

ПЕРСПЕКТИВНЫЕ ОБЛАСТИ ИССЛЕДОВАНИЙ ПОВЫШЕНИЯ ПРОИЗВОДИТЕЛЬНОСТИ БАЗ ДАННЫХ

Институт компьютерных технологий
Национального авиационного университета

Анализ современных и перспективных методов, повышающих производительность и пропускную способность баз данных.

Введение

На примере виртуальной лаборатории Земли и планетарных материалов (ВЛЗПМ) рассматривается система распределения данных на основе фаз.

Системные вычисления основываются на фазах, которые могут быть разьединенными и выполняются одновременно.

Каждая фаза генерирует значительное число выходных файлов, разбросанных по всей площади распределённой компьютерной базы данных. Эти файлы должны быть пересобраны между фазами для того, чтобы вызвать следующий ввод.

Сервис-ориентированная архитектура SOA [1], киберинфраструктура (SOAC) разработаны для выполнения вычислений такого плана.

Метаданные

Рассматривается описание метаданных и алгоритмов управления ими, необходимых для управления распределённым выполнением задач, являющихся частью многофазных потоков, в том числе перераспределением данных между фазами. Метаданные - это учётная информация об использовании системных ресурсов данными. Вместо ссылочных списков, чьи элементы указывают на фактические данные, метаданные используются для того, чтобы обеспечить структуру большей динамикой и экономичностью. В традиционных сильносвязанных параллельных системах параметры доменов напрямую присоединены к узлам и метаданные обычно просты. Однако, в распределённых сетях, специфические ресурсы абстрактны, прямое соответствие между доменами и ресурсами теряется, поэтому использование тщательно разработанных метаданных явля-

ется, в данном случае, наиболее рациональным выходом.

Введение метаданных является решающим в разработке системы, требующей поддержки одновременного выполнения и работы системы в распределённой разнотипной среде.

Для поддержки ресурсов общего доступа метаданные должны включать набор семафоров. Один семафор включается в дескриптор проекта. Для каждого приёма данных необходим собственный семафор. Следовательно, существуют семафоры метаданных, относящиеся к каждой l задаче, включая и проект. Каждый из этих семафоров, управляет доступом к соответствующей задаче и задаче данных.

Очень важно хранить метаданные в центральной базе данных с момента, когда управление проектом начинает оперировать с кэшированной копией приёма проекта. Вследствие этого, дополнительно к семафору при каждой операции приёма и в дескрипторе проекта включается счётчик. Счётчики считают число изменений, но предварительно контролируется операция приёма и тестируется дескриптор проекта.

Для проверки последовательности кэшированной доставки достаточно просто сравнить состояние счётчика в базе данных с кэшированной копией. Доступ к семафору осуществляется путём его поворота. Гарантия последовательности обеспечивается механизмом блокировки данных, который доступен на сервере базы данных.

Для передачи принятой информации с сервера на сервер и хранения в базе данных нужно конвертировать полученные данные в некоторый формат, отличающийся от *java* класса. Для этой цели

ВЛЗПМ использует XML. Библиотека [2] предоставляет возможность хранения содержимого атрибутов *java* класса в XML файла. Структура XML файла иерархической структуры исходного *java* класса совпадает с каждым XML тегом, связанным с соответствующим атрибутом класса. Далее XML файл может быть передан или сохранён в базе данных. Библиотека [2] также предоставляет возможность обратной конвертации снова в *java* для дальнейшей обработки.

В базе данных хранится дескриптор проекта храниться в базе данных, включая все прикрепленные данные и важные метаданные. Этим обеспечивается возможность сохранения данных для их применения при получении следующих данных. В основном рассматриваются метаданные, направленные на распределённое выполнение, совместный контроль и управление проектами в распределённых средах.

Распределённая объектно – ориентированная система баз данных (РООСБД)

Основным свойством данной архитектуры является использование работающих сообща агентов для выполнения запросов и транзакций базы данных, использование механизма удалённого объекта для совместного обмена информацией между агентами многоуровневых транзакций.

