

Инкапсуляция параллелизма в СУБД с открытым исходным кодом

А.В. Лепихов

По своим функциональным возможностям объектно-реляционная СУБД PostgreSQL стоит в одном ряду с такими коммерческими продуктами, как InterBase, SyBase и MS SQL Sever. СУБД PostgreSQL распространяется под лицензией BSD, что позволяет разработчикам выпускать коммерческие версии PostgreSQL под своим именем и осуществлять коммерческую поддержку [1]. Данная СУБД поддерживает инструменты для работы с очень большими базами данных и зачастую используется при организации терабайтных хранилищ данных.

На сегодняшний день в PostgreSQL обеспечена возможность масштабирования на вычислительных системах с общей памятью. Известны коммерческие модификации PostgreSQL для кластерных систем (см. например [3]). Однако данные системы основываются на синхронной репликации, осуществляют перераспределение запросов между вычислительными узлами и не имеют механизмов параллельной обработки запросов.

Параллельная обработка запросов осуществляется на основе фрагментации всех отношений базы данных по узлам вычислительного кластера. Все операции с данными, такие как сканирование таблиц и индексов, проекция, выборка, соединение и пр. выполняются независимо на каждом узле. Параллелизм операций инкапсулируется в специальном операторе *exchange* [2], который осуществляет обмен данными между вычислительными узлами в процессе обработки запроса. Оператор *exchange* реализуется на основе использования стандартного узла в последовательном плане запроса и может быть добавлен в качестве узла в любое место дерева запроса. Оператор *exchange* имеет два специальных параметра, определяемых пользователем: номер *порта обмена* и указатель на *функцию распределения*. Функция распределения для каждого кортежа вычисляет логический номер процессорного узла, на котором данный кортеж должен быть обработан. Параметр «порт обмена» позволяет включать в дерево запроса произвольное количество операторов *exchange*.

Проблема балансировки загрузки решается при помощи алгоритма балансировки, основанного на методе частичного зеркалирования [4]. Данный метод предполагает использование горизонтальной фрагментации отношений и частичного зеркалирования дисков по узлам и уровням вычислительного кластера.

Предлагаемую нами схему работы параллельной модификации PostgreSQL для кластерных систем можно описать следующим образом. Идентичные экземпляры СУБД PostgreSQL запускаются на всех узлах кластера. Один из узлов выбирается в качестве ведущего, остальные узлы считаются ведомыми. Ведущий узел получает команды непосредственно от клиентского приложения. После инициализации клиентского соединения, ведущий узел начинает работать в следующем режиме. Получая от клиента команду на выполнение запроса или транзакции, ведущий узел пересылает эту команду на все ведомые узлы, после чего начинает обрабатывать полученный запрос над своей базой данных. После обработки запроса над своей базой данных ведущий узел принимает результаты обработки запроса с остальных узлов и передает их клиентскому приложению. Ведомые узлы обрабатывают запрос над своей базой данных и передают результат ведущему узлу.

Реализация описанных выше методов в PostgreSQL позволит создать дешевую СУБД способную работать на вычислительных системах с неразделяемыми ресурсами.

Литература

1. Официальный сайт PostgreSQL. [<http://www.PostgreSQL.org>]
2. Соколинский Л.Б. Организация параллельного выполнения запросов в многопроцессорной машине баз данных с иерархической архитектурой // Программирование. –2001. –№. 6. –С. 13–29.
3. Continuent portal. [<http://www.continuent.com>]
4. Лепихов А.В., Соколинский Л.Б. Стратегия размещения данных в многопроцессорных системах с симметричной иерархической архитектурой // Научный сервис в сети Интернет: технологии параллельного программирования: Труды Всероссийск. науч. конф. (18-23 сентября 2006 г., г. Новороссийск). –М.: Изд-во МГУ. –2006. –С. 39–42.