

О подборе наглядных задач для вводного курса параллельных вычислений

Е.А. Еремин

Пермский государственный гуманитарно-педагогический университет

При изучении технологий решения сложных задач, в том числе и параллельных вычислительных технологий, существует проблема выбора иллюстративных примеров. В данной работе делается попытка найти простую задачу для вводного курса, подходящую для наглядной демонстрации преимуществ параллельной обработки. Ускорение расчетов оценивается при помощи разработанного автором учебного программного обеспечения.

Ключевые слова: параллельные вычисления, основы, вводный курс, задачи, школьная информатика.

1. Введение

Переход от последовательных вычислений к параллельным нетривиален: требуется принципиально другое мышление, причем навыки разработки классических алгоритмов не только не помогают, но могут даже дезориентировать программиста. Очень важно поэтому «... как можно раньше обратить внимание обучающихся на необходимость критического отношения к философии последовательных вычислений. Ведь именно с этой философией им приходится сталкиваться на протяжении всего образования как в школе, так и в вузе.» [1] Следовательно, требуется уделять внимание не только профессиональным вузовским курсам, но также и общеобразовательным разделам информатики, включая школу: «разработка целостной методики пропедевтической подготовки учащихся основной школы в области параллельных вычислений – задача нескольких ближайших лет» [2]. Все более активно обсуждается вопрос о необходимости изучения основ параллельной обработки данных в педагогических вузах. В [3], в частности, описана технология строгого отбора содержания учебного материала к курсу «Основы параллельного программирования» для учителей математики и информатики при помощи современных методов построения семантических сетей.

Передовые школьные учителя с энтузиазмом воспринимают возможности новой темы. В литературе опубликованы различные интересные игровые подходы к одновременному выполнению работы несколькими исполнителями (см., например, [4]). В то же время, по мнению автора, подобные методики изучения параллельной обработки на базе рассмотрения совместной повседневной деятельности недостаточны. Необходимо дополнительно дать ученикам хотя бы минимальные представления об аппаратных основах параллельных вычислений (особенно в связи с широким распространением многоядерных процессоров), а также об особенностях параллельных алгоритмов по сравнению с традиционными последовательными, широко изучаемыми в рамках школьного курса.

Думается, против сформулированного предложения особых возражений нет. Проблема в том, как его реализовать на практике. Главная методическая трудность заключается в том, что необходимо продемонстрировать школьникам преимущества параллельной обработки на какой-нибудь простой конкретной задаче. Рассказ учителя о расчетах обтекания болидов «Формулы-1» или расшифровке генома впечатляет учеников, но не убеждает; к тому же, не имея представления о методах решения задачи, ученик не может оценить, как именно параллелизм ускоряет ее решение. Возникает некоторое противоречие: преимущества параллельных вычислений наиболее отчетливо видны на сложных задачах, но продемонстрировать суть этих преимуществ неподготовленной аудитории невозможно. Напротив, взяв простую доступную задачу вроде суммирования массива, мы с удивлением обнаруживаем, что по известным техническим причинам ускорение вычислений оказывается далеко не таким значительным, как ожидалось [5, 6]. Попутно заметим, что подобная проблема почти всегда возникает при первоначаль-

ном изучении технологий решения сложных задач, например, основ объектно-ориентированного программирования.

В данной публикации обсуждаются примеры простых задач, подходящих для описанных выше целей. Автору также очень хотелось бы привлечь внимание к проблеме специалистов в области суперкомпьютеров – призвать их спуститься на некоторое время с вершин наук и помочь в разработке материала для изучения основ параллельных вычислений. Это важно, ибо от удачных вводных курсов может зависеть интерес к области параллельных вычислений и дальнейшие успехи в ее освоении.

2. Принципы подбора задач

Хотя принципы эти достаточно очевидны, очень кратко обсудим их.

