Технический отчет

Руководство по сайзингу для SQL Server 2005/2008 и систем хранения NetApp

John S. Parker, NetApp

June 2009 | TR-3779

Краткое содержание

Этот технический отчет описывает методологию сайзинга (количественного анализа) для Microsoft® SQL Server® 2005/2008. Он содержит обзор архитектуры Microsoft SQL Server, руководство по определению типа нагрузки и описание процесса сбора статистических данных с помощью PerfMon и sqlstatspack.sql.

ОГЛАВЛЕНИЕ

1 ВВЕДЕНИЕ

Этот технический отчет содержит руководство по наилучшим практическим методам сайзинга (количественного анализа). Этот отчет поможет тем, кто занимается количественной оценкой размеров и производительности систем хранения NetApp® для использования их с Microsoft SQL Server. Он рассматривает процедуры и методологию, а также требования по производительности к системе хранения.

2 АРХИТЕКТУРА И КОМПОНЕНТЫ

SQL Server - это система управления реляционными базами данных, имеющая достаточную гибкость для применения от уровня небольшого отдела, до большого корпоративного хранилища данных. Гибкость архитектуры обеспечивается уровнями компонент и сервисов, предлагаемых продуктом SQL Server.

2.1 Компонентная модель SQL SERVER

Компонентная модель SQL Server (Рис. 1) поможет вам понять составные части системы при планировании развертывания сервера SQL. Базовая инсталляция SQL Server обычно располагается на локальном диске. Компоненты могут размещаться и перемещаться вокруг его ядра по мере необходимости.

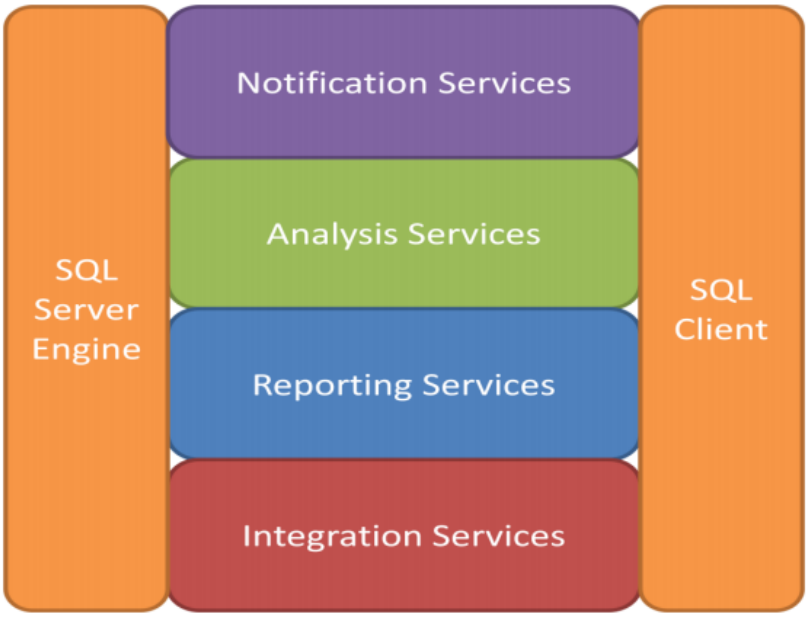

Рис. 1) Схема SQL Server.

Основа SQL Server - это database engine. Ядро СУБД входит в начальные 150MB, необходимые операционной системе для запуска. Глава 2 (стр. 30–32) документа **SQL Server 2008 Administrator's Guide** описывает в деталях необходимость использования высокоскоростных дисков и показатели производительности различных типов дисков. Это руководство не дает рекомендации по выбору какого-то определенного типа дисков, но перечисляет аспекты, которые необходимо учесть при развертывании SQL Server. Важно понять, что дизайн хранилища данных под SQL Server определяется факторами, лежащими за пределами собственно системы SQL Server.

Возможности конфигурировать размещение определенных файлов добавляет гибкости системе SQL Server. При планировании и оценке потребностей компонентов, нужно учесть потребности таких модулей, как SQL Server engine, Analysis Services, Reporting Services, и, в некоторых случаях, Integration Services.

