Университет ИТМО
Кафедра вычислительной техники

Отчет по прохождению практики

Параллельные вычисления

Выполнил студент
группы P3310
Шаймарданов Р. Р.

Руководитель
Соснин В. В.

Санкт-Петербург
2016

Содержание

1 Введение

Цель: составить составить список из наиболее авторитетных литературных источников по "Параллельным вычислениям".

Задачи

1. Описать инструментарий, необходимый для выполнения практического задания.
2. Составить перечень современных (новее 2010 года) англоязычных источников, посвящённых параллельным вычислениям в системах с общей памятью.
3. Разбить найденные источники на четыре группы:
   - платные книги
   - журналы
   - бесплатные книги
   - онлайн-курсы
4. Сформулировать критерий уровня авторитетности для найденных материалов, выполнить ранжирование источников внутри групп.
5. Найти, провести разбиение и ранжировать русскоязычные источники, аналогично с иностранными.
6. Провести сравнение работы потоков при различных запусках ОС.

2 Инструментарий

2.1 Система компьютерной верстки TEX(LATEX)

2.1.1 Описание

TEX— это низкоуровневый язык разметки и программирования, созданный Дональдом Кнутом для единообразной вёрстки документов. Кнут начал разрабатывать систему набора текста TEXв 1977 году для исследования потенциальных возможностей оборудования цифровой печати, которое в то время начинало проникать в издательское дело. Главным образом он надеялся улучшить качество печатной продукции, которое расстраивало в его собственных книгах и статьях. После выпуска в 1989 году поддержки восьмибитных символов разработка TEXприостановилась, только иногда выходили версии с исправленными ошибками.

LATEX— основанный на TEXпакет макросов, созданный Лесли Лампортом. Основная цель — упростить вёрстку текста, особенно в документах с математическими формулами. Значительно позднее авторы разработали для LATEXрасширения, которые называются пакетами или стилями. Некоторые из них распространяются вместе с большинством дистрибутивов TEX/LATEX.

Так как LATEXсодержит часть команд TEX, то создание документа в LATEX- тоже программирование: создаётся текстовый файл в LATEXразметке, макросы LaTeX обрабатывают его и производят конечный документ.

2.1.2 Сравнение с другими программными средствами

Подход LATEXк созданию документа называется WYSIWYM[1]: во время набора текста Вы не видите окончательный вариант документа, толь-

[1] What You See Is What You Mean (То, что ты видишь, есть то, что ты имеешь в виду)

ко логическую структуру этого документа. Оформлением занимается сам LaTeX. Такой подход имеет как достоинства, так и недостатки по сравнению с WYSIWYG[2] программами, такими как Openoffice.org Writer или Microsoft Word.

Достоинства:

- Файлы с исходными текстами можно просмотреть в любом текстовом редакторе, они понятнее в отличие от сложных бинарных файлов и форматов XML, используемых WYSIWYG программами.
- Вы полностью сосредотачиваетесь на структуре и содержании документа и забываете о том, как будет выглядеть печатный вариант.
- Легко скопировать структуру документа в другой документ, в WYSIWYG программах не всегда ясно, какое именно было использовано форматирование.
- Так как исходный документ содержит просто текст, с помощью программных средств на любом языке программирования можно создать таблицы, рисунки, формулы и т.д.

Недостатки:

- Во время редактирования документа нельзя (обычно) увидеть его окончательный вариант без компиляции.
- Необходимо знать нужные команды разметки LaTeX.
- Иногда сложно получить требуемый вид документа.

Документ LaTeX— обычный текстовый файл, в котором указано содержание документа вместе с дополнительной разметкой. При обработке исходного файла макросами LaTeXможно получить документ в разных форматах. Изначально LaTeXподдерживает форматы DVI и PDF, но при использовании другого ПО можно легко получить PostScript, PNG, JPG и т.д.

[2]What You See Is What You Get (Что видишь, то и получишь)

2.1.3 Выбор инструмента редактирования

В ходе изучения всех возможных вариантов работа с LaTeX для создания данного отчета, была выбрана программа Textmaker

Выбор Textmaker'а обусловлен следующими его особенностями:

- Автоматическая подсветка синтаксиса
- Функция автодополнения команд LaTeX
- Сокрытие блоков кода
- Быстрая навигация по структуре документа
- Указание на строку с ошибкой, для быстрой отладки
- Интегрированный просмотр PDF

2.2 Система контроля версий Git

2.2.1 Описание

Система управления версиями — программное обеспечение для облегчения работы с изменяющейся информацией. Система управления версиями позволяет хранить несколько версий одного и того же документа, при необходимости возвращаться к более ранним версиям, определять, кто и когда сделал то или иное изменение, и многое другое.

