

Анализ производительности серверов Microsoft SQL Server в MoLHSA.

Отчет подготовлен для Ministry of Health, Labour and Social Affairs, Georgia.

В первой части отчета предоставлена сводная информация о основных проблемах, связанных со всеми серверами и рекомендации по их устранению.

Вторая часть отчета содержит более подробную и детальную информацию о сервере REPORTING, отмеченный как наиболее важный для заказчика.

Найденные проблемы и рекомендации для их устранения (общие для всех серверов, отмеченных в таблице).

System	Power plan	Huge Memory Grant	Optimizer Memory Leak	Disk Latency	MAXDOP	TOP Waits
Ambulatory	x	x	x		1	CPU
Billing	x	x	x		1	CPU
Case Registration	x	x	x		1	Page I/O latch
Cloud	x		x		1	Page I/O latch
Doctors	x		x		1	Network I/O (Client fetch)
Log	x		x		0	WRITELOG
Referrals	x	x	x		0	CXPACKET
Reporting	x		x	x	1	CPU
User Management	x		x		1	WRITELOG

Issue:

Включен режим энергосбережения (**Power plan**) - Balanced (по умолчанию).

Description:

Для SQL-серверов, особенно с высокой загрузкой рекомендуется использовать план High Performance

Recommendation:

Изменить Power plan на High Performance

Additional information:

[Slow Performance on Windows Server 2008 R2 when using the “Balanced” Power Plan](#)

Issue:

Huge Memory Grant. Выделение большого количества памяти для выполнения запроса

Description:

Для выполнения любого запроса SQL Server выделяет память, не менее значения выставленного в настройках сервера (Minimum memory per query (in KB), default 1024Kb).

После создания плана выполнения, оптимизатор запросов рассчитывает, какое реальное количество памяти нужно для выполнения запроса. Память берется из общего пула, и при отсутствии достаточного количества памяти, процессы, ожидающие память, становятся в очередь. Таким образом, при недостаточном количестве памяти, выделенной серверу СУБД или при выполнении запросов, требующих много памяти (например, запросы, требующие сортировки очень большого количества строк) возникает деградация производительности сервера связанная с отсутствием необходимых ресурсов.

Recommendation:

Необходимо найти запросы, требующие выделения большого количества памяти для их выполнения (memory grant). Для этого можно воспользоваться динамическим представлением [sys.dm_exec_query_memory_grants](#).

Такие запросы требуют оптимизации, например, ограничение входных данных выбором только тех столбцов, которые нужны для выполнения запроса взамен *; исключение ненужных сортировок и др.

Additional information:

[Understanding SQL server memory grant](#)
[SQL Server, Memory Manager Object \(perfmon counters\)](#)

Issue:

На сервере используется SQL Server 2012 RMT, в котором обнаружена возможная утечка памяти **Optimizer memory Leak**.

Description:

В определенных сценариях в SQL Server 2012 RTM может происходить утечка памяти. Проблема устранена в SQL Server 2012 CU1 for Service Pack 1 и выше.

Recommendation:

Рекомендуется установить последний пакет обновлений. На текущий момент это [SQL Server 2012 Service Pack 2](#).

Так же необходимо учитывать, что версия RTM уже не поддерживается Microsoft. Поддержка SQL Server 2012 SP1 заканчивается 15.07.2015

Additional information:

[FIX: Memory leak when you run queries against a temporary table in a nested stored procedure in SQL Server 2012](#)
[Microsoft SQL Server 2012 Support Lifecycle](#)

Issue:

Параметры **Max Degree of Parallelism (MAXDOP)** могут быть настроены не оптимальным образом (настройка сервера). На данном сервере значение параметра =1 (использовать 1 CPU).

Description:

Параметр MAXDOP отвечает за возможность выполнения запроса в несколько потоков. По умолчанию он имеет значение 0, что означает использование всех ядер процессора для выполнения одного запроса, что должно приводить к увеличению производительности. Однако в реальной жизни, особенно для OLTP-систем, это приводит к повышенной загрузке CPU, высокому числу блокировок (BLOCKING) и взаимных блокировок (DEADLOCK). А для синхронизации параллельных потоков требуются дополнительные ресурсы (RAM). А в некоторых случаях время синхронизации этих потоков больше, чем если бы запрос выполнялся в один поток. Именно поэтому важно подобрать оптимальное значение для MAXDOP.

