Maximum Degree of Parallelism for Blackbaud CRM

Blackbaud Enterprise Performance Team

Overview
The SQL Server Maximum Degree of Parallelism (MaxDOP) setting is relevant to multi-core servers, which is just about all servers these days. When the query planner determines that a query will benefit from parallelism, it inserts parallel operators into the query plan. Then at execution time, it determines the actual amount of parallelism to use based on the number of schedulers available on the system.

When you set MaxDOP for a SQL Server instance, it applies to all databases that the instance supports. You can also specify MaxDOP as a hint for queries and indexing operations, but since Blackbaud CRM statements don’t include MaxDOP hints, this investigation focuses on setting the value for the system.

The default MaxDOP value of 0 lets the system use as many SQL Server schedulers as seem appropriate. You can set MaxDOP to a different value to indicate the maximum number of schedulers that can execute a query. In general, the value should not be greater than the number of physical cores in the server (not the number of logical cores if hyperthreading is used). The “wisdom of the web” suggests that expensive, long-running queries run faster with more parallelism and that shorter transactions run better with MaxDOP = 1. Blackbaud CRM uses a mix of short transactions, business processes, ad-hoc queries, etc. Hence it’s a bit unclear what value would be most appropriate.

The Enterprise Performance Team wanted to determine the optimal MaxDOP setting for Blackbaud CRM.

Experimental Setup
To seek a recommendation for MaxDOP for Blackbaud CRM, we conducted a number of experiments to assess different types of operations.

- For everyday usage by online users, we used a workload that models interactive users on the system.
- Another workload modeled users of Enhanced Revenue Batch functionality.
- For long-running ad-hoc queries, we reused a set of queries from a previous investigation.
- For re-indexing operations, we used a set of longer-running ALTER INDEX … REBUILD statements. Re-indexing operations typically occurs in off-prime hours, but some sites run so many business processes during that time that it is important to understand the impact of MaxDOP on re-indexing.
We ran each workload with MaxDOP settings of 0, 1, 2, 4, and 8 on a test rig where the OLTP server is configured with 12 physical cores and is not hyper-threaded.

Summary and Recommendation
The following table summarizes the four workloads performed with each of the MaxDOP settings we tested. A “+” indicates a good result, “++” is super-good, “–” is bad, and “.” is insignificant. The sections that follow provide more details about the results for each workload.

<table>
<thead>
<tr>
<th></th>
<th>MaxDOP 0</th>
<th>MaxDOP 1</th>
<th>MaxDOP 2</th>
<th>MaxDOP 4</th>
<th>MaxDOP 8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interactive Workload</td>
<td>.</td>
<td>+</td>
<td>++</td>
<td>++</td>
<td>+</td>
</tr>
<tr>
<td>ERB Workload</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>Long-running Queries</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>Index Rebuilds</td>
<td>.</td>
<td>–</td>
<td>.</td>
<td>+</td>
<td>.</td>
</tr>
</tbody>
</table>

Across the workloads, MaxDOP = 4 seems to be is the best overall setting. Of course, this is based on the presumption that our test rig is generally similar to a Blackbaud hosting organization (SDO) system or a customer system and that our workloads are representative. We tried to achieve both of these goals.

When you consider a change to MaxDOP, we recommend that you compare before and after performance measurements for key operations – including interactive functionality that users access routinely and large business processes that may take significant time to run. Keep in mind that for many sites, the staging system runs on a smaller server than the production system, so measurements in staging may not translate well to production.

MaxDOP and Interactive Usage
The Interactive workload represents 135 distinct user activities. To identify the most commonly performed operations, we analyzed the Internet Information Services (IIS) logs for a large organization that fights hunger and poverty. To represent peak usage time, the workload modeled 162 users performing these activities. The workload ran against a “masked” copy of the customer’s actual production Blackbaud CRM database. In other words, the workload ran against real data but with changes to conceal any personally identifying information in the database. The workload used load generator systems to execute work against a set of servers that are configured similarly to how Blackbaud’s hosting organization (SDO) or an on-premises customer would configure their servers.