Git — это гибкая, распределённая система управления версиями. Проект был создан Линусом Торвальдсом для управления разработкой ядра Linux, первая версия выпущена 7 апреля 2005 года. На сегодняшний день его поддерживает Джунио Хамано. Программа является свободной и выпущена под лицензией GNU GPL версии 2.

У каждого разработчика, использующего Git, есть свой локальный репозиторий, позволяющий локально управлять версиями. Затем, сохраненными в локальный репозиторий данными, можно обмениваться с другими пользователями. Часто при работе с Git создают центральный репозиторий, с которым остальные разработчики синхронизируются. В этом случае все участники проекта ведут свои локальны разработки и беспрепятственно скачивают обновления из центрального репозитория. Когда необходимые работы отдельными участниками проекта выполнены и отлажены, они, после удостоверения владельцем центрального репозитория в корректности и актуальности проделанной работы, загружают свои изменения в центральный репозиторий. Работа над версиями проекта в Git может вестись в нескольких ветках, которые затем могут с легкостью полностью или частично объединяться, уничтожаться, откатываться и разрастаться во все новые и новые ветки проекта.

2.2.2 Сравнению с другими системами контроля версий

Достоинства:

- Надежная система сравнения ревизий и проверки корректности данных, основанные на алгоритме хеширования Secure Hash Algorithm 1.
- Гибкая система ветвления проектов и слияния веток между собой.
- Наличие локального репозитория, содержащего полную информацию обо всех изменениях, позволяет вести полноценный локальный контроль версий и заливать в главный репозиторий только полностью прошедшие проверку изменения.
- Высокая производительность и скорость работы.
- Удобный и интуитивно понятный набор команд.
- Множество графических оболочек, позволяющих быстро и качественно вести работы с Git'ом.
- Возможность делать контрольные точки, в которых данные сохраняются без дельта компрессии, а полностью. Это позволяет уменьшить скорость восстановления данных, так как за основу берется ближайшая контрольная точка, и восстановление идет от нее. Если бы контрольные точки отсутствовали, то восстановление больших проектов могло бы занимать часы.
- Широкая распространенность, легкая доступность и качественная документация.
- Гибкость системы позволяет удобно ее настраивать и даже создавать специализированные контроля системы или пользовательские интерфейсы на базе git.

- Универсальный сетевой доступ с использованием протоколов http, ftp, rsync, ssh и др.

Недостатки:

- Возможные (но чрезвычайно низкие) совпадения хеш - кода отличных по содержанию ревизий.
- Не отслеживается изменение отдельных файлов, а только всего проекта целиком, что может быть неудобно при работе с большими проектами, содержащими множество несвязных файлов.
- При начальном (первом) создании репозитория и синхронизации его с другими разработчиками, потребуется достаточно длительное время для скачивания данных, особенно, если проект большой, так как требуется скопировать на локальный компьютер весь репозиторий.

2.2.3 Основные команды

add: Добавляет содержимое рабочей директории в индекс для последующего коммита.

status: Показывает состояния файлов в рабочей директории и индексе: какие файлы изменены, но не добавлены в индекс; какие ожидают коммита в индексе. Вдобавок к этому выводятся подсказки о том, как изменить состояние файлов.

diff: Используется для вычисления разницы между любыми двумя Git деревьями.

difftool: Запускает внешнюю утилиту сравнения для показа различий в двух деревьях, на случай если вы хотите использовать что-либо отличное

от встроенного просмотрщика git diff.

commit: Берёт все данные, добавленные в индекс с помощью git add, и сохраняет их слепок во внутренней базе данных, а затем сдвигает указатель текущей ветки на этот слепок.

reset: Используется в основном для отмены изменений. Она изменяет указатель HEAD и, опционально, состояние индекса.

rm: Используется в Git для удаления файлов из индекса и рабочей директории. Она похожа на git add с тем лишь исключением, что она удаляет, а не добавляет файлы для следующего коммита.

mv: Удобный способ переместить файл, а затем выполнить git add для нового файла и git rm для старого.

*clean:*Удаление мусора из рабочей директории. Это могут быть результаты сборки проекта или файлы конфликтов слияний.