Recommendation:

Значение параметра MAXDOP=1 обычно положительно влияет на производительность OLTP-систем, но все-таки накладывает определённые ограничения на возможности сервера. Например, запросы, выполняющие сканирование таблицы или индекса и имеющие большую стоимость выигрывают от использования параллелизма, операции обновления статистики так же выполняются быстрее, при использовании нескольких CPU (использовать опцию OPTION MAXDOP=XX для UPDATE STATISTICS невозможно, в отличие от ALTER INDEX REBUILD).

Таким образом, рекомендуется рассмотреть возможность увеличить значение MAXDOP, например, до 4 и одновременно с этим изменить значение **Cost Threshold For Parallelism** в большую сторону (по умолчанию = 5). Значение параметр **Cost Threshold For Parallelism** указывает на стоимость запроса, ниже которого параллелизм использовать не нужно (стоимость можно посмотреть в планах выполнения запросов). Если подобрать оптимально значение этого параметра, и увеличить MAXDOP, то запросы с высокой стоимостью будут выполняться быстрее, а запросы, которым это не требуется, будут выполняться в один поток.

Additional information:

[Recommendations and guidelines for the "max degree of parallelism" configuration option in SQL Server](#)

[Configure the cost threshold for parallelism Server Configuration Option](#)

[OLTP Blueprint - A Performance Profile of OLTP Applications](#)

Issue:

Высокое ожидание **WRITELOG** на нескольких серверах.

Description:

WRITELOG указывает на время, которое SQL-сервер ожидает завершения записи транзакций в журнал транзакций (на диске) из памяти. Наличие быстрого диска для журнала транзакций позволяет снизить задержки выполнения транзакций.

WRITELOG waits represents the time, SQL Server is waiting on the I/O to complete after a Log block in cache is flushed from Memory to Disk (Thus why having a ultra-fast disk like Flash based disks to store your LDF can speed up this I/O and it reduces your transaction latency)

Recommendation:

Для серверов, где журнал транзакций размещен на локальном диске C или на диске с базами и tempdb необходимо вынести на отдельный LUN с минимальным временем отклика (по возможности <5ms).

При помощи функции fn_virtualfilestats или динамического представления sys.dm_io_virtual_file_stats можно выявить файлы с наибольшей задержкой (файлы журналов транзакций). Это первые кандидаты для размещения на отдельном диске.

Additional information:

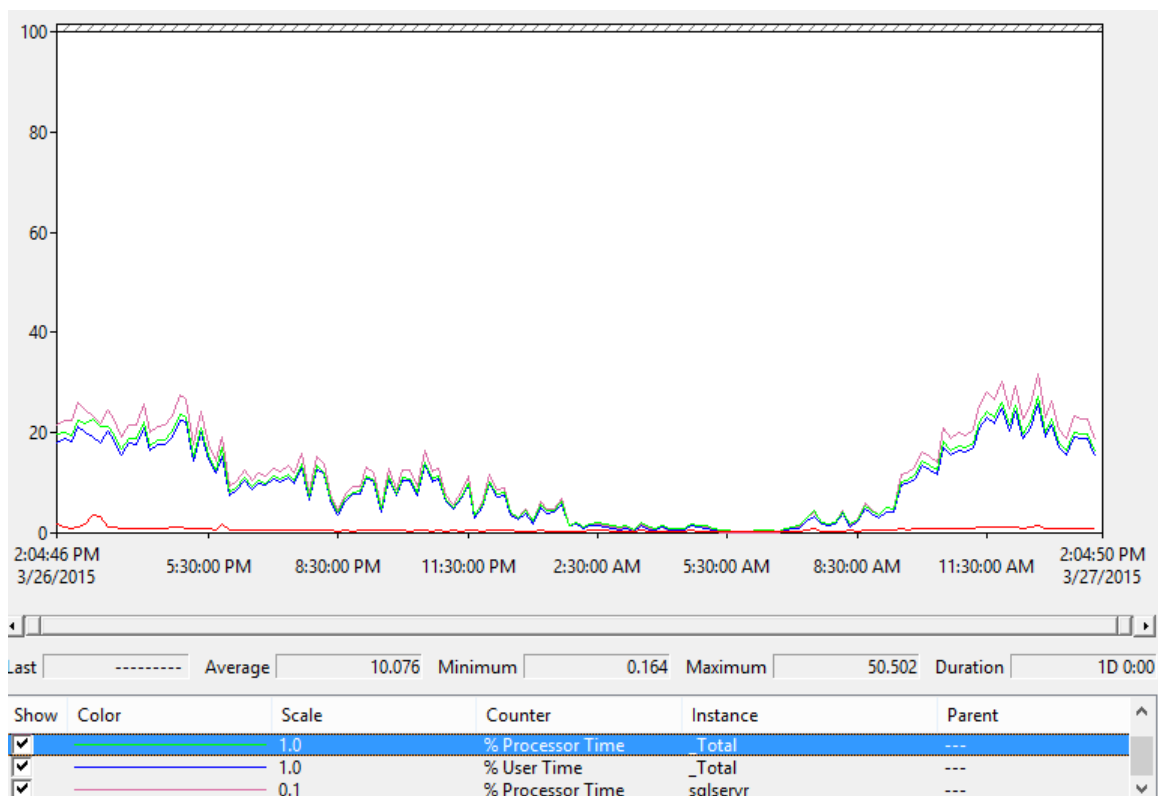
[Diagnosing Transaction Log Performance Issues and Limits of the Log Manager](#)

