

Microsoft®
tech·ed
Russia | 2011

9-10 НОЯБРЯ 2011
МОСКВА

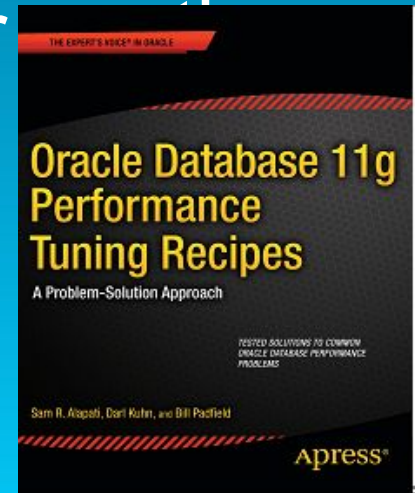


Оптимизация инфраструктуры SQL Server

Дмитрий Артемов
Старший консультант
dima@microsoft.com



Analyzing **Oracle** wait events is the most important performance tuning task you'll perform when troubleshooting a slow-running query. When a query is running slow, it usually means that there are excessive waits of one type or



Зачем я здесь?

Первая из двух презентаций, в которых я постараюсь дать сводную картину инструментария, доступного в SQL Server 2008 R2 для анализа ситуации, выявления проблем и оптимизации инфраструктуры и приложения (кода и индексной схемы)



План

- Введение – DMV
 - DMV уровня платформы
- Оптимизация конфигурации SQL Server
 - Как определить текущую конфигурацию сервера и SQL Server
 - Что может оказать существенное влияние на производительность
 - Как найти ошибки в конфигурации
- Оптимизация дисковой подсистемы
 - Что влияет на производительность
 - Какими средствами определить проблемы
- Оптимизация работы Tempdb
 - Особенности организации работы с данными в Tempdb
 - Как используется Tempdb и как это влияет на производительность
 - Какими средствами можно определить нагрузку в Tempdb
- Взаимодействие с внешним миром
- Интерпретация результатов от DMV
- Как увязать вместе все что открывает SQL Server
 - С чего начать



Введение – DMV/DMF

- DMV/DMF – системные представления/функции, позволяющие заглянуть внутрь SQL Server
 - SQL Server 2008 R2 + SP1: 141 штука
 - Не все описаны в документации, которая стоит у вас на компьютере
 - Заглядывайте в Интернет
 - SQL Server 2012 : 174 штуки
- Именованы по подсистемам: dm_db / os / io / exec...*
- В этой части мы будем в первую очередь рассматривать DM_OS_*, DM_IO_*, частично DM_EXEC_*
- Вторая презентация будет рассматривать DM_EXEC_*, DM_TRAN_*, DM_DB_*,...
- Деление довольно условное



Конфигурация



Инструменты определения конфигурации

- SP_CONFIGURE – по-прежнему важный инструмент
- Представления, открывающие конфигурацию системы
- Часть представлений – чисто информационные
- sys.dm_os_sys_info – общая информация о системе
 - Поля virtual_machine_type, virtual_machine_type_desc почему-то возвращают 1, HYPERVISOR даже на физической машине



Текущая конфигурация

- sys.dm_os_windows_info информация по ОС
 - SQL Server 2008 R2 SP1
- sys.dm_server_registry – список значений ключей реестра (для текущего экземпляра SQL Server)
 - SQL Server 2008 R2 SP1
 - В документации масса ошибок в именах полей
- sys.dm_server_services – список установленных служб SQL Server, их состояние и настройка
 - SQL Server 2008 R2 SP1

```
DECLARE @returnValue NVARCHAR(500)
EXEC master..xp_instance_regread
    @rootkey = N'HKEY_LOCAL_MACHINE',
    @key = N'SOFTWARE\Microsoft\BestPractices',
    @value_name = N'ModelsRoot', @value = @returnValue output
PRINT @returnValue + 'НЕ ДОКУМЕНТИРОВАННОЕ И НЕ
ПОДДЕРЖИВАЕМОЕ'
```

