

УДК 004.45(045)

Чабан Т. О.

Національний авіаційний університет, Київ

ОСОБЕННОСТИ ПРОЕКТИРОВАНИЯ КЛИЕНТ-СЕРВЕРНЫХ СИСТЕМ ОПЕРАТИВНОЙ ОБРАБОТКИ ТРАНЗАКЦИЙ

В статье рассматриваются особенности проектирования клиент-серверных систем оперативной обработки транзакций (OLTP). Выделены основные отличительные особенности OLTP-систем от других существующих СУБД. Даны рекомендации касательно возможных путей повышения производительности OLTP-систем, а также выявлены основные тенденции и направления в развитии систем оперативной обработки транзакций.

Введение. Эффективность руководства компанией напрямую зависит от того, как успешно компания управляет информационными потоками. Без правильной обработки информацией невозможно принять верное управленческое решение. В современных условиях сбор, накопление, хранение информации уже не являются ведущей целью информационного процесса. Сегодня главной задачей систем управления информацией является анализ данных и принятие управленческих решений на их основе. С этой целью используются клиент-серверные системы оперативной обработки транзакций (OLTP – OnLine Transaction Processing).

Цель данного исследования – выявление основных характеристических особенностей проектирования клиент-серверных систем оперативной обработки данных. Предмет данной работы – особенности, тенденции и пути проектирования и развития OLTP-систем полезность баз данных, подходящих под классификацию "корпоративные". Объект исследования – основополагающие аспекты при построении клиент-серверных систем оперативной обработки транзакций.

Теоретическая ценность данной работы заключается в том, что в ней наиболее полно собраны особенности проектирования клиент-серверных систем оперативной обработки данных, а также тенденции и пути их развития. Практическая ценность работы состоит в том, что она раскрывает особенности проектирования OLTP-системы варианты подхода к их созданию и может использоваться в качестве вспомогательного материала при изучении клиент-серверных систем оперативной обработки транзакций.

Анализ последних исследований и публикаций. Проводилось несколько исследований уязвимых мест для производительности современных систем баз данных. Работы Мэйнгольда [1] и Эйламаки [2] показали, что возрастание объемов доступной основной памяти практиче-

ски не способствует повышению производительности систем баз данных. В исследованиях Мак-Вертера [3] анализируются места уязвимости, возникающие при конкуренции за различные ресурсы (блокировки, синхронизация ввода-вывода и другие), с точки зрения клиента (воспринимаемые задержки из-за операций ввода-вывода или вытесняющее планирование других параллельно обрабатываемых запросов).

Существует также большое число публикаций, посвященных системам баз данных в основной памяти. В исследованиях индексных структур в основной памяти анализировались Ахом [4] и Леманом [5]. Другие методы применения основной памяти подробно были описаны Биттоном [6]. К числу полных систем баз данных в основной памяти относятся DataBlitz [7], MARS [8] и TimesTen [9]. Обзор этой области содержится в исследованиях Гарсии-Молины и Салема [10]. Однако ни в одной из этих работ не предпринимались попытки изолировать компоненты накладных расходов, что является основным вкладом данной статьи.

Системы оперативной обработки транзакций – это способ организации и проектирования БД, в котором система работает с небольшими по размеру транзакциями, идущими большим потоком.

Основная функция OLTP-систем заключается в одновременном выполнении большого количества коротких транзакций, производимых большим числом пользователей одновременно. Сами транзакции выглядят отличаются относительной простотой. Например, «снять сумму денег со счета А, положить ее на счет В».

Как следствие, клиенту необходимо от системы максимально быстрое время ответа. Так как OLTP-системы используют обработки информации в режиме реального времени, то основным показателем эффективности OLTP-приложения

является количество выполняемых за секунду транзакций.

OLTP системы создавались для ввода, структурированного хранения и обработки информации в режиме реального времени. OLTP-системы работают с запросами типа «сколько», «где» и им подобными. Предоставляя данные из постоянно обновляемых БД, такие системы не отслеживают динамику изменения процессов на больших временных промежутках и практически не производят обработку данных, не считая определенных расчетов. Главным отличием OLTP-систем от OLAP является то, что они не формируют выводы по имеемым данным, оставляя эту функцию принятия решения самому пользователю.

