

охлаждающих жидкостей, использована на противопожарные нужды, для мойки оборудования, полов, первичной промывки деталей после нанесения гальванопокрытий, а также при приготовлении технологических растворов травления и обезжиривания сталей, малочувствительных к избыточному содержанию  $Cl^-$  и  $SO_4^{2-}$  - ионов.

Что касается утилизации образующихся шламов, следует прежде всего отметить недостаточный опыт утилизации сульфидных шламов. Гидроксидный шлак, образующийся после обезвреживания цианидов, который представлен главным образом гидроксидом кадмия (~75%), можно использовать на специализированных предприятиях по получению солей кадмия. Остальное количество гидроксидного шлака и железоокисный шлак могут быть переработаны как отдельно, так и совместно. При отдельной переработке гидроксидный шлак, обычно содержащий ~60% воды, должен быть высушен и может найти применение в качестве замены минерального порошка при производстве асфальтобетонных смесей для строительства дорог. Железоокисный шлак с небольшим количеством гидроксидов тяжелых металлов может быть использован для получения железоокисного пигмента желтого цвета. Гидроксидный шлак, смешанный с железоокисным шлаком (влажность ~60%) является хорошей вспучивающей добавкой при производстве керамзита.

Таким образом, повышение эффективности очистки воды оптимизацией гидроксидного метода и адсорбцией на гидроксиде железа (III) снижает отрицательное воздействие на окружающую среду, уменьшает водопотребление и позволяет получить ряд продуктов, имеющих большое значение в промышленности.

### ОПТИМИЗАЦИЯ АЛЬТЕРНАТИВНЫХ СОЕДИНЕНИЙ В ЗАПРОСАХ РЕЛЯЦИОННЫХ СИСТЕМ

Погодаев А.К., Муравейко А.Ю., Дятчина Д.В.

*Липецкий государственный  
технический университет,  
Липецк*

Существующие подходы оптимизации запросов предполагают инвариантную схему соединения таблиц [1]. Однако, в базах данных (БД) сложных структур при динамичном изменении объема таблиц ранее запланированные варианты операций соединения с течением времени могут оказаться не оптимальными в плане скорости их выполнения.

Время выполнения запроса можно представить в виде формулы:

$$t = \sum_{i=1}^n \left( [t_{откр\_i} + t_{закр\_i} + \frac{V_i * t_{блока}}{V_{блока}}] * x_i \right) + t_{соед},$$

где  $x_i=1$ , если  $i$ -ая таблица, принадлежит запросу; 0 – иначе;  $n$  – количество таблиц;  $V_{блока}$  – объем блока;

$V_i$  – объем  $i$ -й таблицы;  $t_{откр\_i}$  – время открытия  $i$ -й

таблицы;  $t_{закр\_i}$  – время закрытия  $i$ -й таблицы;  $t_{блока}$  – время чтения блока;  $t_{соед}$  – общее время выполнения операций соединения.

Для выбора оптимального маршрута соединения таблиц из нескольких семантически альтернативных, представим схему БД в виде графа, выполнив переход от таблиц к вершинам и от связей к дугам. Каждой вершине графа сопоставим нагрузку  $e_i$  – время доступа и чтения таблицы, каждой дуге сопоставим нагрузку  $p_j$  – время на соединение инцидентных ей таблиц. Таким образом, для выбора оптимального маршрута соединения необходимо решить задачу оптимизации на графе с нагруженными вершинами и дугами.

Задача оптимизации на графе состоит в выборе минимально нагруженного подграфа при условии, что результирующий подграф является связным:

$$f(G) = \sum_{i=1}^n x_i * e_i + \sum_{j=1}^m y_j * p_j \rightarrow \min \quad (1)$$

где  $e_i = t_{откр\_i} + t_{закр\_i} + t_{блока} \frac{V_i}{V_{блока}}$ ,  $e_i$  – нагрузка на  $i$ -ю вершину;  $x_i = 1$ , если  $i$ -ая вершина, принадлежит подграфу, 0 – иначе;  $n$  – количество вершин;

$y_j = 1$ , если  $j$ -ая дуга принадлежит подграфу, 0 – иначе;  $m$  – количество дуг;  $p_j$  – нагрузка на  $j$ -ю дугу.

Для задачи (1) существуют методы решения (например [2]), но они ограничены определенной предметной областью и специфической структурой графа. Поэтому для случая, когда граф имеет произвольную структуру, разработан следующий алгоритм оптимизации на графе.

В основе данного алгоритма используется поиск на графе в ширину, модифицированный для учета суммарной нагрузки на вершинах и дугах маршрута достижения искомой цели. Кроме того, кратчайший путь находится между несколькими отмеченными вершинами. В результате работы данного алгоритма получается минимальный маршрут, соединяющий все отмеченные вершины, т.е. те, которые используются в запросе.

Выводы: разработаны методика выбора оптимального маршрута соединения таблиц в БД, имеющих сложную структуру организации данных; алгоритм поиска оптимального маршрута соединения отмеченных вершин на графе, имеющем циклы, с нагруженными вершинами и дугами.

### СПИСОК ЛИТЕРАТУРЫ

1. Гарсия-Молина Г., Ульман Д., Уидом Д. Системы баз данных. Полный курс. Пер. с англ.- М.: Издательский дом «Вильямс», 2003 – 1088 с.
2. Погодаев А.К., Анненков А.В. Метод оптимизации графов с нагруженными вершинами // Вестник ЛГТУ – ЛЕГИ 2001 №1(7) – 37-39с.