Для работоспособности базы данных и выполнения концепции многоуровневых транзакций применительно к объектам рекомендуется использование множества уровней. Представленная система использует гибридный многоуровневый параллельный протокол *FoPL*, который объединен с многоуровневой системой восстановления, основанной на использовании системы *ARIES*.

Целесообразным считается создание системы на основе применения базовой объектно-ориентированной модели [3]. Одним из её преимуществ, по сравнению, с моделью *ODMG's* является её математическая строгость.

В сущности предлагаемая система, как и все современные базы данных име-

ет многоуровневую внутреннюю архитектуру.

База данных может быть представлена как совокупность физических объектов, хранящихся в постоянном запоминающем устройстве. Доступ к этим объектам возможен с помощью средств устройств управления памятью.

Система имеет дело с дисковым пространством и поддерживает высокий уровень доступа к физическим объектам.

Модуль кеширования кроме управления страницей в главной памяти поддерживает принцип записей. Записи, которые используются в большинстве случаев, отображаются вместе на одной странице (если это возможно). С того момента, когда множество записей занесены на одну страницу, протоколы операций должны быть синхронизированы. Это достигается путём использования кратковременных регистров-защёлок.

Модуль кеширования имеет два чётко определённых интерфейса: интерфейс страницы и интерфейс записи. Интерфейс страницы доступен всем высшим уровням. Среди других он используется для хранения оперативных, организационных и вспомогательных данных.

Интерфейс записи используется для всех запросов данных. Он доступен только модулю на следующем высоком уровне - файловому объекту хранения (ФОХ), который делает возможным другой уровень абстракции по средствам поддержки хранения объектов.

Объект в памяти конструируется из записей и имеет уникальный идентификатор объекта. ФОХ поддерживает прямые физические ссылки между хранимыми объектами и предлагает связанный с объектом ассоциативный и навигационный доступ к объектам. Доступ использует связь между объектами в памяти таким образом, чтобы осуществить реконструкцию объектов более сложной структуры.

Доступ, связанный с объектом, использует прямой доступ к объекту, используя идентификаторы объекта. Ассоциативный доступ обеспечивает хорошо известный доступ через ключевые значе-

ния. Навигационный доступ связан с передачей данных по физическим ссылкам.

Движок оценки запросов постоянно находится на вершине ФОХ. Этот элемент выполняет запросы клиентских приложений, как только они передаются с высших уровней. Движок оценки запросов использует большое количество агентов, которые объединяются всякий раз, когда это возможно, и выполняет работу, требуемую приложениями.

Агенты могут реализовываться как потоки. Совокупность агентов, которая работает совместно по запросу приложения, на отдельном узле РООСБД принадлежит к одному и тому же процессу. В рамках процесса агенты классифицируются в зависимости от их ролей (*Master* или *slave*) и в соответствии от их вида (локальный или распределённый). Агенты с распределённым отображением имеют способность распределять запросы по удалённым узлам РООСБД, так как, они оборудованы механизмом удалённого вызова объекта. Агенты с локальным отображением используются для выполнения подзапросов, которые включают только объекты локальных данных. Объекты локальных данных могут быть извлечены только с помощью агентов с локальным отображением. Таким образом, на локальном или любом удалённом узле РООСБД агенты с распределённым отображением имеют объединение, по крайней мере, с одним локальным агентом. Как результат, движок оценки запросов является низшим уровнем в системе базы данных.

Агенты оптимизируются для выполнения запросов (запрос алгебры операций, методы, назначенные классам и т.д.) отдельных типов объектов, соответствующих их отображению. Агенты с распределённым отображением выполняют запросы для более сложных объектов, которые включают объекты, хранимые в памяти локально и удалённо. Они используют распределение, параллелизм, многопоточность и поддерживают концепцию транзакций. Агенты с локальным отображением используют параллелизм и многопоточность для улуч-

шения выполнения запросов локальных объектов, которые содержат только объекты, хранимые локально. Имея множество уровней объектов предоставляется возможность использования более усовершенствованной системы управления транзакциями, основанной на многоуровневой модели транзакций.

