SYBASE®
An SAP Company

ИНФОРМАЦИОННЫЙ ДОКУМЕНТ

Практическое руководство по определению характеристик аппаратных средств для Sybase IQ 15

Sybase IQ и RAP Store в составе RAP — The Trading Edition

Марк Мьюми,
ведущий консультант по системам,
Sybase, Inc.

ОГЛАВЛЕНИЕ

Предисловие

Назначение настоящего руководства

Настоящий документ — это попытка осветить основные аспекты определения требований к характеристикам (сайзинга) систем на основе Sybase IQ. Данное руководство применимо также к компоненту RAP Store пакета RAP — The Trading Edition. Компонент RAP Store основан на Sybase IQ и потому его сайзинг выполняется точно так же, как и сайзинг отдельного экземпляра Sybase IQ.

Как правило, при подборе характеристик системы на базе Sybase IQ в расчет берутся центральный процессор, оперативная и дисковая память.

Необходимо учесть, что работа над настоящим руководством ведется непрерывно, а большинство положений взяты из реальной практики. Следовать ли изложенным здесь указаниям — решает исключительно читатель и группа внедрения, под свою ответственность.

Термины

В настоящем документе будет использоваться термин ЦП (центральный процессор). По традиции в документации и технической литературе по Sybase IQ словами «ЦП», «процессор» и «ядро» обозначается одно и то же физическое устройство, а именно процессорное ядро, исполняющее компьютерную программу. Вычислительные системы могут быть оснащены как одноядерными, так и многоядерными процессорами (с 2, 4, 8 или более ядрами). Мы всюду будем придерживаться этой традиции, понимая под центральным процессором одно процессорное ядро.

Логические процессоры систем с гиперпоточностью (hyper-threading) здесь не рассматриваются и термин ЦП в данном документе к ним не относится. Системы с гиперпоточностью не дают для Sybase IQ сколько-нибудь заметного прироста быстродействия, и потому наличие этой функции в приводимых здесь рекомендациях и алгоритмах не учитывается.

Процессорное ядро с гиперпоточностью должно рассматриваться как одно ядро (один ЦП). Для того чтобы настроить IQ соответствующим образом, следует передать оптимизатору число ядер, используя параметр –iqnumbercpus. С точки зрения операционной системы 4-ядерная машина с гиперпоточностью имеет 8 ядер (8 ЦП). В этом случае параметр –iqnumbercpus следует установить равным четырем, чтобы избежать чрезмерной нагрузки на процессоры.

В частности, следует учитывать проблему гиперпоточности и использовать параметр –iqnumbercpus при работе с системой AIX на платформах Power 6 и Power 7. Сегодня в большинстве систем на платформе IBM применена технология SMT (SMT2 на Power 6 и SMT2 либо SMT4 на Power 7), и IQ по умолчанию рассматривает все потоки как процессорные ядра.

Сведения об авторских правах

Нововведения в IQ 15

Параллельная обработка

В 15 версии Sybase IQ реализованы фундаментальные изменения в процессах как загрузки данных, так и обработки запросов. Введен механизм, позволяющий параллельно выполнять значительно большее число операций, чем когда-либо ранее. В версиях 15.1 и 15.2 число параллельно выполняемых операций увеличено.

Реализованный в IQ 15 параллелизм имеет целью способствовать достижению равновесия между скоростью обслуживания отдельных пользователей и способностью системы обслуживать множество пользователей одновременно. Для этого применяется динамическое перераспределение ресурсов. Например, первый пользователь, обратившийся к системе, получает в свое распоряжение все ядра, необходимые для операции загрузки или обработки запроса. Второму пользователю не придется ожидать завершения обслуживания первого пользователя: механизм и оптимизатор высвободят часть мощностей, занятых первым пользователем, и передадут их второму, так что процессорные ресурсы будут распределены между обоими пользователями пропорционально. Подобным же образом система поступит при подключении третьего пользователя, так что в результате на каждого пользователя придется примерно по одной трети вычислительных мощностей. По завершении каждой операции ресурсы высвобождаются и передаются выполняющимся задачам, если это будет способствовать ускорению их выполнения.

Более высокая степень параллелизма влечет более тесную зависимость между количеством процессорных ресурсов, оперативной и дисковой памяти. Для повышения быстродействия может оказаться недостаточно увеличить число или тактовую частоту процессоров, — при этом будет потребляться больше оперативной и дисковой памяти, и может потребоваться более высокая пропускная способность каналов ввода-вывода. Поэтому при наращивании мощности системы необходимо принимать во внимание все ее составляющие.

Мультиплексная архитектура

Как уже отмечалось, в Sybase IQ 15 был реализован ряд нововведений. При расчете требуемых характеристик системы следует принять во внимание два из них: наличие узла-координатора и нескольких записывающих узлов.

Согласно Sybase IQ Multiplex Guide, Sybase IQ поддерживает транзакции чтения-записи, инициируемые несколькими серверами в составе мультиплекса. Ведущий сервер, или координатор, управляет всеми транзакциями чтения-записи в масштабе мультиплекса и поддерживает общий каталог и общую область метаданных. Журнал таблицы версий (TLV) также ведется координатором; в нем хранится информация о DDL-операциях, а также необходимая вторичным серверам (считывающим или записывающим узлам) информация о новых версиях таблиц.

Такую конфигурацию можно назвать «асимметричным кластером», поскольку функции узлов могут различаться, в отличие от других кластерных архитектур баз данных, где обычно либо применяется принцип разделения всех ресурсов, либо разделение ресурсов отсутствует.

Следует подчеркнуть, что все DDL-операции выполняются на координаторе, вне зависимости от того, какой экземпляр IQ фактически выдал SQL-команду. Обычно это не представляет проблемы, за исключением таких операций, например, как создание индексов для целиком заполненных таблиц — всю работу и в этом случае будет выполнять координатор, а не записывающий узел, на котором была запущена команда.

Об этом важно помнить при определении характеристик машины, которая будет служить узлом-координатором IQ. Отсюда вытекают два правила. Во-первых, координатор следует размещать на изолированном сервере, рассчитанном на все операции обслуживания администратором и на любые возможные DDL-операции реального времени, такие как создание индексов. Во-вторых, координатор должен играть роль записывающего узла.

Как было отмечено выше, Sybase IQ 15 допускает запись или изменение данных несколькими узлами одновременно. Однако, как и в IQ 12.x, данные или структура каждого объекта могут изменяться только одним пользователем; эта архитектурная особенность пока сохраняется в силе. Впрочем, для баз с большим количеством таблиц, данные в которых постоянно изменяются, вполне возможно построить систему с двумя или более записывающими узлами, которые будут заниматься исключительно обработкой изменений, в то время как все прочие узлы будут иметь статус считывающих и, соответственно, только обрабатывать запросы.

Прочие возможности, которые следует принять во внимание

В IQ 15 был введен также механизм разделения (партиционирования) баз данных. Впрочем, при сайзинге это не учитывается, разве только в отношении наличия нескольких пространств БД и размещения разделов.

Для поддержки многоярусного хранения, партиционирования и размещения объектов на диске в Sybase IQ 15 введены множественные пространства БД (dbspaces). Для хранения объектов данных администратор может создавать любое число пространств БД. В роли объектов могут выступать таблицы, столбцы, индексы и разделы.

Подчеркнем, что приведенные здесь указания по определению характеристик систем хранения рассчитаны на сайзинг для отдельного пространства БД, а не всех пространств БД, составляющих базу данных.

Общие правила определения характеристик

В справочных целях приведем краткие указания по расчету характеристик, которые будут подробно раскрыты в дальнейшем изложении.

Общие указания по характеристикам оборудования	
ЦП	Минимум на 25% меньше, чем в существующем хранилище данных, нет необходимости использовать коэффициент 1 ЦП/1 Тбайт
Серверная платформа	Windows (Intel/AMD), Linux (Intel/AMD/IBM), AIX, HP-UX (PA-RISC/Itanium), Solaris (SPARC/AMD)
Тип системы хранения	Любое современное оборудование, поддерживающие диски FC или SATA со встроенным контроллером RAID и кэш-памятью
Уровень RAID	RAID 1 или RAID 1+0 дают наивысшее быстродействие, RAID 5 — наиболее выгоден экономически для хранения данных IQ
Управление томами	Для главных и временных устройств IQ не требуется
ПО кластеризации и обеспечения высокой готовности	Не требуется, но может использоваться для обеспечения высокой готовности и аварийного восстановления

Объем дискового пространства для IQ	
Размер хранилища данных (данные и индексы)	На 20-70% меньше объема входных данных
Общий объем системы хранения	Размер хранилища данных IQ плюс 15% дополнительно для RAID 5 или 100% для RAID-зеркалирования
Размер блока системы хранения	Максимально возможный. Основное правило — размер блока должен равняться размеру страницы IQ или вдвое его превосходить.

Общие указания по конфигурированию БД	
Размер страницы базы данных	64-512 Кбайт (по умолчанию 128 Кбайт)
Размер блока базы данных	По умолчанию 1/16 размера страницы (4-32 Кбайт)
Чувствительность БД к регистру букв	Рекомендуется значение «Case Respect» (различение регистра)
Размер файла подкачки	1-2 объема физической оперативной памяти
Общее правило определения размера ОЗУ	4-8 Гбайт на каждое процессорное ядро

Количество ЦП	
Количество ЦП для обработки запросов	0,1—1,5 ЦП на запрос
Количество ЦП для загрузки данных	0,05—0,5 ЦП на каждый индекс и столбец. Обработка индексов HG, WD и TEXT параллелизуется полностью и может занять все процессоры системы

Параметры оперативной памяти	
Всего памяти для всех операций IQ	Общий объем ОЗУ за вычетом 10-20% для ОС
Основной и временный кэш при обработке запросов	40% для основного кэша, 60% временного кэша оставшегося объема ОЗУ
Основной кэш при загрузке	5-10 страниц на каждый индекс и столбец
Временный кэш при загрузке	Каждый HG: (8 + sizeof(datatype)) * numberRowsBeingLoaded Каждый WD: (8 + sizeof(datatype)) * numberRowsBeingLoaded * numberTokens (округлить до следующей верхней границы страницы)
Память для загрузки	binaryTableWidth * 10 000 * 45 / 1024 / 1024
Память битовых массивов при загрузке	8 192 байт на отдельное значение, загружаемой в каждый индекс LF, HNG, DTTM, DATE, TIME и CMP
Резервная память	TmpVal = max(2 * numberOfCPUs, 8 * numberOfMainLocalSpaces) Объем_памяти = (TmpVal * 20) * (block_factor * block_size)
Кэш каталога (–c или –cl/–ch)	2-16 размеров файла каталога (.db)

Параметры системы хранения и ввода-вывода	
Требуемая пропускная способность в расчете на каждое процессорное ядро	20-40 Мбайт/с
Общая полоса пропускания для каждого сервера	число_ядер * 20-30 Мбайт/с
Минимальное число физических FC-дисков на процессорное ядро	0,3-1 диск Вычисляется отдельно для основного и временного хранилища
Минимальное число физических SAS-дисков на процессорное ядро	0,5-3 диска Вычисляется отдельно для основного и временного хранилища
Минимальное число физических SATA-дисков на процессорное ядро	2-5 дисков Вычисляется отдельно для основного и временного хранилища
Максимальное число процессорных ядер для основного хранилища	4-8 ядер Вычисляется отдельно для основного и временного хранилища
Число логических устройств (LUN) для основного/ локального хранилища IQ	3 + (число_ядер / 10) (округлить_вверх)
Число логических устройств (LUN) для временного хранилища IQ	3 + (число_ядер / 10) (округлить_вверх)

Управление оперативной памятью в Sybase IQ

Важно хорошо представлять, как используется память, выделенная IQ при запуске, и как распределяется динамически выделяемая память в ходе работы СУБД. Неправильная настройка использования памяти, как правило, снижает быстродействие из-за частых операций подкачки.

Память операционной системы и память, не предназначенная для Sybase IQ

Во-первых, следует установить, какие программы работают на машине (помимо Sybase IQ) и сколько памяти они используют. Обычно это следующие программы:

- Операционная система
- Серверы OLAP
- Промежуточное ПО
- Прочие программы
- Средства мониторинга
- Оболочки, скрипты и т. п.

Следует просуммировать объем памяти, занимаемой этими программами, и вычесть его из общего объема оперативной памяти, установленной в машине. Из оставшегося объема свободной памяти следует вычесть память, потребляемую операционной системой. Размер кэша файловой системы на большинстве платформ составляет 20%. Его также следует вычесть из доступного объема ОЗУ. Кроме того, ОС использует ОЗУ для ядра и базовых функций. Грубо эту цифру можно оценить в 500 Мбайт — 1 Гбайт. Оставшаяся память будет доступна Sybase IQ.

Память Sybase IQ

На карте рабочей памяти IQ можно выделить пять основных областей: область конфигурации сервера, область управления версиями, загрузочная память, память битовых массивов и резервная память.

Память Sybase IQ состоит из следующих областей:

- кэш каталога (опции –c/–ch/–cl в файле конфигурации);
- память потоков (размер стека * число потоков IQ);
- основной кэш (опция –iqmc в файле конфигурации);
- временный кэш (опция –iqtc в файле конфигурации);
- память для управления версиями;
- загрузочная память (не используется начиная с выпуска 15.2);
- память, используемая для резервного копирования.

