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 реляционной СУБД на основе фрагментного параллелизма**

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 реляционной СУБД на основе фрагментного параллелизма.

1. **Срок сдачи студентом законченной работы:** 04.06.2014.
2. **Исходные данные к работе2**
3. **Перечень подлежащих разработке вопросов3**
4. Запуск теста Graph500 на суперкомпьютере «Торнадо ЮУрГУ».
5. Реализация теста Graph500 для параллельной СУБД PargreSQL.
6. **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

[1. Введение 4](#_Toc379952477)

[1.1. Спецификация теста Graph500 4](#_Toc379952478)

[2. Сравнение референсных реализаций 6](#_Toc379952479)

[3. Литература 7](#_Toc379952480)

1. Введение

Графовые задачи характеризуются нерегулярной структурой графа, низкой локализацией данных и превалированием доступа к данным над вычислениями.

Для оценки производительности суперкомпьютеров на графовых задачах был специально создан рейтинговый список Graph500, в качестве вычислительного ядра реализации которого используется тест «поиск вширь» в графе. Рейтинг суперкомпьютеров Graph500 публикуется два раза в год, а места в списке распределяются с учетом размера графа и полученной производительности суперкомпьютера при его обработке (единица измерения — TEPS, то есть количество пройденных дуг в секунду), причем размер имеет более приоритетное значение, чем производительность.

поиск вширь в графе, лежащий в основе Graph500, не в полной мере соответствует реальным графовым задачам. В настоящее время создатели Graph500 работают над добавлением новых ядер, учитывающих природу реальных графовых задач.

* 1. Спецификация теста Graph500

The intent of this benchmark problem ("Search") is to develop a compact application that has multiple analysis techniques (multiple kernels) accessing a single data structure representing a weighted, undirected graph. In addition to a kernel to construct the graph from the input tuple list, there is one additional computational kernel to operate on the graph.

This benchmark includes a scalable data generator which produces edge tuples containing the start vertex and end vertex for each edge. The first kernel constructs an undirected graph in a format usable by all subsequent kernels. No subsequent modifications are permitted to benefit specific kernels. The second kernel performs a breadth-first search of the graph. Both kernels are timed.

1. Generate the edge list.

2. Construct a graph from the edge list (timed, kernel 1).

3. Randomly sample 64 unique search keys with degree at least one, not counting self-loops.

4. For each search key:

1. Compute the parent array (timed, kernel 2).

2. Validate that the parent array is a correct BFS search tree for the given search tree.

5. Compute and output performance information.

1. Сравнение референсных реализаций
2. Литература