

Распространенное заблуждение состоит в том, что под автоматизацией процессов, предусмотренных бизнес-моделью предприятия, понимают совершенствование оперативной деятельности. С позиций управления бизнесом намного важнее провести содержательный анализ ретроспективной информации, находящейся в распоряжении ведущих менеджеров. Технологически процесс получения, обработки и применения на практике объектно-ориентированных массивов данных является пошаговой задачей. Модель постепенного развития корпоративных систем управления диктует выбор аппаратных средств.

Анализ консолидированных данных: тестируем платформу AMD Opteron

Игорь Нишпенко, Сергей Толокунский

Автоматизированные системы оперативной деятельности предприятия чаще всего основываются на обработке транзакций в режиме реального времени (On-Line Transaction Processing – OLTP). Такие системы ориентированы прежде всего на эффективное выполнение ограниченного набора операций, например формирование счета на оплату, просмотр накладных к нему и т. д. Какой бы ни была практическая реализация системы – менеджер записей, реляционная база данных, – такое решение оптимально для задач, которые ежедневно в огромном количестве выполняет программный комплекс коммерческого предприятия. С его помощью можно с успехом готовить стандартную оперативную отчетность по заранее созданным формам.

Требования к аппаратной части при автоматизации оперативной деятельности достаточно прогнозируемы еще на этапе обследования. Вряд ли, например, торговая компания может изменить за небольшой срок требования к количеству одновременно работающих пользователей с одного десятка до 3–4, а тем более до сотни. И вряд ли так же радикально изменится функциональный состав проекта. Можно смело утверждать, что подобного рода изменения в проекте автоматизации оперативной деятельности – это просчет на этапе его постановки.

Иное дело – хранилища данных. Проекты в этой области исторически возникли позже, и здесь чаще используется инкрементный подход. Сделали

что-то небольшое, посмотрели и решили, стоит ли развивать проект дальше. Но небольшое, казалось бы, изменение в проекте с хранилищем данных может привести к скачкообразному увеличению объемов. Например, потребуется более высокая детализация данных, или охват хранилища расширится от исключительно финансовой сферы до общей, включающей сбыт и закупки.

Таким образом, в проектах по хранилищам данных успех разработчиков зачастую приводит к росту объема самого хранилища. То есть действует закономерность, прямо противоположная таковой в оперативных системах. Успешная деятельность разработчиков в данном случае может привести к значительному увеличению требований к серверу. При этом они возникают далеко не сразу, а только после нескольких успешных этапов работы. И предсказать заранее, на ка-

Под термином «хранилище данных» мы понимаем предметно-ориентированный, интегрированный, зависимый от времени набор данных, предназначенный для поддержки принятия решений различными группами пользователей. Так как хранилище носит предметно-ориентированный характер, его организация нацелена на содержательный анализ информации, а не на автоматизацию бизнес-процессов. Это свойство определяет архитектуру построения хранилища и принципы проектирования модели данных, отличные от тех, что применяются в оперативных системах. *Источник – www.olap.ru*

ком этапе проект достигнет своего «потолка», достаточно трудно. Поэтому наличие у коммерческой организации, внедряющей у себя хранилище данных, возможности поэтапно наращивать производительность своих серверов особенно желательно.

AMD Opteron

Процессоры AMD Opteron и системы на их основе – давно уже не новички на рынке серверов. Характеристики и особенности архитектуры AMD64 изучены и описаны, проведены и опубликованы сравнительные тестирования в различных приложениях.

Эволюционные особенности AMD64 представляют большой интерес. Аппаратная совместимость процессоров AMD Opteron с существующими 32-разрядными приложениями и возможность легкого перехода на 64-разрядные позволяют говорить о более продолжительном по сравнению с традиционными архитектурами жизненном цикле платформы, а следовательно, и о снижении совокупной стоимости владения.

