Математические вопросы кибернетики и вычислительной техники 30, 105--111, 2008.

Разработка пользовательского интерфейса управления распределенными базами данных в кластерных системах

Г. Петросян

Институт проблем информатики и автоматизации НАН РА
e-mail: gurgen@sci.am

Аннотация

В статье описывается пользовательский интерфейс, разработанный для управления распределенными базами данных в кластерных системах. Предлагается двухуровневая модель распределенной обработки баз данных для нескольких кластеров с использованием созданного интерфейса.

1. Введение

В последнее время в различных областях науки все чаще возникают задачи, требующие обработки больших объемов данных. Например, анализ данных, полученных с телескопов (*Астрофизика*), исследования поведения земной коры для составления прогноза землетрясений (*Сейсмология*), исследования поведения квантовых частиц (*Физика*), корпоративные системы электронного документооборота крупных организаций (*Делопроизводство*), изучение межмолекулярных связей (*Молекулярная биология*), автоматизированные системы управления реального времени и т.д. Во всех вышеперечисленных задачах актуальной является высокая скорость обработки больших объемов данных. Для быстрых расчетов и обработки данных в современной науке часто используются высокопроизводительные вычислительные кластеры.

В данной статье рассматривается применение одной из реализаций распределенных баз данных для кластерных систем. Описывается созданный пользовательский интерфейс для управления распределенной базой данных в кластерных системах. Предлагается также двухуровневая модель распределенной обработки баз данных для нескольких кластеров, для случаев, когда ресурсов одного кластера недостаточно.

2. Пример реализации распределенной базы данных для кластерных систем

Для хранения и обработки данных в задачах, требующих быстрой обработки данных больших объемов, используются распределенные базы данных. Есть ряд реализаций распределенных баз данных, таких как MySQL Cluster, Oracle Real Application Cluster, DB/2 [7][8] и т.д. Из перечисленных продуктов только программный пакет MySQL Cluster является свободно распространяемым продуктом с открытым исходным кодом, подпадающим под лицензию GPL. Он также имеет ряд преимуществ по сравнению с остальными, поэтому в данной статье рассматривается именно этот программный продукт. MySQL Cluster представляет собой обычный MySQL сервер установленный со специальными параметрами, дающими возможность запускать MySQL на нескольких серверах и создавать распределенные базы данных. В MySQL Cluster также добавлен новый тип таблицы NDB или NDBCLUSTER [7]. Если таблица в базе данных имеет этот тип, то она хранится на узлах кластера в распределенном виде.

Рассмотрим структуру распределенной базы данных на основе пакета MySQL Cluster:

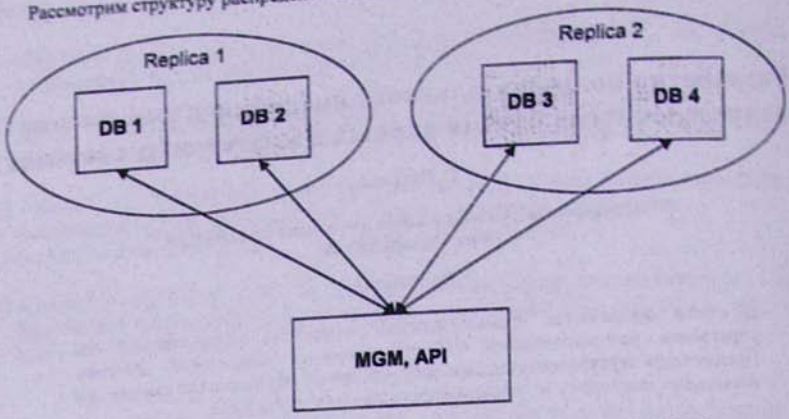

Рис. 2.1 Структура распределенной базы данных на основе пакета MySQL Cluster

На рисунке 2.1 приведен пример такой структуры для одного управляющего узла (MGM,API) и четырех узлов хранения данных (DB). Узлы хранения данных (DB) разбиты на две группы по два узла, в которых организована репликация данных. Реплика - это точная согласованная копия базы данных. Вышеописанная структура распределенной базы данных была установлена в высокопроизводительной вычислительной системе «Армкластер» для организации ряда экспериментов и организации пользовательского интерфейса.