The Performance Index chart below illustrates the overall changes in the performance of a test run of the workload. We performed a baseline set of runs with Blackbaud CRM 3.0 SP7, and we measured the response time of each transaction. The Performance Index of any test run is the average response time of transactions in the baseline divided by the average response time of transactions in the test run. So overall, a higher Performance Index means that the test run was faster than the baseline on average. The red and green lines on the chart show the statistical variability of the baseline runs. A result outside those bounds is likely to be of interest, while a result inside indicates no change from the baseline.
For each MaxDOP value that we studied (0, 1, 2, 4, 8), we performed four test runs to show the consistency of the result. MaxDOP = 0 is the default value use for the baseline runs, so it is no surprise that the MaxDOP 0 runs fall within the expected variability of baseline runs.

The chart shows performance improvements for all non-default settings. The largest improvement is for MaxDOP = 4, which nearly doubles the average performance. MaxDOP = 2 performs almost as well.

MaxDOP and the ERB Workload
The ERB workload collects 11 distinct user activities. To identify the most commonly performed operations for Enhanced Revenue Batch functionality, we analyzed the IIS logs for a large university. To represent a very high level of usage, the workload modeled 150 users performing these activities. The workload ran against a “masked” copy of the customer’s actual production Blackbaud CRM database. In other words, the workload ran against real data but with changes to conceal any personally identifying information in the database. The workload uses load generator systems to execute work against a set of servers running Blackbaud CRM 4.0 that are configured similarly to how SDO or an on-premise customer would configure their servers.

The Performance Index chart below illustrates the overall changes in the performance of a test run of the workload, as described in the previous section.

For each MaxDOP value that we studied (1, 2, 4, 8), we performed three test runs to check for consistency in the results. We compared these runs to a baseline set that used MaxDOP = 0.

We can see that MaxDOP settings have essentially no effect on this workload overall. We do not see performance improvements, and the data does not indicate any harm from different settings.
MaxDOP and Long-running Ad-hoc Queries

The chart below illustrates the time that is required to run a number of long-running ad-hoc queries against the large university’s masked database. The blue bars show how long the queries ran for each MaxDOP setting. To make it easier to compare the results to the default, the red line illustrates how long each query ran with MaxDOP = 0.

For each query, we studied multiple MaxDOP settings (0, 1, 2, 4, 8). It is apparent that MaxDOP settings have essentially no effect on the execution of these queries. We do not see performance improvements, and the data does not indicate any harm from different settings.

When these queries ran, casual observation found that a single CPU core was used for an overwhelming majority of the run and that very little disk activity occurred. Based on that observation, it’s easy to see why limiting the available parallelism had no effect: Only one thread was used at a time anyway.

Despite the consistency of these measurements, we should use caution in assuming that MaxDOP settings never affect long-running ad-hoc queries. This set of queries was expressly selected for another project because they were long-running, and it is possible that they are long-running because they don’t parallelize well.
MaxDOP and Index Rebuilds

The chart below illustrates the time that is required to rebuild indexes for a sample of 10 indexes from the large university’s masked database. We selected some longer-running indexes from a variety of tables. Although the rebuilds are pretty quick (most are less than 20 seconds), rebuild times are cumulatively important to management overhead because the Blackbaud CRM database includes tens of thousands of indexes.

The blue bars show how long it took to rebuild the indexes for each MaxDOP setting. To make it easier to compare the results to the default, the red line illustrates how long it took to rebuild each index with MaxDOP = 0.
For many indexes, MaxDOP = 1 actually increases the rebuild time. Meanwhile, MaxDOP = 4 generally performs better than MaxDOP = 0.

In the chart above, it’s not easy to assess the total impact of MaxDOP settings across the set, so the following chart adds up the time for all 10 indexes. Based on this set of indexes, the best value for index rebuilds appears to be MaxDOP = 4.
MaxDOP is a setting that has different impacts for different work in the database. We tried to represent different types of work with different databases, but more is always possible. As the Enterprise Performance Team continues to build its set of workloads, we expect to do follow-up testing with MaxDOP.

Some “tribal knowledge” asserts that MaxDOP = 1 is a good thing, and some hosted and on-premise customer systems use this setting, based on prior observations from Support and SDO. For those systems, MaxDOP = 1 probably was a reasonable choice and was almost certainly better than the default. The ability to run workloads and take reproducible measurements is new, so we believe this investigation features is an element of scientific rigor in recommending MaxDOP = 4 that was not previously possible. However, as we already noted, there is not a single correct answer for MaxDOP. The right setting depends on the overall workload, and we will add to the set of available workloads in the future.

In any event, the default MaxDOP = 0 does not appear to be optimal.