Система управления транзакциями управляет и контролирует выполнение операций, выполняемых движком оценки запросов и файловым объектом хранения. Следовательно, система обеспечивает локальный и глобальный сериализм. Система состоит из двух компонентов: менеджер транзакций и менеджер восстановления. Менеджер транзакций обладает преимуществом обнаружения псевдоконфликтов (конфликтов, которые не происходят из конфликта высокого уровня) и возможностью планирования многоуровневого сериализма с помощью сериализации конкретных операций (подтранзакций) уровень за уровнем. Вследствие этого могут быть использованы различные протоколы специфических уровней. В настоящее время поддерживается определённый двухфазный блокировочный протокол *str - 2PL* и гибридный *FoPL*. Их можно использовать во всех возможных комбинациях на соответствующем уровне в зависимости от вероятности конфликта. Менеджер восстановления гарантирует атомарность, устойчивость и последовательность данных. Это достигается с помощью поддержки локальных записей, отражающих обновления объектов на всех уровнях, поддержки полных и частичных возвратных операций (подтранзакций, повторно выполняемых операций подтранзакций, восстановление после сбоя и т.д.). Для их функционирования используется восстанавливающий *ARIES/ML* механизм. Он является расширением хорошо известного алгоритма восстановления *ARIES* для многоуровневых систем.

На логическом уровне данные изображаются в терминах модели данных. Представленная система основана на обобщённой объектно-ориентированной модели данных ООМД [3]. Эта модель

рассматривает объекты, как абстракции объектов реального мира. ООМД различает величины и объекты. Каждый объект состоит из уникального постоянного идентификатора, набора (тип, значение) пар, набора (свойство, объект) пар и набора методов. ООМД основывается на произвольном типе любой базовой системы. Типы используются для структуризации значений. Классы служат структурированными базовыми элементами для объектов, имеющих одинаковую структуру и принцип работы. Операции, обеспечиваемые системой приведённого ниже типа с единичным оператором присоединения, позволяют определить соответствующий обобщённый запрос порождающей алгебры. Для того чтобы осуществить распределение, применяется определённая техника фрагментации. Это разбиение горизонтальной и вертикальной фрагментации. Для выполнения этих задач используются классы. Каждый класс назначается для определённого РООСБД узла, а в случае репликации нескольких РООСБД узлов, они объединяются в сеть. Следовательно, фрагментация разбивает на составные части (несколько локальных объектов), глобальные объекты, которые соответствуют фрагментированной схеме. Произведя фрагментацию и отношение класс/узел, необходимо распределить фрагменты, включая фрагментированные методы, по соответствующим узлам РООСБД.

Объекты, образованные в результате фрагментации, прямо не соответствуют объектам, обрабатываемым агентами движка оценки запросов. Кроме того, запросы высокого уровня, транзакции, объектные методы и т.д. должны быть переведены в код, который может быть интерпретирован этими агентами. Концептуальный интервал между логическим ООМД уровнем и движком оценки запросов соединен мостом с модулем отражения. Макроязык формулирует транзакции высокого уровня (например, общие операции обновлений и алгебра высокого уровня). Модуль обработки запросов поддерживает ООМД, фрагмента-

цию и распределение. Кроме того, используется оптимизатор запросов, который формирует план выполнения для оценки запроса пользователя. План выполнения используется модулем отражения, в котором конструкции высокого уровня замещаются макросами. Далее движок оценки запросов будет использовать план выполнения с макросами для определения количества запросов пользователя.

Внутреннее представление объектов, копирование, распределение данных и т. д. скрыто от пользователя. Это осуществляется с помощью пользовательского интерфейса. Наряду с этим, интерфейс предоставляет диалоговые интерфейсы, основанные на диалоговых объектах, которые определяются расширенными обозревателями. Эти диалоговые объекты могут быть созданы в любом сегменте сети. Вызов операции, связанной с таким диалоговым объектом, создаст основного агента и приступит к выполнению транзакции высокого уровня (запрос пользователя).