2.2.4 GitHub

GitHub — крупнейший веб-сервис для хостинга IT-проектов и их совместной разработки. Основан на системе контроля версий Git и разработан на Ruby on Rails и Erlang компанией GitHub, Inc (ранее Logical Awesome).

Для выполнения практической работы создан репозиторий в аккаунте RandomRuslan на GitHub'e.

3 Перечень обучающих источников

В ходе данной работы необходимо найти современные (новее 2010 года) материалы, оказывающие поддержку в изучении параллельных вычислений в системах с общей памятью. Все источники должны быть разбиты на иностранные и русскоязычные, а внутри этих групп на платные и бесплатные книги, журналы и онлайн-курсы. Выполнить внутри каждой группы ранжирование источников по авторитетности, а также сформулировать данный критерий на основе различных показателей материалов.

3.1 Критерий сравнения

Так как все исследуемые материалы имеют различные особенности, то и критерии для сравнения и сортировки будут различными. Каждый критерий формируется на основе различных численных значений, играющих важную роль в оценке авторитетности данного материала. Среди них:

- *Цитирование* — число ссылок на данную работу. Данное число вычисляется с использованием бесплатных общедоступных баз данных в Интернете, с помощью Google Scholar.

- *h-Индекс (Индекс Хирша)* — наукометрический показатель, представляющий собой суммарное число ссылок на работы учёного. Критерий основан на учёте числа публикаций исследователя и числа цитирований этих публикаций, является количественной характеристикой продуктивности учёного за весь период научной деятельности. Учёный имеет индекс h, если h из его N статей цитируются как минимум h раз каждая, в то время как оставшиеся (N – h) статей цитируются менее, чем h раз каждая. Данное число также вычисляется с использованием Google Scholar. Если у источника несколько авторов, берется наибольший из h-Индексов.

- *g-Индекс* — показатель, схожий с Индексом Хирша, рассчитываемый на

основе распределения цитирований, полученных публикациями ученого. Для множества статей, отсортированного в порядке убывания количества цитирований, которые получили эти статьи, g-индекс это наибольшее число, такое, что g самых цитируемых статей получили суммарно не менее g^2 цитирований. Данное число также вычисляется с использованием Google Scholar. Если у источника несколько авторов, берется наибольший из g-Индексов.

- *Оценка на Amazon.com* — среднее значение выставленных пользователями оценок по пятибальной шкале для данного товара на сайте.

- *SNIP* (Source Normalized Impact per Paper), разработанный Центром CWTS, отражает влияние контекстной цитируемости журнала, что позволяет непосредственно сравнивать журналы различной тематики, принимая во внимание частоту, с которой авторы цитируют другие источники, скорость развития влияния цитаты и степень охвата литературы данного направления базой данных. Данное число вычисляется с использованием бесплатных общедоступных баз данных в Интернете, с помощью www.journalmetrics.com.

- *SJR* (SCImago Journal Rank) представляет собой рейтинг журналов, разработанный исследовательской группой SCImago. Он дает возможность оценить научный престиж работ ученых, исходя из количества весомых цитат на каждый документ. Журнал наделяет собственным «престижем» или статусом другие журналы, цитируя опубликованные в них материалы. Фактически это означает, что цитата из источника с относительно высоким показателем SJR имеет большую ценность, чем цитата из источника с более низким показателем SJR. Данное число вычисляется с использованием бесплатных общедоступных баз данных в Интернете, с помощью www.journalmetrics.com.

3.2 Иностранные материалы

3.2.1 Книги

Материалы данного раздела разделены на две категории:

- Посвящённы параллельным вычислениям, разбирая системы с общей памятью Таблица 1
- посвящённых параллельным вычислениям в целом (возможно, для систем с общей памятью) Таблица 2

Критерий оценивания формируется по формуле:

$K = 10 * C * A + g/10^n$, где

C — количество цитирования книги

A — оценка книги на Amazon.com

g — g-Индекс автора

n — количество разрядов в наибольшем g-index

В итоге, критерий представляется десятичной дробью, где:

- *Целая часть* $(10 * C * A)$

  Оценка на Amazon.com выступает в роли коэффициента количества цитирования. Если оценка отсутствует, коэффициент принимается равным 3-м, так как это среднее возможное значение оценки(т. е. оценка выше увеличит значение кол-ва цитирования). Умножение на 10 необходимо, чтобы произведение оставалось целым числом.

