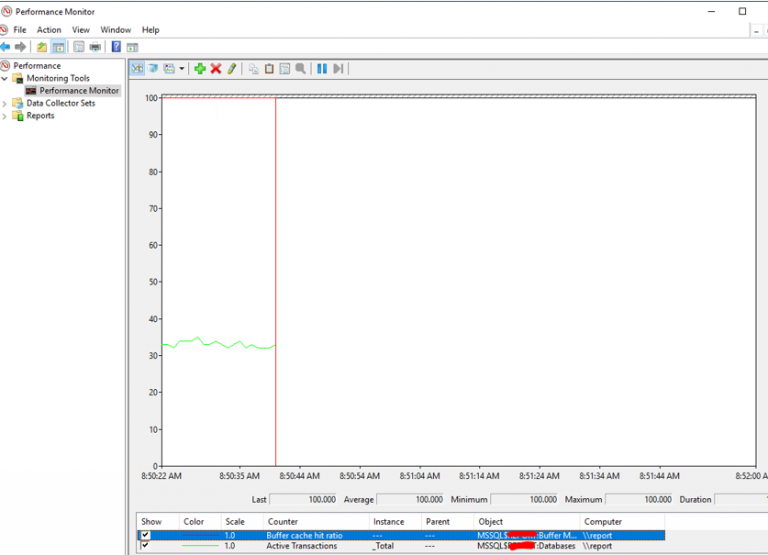
Мониторинг производительности и диагностика проблем SQL Server

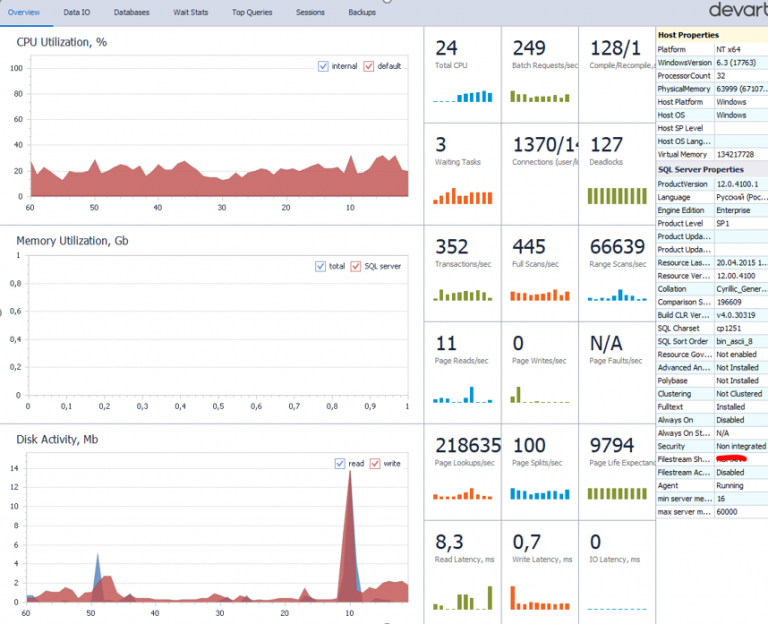
Инструменты для диагностики SQL Server

Если вы правильно диагностировали проблему, то половина работы уже сделана. Рассмотрим какие инструменты обычно используются системным администратором для диагностики различных проблем в SQL Server:

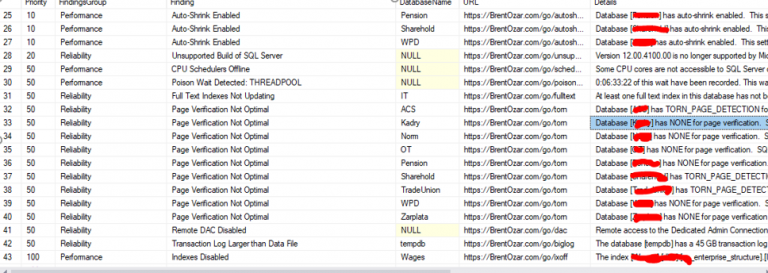
* T-SQL — самый мощный, простой и незаменимый инструмент для поиска проблем и анализом производительности SQL Server. Практически все другие инструменты для работы с SQL Server используют T-SQL. Нет ничего такого, чтобы вы не смогли сделать с помощью T-SQL.
* SQL Server Management Studio — без SSMS практически невозможно работать с SQL Server. С помощью SSMS вы можете посмотреть Activity monitor, проанализировать план запроса, посмотреть параметры сервера или базы данных и многие другие вещи.
* Журналы ошибок SQL Server и Windows – если что-то идёт не так, журнал ошибок — это первое место, куда смотрит системный администратор. Журнал ошибок SQL Server можно посмотреть через SSMS. Журналы Windows можно посмотреть через оснастку eventvwr.msc.
* Монитор ресурсов Windows — resmon.exe незаменимый инструмент Windows для быстрой оценки состояния ресурсов сервера. Использование оперативной памяти и процессора можно посмотреть и через Диспетчер задач, но детальное использование сети и жесткого диска можно посмотреть только через resmon и perfmon.
* Системный монитор Windows (Performance Monitor) — Perfmon.exe это основное средство мониторинга Windows, он содержит в себе разнообразные “счетчики”, как системных метрик, так и приложений, включая SQL Server. Обычно счетчики perfmon обрабатывают с помощью других систем мониторинга, например, Zabbix, так как в perfmon неудобно хранить и смотреть данные за прошедшее время.