registry_key	value_name	value_data
HKLM\SYSTEM\CurrentControlSet\Services\MSSQL\$...	ObjectName	.\serviceuser
HKLM\SYSTEM\CurrentControlSet\Services\MSSQL\$...	ImagePath	"C:\Program Files\Microsoft SQL Server\MSSQL10_50_KATMAI\MSSQL\Binn\sqlservr.exe" -sKATMAI
HKLM\SYSTEM\CurrentControlSet\Services\MSSQL\$...	Start	3
HKLM\SYSTEM\CurrentControlSet\Services\SQLAge...	ObjectName	.\serviceuser
HKLM\SYSTEM\CurrentControlSet\Services\SQLAge...	ImagePath	"C:\Program Files\Microsoft SQL Server\MSSQL10_50_KATMAI\MSSQL\Binn\SQLAGENT.EXE" -i KATMAI
HKLM\SYSTEM\CurrentControlSet\Services\SQLAge...	Start	3
HKLM\SYSTEM\CurrentControlSet\Services\SQLAge...	DependOnService	MSSQL\$KATMAI
HKLM\Software\Microsoft\Microsoft SQL Server\MSS...	CurrentVersion	10.50.2500.0
HKLM\Software\Microsoft\Microsoft SQL Server\MSS...	SQLArg0	-dE:\SQLData\SQL2008\MSSQL10_50_KATMAI\MSSQL\DATA\master.mdf
HKLM\Software\Microsoft\Microsoft SQL Server\MSS...	SQLArg1	-eE:\SQLData\SQL2008\MSSQL10_50_KATMAI\MSSQL\Log\ERRORLOG
HKLM\Software\Microsoft\Microsoft SQL Server\MSS...	SQLArg2	-H E:\SQLData\SQL2008\MSSQL10_50_KATMAI\MSSQL\DATA\mastlog.ldf
HKLM\Software\Microsoft\Microsoft SQL Server\MSS...	TcpDynamicPorts	63490
HKLM\Software\Microsoft\Microsoft SQL Server\MSS...	DisplayName	TCP/IP
HKLM\Software\Microsoft\Microsoft SQL Server\MSS...	Enabled	0

servicename	startup_type	startup_type_d...	status	status_de...	process...	last_startup_time	service_acc...	filename
SQL Server (KATMAI)	3	Manual	4	Running	3316	2011-09-20 09:00:07.4719326 +04:00	.\serviceuser	"C:\Program Files\Micr...
SQL Server Agent (KATMAI)	3	Manual	1	Stopped	NULL	NULL	.\serviceuser	"C:\Program Files\Micr...

NUMA, память

- `sys.dm_os_nodes` – NUMA конфигурация сервера
 - Маски Affinity для узлов NUMA (`cpu_affinity_mask`)
 - Загрузка по узлам (`online_scheduler_count`, `idle_scheduler_count`, `active_worker_count`, `avg_load_balance`)
- `sys.dm_os_sys_memory` – данные об установленной и используемой памяти
 - Поле `system_memory_state_desc` показывает общее состояние памяти SQL Server :

<code>system_low_memory_signal_state</code>	<code>system_memory_state_desc</code>
0	Available physical memory is high
 - Available physical memory is high – памяти достаточно
 - Available physical memory is low – памяти не хватает
 - Physical memory state is steady – состояние стабильно, все в норме
 - Physical memory state is transitioning – переходное состояние
- `available_physical_memory_kb`, `available_page_file_kb` помогают понять все ли хорошо с точки зрения памяти

Память

- `sys.dm_os_process_memory`
 - Информация об использовании памяти SQL Server – как процесса операционной системы
 - `physical_memory_in_use` - общее потребление памяти (включая AWE и большие страницы)
 - `large_page_allocations_kb` – использование памяти, набранной большими страницами
 - `locked_pages_allocations_kb` – использование памяти, полученной через AWE API
 - `process_physical_memory_low = 1` внешнее давление по памяти
 - `memory_utilization_percentage` ниже 100% при нормальной нагрузке может быть сигналом о необходимости расследования, нет ли активных процессов, отнимающих (пытающихся отнять) память у SQL Server

