SQL Server Parallelism Settings - Video Tip

Cost threshold for parallelism and the max degree of parallelism are 2 important setting for your SQL Server. Find out how to set these and what they should be set to.

Transcription:

Steve Stedman 0:09
Hi, this is Steve Stedman and I'm here to talk about a couple of the daily checkup or database health quick scan items that we come across occasionally. These are settings that we optimize or configure on the SQL Server in order to help with performance. Usually, the first are two that really play well together. First one is the cost threshold for parallelism being set to the default of five. The next one is the max degree of parallelism means that the default of zero. And what these relate to is specifically, when a query is run, and it's going to be run across multiple cores or processors, it needs to be split up, and that splitting it up is what's usually referred to as parallelism. So in the middle 90s, when Microsoft set these up as the defaults, it was really rare for computers to have the number of processors or cores that we have today. And also, it was really rare for people to do very big queries at all, because the servers were so limited. So the default for the cost threshold for parallelism is set to five. It's been this way since pretty much day one of SQL Server. But what this means is the cost is the amount of work involved in processing that query. So if you ever look at a execution plan, and it shows a specific cost number in there, that's a cost of five, anything with a cost greater than five, SQL server will send that to multiple threads, or multiple cores to do the processing of that. And what's bad about that, as you get some really small queries that don't take an awful lot of work that are being sent off to be processed in parallel. And what we like to do is we'd like to adjust the setting to about 50, or 60, or 70, somewhere kind of in that range, and then see how things go. And what you're going to see if this is set too low is you're going to see a whole lot of cx packet weights on your system, were too small of queries are being forced to go parallel, when what you really want is just want one big query to go parallel. Now the next setting is the max degree of parallelism. And what this is saying is, how many threads or cores is the processing going to be split off to be run amongst. So if you have four cores on your server, and it set to zero that just says take however many SQL Server fields, it needs to do the processing. But if you've got a server with, let's say, 48 cores, or 64 cores, or crazy big numbers like that, it doesn't make sense necessarily to use all of those cores for parallel processing on the same query. So oftentimes, we'll set that number to around eight, maybe as a max, I know on newer versions of SQL Server, we can set it higher, maybe up to around 16, even based off the number of cores in the system, but

you never want it to be more than the number of cores, and you never want it to be too big. It's one of those sort of Goldilocks type settings where you don't want it too hot, you don't want it too cold, you just want to just write. Now, those two work together, because if you have a cost threshold that's too low, and a max degree of parallelism that's too high, you'll get a lot of really small queries being split up, going to multiple cores to do all the work and then having to come back and bring the results back to one place. So we recommend adjusting the cost threshold to somewhere around 50 to 70 and monitor it and see how things go and setting the max degree of parallelism up to the number of cores with a max of like, depending on the version of SQL server somewhere between eight and 16 typically. Alright, this is Steve Stedman. Hopefully these SQL Server tips will help you out today. Have a great day.