МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение  
высшего профессионального образования

**«Южно-Уральский государственный университет»**

**(национальный исследовательский университет)**

**Факультет Вычислительной математики и информатики**

**Кафедра системного программирования**

|  |  |
| --- | --- |
| РАБОТА ПРОВЕРЕНА  Рецензент  <ученая степень, ученое звание>  \_\_\_\_\_\_\_\_\_\_ <И.О. Фамилия рецензента>  “\_\_\_”\_\_\_\_\_\_\_\_\_\_\_ 2014 г. | ДОПУСТИТЬ К ЗАЩИТЕ  Заведующий кафедрой, д.ф.-м.н., профессор  \_\_\_\_\_\_\_\_\_\_ Л.Б. Соколинский  “\_\_\_”\_\_\_\_\_\_\_\_\_\_\_ 2014 г. |

**Реализация теста GRAPH500 для параллельной СУБД PargreSQL**

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

ЮУрГУ – 010400.62.2014.08-036-15701.ВКР

|  |  |
| --- | --- |
|  | Научный руководитель  кандидат физ.-мат. наук, доцент  \_\_\_\_\_\_\_\_\_\_\_\_\_М. Л. Цымблер  Автор работы, студент группы ВМИ-456  \_\_\_\_\_\_\_\_\_\_\_\_\_А. Ю. Сафонов  Ученый секретарь  (нормоконтролер)  \_\_\_\_\_\_\_\_\_\_\_\_\_ О. Н. Иванова  “\_\_\_”\_\_\_\_\_\_\_\_\_\_\_ 2014 г. |

Челябинск-2014

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение  
высшего профессионального образования

**«Южно-Уральский государственный университет»**

**(национальный исследовательский университет)**

**Факультет Вычислительной математики и информатики**

**Кафедра системного программирования**

УТВЕРЖДАЮ

Зав. кафедрой СП

\_\_\_\_\_\_\_\_\_\_ Л.Б. Соколинский

08.02.2014

**ЗАДАНИЕ1**

**на выполнение выпускной квалификационной работы бакалавра**

студенту группы ВМИ-356 Сафонову Александру Юрьевичу,

обучающемуся по направлению 010400.62 «Информационные технологии»

1. **Тема работы** (утверждена приказом ректора от <дата и номер приказа>)

Реализация теста Graph500 для параллельной СУБД PargreSQL.

1. **Срок сдачи студентом законченной работы:** 04.06.2014.
2. **Исходные данные к работе2**
3. Пан К.С., Цымблер М.Л. Использование параллельной СУБД PargreSQL для интеллектуального анализа сверхбольших графов // Суперкомпьютерные технологии в науке, образовании и промыш-ленности, 2012. № 1. С. 125-134.
4. Соколинский Л.Б. Параллельные системы баз данных. М.: Издательство Московского университета, 2013. 184 с.
5. **Перечень подлежащих разработке вопросов3**
6. Изучить архитектуру параллельной СУБД PargreSQL и спецификацию теста Graph500.
7. Разработать схему базы данных для хранения графа и промежуточных данных в соответствии со спецификацией теста Graph500.
8. Выполнить проектирование и разработку алгоритмов на языке SQL, реализующих тест Graph500 для параллельной СУБД PargreSQL.
9. Провести вычислительные эксперименты на суперкомпьютере “Торнадо ЮурГУ”, исследующие эффективность параллельной СУБД PargreSQL на тесте Graph500.
10. **Дата выдачи задания:** 08.02.2014.

**Научный руководитель4**

Зам. заведующего кафедрой СП,

кандидат физ.-мат. наук М. Л. Цымблер

**Задание принял к исполнению** А. Ю. Сафонов

Оглавление

[Введение 5](#_Toc383139757)

[1. Описание существующих технологий 6](#_Toc383139758)

[2. Алгоритмы, реализующие теста Graph500 для параллельной СУБД PargreSQL 8](#_Toc383139759)

[3. Вычислительные эксперименты 9](#_Toc383139760)

[3.1. Аппаратная платформа экспериментов 9](#_Toc383139761)

[3.2. План экспериментов 9](#_Toc383139762)

[3.3. Результаты 9](#_Toc383139763)

[Заключение 10](#_Toc383139764)

[Литература 11](#_Toc383139765)

[Приложение 13](#_Toc383139766)