2.2 Аспекты сайзинга базы данных

Необходимо помнить о нескольких моментах, когда вы проводите сайзинг для системы SQL Server. Примите во внимание базы данных (system и user) и файлы transaction log, данные file stream, компрессию базы данных (в SQL Server 2008), и директорию SnapInfo, если вы используете SnapManager® for SQL Server. Для подробностей по конфигурированию и развертыванию системы SQL Server смотрите **TR-3696: Microsoft SQL Server 2005 Engine: Storage Fundamentals for NetApp Storage Systems.**

Основные объекты хранения для систем NetApp.

- Data files (таблицы и индексы) - это основные файлы, используемые SQL Server storage engine. Каждая база может располагаться в нескольких файлах, и быть распределена на множество LUN-ов и томов.
- Transaction log. Доступ к файлам журнала транзакций осуществляется последовательно. Резервная копия журнала транзакций сохранятся в директории SnapInfo.
- Database compression - это новая возможность в SQL Server 2008, позволяющая сжимать базу данных, давая возможность администратору базы данных возможность растягивать процесс роста системы хранения, без значительного влияния на базу данных. Уровень компрессии может быть различен, в зависимости от характера данных, так что необходимо предварительное тестирование.
- Резервное копирование tempdb не требуется, так как она пересоздается, когда перезагружается инстанс SQL. Размер tempdb зависит от таких факторов, как онлайновое обслуживание (maintenance) базы, хранение версий, и план запросов (database query plans).
- Filestream data (SQL Server 2008): Этот тип данных представляет собой расширение базы для хранения больших объектов, таких как битмапы, текстовые файлы, музыка, видео или аудиофайлы. Они могут управляться через операции insert, update, delete, и select, также как обычными процедурами резервного копирования SQL.
- SnapInfo—рассчитайте размер директории SnapInfo, в том случае, если вы используете SnapManager for SQL Server. Это можно сделать по следующей формуле: SnapInfo_LUN_Size = ((DB_Size * Daily_Chg_Rate * Days_per_TA-Log_Snapshots) + system databases (master, model, MSDB, и т.д.) + SnapInfo metadata)

Когда вы проводите сайзинг вашей системы SQL Server, любые дополнительные данные, которые вы соберете, могут помочь определить свойства и профиль планируемой системы. Определенный набор данных требует для своей работы инструмент *database sizer tool*. Хорошее понимание того, как система работает, поможет провести правильный процесс сайзинга.

2.3 Руководство по основам SQL SERVER

Сайзинг - это только один аспект планирования новой системы SQL Server. Для лучшего понимания того, как следует организовывать систему для наилучшей производительности, смотрите документы **TR-3696: Microsoft SQL Server 2005 Engine: Storage Fundamentals for NetApp**

Storage Systems. TR-3696 содержит руководство по размещению tempdb и системной базы, а также рекомендации по конфигурациям аггрегейтов, томов и LUN-ов, которые помогут вам достичь необходимой производительности планируемой системы.

3 РУКОВОДСТВО ПО САЙЗИНГУ РАБОЧЕЙ НАГРУЗКИ

3.1 Обзор операций ввода-вывода в SQL SERVER

SQL Server очень чувствителен к задержкам в операциях ввода-вывода, что вызвано параллельно-транзакционным характером ядра SQL Server. SQL Server построен на сложной системе записей, страниц, экстентов и механизмов локирования таблиц, которые обеспечивают транзакционную консистентность всей системы SQL Server. Неудачная структура ввода-вывода (например, когда ввод-вывод занимает слишком много времени) вызывает то, что ресурсы удерживаются дольше необходимого, что в результате ведет к блокированию в системе. Когда это происходит, то часто бывает неочевидно, что причиной низкой производительности является подсистема ввода-вывода

Пользователи SQL Server, впервые знакомящиеся с NetApp могут нетвердо понимать разницы между RAID-DP® и другими типами RAID

Производительность SQL Server всегда зависит от производительности операций ввода-вывода. Эта производительность может быть улучшена увеличением количества используемых дисковых шпинделей, или увеличением скорости их вращения. В этом причина того, отчего в традиционных системах SQL мало распространены RAID 5 или RAID 6. NetApp RAID-DP, независимая реализация RAID 6, это стандартная возможность Data ONTAP®, которая предотвращает потери данных при отказе двух дисков, без излишних затрат на избыточность, и, при этом, в отличие от RAID-5 и RAID-6, не ухудшает общее быстродействие дисковой подсистемы.