* Сторонние приложения — существует много платных и бесплатных приложений для мониторинга SQL Server. Например, одним из бесплатных приложений является dbForge Monitor от компании Devart. Приложение устанавливается как дополнение к SSMS и позволяет выводить очень удобный дашборд для отображения текущего состояния вашего SQL Server (информация об использовании памяти, CPU, нагрузках, блокировках, процессах, информацию о бэкапах, “тяжелых SQL запросах”, производительности дисковой подсистемы и т.д.).



* Скрипты Brentozar – это популярное решение для диагностики настроек и работоспособности SQL Server. У brentozar есть много скриптов для различных задач, но для диагностики нас интересует “sp\_blitz”. Скачать можно бесплатно с официального сайта https://www.brentozar.com/blitz/. Запустите sp\_Blitz.sql чтобы установить необходимые процедуры и выполните их exec sp\_blitz для диагностики. Этот инструмент бесплатный и поддерживается сообществом SQL Server. Sp\_blitz определит все популярные проблемы с вашим сервером и посоветует как их решить.



* Наборы T-SQL скриптов — удобно иметь под рукой коллекции разнообразных T-SQL запросов для диагностики SQL Server, так как не всегда есть время писать собственные запросы, лучше вооружиться заранее. <https://github.com/SQLadmin/AwesomeSQLServer>

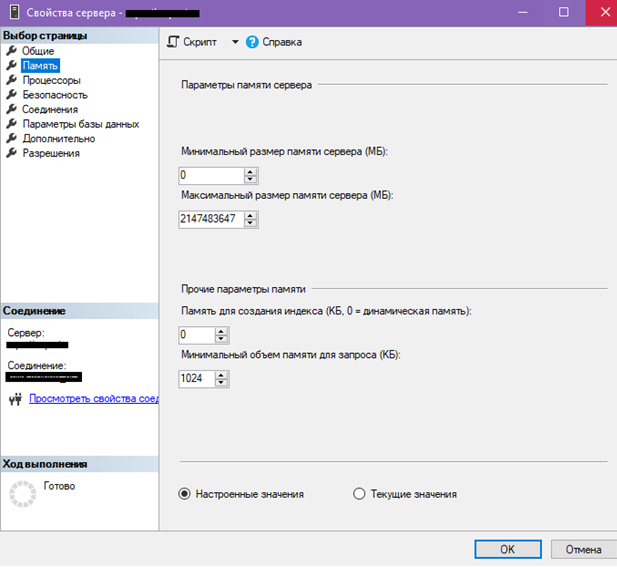
Обнаружение и решение проблем с производительностью SQL Server

Самой распространенной проблемой с которой сталкивается системный администратор, работающий с SQL Server, это жалобы пользователей на производительность запросов и самого сервера: “тормозит”, “долго выполняется запрос“, и так далее.

Прежде всего нужно убедиться, что серверу хватает ресурсов. Рассмотрим, как в SQL Server быстро проанализировать использование памяти, CPU, дисков и наличие блокировок.

Анализ использования оперативной памяти SQL Server

Для начала нужно определить сколько памяти доступно SQL Server. Для этого запустите SSMS (SQL Server Management Studio), зайдите на сервер и зайдите в свойства сервера (ПКМ по названию сервера в Обозревателе объектов).



Сам по себе доступный объём RAM вам ничего не скажет. Нужно сравнить это число с используемой памятью в Диспетчере Задач и самим движком SQL Server с помощью DMV.

В Диспетчере задач, во вкладке Подробности, найдите sqlservr.exe и посмотрите сколько оперативной памяти использует этот процесс.

* Если на сервере, например, 128 GB оперативной памяти, а процесс sqlservr.exe использует 60 GB и ограничений по RAM у SQL Server нет, то оперативной памяти вам хватает.
* Если SQL Server использует 80-90% RAM от заданной или максимальной, то в таком случае нужно смотреть DMV. Имейте в виду, что sqlservr.exe не сможет использовать всю оперативную память. Если на сервере 128 GB, то sqlservr.exe может использовать только 80-90% (100-110 GB), так как остальная память резервируется для операционной системы.

Имейте в виду, что процесс SQL Server’a не отдаёт оперативную память обратно в систему. Например, ваш SQL Server обычно использует 20 GB памяти, но при месячном отчете он увеличивает потребление до 100 GB, и даже когда вычисление отчета закончится и сервер будет работать в прежнем режиме, процесс SQL Server’a всё равно будет использовать 100 GB до перезагрузки службы.