Найденные проблемы и рекомендации для их устранения - REPORTING

1. Общее состояние ресурсов (Reporting).

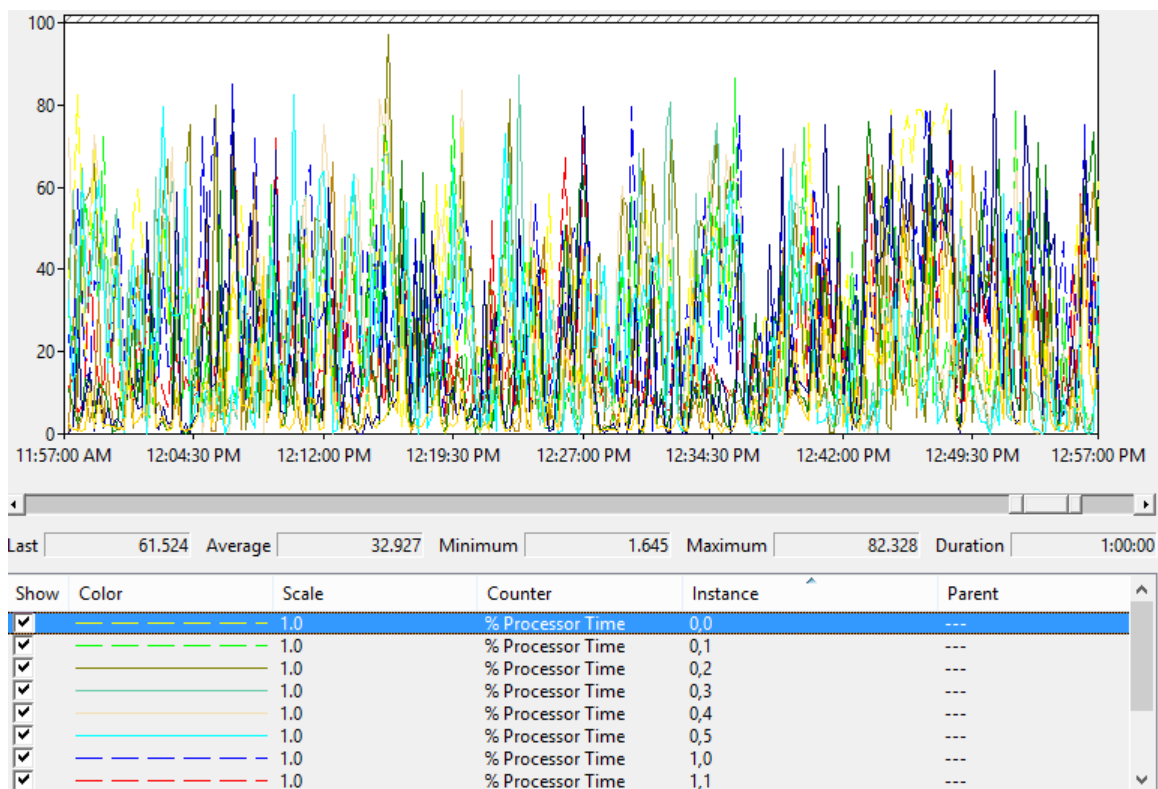
CPU

Общая загрузка CPU в допустимых пределах. Большая часть нагрузки приходится на дневное время и не превышает 50%. (длительная нагрузка выше 60-80% требует пристального наблюдения, выше 80% считается критичной)



