaz.com%2farticles%2fhow-to-partition-table-non-partitioned-table-sql-server-2008.aspx) **said:**  Partisionning is a great feature of SQL Server, but it is somehow difficult for DBAs to use this performance enhancing feature because of the reasons just like Durgesh has noted. I agree with Mike that it is still an answer to this faq. I have demonstrated an example on this at <http://www.kodyaz.com/articles/how-to-partition-table-non-partitioned-table-sql-server-2008.aspx> . You will realize that it is easy to convert it to a partitioned table, but it requires some pre-work. |