Конфигурация сервера

При запуске сервер IQ считывает ряд параметров, задающих количество памяти, используемой в ходе работы. Как правило, эти параметры указываются в файле конфигурации. Они определяют следующие аспекты:

- **Кэш каталога. Параметры** –c, –cl и –ch определяют, сколько оперативной памяти сервер выделит каталогу. Параметр –ca определяет, будет ли эта область памяти статической или динамической.
- **Память потоков**. Каждому потоку, формируемому при запуске сервера, выделяется участок памяти для стека. Общий объем памяти, выделяемой потокам, вычисляется по формуле: размер стека * число потоков IQ. Размер стека задается параметром –iqtss, число потоков IQ — параметром –iqmt.
- **Основной кэш**. Размер этой области памяти, используемой для обработки постоянно хранимых данных, задается параметром –iqmc.
- **Временный кэш**. Эта область памяти служит целям обработки временных данных, ее размер задается параметром –iqtc.

Основной и временный кэш

Основной кэш используется для размещения статических, постоянно хранимых пользовательских данных, а также системных структур, управляющих доступом к данным посредством индексов. Временный кэш — это область памяти, где хранятся временные данные — временные таблицы, внутренние рабочие таблицы и прочие временные по своей природе структуры данных.

После того как определен объем оперативной памяти, которую можно выделить обоим кэшам, следует установить соотношение между основным и временным кэшами. Для типовых конфигураций память первоначально распределяют в соотношении 40% для основного кэша и 60% для временного.

Размер основного кэша влияет на использование 3-байтового индекса FP. Чем больше размер (см. ниже), тем больше разных значений может быть представлено в 3-байтовых структурах FP. Поэтому часто рекомендуют на время загрузки исходных данных в таблицы увеличить размер основного кэша до 50% и, соответственно, уменьшить размер временного кэша также до 50%.

В системах, где основную нагрузку составляют операции ввода данных, а не обработки запросов, как правило, больше памяти выделяют для временного кэша — для облегчения обработки индексов HG и WD. При этом следует помнить, что уменьшение основного кэша может ограничить возможность использования 3-байтового индекса FP.

В системах, где обработка запросов преобладает над вводом данных, как правило, больше памяти выделяется основному кэшу — это сокращает число операций обращения к основному хранилищу для загрузки пользовательских данных. Исключением являются случаи, где используется большое число временных таблиц, операций группировки и упорядочения. В этих случаях рекомендуется вернуться к соотношению 40:60.

Во всех случаях следует постоянно контролировать использование кэша и регулировать его параметры в соответствии с потребностями.

Использование памяти основного кэша оптимизированным индексом FP

В IQ 12 использование оптимизированного индекса Fast Projection (FP) подчиняется следующему правилу: при увеличении мощности связей и достижении предельного числа значений индекса (255 для однобайтового, 64K для двухбайтового) однобайтовый индекс FP (FP1) конвертируется в двухбайтовый (FP2), а двухбайтовый — в плоский.

В IQ 15, где введен оптимизированный 3-байтовый индекс FP, этот алгоритм меняется.

Для управления размером области основного кэша, используемой для хранения поисковых страниц оптимизированных индексов FP введены две новых опции: FP_LOOKUP_SIZE и FP_LOOKUP_SIZE_PPM. Первая из них задает объем оперативной памяти в абсолютном выражении (по умолчанию 16 Мбайт), выделяемый поисковым страницам, а вторая (2500 миллионных доли по умолчанию) определяет тот же объем в миллионных долях основного кэша.

При указании обеих опций FP_LOOKUP_SIZE и FP_LOOKUP_SIZE_PPM выбирается минимальное абсолютное значение. При достижении любым из оптимизированных индексов FP порогового значения он автоматически конвертируется в плоскую FP-структуру.

Для определения объема оперативной памяти, необходимой для хранения оптимизированного индекса FP, можно использовать следующий алгоритм:

Объем памяти в мегабайтах = ((NumDistinctValues * (Column_Data_Size + Cardinality_Size))

/ 1024 / 1024), где:

- Column_Data_Size — размер типа данных в байтах (4 байта для integer, 10 байт для char(10) и т. п.);
- Cardinality_Size (размер мощности связей) — 4 или 8 байт (число разных значений столбца может храниться в 4-байтовом или 8-байтовом поле). Значение 8 всегда безопасно и потому мы будем использовать именно его.

Приведем несколько примеров:

- Для типа integer с 250 000 разных значений:

  (250 000 * (4 + 8)) / 1024 / 1024 = 2,86 Мбайт

- Для типа integer с 2 000 000 разных значений:

  (2 000 000 * (4 + 8)) / 1024 / 1024 = 22,88 Мбайт

- Для типа varchar(50) с 250 000 разных значений:

  (250 000 * (50 + 8)) / 1024 / 1024 = 13,82 Мбайт

- Для типа varchar(50) с 2 000 000 разных значений:

  (2 000 000 * (50 + 8)) / 1024 / 1024 = 100,62 Мбайт

Как видно, при обработке типов данных большой ширины стандартных значений опций FP_LOOKUP_SIZE и FP_LOOKUP_SIZE_PPM может быть недостаточно. При использовании значения по умолчанию — обычно 16 Мбайт — для области хранения индекса FP таблица с 2 млн. различных целых потребует 22 Мбайт, так что оптимизированный 3-байтовый индекс FP будет конвертирован в плоский.

В таблице ниже, заимствованной из второго тома Руководства по администрированию Sybase IQ (Sybase IQ Administration Manual), приведены максимальные количества различных значений каждого столбца для разных значений FP_LOOKUP_SIZE. Предполагается, что размер мощности связей — 8 байт. Для расчета используется следующая формула:

Cardinality = FP_LOOKUP_SIZE / (Column_Data_Size + Cardinality_Size),

при этом FP_LOOKUP_SIZE преобразуется в байты.

Таблица 1. Максимальное количество уникальных значений индекса FP(3)

	Ширина типа данных столбца, байт					
FP_LOOKUP_SIZE	**4**	**8**	**32**	**64**	**128**	**255**
1 Мбайт	87381	65536	26214	14563	7710	3986
4 Мбайт	349525	262144	104857	58254	30840	15947
8 Мбайт	699050	524288	209715	116508	61680	31895
16 Мбайт	1398101	1048576	419430	233016	123361	63791
32 Мбайт	2796202	2097152	838860	466033	246723	127583
64 Мбайт	5592405	4194304	1677721	932067	493447	255166
128 Мбайт	11184810	8388608	3355443	1864135	986895	510333
256 Мбайт	16777216	16777216	6710886	3728270	1973790	1020667

Настройка обработки запросов опцией Max_Hash_Rows

Текущая версия IQ имеет опцию Max_Hash_Rows, определяющую долю кэша, которую можно использовать для выполнения различных алгоритмов соединения хэшированием в ходе исполнения. Значение по умолчанию установлено как для систем с 4 Гбайт ОЗУ. Большинство современных серверов оснащено значительным объемом ОЗУ, и для повышения пропускной способности значение этой опции следует соответственно увеличить.

Для задания нового значения Max_Hash_Rows в качестве отправной точки можно использовать как раз эти 4 Гбайт:

Новое значение = 2,5 млн. * (IQ_memory / 4 Гбайт),

где IQ_memory равно сумме объемов основного и временного кэша IQ в гигабайтах.

Кэш каталога

Кэш каталога — это область памяти, зарезервированная для работы, которая должна выполняться вне среды IQ. Каталог — это участок механизма IQ, управляемый и поддерживаемый программой Sybase SQL Anywhere (также называемой ASA или SA). Каталог обрабатывает все операции, не связанные с объектами и структурами данных Sybase IQ, кроме того, он управляет выдачей результатов клиентскому приложению.

Обычно каталог используется с невысокой интенсивностью, однако на случай интенсивного его использования полезно знать, как определять его размер.

В большинстве систем объем кэша каталога следует устанавливать в 2—8 раз большим размера файла каталога. Если файл каталога (файл .db) имеет размер 25 Мбайт, то размер кэша каталога следует установить в 50—200 Мбайт. В системах, обслуживающих более 50—100 одновременно работающих пользователей, коэффициент следует увеличить с 2—8:1 до 4—16:1.

Размер кэша каталога определяет также, до известной степени, число подключений и пользователей, которые могут активно выдавать запросы к IQ или ASA. Следствием недостаточного размера кэша могут быть снижение быстродействия или даже ошибки из-за нехватки ресурсов.

В сервере Sybase IQ объем кэш-памяти каталога, доступный каждому активному запросу, вычисляется с учетом параметров серверного кэша (–c) и числа активных запросов (–gn). Если через подключение поступил запрос, а кэш-память исчерпана, сервер отклонит запрос, выдав сообщение об ошибке «Statement size or complexity exceeds serverlimits» («Уровень сложности команды превышает установленные предельные возможности сервера»). Для устранения этой ошибки можно как увеличить объем кэша каталога, так и уменьшить допустимое число активных запросов, задав при запуске сервера соответствующие опции.

Как правило, параметр –gn неявно задается стартовым скриптом IQ как –gm + 5. Если –gm установлен в 100, то значение –gn по умолчанию будет равняться 105.

Лучший способ определить объем кэша каталога, используемого запросами — измерить его на основании типичной дневной рабочей нагрузки. Для сбора требуемой статистики предусмотрены специальные свойства подключения и сервера:

- UnschReq — число запросов, находящихся в очереди и ожидающих свободного серверного потока.
- ActiveReq — число серверных потоков, занятых обработкой запросов.
- LockedHeapPages — число заблокированных динамических страниц в кэше.
- CurrentCacheSize — текущий размер кэша в килобайтах.

- PageSize — размер страниц кэша сервера БД в килобайтах (страниц каталога, а не IQ).

Средний уровень использования кэша запросами вычисляется на основании текущего использования кэша и числа активных запросов по следующей формуле:

PagesPerRequest = LockedHeapPages / ActiveReq

Указанные свойства можно получить с помощью функции property() SQL Anywhere. Поскольку их значения постоянно меняются, для оценки максимальной нагрузки следует запрашивать их периодически.

Нижеследующий фрагмент кода на SQL считывает 100 образцов с 5-секундным интервалом и вычисляет максимальное использование кэша:

```
begin
        declare c1 int;
        declare local temporary table #tmp1(ar bigint, lp bigint) in SYSTEM;
        set c1 = 0;
        lp: loop
        set c1 = c1 + 1;
        if c1 > 100 then
                leave lp
        end if;
        insert #tmp1 select property('ActiveReq'), property('LockedHeapPages');
        waitfor delay '00:00:05';
        end loop;
        select max(lp) as MaxLockedHeapPages,
        max(ar) as MaxActiveReq,
        max(lp)/max(ar) as AvgPagesPerRequest
        from #tmp1;
end;
```

Теперь максимальное число запросов, которые могут исполняться одновременно, можно вычислить, используя общий объем кэша и объем, потребляемый запросом:

MaxActiveReq = CurrentCacheSizeKB / PageSizeKB / AvgPagesPerRequest

Пусть объем кэша установлен в 256 Мбайт, размер страницы каталога — 4 Кбайт, а среднее число страниц на запрос — 500. Тогда максимальное число одновременных активных запросов будет таким:

MaxActiveReq = (256 * 1024) / 4 / 500;

MaxActiveReq = 131,072 ≈ 131 (округлено в сторону уменьшения).

Управление версиями

Память для управления версиями выделяется на время работы механизма Sybase IQ. Как правило, это очень небольшая область ОЗУ (1—3 Кбайт на каждую сохраняемую версию). Обычно общий объем памяти для управления версиями исчисляется килобайтами или единицами мегабайт. Однако в системах с сотнями, тысячами или миллионами активных версий объем этой памяти может достигать значительной величины.

Загрузочная память

Начиная с версии 15.0 Sybase IQ отменена большая часть требований к загрузочной памяти для операций массовой загрузки.

В версии 15.2 отменены все требования к загрузочной памяти.

В IQ 15.0 и 15.1 загрузочная память используется только для загрузки данных фиксированной ширины. Все операции по загрузке данных с разделителями и двоичных данных используют в качестве загрузочной памяти временный кэш IQ.

При загрузке данных фиксированной ширины следует рассчитывать объем загрузочной памяти, как и раньше. Алгоритмы и указания по расчету приведены ниже.

Загрузочная память для загрузки данных фиксированной ширины — это фиксированный объем памяти, выделяемый для каждой массовой загрузки и вычисляемый по особой формуле. Эта память находится вне кэшей IQ и выделяется с помощью функций выделения памяти операционной системы. По окончании загрузки память высвобождается. Формула расчета объема загрузочной памяти следующая:

TableWidth * 10 000 * 45 / 1024 / 1024, где:

- TableWidth — ширина таблицы в байтах, как ее возвращает функция IQ tablewidth();
- 45 — общее число корзин для хранения данных;
- 10 000 — число записей по умолчанию в каждой корзине.

