

4. Kouvatso D. D. A maximum entropy analysis of the G/G/1 queue at equilibrium // J. Opl. Res. Soc. 1988. Vol. 39, N 2. P. 183—200.

5. Клейнрок Л. Теория массового обслуживания. М.: Машиностроение, 1979. 432 с.

Рекомендована кафедрой
электронной вычислительной техники

Поступила в редакцию
02.07.07 г.

УДК 004.4'22, 004.65

В. Б. НОВОСЕЛЬСКИЙ

*Санкт-Петербургский государственный университет
информационных технологий, механики и оптики*

МЕТОД ПРОЕКТИРОВАНИЯ ПРОЦЕССА РАСПРЕДЕЛЕНИЯ РЕЛЯЦИОННОЙ БАЗЫ ДАННЫХ

Рассматривается вопрос распределения реляционной базы данных по узлам вычислительной сети. Предлагается генетический алгоритм решения задачи проектирования, позволяющий учесть взаимозависимость схем фрагментации данных, размещения фрагментов и стратегий исполнения запросов. Исследуются качество получаемых решений и вычислительная сложность алгоритма.

Технологии баз данных (БД) в настоящее время используются практически во всех организациях. Все большую значимость приобретают процессы децентрализации, требующие создания приложений, доступ к которым осуществляется из различных географических мест. Увеличиваются требования к оперативности и достоверности информации. Задачи информационной интеграции БД и проектирования географически распределенных БД (РБД) являются наиболее актуальными для разработчиков программного обеспечения в течение почти трех десятилетий.

Большинство авторов под проектированием РБД понимают фрагментацию и размещение, т.е. разбиение БД на фрагменты и принятие решения о том, где будут храниться эти фрагменты. Однако проектирование схем фрагментации и размещения отношений реляционной базы данных основывается на информации о способах и методах использования РБД. Эти методы зависят от стратегии исполнения запросов, которая, в свою очередь, должна учитывать схемы фрагментации и размещения. Следовательно, проектирование схем фрагментации и размещения, а также определение стратегии исполнения запросов должны производиться одновременно. Таким образом, задачу проектирования РБД следует формулировать так: для заданной логической схемы БД, множества запросов и конфигурации вычислительной сети (ВС) описать схему фрагментации, схему размещения фрагментов и сформировать стратегии исполнения каждого запроса таким образом, чтобы оптимизировать целевую функцию.

Все три задачи, входящие в процесс проектирования РБД, NP-полные [1, 2], т.е. с ростом размерности задач их вычислительная сложность растет экспоненциально. Во многих работах (например, [3—5]) отмечается, что для решения задач кластеризации и компоновки наиболее успешно применяются генетические алгоритмы.

В основу генетических алгоритмов положены идеи естественного отбора в биологических популяциях. В таких алгоритмах любое решение задачи синтеза представляется особями, которые характеризуются хромосомами. Хромосомы состоят из генов, значениями генов являются значения проектных параметров. Направленный перебор решений осуществляется с помощью генетических операторов выбора родителей, скрещивания, мутации, селекции,

переупорядочения. Особи оцениваются с использованием функции приспособленности, определяющей, насколько хорошо особь подходит для решения данной задачи.

Помимо NP-полноты, задачи проектирования РБД обладают свойством взаимозависимости, т.е. входными данными для следующей задачи является решение предыдущей. Таким образом, для получения проекта РБД, близкого к оптимальному, целесообразно применять вложенные генетические алгоритмы.

В соответствии с разработанным автором настоящей статьи методом для каждой хромосомы из внешнего алгоритма, который сформирован схемами фрагментации и размещения, запускается вложенный генетический алгоритм, предназначенный для формирования оптимальной стратегии исполнения запросов. Хромосомы вложенного алгоритма представляют собой решения, принимаемые при исполнении запроса. Комбинация хромосом из внешнего и вложенного алгоритмов полностью определяет решение поставленной задачи.

Сначала необходимо сформировать пространство поиска схем фрагментации, размещения и стратегии исполнения запросов, затем схемы должны быть представлены в виде хромосом. Пространство поиска схемы фрагментации формируется разбиением отношений на минимальные, не делимые далее, фрагменты. Для получения такого набора фрагментов для каждого отношения R на основании транзакций определяются набор минтермов предикатов и группы атрибутов. Разбиение на предикаты характеризует горизонтальную фрагментацию, а группы атрибутов — вертикальную.

Схема размещения кодируется бинарной матрицей D , строками которой являются узлы ВС, а столбцами — сформированный набор фрагментов. Единица в ячейке матрицы означает наличие определенного фрагмента в узле ВС. Для обеспечения полноты схемы размещения каждый фрагмент должен находиться хотя бы в одном узле (в каждом столбце должна быть хотя бы одна единица).

При исполнении реляционного запроса необходимо принять следующие решения.

1. В какой последовательности производить операции соединения отношений.
2. В каком узле производить операцию соединения.

3. Применять ли технику semi-join (неполное соединение) [6] при соединении. Эта техника используется при соединении отношений, находящихся в разных узлах: пусть отношение A находится в узле 1 и отношение B находится в узле 2; принцип semi-join заключается в отправке на узел 2 столбцов отношения A , по которым осуществляется соединение, нахождение и отправка на узел 1 только тех записей из B , которые удовлетворяют условию соединения.

