

УДК 004

СПОСОБЫ ОПТИМИЗАЦИИ АККУМУЛИРОВАНИЯ ИЕРАРХИЧЕСКИХ СУЩНОСТЕЙ В БАЗЕ ДАННЫХ

А.С. Оту

Научный руководитель: А.А. Напрюшкин, к.т.н., инженер кафедры ИКСУ ИК ТПУ

Национальный исследовательский Томский политехнический университет

E-mail: eld0727@mail.com

This article is about three ways of solving problem of accumulating hierarchical data in SQL database. Those methods were designed to make general accumulating calculation over hierarchical big data for analytic software for company Ltd. "SamaraNIPIneft".

Keyword: SQL database, bid data, hierarchical data, data cache.

Ключевые слова: SQL база данных, большие данные, иерархические данные, кеширование данных.

Введение

На сегодняшний день существует огромное количество областей, где требуется анализ данных. Причем эти данные устроены так, что сущности, которые нужно анализировать разбиваются на более мелкие части, которые в свою очередь разбиваются еще на более мелкие части, и этот процесс может продолжаться очень долго, образуя иерархию данных. И основной проблемой при работе с этой иерархией является, количество записей в базе данных, которые нужно обработать, чтобы получить статистические данные для верхних уровней.

Для решения этой задачи были рассмотрены три способа решения этой проблемы.

Описание проблемы

Рассмотрим нашу проблему на примере этой простой схемы базы данных:

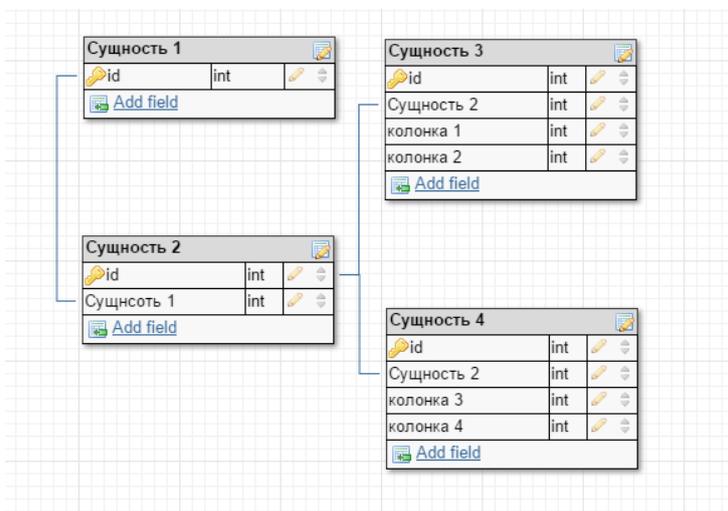


Рис. 1. Схема базы данных

В этой схеме «Сущность 1» разбивается на несколько записей «Сущность 2» в свою очередь «Сущность 2» разбивается на «Сущность 3» и «Сущность 4». Итак, наша задача собрать общую информацию из колонок «колонка 1» и «колонка 2» таблицы «Сущность 3» и «колонка 3», «колонка 4» из «Сущность 4». Даже для этой схемы если в таблицах «Сущность 3» и «Сущность 4» к каждой «Сущность 2» будет привязано по несколько тысяч записей, и в свою очередь «Сущность 1» будет разбита на одну тысячу записей «Сущность 2» то полу-

читаться, что нужно будет собрать данные из $1000 \times (2000 + 2000) = 4000000$ записей на одну строчку «Сущность 1». И как можно предположить этот процесс займет заметное время даже для одной записи. И для решения этой проблемы существует несколько способов.

Увеличение количества серверов баз данных

Одним из самых очевидных способов является увеличение производственных мощностей, за счет увеличения серверов баз данных. Приложение будет обращаться за разными частями данных к разным серверам. Этот метод называется Map-Reduce, который предполагает сбор информации с разных источников с последующим ее суммированием. Плюсом этого подхода является линейное увеличение скорости чтения данных в зависимости от количества серверов.

Минусы:

- Стоимость закупки и поддержки серверов базы данных
- Возможные проблемы сети, что может увеличить время ожидания данных
- Необходимость делать репликацию данных, чтобы поддерживать актуальность данных на всех серверах.

Кэширование данных

Следующим способом ускорения получения информации является кэширование данных. На сегодняшний день кэширование на уровне базы данных предлагают все популярные СУБД. Плюсом этого подхода будет увеличение скорости чтения данных, и меньшая стоимость в сравнении с закупкой серверов. Минусами является увеличение скорости времени записи в базу данных, за счет того, что требуется поддерживать целостность кэша. Также можно использовать кэширование посчитанных данных на стороне клиента, это позволит разгрузить сервера баз данных. Но при этом увеличится объем памяти, который потребляет приложение, и увеличивается сложность разработки за счет того, что нужно следить за состоянием кэша, чтобы он не содержал устаревших данных.