Введение

Актуальность

Большинство существующих тестов производительности суперкомпьютеров направлены на измерение производительности вычислительных мощностей (тесты LINPACK, HPCC RandomAccess). Однако они не позволяют адекватно измерить производительность на задачах с интенсивной обработкой данных (Data Intensive). Примерами таких задач могут служить задачи на сверхбольших графах. Чтобы иметь возможность улучшать производительность на задачах данного типа, требовалось разработать новый сравнительный тест (бенчмарк), который бы учитывал специфику задач с интенсивной обработкой данных. В 2010 около 30 международных экспертов в области высокопроизводительных вычислений из научной сферы и сформировали специальный комитет, с помощью которого была разработана спецификация подобного теста, названного Graph500. Целью данного теста является разработка приложения, состоящего из нескольких основных анализирующих частей (так называемых «ядер»), которые обращаются к единой структуре данных, представляющей собой взвешенный, ненаправленный граф. Текущая спецификация теста содержит 2 ядра, одно ядро из входного списка ребер создает представление графа, которое будет использоваться в дальнейшем, а второе — производит вычисления на этом графе.

Данный сравнительный тест включает в себя масштабируемый генератор данных, который генерирует список ребер, где для каждого ребра указана начальная и конечная вершины. Первое ядро переводит ненаправленный граф в представление, которое будет использоваться всеми следующими ядрами. Запрещены любые дальнейшие модификации этого представления, которые могли бы ускорить работу следующих ядер. Второе ядро осуществляет поиск в ширину на графе. Время работы каждого из ядер должно быть измерено.

Согласно размеру входных данных задача разделяется на 6 классов, описанных в таб. 1.

Таб. . Характеристики классов задач

|  |  |  |  |
| --- | --- | --- | --- |
| **Класс задачи** | **Количество вершин** | **Количество ребер** | **Память** |
| Игрушечный (уровень 10) | 226 | 230 | 17 ГБ |
| Минимальный (уровень 11) | 229 | 233 | 137 ГБ |
| Маленький  (уровень 12) | 232 | 236 | 1 ТБ |
| Средний  (уровень 13) | 236 | 240 | 17 ТБ |
| Большой  (уровень 14) | 239 | 243 | 140 ТБ |
| Огромный  (уровень 15) | 242 | 246 | 1 ПБ |

На основе указанных классов определяются основные параметры генератора графа.

После генерации списка ребер, первое ядро программы переводит его во внутреннее представление, которое будет использоваться всеми последующими ядрами и которое не может быть изменено впоследствии.

Перед началом работы второго ядра в графе случайно выбирается 64 вершины со степенью не меньше 1 (не считая петли), чтобы исключить работу алгоритма на изолированных вершинах.

Цель и задачи

1. Изучить архитектуру параллельной СУБД PargreSQL и спецификацию теста Graph500.
2. Разработать схему базы данных для хранения графа и промежуточных данных в соответствии со спецификацией теста Graph500.
3. Выполнить проектирование и разработку алгоритмов на языке SQL, реализующих тест Graph500 для параллельной СУБД PargreSQL.
4. Провести вычислительные эксперименты на суперкомпьютере “Торнадо ЮурГУ”, исследующие эффективность параллельной СУБД PargreSQL на тесте Graph500.

Структура и объем

Содержание

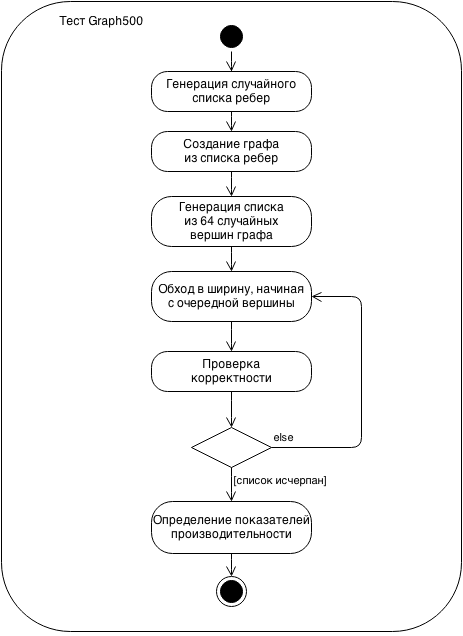


Рис. . Тест Graph500

1. Описание существующих технологий

Базисной концепцией параллельной обработки запросов в реляционных системах баз данных является фрагментный параллелизм [5]. Принципиальная схема обработки запроса с использованием фрагментного параллелизма выглядит следующим образом (см. рис. 2).