В распределенных базах данные в основном хранятся в оперативной памяти для более эффективной и быстрой обработки [2].

Таким образом, например, для хранения данных объемом в 1Гб необходимо, чтобы суммарный размер оперативной памяти узлов хранения данных, входящих в одну и ту же группу реплики, был не меньше чем 1Гб. Такая организация распределенных баз данных при наличии баз данных больших объемов подразумевает:

Во первых резкое ограничение объема хранимых баз данных на кластере (т.к оперативная память в основном в десятки раз меньше,чем, например, память на жестком диске);

Во вторых при больших объемах данных необходимо постоянно занимать узлы кластера, не имея возможности делать расчеты в те временные промежутки когда база данных не используется.

В связи с вышесказанным, для организации более эффективной работы распределенных баз данных был создан специализированный пользовательский интерфейс, предназначенный для оптимизации использования распределенных баз данных в кластерных системах.

2. 1. Проведенные эксперименты

Для изучения поведения распределенной базы данных [2] в кластерных системах, а также для определения влияний запросов различной степени сложности на их время выполнения, был проведен ряд экспериментов [1][3].

В ходе экспериментов, проведенных в вычислительной системе «Армкластер», была создана база данных объемом в 3,2Гб на 4-х узлах и проведены эксперименты, по итогам которых, были получены следующие результаты:

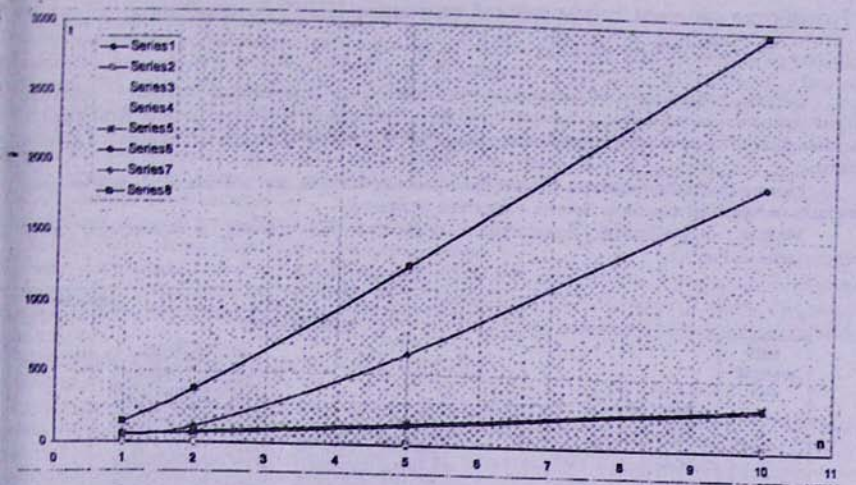

Рис 2.2. Общий график зависимости количества одновременных подсоединений от задержек выполнения различных запросов

На Рис. 2.2 показаны 8 кривых, каждая из которых представляет собой зависимость количества одновременных соединений от времени выполнения данного запроса. Ниже представлены запросы, соответствующие кривым.

1 - SELECT * FROM neds2 LIMIT 500,600;
2 - SELECT max(a1) FROM neds2;
3 - SELECT min(a1) FROM neds2;
4 - SELECT sum(a1) FROM neds2;
5 - SELECT * FROM neds2 WHERE a1=a2;
6 - SELECT * FROM neds2 WHERE a1 LIKE a2;
7 - SELECT * FROM neds2 ORDER BY a1 LIMIT 500,600;
8 - SELECT * FROM neds2 ORDER BY a1,a2,a3,a4 LIMIT 500,600;

Как видно из графика, количество соединений особенно сказывается на времени выполнения запроса при запросах 7 и 8; это те запросы, где есть операция сортировки. При остальных запросах количество соединений не сказывается, а в ряде случаев, как, например, запрос 1 – количество соединений (в диапазоне от 1 до 10) практически не влияет на время выполнения запроса.