Агрегация данных вверх по иерархии

Одним из наиболее эффективных способов ускорения аккумулирования иерархических данных является агрегация необходимых данных вверх по иерархии:

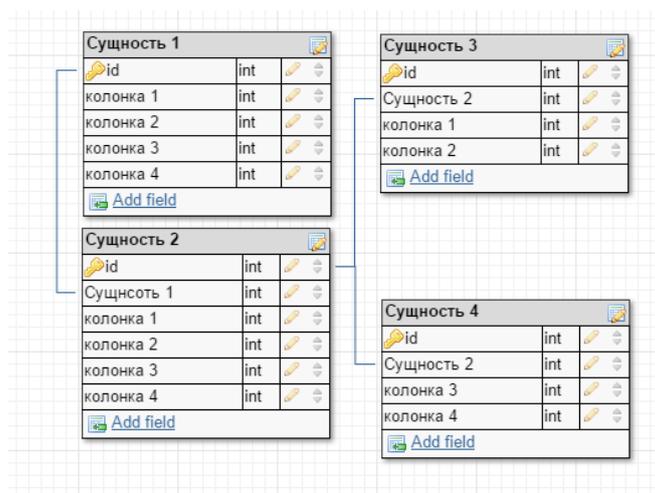


Рис. 2. Агрегация данных вверх по иерархии

При данном способе можно получить доступ к любым данным за одну операцию чтения из базы, что позволяет существенно ускорить этот процесс. Но появляется огромная

сложность поддержания консистентности данных на всех уровнях иерархии, которая ложится на разработчика.

Заключение

Проблема работы с большими данными существовали всегда, с момента появления серверов, способных хранить и обрабатывать их. Но при должном проектировании базы данных и возможности применения способов оптимизации работы с базами данных, можно избежать большего количества проблем, связанных с иерархичными большими данными.

УДК 004

СРАВНЕНИЕ ЭФФЕКТИВНОСТИ ПИД-РЕГУЛИРОВАНИЯ И НЕЧЕТКОГО РЕГУЛИРОВАНИЯ В СИСТЕМЕ УПРАВЛЕНИЯ ПРОЦЕССОМ РЕКТИФИКАЦИИ НЕФТИ

А.А. Наумовская

Научный руководитель: В.А. Рудницкий, доцент кафедры ИКСУ ИК ТПУ

E-mail: nastyush.naumovskya.1994@mail.ru

In given article author consider an efficiency of use of Fuzzy-regulators in relation to classical PID-controllers in a control system of fuel consumption in a distillation column.

Keywords: PID-controller, fuzzy-logic controller, distillation column.

Ключевые слова: ПИД-регулятор, нечёткий регулятор, ректификационная колонна.

Введение

В химической, нефтехимической и других отраслях промышленности часто возникает необходимость разделения смесей на отдельные составляющие. Наиболее ярким примером является необходимость разделения нефтепродуктов на отдельные фракции в нефтехимической промышленности. Этот процесс называется ректификацией и производится в специальных аппаратах, называемых ректификационными колоннами.

Разработка систем автоматического управления процессом ректификации нефти является сложной задачей, так как математическая модель ректификационной колонны [1] является нелинейной, содержит много эмпирических коэффициентов, изменяющихся в большом диапазоне и вызывающих большие затруднения при идентификации. В процессе управления нелинейными и сложными системами достаточно трудоемко осуществлять настройку ПИД-регулятора. Сложный процесс настройки приводит к ухудшению качества получаемого топлива [2]. Решением этой проблемы стало применение нечетких регуляторов в таких системах.

Моделирование процесса ректификации нефти

Пусть имеется контур автоматического регулирования фазового состава сырой нефти, который позволит сравнить эффективность классического регулирования и нечеткого регулирования (рис. 1). Исполнительный механизм данной системы – это клапан, расположенный на линии подачи топлива в печь. Звено чистого запаздывания характеризует транспортные задержки в трубопроводе реальной системы. Печь характеризуется передаточной функцией инерционного звена первого порядка. В качестве задания для регулятора будет величина, характеризующая перепад давления на прямом вертикальном участке трубопровода подачи нефти из печи в колонну, что фактически отражает соотношение газовой и жидкой фаз в нагретой нефти [2]. С целью упрощения анализа на вход подадим сигнал в виде единичного ступенчатого воздействия. Найдем коэффициенты ПИД-регулятора методом Циглера-Никольса и получим следующие значения: $K_p = 1,215$, $K_i = 0,81$, $K_d = 0,456$. Для настройки