- SQL Server reads: Когда происходит чтение из баз SQL Server, запрос клиента сперва попадает в буферный кэш. Если данные не обнаруживаются в кэше, то SQL Server обращается в подсистему ввода-вывода для того, чтобы извлечь данные. Операция считается не выполненной до тех пор, пока все данные не будут считаны полностью; следовательно, пользовательское подключение или процесс будет находиться в состоянии ожидания ввода-вывода (I/O wait state) до ее завершения.
- SQL Server writes: Пользователь записывает данные в журнал транзакций и буфера кэша. Если модифицируемые данные не находятся уже в буферах кэша, тогда они должны быть считаны в кэш из подсистемы ввода-вывода. Менеджер буфера обеспечивает то, что журнал транзакций записывается первым, до того, как изменения пишутся в базу данных. Это принято называть write-ahead logging. Когда пользователь вносит изменения в данные, и выполняет COMMIT, совершается операция записи в журнал, показывающая, что изменения занесены, позволяя операции COMMIT завершиться. Когда COMMIT завершен, пользовательский процесс может перейти к следующей стадии или команде без ожидания того, что изменения будут занесены на диск. Откат операций (ROLLBACK) идет тем же путем, что и COMMIT, но в обратном порядке. Менеджер буфера перемещает данные из кэша на диск. Это сопровождается записью log sequence numbers (LSNs) для каждой записи в журнале.
- Transaction log: Журнал транзакций в SQL Server характеризуется интенсивными записями, последовательного характера. Журнал транзакций используется для

возможности восстановить данные в случае сбоя базы данных или инстанса (экземпляра) базы.

3.2 Online Transaction Processing (OLTP)

База типа Online Transaction Processing (OLTP) ориентирована на проведение максимального количества транзакций в системе базы данных за наименьшее возможное время.

Примеры различных типов баз OLTP включают в себя такие системы, как заказы в интернете и системы управления производством. Системы типа OLTP могут иметь очень большие объемы транзакций в секунду, и пропускная способность для OLTP стоит во главе угла. Для таких транзакций SQL Server требует высокоэффективную подсистему ввода-вывода. Согласно статье **Microsoft SQL Server best practices** написанной Mike Ruthruff, транзакция профиля типа OLTP состоит из следующих паттернов:

- Работа OLTP характеризуется в основном случайными обращениями к данным на запись и чтение.
- Активность чтения (Read activity) в большинстве случаев консистентна и «атомарна»; как правило, она не содержит больших, занимающих много времени запросов.
- Активность записи происходит во время операций чекпойнтов (частота определяется настройками интервала восстановления (recovery interval settings)).
- Записи в журнал имеют последовательный характер и различный размер, который зависит от характера нагрузки (sector aligned up to 60KB).
- Чтения журнала также последовательны (sector aligned up to 120KB).

3.3 Decision Support System (DSS)

Статья **SQL Server best practices** описывает природу decision support system (DSS) следующим образом:

- Чтения и записи преимущественно последовательные, и, в общем случае, являются результатом сканирования таблиц и индексов, а также операций обширного insert.
- Размер ввода-вывода варьируется, но, в общем случае, больше 8KB. Операция упреждающего чтения (Read-ahead) по объемам кратна величине 8KB и достигает значений в 256KB (1024KB для Enterprise edition). Операция Bulk load кратна 8KB и достигает значений в 128KB.

3.4 Смешанный (MIXED)

В системе смешанного характера нагрузки вам необходимо использовать также смешанный подход для управления вводом-выводом в SQL Server. Количество операций чтения и записи усреднено, и находится в районах соотношения от 40% к 60% до 65% к 35%.

Хотя в смешанной среде соотношения нагрузки чтения-записи бывают различны, вариант 50-50 для чтения и записи хороший вариант для первого приближения. Ниже вы можете видеть различия, между этими режимами нагрузки.

Тип базы данных	операций чтения	операций записи
DSS	80%	20%
OLTP	66%	33%
MIXED	50%	50%

Таблица 1) Процентное соотношение операций чтения/записи.

4 УСРЕДНЕНИЕ РАЗМЕРОВ ДАННЫХ

4.1 Оценка объемов ввода-вывода

Оценка объемов ввода-вывода, требуемых для системы SQL Server, весьма критична для сайзинга базы данных. Этот раздел поможет администратору понять, как поддерживать производительность экземпляра базы данных в приемлемых пределах. Вы сможете оценить объемы ввода-вывода, в том случае, когда вы не можете получить действительные физические данные по вводу-выводу с реальной системы. Такое может случиться, например, когда создается новая система. Воспользуйтесь формулами для оценки, приведенными в **SQL Server 2005 Administrator's Guide**.

