

ФИЗИКА НАШИХ ДНЕЙ

50/54+681.142

ТЕОРЕТИЧЕСКАЯ ФИЗИКА И РАСЧЕТЫ НА ЭВМ*)*К. Г. Вильсон*

Необычные изменения происходят в деловом мире. Их причиной является совместное наступление Японии и компьютеров. Эти изменения не всегда, однако, бывают замечены в научных и правительственных кругах, обладающих менее быстрой реакцией.

Характерное время, необходимое для разработки идеи и ее внедрения, быстро уменьшается. Примером разработки и внедрения по старинке могут служить лазеры, изобретенные более двадцати лет назад. Сейчас назревают революционные изменения в области передачи сообщений в связи с предполагаемым использованием для этих целей лазеров и волоконной оптики. За этот двадцатилетний период лазеры из объекта любопытства ученых стали объектом стандартных промышленных Р и В**). Однако сейчас (и в особенности в области компьютерной техники) никто уже не имеет двадцати лет на Р и В, а всего лишь примерно от трех до пяти лет.

Изделие «живет» три-пять лет, а потом опять «возвращается на чертежную доску». Такие условия означают, что Р и В, опирающиеся на хорошо изученные явления, уже несовременны. Для успешного продвижения в новых областях необходимо более глубокое понимание научных проблем.

Стандартная ситуация в промышленности состоит в том, что все в производстве является секретом. Однако сейчас для того, чтобы успешно развивать производство, компаниям необходимо иметь заблаговременную информацию о новых разработках, и это становится более важным, чем обеспечение секретности. Промышленность должна теперь обеспечить контакт своих представителей с внешним миром для того, чтобы обеспечить приток идей.

Эти изменения означают, что роль науки и ученых в обществе становится совсем иной, чем прежде. Научные разработки должны быть в гораздо большей степени связаны с производством, поскольку промышленность ищет информацию о научных разработках и использует ее.

Первой областью, в которой должен установиться такой порядок вещей, является производство ЭВМ, поскольку именно в этой области характерное время Р и В наиболее мало.

Каким образом внедрение компьютеров повлияет на фундаментальные разработки? В физике элементарных частиц экспериментаторам приходится

*) Wilson K. G. Theoretical Science and the Future of Large Scale Computing. — CERN Courier, 1983, v. 23, pp. 172—177. — Перевод А. В. Леонидова.

Кеннет Г. Вильсон — профессор физики Корнеллского университета, США, лауреат Нобелевской премии по физике за 1982 г.

**) Так автор здесь и далее обозначает разработку и внедрение. (Прим. перев.)

© CERN Courier 1983.

© Перевод на русский язык, издательство «Наука». Главная редакция физико-математической литературы, «Успехи физических наук», 1984.

анализировать события, в которых участвуют сотни частиц. Розыгрыш по Монте-Карло таких событий «съедает», естественно, большую часть мощности ЭВМ (с учетом закона сохранения энергии, благодаря которому можно рассматривать сотни частиц, а не миллионы). В процессе машинного эксперимента теоретикам приходится разыгрывать такие события на очень малых временных интервалах. Свойства системы, состоящей из сотен частиц, уже невозможно описать аналитически простой теорией вроде теории атома водорода или теории движения Земли вокруг Солнца. Остается полагаться, по крайней мере частично, на розыгрыш событий на ЭВМ.

При компьютерном розыгрыше теоретик должен интересоваться не столько окончательным результатом, сколько главными факторами, определяющими тот или иной результат. На очень коротком временном интервале сохранение энергии уже не ограничивает число участвующих частиц. Например, при изучении короткоживущих глюонов в протоне или нейтроне, рассмотреть десять или пятьдесят глюонов недостаточно. Для того чтобы иметь конечное число глюонов, необходимо на самом деле вводить решетку. Но даже конечная решетка внутри протона включает тысячи, если не миллионы узлов, и глюоны должны присутствовать в каждом из них. Быстродействие компьютеров, необходимое для этих теоретических вычислений (это по-прежнему розыгрыш по Монте-Карло) намного больше, чем у соответствующей экспериментальной аппаратуры, просто потому, что число объектов, участвующих в машинном эксперименте, намного больше числа объектов в реальном эксперименте. Именно поэтому я начал размышлять о том, как оправдать привлечение колоссальных компьютерных мощностей для нужд теории.

Когда я определил, в каком компьютере нуждаюсь, я обнаружил, что такого компьютера еще никто не сделал. Так что первая проблема состояла не в том, что у меня не было денег, а в том, что мне не на что было их потратить. Размышляя в этом направлении, я постепенно пришел к мысли дать общий обзор ситуации, сложившейся вокруг ЭВМ не только в университетах, но и в промышленности, где в них наблюдается большая нужда.