Для задания предельного объема виртуальной памяти, который может использовать подключение, вводящее данные, можно динамически задавать параметр периода исполнения Load_Memory_MB. Его максимальное допустимое значение

— 2000 (Мбайт). Если Load_Memory_MB имеет ненулевое значение, а используемая память вышла за указанный предел, операция ввода будет прекращена с выдачей сообщения об ошибке. Если задано нулевое значение Load_Memory_MB, значит, ограничение на использование виртуальной памяти отсутствует, и IQ будет использовать столько памяти, сколько потребует алгоритм, буферизующий вводимые данные.

Однако весь объем памяти, вычисляемый по алгоритму, требуется не всегда. Вне зависимости от того, вычисляется ли объем загрузочной памяти системой по вышеприведенной формуле или задается ли он пользователем посредством опции, загрузочная память ограничена областью ОЗУ, выделяемой для загрузки данных. В ходе загрузки данных IQ динамически выделяет виртуальную память каждой корзине для кэширования поступающих данных в одной из 45 возможных корзин. Если загрузочный файл содержит 20 000 записей, то по умолчанию будут выделены только 2 корзины (по 10 000 записей в каждой).

Механизм выделения корзин по мере потребности повышает эффективность использования виртуальной памяти. Кроме того, он позволяет разбить процесс загрузки данных на две параллельные задачи: собственно ввода и построения индексных структур. Для построения разных типов индексов требуется проделать разный объем работы, поэтому время построения различается. Например, индекс FP или LF строится значительно быстрее, чем HG или DATE. Использование корзин позволяет кэшировать данные, благодаря чему в процессе построения индекса нет необходимости ожидать готовности диска и завершения операций считывания данных; кроме того, индексам для оптимальной работы не требуется ждать завершения каких-либо процессов, помимо собственно ввода данных.

При расчете требуемого объема загрузочной памяти целесообразно ориентироваться на то, что в ходе загрузки будут выделяться все 45 корзин. Возможно, серверу они и не потребуются, однако соответствующий запас виртуальной памяти гарантирует корректную работу в любых обстоятельствах.

Использование кэш-памяти во время загрузки данных

Следует подчеркнуть, что в процессе загрузки данных используется также память из области основного кэша. В зависимости от требований к памяти, предъявляемых системой, а также числа столбцов в загружаемых таблицах, это может повлиять на необходимый для основного кэша минимальный объем ОЗУ.

IQ выделяет по одной странице для каждого индекса FP и по одной странице для каждого отдельного значения индекса LF. Для оптимизации быстродействия эта память выделяется непосредственно из основного кэша. Для систем с большим количеством одновременных операций загрузки или с большой шириной таблиц потребность в этой области памяти может быстро увеличиться. Если объем кэша окажется недостаточен для того, чтобы уместить эти страницы в ОЗУ, быстродействие снизится, так как IQ потребуется вновь считывать страницы с диска.

Индексы HG и WD используют в процессе загрузки временный кэш. Необходимый объем для конкретных индексов приближенно вычисляется по такой формуле:

```
( 8 + sizeof( datatype ) ) * numberRowsBeingLoaded
```

Загрузка 100 записей типа integer (размер 4 байта) потребует приближенно:

```
(8 + 4) * 100 -> 1 200 байт
```

Загрузка 100 записей типа varchar(20) потребует приближенно:

```
(8 + 20) * 100 -> 2 800 байт
```

Индекс WD использует существенно больше памяти, поскольку каждое слово (токен) значения данных занимает некоторый объем временного кэша. Каждый токен требует столько же памяти, сколько и индекс HG. Таким образом, общий объем требуемой памяти составит:

```
numberTokens * ( 8 + sizeof( datatype ) ) * numberRows
```

Возьмем для примера символьное поле char(20) с 4 токенами в каждой записи. Требуемый объем временного кэша в этом случае приближенно составит:

```
4 * (8 + 20) * 100 -> 11 200 байт
```

Поскольку фактический объем используемой памяти зависит от числа загружаемых токенов, потребление памяти индексом WD предсказать весьма трудно.

Память битовых массивов

На время загрузки данных выделяется также дополнительный объем виртуальной, или динамической памяти, для хранения битовых массивов. Эта память выделяется помимо всей прочей памяти, выделенной для IQ.

Опция Load_Memory_MB задает количество виртуальной памяти, которую команда загрузки может использовать для сущностей, информация о которых уже имеется. Однако есть и такие сущности, информация о которых становится доступна только в момент загрузки данных. Они представлены в форме битовых массивов, хранящих разные значения загружаемых данных.

Память битовых массивов при загрузке данных необходима при использовании следующих типов индексов: LF, HNG, DTTM, DATE, TIME и CMP.

В качестве примера индекса с битовыми массивами рассмотрим индекс LF. Для каждого отдельного значения в нем будет свой битовый массив. Для каждого отдельного значения и группировки записей в процессе загрузки будет установлен бит в соответствующем массиве — именно в том, который связан с отдельным значением.

Установка битов в массиве поодиночке — довольно медленная операция. Для повышения быстродействия в ходе загрузки данных биты в массиве устанавливаются группами. Группы хранятся в виртуальной памяти. Размер хранилища для одной группы — 8 192 байта.

Как это влияет на объем загрузочной памяти? Пусть есть индекс LF, охватывающий в ходе загрузки N разных значений. Тогда объем виртуальной памяти, используемой этим индексом, составит:

(8 192 * N) байт.

Если имеется 500 индексов LF и каждый столбец содержит N разных значений, требуемый объем виртуальной памяти составит:

8 192 * 500 * N = 4 096 000 * N.

Как видно, даже относительно небольшое число разных значений потребует заметного объема вирутальной памяти. Так, для 100 различных значений в каждом столбце в сумме потребуется около 400 Мбайт (4 096 000 * 100 = 409 600 000 байт).

Поскольку до загрузки загрузочный механизм не располагает информацией о количестве различных значений, Sybase IQ не может заранее предсказать, сколько виртуальной памяти потребуется для индекса LF.

Для индексов LF объем битового кэша определяется параметром LF_BITMAP_CACHE_KB. Значение по умолчанию — 4 Кбайт с максимумом 8 Кбайт при группировке.

Используются и другие, относительно малые области кэш-памяти, при этом их количество и размер не зависят от числа различных значений. Например, индексы HG/WD используют некоторый объем кэша для обновления страниц Btree в ходе загрузки. Эти дополнительные кэш-области виртуальной памяти относительно малы и незначительно влияют на общий объем выделяемой памяти.

Если объемы загрузочной памяти и соответствующей памяти битовых массивов рассчитаны недостаточно точно, то в процессе загрузки данных возможны операции с файлом подкачки (обмен страницами между оперативной памятью и диском) на уровне операционной системы. Их можно отследить с помощью средств ОС по контролю за использованием памяти (так, информацию об операциях с файлом подкачки выдает программа vmstat) и подсистемой ввода-вывода (активность устройств подкачки можно зафиксировать с помощью программы iostat). Кроме того, для выявления нехватки памяти можно использовать хранимую процедуру sp_iqstatus. В строке «IQ Dynamic Memory» будет выведен текущий и максимальный объемы оперативной памяти, которые были выделены IQ для всех областей памяти (кэши, загрузочная память, резервная память и память битовых массивов). Если в процессе наблюдения этих показателей окажется, что цифра превышает физический объем ОЗУ или объем ОЗУ, который система предположительно должна была использовать, значит, требуется вмешательство.

Если утилиты покажут, что ОС интенсивно работает с файлом подкачки (или вообще задействует устройство подкачки), это означает, что памяти не хватает и необходимо принять меры к повышению быстродействия: увеличить объем оперативной памяти на машине, уменьшить предельно допустимое количество загрузочной памяти, либо уменьшить размер основного и временного кэшей.

Память для резервного копирования

В идеальной ситуации количество виртуальной памяти, используемой процессом резервного копирования, является функцией следующих переменных:

- число ЦП;
- число пространств БД главного или локального хранилища, подлежащих резервному копированию;
- блок-фактор (число блоков, записываемых за один раз);
- размер блока IQ (выводится в поле ‘block_size’ в sys.sysiqinfo).

Приближенно требуемый объем виртуальной памяти можно вычислить по следующему алгоритму:

y = max(2 * number_of_cpus, 8 * number_of_main_or_local_dbspaces),

z = (y * 20) * (block factor * block_size),

где z — грубая оценка объема виртуальной памяти, используемой при резервном копировании.

Пример: пусть система имеет следующие характеристики:

- число dbspace = 50;
- блок-фактор = 100;
- число ЦП = 4;
- размер блока = 8 192.

Тогда, если руководствоваться вышеприведенными допущениями,

y = max(8, 400) = 400,

и количество ОЗУ, необходимое для резервного копирования, составит:

z = (400 * 20) * (100 * 8 192) = 6,5 Гбайт.

Весь этот объем выделяется из пула операционной системы: это динамическая память, рассматриваемая как загрузочная в том отношении, что она высвобождается лишь тогда, когда операция будет закончена.

Единственный параметр, которым здесь можно манипулировать — блок-фактор, впрочем, этого вполне достаточно. Если в нашем примере изменить его значение на 10, тогда объем требуемой памяти сократится до:

(400 * 20) * (10 * 8 192) = 655 Мбайт.

Уменьшение допустимого объема памяти может замедлить резервное копирование (в случае если подсистема дискового ввода-вывода не является самым узким местом). Скорость резервного копирования определяется скоростью считывания и записи блоков подсистемой дискового ввода-вывода. Для сокращения числа операций ввода-вывода процесс резервного копирования пытается считывать и записывать блоки последовательными группами за одну операцию.

Итак, для сокращения или увеличения доступного объема памяти резервного копирования можно манипулировать значением блок-фактора. Как малое, так и большое значения имеют свои компромиссы относительно затрат и быстродействия, и оптимальный вариант можно выбрать лишь с учетом этих компромиссов.

Полная карта распределения памяти

Для каждой одновременной операции загрузки следует вычислить требуемое количество загрузочной памяти и полученные значения просуммировать. Сумму следует затем вычесть из общего объема ОЗУ, который может использовать IQ. Остаток может быть выделен кэшам IQ.

Операционная система	**От 0,5 до 1 Гбайт ОЗУ**
Кэш файловой системы	**20% ОЗУ**
Прочие программы	
Память каталога IQ	**параметры –c/–cl/–ch**
Память потоков IQ	**размер стека * число потоков**
Загрузочная память	**на одновременную загрузку**
Память битовых карт	**на одновременную загрузку**
Основной кэш IQ	**40% оставшегося ОЗУ**
Временный кэш IQ	**60% оставшегося ОЗУ**
Память резервного копирования	**На каждый экземпляр**

Требования к оперативной памяти

Есть два подхода к определению требуемого размера ОЗУ для IQ: точно рассчитать объем всех составляющих и вычислить в соответствии с этим объем памяти, либо взять приблизительно подходящие объемы и регулировать их по мере необходимости. Расчеты размеров различных областей памяти мы рассмотрели во всех подробностях. Если эта процедура обещает занять слишком много времени или вообще выглядит невыполнимой, можно поступить иначе: выделить память исходя из взвешенной оценки и изменять параметры, руководствуясь статистикой использования.

При приблизительной оценке объемов памяти руководствуются следующим правилом: основному и временному кэшам IQ надо выделять не более 2/3 общего объема ОЗУ. Так, в системе с 64 Гбайт памяти совокупный размер основного (–iqmc) и временного (–iqtc) кэшей не должен превышать 48 Гбайт.

При этом IQ получит вполне достаточный объем ОЗУ для своих операций, не перегружая систему. Поскольку области кэш-памяти IQ занимают порядка 80% всей выделяемой памяти или более, слишком большой их размер может привести к значительному снижению быстродействия, так как операционная система будет вынуждена выгружать страницы на диск.

Определение размера файла подкачки

Обычно рекомендуется устанавливать размер файла подкачки вдвое большим объема установленной оперативной памяти. Однако в системах с большим объемом ОЗУ (64 Гбайт или более) должно быть достаточно файла размером с ОЗУ. Тогда, если будет задействован механизм подкачки, размер буфера будет достаточно велик, чтобы администраторы смогли найти и устранить причину нехватки памяти. Как было показано в предыдущих разделах, IQ может оперировать значительным объемом ОЗУ вне областей основного и временного кэшей — загрузочной памятью, памятью битовых карт и памятью резервного копирования. Чтобы предотвратить ситуацию, когда вся виртуальная память будет использована, размер файла подкачки должен быть примерно вдвое большим объема ОЗУ. При этом система будет иметь достаточно ресурсов, чтобы компенсировать непредвиденные ситуации, когда объем динамической памяти превышает объем ОЗУ. Конечно, быстродействие снизится, однако работа не остановится по причине нехватки памяти (физической или виртуальной).

Пример

Возьмем для примера две разные системы: одну с 16 Гбайт ОЗУ и другую с 32 Гбайт ОЗУ. Предположим, что будет пять одновременных процессов загрузки, и размер загрузочной памяти установим в 200 Мбайт. Предположим также, что при загрузке не потребуется значительного объема памяти битовых массивов.

