РАЗРАБОТКА ПРОГРАММНЫХ СРЕДСТВ СБОРА СТАТИСТИКИ РАБОТЫ РАСПРЕДЕЛЕННОЙ БАЗЫ ДАННЫХ

Остроухова Я.И., группа ИУС-05м
Руководитель доц. каф. АСУ Телятников А.О.

Основной причиной разработки систем, использующих базы данных, является стремление интегрировать все обрабатываемые в организации данные в единое целое и обеспечить к ним контролируемый доступ. Создание компьютерных сетей приводит к децентрализации обработки данных. Децентрализованный подход отражает организационную структуру компании, логически состоящую из отдельных подразделений, отделов, проектных групп, которые физически распределены по разным офисам, отделениям, предприятиям или филиалам, причем каждая отдельная единица имеет дело с собственным набором обрабатываемых данных. Разработка распределенных баз данных (РБД), отражающих организационные структуры предприятий, позволяет сделать данные, поддерживаемые каждым из существующих подразделений, общедоступным, обеспечив при этом их сохранение именно в тех местах, где они чаще всего используются. Подобный подход расширяет возможности совместного использования информации, одновременно повышая эффективность доступа к ней.

Распределенная база данных представляет собой набор логически связанных между собой разделяемых данных, которые физически распределены в некоторой компьютерной сети при помощи репликации и фрагментации [1]. Основными процессами, протекающими в распределенной базе данных, являются выполнение запросов к распределенным данным и распространение обновлений к множеству копий данных, расположенных на разных узлах компьютерной сети. Производительность распределенной базы данных зависит не только от параметров технических средств, но и от того, насколько рационально распределены данные по

узлам компьютерной сети. Обеспечение роста производительности таких систем за счет повышения эффективности обработки информации в базах данных связано с решением одной из основных проблем — проблемой рационального размещения данных по узлам компьютерной сети, что дает возможность, не увеличивая стоимость сети, повысить скорость обработки данных. Вопросам оптимизации распределенных баз данных посвящен ряд научных работ и публикаций. Весомый вклад в развитие этого направления внесли Г.Г. Цегелик, А.Г. Мамиконов и другие ученые. Несмотря на проведенные исследования задача оптимального распределения данных не получила окончательного решения. Одним из нерешенных вопросов остается сбор исходных данных (параметров распределенной базы данных), которые на данный момент могут быть получены только в приближенном виде.

В общем виде процесс оптимизации схемы распределения данных представлен на рис. 1. Для оптимизации распределенной базы данных с критерием эффективности — минимальное суммарное среднее время выполнения запросов и распространения обновлений, совместно с модифицированным генетическим алгоритмом используется объектная модель распределенной базы данных. Генетический алгоритм формирует набор хромосом, кодирующий распределение данных по узлам сети, а с помощью объектной модели вычисляется значение целевой функции [2,3]. Входными данными для моделирования и оптимизации является статистика выполнения запросов и распространения обновлений, а результатом — модифицированная схема распределения данных. Сбором статистики путем сложного анализа функционирования реальной РБД и ее модификацией на основании полученной субоптимальной схемы распределения данных занимается эксперт. Недостатком такого подхода является отсутствие возможности получить достоверную статистику работы распределенной базы данных. С целью упрощения процесса сбора исходных данных для моделирования и повышения их точности предлагается выполнить автоматизацию сбора статистики с помощью программных средств.