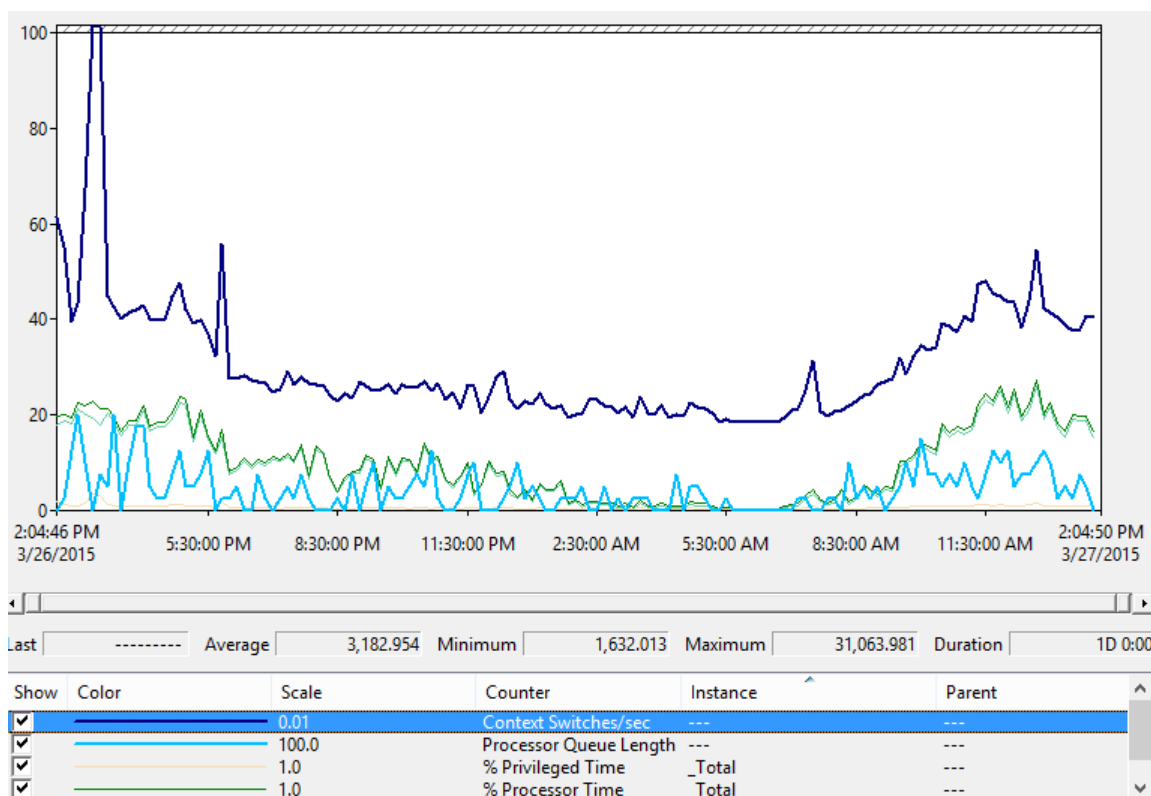
Основной потребитель CPU - SQL Server.

Средняя загрузка CPU для каждого из ядер 20-35%. Несмотря на отдельные пики, длительная загрузка отдельных ядер не обнаружена.



Иногда есть незначительные очереди к ресурсам CPU (не более 3-х, один из показателей возможного недостатка ресурсов CPU).