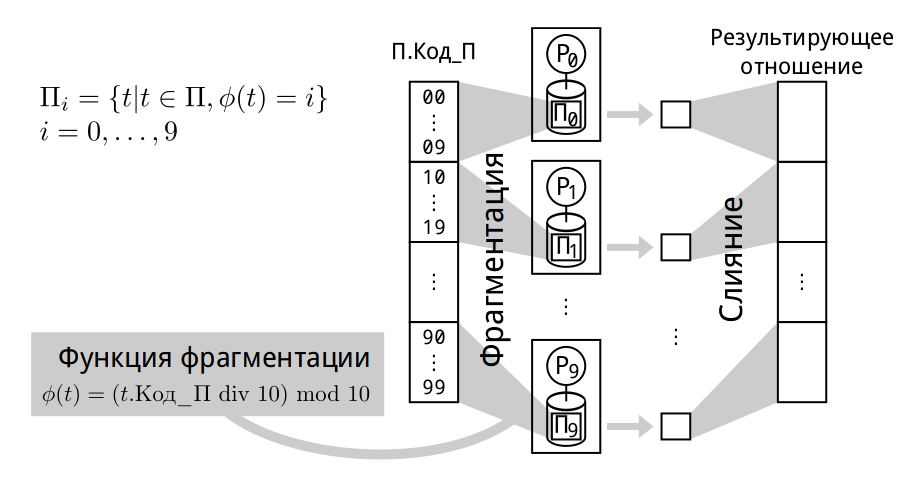


Рис. . Параллельная обработка запроса на основе фрагментного параллелизма

Реляционные отношения, хранящиеся в базе данных, подвергаются горизонтальной фрагментации по дискам многопроцессорной системы. Способ фрагментации определяется функцией фрагментации, которая для каждого кортежа отношения вычисляет номер процессорного узла, на котором должен быть размещен этот кортеж. Запрос параллельно выполняется на всех процессорных узлах в виде набора параллельных агентов [4], каждый из которых обрабатывает отдельный фрагмент отношения на выделенном ему процессорном узле. Полученные агентами результаты сливаются в результирующее отношение.

Функция фрагментации отношения

вычисляет номер процессорного узла, на котором должен храниться кортеж. Величина (количество процессорных узлов) называется степенью фрагментации. Для фрагментации целесообразно использовать атрибутную фрагментацию, которая предполагает, что

где является некоторой функцией, определенной на домене атрибута . То есть атрибутная фрагментация предполагает, что функция фрагментации зависит от определенного атрибута фрагментируемого отношения. Преимущество атрибутной фрагментации в том, что она допустима основными реляционными операциями: группировка, удаление дубликатов, естественное соединение.

Несмотря на то, что каждый параллельный агент в процессе выполнения запроса независимо обрабатывает свой фрагмент отношения, для получения корректного результата необходимо выполнять пересылки кортежей между процессорными узлами. Для организации таких пересылок в соответствующие места дерева плана запроса вставляется оператор *exchange*.

Оператор *exchange* однозначно задается номером порта обмена и функцией пересылки. Функция пересылки для каждого входного кортежа вычисляет логический номер процессорного узла, на котором данный кортеж должен быть обработан. Порт обмена позволяет включать в дерево запроса произвольное количество операторов *exchange* (для каждого оператора указывается свой уникальный порт обмена).

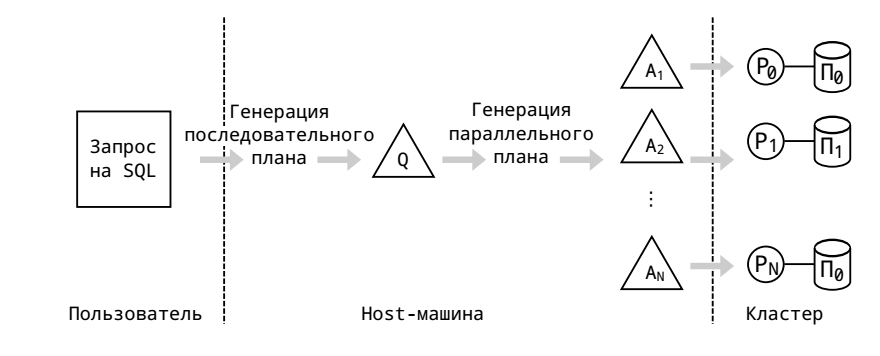


Рис. . Схема обработки запроса в параллельной СУБД.  − последовательный физический план,  − параллельный агент,  ‑‑ вычислительный узел,  − диск

Общая схема организации обработки запросов в параллельной СУБД выглядит следующим образом [4]. Мы будем полагать, что вычислительная система представляет собой кластер из вычислительных узлов (см. рис. 3), и каждое отношение базы данных, задействованное в обработке запроса, фрагментировано по всем этим узлам. В соответствии с данной схемой обработка запроса состоит из трех этапов.