Типичная вычислительная задача состоит из ввода данных, выполнения вычислений и вывода результатов. В случае параллельного алгоритма отдельно взятый процессор тоже подчиняется этой схеме: получает исходные данные извне, обрабатывает их и передает полученные результаты внешним к нему узлам вычислительной системы. Отсюда немедленно вытекает ряд простых рекомендаций по подбору подходящих задач. Исходных данных должно быть как можно меньше, а в идеале вообще не быть – все величины желательно вычислять по входящим в программу формулам. Результатов тоже хочется видеть как можно меньше (в простейшем случае – единственную величину: сумму, количество удовлетворяющих некоторому условию чисел, максимум и т.п.). Кроме того, удобно, когда все вычисления в каждом процессоре слабо зависят или даже совсем не зависят от результатов обработки данных в других процессорах.

В свете сформулированных требований немедленно приходит в голову задача о численном нахождении определенного интеграла (сумма значений подынтегральной функции) или вычисление площади фигуры методом Монте-Карло (подсчет количества случайных значений, удовлетворяющих некоторым условиям попадания внутрь фигуры). Когда аудитория имеет слабую математическую подготовку, лучше, чтобы объяснять численный метод не приходилось, для чего удобно взять более простую по содержанию задачу вроде подсчета количества «счастливых» билетов [7].

Все названные задачи дают ускорение, близкое к числу используемых процессоров. Тем не менее, существуют причины, по которым хочется показать и другие примеры. Во-первых, таких «удобных» для параллельной обработки задач на практике не так много. А во-вторых, именно грамотное взаимодействие процессоров, а не примитивные вычисления по независимым формулам, представляет наибольший интерес с точки зрения обучения основам параллельного программирования. Именно поэтому далее будут рассмотрены другие, более реалистичные и наглядные примеры учебных задач.

3. Задача о составлении частотного словаря

В книге [8] в качестве примера так называемого фиксированного параллелизма рассмотрена сортировка слов по алфавиту, причем каждый из 26 использованных процессоров упорядочивает слова на определенную латинскую букву. Эта задача была несколько усовершенствована и подробно обсуждалась применительно ко вводу курсу параллельных вычислений в статье [9].

Проблема формулируется следующим образом: составить частотный словарь текста, т.е. алфавитный список всех имеющихся слов с указанием, сколько раз каждое встречается.

Классическое «однопроцессорное» решение состоит в сортировке массива и объединении одинаковых слов. Пусть теперь в нашей воображаемой параллельной вычислительной машине имеется 27 процессоров. Один из них процессор-мастер (ПМ), а остальные, работающие под его управлением, – процессоры-рабочие (ПР). ПМ считывает из текста очередное слово и по первой букве определяет, какому из ПР передать его на обработку. ПР упорядочивают «свои» слова независимо друг от друга по тому же самому алгоритму, а затем все результаты собираются в ПМ.

В [9] подробно обсуждаются особенности параллельного алгоритма решения данной задачи. Кроме того, предложен простой метод оценки ускорения вычислений на основе подсчета

количества наиболее «трудоемких» строковых операций (для грубых оценок обычно рекомендуется именно такой подход [10]). Разработано наглядное программное обеспечение, которое демонстрирует учащимся ход процесса и подсчитывает полученное от параллельности программы ускорение. Убедительно показано, что многопроцессорная обработка существенно уменьшает время вычисления результата.

Тем не менее, при попытке представить удачную, как казалось, методическую разработку на конференцию ПаВТ'2016 возникло неожиданное осложнение. Один из экспертов, прочитав предложенный текст, справедливо отметил, что фактически при переходе к описанному выше 27-процессорному алгоритму использован другой, так называемый блочный или «карманный» алгоритм сортировки. Иными словами, выигрыш во времени обусловлен не только параллельностью программы, но и удачным усовершенствованием метода решения. Справедливости ради следует сказать, что автор видел некоторую некорректность сравнения, и в рецензируемом тексте говорилось о планах «оценить время однопроцессорного алгоритма, который можно написать, глядя на разработанный параллельный алгоритм. Он состоит в следующем: по первой букве слова заносятся методом вставки в один из 26 массивов, а затем все они объединяются в общий результирующий массив.»