Число контекстных переключений в пределах допустимой нормы (около 31000, для данного сервере рекомендуется не более $50000 = 5000 \times \text{number of CPU}$). Большое число контекстных переключений может указывать на проблемы с драйверами устройств, в результате чего, ресурсы CPU тратятся на переключения между привилегированным и пользовательским режимами, вместо выполнения полезной работы.



Вывод:

Проблем не обнаружено.

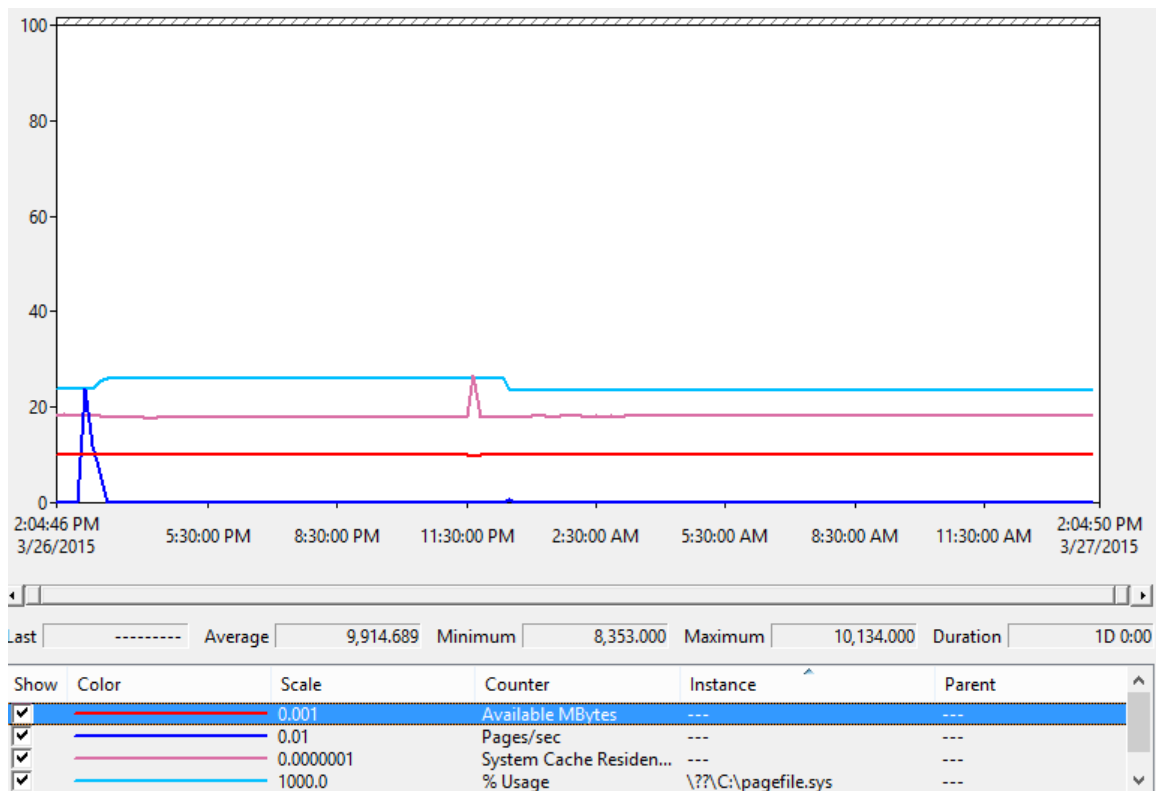
На текущий момент процессорных ресурсов достаточно для нормальной работы.

RAM

На сервере достаточное количество доступной памяти (8Гб в минимуме, рекомендуется не менее 300Мб, счетчик Memory\Available Mbytes). Изменения в течении суток незначительные.

Файл подкачки фактически не используется.

Файловый кэш используется максимум до 350Мб.



Вывод:

На сервере достаточное количество физической доступной памяти.

Рекомендуется часть памяти отдать серверу SQL, но не более 6Гб. В дальнейшем необходим мониторинг доступной памяти. Крайне желательно не допускать снижение этого показателя ниже 64Мб, так в этом случае наступит Low-Memory Condition, при котором ОС начинает агрессивно урезать рабочие наборы памяти процессов (в том числе и сервера SQL), освобождая память. Часть памяти сбрасывается в файл подкачки.