Предметом изучения этого материала стали вопросы практического использования серверов на основе процессоров AMD Opteron в области корпоративных систем управления. Первое, что приходит в голову при разговоре о таких серверах, – базы данных, поддерживающие одновременную работу большого количества

пользователей. Широкое распространение получили также системы с тонким клиентом, которым необходима значительная вычислительная мощность центральных компьютеров. Высокие требования к гибкости таких систем — как основе конкурентоспособности и выживаемости предприятия в сложном современном мире — обусловили развитие самых разнообразных сервисов, от поддержки архитектуры терминального доступа с графическим интерфейсом до Интернета, XML и многих других.

Самостоятельное исследование производительности серверных платформ даже в ограниченной области баз данных — задача ресурсоемкая. К тому же существует консорциум **TPC.ORG**, прерогатива которого — проведение всестороннего независимого тестирования серверных систем. Мы же сконцентрировали усилия в узкой области, попытавшись изучить производительность не СУБД, а типичного сервера определенной конфигурации на основе процессоров AMD Opteron применительно к хранилищам данных в 32- и 64-разрядном окружении. Исследования проведены на основе СУБД Oracle 9i.

Почему Oracle?

Может быть, это один из первых вопросов, который возникнет у читателей. Не секрет, что при построении хранилищ достаточно эффективно используется продукт корпорации Microsoft — MS SQL Server. Начиная с версии 7 Microsoft предлагает достаточно мощный продукт, который считается к тому же недорогим и простым в освоении для решений своего класса. Мы надеемся в дальнейшем испытать и его на платформе AMD Opteron.

Но на данный момент коммерческие версии СУБД в обеих архитектурах — x86 и AMD64 — у Microsoft в отличие от Oracle отсутствуют. Немаловажно и то, что СУБД Oracle обладает богатой функциональностью для построения кластерных систем — их основу могут составлять серверы, подобные тому, на котором проводились испытания. С использованием СУБД Oracle возможно построение нескольких типов кластерных моделей, в том числе отказоустойчивых и систем с балансировкой нагрузки. При этом база не подвергается каким-либо модификациям, не требуется какая-либо привязка к узлам или к их количеству. То есть одна и та же база данных без мо-

дификаций может эксплуатироваться и на одиночном сервере, и на кластере из двух узлов, и на кластере из 32 узлов. Подобного решения у MS SQL пока нет.

Стало быть, если речь идет о масштабируемых системах, обслуживающих запросы крупных производственных предприятий, банков, биллинговых компаний, предусматривающих высокую степень отказоустойчивости, то применение СУБД Oracle выглядит вполне уместным.

Описание подхода к испытаниям

В этом материале сделана попытка провести сравнение производительности систем на процессорах AMD Opteron при работе с большими базами данных в 32- и 64-разрядном окружении. В качестве программной базы для исследования использована связка SuSE Linux Enterprise Server 8.0 и Oracle 9i — эти приложения имеют коммерческий статус и доступны сегодня в обеих версиях, 32- и 64-битной.

Подчеркнем — задача состояла не в сравнении производительности различных СУБД. нас интересовала прежде всего производительность 64-битных и 32-битных приложений на одной и той же аппаратной платформе. Замеры должны были продемонстрировать преимущества или отсутствие таковых при переходе с 32-разрядных платформ на 64-разрядные применительно к серверам корпоративных информационных систем.

Что мы испытывали

Готовя постановку тестов, мы пошли на сознательное упрощение, ограничившись только выбором основных и/или наиболее ресурсоемких операций.

При проведении испытаний интересной представлялась возможность упрощения структуры баз данных за счет большей мощности сервера (имеются в виду продвинутые функции Oracle — секционированные таблицы и прочие). Дело в том, что многие тонко настроенные структуры очень хорошо ведут себя в одних условиях, но очень плохо — в других. Поэтому возможность работать на более простых структурах не только облегчает проект и уменьшает его стоимость, но и делает его более устойчивым к изменениям условий.

Для обоих релизов использовалась одна и та же база данных (что естественно), абсолютно одинаковой была и аппаратная часть. Различия касались только программного обеспечения — в частности, конфигураций Oracle.