	Система с 16 Гбайт	**Система с 32 Гбайт**
Операционная система	1 Гбайт	1 Гбайт
Кэш файловой системы	1 Гбайт	2 Гбайт
Прочие программы	500 Мбайт	500 Мбайт
Память каталога IQ	128 Мбайт	256 Мбайт
Память потоков IQ	250 Мбайт	500 Мбайт
Загрузочная память	1 Гбайт	1 Гбайт
Память битовых массивов	256 Мбайт	256 Мбайт
Основной кэш IQ	4,4 Гбайт	10,4 Гбайт
Временный кэш IQ	6,6 Гбайт	15,6 Гбайт
Память резервного копирования	—	—
Итого	**15,125 Гбайт**	**31,5 Гбайт**

Работа Sybase IQ с диском

В Sybase IQ используются хранилища двух основных типов: основное хранилище (main store) и временное хранилище (temporary store).

На устройствах основного хранилища (и его контрагента на считывающем узле многоузловой системы — локального основного хранилища) находятся данные пользователя и индексы, являющиеся частью пользовательских таблиц. Данные в основном хранилище хранятся постоянно, оставаясь там и после перезагрузки системы.

Устройства временного хранения используются для размещения данных и индексов в составе временных и глобальных временных таблиц. В ходе загрузки данных временное хранилище применяется для кэширования промежуточных данных, подлежащих загрузке в индексы HG и WD. При обработке запросов оптимизатор использует временное хранилище для сортировки и вспомогательных репозиториев, создаваемых лишь на время выполнения конкретных операций. Такие репозитории используются для обновляемых курсоров, промежуточных наборов результатов и управления индексами соединений (команда SYNCHRONIZE JOIN INDEX).

Операции чтения-записи

Все операции чтения-записи Sybase IQ выполняет через кэш — непосредственно на диск никакие данные не попадают. Операции с основным и временным хранилищем выполняются через буферный кэш IQ. На диск данные из кэша записываются при выполнении одного из следующих двух условий:

- количество модифицированных страниц кэша превысило допустимый предел;
- выполняется фиксация транзакции, изменившей таблицу.

Если объем памяти для основного или временного кэша достаточно велик, хранилище будет использоваться редко. Поскольку в большинстве систем недостаточно памяти для кэширования всех операций, следует тщательно продумать распределение места на диске и выбрать местоположение хранилища так, чтобы обеспечить оптимальное быстродействие.

После того как данные попали в кэш, они будут записаны на диск. Подготавливая страницу памяти (содержащую пользовательские данные, индексы, битовые массивы и т. п.) к записи, IQ сжимает ее по специальному алгоритму. Размер страницы делится на 16 без остатка (по умолчанию — 16 блоков на страницу) и кратен размеру полученного в результате блока данных. Для 256-килобайтных страниц размер каждого блока будет равен 16 Кбайт. Чтение и запись всегда выполняются массивами, кратными длине блока. Длина массива может быть минимум 1 блок (максимальное сжатие) или максимум 16 блоков (без сжатия). Как чтение, так и запись осуществляются единственной операцией ввода-вывода. Коэффициент сжатия и, следовательно, размер записываемого массива определяются характером данных страницы.

Страницы, которые сжимаются до одного блока, встречаются крайне редко, так что IQ весьма редко считывает или записывает массив данных размером в 1 блок. Контроль работы подсистемы хранения покажет, что считываются и записываются массивы переменной длины. Степень различия длины будет зависеть от уровня сжатия.

IQ_SYSTEM_MAIN

Говоря о IQ 12.7, мы оперировали понятием пространств БД (dbspaces). Пространство БД — это плоский файл или raw-устройство, отображение в которые выполняется один к одному.

В IQ 15 пространства БД организованы иначе. Dbspace — это логический контейнер для хранения данных, который может состоять из одного или более плоских файлов или raw-устройств. С точки зрения системы dbspace состоит из файлов. Файлы (иногда называемые БД-файлами — dbfiles) являются физическими хранилищами, а пространства БД — логическими. Файл в IQ может быть как файлом файловой системы, так и raw-устройством.

В IQ 15 может существовать лишь одно временное пространство БД. Как и прочие пространства БД, оно может состоять из одного и более файлов (плоских файлов или raw-устройств). Чтобы добавить к dbspace файл (и таким образом увеличить размер временного хранилища IQ), используют команду ALTER DBPSACE.

Значение понятия «dbspace» меняется в зависимости от используемой версии IQ. В Sybase IQ 12.7 dbspace соответствует одному файлу базы данных. При включенной опции DEFAULT_DISK_STRIPING эта система автоматически распределяла данные между всеми доступными пространствами БД в целях оптимизации быстродействия и простоты администрирования, при этом назначить какое-либо dbspace конкретной таблице или индексу невозможно.

Под файлом, с присвоенным ему логическим именем файла и соответствующим путем, понимается каждый файл операционной системы, используемый для хранения данных.

Все имена dbspace, файлов и физические пути к файлам должны быть уникальны. Имя файла может совпадать с именем пространства БД.

Хранилищем (store) называется одно или более пространств БД, где хранятся постоянные или временные данные, используемые для конкретных целей. В Sybase IQ есть три типа хранилищ:

- хранилище каталога — содержит пространство БД SYSTEM и до двенадцати дополнительных пространств БД;
- главное хранилище IQ — содержит пространство БД IQ_SYSTEM_MAIN и пользовательские пространства БД;
- временное хранилище IQ — содержит пространство БД IQ_SYSTEM_TEMP.

Настоятельно рекомендуется не хранить в IQ_SYSTEM_MAIN данные пользователей. Для этих целей в Sybase IQ предусмотрено создание пользовательских пространств БД.

IQ_SYSTEM_MAIN лучше всего рассматривать как системную область, подобную master-области (и файлу master.dat) в Sybase ASE, ни при каких обстоятельствах не предназначенную для хранения пользовательских данных. В большинстве реляционных СУБД, как правило, данные и структуры пользователей не смешивают с данными и структурами системы. Как в 15-й, так и в последующих версиях Sybase IQ этот принцип должен остаться без изменений.

В обновленном IQ_SYSTEM_MAIN помещаются пространство БД и свободный список файлов. Там же находится пространство для управления версиями, осуществляется часть межузловых коммуникаций, воспроизведение TLV и другие функции. По умолчанию для этих целей зарезервировано 20% IQ_SYSTEM_MAIN. Если для хранения пользовательских данных требуется дополнительное место в IQ_SYSTEM_MAIN, оно будет зарезервировано. При увеличении IQ_SYSTEM_MAIN следует остановить весь кластер и синхронизировать узлы.

Именно по этим причинам и рекомендуется оставить IQ_SYSTEM_MAIN роль исключительно системной области с минимумом пользовательских данных или вообще без таковых.

Определение характеристик процессоров и ядер

Загрузка и изменение данных

Единичные изменения данных

Единичное изменение данных (модификация одной записи в таблице) в Sybase IQ выполняется посредством команд вставки (INSERT), изменения (UPDATE) и удаления (DELETE). Количество ЦП и объем оперативной памяти для этих операций большой роли не играет. Чтобы обслуживать таким образом пользователя, достаточно одного-двух ЦП. Следует заметить, однако, что подобный способ изменения данных в Sybase IQ неоптимален и к нему следует прибегать лишь тогда, когда не требуется высокого быстродействия.

Операции массовой загрузки

Под операциями массовой загрузки понимаются все операции загрузки данных, выполняемые с помощью команд LOAD TABLE, INSERT FROM LOCATION и INSERT SELECT. При этом массовый загрузчик IQ применяет к базе данных множество изменений сразу. В эту категорию попадают также операции обновления и удаления, затрагивающие несколько записей, благодаря своей параллельной природе. (Мы относим их сюда по логическому признаку, отсюда не следует, что для команд изменения и удаления используется массовый загрузчик IQ.)

Расчет характеристик системы IQ для загрузки данных очень прост. Количество процессорных мощностей, необходимое для единичной операции загрузки с достаточным быстродействием, выводится непосредственно из числа индексов в таблице.

Порядок приблизительного расчета количества ЦП при условии выполнения операций загрузки в монопольном режиме:

- 1 ЦП на каждые 5-10 столбцов в загружаемой таблице (индекс FP по умолчанию);
- 1 ЦП на каждые 5-10 индексов (HNG, LF, CMP, DATE, TIME, DTTM) в таблицах, информация о которых отсутствует.
- Индексы HG, WD и TEXT подлежат массово-параллельной обработке и могут, в целях повышения быстродействия, использовать все доступные мощности ЦП сервера. Правило приближенного расчета здесь таково: одно процессорное ядро на каждые 1-2 индекса HG, WD или TEXT. Когда известен шаблон загрузки, можно увеличить или уменьшить количество процессорных ресурсов в соответствии с потребностями.

Следует подчеркнуть, что перечисленные правила рассчитаны на системы, в которых операции загрузки занимают почти 100% системы-источника и выполняются в монопольном режиме. Они должны быть сбалансированы по отношению к периодам времени, в которые будет выполняться загрузка.

Для расчета характеристик системы можно использовать и другой алгоритм, который основывается на объеме подлежащих загрузке в IQ исходных данных. При этом ставится целью скорее обеспечить загрузку в заданный срок, чем обеспечить максимальное быстродействие системы.

- В системах с 4 или менее ЦП ожидаемая скорость загрузки — приблизительно 10-15 Гбайт в час на каждый ЦП. 4-процессорная система должна загружать около 40 Гбайт исходных данных в час.
- В системах с 8 или более ЦП ожидаемая скорость загрузки — приблизительно 15-40 Гбайт в час на каждый ЦП. 8 процессорная система должна загружать 120—200 Гбайт исходных данных в час.

Время загрузки может меняться в широких пределах — в зависимости от числа процессоров, а также количества и типов индексов загружаемых таблиц.

Sybase IQ поддерживает неструктурированные данные — двоичные и символьные большие объекты (BLOB и CLOB). При загрузке в IQ BLOB и CLOB скорость напрямую зависит от числа ЦП и производительности дисковой подсистемы (в том числе быстродействия дискового массива, числа шпинделей, пропускной способности шинных адаптеров и т. д.).

Для каждого загружаемого в IQ BLOB или CLOB необходим один ЦП. Система с 8 ЦП будет загружать одни и те же данные BLOB/CLOB примерно вдвое быстрее системы с 4 ЦП. При определении характеристик для работы с данными BLOB и CLOB зависимость от ожиданий пользователя к скорости обработки этих типов данных является квадратичной.

Начиная с 15-й версии Sybase IQ все операции загрузки, в том числе второй проход процесса загрузки индексов HG и WD, выполняются параллельно. Скорость загрузки индекса HG можно значительно увеличить не только путем выполнения приведенных здесь рекомендаций по сайзингу, но и простым увеличением процессорных ядер в записывающем узле. При этом обработка индекса будет в значительно большей мере распараллелена, благодаря чему время загрузки сократится.

Запросы

Чтобы лучше понять, сколько процессорных мощностей отнимает обработка каждого запроса, следует их классифицировать. При этом трудно выделить критерии, не наблюдая непосредственно за ходом обработки запросов. Нельзя, например, заранее утверждать, что все запросы с объединением 17 таблиц будут сложными и будут долго выполняться. Они могут быть обработаны как за 30 секунд, так и за 30 минут.

Обычно предполагают, что любой запрос на время своего исполнения занимает 1-2 ЦП. Время выполнения запроса может измеряться миллисекундами, секундами и даже часами — оно зависит от того, какой объем данных для этого необходимо считать. Для уточнения скорости обработки запросов и профилей их исполнения классифицируем запросы в зависимости от времени обработки.

Определим пять классов запросов:

- **сверхбыстрые** — запросы, которые обычно выполняются менее чем за 5 с;
- **быстрые** — обычно выполняются менее чем за 3 мин.;
- **средние** — обычно выполняются за 3-10 мин.;
- **долгие** — обычно выполняются за 10-30 мин.;
- **сверхдолгие** — обычно выполняются более чем за 30 мин.

Зададим для этих классов требования к одновременной обработке.

- **сверхбыстрые** — каждый ЦП должен обрабатывать 10 или более таких запросов одновременно;
- **быстрые** — каждый ЦП должен обрабатывать 5—10 запросов одновременно;
- **средние** — каждый ЦП должен обрабатывать 2—5 запросов одновременно;
- **долгие** — каждый ЦП должен обрабатывать 1—2 запроса одновременно;
- **сверхдолгие** — каждый ЦП должен обрабатывать максимум один запрос; как правило, исполнение запроса требует более одного ЦП.

При расчете следует учитывать возможность параллельного исполнения запросов. Если есть свободные ЦП и запрос может быть разбит на независимо выполняемые части, то оптимизатор может заменить последовательное выполнение несколькими параллельными процессами. Скорость обработки параллелизуемого запроса на системах с 4, 8 и 12 ЦП будет существенно различаться.

Общее быстродействие системы будет также меняться по мере увеличения числа пользователей одного экземпляра СУБД. При росте количества пользователей и параллельно выполняемых запросов будет уменьшаться пул свободных ресурсов, необходимых системе для обработки запросов.

15-й выпуск Sybase IQ продолжает совершенствоваться, и число составляющих запросы операций, которые могут выполняться одновременно, увеличивается. Приведенные выше рекомендации — хорошая отправная точка для определения характеристик системы, однако важно помнить, что каждый отдельный пользователь выигрывает при увеличении количества свободных процессорных ядер.