AWE или ... AWE

- До сих пор у SQL Server есть настройка “AWE Enabled”
- Она есть и в x86 и в x64 версиях
- Как быть?
- Документация говорит:
 - Support for AWE...only applies to 32-bit operating systems
 - Although it is not required, we recommend locking pages in memory when using 64-bit operating systems
- И как все это вяжется?
- В ситуации с x64 мы используем AWE API для выделения памяти, а ему нужна привилегия “Lock pages in memory”
- Это совсем не значит что без привилегии мы не сможем работать с большими объемами, просто так быстрее и нет риска Swapping

А “awe enabled” на x64 мы просто игнорируем 😊

AWE – lock pages in memory

- SQL Server Standard edition поддерживает этот функционал, начиная с SQL Server 2008 SP1 + Cumulative Update 2
- Подробности см:
<http://support.microsoft.com/kb/970070>



Дисковая подсистема



Основные требования

- Хорошо – задержки до 20 мсек на файлах данных, до 10 мсек на журнале транзакций
- Выравнивание секторов (Windows до 2003 включительно)
- 64К размер сектора
- Побольше шпинделей для OLTP
- Пошире канал ввода/вывода для аналитических/отчетных систем
- И то и другое для смешанных приложений

**Тебе что намазать?
Мёду или сгущенного
молока?**

**И того, и
другого, и
можно без
хлеба.**



Размещение

- Сколько чего где лежит
- Часто необходимо определить число записей в таблице или объем хранения на диске
 - `SP_SPACEUSED` – по всей БД или для конкретного объекта
 - `sys.dm_db_partition_stats` – более детальная информация с возможностью агрегирования как вам удобно
 - Из этого представления можно извлечь число записей, гораздо быстрее чем средствами `SELECT COUNT(*)`, но с некоторой погрешностью т.к. обновление делается асинхронно



Новые средства

- `sys.dm_os_volume_stats (database_id, file_id)`
 - Появился в SQL 2008 R2 SP1
 - Дает информацию о логических дисках, где лежат файлы БД
 - Удобен для мониторинга свободного пространства

<code>total_bytes</code>	Общий размер тома в байтах.
<code>available_bytes</code>	Доступное свободное место тома.
<code>supports_compression</code>	Поддерживает ли том сжатие на уровне операционной системы.
<code>supports_alternate_streams</code>	Поддерживает ли том дополнительные потоки.
<code>supports_sparse_files</code>	Поддерживает ли том разреженные файлы.
<code>is_read_only</code>	Помечен ли том как доступный только для чтения.
<code>is_compressed</code>	Сжат ли том в настоящее время.

Дисковая активность

- Представления показывают дисковую активность
 - Logical read – чтение из кеша
 - Physical read – чтение с диска
- `sys.dm_io_pending_io_requests`
 - Текущее состояние ожиданий на дисковые операции
 - `io_pending = 1` означает, что ожидание на стойке, `= 0` – стойка выполнила свою работу. SQL Server не может быстро обработать результат
 - `io_type` – тип задержки Network/Disk



Нагрузка

- `sys.dm_io_virtual_file_stats` – накопленная с момента старта SQL Server информация о дисковых операциях
 - На уровне индивидуальных файлов
 - Показывает накопленные задержки ввода/вывода
 - `select * from sys.dm_io_virtual_file_stats(-1,-1)` – показывает по всем БД и всем файлам
 - `select * from sys.dm_io_virtual_file_stats(DB_Id,File_id)` – по конкретному файлу конкретной БД
- ```
select io_stall_read_ms/num_of_reads [Delay], *
from sys.dm_io_virtual_file_stats(-1,-1) order by
[Delay];
```