- *Дробная часть* $(g/10^n)$

  Деление на степень десяти необходимо, чтобы сместить g-Индекс в дробную часть критерия. Является дополнительным, учитывается только в случае, если целые части равны.

Таблица 1

Название	Автор	Цитирование	Амазон	g-index	Критерий
The Art of Multiprocessor Programming, Revised Reprint 1st Edition	Maurice Herlihy, Nir Shavit	951	4.1	50	38991.5
An Introduction to Parallel Programming 1st Edition	Peter Pacheco	196	4.0	35	7840.35
Multicore Application Programming: for Windows, Linux, and Oracle Solaris (Developer's Library) 1st Edition	Darryl Gove	33	4.5	13	1485.13
Multicore and GPU Programming: An Integrated Approach 1st Edition	Gerassimos Barlas	4	3.7	26	148.26
Programming Models for Parallel Computing (Scientific and Engineering Computation)	Pavan Balaji	-	-	39	0.39
Shared Memory Application Programming: Concepts and Strategies in Multicore Application Programming 1st Edition	Victor Alessandrini	0	-	18	0.18

Multicore Software Development Techniques: Applications, Tips, and Tricks (Newnes Pocket Books) 1st Edition	Robert Oshana	-	-	13	0.13

Таблица 2

Название	Автор	Цитирование	Амазон	g-index	Критерий
Parallel Programming: for Multicore and Cluster Systems 2nd ed.	Thomas Rauber, Gudula Rünger	244	5	34	12200.34
Structured Parallel Programming: Patterns for Efficient Computation 1st Edition	Michael McCool, James Reinders, Arch Robison	162	4.7	50	7614.5
Masterkurs Parallele und Verteilte Systeme: Grundlagen und Programmierung von Multicore-Prozessoren, Multiprozessoren, Cluster, Grid und Cloud (German Edition) 2nd Edition	Günther Bengel, Christian Baun, Marcel Kunze, Karl-Uwe Stucky	27	-	47	810.47

Parallel Computer Organization and Design 1st Edition	Michel Dubois, Murali Annavaram, Per Stenström	0	4.8	50	0.5
Parallel Computing for Data Science: With Examples in R, C++ and CUDA (Chapman & Hall/CRC The R Series)	Norman Matloff	0	3	31	0.31

3.2.2 Бесплатные книги

Критерий оценивания формируется по той же формуле, что для платных иностранных книг.

Таблица 3

Название	Автор	Цитирование	Амазон	g-index	Критерий
Introduction to Parallel Computing	Blaise Barney	201	-	21	6030.21
Introduction to High-Performance Scientific Computing	Victor Eijkhout	27	3.5	50	945.5
The Practice of Parallel Programming	Sergey Babkin	3	4.5	2	135.02
Programming on Parallel Machines : GPU, Multicore, Clusters and More	Norm Matloff	0	-	7	0.07

3.2.3 Журналы

Критерий для данного раздела формируется на основе двух популярных показателях оценки журналов — SNIP и SJR. Так как они равноправны и близки по значению (не отличаются порядком), критерий формируется как простая сумма данных показателей.

Таблица 4

Название	Редактор	SNIP	SJR	Критерий
Transactions on Parallel and Distributed Systems	David A. Bader	3.892	2.017	5.909
Parallel Computing	Jeffrey Hollingsworth	2.141	1.232	3.373
Journal of Parallel and Distributed Computing	Viktor Prasanna	1.991	1.093	3.084
Parallel Processing Letters	Selim G. Akl	1.453	0.499	1.952

3.2.4 Онлайн-курсы

Формирование данного критерия авторитетности осложнено тем, что разные ресурсы предоставляют разную информацию о курсах. Из общих показателей возможно выделить только g-Индекс и h-Индекс авторов и лекторов курсов, которые и лягут в основу критерия авторитетности. Так как они равноправны и близки по значению (не отличаются порядком), критерий формируется как простая сумма данных показателей.

Таблица 5

Название	Автор/Лектор	g-index	h-index	Критерий
Heterogeneous Parallel Programming	Wen-mei W. Hwu	50	50	100
Parallel Computing	Alan Edelman	50	38	88
Parallel Programming for Multicore	Kathy Yelick	8	36	44
Multicore Programming	Tom Van Cutsem	14	26	40
Parallel Programming for Multicore Machines Using OpenMP and MPI	Constantinos Evangelinos	10	27	37
High Performance Scientific Computing	Randy LeVeque	2	3	5

3.3 Российские материалы

3.3.1 Книги

Из-за недостатка показателей для русскоязычных материалов, критерий оценивания формируется по формуле:

$K = C + g/10^n$, где

C — количество цитирования книги

g — g-Индекс автора

n — количество разрядов в наибольшем g-index

В итоге, критерий представляется десятичной дробью, где:

- *Целая часть* (C)
  Число ссылок на данную работу, показывающее ее значимость.