Различия в конфигурациях были вызваны только наличием «избыточной» памяти. На сервере физически было установлено 6 GB ОЗУ. Поскольку нас не интересовали ни архитектурные изыски Oracle, ни различия в функциональности релизов, а только сравнение возможностей 32-битного и 64-битного ПО с точки зрения утилизации функций конкрет-



**НОВОРІЧНА
ПОРАДА
ВІД
ДІДА
МОРОЗА**

**Комп'ютери ІРБІС
найкращий подарунок до
Нового Року!**



м.Київ, вул.Струтинського, 8, 7 пов
тел. (044) 537-24-05, 537-24-07
info@compirbis.kiev.ua
www.compirbis.kiev.ua

ного сервера, то мы конфигурировали Oracle в соответствии с особенностями каждой архитектуры. То есть в 64-битном режиме мы смогли увеличить разрешенное пространство памяти для Oracle дополнительно на 2 GB.

Отдельно хотелось бы сказать, что существуют реальные проблемы утилизации возможностей архитектур с расширенной адресацией. Имеется в виду не только 64-битная архитектура, но даже и 32-битная в сравнении с 16-битной. Так, например, известно, что при переносе OS/2 на 32-битную архитектуру разработчики Microsoft SQL Server имели очень серьезные проблемы с быстродействием (см. Kalen Delaney, Inside Microsoft SQL Server 2000). Еще более показательны факты, предоставленные специалистами компании «Ай-Текко» (см. www.i-teco.ru/article34.html) – 2-процессорные серверы HP ProLiant на Intel Xeon продемонстрировали одинаковые результаты и на 32-, и на 64-разрядной платформе (неиспользовались гестовые версии Windows, NET Server и SQL Server 2000 64-bit edition beta).

Безусловно, репутация компании Oracle – многолетнего поставщика коммерческих релизов 64-битных РСУБД – давала повод надеяться на положительные результаты, но могло получиться и так, что именно при объеме памяти в 6 GB накладные расходы расширенной архитектуры превысили бы преимущества дополнительной памяти, и результат оказался бы нулевым или даже отрицательным.

Возникает вопрос – а почему именно 6 GB? Дело в том, что 6 GB – достаточно распространенная конфигурация. Конечно, тестируемый сервер может содержать в себе и 16 GB ОЗУ, но разрыв между этими конфигурациями в обоих режимах мог оказаться настолько велик, что какие-либо результаты сопоставления были бы уже предопределенными. А мы хотели выяснить, какие перспективы сулит возможность постепенного наращивания мощности подсистемы памяти, открываемая архитектурой AMD Opteron.

Как шли испытания

Испытания проводились в пользовательской среде. Все запросы к серверу были динамическими и выполнялись последовательно (следующий по завершении предыдущего) только с удаленного компьютера.

Запросы строились так, чтобы минимально задействовать ресурсы процессоров.

Для сравнения были отобраны следующие операции:

- имитация начального наполнения хранилища путем создания таблиц фактов. Источник данных – реляционная СУБД;
- имитация пополнения хранилища путем добавления в таблицы фактов. Источник данных – реляционная СУБД;
- выборки из хранилища, находящегося в «сыром» состоянии (при отсутствии каких-либо индексов);
- индексация хранилища;
- выборка из хранилища в индексированном состоянии.

Результаты испытаний приведены в таблицах.

Одни и тот же запрос в каждом замеры производился несколько раз. Результат первого просто игнорировался, поскольку на его характеристиках могли сказаться любые случайности. То есть первый запрос служил только для заполнения кэшей – Oracle и Linux. Затем выполнялись подряд еще два запроса. Если время ответа на них оказывалось близким, вычислялось усредненное значение по результатам двух последних. Если же возникали большие отличия – производился четвертый запрос (третий значимый). По нему уже принималось решение – либо разбираться с оптимизатором (смотреть план запроса), либо попытаться перезагрузить Oracle и начать замер сначала, т. е. исключить возможные проблемы со стороны операционной системы.