Реализованный в IQ 15 параллелизм имеет целью достичь равновесия между скоростью обслуживания отдельных пользователей и способностью системы обслуживать множество пользователей одновременно. Для этого применяется динамическое перераспределение мощностей в процессе работы системы. Например, первый пользователь, обратившийся к системе, получает в свое распоряжение все ядра, необходимые для операции загрузки или обработки запроса. Второму пользователю не придется ожидать завершения обслуживания первого пользователя: оптимизатор высвободит часть мощностей, занятых первым пользователем, и передаст их второму, так что процессорные ресурсы будут распределены между обоими пользователями пропорционально. Аналогичным образом система поступит при подключении третьего пользователя, так что в результате на каждого пользователя придется примерно по одной трети вычислительных мощностей. По завершении счета ресурсы высвобождаются и передаются выполняющимся запросам.

Определение объема памяти

Загрузка и изменение данных

Единичные операции

Как правило, единичные (затрагивающие одну запись) операции в IQ не требуют значительного объема памяти. Количество памяти, необходимой для каждой единичной операции вставки, изменения или удаления, рассчитывается по тому же алгоритму, что и для массовых операций. Разница заключается лишь в том, что при выполнении единичных операций изменения вносятся только в одну запись.

Массовые операции

Как правило, чем больше объем ОЗУ, тем лучше. Однако для операций загрузки это утверждение справедливо не всегда. Для хранения данных, используемых для построения индексов HG и WD, необходимо задать большой объем временного кэша. Основной кэш должен быть достаточно велик, чтобы уместить страницы-заголовки для всех индексов и их не требовалось бы подгружать с диска в случае, если они окажутся выгружены туда в результате операций подкачки. Кроме основного и временного кэшей, следует предусмотреть достаточно ОЗУ вне области IQ для загрузочной памяти, которая выделяется при каждой загрузке данных (до версии 15.2).

Во время загрузки промежуточные данные, необходимые для индексов HG, хранятся во временном кэше. При первом проходе процесса загрузки туда помещаются данные и информация, требуемые для построения индекса HG из поступающих данных. На втором проходе эти данные объединяются с имеющимися структурами индекса и затем записываются на диск. Если объем временного кэша окажется недостаточен для хранения данных на первом проходе, сервер запишет страницы во временное хранилище на диске. Для оптимизации загрузки рекомендуется задать такой размер временного кэша, чтобы выгрузка страниц из памяти на диск была минимизирована или вообще исключена. При загрузке больших объемов данных накладные расходы могут оказаться велики, так что следует оценить необходимое быстродействие и ресурсы и найти компромиссный вариант.

Определение объема временного кэша осуществляется весьма просто. Нижеприведенная формула применяется ко всем загружаемым столбцам, содержащим индекс HG или WD:

total_pages = 1 + ((number_of_rows_in_load_files * width_of_columns_in_hg) / page_size_in_bytes), где:

- **number_of_rows_in_load_files** — общее число записей, загружаемых командой LOAD TABLE или INSERT;
- **width_of_columns_in_hg** — совокупная ширина в байтах всех столбцов индекса HG. Для индексов HG (первичный ключ, ограничение уникальности, внешний ключ и т. д.), поддерживающих несколько столбцов, следует учитывать каждый столбец;
- **page_size_in_bytes** — размер страницы в байтах, указанный при создании БД.

Для примера предположим, что в таблицу с одним индексом HG для столбца типа integer загружается 10 млн. записей, а размер страницы БД — 256 Кбайт. Общий объем временного кэша, необходимый для загрузки данных, рассчитывается так:

total_pages = 1 + ((10 000 000 * 4) / 262 144);

total_pages = 1 + (40 000 000 / 262 144);

total_pages = 153 (256-килобайтных) страниц, или 38 Мбайт.

Если тот же объем данных будет загружаться в ту же самую таблицу, но индексом будет первичный ключ для столбцов типа integer и char(10), то необходимый объем памяти для индекса будет таким:

total_pages = 1 + ((10 000 000 * (4 + 10)) / 262 144);

total_pages = 1 + (140 000 000 / 262 144);

total_pages = 535 (256-килобайтных) страниц, или 134 Мбайт.

При определении размера основного кэша отталкиваются от типов индексов, примененных в загружаемой таблице:

- FP — 1 страница для каждого индекса FP (плюс 3 во временном кэше для каждого оптимизированного FP);
- LF — 1 страница для каждого отдельного значения, загружаемого в индекс LF;
- HNG, DATE, TIME и DTTM — 1 страница для каждого бита в битовом массиве;
- CMP — 3 на индекс;
- HG и WD — на первом проходе используется временный кэш (см. выше), на втором, при построении окончательных страничных структур, — основной кэш. Поскольку используется временный кэш, требования к основному кэшу для индексов HG и WD минимальны.

Примерное правило — 5-10 страниц основного кэша для каждого индекса загружаемой таблицы (учитываются все типы индексов, в том числе индекс FP по умолчанию). Если загружается большое число взаимно различных значений, эту цифру следует увеличить. Мы привели лишь приблизительную оценку для использования в качестве отправной точки.

Для таблицы с 50 столбцами и 25 дополнительными индексами расчет будет таким:

(50 индексов FP + 25 дополнительных индексов) * (5,10) -> 375-750 страниц;

375 страниц * размер страницы 128K -> 46,875 Мбайт;

750 страниц * размер страницы 128K -> 93,75 Мбайт.

Запросы

При определении объема памяти для обработки запросов базовое правило следующее: на каждый ЦП должно приходиться минимум 4-8 Гбайт ОЗУ. Для систем с малым количеством ЦП (менее 8) рекомендуется выбирать объем ОЗУ ближе к 8 Гбайт в расчете на процессор.

Конфигурация с несколькими серверами

Sybase IQ может работать на нескольких серверах, совместно использующих основное дисковое пространство. Такую конфигурацию называют Sybase IQ Multiplex, мультиплексом или просто многоузловой конфигурацией.

Выполняя вышеприведенные указания относительно требований к ЦП для обработки запросов, можно разместить необходимое количество ЦП на разных машинах — нет необходимости иметь серверы с множеством процессоров. Данный метод позволяет также распределять нагрузку между серверами. На прикладном уровне можно также назначить конкретным серверам функции обработки конкретных типов запросов.

Предположим для примера, что для хранилища данных требуются 16 ЦП. Для большей части запросов все равно, будут ли все эти процессоры размещаться на одном или на нескольких машинах. Можно использовать 4 системы, каждая из которых будет иметь 4 ЦП и 16-32 Гбайт ОЗУ. Такой вариант может быть значительно дешевле, чем одна 16-процессорная система с 64-128 Гбайт ОЗУ.

Определение характеристик системы хранения

Определение размера IQ_SYSTEM_MAIN

Необходимый размер основного хранилища по умолчанию определяется относительно просто — он основывается на общем размере пользовательских пространств БД, а также зависит от числа узлов в мультиплексе.

Для БД объемом менее 100 Гбайт рекомендуется устанавливать размер IQ_SYSTEM_MAIN по меньшей мере 4 Гбайт; как правило, его объем должен составлять 5-10% объема пользовательского основного пространства (5-10 Гбайт для 100-гигабайтной базы данных). Если этот экземпляр БД переносится на мультиплекс, то необходимо добавить по 1 Гбайт на каждый узел. Таким образом, для двухузловой системы со 100-гигабайтной базой данных размер основного хранилища должен составлять 7-12 Гбайт.

Для баз данных объемом более 100 Гбайт рекомендуется устанавливать размер IQ_SYSTEM_MAIN по меньшей мере 8 Гбайт для одномашинной конфигурации и 16 Гбайт для мультиплекса. В этом случае объем IQ_SYSTEM_MAIN должен составлять 1-2% пользовательского основного пространства (10-20 Гбайт для БД объемом 1 Тбайт). Если такая же база развернута на мультиплексе, в расчете на каждый узел следует добавить 0,1-0,03% дискового пространства (1-3 Гбайт на 1 Тбайт). Так, для 4-узловой системы с 1-терабайтной базой данных размер основного хранилища должен составлять минимум 16 Гбайт плюс 1-3 Гбайт на узел, то есть всего 20-28 Гбайт.

Общие указания по системе хранения

Требования к диску для Sybase IQ меняются по мере роста быстродействия ЦП. Приведенные в данном разделе значения являются базовыми величинами для всех типов систем. Если создаваемая система должна иметь особо высокое быстродействие, то возможно, некоторые значения целесообразно увеличить — с учетом увеличенного объема работы, выполняемого благодаря более быстрому ЦП и большему объему ОЗУ. Как правило, быстродействие Sybase IQ зависит от ЦП, а не от диска или системы ввода-вывода. По мере роста скорости работы и пропускной способности ЦП узкое место смещается ближе к дисковой подсистеме.

При наличии нескольких шинных адаптеров FC они должны быть установлены на разных шинах, чтобы не было задержек при передаче по шине.

Каждый ЦП может обрабатывать (в среднем) 10-20 Мбайт данных Sybase IQ в секунду. Поэтому, как правило, дисковую ферму следует рассчитывать так, чтобы каждый ЦП мультиплекса мог получать поток данных 10-20 Мбайт/с. Ширину совокупной полосы пропускания диска в Мбайт/с можно получить, умножив число ЦП в мультиплексе на 10.

Использование диспетчера логических томов (LVM) для Sybase IQ не обязательно и, как правило, не повышает эффективности работы системы (хотя это не означает, что LVM не дает никаких преимуществ при работе иных программ). В многомашинных конфигурациях применение LVM может увеличить стоимость системы из-за необходимости использования специальных кластеризованных версий LVM.

Raw-устройства характеризуются более быстрым вводом-выводом по сравнению с файловыми системами, а также позволяют легко обеспечить совместный доступ нескольких узлов к одним и тем же устройствам хранения, не прибегая к такому сложному решению, как общие файловые системы. Впрочем, в версии IQ 15.1 ESD 3 и более поздних реализован прямой ввод-вывод для систем, которые не могут использовать raw-устройства. Чтобы гарантировать максимальное быстродействие при работе с диском, необходимо правильно настроить файловые системы. Обычно файловая система должна быть настроена на поддержку операций с очень большими блоками данных, не разбиваемых на подблоки меньшего размера.

При выборе дисков для Sybase IQ можно использовать диски высокой емкости (146, 300, 500, 750 и более Гбайт) без опасений относительно скорости доступа. Тестирование Sybase IQ с новыми моделями дисков с малой скоростью вращения (7200 об./мин.) показало минимальное влияние на быстродействие. Система не требует использования дисков с высокой скоростью вращения (10 или 15 тыс. об./мин.). Диски высокой емкости с меньшей скоростью вращения обеспечивают более низкую удельную стоимость хранения. Безусловно, если уже есть диски меньшей емкости, то их вполне можно использовать.

В следующих параграфах приведен базовый алгоритм расчета количества устройств, необходимых для основного и временного хранилищ IQ. Для систем с высокой нагрузкой и большим количеством запросов, а также выполняющих загрузку данных в большие объекты (данные типов BLOB и CLOB) полученные значения следует увеличить — так, чтобы полоса пропускания системы ввода-вывода была достаточна для оптимального решения этих задач.

Размер полосы, ширина полосы и размер блока подсистемы хранения

Цель оптимизации процесса записи на диск, выполняемого IQ, состоит в том, чтобы запись осуществлялась на возможно большее количество дисков одновременно, и чтобы при этом между дисками не возникало конкуренции за ресурсы канала ввода-вывода. Для этого размер полосы, ширина полосы и размеры блоков подсистемы хранения должны быть выбраны соответствующим образом.

Мы будем описывать процесс определения характеристик подсистемы хранения в терминах, используемых администраторами систем хранения данных. Приведем здесь определения этих терминов (их значения, принятые в IQ, отличаются).

Ширина полосы — количество физических дисков, составляющих группу RAID для одного пространства БД IQ. RAID 5 в конфигурации 4+1 будет иметь ширину 4. 8 дисков в конфигурации RAID 0+1 будут также иметь ширину 4 (4 основных и 4 зеркальных).

Размер блока — количество данных, записываемых в группу RAID за одну операцию. Необходимо максимально приблизить это значение к размеру страницы IQ.

Размер полосы — количество данных, записываемых на каждый диск в группе RAID.

При проектировании конфигураций IQ ширина полосы группы RAID, как правило, в расчет не берется, если удовлетворяются минимальные требования к общему числу дисков (шпинделей), необходимых IQ для оптимального быстродействия.

При использовании большинства современных систем хранения рекомендуется, чтобы размер полосы был максимально приближен к размеру страницы IQ, но не превышал его. Это позволяет достигнуть наивысшего быстродействия благодаря тому, что большие массивы данных могут записываться на диск едиными непрерывными блоками.

Если размер полосы может быть увеличен, то на этот случай IQ позволяет за один раз записывать на диск массивы, превышающие размер страницы. По умолчанию IQ записывает за одну операцию вывода одну страницу. Опция DEFAULT_KB_PER_STRIPE (значение по умолчанию 1 Кбайт) гарантирует, что при любом выбранном размере страницы IQ запишет очередную выводимую порцию данных в следующий файл IQ данного пространства БД. Страница при этом не разбивается на блоки или массивы меньшего размера.

В противоположность значимости этого фактора, его значение по умолчанию не слишком мало. IQ выполняет операцию записи так: сжимает одну страницу и записывает ее на диск за одну операцию ввода-вывода. Страница всегда записывается в один файл, и каждая операция ввода-вывода совершается по отношению к одному файлу.