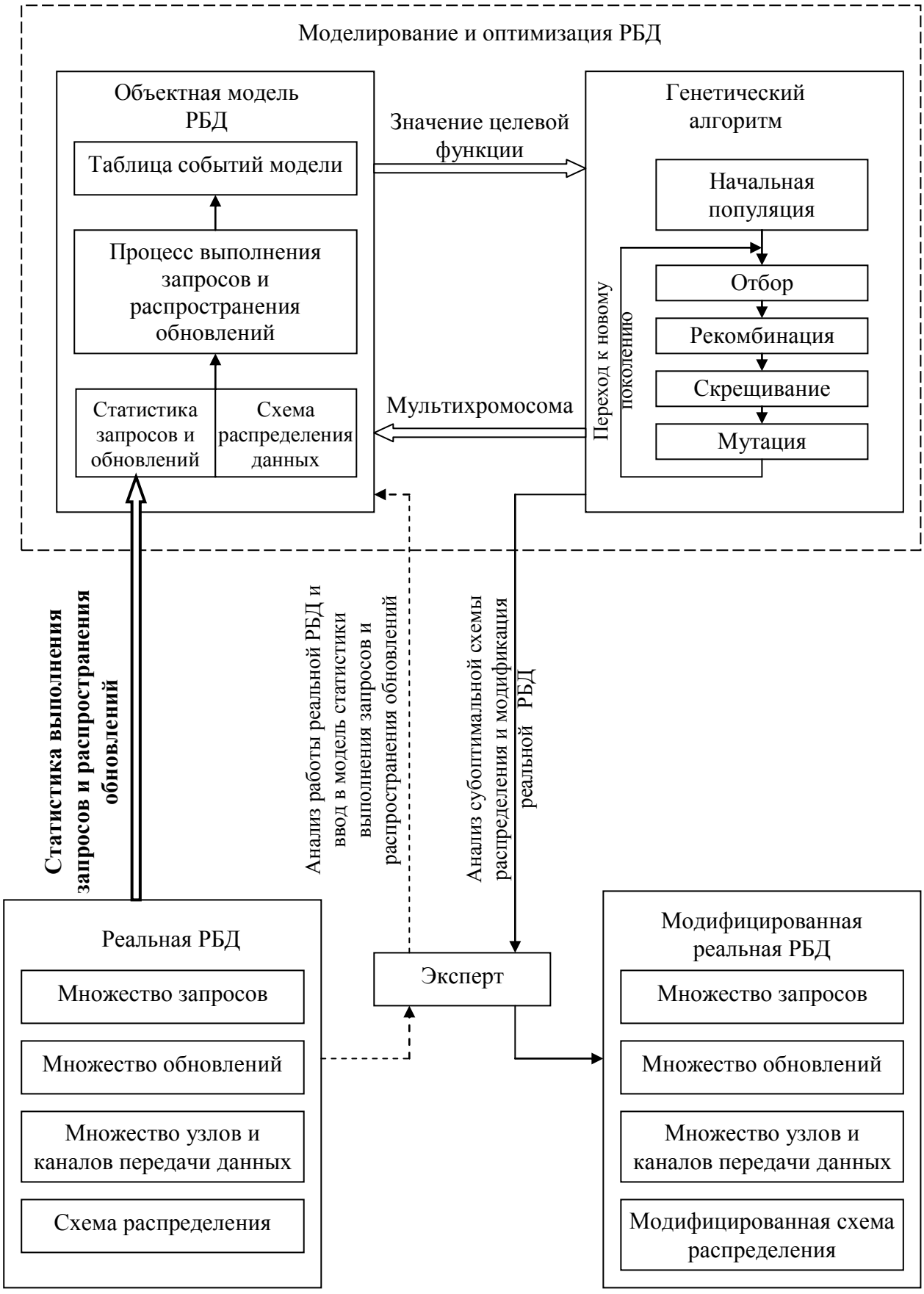

Рисунок 1 — Процесс оптимизации схемы распределения данных по узлам сети

В настоящее время большинство современных СУБД (IBM DB2, Oracle, Ingress, SQL Server 2000) обеспечивают поддержку специфических функций, необходимых для работы с распределенными базами данных. Для автоматизации сбора статистики необходимо изучить функционирование реальной распределенной базы данных в конкретной СУБД, так как каждая из них имеет ряд особенностей. Одной из мощных СУБД, обеспечивающих поддержку распределенных баз данных, является SQL Server 2000. Для реализации распределенных баз данных в SQL Server 2000 существует система репликации, которая представляет собой совокупность механизмов, обеспечивающих отображение изменения данных, сделанных на одном сервере, на другие сервера. Терминология репликации использует три основных понятия: издатель, подписчик и дистрибьютор. Издателем является сервер, предоставляющий расположенную на нем информацию другим серверам. Администратор конфигурирует на издателе публикацию, включая в нее одну или более статей. Подписчиком является сервер, который принимает данные от издателя. Дистрибьютор же служит промежуточным звеном между издателем и подписчиком, его роль сводится к сбору всей информации, которая должна быть скопирована подписчикам от издателя.

В SQL Server 2000 применяется три основных типа репликации: репликация моментальных снимков, репликация транзакций и репликация сведением. Наиболее универсальной является репликация сведением, которая может работать как при наличии постоянного физического соединения, так и без него, а главным преимуществом по сравнению с остальными типами репликации является возможность вносить изменения в публикуемые данные не только на издателе, но и на подписчике.

В зависимости от используемого типа репликации при создании реальной распределенной базы данных на издателе, подписчике и дистрибьюторе создается специфический набор системных таблиц, хранящих информацию обо всех публикациях и подписках, а также изменениях, сделанных в них.