Тенденции в области OLTP. Истоки OLTP-систем восходят к системам 1970-х годов. В таких системах использовались дисковые индексные структуры и неупорядоченные файлы, а транзакции поддерживались на основе журнала и управление параллелизмом с блокировкой. В настоящее время OLTP-системы развиваются по совершенно другим путям, нежели 30 лет назад. Аналогичные наблюдения описывались Стоунбрейкером в книге «Конец архитектурной эпохи, или Наступило время полностью переписывать системы управления данными» [12]. К основным тенденциям развития OLTP-систем можно отнести следующие:

а) кластерные вычисления.

В последние двадцать лет появились OLTP-системы с архитектурой без общих ресурсов, так называемые «shared nothing» системы и кластеры из персональных компьютеров массового класса для решения многих масштабных вычислительных задач. Любая будущая система баз данных, предназначенная для использования на этих кластерах, должна разрабатываться с нуля;

б) базы данных, хранимые в основной памяти.

Существенный рост размеров доступной основной памяти за последние несколько десятилетий позволил многие базы данных OLTP разместить основной памяти, в особенности, в совокупной основной памяти крупного кластера. Это происходит благодаря тому, что размер большинства баз данных OLTP растет далеко не так быстро, как объем доступной основной памяти. С учетом этого фактора поставщикам OLTP имеет смысл создавать системы, оптимизируемые в расчете на распространенную ситуацию, когда база данных целиком помещается в основной памяти;

в) однопоточный режим в системах OLTP.

Во всех современных OLTP-системах имеется обширная поддержка многопоточного режима. В том числе, набор протоколов транзакционного управления параллелизмом, а наличие в код команд кратковременных блокировок, так называемых «защелок», обеспечивающими корректность доступа нескольких потоков управления к совместно используемым структурам (буферные пулы и индексные страницы). Традиционные аргументы в пользу многопоточного режима состоят в том, что он позволяет избежать простоя при обработке нескольких транзакций, когда одна или несколько транзакций ждут поступления данных с диска, а также предотвратить торможение длинными транзакциями выполнения коротких транзакций. Но по мнению многих современных специалистов в некоторых классах приложений баз данных не требуется поддержка многопоточного режима. В таких системах унаследованный код для поддержки блокировок и защелок становится неоправданным накладным расходом;

г) высокая доступность в противовес журнализации.

Для производственных систем обработки транзакций требуется круглосуточная доступность баз данных. По этой причине в большинстве систем используется некоторая форма обеспечения высокой доступности, как правило, основанная на наличии двукратной (или большей) избыточности аппаратуры для гарантированного наличия доступной резервной системы в случае отказа;

д) варианты транзакций.

Хотя для многих систем OLTP, несомненно, требуется транзакционная семантика, в последнее время появились предложения по поводу систем управления данными с ослабленной согласованностью. Обычно требуется некоторая форма конечной согласованности при наличии убежденности в том, что доступность более важна, чем транзакционная семантика. Системы баз данных для таких сред, по всей вероятности, мало нуждаются в механизмах, разработанных для поддержки транзакций.

И, наконец, современные исследования показывают, что имеются ограниченные формы транзакций, для поддержки которых требуется гораздо меньше механизмов, чем для поддержки стандартных транзакций над базами данных. Выбор типа клиент-серверной архитектуры зависит от географического расположения узлов сети пред-

приятія, вимог до надійності, швидкодії та обслуговуванню. Існують наступні типи реалізації OLTP-технологій: файл-серверна архітектура, двохрівнева і трьохрівнева клієнт-серверні архітектури.

Сервером є обслуговуючий інформаційну потребу клієнта процес. Клієнтом, як правило, є кінцевий користувач, що надсилає запит на обслуговування або застосування, викликане кінцевим користувачем.