  - Показывает время отклика стойки на конкретных файлах
  - Perfmon показывает задержки на уровне логического диска
- Для выяснения задержек за некий период снимаем «до» и «после» – вычисляем дельту

# Обслуживание

- Фрагментация
  - Всегда ли она вредна? (да, всегда)
  - Всегда ли от нее нужно избавляться? (нет, не всегда)
  - Уровень фрагментации определяется из `sys.dm_db_index_physical_stats` – подробнее в следующей презентации



# Секционирование

- Секционирование
  - Помогает в первую очередь при массивных перемещениях данных
  - Может помочь при очень интенсивных вставках
    - Если значение поля секционирования монотонно увеличивается
  - Может помочь при манипулировании кусками огромных таблиц
  - Хорошо совмещается с Filestream
  - Важно найти оптимальный ключ секционирования
  - Манипулирование секциями требует блокировок, несовместимых с (нормальной) жизнью (клиентов)

|       | NL | SCH-S | SCH-M | S | U | X | IS | IU | IX | SIU | SIX | UIX | BU |
|-------|----|-------|-------|---|---|---|----|----|----|-----|-----|-----|----|
| NL    | N  | N     | N     | N | N | N | N  | N  | N  | N   | N   | N   | N  |
| SCH-S | N  | N     | C     | N | N | N | N  | N  | N  | N   | N   | N   | N  |
| SCH-M | N  | C     | C     | C | C | C | C  | C  | C  | C   | C   | C   | C  |

**Key**

- N No Conflict
- I Illegal
- C Conflict

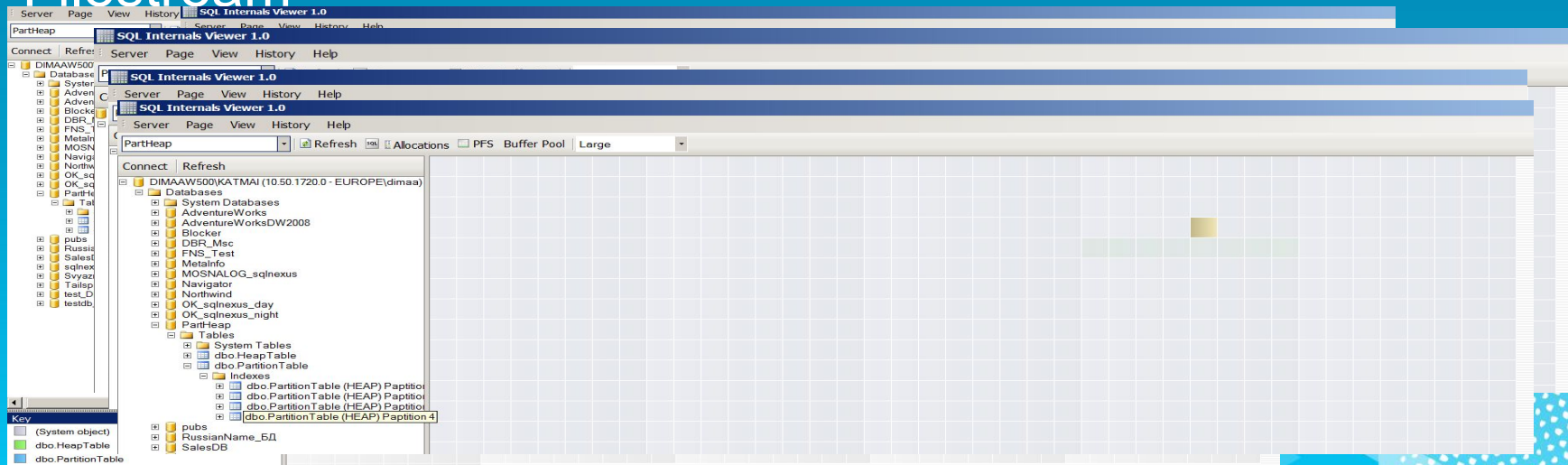
← Это нужно INSERT, UPDATE, SELECT, DELETE