Для оценки объемов ввода-вывода в новой базе данных:

1. Оцените количество транзакций за определенный момент времени.
2. Умножьте число транзакций на коэффициент насыщения 0,85, и поделите это на число секунд в рабочем дне.

Секунды в дне определяются рабочими часами базы данных. Если база данных работает круглосуточно, то это число 86400. Формула для оценки ввода-вывода такова:

Общий I/O = (*ожидаемое количество транзакций* * 0,85)/*секунд в рабочем дне*

Например, если у вас 40000 транзакций в системе, работающей 24 часа в сутки, формула будет:

(40000 * 0,85)/86400 = 0,3935 IOPS

3. После определения количества операций ввода-вывода, определим, систему какого типа мы создаем. Если создается система типа OLTP, определим число операций записи и чтения, умножив полученное число общих операций ввода-вывода на составляющие проценты соответствующих режимов (OLTP/DSS/mix). Таблица 1, приведенная ранее, показывает процентное соотношение для каждого из трех типов.

Формула ввода-вывода в мегабайтах:

((*число транзакций* * 0.85)/*секунд в дне*) * *тип* % = I/O мегабайт.

Например, для определения объемов чтения и записи для системы типа DSS, формула будет:

((40000 * 0,85)/86400)* 0,80) = 0,3148MB чтения

((40000 * 0,85)/86400)* 0,20) = 0,0787MB записи

4.2 Сбор статистики

Когда вы проводите сайзинг уже существующей базы данных, понимание типа ее рабочей нагрузки и интерпретация статистических данных будет весьма полезна. Инструмент database sizer tool не определяет тип системы по загруженным в него данным. Сравнив значения физических чтений и записей со значениями из Таблицы 1, вы сможете определить тип рабочей нагрузки системы. Это важно для сбора данных в периоды максимальной стрессовой нагрузки. Скрипт

sqlstatspack.sql соберет средние значения на протяжении всего интервала наблюдаемого времени.

PerfMon позволяет вам видеть верхние пороги значений за наблюдаемые периоды времени. Для подробностей по использованию PerfMon, смотрите **Приложение С**. Введите наивысшие значения величин по чтению и записи в приложение database sizer tool.

5 ПОСТРОЕНИЕ МОДЕЛЕЙ РОСТА ЕМКОСТИ

5.1 Планирование емкости

В соответствии с **Microsoft SQL Server 2005 Administrator's Companion**, основы сайзинга и планирования емкости основываются на теории очередей (queuing theory). Queuing theory содержит несколько терминов, которые следует понимать: время обслуживания (service time) и время ожидания (wait time), известное также как время в очереди (queue time). Главное правило:

Время ответа = время обслуживания + время в очереди

Queuing theory также имеет два ключевых компонента, согласно **Microsoft SQL Server 2005 Administrator's Companion**:

- Шансы «встать в очередь» экспоненциально растут, когда емкость ресурса близка к заполнению.
- Время отклика (response time) равно сумме времени обслуживания и времени в очереди. (См. Chapter 6 [pages 108–111] в **Microsoft SQL Server 2005 Administrator's Companion**.)

Наблюдение за подсистемой ввода-вывода (I/O subsystem) с помощью PerfMon это отличный способ разработать профиль поведения для подсистемы ввода-вывода SQL Server. Счетчики группы Physical disk, более всего помогающие понять характер ввода-вывода:

- Disk reads/sec: IOPS чтения для диска или дисков
- Disk transfers/sec: Общий уровень IOPS *чтения плюс записи* для каждого диска
- Disk writes/sec: IOPS записи для каждого диска
- Avg. disk sec/read: Задержки чтения или среднее время для каждой операции чтения
- Avg. disk sec/write: Задержки записи или среднее время для каждой операции записи
- Average disk queue length: Среднее количество запросов чтения и записи, попадающих в очередь к диску.

Для хорошо настроенной подсистемы ввода-вывода, величины задержек чтения и записи (read and write latency) должны быть в пределах от 5 до 10 ms. Задержки выше 20 ms говорят о том, что система может испытывать проблемы с вводом-выводом. Задержки выше 30 ms неприемлемо высоки. Поскольку все величины относительны, то влияние измеренных задержек зависит от типа операций, совершаемых на системе хранения, и того, что они значат для пользователей.