Опция DEFAULT_KB_PER_STRIPE означает, что все последовательные операции записи вплоть до указанного значения применяются по отношению к одному и тому же файлу.

При настройке системы хранения значение этой опции следует выбирать таким, чтобы IQ гарантированно выполняла запись в тот же самый файл перед переходом к следующему.

Можно установить значение этой опции в 256 Кбайт. В этом случае IQ станет записывать страницы в dbfile по достижении объема в 256K (сжатого в блоки); после этого будет осуществлен переход к следующему файлу. В системах, где загрузка меняется в широких пределах, а диски рассчитаны на запись очень большими блоками, можно определить оптимальное значение DEFAULT_KB_PER_STRIPE, увеличивая его шагами, соответствующими размеру страницы IQ. Эта опция динамическая, то есть ее изменения учитываются во всех последующих операциях записи.

При использовании дисковых массивов RAID 5 ширина полосы должна быть на единицу меньше, чем общее число дисков в группе (с учетом диска четности).

Размер блока должен быть равен размеру страницы IQ, заданному при создании базы данных. Некоторые системы могут не поддерживать настолько большие блоки. В этом случае следует выбрать размер блока, равный 64K или более.

Чтобы проиллюстрировать процесс записи, приведем схему. IQ выполняет запись на диск. В ходе этой операции SAN или операционная система могут разбить массив, записываемый одной операцией ввода-вывода IQ, на блоки меньшего размера. Вне зависимости от того, будет ли выполнено такое разбиение, этот блок затем разбивается на подблоки для записи на отдельные диски.

На схеме приведены варианты для двух наиболее часто используемых с IQ типов RAID: RAID 5 и RAID 0+1.

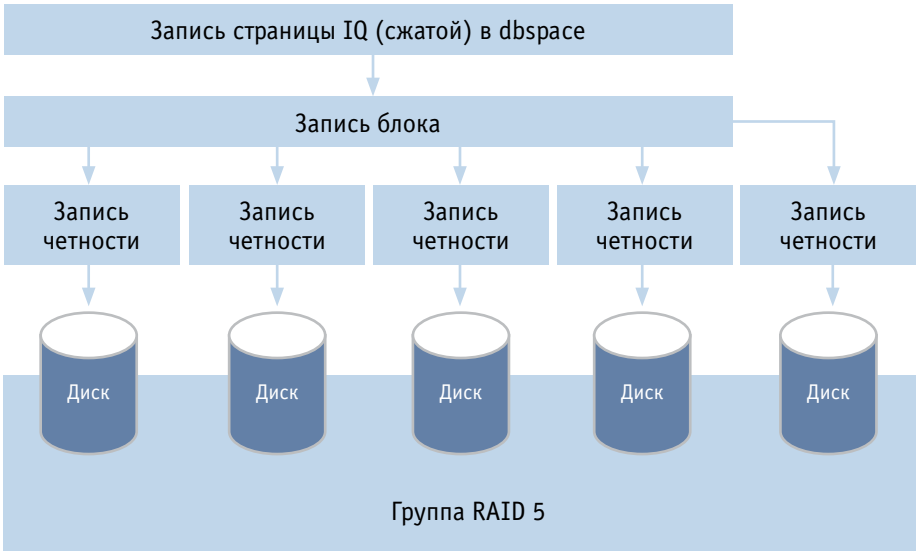

Рис. 1. Пример записи в RAID 5 (4+1)

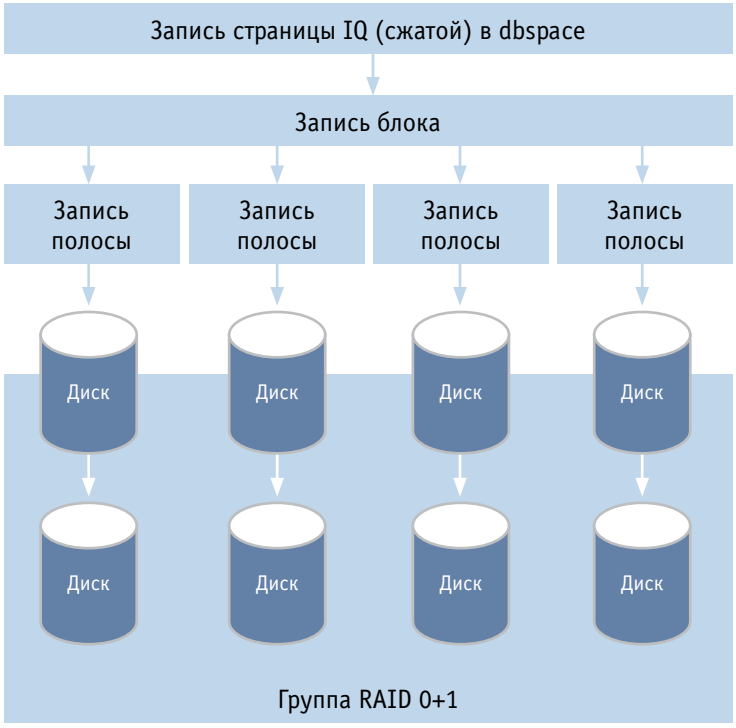

Рис. 2. Пример записи в RAID 0+1 (4 диска)

Физические диски и контроллеры устройств

При использовании дисков Fiber Channel на каждое процессорное ядро должно приходиться минимум 0,3-1 жестких диска, как в одномашинной, так и в многомашинной конфигурации. Если используются диски SAS, количество дисков на ядро следует увеличить до 0,5-3. В случае же применения дисков SATA на каждое ядро следует иметь 2-5 диска. После того, как число дисков на ядро и количество ядер перемножены, следует округлить полученное значение вверх до ближайшего целого.

В системах с высокой нагрузкой указанное количество дисков следует увеличить на 50-100% — это необходимо для обработки обращений к диску, инициируемых дополнительными пользователями, или операций интенсивного ввода-вывода.

Например, 20 процессорных ядер потребуют 7-20 дисков FC, 10-50 дисков SAS или 40-100 дисков SATA. Если же ожидается интенсивный ввод-вывод, то число дисков должно быть таким: 10-40 FC, 15-100 SAS или 60-200 SATA.

Для управления дисками используются дисковые контроллеры (контроллеры FC, шинные адаптеры, или HBA). На каждые 5-10 ЦП должен приходиться один такой контроллер. В высоконагруженных системах число контроллеров должно быть больше.

Быстродействие IQ зависит также от системы хранения, в которую устанавливаются диски и средства сопряжения с серверами. Большое количество дисков и контроллеров окажется бесполезным, если ширина полосы пропускания канала, соединяющего сервер с системой хранения, или внутреннего канала системы хранения недостаточна.

Обычно совокупная полоса пропускания системы хранения вычисляется по такой формуле:

number_of_total_cores * 20 Мбайт/с,

где number_of_total_cores — общее число процессорных ядер во всей системе IQ. В случае одномашинной конфигурации это число ЦП сервера. Для многомашинной конфигурации это сумма количеств ЦП каждого сервера с IQ. Для высоконагруженных систем базовое значение 10-20 Мбайт/с следует увеличить соответственно нагрузке.

При определении количества физических дисков для устройств как основного, так и временного хранилищ используется один и тот же алгоритм. Однако для оптимизации быстродействия и сегрегации нагрузки сайзинг дисков основного и временного хранилищ следует выполнять раздельно.

Устройства для одномашинной конфигурации

В настоящее время принят следующий базовый порядок определения количества устройств основного и временного хранения IQ (dbspaces) для одномашинной конфигурации:

IQ_dbspaces = 3 + (number_of_singlehost_cores / 10).

Полученное значение следует округлить вверх до ближайшего целого. Для устройств основного и временного хранения значения следует рассчитывать отдельно.

Например, 4-ядерная машина потребует минимум 3 + (4/10) = 3,4 ≈ 4 устройства для основного и 4 для временного хранилища.

Устройства для многомашинной конфигурации

При определении количества устройств основного хранения для многомашинных конфигураций следует принять в расчет все ЦП (ядра). Алгоритм расчета следующий:

1. Вычислить общее количество ЦП на всех машинах.
2. Применить к нему формулу:

   IQ_dbspaces = 3 + (number_of_multihost_cores / 10)

Например, если мультиплекс состоит из шести 4-процессорных машин, всего получится 24 ЦП. Тогда необходимое число устройств основного хранилища будет 6 (3 + (24/10)).

При расчете количества устройств временного хранения используется тот же алгоритм, что и для одномашинной конфигурации. Каждый узел должен иметь собственную совокупность устройств временного хранения, размер которой зависит от числа ЦП на данном узле.

IQ_dbspaces = 3 + (number_of_singlehost_cores / 10)

Результат следует округлить вверх до ближайшего целочисленного значения.

Пусть есть четыре 4-процессорных машины. Каждой из них потребуются 4 устройства временного хранения, то есть всего будет 16 устройств временного хранения.

Устройства локального хранения

В настоящее время принят следующий базовый порядок определения числа устройств локального хранения IQ (dbspaces) для одномашинной конфигурации:

IQ_dbspaces = 3 + (number_of_singlehost_cores / 10)

Результат следует округлить вверх до ближайшего целочисленного значения.

Например, 4-ядерная (4-процессорная) машина потребует минимум 3 + (4/10) = 3,4 ≈ 4 устройства локального хранения.

Размещение устройств IQ

Рекомендуется размещать устройства основного и временного хранения IQ так, чтобы они были физически отделены друг от друга, находились на разных логических устройствах (LUN) и физических дисках. Это не обязательное условие, однако его выполнение позволит исключить конфликты, неизбежные при совместном использовании дисков обоими устройствами, и повысить общую пропускную способность. Кроме того, большинство систем хранения имеют механизмы кэширования, которые могут настраиваться для каждого устройства отдельно. При соблюдении указанного условия это позволит настраивать кэши дискового массива по-разному для разных типов устройств IQ.

Дополнительные сведения по настройке устройств хранения, логических устройств (LUN) и сетей хранения данных (SAN) можно получить, введя в поле поиска на сайте http://www.sybase.com строки «IQ Reference Architecture Sizing Guide», «Reference Architecture Implementation Guide» и «NonStopIQ».

Соответствие дисковых устройств

Проектируя подсистему хранения IQ, важно точно определить характеристики всей системы хранения. Не менее важно — для обеспечения быстродействия — распределить нагрузку между файловыми системами, а также устройствами основного и временного хранения. При этом необходимо применять раздельные шинные адаптеры или контроллеры ввода-вывода, а также разные пути в инфраструктуре SAN. Кроме того, физические диски или шпиндели SAN, находящиеся в распоряжении IQ, не должны использоваться какими-либо другими программами.

Приведем в качестве примера высокоуровневую концептуальную диаграмму возможной организации подсистемы хранения IQ. Основное хранилище IQ состоит из четырех разных группировок RAID 5. Каждая группировка является 5-дисковой конфигурацией — 4 + 1 диск четности. Группа RAID в целом представляется каждой машине, подключенной к SAN, одним устройством, так что они воспринимают содержимое четырех дисков (плюс одного диска четности) как единое целое.

«Диск» RAID 5 выглядит для операционной системы логическим устройством (LUN). Последнее, в свою очередь, используется IQ как единое пространство БД (dbspace). При этом нет необходимости использовать диспетчер логических томов (LVM). Если же LVM есть, то задачу представления системе IQ логического устройства как единого диска (без чередования по нескольким дискам) выполняет именно он.

Важно помнить, что в многомашинной конфигурации может возникнуть необходимость в одновременном доступе нескольких серверов к одним и тем же устройствам основного хранения. При увеличении числа узлов критическую роль для быстродействия будут играть добавление портов и увеличение пропускной способности SAN и ее коммутаторов.

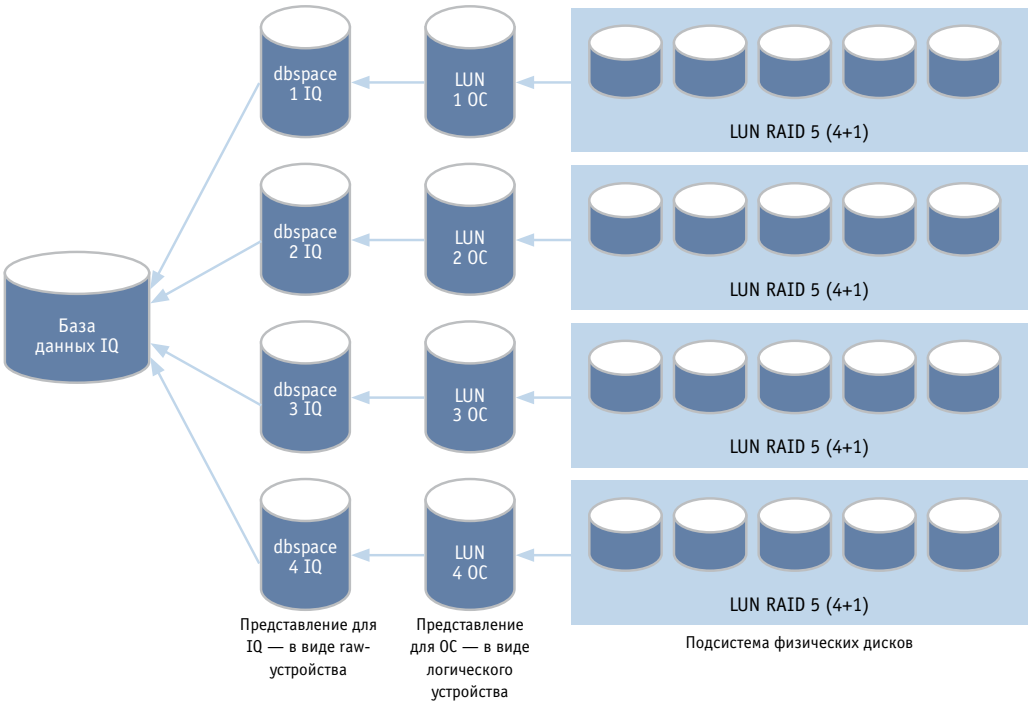

Рис. 3. Соответствие дисков основного хранилища IQ в многомашинной конфигурации

Для временного хранилища IQ дисковая подсистема и представление дисков организованы почти так же. Разница заключается в том, что устройства временного хранения для каждого сервера свои, а устройства основного хранения используются всеми узлами совместно.