Полезные ссылки:

[Preventing low memory problems](#)

[Evaluating Memory and Cache Usage](#)

[How to reduce paging of buffer pool memory in the 64-bit version of SQL Server](#)

DISKS

Рекомендуемые задержки для дисков, на которых распложены файлы СУБД - 4-20 ms, 1-5 ms для журналов транзакций.

Общие рекомендации и в частности для дисков можно посмотреть здесь:

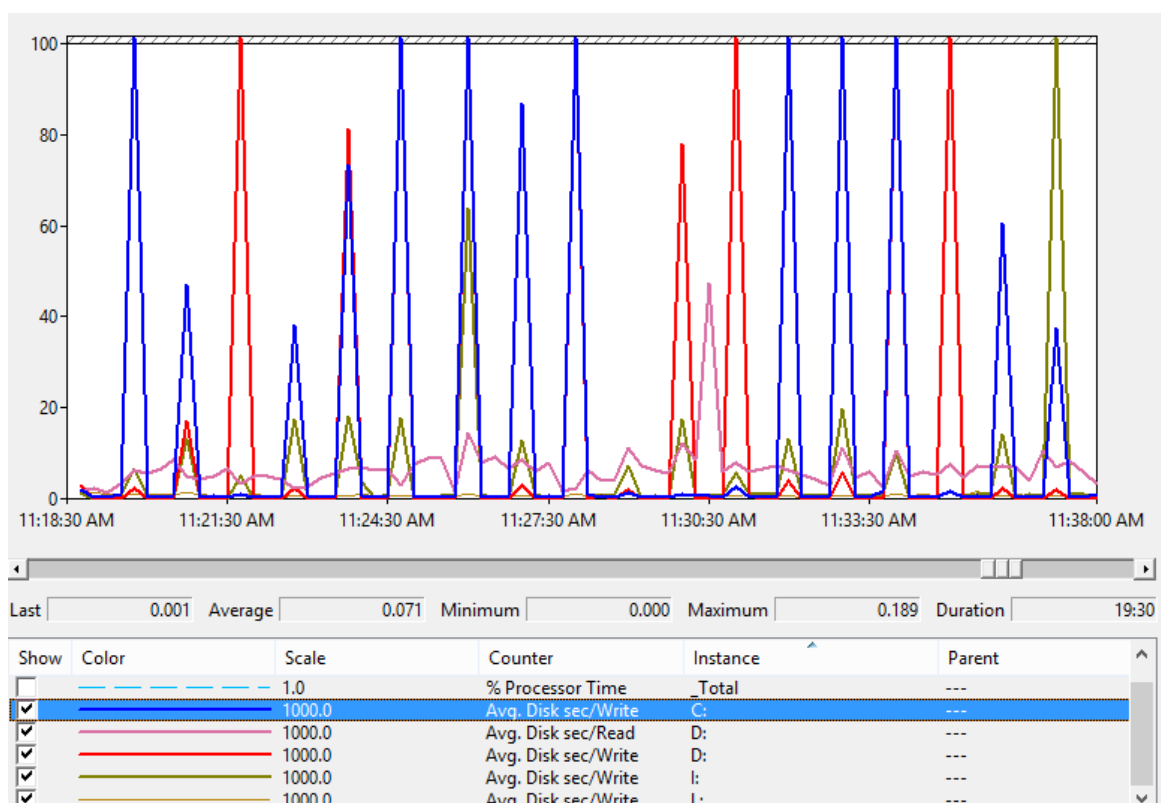
[SQL Server Best Practices Article](#)

[Storage Top 10 Best Practices](#)

На данном сервере зафиксированы значительные задержки дисков, в десятки раз превышающие рекомендуемые значения.

Сводная таблица дисковых задержек в рабочие часы:

Disk	Read latency		Write latency	
	Average	maximum	Average	maximum
C	4	11	71	189
D	8	47	72	168
I	3	10	42	276
L	1	2	1	2



Вывод:

Дисковая подсистема не удовлетворяет требованиям СУБД MS SQL и не справляется с текущим IO.