Вопросы, которые необходимо задать пользователю при проведении планирования:

- Каков размер базы или баз данных?
- Сколько пользователей используют эти базы?
- Когда наступает время пиковой активности использования?

- Каков характер нагрузки для каждой из баз? OLTP, DSS, batch jobs, misc.?
- Каков темп прироста данных?
- Каков темп прироста пользователей?
- Это выделенное решение под SQL, или совместно используемое с другой задачей?
- Есть ли необходимость в решении высокой доступности, масштабируемости и/или катастрофоустойчивости?
- Будет ли база реплицироваться?

5.2 Модели роста

Использование моделей рост позволяет администратору разрабатывать и оценивать потенциальный рост базы данных. Единственная абсолютно точная модель роста это «историческая» модель, но она позволят вам сказать, что было, а не то, что будет. Однако администраторы могут взять данные о приросте из прошлого, вставить их в модель и проэкстраполировать их на будущее, чтобы получить наилучшее соответствие требованиям.

Ограничением любой модели будет то количество данных, которые вы в нее занесете. Чем больше данных о системе вы соберете, тем точнее будет оценка. В этом документе мы рассмотрим две различных модели: модель линейного роста, и модель геометрического роста (Hotek and Makin).

Модель линейного роста

Эта модель реализует модель прямого линейного прироста. Она основывается на данных объемов прироста данных в системе. Формула вычисления линейного прироста такова:

Будущее использование = *текущее использование* + (*величина прироста* * *число периодов*)

С помощью этой формулы можно определить суммарный размер базы данных, исходя из текущего ее состояния. Процедура для сбора информации состоит в использовании sp_helpdb. Это даст вам суммарные значения для каждой базы. Далее, возьмите сумму дискового пространства, использованного всеми базами. Величина прироста может быть найдена отслеживанием размеров за определенный период времени. Возьмите этот прирост и используйте его вместе с числом месяцев, на которые вы хотите оценить будущее.

Например, для базы, в настоящий момент занимающей 400GB, с приростом в 50GB в месяц на протяжении трехмесячного периода (Таблица 2), формула будет такой:

Ожидаемое будущее использование пространства = 400GB + (50GB X 3) = 550GB через 3 месяца

Параметр	Величина
Текущее использование	400GB
Прирост	50GB/месяц
Период	3 месяца

Таблица 2) Пример 1.

Выводы, сделанные в результате этого упражнения, дадут основания для оценки того, сколько пространства для базы данных следует предусмотреть, в случае, если база будет расти с заданными параметрами в течение трех месяцев.

Модель геометрического роста

Исследование модели «геометрического» роста позволяет более точно настроить объемы изменения в системе. Один из способов это посмотреть на темп прироста как на процент прироста за определенный период времени. Возьмите данные для определенного периода времени для всей базы, и сравните с разницей за предыдущий период. Если у вас нет данных за предыдущие интервалы времен, то можно оценить размеры, основываясь на данных размеров резервных копий базы msdb. Если вы регулярно делаете резервные копии для базы данных, вы сможете оценить темп прироста по этим сведениям.

Для того, чтобы оценить прирост, подставьте в формулу текущее использование пространства, и темпы прироста, а также планируемый срок, и вы получите показатели geometric view для темпов прироста.

Будущее использование = *текущее исп.* * (1 + *величина прироста* /*текущее исп.*) число периодов

Например, при использовании величин из Таблицы 2, формула будет такой:

Ожидаемое будущее использование пространства = 400 * $(1 + 50/400)^3$ = 569.53GB через 3 месяца

6 ФИЗИЧЕСКАЯСТРУКТУРА ХРАНЕНИЯ ДЛЯ БАЗ SQL SERVER

Планирование новой системы требует оценки схемы размещения данных SQL Server на системе хранения. Для руководства в том, как разместить данные по aggregates, томам и LUN-ам, смотрите документ **TR-3696: Microsoft SQL Server 2005 Relational Engine: Storage Fundamentals for NetApp Storage Systems**. Этот документ описывает, как распределить пространство и сконфигурировать SQL Server instance для достижения максимальной производительности.