Наслідки для майбутніх серверів баз даних OLTP:

а) управління паралелізмом. Великий вигода може принести виявлення сценаріїв, таких як коммутативність застосувань або чергова обробка транзакцій, для яких дозволено відключити управління паралелізмом. Однак існує багато застосувань баз даних, які недостатньо добре дотримуються подібних приличій або не можуть працювати в режимі виконання транзакцій по одній. В таких випадках цікавим питанням є те, який із протоколів управління паралелізмом є найкращим? Двадцять років тому різні дослідники виконали всебічне імітаційне моделювання, яке чітко показало перевагу динамічних блокувань над іншими методами управління паралелізмом. Однак в цих дослідженнях передбачалося збереження баз даних на дисках і наявність відповідних простих транзакцій в очікуванні завершення введення-виведення, і це, очевидно, суттєво вплинуло на результати. Було б дуже бажано знову створити такі моделювання, маючи на увазі робочу навантаження з базами даних в основній пам'яті і виявити деяку різновидність оптимістичного управління паралелізмом;

б) підтримка багатоядерних процесорів. Зростаюча поширеність багатоядерних комп'ютерів призводить до наступного питання: що будуть робити з кількома ядрами майбутні сервери баз даних OLTP? Один варіант полягає в паралельному виконанні кількох транзакцій на різних ядрах в одному вузлі, як це робиться сьогодні. Але такий паралелізм вимагає використання замикачів і призводить до потреби рішення багатьох проблем розподілу ресурсів, хоча накладні витрати механізму замикачів не особливо великі, але вони залишаються перешкодою для досягнення суттєвого підвищення продуктивності систем баз даних OLTP. Другий варіант оснований на

використанні віртуалізації на рівні операційної системи або СУБД, щоб кожне ядро представляло як однопоточна машина. Але дуже незрозуміло, які наслідки матиме цей підхід для продуктивності. Третій варіант полягає в тому, щоб спробувати використовувати внутрізапитний паралелізм, який може бути виправдан, навіть якщо в системі транзакції виконуються по черзі. Однак обсяг внутрізапитного паралелізму, доступного в типовій транзакції OLTP обмежений;

в) управління реплікацією. В традиційних системах баз даних прийнято підтримувати реплікацію на основі схеми «активний-пасивний» шляхом наділення журнальної інформації. У кожного об'єкта існує «активна» первинна копія, до якої, в першу чергу, направляються всі зміни. Далі журнал змін передається по мережі в один або кілька «пасивних» резервних вузлів. Після цього в віддалених базах даних по журналу створюються зміни, виконані в первинному вузлі. Але дуже корисно було б дослідити альтернативи реплікації «активний-пасивний», такі як підхід «активний-активний». Основною причиною того, що в минулому використовувалася реплікація «активний-пасивний» з наділенням журналу, є те, що створення змін по журналу обходиться значно дешевше виконання над реплікою логіки транзакції. Однак в системі баз даних в основній пам'яті час виконання транзакції зазвичай становить менше однієї мілісекунди, що, ймовірно, незначно перевищує час, необхідний для створення змін по журналу. В цьому випадку альтернативна архітектура «активний-активний» здається цілком осмисленою. При цьому підході всі репліки є «активними», і транзакція виконується синхронно над усіма репліками. Перевагою цього підходу є майже миттєва обробка відмов, і відсутність вимоги напрямлення змін спочатку в вузол з первинною копією;

г) слабка узгодженість. В більшості великих Web-орієнтованих магазинів для досягнення високої доступності і забезпечення відновлення після аварійних відмов використовується реплікація баз даних OLTP, зазвичай в глобальній мережній середі. Однак, ніхто не платить за транзакційну узгодженість в глобальній мережі. Широко поширеним референтом в Web-застосуваннях є «кінцевий

ная согласованность». Обычно сторонники такого подхода выступают за разрешение несогласованности не техническими средствами; например, дешевле предоставить кредит недовольному клиенту, чем поддерживать стопроцентную согласованность. Другими словами, реплики, в конечном счете, становятся согласованными, по видимому, тогда, когда система переводится в пассивное состояние.