- *Дробная часть* $(g/10^n)$
  Деление на степень десяти необходимо, чтобы сместить g-Индекс в дробную часть критерия. Является дополнительным, учитывается только в случае, если целые части равны.

Таблица 6

Название	Автор	Цитирование	g-index	Критерий
Технологии параллельного программирования MPI и OpenMP	Александр Антонов	18	25	18.25
Современные языки и технологии параллельного программирования	Виктор Гергель	10	10	10.1

Инструменты параллельного программирования в системах с общей памятью	Кирилл Корняков, Валентина Кустикова, Иосиф Мееров, Алексей Сиднев, Александр Сысоев, Александр Шишков	8	12	8.12
Введение в параллельные методы решения задач. Учебное пособие	Михаил Якобовский	7	4	7.04
Практикум по методам параллельных вычислений	Александр Старченко, Евгений Данилкин, Валентина Николаева, Сергей Проханов	3	23	3.23
Основы параллельного программирования. Учебное пособие	Кирилл Богачев	2	9	2.9
Вычисления на многопроцессорных компьютерах. Параллельные вычисления на основе технологии OpenMP: учебное пособие	Алексей Геннадьевич Абрамов	0	3	0.3

3.3.2 Бесплатные книги

Критерий оценивания формируется по той же формуле, что для платных русскоязычных книг.

Таблица 7

Название	Автор	Цитирование	g-index	Критерий
Модели параллельного программирования	Федотов Илья	2	2	2.2

3.3.3 Онлайн-курсы

Формирование данного критерия авторитетности осложнено тем, что разные ресурсы предоставляют разную информацию о курсах. Из общих показателей возможно выделить только g-Индекс авторов и лекторов курсов. В итоге, критерий формируется по формуле:

$K = g + S/10^n$, где

g — g-Индекс автора

S — среднее арифметическое просмотров каждой лекции(округленное до целого) n — количество разрядов в наибольшем среднем просмотров

В итоге, критерий представляется десятичной дробью, где:

- *Целая часть* (g)
  g-Индекс автора

- *Дробная часть* $(S/10^n)$
  Деление на степень десяти необходимо, чтобы сместить среднем просмотров в дробную часть критерия. Является дополнительным, учитывается только в случае, если целые части равны.

Таблица 7

Название	Автор/Лектор	g-index	Среднее просмотров	Критерий
Высокопроизводительные вычисления для многопроцессорных многоядерных систем	Гергель Виктор Павлович, Сысоев Александр Владимирович, Козинов Евгений Александрович, Лабутина Анна Андреевна	15	-	15

Инструменты параллельного программирования в системах с общей памятью	Сысоев Александр Владимирович, Мееров Иосиф Борисович, Сиднев Алексей Александрович, Шишков Александр Валерьевич, Корняков Кирилл Владимирович, Кустикова Валентина Дмитриевна	12	-	12
Параллельное программирование	Евгений Калишенко, Алексей Злобин	2	2416	2.2416
Параллельное программирование	Роман Елизаров	0	3174	0.3174
Эффективные параллельные алгоритмы: методика BSP	Александр Тискин	0	1573	0.1573

Примечание

Русскоязычные журналы на данную тематику найдены не были

4 Практическая часть

4.1 Описание

Необходимо провести сравнение времени работы программы для различных входных параметров (количества экспериментов и потоков) и различных запусков ОС. По данным исследованиям построить графики времени работы с границами доверительного интервала и параллельного ускорения.