7 ПРИМЕР САЙЗИНГА

Теперь, когда мы рассмотрели все эти аспекты, давайте проведем примерный сайзинг для базы SQL Server на системе хранения NetApp. Первым шагом соберем данные, которые потребуются для инструмента сайзинга.

Начнем со сбора информации о количестве записей и чтений базы, и о ее размерах. Рассмотрим свежеустановленную систему, как это требуется при оценке данных сайзера.

1. Определите число транзакций базы данных. База работает 24 часа в сутки, и оцениваем количество транзакций в 40 000 за 24-часовой период. Эти транзакции включают в себя все операции INSERT, UPDATE и DELETE, происходящие в базе.
2. Определите количество чтений и записей. Введите ожидаемое число транзакций, 40 000, в следующую формулу:
   Общее количество I/O = (*ожидаемое число транзакций* *0,85)/*секунд в рабочем дне*

То есть:

(40000 * 0,85)/86400 = 0,3935

Это даст нам общее число операций ввода-вывода в мегабайтах за 24-часовой период.

Используйте этот результат в формуле, использующей тип рабочей нагрузки базы данных DSS, OLTP, или mixed:

((*число транзакций* * 0,85)/*секунд в рабочем дне*) * *тип нагрузки* % = *I/O мегабайт*

Например, для типа нагрузки DSS будет:

((40000 * 0,85)/86400)* 0,80) = 0,3148MB reads

((40000 *0,85)/86400)* 0,20) = 0,0787MB writes

3. Определите размер базы данных и журналов транзакций. Теперь мы знаем объемы ввода-вывода системы, которые установили на шаге 2, требуются размеры базы и журналов. Возьмем размер базы равным 500GB, и журналов транзакций в среднем в 15% от размера базы, то есть 75GB.
4. Определим число журналов транзакций хранимых в онлайне. Для нашего темпа изменений оценим количество журналов, доступных в онлайне в 7.
5. Определим число сохраняемых Snapshot™. В этом примере мы определим возможность восстановления из полной ежесуточной копии в течение 14 дней. Таким образом, мы должны сохранять 14 Snapshot-ов, для возможности восстановления в любой момент.
6. Определим величину прироста базы за год. Так как мы рассматриваем новую систему, то требуется интерполировать данные, чтобы вставить их в модель роста. Для вычисления годовой величины изменений, формула количества записей в мегабайтах в секунду изменена, чтобы определить величину дневных изменений, которая, умноженная на 365, дает процент годовых изменений данных базы.

Формула, измененная для вычислений процентной величины ежедневного объема изменений:

((*число транзакций* * 0,85) * *тип нагрузки* %)/*размер базы в MB*) = *ежедневные изменения* (%)

(*ежедневный прирост* * 365) = *ежегодовой прирост* (%)

Если ежегодовой прирост менее 10%, то возьмите его равным 10%.

Пример:

Дневной прирост записей: ((40000 * 0,85) * 0,20)/500000 = 0,0136%

Годовой прирост записей: (0,0136 * 365) = 4,964%. Берем 10%, если получилось менее 10%.

Дневной прирост чтений: ((40000 * 0,85) * 0,80)/500000 = 0,0544%

Годовой прирост чтений: (0,0544 * 365) = 19,856%. Берем 19,856%, так как получилось больше 10%.

7. Рекомендуется хранить данные и журналы на отдельных томах, и разные инстансы базы на отдельных aggregates. Это зависит от того, насколько загружена ваша база данных; постарайтесь делать ваши aggregates настолько большими, насколько это возможно, чтобы получить положительный эффект от распределения нагрузки ввода-вывода базы по множеству входящих в aggregate дисков.
8. После того, как вы определились с этими входными данными, осталось задать несколько общих параметров, таких как контроллер системы хранения, максимальное количество контроллеров, тип дисков, и так далее.

Общая спецификация системы	
Версия Data ONTAP	7.3.x
Платформа	Любая (any)
Максимальное число контроллеров (Max # of Heads)	10
Запас по CPU (CPU Headroom %)	30%
Тип дисков (Disk Type)	Любые (any)
Тип RAID	RAID-DP
Размер RAID-группы (RG Size)	16
Количество запасных дисков (# Spares)	1
Считать полными полками? (Full shelf?)	Нет
Запас по дисковой емкости (Disk headroom)	10%
Кластерная система (Local Cluster?)	Нет
Режим работы при кластерном файловере (CFO Perf)	Degraded
Metrocluster	Нет
Число модулей PAM	0

Внимание: данные в этой таблице просто примеры величин, которые нужны для работы сайзера базы данных.