Основные задержки связаны с операциями записи.

Рекомендуется СХД с лучшими характеристиками (большее количество поддерживаемых IOPS).

Необходимо рассмотреть вариант размещения файлов БД на большем количестве дисков (отдельных LUN).

NETWORK

Текущая утилизация сетевого интерфейса (1Gbps) на отправку данных достигает 57% , что близко к критическим значениям (>60% - Critical). На прием - 2%.

Со стороны SQL Server так же присутствуют ожидания, связанные с сетью.

Вывод:

Необходим сетевой интерфейс с большей пропускной способностью.

2. Другие проблемы (Reporting).

Issue:

Некоторые файлы данных размещены на диске C (бд HMIS_Reporting), некоторые файлы журналов транзакций размещены на одном диске с данными (диск I: и L:, базы данных HMIS_Reporting и tempdb)

Description:

В файлы журналов транзакций в основном производится последовательная запись и время отклика (latency) рекомендуется <5ms, так как от этого зависит время выполнения транзакций. С файлами данных выполняются операции случайного (random) чтения/записи, а соотношение (для оптимизации кэша) можно вычислить при помощи счетчиков perfmon: LogicalDisk\Disk reads/sec и Disk writes/sec. БД tempdb чаще всего является одной из наиболее загруженной на экземпляре. Именно поэтому рекомендуется размещать файлы данных, журналы транзакций и tempdb (data+log) на отдельных разделах и в идеале на отдельных физических LUN.

Recommendation:

Переместить файлы данных бд HMIS_Reporting с диска C на отдельный раздел. Разместить файлы данных и журналы транзакций на отдельных дисках. Базу tempdb разместить на отдельном диске.

Additional information:

[Place Data and Log Files on Separate Drives](#)

[Separate database and transaction log files on different drives for optimal performance and disaster recovery](#)

[Moving Database Files](#)

Issue:

Количество создаваемых Workfiles Created/sec выше рекомендуемых (>20 per sec).

Description:

Workfiles - объекты, создаваемые в tempdb для хранения результатов хэширования при операциях соединения (JOIN) в случае недостаточного количества оперативной памяти для их размещения. Операции HASH JOIN сами по себе являются требовательными к ресурсам (CPU, I/O), а файл tempdb более медленный ресурс, чем RAM. В результате получаем деградацию производительности.

Как правило, причиной создания большого количества Workfiles является отсутствие необходимых индексов для выполнения операций JOIN на таблицах, участвующих в соединении.

Recommendation:

Необходимо выявить запросы, использующие большее количество ресурсов (CPU, I/O). Проанализировать их планы выполнения на предмет использования операций HASH JOIN (особенно с событиями Hash warning). Построить (изменить существующие) индексы по полям, которые используются в предикате соединения. Возможно изменить код запросов для ограничения количества входных данных.

Так же существует вероятность устранения проблемы добавлением оперативной памяти серверу SQL.

Additional information:

[SQL Server Perfmon Counters of Interest](#)

Issue:

На дисках существенные задержки (Disk Latency).

Об это говорят показания счетчиков perfmon, так же в SQL-сервер присутствуют задержки PAGEIOLATCH_XX для БД Hmis_Reporting для ряда файлов.

В журналах ошибок зафиксировано большое количество предупреждений для диска D типа: *2015-03-22 16:41:47.060 spid8s SQL Server has encountered 423 occurrence(s) of I/O requests taking longer than 15 seconds to complete on file [D:\Data\Hmis_Reporting_Sec_61.ndf] in database [Hmis_Reporting] (7). The OS file handle is 0x00000000000000C4C. The offset of the latest long I/O is: 0x00000126f80000*

Description:

Минимальное время отклика дисковой подсистемы очень важно для SQL-сервера.

Общая рекомендация - < 8 ms.

Для дисков, на которых размещены журналы транзакций и tempdb рекомендуется <= 5 ms. Значения выше > 20 ms являются критичными для SQL Server и требуют анализа и устранения.

Оценить задержки дисков можно при помощи счетчиков LogicalDisk\Avg. Disk sec/Reads и Avg. Disk sec/Writes

Ожидания PAGEIOLATCH_XX возникают при чтении данных в буферный пул (data cache).