Ускорение $S(n)$ определяют как отношение:

$$S(n) = \frac{T_1(n)}{T_2(n)}, \text{ где}$$

n — рассматриваемый входной параметр
T_1 — время первого измерения
T_2 — время второго измерения

Замеры необходимо провести для Linux-системы:

1. без виртуализации
2. с виртуализацией

4.2 Программа

```
#include <omp.h>
#include <stdio.h>
double run_parallel_experiment(int amount_of_threads, unsigned int experiment_size){
    unsigned int i;
    double sum_of_squares = 0;

    omp_set_num_threads(amount_of_threads);
    #pragma omp parallel for reduction(+:sum_of_squares)
    for (i = 0; i < experiment_size; ++i) {
        sum_of_squares += i*i;
    }
    return sum_of_squares;
}
int main() {
    double result;
    double t1;
    double t2;

    t1 = omp_get_wtime(); // timestamp before the experiment
    result = run_parallel_experiment(2, 100000000);
    t2 = omp_get_wtime(); // timestamp after the experiment

    printf("result=%e, time_in_milliseconds=%f\n", result, 1000*(t2 - t1));
    return 0;
}
```

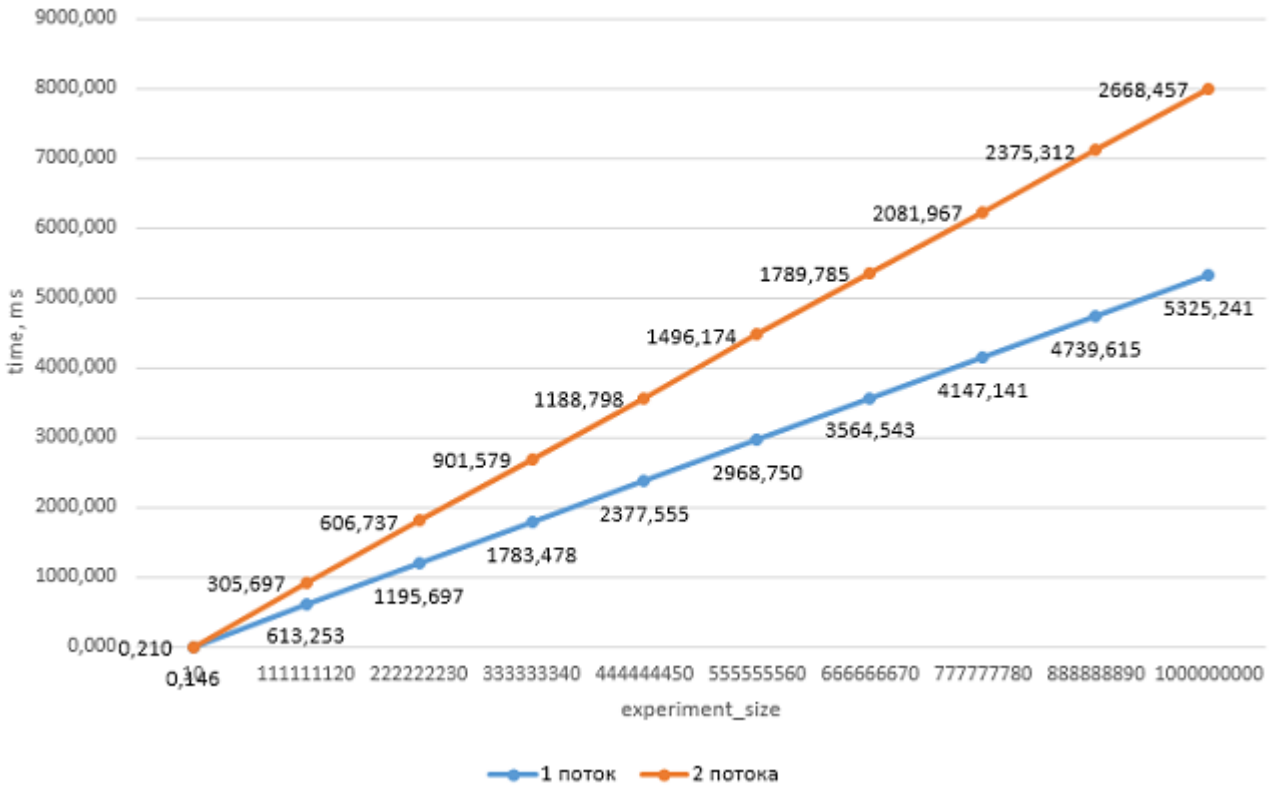
Для проведения сравнения данная программа запускалась для десяти значений experiment_size (от 10^1 до 10^9 с шагом 111111110). Количество потоков (amount_of_threads) принималось равным 1 или 2.