Комментарии

Хотелось бы сразу сказать, что приведенная таблица результатов дает информации больше, чем диаграммы (полная таблица будет размещена на сайте itc.ua). В ней представлены некоторые неожиданные результаты, не вошедшие в диаграммы. Так, например, при выборке из большой таблицы с битовым индексом 64-битный релиз показал непомерно большое время запроса. Как оказалось, этот релиз считал более эффективным способом просканировать индекс, а не таблицу, и ошибся. Что ж, это подтверждает, что разные релизы могут принимать различные решения в сходных ситуациях. Правда, на принятие решения могли повлиять и разные объемы кэшей, которые им были доступны.

Конфигурации и параметры

Описание аппаратной части

Процессоры: 2×AMD Opteron 244 (1,8 GHz)

Системная плата: TYAN Thunder K8S Pro (S2882)

ОЗУ: 6 GB DDR PC3200

Дисковая подсистема:

Western Digital WD400JB (40 GB), reiserfs, системный

Western Digital WD1200JB (120 GB), ext3, для данных

Примечание. При построении коммерческой системы, работающей под управлением Oracle, использование IDE-дисков, конечно же, неприемлемо. Оперативные системы обычно базируются на RAID-массивах из производительных SCSI-дисков, хранилища могут строиться и на массивах из дисков SCSI или SATA, возможно, вынесенных во внешне выделенное устройство хранения. Медленная дисковая подсистема была выбрана для замеров специально, чтобы ярче проявились потенциальные преимущества 64-разрядной среды при работе с памятью.

Программная часть 32-битной архитектуры состояла из операционной системы SuSE Linux Enterprise Server 8 + SP3 и СУБД Oracle 9i (9.2.0.4), а 64-битной – SuSE Linux Enterprise Server 8 + SP3 и СУБД Oracle 9i (9.2.0.4) (для AMD64)

В конфигурации Oracle после его установки и создания базы менялось только два параметра:

- `pga_aggregate_target`;
- `db_cache_size`.

Первый из них определял максимальное количество оперативной памяти, которая могла использоваться Oracle для сортировок, второй параметр – максимальное количество данных для кэширования.

Для 32-битного релиза:

- `pga_aggregate_target` = 2002 MB;
- `db_cache_size` = 1440 MB.

Для 64-битного релиза:

- `pga_aggregate_target` = 3072 MB;
- `db_cache_size` = 2384 MB.

Таким образом, для 32-битного релиза разрешалось использовать более 3442 MB ОЗУ, для 64-битного релиза – более 5456 MB. Слово «более» здесь употреблено потому, что существуют другие кэши, на которые расходуется память, но их значения были одинаковыми для обоих релизов – те, которые устанавливаются по умолчанию при инсталляции.

Диаграммы и пояснения



Диаграмма № 1. Производительность импорта данных

Такие действия производятся при импорте данных в базу.

Этот замер чрезвычайно важен. Он позволяет сравнить производительность подсистем ввода/вывода в 32- и 64-разрядном окружении. Как свидетельствуют полученные данные, различия незначительны. Таким образом, можно не принимать во внимание скорость работы дисковой системы.

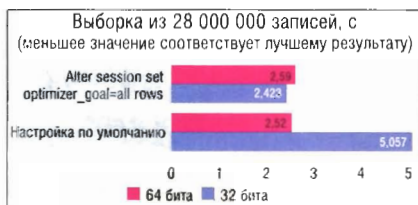


Диаграмма № 2. Производительность выборки данных из неиндексированных таблиц (28 миллионов записей)

Второй шаг при создании хранилища – проверка успешности ввода всех данных. Существует много разных методов, самый примитивный из которых – пересчет количества импортированных записей. Именно такая операция и была проделана.

На практике этот способ используется очень часто. Кроме того, он довольно прост и показателен. Условность замера состоит в том, что процессоры почти не загружены – нет каких-либо сложных расчетов или объемных суммирований.

В этом замере были выявлены различия в методах работы с данными разных релизов. Оказалось, что 32- и 64-битный релиз Oracle используют различные стратегии, причем в первом случае предполагается значительно большее количество физических обращений к диску. Для того чтобы нормализовать данные, пришлось вручную указать параметры оптимизации. После этого 32-битный релиз оказался чуть быстрее.