Длительные ожидания указывает на проблемы с дисковой подсистемой (как правило на задержки чтения с диска).

Увеличение количества памяти серверу SQL так же снижает количество дисковых операций в результате более эффективного кэширования страниц в буферном пуле. В общем случае, если используются настройка Max Server Memory и в ОС достаточное количество доступной физической памяти, то часть памяти можно выделить серверу SQL.

Recommendation:

Необходимо уменьшить время отклика дисковой подсистемы и добавить памяти серверу SQL.

Рекомендуется разместить файлы данных, журналы транзакций и tempdb на отдельных дисках. Далее необходимо идентифицировать диски с наибольшим временем отклика. При помощи SQL запроса с использованием функции fn_virtualfilestats или динамического представления sys.dm_io_virtual_file_stats необходимо выявить наиболее нагруженные файлы на диске с высокими задержками. Эти файлы могут быть кандидатами для размещения на отдельном диске (LUN). В случае необходимости вынести эти файлы на отдельный LUN.

Так же рекомендуется выявить запросы, создающие максимальную нагрузку на дисковую подсистему (большое количество физических и логических чтений). Такие запросы желательно оптимизировать: сократить размер выборки, создать недостающие индексы. Добавление оперативной памяти серверу SQL так же может снизить количество дисковых операций (память, выделенная SQL Server, не путать с памятью в ОС). На текущем сервере доступная физическая память - 8Гб. Необходимо изменить параметры Max Server Memory с 21Гб до 26Гб. На данном сервере это возможно, так как в ОС есть около 9Гб доступной памяти.

Additional information:

По возможности измерить соответствие дисковой подсистемы при помощи SQLIO.exe или Iometer.exe (<http://www.iometer.org>) в часы минимально нагрузки сервера.

Issue:

Зафиксированы события автоматического прироста файлов в рабочие часы.

В журнале ошибок зафиксированы предупреждения:

2015-03-22 11:47:46.800 spid60 Autogrow of file 'tempdev_1' in database 'tempdb' took 62870 milliseconds. Consider using ALTER DATABASE to set a smaller FILEGROWTH for this file.

Description:

Рекомендуется создавать файлы данных и журналы транзакций с некоторым запасом, чтобы события автоприроста возникали реже, а в идеале чтобы не возникали никогда. В момент прироста файл блокируется в файловой системе и до завершения операции и в него нельзя писать/читать.

Для файлов данных время блокирования можно сократить включив [Instant File Initialization](#). Так же важно правильно подобрать параметры автоматического прироста. Рекомендуется использовать фиксированные размеры вместо % и не более 1Гб (для файлов данных с включенной мгновенной инициализацией файлов может быть больше).

Recommendation:

[Включить мгновенную инициализацию файлов](#), если это еще не сделано, добавив учетную запись, под которой работает служба SQL в локальную политику Perform volume maintenance tasks.

Изменить параметры автоматического прироста. Для журналов транзакций значение не должно превышать 1Гб.

Additional information:

[Considerations for the “autogrow” and “autoshrink” settings in SQL Server Database File Initialization](#)

Issue:

Initial Size для tempdb очень маленький для текущего сервера.

В процессе работы сервера происходит регулярный рост файлов tempdb вызывающий деградацию производительности.

В журнале ошибок зафиксированы предупреждения:

2015-03-22 11:47:46.800 spid60 Autogrow of file 'tempdev_1' in database 'tempdb' took 62870 milliseconds. Consider using ALTER DATABASE to set a smaller FILEGROWTH for this file.

Это 62 секунды на прирост!

Description:

Автоматический прирост файлов всегда приводит к деградации производительности, и особенно файлов tempdb. Очень важно правильно подобрать начальный размер tempdb, который используется при создании бд tempdb в момент старта сервера SQL.

Recommendation:

Необходимо увеличить начальный размер файлов tempdb до их "рабочего состояния" (требуется мониторинг размера tempdb в течении некоторого времени). Предварительно необходимо включить мгновенную инициализацию файлов.

Для журнала транзакций очень большой размер лучше не задавать (более 10Гб).

Additional information:

[Optimizing tempdb Performance](#)