Необходимо, однако, обеспечить, чтобы диски (шпиндели), предоставленные устройствам временного хранения каждого сервера IQ, не использовались другими экземплярами IQ или иными программами.

Мы построили пример устройства временного хранения IQ на основе вышеприведенного примера с 4 группировками RAID 5 из 5 дисков (4 + 1 диск четности).

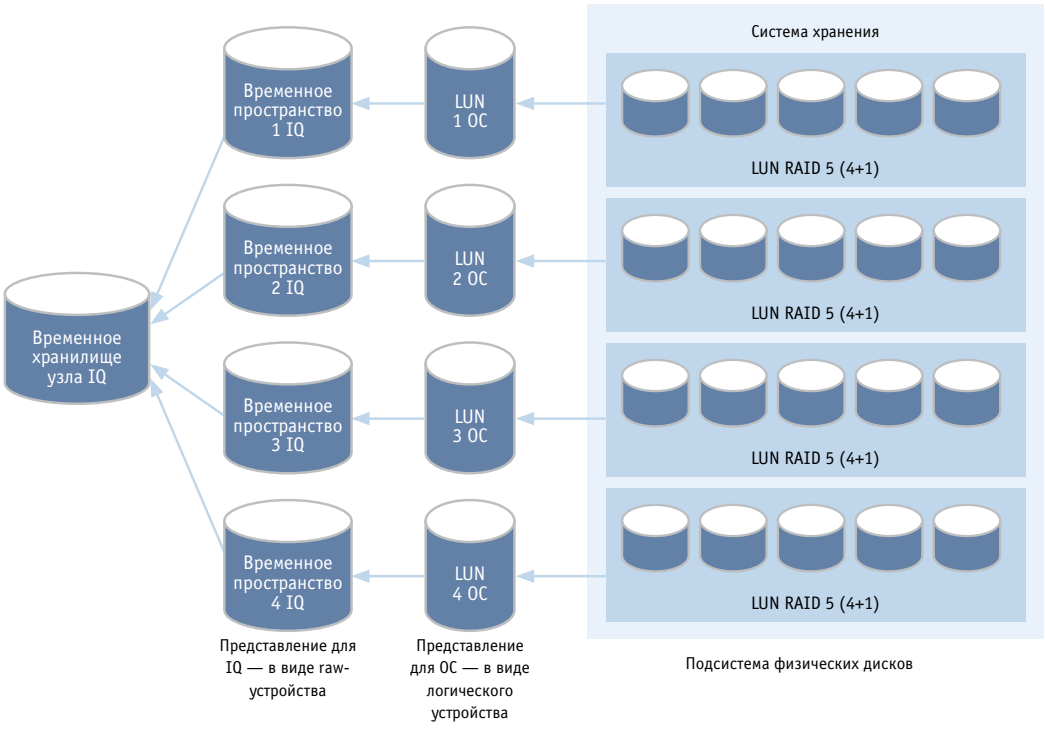

Рис. 4. Соответствие дисков временного хранилища IQ для каждого узла

Определение характеристик сети

Быстродействие

Объем межузлового трафика в мультиплексе IQ настолько незначителен, что не играет существенной роли при конфигурировании сети, связывающей серверы IQ. Впрочем, есть два аспекта работы мультиплекса, для которых топология сети и ее быстродействие имеют значение — это запросы и удаленная загрузка данных (командой INSERT … LOCATION).

При обработке запросов и удаленной загрузке данных скорость получения данных от клиента определяется пропускной способностью сети. Например, передача 100 Мбайт данных займет:

- 80 секунд по 10-мегабитной сети;
- 8 секунд по 100-мегабитной сети;
- 0,8 секунды по гигабитной сети.

При обработке запросов, возвращающих большой объем данных, а также при загрузке больших объемов данных с удаленных серверов сеть может оказаться узким местом. Чем быстрее сетевые контроллеры и ЛВС в целом, тем выше показатели обслуживания в многопользовательском режиме, благодаря большей доступной полосе пропускания для каждой операции.

Для повышения быстродействия и исключения потенциального соперничества за использование ресурсов следует:

- применять быстрые сетевые контроллеры;
- использовать высокоскоростную сеть (1 Гбит вместо 100 Мбит);
- оснастить серверы IQ несколькими сетевыми контроллерами и настроить приложения на использование нескольких IP-интерфейсов;
- увеличить размер пакета клиентского приложения (для соединений ODBC, JDBC и Open Client эти настройки выполняются по-разному);
- увеличить размер пакета удаленной загрузки данных, используя параметр «packetsize» команды INSERT … LOCATION. Следует подчеркнуть, что это документированная возможность IQ, однако в настоящее время максимальный размер пакета ограничен 512 байтами; в будущих выпусках продукта он будет увеличен.

Коммутаторы контента и IQ Multiplex

В настоящее время в Sybase IQ отсутствует встроенный балансировщик загрузки и диспетчер нагрузки. Архитектура Sybase IQ такова, что приложение подключается к одному узлу мультиплекса и вся рабочая нагрузка от пользователя поступает на этот узел. Этот принцип дает приложению и администраторам БД контроль над использованием ресурсов мультиплекса и позволяет, в целях эффективного распределения нагрузки, назначать конкретным пользователям или приложениям определенные серверы. Если же вдруг назначенный сервер оказывается недоступен, то приложение или пользователь могут подключиться к другому действующему узлу Sybase IQ.

Во многих случаях процесс ручной настройки неприемлем, и в то же время создание программы управления нагрузкой может оказаться слишком трудной или вообще непосильной задачей. Нижеприведенная схема иллюстрирует один из возможных вариантов преодоления этой архитектурной проблемы.

Многие заказчики применяют для балансировки нагрузки коммутаторы контента. Изготовители сетевого оборудования, в частности Cisco, F5 и ServerIron, поставляют коммутаторы контента, используемые, как правило, в Web- и FTP-фермах. Такие коммутаторы предоставляют клиентским приложениям один IP-адрес и порт вместо нескольких подключений TCP/IP, передавая затем нагрузку одному из нескольких серверов. Количество, состав и конфигурация этих серверов для приложений значения не имеют и полностью скрыты от них коммутатором.

Такая схема позволяет подключаться к ферме IQ Multiplex любому приложению, не располагающему информацией об отдельных узлах. Она также обеспечивает «бесшовное» переключение нагрузки при простоях, как плановых, так и внеплановых. Когда какой-либо узел отключается, коммутатор контента автоматически соединяет клиента с доступным рабочим узлом. Отключившийся узел помечается как недоступный, и соединения с ним не производятся до тех пор, пока он не будет готов к работе и помечен как доступный.

Для дальнейшего распределения рабочей нагрузки можно использовать также несколько портов. Например, можно добавить один порт для пользователей, создающих большую нагрузку. Поскольку все перенаправления в коммутаторе контента выполняются на уровне портов, этот порт можно зарезервировать для определенных клиентских машин.

Короче говоря, данная архитектура не уменьшает возможности распределения нагрузки, а наоборот, увеличивает их. Как и раньше, можно разделять нагрузку приложений и пользователей, но помимо этого, в инфраструктуре реализована возможность бесшовного аварийного переключения нагрузки.

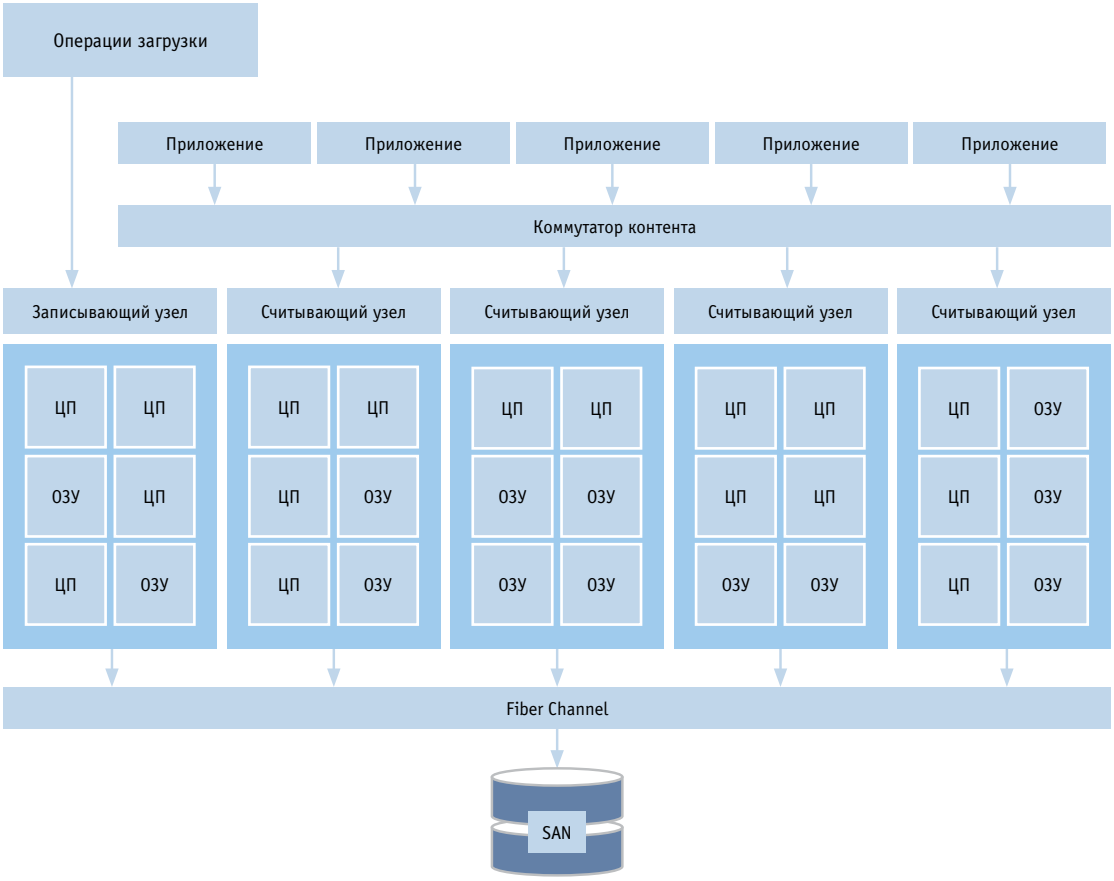

Рис. 5. IQ Multiplex с коммутатором контента

Размеры страниц IQ

Sybase IQ позволяет при создании базы данных задать размер страницы, используемый для дисковых устройств основного и временного хранилищ IQ. В настоящее время размер страницы по умолчанию 128 Кбайт. Пользователю доступны также размеры 64, 256 и 512 Кбайт.

Использование 64-килобайтных страниц оправданно сегодня в крайне редких случаях. Как правило, этот размер применим в системах с очень малым объемом ОЗУ (менее 4-8 Гбайт), в 32-разрядных системах, в системах с ограниченной дисковой и оперативной памятью, обслуживающих большое число пользователей одновременно, а также в очень малых базах данных.

Пересылая данные из памяти на диск, IQ сжимает страницы, превращая их в блоки. Страница (содержащая пользовательские данные, индексы, битовые массивы и т. п.), компрессируется по специальному алгоритму. Размер страницы делится на 16 без остатка (настройка по умолчанию — 16 блоков на страницу) и кратен размеру полученного в результате блока данных.

Результатом операции сжатия всегда является непрерывный блок данных, подлежащий записи на диск. Размер блока зависит от того, насколько хорошо данные страницы поддаются сжатию. При этом он всегда будет кратен размеру блока IQ (по умолчанию 1/16 размера страницы).

Хотя при создании базы данных можно задать размер блока IQ, как правило, в этом нет необходимости. В большинстве случаев размер блока по умолчанию (1/16 установленного размера страницы IQ) вполне подходит.

Размер страницы IQ действителен как для основных, так и для временных устройств хранения. Об этом важно помнить, поскольку, например, система с большим числом одновременно работающих пользователей может размещать значительную часть временных и рабочих таблиц во временном хранилище. Как правило, временные таблицы содержат незначительную долю данных исходных таблиц, лишь в очень немногих случаях это могут быть десятки тысяч или хотя бы тысячи записей. Столь малое число записей может потребовать больших накладных расходов по причине минимальных требований к хранению данных на странице или нескольких страницах.