Особенностью файл-серверной архитектуры является такое проектирование СУБД, при котором приложения пользователей вместе с системой управления находятся на клиентской части, а сами файлы данных располагаются на сервере. При данном подходе к проектированию СУБД управляющие программы располагаются в оперативной памяти рабочих станций (PC) локальной сети, а файлы баз данных – на магнитных дисках файл-сервера. Особенность файл-серверов состоит в том, что вся обработка данных производится на PC, а файл-сервер выполняет лишь функцию накопителя данных. В результате расположения клиентов и серверов в различных узлах сети, файл серверная технология является распределенной и реализует многопользовательский режим работы. Синхронизацию совместного использования данных файл-серверов выполняет СУБД. Она обеспечивает блокирование записей для других рабочих станций во время внесения в записи БД изменений.

Выводы. Основы проектирования OLTP-систем в таком виде, как они преподаются сегодня в высшей школе, существовали несколько десятков лет назад – в 70-х годах XX ст., когда зародились основы теории всех практически СУБД. Сегодня же тенденции в развитии файл-серверных систем обработки транзакций и особенности их проектирования кардинально поменялись. Поэтому преподавателям высших учебных заведений при составлении учебных планов весьма желательно было бы учитывать аспекты развития рынка OLTP-систем, изложенные в данной статье.

Как показывают последние исследования в области OLTP-систем, современные (с хранением данных в основной памяти) системы баз данных могут демонстрировать высокую производительность не только за счет отсутствия обме-

нов с дисками. Ведь в традиционных системах существенное время тратится на журнализацию, блокировки, защелки, двухфазную фиксацию, поддержку B-деревьев и др. Исключая их из работы системы можно добиться существенного повышения производительности OLTP-систем.

Список литературы

1. Boncz, P. A., Manegold, S., and Kersten, M. L. "Database Architecture Optimized for the New Bottleneck: Memory Access." In Proc. VLDB, 1999.
2. Ailamaki A., DeWitt D. J., Hill M. D., and Wood D. A. "DBMSs on a Modern Processor: Where Does Time Go?" In Proc. VLDB, 1999, pp. 266-277.
3. McWhorter D. T., Schroeder B., Ailamaki A., and Harchol-Balter M. "Priority Mechanisms for OLTP and Transactional Web Applications". In Proc. ICDE, 2004.
4. Aho A. V., Hopcroft J. E., and Ullman J. D. "The Design and Analysis of Computer Algorithms." Addison-Wesley Publishing Company, 1974. Имеется перевод на русский язык: Ахо А., Хопкрофт Дж., Ульман Дж. Построение и анализ вычислительных алгоритмов. М.: Мир, 1979.
5. Lehman T. J. and Carey M. J. "A study of index structures for main memory database management systems". In Proc. VLDB, 1986.
6. Bitton, D., Hanrahan, M., and Turbyfill, C. "Performance of Complex Queries in Main Memory Database Systems". In Proc. ICDE, 1987.
7. Baulier J. D., Bohannon P., Khivesara A., et al. "The DataBlitz Main-Memory Storage Manager: Architecture, Performance, and Experience". In The VLDB Journal, 1998.
8. Eich M. H. "MARS: The Design of A Main Memory Database Machine". In Proc. of the 1987 International workshop on Database Machines, October, 1987.
9. Oracle TimesTen, 2007. http://www.oracle.com/us/corporate/Acquisitions/timesten/index.html?origref=http://citforum.ru/database/articles/oltp_lg/links.shtml
10. Garcia-Molina H. and Salem K. "Main Memory Database Systems: An Overview". IEEE Trans. Knowl. Data Eng. (1992). – С. 509-516.
11. OLTP – системы оперативной обработки транзакций – <http://www.itstan.ru/it-i-is/oltp-sistemy-operativnoj-obrabotki-tranzakcij.html>
12. Stonebraker M., Madden S., Abadi, D. J., Harizopoulos S., Hachem N., and Helland P. "The End of an Architectural Era (It's Time for a Complete Rewrite)". In Proc. VLDB, 2007.

Научный руководитель – Труш А. И., канд. техн. наук, доц.