Для эффективной обработки данных была определена формула расчета оптимального количества необходимых узлов, исходя из объема базы данных:

$$N = \left[\frac{S + I}{M - m} * R \right] + 1 \qquad (1)$$

где S – размер предполагаемой базы данных (Гб), I – размер индексов базы (Гб), R=[1, 2, 4] – количество реплик, M – оперативная память каждого узла (Гб), m – константа, оперативная память, используемая операционной системой каждого узла, принято значение 0,2Гб, N – количество необходимых узлов, [x] – целая часть числа x.

3. Разработка системы динамической генерации (СДГ) баз данных в кластерных системах.

Система динамической генерации баз данных в кластерных системах призвана оптимизировать использование узлов кластера, дать возможность иметь на кластере базы данных больших объемов в архивированном виде, превышающих размер суммарной оперативной памяти кластера.

При необходимости использования базы данных, система дает возможность динамически создавать распределенные базы данных в кластере из архива.

Модель динамической генерации и обработки баз данных в кластерной среде представлена на Рис 3.1.

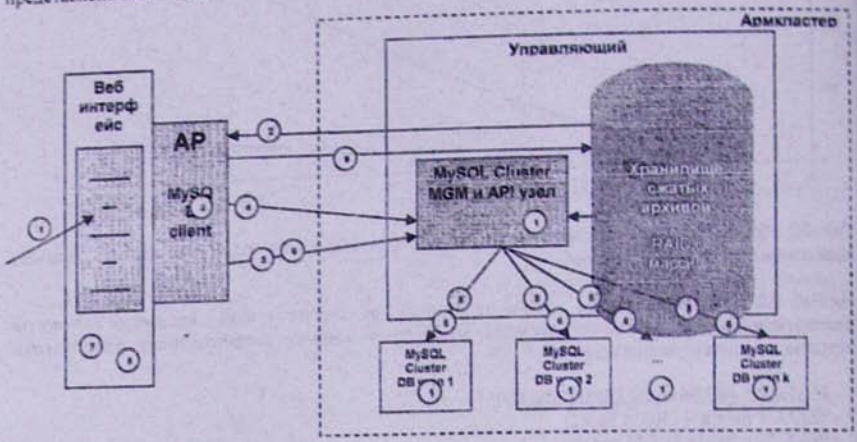

Рис. 3.1 Модель динамической генерации и обработки баз данных в кластерной среде

На Рис 3.1 представлены высокопроизводительная вычислительная система «Армкластер» [5], где показаны узлы кластера и управляющий компьютер, а также сервер доступа (*Access point*, AP). На управляющем компьютере кластера установлены и запущены MGM и API процессы распределенной базы данных для кластерных систем MySQL Cluster, а на узлах кластера - DB процессы, кроме того, на управляющем компьютере кластера находится хранилище данных (RAID массив). На сервере доступа установлен созданный пользовательский интерфейс. Пошагово представим работу пользовательского интерфейса, предетавленного на Рис. 3.1.

1. Выбор архива БД из списка

На сервере доступа к системе «Армкластер» установлен командный интерфейс пользователя, на основе которого создан специализированный веб интерфейс. С помощью этого веб интерфейса пользователь из списка баз данных, находящихся в хранилище сжатых архивов, выбирает ту базу данных, которая требуется для обработки на кластере.

2. Разархивирование (декомпрессия и подготовка к установке)

После выбора базы данных из списка, СДГ обращается в хранилище сжатых архивов и разархивирует необходимый файл базы данных. Сервер доступа имеет прямой доступ к хранилищу управляющего компьютера кластера, т.к. между управляющим компьютером кластера, сервером доступа и узлами кластера установлена система NFS.

3. Расчет количества необходимых узлов и создание конфигурационного файла MySQL Cluster-а

Расчет количества необходимых узлов производится на основании объема разархивированных данных по формуле (1). Полученное значение N является количеством необходимых узлов на кластере. На основании количества узлов и объема базы данных генерируется специальный конфигурационный файл для MySQL Cluster-а.