Нужно обратить внимание на то, что в этом замере файловая подсистема уже не играла никакой роли. Таблица полностью помещалась в кэш, а первые замеры, напомним, отбрасывались. Таким образом, при обработке таблицы, все записи которой помещались в кэш, младшая архитектура выиграла, но незначительно. Это обстоятельство подтверждает, что при работе с

малыми объемами данных накладные расходы 64-разрядной архитектуры могут нивелировать предполагаемый выигрыш в производительности.

Результаты замеров при неустановленном параметре оптимизации иллюстрируют, что при работе с большими таблицами мелкие различия в стратегии оптимизатора могут играть весьма значительную роль.



Диаграмма № 3. Производительность выборки данных из неиндексированных таблиц

В этом замере таблица уже не помещалась в кэш Oracle ни в одной из платформ. Но кэш данных у 64-разрядной архитектуры был больше – 2384 МВ против 1440 МВ. Как и ожидалось, запросы на 32 бита оказались медленнее – от 15 до 20% в зависимости от их вида. Причем разница была более существенной на примитивном запросе.

В данном замере был получен результат, подтверждающий ожидания, – 64-битная архитектура дает возможность быстрее обрабатывать большие таблицы.



Диаграмма № 4. Производительность индексации

Этот замер – ключевой. После индексации, особенно множественной, многие запросы существенно ускоряются. Но вот сама индексация происходит очень и очень долго. Зачастую именно она является основным ресурсоемким этапом выведения хранилища в рабочую фазу.

Как видно из диаграммы, в обоих случаях 64-битная архитектура вышла победителем. Причем на малой таблице выигрыш оказался значительнее, чем на большой. Это позволяет предположить, что с увеличением объема памяти скорость индексации большой таблицы будет расти быстрее, чем производительность выборки из неиндексированной большой таблицы.

В целом обращает внимание очень небольшое время, потраченное на создание индексов. Например, для меньшей таб-

лицы время создания индекса превосходит время выборки из неиндексированной таблицы всего-то в 5–6 раз, но для большой таблицы результат становится поразительным – индексация лишь на 20% продолжительнее выборки! Косвенно это указывает на очень мощную подсистему управления памяти архитектуры AMD Opteron.



Диаграмма № 5. Производительность выборки по индексам

Заключительный этап – сравнение производительности выборок на таблицах в рабочем режиме.

Результат замера «Выборка по индексированной таблице» оказался неожиданным. Так, на малой таблице в 32-битном окружении скорость была на порядок меньше, чем в 64-битном! В разных релизах оптимизатор снова выбирал различные планы исполнения запроса. При принудительном выставлении плана запросов картина сразу выравнилась: 64-разрядный релиз выполнял их на 10–20% быстрее.

Этот курьез показал нам еще один интересный момент. Дело в том, что bit-map-индексы создавались для поддержки колонок с низким значением cardinality (количеством уникальных элементов). В нашем хранилище использовались две таблицы фактов – sales_fact_1997 и sales_fact – с разным количеством записей, но с одинаковым количеством уникальных значений. То есть в случае с большей таблицей (sales_fact) мы имели гораздо меньшую селективность (а именно – в 8 раз). В конечном счете получилось, что для меньшей таблицы эффективней оказалась стратегия сканирования индекса, а для большей – наоборот, стратегия сканирования таблицы.

Наше объяснение этому таково. Вся малая таблица помещалась в кэш Oracle, поэтому просмотр индекса и последующий доступ к записям по идентификатору, не требующие обращения к диску, были эффективнее. В случае, когда таблица заметно выходила за пределы кэша, просмотр индекса и последующий доступ к записям по идентификатору, требующие теперь случайного обращения к диску, оказались менее эффективными, чем полное сканирование таблицы, при котором используются такие изолированные технологии, как многоблочное чтение.