Кластеризация

Кластеры и решётки рабочих станций предоставляют ресурсы, доступные для работы с данными. Для использования этих ресурсов необходимы новые алгоритмы и детальная разработка способа распределения данных и дальнейшего использования этого распределения. Рассматривается дублированный алгоритм последовательной кластеризации. Основным признаком интеллектуального разделения является максимальная однотипность разделённых фрагментов. Этот признак позволяет параллелизовать задачу, обычно встречающуюся в базе данных.

Кластеризация – это процесс разделения данных на отдельные группы (кластеры) таким образом, что объекты внутри одного и того же кластера подобны, но не подобны объектам из других кластеров.

При кластеризации используются иерархические методы, изначально предусматривающие разделение кластеров уникальных типов и осуществляющие

объединение с соседними кластерами до тех пор, пока не встретится завершающее условие.

Методы разделения изначально рассматривают разделение с одним кластером, содержащим в себе все экземпляры класса и осуществляющих повторное перемещение кластеров до завершения. Общий итеративный алгоритм последовательной распределённой кластеризации приводит к глобальному распределению базы данных, путём привязки двух разделённых частей на каждом повторении.

Выводы

Основным преимуществом и перспективным направлением в развитии метаданных является использование метаданными ссылок на фактические данные, что обеспечивает улучшение структуры и повышение экономичности.

Распределённая объектно-ориентированная система баз данных является перспективным и развивающимся направлением за счёт общего использования агентов, которые управляют транзакциями в базах данных.

Общий алгоритм последовательной распределённой кластеризации обеспечивает качественный результат, такой же как и увеличение скорости системы путём повышения параллельных и распределённых вычислений. При этом повышается уровень сложности и коэффициент усиления в задаче ассоциативных правил. Фаза кластеризации должна распределяться и выполняться быстро, чтобы не было снижения общего времени системного выполнения.

Список литературы:

1. D. Gannon, B. Plale, M. Christie, L. Fang, Y. Huang, S. Jensen, G. Kandaswamy, S. Marru, S.F.L. Pallickara, S. Shirasuna, Y. Simmhan, A. Slominski, Y.M. Sun, Service oriented architectures for science gateways on grid systems, in: Lecture Notes in Computer Science, vol. 3826, 2005. 21-32.
2. See <http://www.castor.org>.
3. See <http://VLab.msi.umn.edu/projects/ITResearch.shtml>.

4. R. Agrawal, J. Gehrke, D. Gunopulos, P. Raghavan, Automatic subspace clustering of high dimensional data, Data Mining and Knowledge Discovery Journal 11 (1) (2005) P494.

5. K.-D. Schewe, B. Thalheim, Fundamental concepts of object oriented databases, Acta Cybernetica 11 (1-2) (1993) P. 49-84.

6. K.-D. Schewe, On the unification of query algebras and their extension to rational tree structures, in: Proceedings of the 12th Australasian Conference on Database Technologies, IEEE Computer Society Press, 2001, pp. 52-59.

7. K.-D. Schewe, Fragmentation of object oriented and semistructured data, in: Proceedings of the Baltic Conference, BalticDB&IS 2002, Institute of Cybernetics at Tallinn Technical University, 2002, pp. 253-266.

8. H. Ma, K.-D. Schewe, A heuristic approach to horizontal fragmentation in object oriented databases, in: J. Barzdins, A. Caplinskis 2004, p 567.

9. Жуков И.-А., Иванкевич О.-В., Аль Шибани Салим, Средства повышения эффективности обработки баз данных большого объёма в информационных системах авиапредприятий Украины, материалы VIII Междун. науч.-практич. конф. – «АВИА-2007. Т. 1. – К.: НАУ, 2007. – С. 13.37-13.40.

10. Жуков И.-А., Иванкевич О.-В., Методы ускорения обработки больших баз данных с применением многопроцессорных модулей, Проблемы информатизации и управления. К.: НАУ, 2005. – Вып. 12. – С.68-76.