4. Загрузка конфигурационного файла на управляющий компьютер

Производится загрузка, созданного на предыдущем шаге конфигурационного файла на управляющий компьютер, где находится MGM узел и запускается система управления распределенной базой данных.

5.Установка структур архива базы данных на узлах кластера

6. Заполнение структуры базы данных на узлах данными

Данные шаги представляют из себя подключение к API узлу (который также находится на управляющем компьютере «Армкластер»-а) и выполнение SQL (Structured Query Language) запросов по созданию структуры и заполнения этой структуры данными из разархивированных файлов.

7. Выполнение набора пользовательских SQL запросов

8. Вывод результатов

В пользовательском интерфейсе есть возможность выполнение онлайн запросов с помощью заполнения формы SQL командами и отправления их распределенной базе данных, а также есть возможность загрузить пользовательский SQL файл заданний. После выполнения этих запросов интерфейс выводит результаты обаботки базы данных.

9. Архивация БД (при необходимости)

После завершения работы над базой данных при необходимости производится создание соответствующих SQL файлов (*dump*) базы данных, а также их архивация и перемещение в хранилище сжатых архивов.

10. Удаление БД, остановка процессов MySQL Cluster-а, освобождение узлов кластера

Последним шагом производится удаление базы данных с узлов кластера и остановка процессов MySQL Cluster-а. Таким образом освобождаются ресурсы кластера для дальнейшего использования.

Созданный пользовательский интерфейс позволяет при необходимости (по команде пользователя) динамически генерировать распределенные базы данных в кластерных системах, производить их обработку, при необходимости архивировать и обновлять базы данных в хранилище, а также после завершения обработки удалять и освобождать узлы кластера.

Данная система позволяет оптимизировать использование ресурсов кластера для обработки баз данных больших объемов, и позволяет иметь в архивированном виде множество баз данных, объем которых в сумме превышает объем оперативной памяти кластера, то есть одновременное использование этих баз данных невозможно. Данная система позволяет по требованию пользователя автоматически создавать, сконфигурировать и запускать нужную распределенную базу даных на кластере.

Пользовательский интерфейс динамической генерации баз данных в кластерных системах справедлив для тех баз данных, объем которых не превышает возможности одного кластера (суммарной оперативной памяти). Если же объем базы данных превышает суммарную оперативную память одного кластера, предлагается двухуровневая модель распределенной обработки данных в межкластерной среде, описанная ниже.

4. Двухуровневая модель распределенной обработки баз данных в межкластерной среде.

В случае, если ресурсов одного кластера недостаточно для обработки базы данных большого объема, то есть необходимость организовать обработку баз данных на нескольких кластерах.

Педлагаемая двухуровневая модель распределенной обработки баз данных в межкластерной среде определяет принципы и организовывает такой процесс [6].

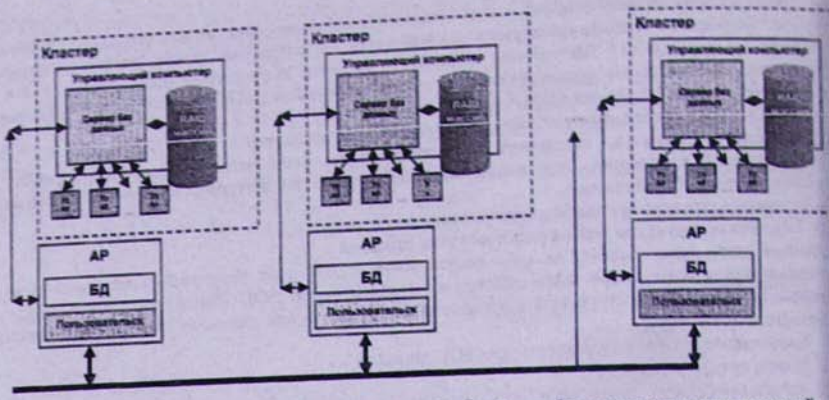

Рис. 4.1 Двухуровневая модель распределенной обработки баз данных в межкластерной среде.