4. Применять ли технику double-pipelined hash-join (двойное конвейерное хэш-соединение) [7] при соединении. Идея double-pipelined hash-join состоит в формировании двух хэш-таблиц Ha и Hb , после чего записи из A и B обрабатываются по одной в каждый момент времени. Для записи из A в таблице Hb ищутся соответствующие ей вложения, после чего запись из A и найденные записи из B возвращаются как результаты соединения, и запись из A помещается в хэш-таблицу Ha . Далее, берется запись из B и обрабатывается аналогичным образом.

5. Какие узлы сети, содержащие необходимые фрагменты, использовать при внутриоператорном параллелизме.

Функцией приспособленности вложенного алгоритма является критерий оптимальности РБД. Для всех запросов производится расчет сокращения времени исполнения за счет применения внутриоператорного параллелизма и расчет коэффициентов использования ресурсов.

После останова вложенного алгоритма значение функции приспособленности лучшей хромосомы используется как оценка функции приспособленности внешнего алгоритма. Критерием останова алгоритмов является схождение популяции.

Для оценки качества предложенного генетического алгоритма был проведен эксперимент, в котором решения, найденные с помощью этого алгоритма, сравнивались с оптимальными. Использовался следующий критерий эффективности БД:

$$\xi = \sum_{i=1}^k \sum_{j=0}^n (c_t Q_{ij} + c_a W_{ij}) f_{ij} \rightarrow \min,$$

где k — количество узлов ВС; n — количество запросов; f_{ij} — частота возникновения j -го запроса в i -м узле; Q_{ij} — временной коэффициент j -го запроса, порожденного в i -м узле; W_{ij} — коэффициент использования ресурсов при обработке j -го запроса, порожденного в i -м узле; c_t и c_a — коэффициенты, определяющие приоритеты времени ответа и готовности транзакции, значения c_t и c_a находятся в интервале $[0 \dots 1]$ и определяются проектировщиком.

Временной коэффициент j -го запроса определяется как $Q_{ij} = T_{ij}^d / T_{ij}^l$, т.е. равен отношению времени ответа на запрос, исполненный в РБД, к расчетному времени ответа на запрос, исполненный локально в узле i , при условии наличия в узле всех необходимых фрагментов. Коэффициент использования ресурсов W_{ij} равен сумме коэффициентов использования центрального процессора и внешнего запоминающего устройства узлов, вовлеченных в операцию, и коэффициента использования сетевых ресурсов.

Эксперимент проводился для следующей вычислительной системы: ВС состоит из трех узлов, равноудаленных друг от друга; логическая схема БД имеет 2 отношения; производится 3 запроса на выборку и 2 запроса на обновление данных, которые формируют 4 группы атрибутов и 5 предикатов.

По результатам измерений были определены оптимальные значения параметров генетических алгоритмов, приведенные в таблице. Запуск алгоритма с другими значениями параметров ухудшает результат или увеличивает время работы.

Параметр	Алгоритм	
	внешний	вложенный
Размер популяции, особь	25—35	100—110
Вероятность скрещивания, %	45—70	55—65
Вероятность мутации, %	0,75—1,00	0,75—1,00

Схождение популяций, при котором 90 % лучших особей имеют расхождение по значению функции приспособленности не более, чем на 10 %, наблюдалось примерно спустя 500—700 поколений внешнего генетического алгоритма и 2500—3000 поколений вложенного алгоритма. Отметим, что для различных значений c_t и c_a удалось найти оптимальное решение, полученное полным перебором. Кроме этого, с увеличением пространства возможных решений (количества фрагментов и узлов ВС) время работы генетического алгоритма увеличивается квадратично, тогда как поиск оптимального решения путем полного перебора растет экспоненциально.

В дальнейшем необходимо исследовать зависимость скорости схождения популяций от параметров генетического алгоритма.

Разработанная прикладная программа, реализующая предложенный алгоритм, может быть положена в основу создания САПР распределенной базы данных.

СПИСОК ЛИТЕРАТУРЫ

1. Ahmad I., Karlapalem K., Kwok Y.K. et al. Evolutionary algorithms for allocating data in distributed database systems // Distributed and Parallel Databases. 2002. Vol. 11, N 1. P. 5—32.
2. Apers P. M. G. Data allocation in distributed database systems // ACM Trans. on Database Systems. 1988. Vol. 13, N 3. P. 263—304.
3. Rho S., March S. T. Optimizing distributed join queries: A genetic algorithm approach // Annals of Operations Research. 1997. Vol. 71. P. 199—228.

4. Wang J.-C., Horng J.-T., Hsu Y.-M. et al. A genetic algorithm for set query optimization in distributed database systems // Proc. of the IEEE Intern. Conf. on Systems, Man and Cybernetics. Beijing, China. 1996. P. 1977—1982.
5. Норенков И. П. Эвристики и их комбинации в генетических методах дискретной оптимизации // Информационные технологии. 1999. № 1. С. 2—7.
6. Bernstein P. A., Goodman N., Wong E. et al. Query processing in a system for distributed databases (SDD-1) // ACM Trans. on Database Systems. 1981. Vol. 6, N 4. P. 602—625.
7. Wilschut A. N., Apers P. M. G. Dataflow query execution in a parallel main-memory environment // Distributed and Parallel Databases. 1993. Vol. 1, N 1. P. 103—128.

Рекомендована кафедрой
информатики и прикладной
математики

Поступила в редакцию
04.04.08 г.