

К ВОПРОСУ ОБ ОПТИМИЗАЦИИ БАЗ ДАННЫХ СЛОЖНЫХ ИНФОРМАЦИОННЫХ СИСТЕМ

А.В. Майстренко, Н.В. Майстренко, О.И. Ерохин

ГОУ ВПО «Тамбовский государственный технический университет», г. Тамбов

Ключевые слова и фразы: база данных; взвешенный неориентированный граф; информационная система; модель «сущность-связь»; CASE-технология.

Аннотация: Анализируется возможность автоматизации проектирования сложных информационных систем. Рассматриваются постановка и алгоритм решения задачи разбиения баз данных информационной системы на физическом уровне с целью оптимизации времени доступа к данным.

Современные информационные системы (ИС) характеризуются сложной структурой. К настоящему времени разработано множество методик для качественного оптимального проектирования ИС с использованием CASE-средств, позволяющих обеспечить модульность разработки, разделив ее на параллельные процессы и минимизировав при этом количество ошибок проектирования.

Обычно к CASE-средствам относят любое программное средство, автоматизирующее ту или иную совокупность процессов жизненного цикла программного обеспечения (ПО) и обладающее рядом характерных особенностей:

- мощными графическими средствами для описания и документирования ИС, обеспечивающими удобный интерфейс с разработчиком и развивающими его творческие возможности;
- интеграцией отдельных компонент CASE-средств, обеспечивающей управляемость процессом разработки ИС;
- использованием специальным образом организованного хранилища проектных метаданных.

Интегрированное CASE-средство (или комплекс средств, поддерживающих полный жизненный цикл программного обеспечения) содержит следующие компоненты [2]:

- репозиторий, являющийся основой CASE-средства. Он должен обеспечивать хранение версий проекта и его отдельных компонентов, синхронизацию поступления информации от различных разработчиков при групповой разработке, контроль метаданных на полноту и непротиворечивость;
- графические средства анализа и проектирования, обеспечивающие создание и редактирование иерархически связанных диаграмм (DFD, ERD и др.), образующих модели ИС;
- средства разработки приложений, включая языки 4GL и генераторы кодов;
- средства конфигурационного управления;
- средства документирования;
- средства тестирования;
- средства управления проектом;
- средства реинжиниринга.

Кроме перечисленных компонентов обязательным требованием к современным CASE-средствам является возможность обработки с их помощью моделей предметной области ИС и построения на ее основе баз данных.

Как правило, на практике при проектировании ИС используется несколько моделей предметной области. Обобщенная (понятийная) модель и модель, представленная схемой базы данных, могут дополняться рядом промежуточных.

Размер баз данных и их файлов, а также их количество – один из наиболее проблемных вопросов, возникающих при проектировании, т.к. зачастую размер баз данных в ИС настолько велик, что время доступа к данным, хранящимся в них, может быть недопустимо большим. Часто для сокращения времени доступа к необходимым данным базы данных разбивают по нескольким файлам в соответствии со статистикой использования тех или иных атрибутов баз данных.

Разбиение содержимого базы данных по отдельным файлам – сложная задача, формализуемая и решаемая при помощи дискретной математики и теории графов [1]. Представим базу данных, содержащую N атрибутов. Вес i атрибута $W_i = l_i \cdot m$, где l_i – длина в байтах значения атрибута, m – число экземпляров, $i = 1, 2, \dots, N$.

Атрибуты и связи между ними можно представить в виде взвешенного ненаправленного графа, причем каждая вершина графа имеет вес d_i и каждое ребро имеет вес w_{ij} . Вес ребра отображает частоту совместного использования данной пары атрибутов в приложениях. Задача оптимизации построения БД заключается в построении файлов (таблиц) БД таким образом, чтобы сумма весов вершин взвешенного ненаправленного графа не превосходила допустимого объема, а сумма весов разрезанных ребер была наименьшей. Таким образом, исходная задача принадлежит классу задач декомпозиции графа.

Пусть задан неориентированный взвешенный граф $G(X, V, w, d)$ порядка n , где $X = \{x_1, \dots, x_n\}$ – множество вершин; $V \subseteq X \times X$ – множество ребер; $w: V \rightarrow R^+$ – отображение, определяющее вес каждого ребра; $d: X \rightarrow R^+$ – отображение, определяющее вес каждой вершины, где R^+ – множество действительных неотрицательных чисел.