Для каждого столбца и каждого индекса этих временных объектов будет выделено по одной странице. Если объект содержит всего 100-1000 записей, большая часть страниц может оказаться пустой, но при этом будет занята. Это может существенно увеличить требования к объему дисковой памяти для временного хранилища IQ и временного кэша для конкретного набора данных по сравнению с хранимыми данными.

Возьмем для примера таблицу с 10 столбцами без индексов. Для нее потребуется минимум 10 страниц (по одной на столбец). В зависимости от типов данных и степени индексной оптимизации страница может уместить вплоть до нескольких десятков тысяч записей. При создании таблицы на несколько сотен записей потребуется тот же объем, что и для нескольких тысяч записей той же структуры.

При выборе размера страницы следует учесть несколько факторов. Одни из них определяют размер страницы, другие являются его следствием. Это:

- число одновременно работающих пользователей;
- число записей в таблице;
- требования к ресурсам памяти (оперативной памяти, емкости временного хранилища, области управления версиями).

Число одновременно работающих пользователей

Прямая связь между числом одновременно работающих пользователей и размером страницы отсутствует — нельзя сказать, что, например, в системе с числом пользователей, меньшим X, размер страницы должен быть таким-то. Однако совокупное влияние числа одновременно обслуживаемых пользователей и размера страницы на работу системы может быть очень заметным. Важно, чтобы эти два показателя, а также объем виртуального хранилища (кэшей и пространств БД) для временного хранилища IQ были сбалансированы.

По мере увеличения размера страницы соответственно увеличиваются необходимые для обработки страниц объемы ОЗУ и дисковой памяти. При увеличении числа одновременно работающих пользователей также потребляется больше оперативной и дисковой памяти.

Число записей в таблице

Хотя в руководствах фактором, определяющим размер страницы, считается предельный размер базы данных, в текущей практике этот расчет используют в последнюю очередь, когда другие возможности определить оптимальный размер страницы отсутствуют. Самый подходящий фактор для определения размера страницы, основанный на характеристиках БД — ожидаемое максимальное число записей в самой большой таблице.

В главе 5 «Choosing an IQ Page Size» («Выбор размера страницы IQ») Руководства по системному администрированию Sybase IQ (Sybase IQ System Administration Guide) приведены следующие рекомендации:

- 64 Кбайт — для баз данных с максимальным размером таблицы до миллиарда записей. Это абсолютный минимум для новой базы данных. На 32-разрядных платформах 64-килобайтный размер страницы обеспечивает наивысшее быстродействие.

- 128 Кбайт — для баз данных на 64-разрядных платформах с максимальным размером таблицы от 1 млрд. до 4 млрд. записей. 128 Кбайт — размер страницы IQ по умолчанию. Как правило, он подходит для большинства систем.
- 256 Кбайт — для баз данных на 64-разрядных платформах с максимальным размером таблицы более 4 млрд. записей.
- 512 Кбайт — для баз данных на 64-разрядных платформах с максимальным размером таблицы более 10 млрд. записей.

Анализ недавних внедрений Sybase IQ показал следующее:

- 64 Кбайт — используется только под управлением 32-разрядной ОС Windows.
- 128 Кбайт (размер по умолчанию) — в большинстве систем используется этот размер страницы. Размер таблиц, как правило, составляет менее 2-3 млрд. записей.
- 256 Кбайт — системы с этим размером страницы имеют таблицы с 4-8 млрд. записей.
- 512 Кбайт — этот размер страницы используется в системах с таблицами, насчитывающими десятки миллиардов записей.

Однако число записей не должно оставаться единственным фактором определения размера страницы IQ. Обычно придерживаются того правила, что при увеличении объема ОЗУ можно увеличить и размер страницы.

Влияние на ресурсы памяти

Увеличение размера страницы IQ сопровождается рядом следствий, которые необходимо принять во внимание — это изменение потребления дисковой памяти, оперативной памяти и необходимого размера области управления версиями.

Потребление дисковой памяти объектами, находящимися в основном хранилище, обычно не представляет проблемы по той причине, что размер их достаточно велик, чтобы заполнять страницы целиком. Однако пространство, потребляемое маленькими поисковыми и справочными таблицами (как правило, длиной в несколько тысяч строк и менее), при увеличении размера страницы может существенно возрасти. В этом случае страница малого размера может быть заполнена данными на 50%, а большого — на 33 или даже на 25%, при одном и том же объеме данных.

Хотя сжатие данных выполняется до того, как они записываются на диск, влияние размера страницы на потребляемую дисковую память может проявиться и здесь. Данные на диске хранятся в блоках. Размер блока по умолчанию составляет 1/16 размера страницы. По мере увеличения размера страницы растет и размер блока. Так, страница данных объемом 128 Кбайт может быть сжата в 8-килобайтный блок. 256-килобайтная страница с теми же данными займет на диске уже 16 Кбайт.

При изменении размера страницы объем потребляемой оперативной памяти не меняется, поскольку основной и временный кэш остаются теми же. Однако если объем оперативной памяти, доступный для кэшей, не увеличить, быстродействие может снизиться.

В системе с небольшим количеством пользователей количество проблем, обусловленных нехваткой памяти, с очень высокой вероятностью будет меньшим, чем в системах с большим числом одновременно работающих пользователей.

При увеличении числа одновременно работающих пользователей растет число страниц, находящихся как в основном, так и во временном кэше. При переходе к большему размеру страницы (например, 256 Кбайт вместо 128) число страниц, умещающихся в одном и том же объеме памяти, уменьшается вдвое.

Если пользователю необходима лишь малая доля данных на странице, то для обработки запроса потребуется обработать большее число страниц. Если не увеличить объем ОЗУ, выделенный для кэшей, то система может оказаться вынуждена перед загрузкой новых данных выгружать наиболее редко используемые страницы.

Таким образом, для надлежащего обслуживания того же самого количества пользователей при удвоенном размере страницы может оказаться необходимым удвоить размер кэшей IQ.

Если размер страницы определяется исключительно на основании числа записей в таблице, может возникнуть нехватка оперативной памяти и обращения к диску участятся. Как правило, системы с объектами большего объема имеют больше доступной памяти и потому увеличение размера страницы не сказывается отрицательно на их работе. Если таблицы могут вырасти до такого предела, при котором обычно увеличивают размер страницы, но ОЗУ увеличить невозможно, необходимо тщательно взвесить все «за» и «против» — возможно, от увеличения размера страницы придется отказаться.

Что касается версионирования, то этим аспектом при определении размера страницы часто пренебрегают. Число страниц, необходимых для записи версий, не меняется; при этом увеличивается объем кэша или дисковой памяти. Если размер страницы удвоить, то объем кэша или дисковой памяти, необходимый для хранения информации о версиях, также удвоится.

Потоки

Выделение потоков при запуске

Число потоков по умолчанию, выделяемых в IQ при запуске, зависит только от двух факторов: числа процессорных ядер и количества пользовательских подключений (параметр запуска –gm). Общее число потоков, выделяемых при запуске, может быть изменено с помощью стартовой опции –iqmt.

По умолчанию параметру –iqmt присваивается следующее значение:

60 * (min(numCores,4)) + 50 * (numCores – 4) + 2 * (numConnections + 2) + 1

На 4-ядерной системе при значении параметра –gm (numConnections) равном 20 получится:

iqmt = 60 * (4) + 50 * (4–4) + 2 * (20+2) + 1
iqmt = 285

На 12-ядерной системе при значении параметра –gm (numConnections) равном 50 получится:

iqmt = 60 * (4) + 50 * (12–4) + 2 * (50+2) + 1
iqmt = 745

Существуют два разных типа потоков: потоки подключений и серверные потоки. При этом они не составляют единого большого пула общего назначения.

Число потоков подключений определяется стандартным расчетом на основе числа подключений, на которое настроен сервер: 2 * (numConnections + 2). Они зарезервированы для подключений и не могут использоваться в качестве рабочих потоков для обработки запросов или загрузки данных.

Проверить количество потоков можно с помощью IQ Buffer Cache Monitor, указав опцию –threads или –debug. Если параметр «Free Threads» равен параметру «Reserved Threads», то остающееся количество — это потоки, предназначенные для подключений, при этом потоков для параллельной обработки не остается.

Число серверных потоков определяется по стандартному расчету на основе числа процессорных ядер: 60*(min(numCores,4)) + 50*(numCores – 4). Эти потоки используются для поддержки параллельной загрузки и параллельной обработки запросов наряду с теми, что задействованы для дискового ввода-вывода. Для параллельной обработки необходимы свободные серверные потоки. Если параметру –iqmt присваивается значение, оно обязательно должно быть больше числа потоков по умолчанию. Если это значение меньше числа потоков по умолчанию, IQ проигнорирует его и запустит сервер с необходимым минимумом потоков, достаточных ТОЛЬКО для обслуживания подключений. Таким образом, чтобы в системе был пул потоков для обработки запросов, загрузки данных, а также подключений, параметр –iqmt должен быть установлен согласно вышеприведенному указанию.

Для 64-разрядных систем верхний предел числа потоков — 4096. Общее число потоков, задаваемое параметром –iqmt, а также стартовым параметром –gn, не должно превышать этого предельного значения. Значение –gn по умолчанию — –gm + 5, где –gn — число потоков, используемое механизмом SQL Anywhere.

Потоки дискового ввода-вывода

Потоки, реализующие дисковый ввод-вывод, делятся на две группы: записи изменений (sweepers) и предварительной выборки (prefetchers). Первые выполняют запись измененных буферов, а вторые считывают информацию в буферы.

Число потоков каждой группы определяется следующими опциями:

SWEEPER_THREADS_PERCENT

PREFETCH_THREADS_PERCENT

Обе опции выражают долю в процентах общего числа потоков, выделенных IQ, не считая потоков SQLAnywhere. Значение по умолчанию в обоих случаях — 10, так что обеим группам выделяется по 10% общего числа потоков IQ.

Как правило, запись и считывание данных выполняются двумя названными группами потоков. Однако если потоки записи изменений не успевают обработать область, подлежащую записи, то диспетчер буферов подаст команду их выгрузки, и запись выполнит приемный поток «inline», после чего буфер можно будет использовать вновь. Точно так же, если потоки предварительной выборки не могут выполнить выборку своевременно, то запрашивающий поток. не обнаружив в памяти требуемого блока, считывает данные сам. Однако в идеальном случае дисковый ввод-вывод должны выполнять только потоки записи изменений и предварительной выборки.

Потоки записи изменений проходят последнюю использовавшуюся цепочку буфер за буфером, постоянно осуществляя поиск буферов, подлежащих записи. Дисковые и потоковые очереди как таковые отсутствуют: потоки только контролируют wash-область и записывают все обнаруженные данные.

Группа предварительной выборки получает информацию о считываемых буферах от оптимизатора. Если оптимизатор не нуждается в данных, то ее потоки будет ожидать задания. Они могут выполнять и другие операции чтения, а не только предварительной выборки. Этим управляет система; пользователь контролировать этот процесс не может.

Получение сведений о количестве потоков

Вывод опции –debug IQ Buffer Cache Monitor имеет раздел, содержащий подробную информацию о потоках. Ту же информацию можно получить с помощью процедуры sp_iqsysmon.

Пример вывода:

```
ThreadLimit= 1499 ( 100,0%)
ThrNumThreads= 1499 ( 100,0%)
ThrReserved= 566 ( 37,8%)
ThrNumFree= 787 ( 52,5%)
NumThrUsed= 712 ( 47,5%)
```

Прокомментируем этот пример.

Во-первых, значение ThreadLimit всегда совпадает с ThrNumThreads и равно значению iqmt (вычисленное общее число потоков по умолчанию), уменьшенному на 1. Один поток IQ резервирует для аварийного подключения. Таким образом, в нашем случае iqmt=1500.

Во-вторых, (ThrNumFree + NumThrUsed) всегда равно ThreadLimit. В любой момент времени сумма числа свободных и используемых потоков равна общему количеству доступных потоков. Для NumThrUsed счетчик отсутствует; это значение вычисляется как (ThreadLimit – ThrNumFree).

Число зарезервированных потоков отражено в показателе ThrReserved. Как правило, этот показатель представляет число потоков в пуле потоков подключений. Он не связан непосредственно с другими; его значение может слегка изменяться, при этом прочие счетчики не будут увеличиваться либо уменьшаться пропорционально. Изменение значения ThrReserved происходит из-за потоков, резервируемых для операций ввода-вывода.

Количество свободных потоков отражает показатель ThrNumFree. Эти потоки сервер может использовать в любых целях — как для подключений, так и в качестве серверных потоков. Если значения ThrNumFree и ThrReserved совпадают, значит, все доступные рабочие потоки используются, и доступны лишь потоки, зарезервированные для подключений. Если возникает или обещает возникнуть такая ситуация (когда значения этих двух параметров приближаются друг к другу), рекомендуется увеличить число рабочих потоков с помощью опции –iqmt.

Sybase, Inc.
Worldwide Headquarters
One Sybase Drive
Dublin, CA 94568-7902
U.S.A
1 800 8 sybase

www.sybase.com

Sybase CIS
115114, Москва,
Дербеневская набережная,
д. 7, стр. 16
+7 (495) 797-4774

www.sybase.ru

SYBASE®
An SAP Company