Даже если вы уверены, что оперативной памяти серверу хватает, не будет лишним точно знать объём потребляемой RAM.

Узнать реальное использование RAM можно с помощью **Dynamic Management Views**. DMV это административные вьюверы (представления). С помощью DMV можно диагностировать практически любую проблему в SQL Server.

Посмотрим sys.dm\_os\_sys\_memory, для удобства используем запрос:

SELECT total\_physical\_memory\_kb / 1024 AS

[Total Physical Memory],

available\_physical\_memory\_kb / 1024 AS

[Available Physical Memory],

total\_page\_file\_kb / 1024 AS

[Total Page File (MB)],

available\_page\_file\_kb / 1024 AS

[Available Page File (MB)],

100 - ( 100 \* Cast(available\_physical\_memory\_kb AS DECIMAL(18, 3)) / Cast

(

total\_physical\_memory\_kb AS DECIMAL(18, 3)) ) AS

'Percentage Used',

system\_memory\_state\_desc AS

[Memory State]

FROM sys.dm\_os\_sys\_memory;

Рассмотрим каждый выводимый параметр:

* [Total Physical Memory] – объём оперативной памяти доступный в операционной системе. На некоторых серверах может показывать немного больше реально установленной.
* [Available Physical Memory] – объём оперативной памяти доступный для SQL Server, без учета уже захваченной SQL Server.
* [Total Page File (MB)] – Объём “Сommit limit”. Commit Limit = Оперативная память + все файлы подкачки. То есть, если у вас на сервере 32 GB оперативной памяти и 16 GB файл подкачки, commit limit будет 48 GB.
* [Available Page File (MB)] – Объём файла подкачки.

Percentage Used – процент занятой оперативной памяти. Такого параметра нет в самом sys.dm\_os\_sys\_memory, но он считается по формуле available\_physical\_memory\_kb / total\_physical\_memory\_kb

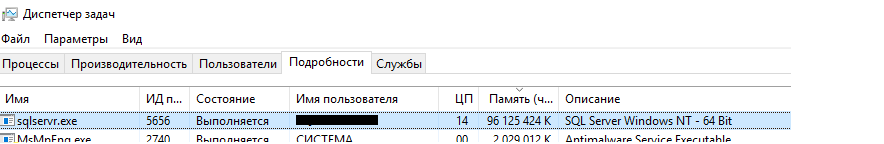
* [Memory State] – Состояние RAM. Поле system\_memory\_state\_desc содержит в себе состояние потребления оперативной памяти в виде текста. Значение этого поля считается исходя из других двух: system\_low\_memory\_signal\_state и system\_high\_memory\_signal\_state. Вы можете выбирать их напрямую, если вам нужен Boolean/bit формат данных.

Все эти данные полезны, если вы хотите точно определить сколько ваш SQL Server потребляет RAM. Чаще всего это используют, если есть подозрения что для экземпляра выделено слишком много оперативной памяти.

Если Вам нужно убедиться, что серверу хватает RAM, вы можете смотреть только на поля **system\_low\_memory\_signal\_state, system\_high\_memory\_signal\_state и system\_memory\_state\_desc.** Если **system\_low\_memory\_signal\_state = 1**, то серверу явно не хватает оперативной памяти.

### Загрузка процессора в SQL Server

Нагрузку на процессор определить проще, так как это можно сделать в Диспетчере задач. Чтобы узнать текущую нагрузку на процессор, найдите в Диспетчере задач процесс sqlservr.exe



Если вы хотите узнать нагрузку за прошедшее время, можно воспользоваться запросом:

DECLARE @ts BIGINT;

DECLARE @lastNmin TINYINT;

SET @lastNmin = 30; // нагрузка за прошедшие n минут

SELECT @ts = (SELECT cpu\_ticks / ( cpu\_ticks / ms\_ticks )

FROM sys.dm\_os\_sys\_info);

SELECT TOP(@lastNmin) Dateadd(ms, -1 \* ( @ts - [timestamp] ), Getdate())AS

[EventTime],

sqlprocessutilization AS

[SQL Server Utilization],

100 - systemidle - sqlprocessutilization AS

[Other Process CPU\_Utilization],

systemidle AS

[System Idle]

FROM (SELECT

record.value('(./Record/@id)[1]', 'int') AS record\_id,

record.value('(./Record/SchedulerMonitorEvent/SystemHealth/SystemIdle)[1]', 'int') AS [SystemIdle],

record.value('(./Record/SchedulerMonitorEvent/SystemHealth/ProcessUtilization)[1]', 'int')AS [SQLProcessUtilization],

[timestamp]

FROM (SELECT[timestamp],

CONVERT(XML, record) AS [record]

FROM sys.dm\_os\_ring\_buffers

WHERE ring\_buffer\_type = N'RING\_BUFFER\_SCHEDULER\_MONITOR'

AND record LIKE'%%')AS x)AS y

ORDER BY record\_id DESC;

В результате мы получим поминутную статистику использования процессора.