← Это нужно для MERGE/SPLIT

- Так, что хоть сама операция выполняется быстро, ждать своей очереди можно долго

# Секционирование

- Кластерный индекс не обязателен
- SQL Server создает отдельные структуры хранения под каждую секцию (внутри одного или разных файлов). В том числе отдельные каталоги для Filestream





# Filestream

- Вставка
  - Медленнее (примерно в два раза, по моим тестам) чем при использовании (N)Varchar(MAX) или XML
- Выборка (не тестировал, возможно также медленнее)
- Обслуживание
  - Единственный способ физически разделить реляционное хранение и массивные неструктурированные данные для секционированной таблицы
- Перемещение
  - Объявление файловой группы как ReadOnly позволяет переносить по желанию через Backup-Restore



# Tempdb



# TempDb

У нас только одна TEMPDB на всех, Берегите ее.

- Хранит явно созданные временные объекты: глобальные или локальные временные таблицы, временные хранимые процедуры, табличные переменные, некоторые типы курсоров.
- Внутренние объекты, созданные самим SQL Server Database Engine
  - Рабочие таблицы для DBCC CHECKDB и DBCC CHECKTABLE.
  - Рабочие таблицы для hash операций (join и aggregation).
  - Рабочие таблицы для статических или keyset курсоров.
  - Рабочие таблицы для обработки объектов Service Broker.
  - Рабочие файлы для обеспечения операций GROUP BY, ORDER BY, UNION, SORT и SELECT DISTINCT.
  - Рабочие файлы для сортировки при создании или перестройки индексов (при указании SORT\_IN\_TEMPDB).
- Версии записей при использовании READ\_COMMITTED\_SNAPSHOT или явном указании SNAPSHOT ISOLATION.
- Версии записей для: online index операции, (MARS) и AFTER триггеры.

# Три основные проблемы при работе с TEMPDB:

- Дисковые очереди на TEMPDB, производительность страдает.
  - Наиболее частая проблема.
- Очереди на работу с основными страницами метаданных в TEMPDB. При интенсивном создании временных объектов. Любые изменения в распределении пространства требует наложения latch на страницы PFS, GAM или SGAM для отражения изменений. Множество таких операций создает «горячие точки» и тормозит приложение.
  - Обычно характерно для OLTP.
- Кончилось место в TEMPDB.
  - С каждым может случиться.



# Что мы можем узнать о

# TempDb

sys.dm\_db\_file\_space\_usage – на что выделено пространство в файле БД  
Работает только для TempDb

- В SQL Server 2012 – для любой БД

**SELECT**

```
SUM (user_object_reserved_page_count) * 8 as usr_obj_kb,
SUM (internal_object_reserved_page_count) * 8 as internal_obj_kb,
SUM (version_store_reserved_page_count) * 8 as version_store_kb
FROM sys.dm_db_file_space_usage
```

- Все ниже перечисленное может не являться проблемой, но знать об этом нужно:
  - Значительный % под пользовательские объекты означает, что имеется потенциальный риск создания «горячих точек» на страницах метаданных.
  - Значительный % под внутренние объекты означает, что планы интенсивно используют TEMPDB. Следует отыскать такие планы.
  - Значительный % под хранилище версий означает, что очистка хранилища версий не справляется с работой. Новые версии поступают слишком быстро. Посмотрите, нет ли слишком длительных транзакций, возможно, интенсивность транзакций слишком велика

# Взаимодействие с внешним миром



# Сессии

- Мы все знаем `master.dbo.sysprocesses`
- Теперь появились `sys.dm_exec_sessions` и `sys.dm_exec_requests`
- Но, старый конь борозды не портит:
  - Показывает `sql_handle` для последнего выполненного запроса на неактивных сессиях
  - Напрямую показывает `Id` потока ОС для соединения
  - Показывает дочерние потоки сессий при параллельном исполнении запросов
  - Показывает открытые транзакции для неактивного соединения (новые DMV этого не умеют 😞)