Таким образом, возникла дополнительная задача – оценить, какая часть ускорения связана с усовершенствованием алгоритма, а какая – с применением параллельных вычислений.

Результаты оценки с помощью описанного в [9] метода приведены на рис. 1. Пользуясь им, можно сравнить количество строковых операций для трех алгоритмов: однопроцессорного стандартного, однопроцессорного блочного и 27-процессорного. Графики отчетливо показывают, что выигрыш в числе операций из-за перехода к блочному методу существенно больше, чем благодаря многопроцессорности.

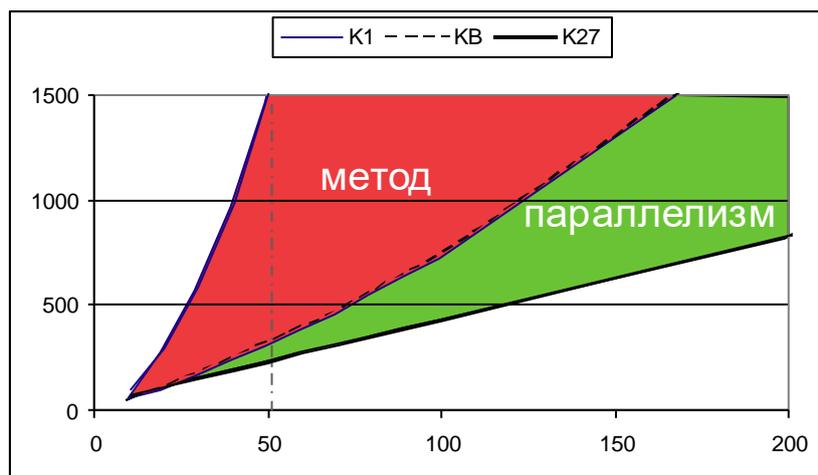


Рис. 1. Зависимость количества операций от числа слов в тексте для однопроцессорных стандартного (K1) и блочного (KB), а также 27-процессорного (K27) алгоритмов

Оказывается, выбранный пример следует признать не самым удачным для демонстрации преимуществ параллельных вычислений.

4. Решение одномерного уравнения теплопроводности

В поисках более убедительного учебного примера, рассмотрим другую задачу – численное решение уравнения теплопроводности. Идея возможности рассмотрения этой задачи была предложена в самом первом школьном учебнике информатики, разработанном большим коллективом авторов под руководством А.П. Ершова и В.М. Монахова [11]. Итоговая расчетная формула для этой задачи оказывается настолько простой и интуитивно понятной, что даже не верится, что она получена из конечно-разностного метода для дифференциального уравнения Лапласа! Для простоты и наглядности ограничимся одномерным случаем уравнения, при этом значение температуры в любой точке есть просто полусумма температур в двух соседних точках. Температуру на границах будем считать фиксированной.

Если для получения параллельного алгоритма разбить расчетную область на несколько частей, то при сопряжении решений в граничных точках в случае распределенной памяти потребуется передавать значения данных между соседними процессорами. Процедура введения для этой цели дополнительных узлов детально разработана (см., например, рис. 9.8 в [12], где рассмотрена даже более сложная ситуация), и мы не будем здесь ее описывать.

Оценка эффективности параллельного решения данной задачи была проведена с помощью предложенной автором ранее модели многопроцессорного учебного компьютера E14 [6, 13]. Модель представляет собой тщательно разработанное и отлаженное программное обеспечение, детально имитирующее работу пяти одинаковых процессоров с простой системой команд, один из которых является главным. В модель заложены различные способы взаимодействия процессоров с оперативной памятью: страничная организация памяти с общей шиной (создание общей памяти) и перенос данных путем межпроцессорного взаимодействия через порты (распределенная память). Описание модели является отдельной темой [5, 13]; здесь ограничимся лишь указанием на то, что для E14 пишется реальная программа на языке низкого уровня, и она весьма правдоподобным образом выполняется на моделирующем ПО. При этом автоматически подсчитывается число выполненных машинных команд для изучаемой программы, что позволяет количественно оценить время ее выполнения. Сравнивая показатели для разных программ (например, сопоставляя результаты для параллельного и классического алгоритмов), легко рассчитать полученное ускорение вычислений.

Кроме того, поскольку мы имеем текст не слишком большой программы из конкретных машинных команд, можно, учитывая число повторений циклов, написать простую формулу, которая позволит вычислять количество выполненных команд программы даже без ее запуска. Некоторая погрешность может здесь появиться из-за наличия циклов ожидания, которые связаны с необходимостью синхронизации частей программы, работающих в разных процессорах. Но, как показывают проверки на моделирующем ПО, различие между тем, что дают формулы, и результатами моделирующей программы, пренебрежимо мало (по крайней мере, для данной задачи).

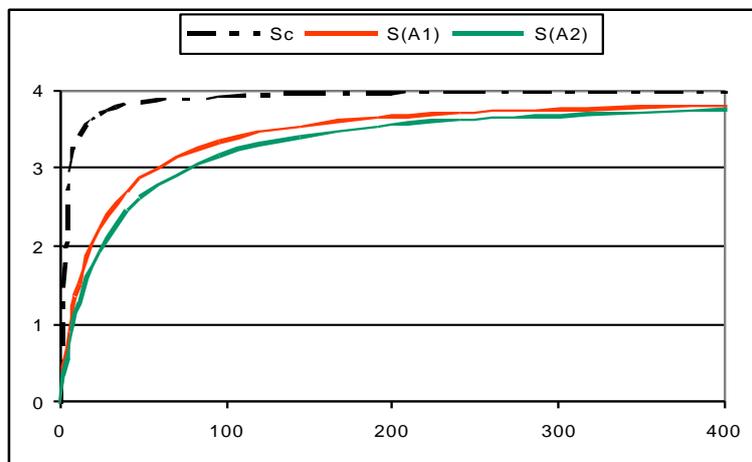


Рис. 2. Зависимость ускорения вычислений от количества расчетных точек для архитектур с общей (A1) и распределенной (A2) памятью; Sc – закон Амдала

В формулу входит число повторений рабочего цикла, которое однозначно связано с количеством расчетных точек N . Таким образом, полученные формулы позволяют проанализировать ситуацию с ускорением вычислений S для всевозможных N даже без проведения массовых расчетов. Результаты для рассматриваемой задачи, описываемой одномерным уравнением теплопроводности, представлены на рис. 2. На нем изображены зависимости $S(N)$ для архитектур с общей и распределенной памятью. Обе кривые очень близки и быстро выходят на асимптоту. Для сравнения на рисунке также приведена кривая классического закона Амдала, построенная по анализу количества команд в последовательной и циклической частях исходной однопроцессорной программы. Из графика очевидным образом следует, что при достаточном количестве расчетных точек удастся получить ускорение S , близкое к числу считающих процессоров. Дополнительно поясним, что пятый процессор E14 непосредственно в вычислениях не участву-

ет: он только организует их (в частности, в алгоритме с распределенной памятью центральный процессор доставляет граничные данные от одного процессора другому).

Итак, результаты моделирования показывают, что выбранная задача хорошо ускоряется, несмотря на обмен данными в ходе вычислений.

5. Заключение

Таким образом, проведенный анализ показывает, что проблема подбора подходящих задач для вводных курсов параллельных вычислений (особенно для школьников) действительно существует. Трудно подобрать задачу, которая одновременно была бы понятной и наглядной и, в то же время, демонстрировала существенное ускорение на многопроцессорной машине. Тестирование показало, что удачным примером может служить задача о численном решении уравнения теплопроводности, если, следуя известному учебнику [11], взять расчетную формулу без вывода.

Если обсуждение на конференции подтвердит, что выбранный пример действительно удачный, автор планирует разработать специализированное учебное программное обеспечение, аналогичное описанному в [9].

Литература

1. Воеводин В.В. Вычислительная математика и структура алгоритмов. М.: Издательство Московского университета, 2006. 112 с.
2. Босова Л.Л. Параллельные алгоритмы в начальной и основной школе // Информатика в школе. 2015. №2. С. 24-27.
3. Сокольская М.А. Содержание курса «Основы параллельного программирования» // Вестник Красноярского государственного педагогического университета. 2011. Том 1, выпуск № 3. С. 180-185.
4. Плаксин М.А. О методике начального знакомства с параллельными вычислениями в средней школе. Информатика в школе: прошлое, настоящее и будущее.: материалы Всеросс. науч.-метод. конф. по вопросам применения ИКТ в образовании. (Пермь, 6-7 февраля 2014 г.). Пермь: ПГНИУ, 2014. С. 256-258.
5. Еремин Е.А. Действительно ли процессор с двумя ядрами сосчитает любую задачу вдвое быстрее? // Информатика. 2015. № 4. С. 12-26.
6. Еремин Е.А. Учебная многопроцессорная модель «Е14» – ПО для изучения основ параллельных вычислений. Параллельные вычислительные технологии (ПаВТ'2015): Труды международной научной конференции (Екатеринбург, 31 марта – 2 апреля 2015 г.). Челябинск: Издательский центр ЮУрГУ, 2015. С. 501.
7. Карабаза П.В. Решение задачи о «счастливых билетах» с помощью параллельных вычислений. // Информатика. 2015. № 4. С.34-39.
8. Лин К., Снайдер Л. Принципы параллельного программирования. М.: Издательство Московского университета, 2013. 408 с.
9. Еремин Е.А. Как можно продемонстрировать школьникам преимущества многопроцессорных систем // Информатика в школе. 2015. № 9. С. 32-36.
10. Левитин А.В. Алгоритмы. Введение в разработку и анализ. М.: Изд. д. Вильямс, 2006. 576 с.
11. Основы информатики и вычислительной техники: Проб. учеб. пособие для сред. учеб. заведений. В 2-х ч. Ч. 1 / А.П. Ершов, В.М. Монахов, С.А. Бешенков и др. М.: Просвещение, 1985. 96 с.
12. Старченко А.В., Берцун В.Н. Методы параллельных вычислений: Учебник. Томск: Издательство Томского университета, 2013. 223 с.

13. Еремин Е.А. О выборе подхода к построению компьютерной поддержки для изучения многопроцессорных систем. // Вестник ПГГПУ. Серия ИКТ в образовании. 2014. Выпуск №.10. С. 5-14.

About selection of demonstrative tasks for an introductory course of parallel computation

E.A. Eremin

Perm State Humanitarian Pedagogical University

There is a problem to select illustrative example during learning of technologies for solving complex tasks including parallel computation technologies. The present paper attempts to find a simple problem for an introductory course, which is suitable for demonstration of parallel processing advantages. Calculation speedup is estimated by means of educational software developed by the author.

Keywords: parallel computation, foundations, introductory course, tasks, school informatics.

References

1. Voevodin V.V. Vychislitel'naja matematika i struktura algoritmov [Calculus Mathematics and Structure of Algorithms]. Moscow, Publishing of the Moscow University, 2006. 112 p.
2. Bosova L.L. Parallelnye algoritmy v nachalnoj i osnovnoj shkole [Parallel Algorithms at Primary and Basic School]. // Informatika v shkole [Informatics at School]. 2015. No. 2. P. 24-27.
3. Sokol'skaya M.A. Soderzhanie kursa «Osnovy parallel'nogo programmirovaniya» [Contents of the course «Foundations of Parallel Programming»] // Vestnik Krasnoyarskogo gosudarstvennogo pedagogicheskogo universiteta [The Bulletin of the Krasnoyarsk State Pedagogical University]. 2011. Vol. 1, No 3. P. 180-185.
4. Plaksin M.A. O metodike nachalnogo znakomstva s parallelnymi vychisleniyami v srednej shkole. [About Methods of Primary Notion with Parallel Calculations at Basic School]. Informatika v shkole: proshloe, nastojashhee i budushhee: materialy Vseross. nauch.-metod. konf. po voprosam primeneniya IKT v obrazovanii (Perm, 6-7 fevralja 2014) [Informatics at School: Past, Present and Future: Proceedings of the All Russian Scientific and Methodical Conference on the Points of Usage of ICT in Education (Perm, Russia, February, 6-7, 2014)]. Perm, Publishing of the Perm State National Research University, 2014. P. 256-258.
5. Eremin E.A. Deystvitel'no li protsessor s dvumya yadrami soschitaet lyubuyu zadachu vdvoe bystree? [Will a Processor with Two Cores Really Calculate Any Task Twice Faster?] // Informatika [Informatics]. 2015. No 4. P. 12-26.
6. Eremin E.A. Uchebnaya mnogoprotsessornaya model' «E14» – PO dlya izucheniya osnov parallel'nykh vychisleniy [Educational Multiprocessor Model «E14» – the Software for Learning Foundations of Parallel Computing]. Parallelnye vychislitel'nye tekhnologii (PaVT'2015): Trudy mezhdunarodnoy nauchnoy konferentsii (Ekaterinburg, 31 marta – 2 aprelya 2015 g.) [Parallel Computational Technologies (PCT'2015): Proceedings of the International Scientific Conference (Yekaterinburg, Russia, March, 31 – April, 2, 2015)]. Chelyabinsk, Publishing of the South Ural State University, 2015. P. 501.
7. Karabaza P.V. Reshenie zadachi o «schastlivykh biletakh» s pomoshch'yu parallel'nykh vychisleniy [The Solution of the Problem about «Lucky Tickets» by Means of Parallel Computing]. // Informatika [Informatics]. 2015. No 4. P. 34-39.
8. Lin C., Snyder L. Principles of Parallel Programming. Addison-Wesley, 2008. 352 p.
9. Eremin E.A. Kak mozjno prodemonstrirovat' shkol'nikam preimushchestva mnogoprotsessornykh sistem [How We Can Demonstrate the Advantages of Multiprocessor Systems to Schoolchildren] // Informatika v shkole [Informatics at School]. 2015. No. 9. P. 32-36.

10. Levitin A.V. Introduction to the Design and Analysis of Algorithms. Addison-Wesley, 2006. 592 p.
11. Osnovy informatiki i vychislitel'noy tekhniki: Prob. ucheb. posobie dlya sred. ucheb. zavedeniy [The Foundations of Informatics and Computing Machinery: Trial School-book for Middle Schools]. In 2 parts. P. 1 / A.P. Ershov, V.M Monakhov, S.A. Beshenkov et al. Moscow, Prosveshchenie [Education], 1985. 96 p.
12. Starchenko A.V., Bertsun V.N. Metody parallel'nykh vychisleniy: Uchebnik [Methods of Parallel Computing: Textbook]. Tomsk, Publishing of the Tomsk University, 2013. 223 p.
13. Eremin E.A. O vybore podkhoda k postroeniyu komp'yuternoy podderzhki dlya izucheniya mnogoprotsessornykh sistem. [About the Choice of Approach to Design of Computer Support for Learning of Multiprocessor Systems] // Vestnik PGGU. Seriya IKT v obrazovanii [The Bulletin of The Perm State Humanitarian Pedagogical University. ICT in Education Series]. 2014. No. 10. P. 5-14.