На первом этапе SQL-запрос передается пользователем на выделенную host-машину, где транслируется в некоторый последовательный физический план [30].

На втором этапе последовательный физический план преобразуется в параллельный план, представляющий собой совокупность параллельных агентов. Это достигается путем вставки оператора обмена *exchange* в соответствующие места плана запроса.

На третьем этапе параллельные агенты пересылаются с host-машины на соответствующие вычислительные узлы, где интерпретируются исполнителем запросов. Результаты выполнения агентов объединяются корневым оператором *exchange* на нулевом узле, откуда передаются на host-машину.

Роль host-машины может играть любой узел вычислительного кластера.

1. Алгоритмы, реализующие теста Graph500 для параллельной СУБД PargreSQL
   1. Алгоритм BFS
2. Вычислительные эксперименты
   1. Аппаратная платформа экспериментов

Эксперименты проводились на суперкомпьютере «Торнадо ЮУрГУ»[1]. Характеристики приведены в таб. 2.

Таб. . Характеристики суперкомпьютера «Торнадо ЮУрГУ»

|  |  |
| --- | --- |
| Число вычислительных узлов | 480 |
| Тип процессора | Intel Xeon X5680 (Gulftown, 6 ядер по 3.33 GHz) — 960 шт. |
| Тип coпроцессора | Intel Xeon Phi SE10X (61 ядро по 1.1 GHz) — 384 шт. |
| Оперативная память | 16.9 TB |
| Дисковая память | 150 TB, твердотельные накопители SSD Intel, параллельная система хранения данных Panasas ActiveStor 11 |
| Тип системной сети | InfiniBand QDR (40 Gbit/s) |
| Тип управляющей сети | Gigabit Ethernet |
| Пиковая производительность комплекса | 473.6 TFlops |
| Производительность на тесте LINPACK | 288.2 TFlops |
| Операционная система | Linux CentOS 6.2 |

* 1. План экспериментов

Запустить MPI-реализацию теста Graph500 на суперкомпьютере.

Запустить PargreSQL-реализацию теста Graph500 на суперкомпьютере.

* 1. Результаты

Результаты поражают воображение.

Заключение

Резюме работы.

Список основных результатов.

Направления будущих исследований (?).

Литература

1. Суперкомпьютер «Торнадо ЮУрГУ» // Лаборатория суперкомпьютерного моделирования ЮУрГУ. 2014. URL: http://supercomputer.susu.ac.ru/computers/tornado/ (дата обращения: 14.03.2014)
2. Соколинский Л.Б. Параллельные системы баз данных. М.: Издательство Московского университета, 2013. 184 с.
3. Graph500 Benchmark [Электронный ресурс] // Graph500 [сайт]. [2014]. URL: http://www.graph500.org/specifications (дата обращения: 14.03.2014)
4. Костенецкий П.С., Лепихов А.В., Соколинский Л.Б. Некоторые аспекты организации параллельных систем баз данных для мультипроцессоров с иерархической архитектурой // Алгоритмы и программные средства параллельных вычислений: [Сб. науч. Тр.]. -Екатеринбург: УрО РАН, 2006. С. 42-83.
5. Пан К.С., Цымблер М.Л. Использование параллельной СУБД PargreSQL для интеллектуального анализа сверхбольших графов // Суперкомпьютерные технологии в науке, образовании и промышленности, 2012. № 1. С. 125-134.
6. Мокрозуб В.Г. Графовые структуры и реляционные базы данных в автоматизированных интеллектуальных информационных системах. – М.: Издательский дом "Спектр", 2011. 108 c.
7. Гладков М., Шибанов С. Сложные структуры в реляционных базах данных // Открытые системы. 2004. URL: http://www.osp.ru/os/2004/02/183939/ (дата обращения: 24.03.2014)

Спецификация Graph500

Учебник/статья ЛБС о ПСУБД

Спецификация тестов TPC

<Торнадо ЮУрГУ>

???

Приложение