Они могут быть подобраны более точно для того, чтобы соответствовать требованиям базы данных.

9. Дополнительные моменты, которые надо не упускать из виду, возможно использование SnapMirror®, SnapDrive®, и SnapManager for SQL Server.
10. Проверьте внимательно форму сайзера, чтобы убедиться, что вы ничего не упустили и все необходимые данные собраны и введены. Когда вы собрали все данные, вы готовы к использованию инструмента database sizer tool.

ПРИЛОЖЕНИЯ

Приложение A: Форма сбора данных для сайзинга

Эта таблица поможет с вычислениями и организацией данных перед тем, как вводить их в database sizing tool.

https://fieldportal.netapp.com/ci/files/18388.xlsx

Внимание: Не обязательно использовать всю таблицу целиком.

Приложение B: STATSPACKSTATSPACK4SQL3.SQL

http://perf-build1.eng.netapp.com/DBSizer/SQLMon.jsp

```
/*
###############################################################################
######################
## Original Create Date: 6/15/05 Original Author: Jonas Irwin
## Current Version: 2.0
## Current Author: Mike Flannery
##
## Purpose:
## Grab IO statistics for SQL Server databases and data files over operator specified
time interval
## Mods:
## 12/20/07 - Flannery - Added UNION at end to make the output more end user readable
## 12/20/07 - Flannery - Tested with SQL Server 2005. Made tables and columns case
consistent
## 3/14/08 - Flannery - Tested with SQL Server 2008
## 3/14/08 - Flannery - Changed SQLIO table from permanent to variable table
## 8/01/08 - Safa – Adjust case in Column calls
## 8/01/08 - Safa – Validate on SQL 2000 and 2005
###############################################################################
######################

*/

DECLARE @total int
DECLARE @now datetime
select @now = getdate()
select @total = 0

DECLARE @SQLIO TABLE
 (dbname varchar(128) ,
 fname varchar(2048),
 startTime datetime,
 noReads1 int,
 noWrites1 int,
 BytesRead1 bigint ,
 BytesWritten1 bigint ,
 noReads2 int,
 noWrites2 int,
 BytesRead2 bigint ,
 BytesWritten2 bigint,
 endtime datetime,
 deltawrites bigint,
 deltareads bigint,
```

```
 timedelta  bigint,
 fileType  bit,
 fileSizeBytes bigint
)

USE master

--grab baseline first sample
INSERT INTO @SQLIO
SELECT
 cast(DB_Name(a.DbId) as varchar),
 b.filename,
 getdate(),
 cast(NumberReads as int),
 cast(NumberWrites as int),
 cast(a.BytesRead as bigint),
 cast(a.BytesWritten as bigint),
 0,
 0,
 0,
 0,
 0,
 0,
 0,
 0,
 'filetype' = case when groupid > 0 then 1 else 0 end,
 cast(b.size as bigint) * 8192
FROM
 ::fn_virtualfilestats(-1,-1) a ,sysaltfiles b
WHERE
 a.DbId = b.dbid and
 a.FileId = b.fileid
ORDER BY 1

/*DELAY AREA - change time at will */
WAITFOR DELAY '000:00:10'


UPDATE @SQLIO
set
 BytesRead2=cast(a.BytesRead as bigint),
 BytesWritten2=cast(a.BytesWritten as bigint),
 noReads2 =NumberReads ,
 noWrites2 =NumberWrites,
 endtime= getdate(),
 deltawrites = (cast(a.BytesWritten as bigint)-BytesWritten1),
 deltareads= (cast(a.BytesRead as bigint)-BytesRead1),
    timedelta = (cast(datediff(s,startTime,getdate()) as bigint))

FROM ::fn_virtualfilestats(-1,-1) a ,sysaltfiles b
WHERE  fname= b.filename and dbname=DB_Name(a.DbId) and
 a.DbId = b.dbid and
 a.FileId = b.fileid


/*dump results to screen - individual results
SELECT
 'Transaction Log Size',
 sum(cast(b.size as float) * 8192)/1024/1024
FROM
 ::fn_virtualfilestats(-1,-1) a ,sysaltfiles b
WHERE
```

```
 a.DbId = b.dbid and
 a.FileId = b.fileid and
 groupid = 0
union
SELECT
 'Database Size',
 sum(cast(b.size as float) * 8192)/1024/1024/1024
FROM
 ::fn_virtualfilestats(-1,-1) a ,sysaltfiles b
WHERE
 a.DbId = b.dbid and
 a.FileId = b.fileid and
 groupid > 0
union
select
 'Read IO Percent',
 (sum(cast(deltareads as float))/(sum(cast(deltawrites as
float))+sum(cast(deltareads as float))))*100
from @SQLIO
union
select
 'Database Read Rate',
 sum(cast(deltareads as float))/max(cast(timedelta as float))/1024/1024
from @SQLIO
union
select
 'Database Write Rate',
 sum(cast(deltawrites as float))/max(cast(timedelta as float))/1024/1024
from @SQLIO
union
select
 'Log Rate',
 (sum(cast(deltawrites as float))/max(cast(timedelta as float))/1024/1024)*3
from @SQLIO
where fileType=0
*/
/*dump results to screen - sizer appropriate results */
select * from @SQLIO
```