Выборки из неиндексированных таблиц (28 000 000 записей, диаграмма № 2)

Тест	Команда	32 бита, с	64 бита, с	Прирост, %	Задействовано строк (rows)
Настройка по умолчанию	select count(*) from sales_fact_1997	5,057	2,52	200,436	28 962 208
Alter session set optimizer_goal=all rows	select count(*) from sales_fact_1997	2,423	2,59	93,408	28 962 208

Выборка из 8×28 000 000 записей (диаграмма № 3)

Тест	Команда	32 бита, с	64 бита, с	Прирост, %	Задействовано строк (rows)
Подсчет общего количества, без наложения условий	select count(*) from sales_fact	170,776	142,62	119,746	231 697 664
Подсчет количества с наложением условий	select count(*) from sales_fact WHERE product_id = 1 and time_id = 741 and customer_id = 614 and store_id = 17	175,873	152,92	115,009	1408

Индексация (диаграмма № 4)

Тест	Команда	32 бита, с	64 бита, с	Прирост, %	Задействовано строк (rows)
Создание битового индекса на таблице	create bitmap index sales_fact_7_product_bit on sales_fact_1997(product_id) nologging pctfree 0;	19,528	19,50	100,154	–
Создание битового индекса на таблице в 28 000 000 записей	create bitmap index sales_fact_7_by_time_bit on sales_fact_1997(time_id)	12,678	10,30	123,147	–
Создание битового индекса на таблице в 8×28 000 000 записей	create bitmap index sales_fact_by_time_bit on sales_fact(time_id) nologging pctfree 0;	196,252	174,89	112,213	–

Выводы

Проведенные исследования позволяют нам сделать несколько выводов:

1. Как и следовало ожидать, существующее 32-разрядное обеспечение, в том числе 32-разрядная версия СУБД Oracle 9i, функционирует на платформе AMD Opteron без всяких проблем. Миграция данных из 32-разрядной базы данных в 64-разрядную происходит абсолютно гладко.
2. Повышение производительности системы при переходе от 32-разрядной к 64-разрядной среде – очевидно. Особенно интересно, что этот переход при наличии соответствующей аппаратной базы осуществляется практически бесплатно, путем несложной миграции данных из одной базы в другую. 64-разрядная среда позволяет по мере необходимости наращивать объем памяти, добиваясь тем самым увеличения скорости обработки данных вместо того, чтобы изобретать хитроумные оптимизации.
3. Сегодня на рынке уже представлены все необходимые компоненты для

создания коммерческих x86-совместимых систем, которые затем без модернизации аппаратной части можно будет перевести в современную 64-разрядную среду, и наращивать производительность дальше. В частности, тестирование подтвердило возможность применения как 32-, так и 64-разрядных систем на основе процессоров AMD Opteron, работающих под управлением Linux и Oracle, в задачах управления и анализа бизнес-процессов.

4. Производительные современные серверы могут сэкономить коммерческим компаниям время и деньги в проектах по модернизации корпоративных систем управления за счет упрощения архитектуры хранилищ данных.

Тесты проводились на оборудовании и при содействии сотрудников Центра кластерных технологий для бизнеса компании Entry

E-mail авторов:

i.pilipenko@pmcg.com.ua, st@entry.kiev.ua



Pragma

MACHINE TRANSLATION SOFTWARE

Різдвяний подарунок

З 1 по 31 грудня всі програми серії Pragma зі знижкою 50%

Pragma Base 3.0

усього за 120 грн ви отримаєте базовий набір перекладача

Pragma Expert 3.0

за 450 грн — можливість перекладу в усіх напрямках — англ – рос – укр

Унікальна можливість

за лічені секунди здійснювати переклад різноманітних документів з англійської, німецької, російської та української мов

Більше ніж 800 тисяч слів у 100 тематичних словниках

ДЕ КУПИТИ

Trident Software

тел-факс: (044) 2524943

trident@kagi.com

www.trident.com.ua

ABBY Ukraine	(044) 4909999
АВЦ-Днепр	(0562) 349359
АМИ	(062) 3854888
Ай Ти Про	(044) 2644427
Алсита	(044) 2161171
Бакотек	(044) 2493418
Київ-ТЕК	(044) 2419145
Софтлайн Інл	(044) 2010300
СофтПром	(044) 2425300
Телец	(044) 5371846

Тестова версія програми знаходиться на web-сайті
Усі можливості перекладача упаковані у 10 Мб!

TRIDENT
Software