# Сессии

- `sys.dm_exec_connections` – показывает текущие подключения к серверу
- Три поля уникально идентифицируют соединение
  - `connection_id` – уникально идентифицирует соединение на уровне экземпляра, используется для соединения с `sys.dm_exec_requests`
  - `session_id` – идентифицирует сессию Ю связанную с соединением, используется в качестве foreign key для ссылки на `sys.dm_exec_sessions` соединения с `sys.dm_exec_connections`
  - `most_recent_session_id` – идентификатор (`session_id`) последнего запроса, связанного с соединением
- Можно выяснить активность соединения НО...

```
select session id, num reads, num writes from sys.dm exec connections where session id = 52
```


| session_id | num_reads | num_writes |
|------------|-----------|------------|
| 1          | 52        | 34         |

← Данные меняются при каждом запросе

```
select session id, logical reads, writes from sys.dm exec sessions where session id = 52
```

| session_id | logical_reads | writes |
|------------|---------------|--------|
| 1          | 52            | 193    |

← Данные меняются только при выполнении реальной работы



Данные о чтении/записи (`num_reads`, `num_writes`) выражены как число сетевых пакетов, которые переданы в рамках соединения

# Сессии

- `sys.dm_exec_sessions` – возвращает информацию о текущих сессиях на сервере
- В зависимости от ситуации (EXECUTE AS) отображает различные данные по пользователю
- Обновляется только после завершения запроса
- Поле `status` может иметь четыре значения:
  - `Running` – во время фактического выполнения запроса (активно использует CPU)
  - `Sleeping` – ждет выделения CPU или нового задания
  - `Dormant` – при очистке (`reset`) сессии при использовании пула соединений
  - `Preconnect` – проходит проверку в классифицирующей функции Resource Governor
- Интересным может быть поле `transaction_isolation_level`
- Поля `original_login_name`, `original_security_id` позволяют определить подмену контекста
- Соединение с `sys.dm_tran_session_transactions` позволяет определить зависшие транзакции
- Фильтровать можно по `session_id > 50` или по `is_user_process = 1`
  - Некоторые системные процессы (например Service broker) могут открывать сессии с номером более 50

# Интерпретация результатов



# Если что-то работает медленно

- Недостаток памяти
- Дисковые задержки
- Слишком тяжело в TempDb
- Может быть банально слишком слабый сервер/дисковая подсистема
- Могут быть неверные настройки
- А может быть, “это всё оттого, что *кто-то слишком много ест!*”
  - За счет чего он “ест” слишком много ресурсов?
  - Чаще всего проблемы в приложении



# Куда смотреть (на уровне сервера)

- В первую очередь определить где мы стоим
  - `sys.dm_os_wait_stats` / `sys.dm_os_waiting_tasks`
  - Счетчики монитора производительности (perfmon)
- Недостаток памяти
  - `sys.dm_os_process_memory`
- Не справляются диски
  - `sys.dm_io_pending_io_requests`
  - `sys.dm_io_virtual_file_stats`
- Процессоры
  - `sys.dm_os_nodes`



# Все ли корректно в настройках

- SP\_CONFIGURE

- max server memory (MB)

- Рекомендуется на каждые 16 Гб физической памяти оставлять не менее 1 Гб операционной системе
    - Но это размер буфера, фактическое потребление может быть больше
    - Если есть иные потребители, учитывайте их

```
SELECT (total_physical_memory_kb / (1024*1024)) -
CEILING((CAST((total_physical_memory_kb / (1024*1024)) AS
numeric(8,2)) / 16))
FROM sys.dm_os_sys_memory
```

# А нет ли проблем в коде?

- Об этом в следующей презентации
- **Сегодня 14:30 – 15:30** Оптимизация приложений на базе SQL Server, **DAT305**







Спасибо, вопросы?

Пожалуйста, заполните форму с  
оценкой сессии