Требуется определить разбиение множества вершин X графа $G(X, V, w, d)$ на k подмножеств $(X_1, X_2, \dots, X_k)$ таким образом, чтобы для кусков графа $G_1(X_1, V_1, w_1, d_1)$, $G_2(X_2, V_2, w_2, d_2)$, ..., $G_k(X_k, V_k, w_k, d_k)$ выполнялись требования:

$$\begin{aligned} X_i \cap X_j &= \emptyset, \text{ для } \forall i \neq j, \text{ где } i, j = \overline{1, k}; \\ \bigcup_{i=1}^k X_i &= X; \\ |X_1| &= n_1, \dots, |X_k| = n_k, n_1 + \dots + n_k = n. \end{aligned} \quad (1)$$

В качестве первого критерия оптимальности F_1 , определяющего эффективность k -разбиения $(X_1, \dots, X_k)$, будем рассматривать суммарный вес всех ребер, целиком попавших в один из подграфов разбиения:

$$F_1(X_1, \dots, X_k) = \frac{1}{2} \sum_{i=1}^k \sum_{x_a \in X_i} \sum_{\substack{x_b \in X_i \\ x_a \neq x_b}} w(x_a, x_b) \quad (2)$$

Суммарный вес вершин, принадлежащих одному подграфу, будем называть весом подграфа. Вес некоторого подграфа X_i будем обозначать через $W(X_i)$. Для уравнивания весов получаемых подграфов будем использовать следующий критерий оптимальности F_2 :

$$F_2(X_1, \dots, X_k) = \min_{i=1, k} W(X_i) = \min_{i=1, k} \left\{ \sum_{x_a \in X_i} d(x_a) \right\} \quad (3)$$

Таким образом, имеем задача оптимизации БД может быть записана в виде:

$$\begin{cases} \max_{(X_1, \dots, X_k) \in D} F_1(X_1, \dots, X_k), \\ \max_{(X_1, \dots, X_k) \in D} F_2(X_1, \dots, X_k) \end{cases} \quad (4)$$

Для решения задач разбиения графов используют, как правило, алгоритм разбиения пополам, алгоритм деления с учетом связности, алгоритм разбиения Кернигана-Лина и их модификации. Результаты сравнительного анализа эффективности перечисленных алгоритмов приведены в табл. 1.

Сравнительный анализ алгоритмов разбиения графов

Алгоритм	Точность	Время выполнения	Возможность распараллеливания
Рекурсивный метод деления пополам	..	.	...
Деление с учетом связности	..	..	..
KL-алгоритм	...	...	...
Примечание. . – плохо; .. – хорошо; ... – очень хорошо.			

Как видно из табл. 1, наибольшей эффективностью обладает алгоритм Кернигана-Лина [3], обладающий лучшим показателем точности, сходимости, возможностью распараллеливания, а также относительной простотой в реализации. Данный алгоритм основан на понятии веса – величины, которая определяет выигрыш от перемещения вершины из одного подмножества в другое. Вес рассчитывается для каждой вершины как количество соединений вершины с другим подмножеством минус количество соединений с подмножеством, в котором вершина находится. Пока есть вершины с положительным весом, алгоритм меняет местами вершины с максимальным весом с вершинами из другого подмножества.

Суть алгоритма Кернигана-Лина состоит в обмене вершинами между подмножествами имеющегося разбиения графа. То есть предполагается, что некоторое начальное разбиение графа уже существует, затем имеющееся приближение улучшается в течение некоторого количества итераций. Используем в качестве начального простейшее разбиение графа на две части без учета каких-либо условий.

На первом шаге алгоритма формируется множество пар вершин для перестановки. Из вершин, которые еще не были переставлены на текущей итерации, формируются все возможные пары. На втором – выполняется построение новых вариантов разбиения графа. Ценность каждого разбиения равна сумме весов дуг внутри кластера разбиения. Каждая пара, подготовленная на шаге 1, поочередно используется для обмена вершин между частями имеющегося разбиения графа для получения множества новых вариантов деления. Для сформированного на шаге 2 множества новых делений графа выбирается лучший вариант (шаг 3). Этот вариант далее фиксируется как новое текущее разбиение графа, а соответствующая выбранному варианту пара вершин отмечается как использованная на текущей итерации алгоритма. На четвертом шаге проверяется наличие в графе вершин, еще не использованных при перестановках, выполнение итерации алгоритма снова продолжается с шага 1. Если же перебор вершин графа завершен, далее следует шаг 5. Среди всех разбиений графа, полученных на шаге 3 проведенных итераций, выбирается (и фиксируется) наилучший вариант разбиения графа.

Именно этот алгоритм предполагается реализовать для оптимизации структуры баз данных при разработке ИС моделирования и проектирования сложных химико-технологических систем.

Список литературы

1. Бурков, В.Н. Теория графов в управлении организационными системами / В.Н. Бурков, А.Ю. Заложнев, Д.А. Новиков. – М. : Синтег, 2001. – 124 с.
2. Федоров, Н.В. Проектирование информационных систем на основе современных CASE-технологий / Н.В. Федоров. – М. : МГИУ, 2008. – 287 с.
3. Schloegel, K. Graph Partitioning for High Performance Scientific Simulations / K. Schloegel, G. Karypis, V. Kumar. – 2000.

About Optimization of Complex Information System Database

A.V. Maistrenko, N.V. Maistrenko, O.I. Erokhin

Tambov State Technical University, Tambov

Key words and phrases: database; scaled undirected graph; information system; «essence-link» model; CASE-technology.

Abstract: The paper analyzes the possibility of automation of designing complex information systems. The problem setting and the algorithm for splitting the information system database on the physical level in order to optimize the time of data access are studied.

© А.В. Майстренко, Н.В. Майстренко, О.И. Ерохин, 2009