Рассмотрим создание средств сбора статистики реальной распределенной базы данных о количестве и длительности выполнения обновлений на примере репликации сведением. Для этой задач главным источником информации будут являться три системные таблицы, описание которых приведено в табл. 1. Набор таких таблиц содержится в базе данных на издателе и на подписчике.

Таблица 1 — Системные таблицы репликации сведением

Название таблицы	Описание
MSmerge_contents	Содержит информацию о выполнении инструкций INSERT и UPDATE по каждой строке реплицируемой таблицы. В случае многократных изменений одной и той же строки, информация содержится только по последнему изменению.
MSmerge_tombstone	Содержит информацию о выполнении инструкции DELETE по каждой строке реплицируемой таблицы.
MSmerge_genhistory	Хранит информацию о каждом сеансе, в котором были произведены модификации реплицируемых данных. Для идентификации сеанса используется три значения: порядковый номер сеанса, глобальный уникальный идентификатор сеанса и локальный уникальный идентификатор сеанса.

При возникновении обновления на узле A информация о таблице, строке и номере сеанса записывается соответственно в таблицы MSmerge_contents или MSmerge_tombstone узла A, после чего изменения передаются дистрибьютору. В свою очередь дистрибьютор, приняв изменение от узла A, направляет его узлу B, на котором в такие же системные таблицы (MSmerge_contents и MSmerge_tombstone) добавляются соответствующие записи. Создав триггеры на такие системные таблицы можно отследить все изменения по каждой строке

реплицируемой таблицы, а для связывания соответствующих изменений на серверах — участниках репликации используется глобальный уникальный идентификатор сеанса, который хранится в таблице MSmerge_genhistory.

Для сбора информации о выполнении распределенных запросов предлагается использовать утилиту SQL Profiler, которая регистрирует события SQL Server и предоставляет ряд возможностей для анализа выполненных операций [4]. Для того чтобы получить возможность отслеживать с помощью SQL Profiler выполнение распределенных запросов, необходимо создать задание трассировки, установив события SQL:StmtCompleted и Execution Plan, и запустить его на выполнение. Событие SQL:StmtCompleted будет фиксировать каждую SQL — инструкцию с указанием временных параметров, а событие Execution Plan для каждой инструкции будет создавать план ее выполнения.

После обработки и преобразования собранных данных исходная информация для решения задачи оптимального распределения данных будет представлена в виде табл. 2. В случае обновления временем обработки и передачи ответа можно пренебречь, так как оно невелико и не окажет значительного влияния на процесс моделирования и оптимизации. То же касается и объема передаваемого ответа о подтверждении обновления. Поэтому поля (завершение обработки, завершение выполнения, объем 2) при добавлении информации об обновлении будут заполняться автоматически, а не определяться комплексом программных средств. В случае выполнения запроса вся приведенная информация будет определяться программно.

Таким образом, автоматизировав процесс сбора статистики реальной РБД с помощью комплекса программных средств, удается получить наиболее точную входную информацию для моделирования работы распределенной базы данных и дальнейшего нахождения оптимального распределения данных.

Таблица 2 — Таблица статистики работы распределенной базы данных

Название поля	Описание
Идентификатор запроса	Идентификатор вызываемого запроса и обновления
Тип запроса	Тип выполняемого запроса: обновление или запрос
Узел 1	Имя узла, на котором инициирован запрос
Узел 2	Имя узла, к которому обращен запрос
Начало выполнения	Время начала выполнения запроса на узле 1
Начало обработки	Время начала обработки распределенного запроса на узле 2
Завершение обработки	Время завершения обработки распределенного запроса на узле 2
Завершение выполнения	Время завершения выполнения запроса на узле 1
Объем 1	Объем данных, переданных с узла 1 на узел 2 для выполнения запроса
Объем 2	Объем ответа на запрос, передаваемого с узла 2 на узел 1

Перечень ссылок

1. Дейт К.Дж. Введение в системы баз данных, 8-е издание.: Пер. с англ. — М.: Издательский дом «Вильямс», 2005. — 1328 с.

2. Телятников А.О. Разработка объектной модели распределенной базы данных // Наукові праці ДонНТУ. Випуск 74. — Донецьк: ДонНТУ, 2004. — С. 192–200.

3. Лаздынь С.В., Телятников А.О. Повышение эффективности распределенных баз данных с использованием объектно-ориентированного моделирования и генетических алгоритмов // Единое информационное пространство: Сб. докл. Междунар. Научно-практич. конф. — Днепропетровск: ИПК ИнКомЦентра УГХТУ, 2003. — С. 23–26.

4. Оутей М., Конте П. Эффективная работа: SQL Server 2000. — СПб.: Питер; К.: Издательская группа BHV, 2002. — 992 с.