Приложение C: PERFMON

PerfMon это инструмент, являющийся частью Windows®. Этот инструмент поможет вам удобно наблюдать за работой SQL Server, собирая данные по активности операций ввода-вывода, необходимые для инструмента database sizer tool.

Инструкции по запуску PerfMon:

1. Запустите PerfMon. PerfMon может использоваться для наблюдения за работой инстанса SQL Server на локальной, либо с удаленной машины. Удаленная система не требует установки на нее SQL Server. Однако обе операционные системы должны иметь одну и ту же OS и версию установленного сервиспака.

2. Выберите следующие счетчики для регистрации:

    - LogicalDisk
        - Avg. disk queue length
        - Current disk queue length
        - Disk reads/sec: Введите это число в database sizer tool.
        - Disk writes/sec: Введите это число в database sizer tool.
    - PhysicalDisk
        - Disk reads/sec: Read IOPS для каждого диска или дисков
        - Disk transfers/sec: Общее число IOPS чтений плюс записей для каждого диска
        - Disk writes/sec: Write IOPS для каждого диска или дисков
        - Avg. disk sec/read: Read latency или average read time для каждой операции
        - Avg. disk sec/write: Write latency или average write time для каждой операции
        - Average disk queue length
    - Processor
        - Percentage processor time (для каждого процессора)
    - SQLServer: Transactions
        - Transactions
    - SQLServer: Wait statistics
    - Network I/O waits
        - Page I/O latch waits (average)
        - Page I/O latch waits (current)

Внимание: Для процедуры сайзинга вам понадобятся только результаты LogicalDisk: Disk reads/sec и LogicalDisk: Disk writes/sec. Остальные данные помогут вам проанализировать общую производительность и состояния системы.

3. После установки желаемых счетчиков PerfMon, запустите наблюдение.

Внимание: При наблюдении, попытайтесь собрать данные в пиковые периоды загрузки, чтобы при использовании этих данных для сайзинга получить желаемую производительность будущей системы.

Приложение D: Дополнительные источники информации

Garvey, Brad, Space Reservation for Microsoft Exchange Server Using a NetApp SAN or IP SAN: TR-3758i, March 2009, http://my.netapp.com/psweb/jsp/DownloadContent.jsp?dDocName=P_053282.

Hotek, Mike, and Makin, J.C., MCITP Self-Paced Training Kit (Exam 70-443): Designing a Database Server Infrastructure Using Microsoft SQL Server 2005, Microsoft Press, 2007.

Isakov, Garcia, Misner, Patel, and Whalen, Microsoft SQL Server 2005 Administrator's Companion, Microsoft Press, 2006.

McPhail, Robert, Microsoft SQL Server 2005 Relational Engine: Storage Fundamentals for NetApp Storage Systems, TR-3696, July 2008, http://media.netapp.com/documents/tr-3696.pdf.

NetApp database sizing tool, http://perf-build1.eng.netapp.com/DBSizer/UnifiedDatabase.jsp.

Ruthruff, Mike, SQL Server Best Practices: Predeployment I/O Best Practices, June 2008, http://technet.microsoft.com/en-us/library/cc966412.aspx.

Considerations for Transactional Replication, http://msdn.microsoft.com/en-us/library/ms151254.aspx.