(October 31st, 2011)

If you know someone who you think would benefit from being an Insider, feel free to forward this PDF to them so they can sign up here.

Quick tips for our Insider friends!

Hey Insiders!

This bi-weekly Quick Tips is coming to you from Las Vegas, where we're at the bi-annual SQL Connections conference (that we co-chair). Vegas is a bit of an interesting place and it wasn't until I'd been here a few times that I got used to the 24-hour environment and the timeless atmosphere inside the casino hotels. We don't gamble, but we do like seeing shows and this year we're going to check out Rod Stewart before he retires – should be good!

The most recent book I've read is Peter Carey's *Parrot and Olivier in America*. It follows two exiles from Paris when Napoleon is in power for the second time and their adventures in the young America. Their situation is complicated by them being occasional master and servant. The language is excellent without being overly verbose and the book does a great job of exploring their past lives and what made them into who they are.

Check out the final part of the newsletter for details of our **new remote DBA service**, plus our projected class schedule through August 2012. Our winter 2012 classes are now open for registration!

Please let us know if you liked what you read/saw here and/or have any suggestions for future Quick Tips.

Paul's Ponderings

Most people equate blocking and resource contention with locks, but the majority of the time it's not locking issues that cause query performance problems.

My favorite module from our performance tuning classes is about a tuning methodology called 'waits and queues'. It's described in detail in:

- The whitepaper Performance Tuning Using Waits and Queues
- My blog post Wait statistics, or please tell me where it hurts and category Wait Stats

The basic premise is that whenever an executing thread in SQL Server needs a resource (e.g. a data file page, a lock, some memory, a transaction log buffer) and the resource isn't immediately available, the thread gets *suspended* and has to wait until the resource is available.

SQL Server is keeping track of all the reasons why threads have to wait as well as how long they have to wait. You can get this information using the `sys.dm_os_wait_stats` DMV and my blog post has a query that aggregates the information and shows it in actionable form.

This information is an excellent starting point for figuring out any performance problem. It will tell you what the most common reason is for threads waiting and point you in the right direction for deeper analysis – saving you from wasting time checking things that aren't contributing to the slowdown.

The very best starting point though is to look at the output from the `sys.dm_os_waiting_tasks` DMV as that will list all threads that are waiting, and what they're waiting for – in other words, what's contributing to slowdowns right now.

Here's an example. Imagine you have a table with a narrow row size and a clustered index on an INT IDENTITY column. This is one of the things we advise really strongly – keep your clustered index key narrow, static, unique, and ever-increasing. This works perfectly well until you get to a pretty high-end workload. For instance, with several thousand concurrent connections (and maybe just hundreds in some cases) all inserting into the table, there's going to be blocking. But if you look at the locks that are held with `sys.dm_tran_locks`, it doesn't show anything amiss, because all the threads are holding page IX locks and key X locks. Things look normal.

However, if you look in `sys.dm_os_waiting_tasks` you'll see that most of the threads are waiting with a wait type of PAGELATCH_EX. This is because a lock is not sufficient to be able to change an in-memory copy of a page. As that page is really a data structure in SQL Server's memory space, a *latch* is required. Latches are akin to lightweight locks and protect access to memory.

As the clustered index is on an identity column, it's going to be an append-only insert pattern with many rows on each data page. This is a good thing. But this means many threads will be trying to insert rows simultaneously on the same data file page. The required locks don't cause blocking because each thread has compatible page locks and row locks on individual rows. However, the threads all need to acquire an exclusive latch on the page – which isn't possible simultaneously. For a 'normal' workload this isn't a big issue, but at the high end, it can cause major blocking.

Blocking does not mean that locks are the issue. Blocking means that something is holding up the workload on your server and you need to find out what.

Summary: if things are running slowly on your server, don't flail around looking at things like fragmentation or profile for long-running queries. Go straight to the DMV `sys.dm_os_waiting_tasks` to see what queries are waiting for – and then dig in based on those wait types.

Call to action: go run the query I've got in the blog post above. It'll show you what the most prevalent wait types are on your server. Use the info in the blog post and in the whitepaper I link to above to figure out what's taking time on your system and blocking your workload.

I'm really interested to know your thoughts on investigating blocking using wait stats analysis—feel free to drop me a line, confidentially as always.

Video Demo

As you all know, we recently hired Joseph Sack. I wanted to introduce him to you so I asked him to record the demo video for this newsletter, which he was kind enough to do. As a former Premier Field Engineer, he has some invaluable experience in dealing with performance troubleshooting. In the video, he presents some methods for troubleshooting deadlocks that you'll find useful. The video is about 12 minutes long.

I produced the video in WMV and MOV formats so everyone can watch. You can get the videos:

- For WMV: here
- For MOV: here

I recommend downloading before watching. And you can get the demo code here.

SQLskills Offerings

We've just launched a new "remote DBA" service where we provide repeated mini-health checks on regular intervals (once a server has been initially health-checked) plus automated monitoring of SQL Server for problems – and what to do when a problem inevitably *does* arise. This is an excellent way to gain access to our team's incomparable expertise and experience – especially if you don't have a full-time DBA on staff. However, even if you do, this gives you an additional set of expert eyes to watch over your critical data.

See here for more details.

Registrations are continuing for our two remaining Immersion Events left this year in Atlanta, GA the week of December 5. During that week, we are running both our Internals and Performance (IE1) plus a *new* BI Immersion Event. Time to use that 2011 budget before you lose it! And, register soon as the early-bird discount expires on midnight PST Friday, November 4th.

For everyone that's been asking – YES – we will be offering all four of our Immersion Events in 2012 with these classes being added to our schedule starting in October. Here's a list of some of the classes and cities we're targeting:

- Available for registration **NOW**:
  - February 27 – March 2, 2012: Internals and Performance (IE1) in Tampa, FL
  - March 5-9, 2012: Performance Tuning (IE2) in Tampa, FL
  - March 12-16, 2012: High Availability/Disaster Recovery (IE3) in Tampa, FL
- Available for registration December/January
  - April 2012: IE1 & IE2 in Chicago, IL
  - May 2012: IE1 & IE2 in London, UK
  - August 2012: IE1-IE2-IE3-IE4 in Redmond/Bellevue (WA) again!

See here for all the details.

Summary

We hope you've enjoyed this issue - we really enjoy putting these together.

If there is anything else you're interested in, we'd love to hear from you - drop us a line.

Thanks - Paul and Kimberly