4.3 Описание операционной системы

Операционной системой для исследования была выбрана ОС Ubuntu (64-bit), запускаемая на 4х-ядерном компьютере с загрузочной флешки и на виртуальной машине с помощью программы Oracle VM VirtualBox, запущенной со следующими характеристиками:
Оперативная память: 3152 Мб
Процессоры: 4
Порядок загрузки: Гибкий диск, Оптический диск, Жесткий диск
Ускорение: VT-x/AMD-V, Nested Paging, PAE/NX, Паравиртуализация KVM

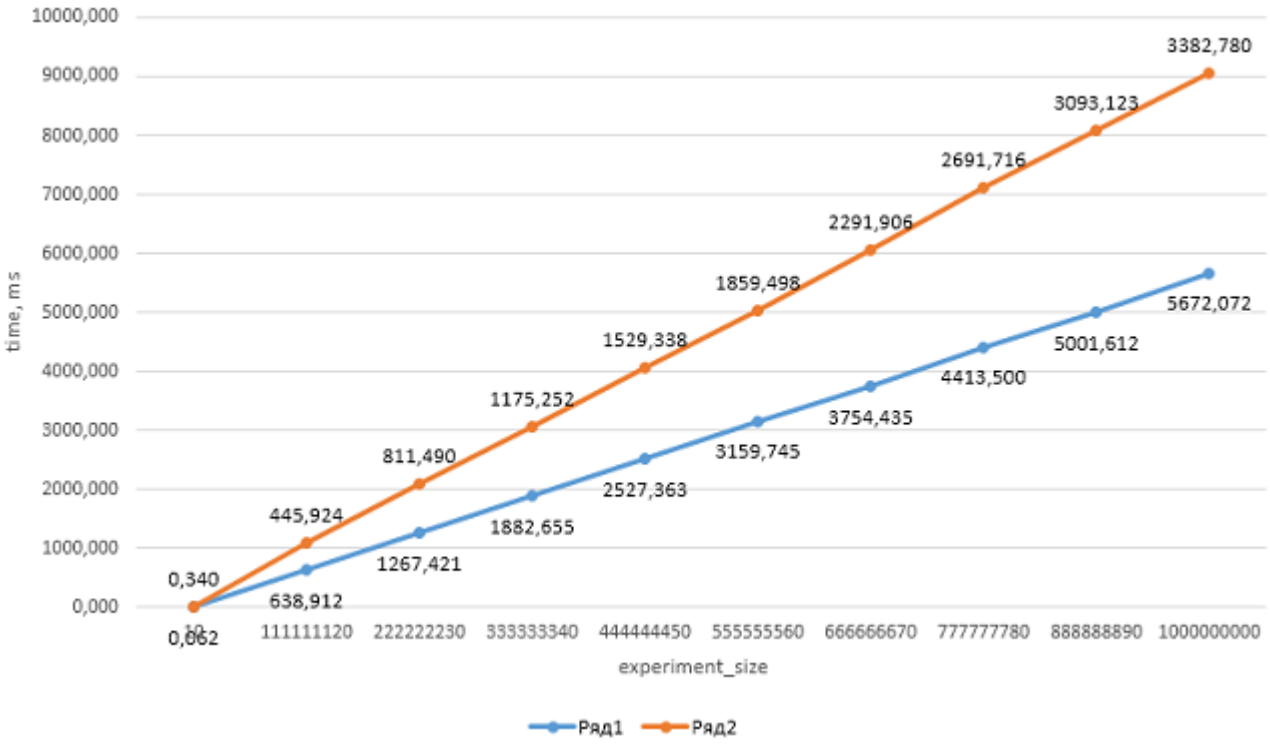
4.4 Сравнение

4.4.1 График работы без виртуализации



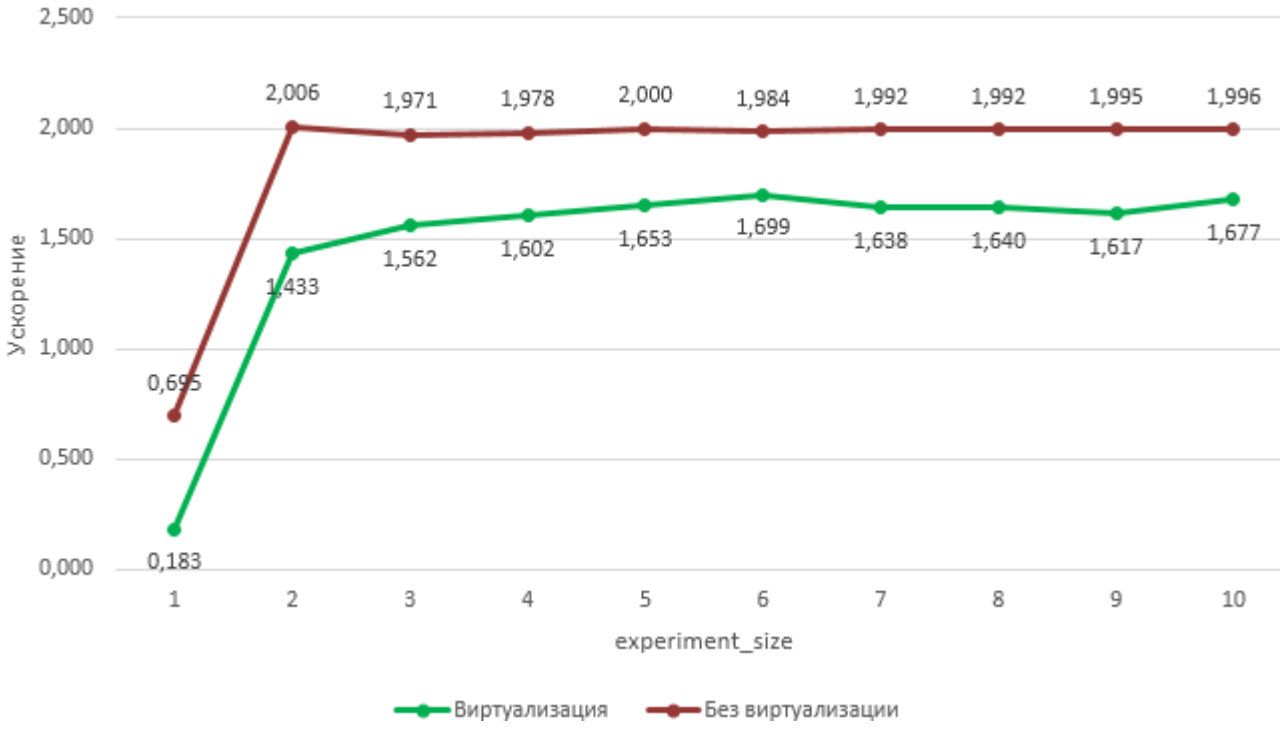
По данному графику видно, что в обоих случаях зависимость времени исполнения от количества потоков линейна. Работа двух параллельно работающих потоков осуществляется быстрее, кроме случая, когда значение experiment_size мало.

4.4.2 График работы с виртуализацией



По графику работы программы на виртуальной машине можно сделать такие же выводы, что и для работы программы без виртуализации. Так же следует заметить, что время работы для аналогичного количества потоков больше с виртуализацией, так как неизбежны растраты ресурсов на поддержание работы самой виртуальной машины.

4.4.3 График ускорения



По графику ускорений можно сделать следующие выводы:

1. Характер графиков ускорений для обоих случаев одинаков.
2. Увеличение количества потоков сказывается лучше для работы программы без виртуализации.