Опишем модель, представленную на Рис. 4.1. На рисунке представлены кластеры, которые связанны между собой через компьютерную сеть. У каждого кластера есть свой управляющий компьютер и узлы, свое хранилище (RAID массив) и свой сервер доступа (*Access point*). На каждом из кластеров установлены распределенные базы данных MySQL Cluster [7], на серверах доступа установлены свои локальные базы данных, а также пользовательский интерфейс, который был представлен в разделе 3.

Принцип работы данной модели следующий. Предлагается разделить данные на n частей, где n - минимальное количество кластеров необходимых для обработки данной базы данных и обработать их на нескольких кластерах.

На каждом из кластеров на основании полученной части базы данных при помощи пользовательского интерфейса автоматически создается распределенная база данных на данном кластере и производится обработка (данная часть точностью повторяет процесс динамической генерации баз данных, описанный в разделе 3). Процесс синхронизации работы между кластерами при обработке данных производится с помощью локальной базы данных, установленной на каждом сервере доступа, которая не требует особого быстродействия и хранит промежуточные результаты обработки.

Связь между серверами доступа осуществляется посредством сети общего доступа.

Данная модель может обеспечить синхронизацию распределенной обработки баз данных больших объемов на нескольких кластерах.

5. Заключение

В данной статье представлен созданный специализированный пользовательский веб интерфейс, позволяющий при необходимости динамически генерировать распределенные базы данных в кластерных системах из хранилища, производить их обработку, при необходимости (в случае изменений) архивировать и обновлять базы данных в хранилище, а также освобождать узлы кластера после завершения обработки, что дает возможность оптимизировать использование узлов кластера для обработки баз данных.

В статье представлена идеология двухуровневой модели распределенной обработки баз данных в межкластерной среде, которая может обеспечить согласованную обработку данных на нескольких кластерах с использованием созданного СДГ распределенных баз данных для кластерных систем.

Автор статьи выражает благодарность Владимиру Григорьевичу Саакяну за постановку задачи и советы в ходе создания и подгодовки данной статьи.

Литература

[1] А.Ю. Пушников, Введение в системы управления базами данных. Часть 1. Реляционная модель данных; Учебное пособие/Издание Башкирского ун-та. - Уфа, 1999.

[2] Г. Ладыженский, Распределенные информационные системы и базы данных http://www.citforum.ru/database/kbd96/45.shtml

[3] В. З. Шнитман, С.Д. Кузнецов, Серверы корпоративных баз данных, информационно-аналитические материалы центра информационных технологий, http://www.citforum.ru/database/skbd/contents.shtml

[4] В. В. Воеводин, Вл. В. Воеводин. Параллельные вычисления; "БХВ-Петербург", Санкт-Петербург, 2004.

[5] H. Astsatryan, Yu. Shoukourian, V. Sahakyan, Creation of High-Performance Computation Cluster and DataBases in Armenia; Proceedings of Second International Conference on Paralel Computations an Control Problems (PACO '2004), pp 466-470, 2004.

[6] G. Petrosyan, V. Sahakyan On Distributed Data Processing Model for Cluster Systems Proceedings of the Sixth International Conference on Computer Science and Information Technologies, 2007.

[7] MySQL AB (2005). MySQL Reference Manual for version 5.1 http://dev.mysql.com/doc/refman/5.1/en/index.html.

[8] Н. Игнатович, DB2 Universal Database ключевые характеристики, IBM 2002.

Կլաստերային համակարգերում տարաբաշխված տվյալների հենքերի համար օգտագործողի ղեկավարման ինտերֆեյսի մշակումը:

Գ. Պետրոսյան

Ամփոփում

Այս հոդվածում ներկայացված է կլաստերային համակարգերի համար տարաբաշխված տվյալաների հենքի համար ղեկավարման ինտերֆեյս, որը թույլ է տալիս դինամիկ կերպով ղեկավարել կլաստերային տվյալների հենքերը: Առաջարկվել է նաև կլաստերային համակարգերի համար տվյալների տարաբաշխած մշակման երկու մակարդականի մոդել: