Моей жене Пегги, которая поддерживала не только мой путь в области высокопроизводительных вычислений, но и нашего сына Джона и дочь Рейчел. Научное программирование далеко от ее медицинских знаний, но она сопровождала меня и совершила это наше путешествие с самого начала. Моему сыну Джону и дочери Рейчел, которые вновь разожгли во мне пламя, и за ваше многообещающее будущее.

– Боб Роби

Моему мужу Рику, который поддерживал меня всю дорогу, спасибо, что брал на себя утренние смены и позволял мне работать по ночам. Ты никогда не позволял мне отказываться от самой себя. Моим родителям и родственникам, спасибо за всю вашу помощь и поддержку. И моему сыну Дереку за то, что он был одним из моих самых больших вдохновителей; ты – вся причина, почему я не просто живу, я наслаждаюсь жизнью.

– Джули Замора

Оглавление

Содержание

Предисловие

От авторов

Боб Роби, Лос-Аламос, Нью-Мексико

> *Выходить за дверь – опасно, Фродо. Выходишь на дорогу, и, если не удержаться на ногах, неизвестно, куда тебя унесет.*
>
> – Бильбо Бэггинс

Я не мог предвидеть, куда нас приведет это путешествие в параллельные вычисления. Я говорю «мы», потому что на протяжении многих лет это путешествие делили со мной многочисленные коллеги. Мое же путешествие в параллельные вычисления началось в начале 1990-х годов, когда я учился в Университете Нью-Мексико. Я написал несколько программ по динамике сжимаемой жидкости для симулирования экспериментов с ударной трубкой и запускал их в каждой системе, которая попадала мне в руки. В результате меня вместе с Брайаном Смитом (Brian Smith), Джоном Соболевски (John Sobolewski) и Фрэнком Гилфезером (Frank Gilfeather) попросили представить предложение по созданию центра высокопроизводительных вычислений. Мы выиграли грант и в 1993 году создали Центр высокопроизводительных вычислений Maui Мое участие в проекте состояло в том, чтобы предлагать курсы и возглавлять подготовку 20 аспирантов по разработке параллельных вычислений в Университете Нью-Мексико в Альбукерке.

1990-е годы были временем становления параллельных вычислений. Я помню выступление Эла Гейста (Al Geist), одного из первых разработчиков параллельной виртуальной машины (PVM) и члена комитета по стандартам MPI[1], как говорил о готовящемся к выпуску стандарте MPI

[1] Message Passing Interface (MPI) – программный интерфейс для передачи информации, который позволяет обмениваться сообщениями между процессами, выполняющими одну задачу. – *Прим. перев.*

(июнь 1994 года). Он сказал тогда, что он никуда не приведет, потому что он слишком сложен. Насчет сложности Эл был прав, но, несмотря на это, он взмыл ввысь, и в течение нескольких месяцев он уже применялся почти во всех параллельных приложениях. Одна из причин успеха MPI заключается в наличии готовых имплементаций. Аргоннская национальная лаборатория разрабатывала Chameleon, инструмент переносимости, который в то время транслировал языки передачи сообщений, включая P4, PVM, MPL и многие другие. Этот проект был быстро изменен на MPICH, став первой высококачественной имплементацией MPI. На протяжении более десяти лет MPI стал синонимом параллельных вычислений. Почти каждое параллельное приложение было построено поверх библиотек MPI.

Теперь давайте перенесемся в 2010 год и увидим появление графических процессоров, GPU. Как-то я наткнулся на статью в журнале Dr. Dobb об использовании суммы Кахана для компенсировпания единственной арифметики с одинарной точностью, доступной на GPU. Я подумал, что, возможно, этот подход мог бы помочь решить давнюю проблему параллельных вычислений, когда глобальная сумма массива меняется в зависимости от числа процессоров. Намереваясь проверить это на практике, я подумал о программном коде по гидродинамике, который мой сын Джон написал в средней школе. Он тестировал сохранение массы и энергии в задаче с течением времени и останавливался, выходя из программы, если масса изменялась более чем на указанную величину. Пока он был дома на весенних каникулах после своего первого курса в Вашингтонском университете, мы опробовали этот метод и были приятно удивлены тем, насколько улучшилось сохранение массы. Для производственных исходных кодов влияние этого простого технического приема окажется важным. Мы рассмотрим алгоритм суммирования с повышенной точностью для параллельных глобальных сумм в разделе 5.7 этой книги.

В 2011 году я организовал летний проект с тремя студентами, Нилом Дэвисом (Neal Davis), Дэвидом Николаеффом (David Nicholaeff) и Деннисом Трухильо (Dennis Trujillo), чтобы попробовать, сможем ли разработать более сложные программные коды, такие как адаптивная детализация расчетной сетки (AMR) и неструктурированные произвольные лагранжево–эйлеровы (ALE) приложения для работы на GPU. Результатом стало CLAMR, мини-приложение AMR, которое полностью работало на GPU. Бóльшая часть приложения была легко переносима. Самой сложной его частью было определение соседа для каждой ячейки. В изначальном CPU-коде использовался алгоритм k-D-дерева, но алгоритмы, основанные на дереве, трудно переносятся на GPU. Через две недели после начала летнего проекта на холмах над Лос-Аламосом вспыхнул пожар в Лас-Кончасе, и город был эвакуирован. Мы уехали в Санта-Фе, и студенты разбежались. Во время эвакуации я встретился с Дэвидом Николаеффом в центре Санта-Фе, чтобы обсудить вопрос переноса на GPU. Он предложил нам попробовать использовать алгоритм хеширования, чтобы заменить древесный исходный код отыскания соседей. В то время я наблюдал за пылающим над городом огнем, и меня не отпускало

беспокойство, что огонь доберется до моего дома. Несмотря на все эти перипетии, я согласился попробовать, и алгоритм хеширования привел к тому, что весь код был запущен на GPU. Техника хеширования была обобщена Дэвидом, моей дочерью Рейчел, когда она училась в средней школе, и мной. Эти алгоритмы хеширования формируют основу для многих алгоритмов, представленных в главе 5.

В последующие годы технические приемы компактного хеширования были разработаны Ребеккой Тамблин (Rebecka Tumblin), Питером Аренсом (Peter Ahrens) и Сарой Харце (Sara Hartse). Более сложная задача компактного хеширования для операций перекомпоновывания (remapping) на CPU и GPU была решена Джеральдом Коломом (Gerald Collom) и Колином Редманом (Colin Redman), когда они только что закончили среднюю школу. Этими прорывами в параллельных алгоритмах для GPU были устранены препятствия для выполнения многих научных приложений на GPU.

В 2016 году я начал программу летней исследовательской стажировки по параллельным вычислениям (PCSRI) в Национальной лаборатории Лос-Аламоса (LANL) вместе с моими соучредителями Хай А Нам (Hai Ah Nam) и Гейбом Рокфеллером (Gabe Rockefeller). Целью программы параллельных вычислений было решение проблемы растущей сложности систем высокопроизводительных вычислений. Программа представляет собой 10-недельную летнюю стажировку с лекциями по различным темам параллельных вычислений с последующим исследовательским проектом под руководством сотрудников Национальной лаборатории Лос-Аламоса. У нас в летней программе участвовало от 12 до 18 студентов, и многие использовали ее в качестве трамплина для своей карьеры. С помощью этой программы мы продолжаем решать некоторые новейшие задачи, стоящие перед параллельными и высокопроизводительными вычислениями.

Джули Замора, Чикагский университет, Иллинойс

> *Если есть книга, которую вы хотите прочитать, но она еще не написана, то вы должны ее написать.*
>
> – Тони Моррисон

Мое знакомство с параллельными вычислениями началось со слов: «Перед тем как начнете, зайдите в комнату в конце 4-го этажа и установите вот эти процессоры Knights Corner в нашем кластере». Эта просьба профессора Корнельского университета побудила меня попробовать что-то новое. То, что я считала простым делом, превратилось в бурное путешествие в высокопроизводительные вычисления. Я начала с изучения основ – с выяснения физического принципа функционирования малого кластера, поднимая 40-фунтовые сервера для работы с BIOS и выполнения моего первого приложения, а затем оптимизации этих приложений по всем установленным мной узлам.

После короткого семейного перерыва, каким бы обескураживающим он ни был, я подала заявление на научно-исследовательскую стажиров-

ку. Будучи принятым в первую программу летней исследовательской стажировки по параллельным вычислениям в Нью-Мексико, я получила возможность разведать тонкости параллельных вычислений на современном оборудовании, и именно там я познакомилась с Бобом. Я пришла в восторг от повышения производительности, которое было возможно при наличии лишь небольших знаний о правильном написании параллельного кода. Я лично провела разведывательный анализ процесса написания более эффективного кода OpenMP. Мое волнение и прогресс в оптимизации приложений открыли двери для других возможностей, таких как участие в конференциях и представление моей работы на собрании группы пользователей Intel и на стенде Intel по суперкомпьютерным вычислениям. В 2017 году меня также пригласили принять участие и выступить на конференции в Салишане. Это была прекрасная возможность обменяться идеями с несколькими ведущими провидцами высокопроизводительных вычислений.

Еще одним замечательным опытом была подача заявки на участие в хакатоне GPU и участие в нем. На хакатоне мы перенесли код на OpenACC, и в течение недели этот код был ускорен в 60 раз. Вы только подумайте – расчет, который раньше занимал месяц, теперь может выполняться за одну ночь. Полностью погрузившись в потенциал долгосрочных исследований, я подала заявление в аспирантуру и выбрала Чикагский университет, осознавая его тесную связь с Аргоннской национальной лабораторией. В Чикагском университете меня консультировали Ян Фостер (Ian Foster) и Генри Хоффман (Henry Hoffmann).

Из своего опыта я поняла, насколько ценным является личное взаимодействие, когда учишься писать параллельный код. Я также была разочарована отсутствием учебника или справочника, в котором обсуждалось бы текущее оборудование. В целях восполнения этого пробела мы написали эту книгу, чтобы сделать ее намного проще для тех, кто только начинает свою деятельность в параллельных и высокопроизводительных вычислениях. Решение задачи создания и преподавания введения в информатику для поступающих студентов Чикагского университета помогло мне получить представление о тех, кто знакомится с этой областью впервые. С другой стороны, объяснение технических приемов параллельного программирования в качестве ассистента преподавателя в курсе «Продвинутые распределенные системы» позволило мне работать со студентами с более высоким уровнем понимания. Оба этих опыта помогли мне обрести способность объяснять сложные темы на разных уровнях.

Я считаю, что у каждого должна быть возможность изучить этот важный материал по написанию высокопроизводительного кода и что он должен быть легко доступен для всех. Мне посчастливилось иметь наставников и советников, которые направляли меня по правильным ссылкам на веб-сайты или передавали мне свои старые рукописи для чтения и изучения. Хотя некоторые технические приемы, возможно, будут сложными, более серьезной проблемой является отсутствие согласованной документации или доступа к ведущим ученым в этой области в качестве наставников. Я понимаю, что не у всех есть одинаковые ре-

сурсы, и поэтому я надеюсь, что создание этой книги заполнит пустоту, которая существует в настоящее время.

Как мы пришли к написанию этой книги

Начиная с 2016 года команда ученых LANL во главе с Бобом Роби разработала лекционные материалы для летней исследовательской стажировки по параллельным вычислениям (PCSRI) в Лос-Аламосской национальной лаборатории (LANL). Большая часть этого материала посвящена новейшему оборудованию, которое быстро выходит на рынок. Параллельные вычисления меняются быстрыми темпами, и к ним прилагается мало документации. Книга, охватывающая эти материалы, была явно необходима. Именно в этот момент издательство Manning связалось с Бобом по поводу написания книги о параллельных вычислениях. У нас был лишь черновой набросок материалов, и работа над ним вылилась в двухлетние усилия по переводу всего этого в формат высокого качества.

Темы и план глав были четко определены на ранней стадии и основывались на лекциях для нашей летней программы. Многие идеи и технические приемы были позаимствованы из более широкого сообщества высокопроизводительных вычислений, поскольку мы стремимся к более высокому уровню вычислений – тысячекратному повышению производительности вычислений по сравнению с предыдущим эталоном петафлопсного масштаба. Это сообщество включает Центры передового опыта Министерства энергетики (DOE), проект по экзомасштабным вычислениям (Exascale Computing Project) и серию семинаров по производительности, переносимости и производительности. Широта и глубина материалов наших компьютерных лекций отражают глубокие проблемы сложных гетерогенных вычислительных архитектур.

Мы называем материал этой книги «глубоким введением». Она начинается с основ параллельных и высокопроизводительных вычислений, но без знания вычислительной архитектуры невозможно достичь оптимальной производительности. По ходу дела мы стараемся осветить проблематику на более глубоком уровне понимания, потому что недостаточно просто двигаться по тропе, не имея ни малейшего представления о том, где вы находитесь или куда направляетесь. Мы предоставляем инструменты для разработки карты и для того, чтобы показать удаленность цели, к которой мы стремимся.

В начале этой книги Джо Шоновер (Joe Schoonover) был привлечен для написания материалов, связанных с GPU, а Джули Замора – главы по OpenMP. Джо предоставил дизайн и компоновку разделов по GPU, но ему пришлось быстро отказаться. Джули написала статьи и предоставила массу иллюстраций о том, как OpenMP вписывается в этот дивный новый мир масштабных вычислений, поэтому указанный материал особенно хорошо подошел для главы книги об OpenMP. Глубокое понимание Джули проблем, связанных с экзомасштабными вычислениями, и ее способность разбирать их по полочкам для новичков в этой области стали решающим вкладом в создание этой книги.

Благодарности

Мы хотели бы поблагодарить всех, кто помог сформировать эту книгу. Первым в нашем списке стоит Джо Шоновер из Fluid Dynamics (Гидродинамики жидкости), который сделал все возможное в преподавании параллельных вычислений, в особенности с GPU-процессорами. Джо был одним из соруководителей нашей программы параллельных вычислений и сыграл важную роль в формулировании того, что должна охватывать эта книга. Другие наши соучредители, Хай А Нам, Гейб Рокфеллер, Крис Гарретт, Юнмо Ку, Люк Ван Рокель, Роберт Берд, Джонас Липпунер и Мэтт Тернер, внесли свой собственный вклад в успех школы параллельных вычислений и ее содержание. Создание летней программы параллельных вычислений не произошло бы без поддержки и видения директора института, Стефана Эйденбенца. Также спасибо Скотту Раннелсу и Даниэлю Израэлю, которые возглавили летнюю школу вычислительной физики LANL и стали пионерами концепции школы, дав нам модель для подражания.

Нам повезло, что нас окружают эксперты в области параллельных вычислений и книгоиздания. Благодарим Кейт Боуман, чей опыт в написании книг помогал вносить изменения в первые главы. Кейт – невероятно талантлива во всех аспектах издательского дела и уже много лет занимается составлением предметных указателей книг. У нас также были неофициальные отзывы от сына Боба Джона, дочери Рэйчел и зятя Боба Берда, часть технической работы каждого из которых упоминается в книге. Муж Джули, Рик, помог предоставить экспертные знания по некоторым темам, а Дов Шлахтер рассмотрел некоторые ранние наброски и предоставил несколько полезных отзывов.

Мы также хотели бы отметить опыт сотрудников, которые приложили свои знания в конкретных главах. Сюда входят Рао Гаримелла и Шейн Фогерти из Лос-Аламоса и Мэтт Мартино из Национальной лаборатории Лоуренса Ливермора, чья работа включена в главу 4. Особая благодарность выражается инновационным работам упомянутых ранее многих студентов, чья работа занимает бо́льшую часть главы 5. Рон Грин из Intel в течение нескольких лет руководил усилиями по документированию