3. Начиная с некоторого значения, ускорение перестает изменяться.
4. Ускорение программы без виртуализации более стабильно, не подвержено скачкам.

5 Выводы

- Существует большой выбор иностранных материалов для изучения параллельных вычислений в системах с общей памятью, как платных, так и выложенных в свободный доступ.
- Выбор среди русскоязычных материалов меньше, однако достаточен для исследования данной тематики.
- Представлены только современные материалы, следовательно вся информация обновляется, не теряя актуальности.
- Выявлены различные критерии сравнения материалов, разной степени важности и достоверности, а также способы их вычисления.
- Увеличение количества параллельных потоков ускоряет выполнение программы для немалых требуемых вычислениях.
- Работа программ при виртуализации проходит медленнее и меньше подвержена ускорению путем добавления дополнительных потоков, из-за затраты ресурсов на поддержание работы виртуальной машины.

6 Используемые сайты

1. ru.wikibooks.org
2. git-scm.com
3. Amazon.com
4. ozon.ru
5. Google Scholar
6. arxiv.org
7. journalmetrics.com
8. computing.llnl.gov
9. tacc.utexas.edu
10. freecomputerbooks.com
11. freetechbooks.com
12. elsevier.com/journals
13. sciencedirect.com
14. computer.org
15. ocw.mit.edu
16. coursera.org
17. cs.berkeley.edu
18. coursetalk.com
19. soft.vub.ac.be/soft

20. books.google.ru

21. elib.spbstu.ru

22. scanlibs.com

23. pp-book.narod.ru

24. lektorium.tv

25. hpcc.unn.ru