Одной из проблем является обучение. Среди физиков бытует убеждение, что для того чтобы всерьез работать с ЭВМ — выполнять численный теоретический эксперимент или анализировать экспериментальные данные, достаточно двухнедельного курса обучения ФОРТРАНу. Сегодняшний численный эксперимент, проводимый теоретиком, очень близок по духу реальному эксперименту. Результат машинного эксперимента является не функцией или идеей, а числом. Так что для начала нужно понимать, как поставить численный эксперимент, результат которого представляет интерес. Это эквивалентно постановке реального эксперимента и определению всех его параметров. Необходимо анализировать выходные данные и иметь дело с проблемами, аналогичными встречающимся при обработке экспериментальных данных.

Никому не придет в голову утверждать, что человек готов к серьезной работе в экспериментальной физике, если он окончил двухнедельные курсы пайки. И тем не менее убеждение в том, что двухнедельного курса ФОРТРАНа достаточно для серьезной работы на ЭВМ, существует!

Какое же обучение необходимо? Обучение должно быть двухступенчатым. На первой стадии оно похоже на обучение эксперименту — несколько лет практической работы в процессе написания диссертации под руководством опытного специалиста, который знает, как поставить эксперимент, как анализировать экспериментальные данные, как убедиться в отсутствии систематических ошибок, искажающих результат, и т. д. Очевидно, что желающие заниматься машинным экспериментом должны пройти такого рода обучение.

На второй стадии необходимо практиковаться в составлении и отладке больших программ. Эта необходимость до сих пор игнорировалась в науч-

ных кругах. В то же время научный мир, как и деловой мир и сотрудники национальных лабораторий, сооружают небоскреб из огромных программ на ФОРТРАНе, которые невозможно прочесть и модифицировать, что существенно тормозит научный прогресс. Это поглощает огромное количество рабочего времени, которое лучше было бы потратить на другие цели, и направляет исследования в жестко фиксированном направлении, в то время как на самом деле хотелось бы сделать нечто другое!

В шестидесятые годы в промышленности и национальных лабораториях создавалось много больших программ такого рода. При этом не очень-то занимались осмыслением происходящего. Около 1971 г. было достигнуто насыщение. В национальных лабораториях США сейчас найдется немного программ, написанных в течение последних десяти лет.

Реализация надежд, возлагаемых на компьютеры, существенно зависит от того, как будет решаться эта проблема. Как писать большие куски программы так, чтобы понимать, что получится в результате и как с этой программой работать? Вместо того, чтобы писать программы на специально придуманных языках, специалисты по ЭВМ обсуждают сейчас проблему создания программного обеспечения, которое допускает использование любого языка. Тогда можно будет при помощи единого алгоритма использовать накопленную за 20 лет информацию.

Каждый дипломник, который готовится использовать ЭВМ в процессе написания диссертации, должен пройти примерно трехмесячный курс обучения работе с ЭВМ. Обучение работе с ЭВМ должно быть также тесно связано с учебными планами колледжей как в период преддипломной практики студентов, так и в период их дипломной работы. Но это означает, что нужно изменить язык или способ программирования, поскольку студент не в состоянии реализовать программу на ФОРТРАНе за время, отведенное на обучение.

Есть еще проблема создания единой вычислительной сети. Многие физики хотели бы, не покидая места основной работы, проводить эксперименты где-нибудь еще, обмениваться информацией, данными, программами с коллегами из других институтов и т. д.

У проблемы единой вычислительной сети есть еще один аспект, с которым незнакомы многие ученые. Это междисциплинарное общение. Например, инженер в Корнелле готовил курс по конструированию ЭВМ. Ему необходимо было описать восемь различных типов компьютеров, а он имел информацию только о четырех из них. Тогда он послал запрос через систему ARPANET с просьбой сообщить информацию об остальных четырех компьютерах. Специалисты по этим четырем компьютерам ответили ему через ARPANET.

Чем больше физики обращаются к использованию компьютеров, тем ближе они к решению своих задач «исходя из первых принципов» *). Это справедливо не только для физики элементарных частиц, где так или иначе используются первопринципы, но и для физики твердого тела, прикладной физики, химии, геохимии и многих областей в инженерном деле. Специалистам, работающим в этих областях, необходимо обмениваться между собой информацией, чтобы не изобретать несколько раз одно и то же. Единая компьютерная сеть нужна для того, чтобы сделать такой обмен информацией возможным.

Это особенно важно для того, чтобы сделать контакты между научным миром и производством более тесными, поскольку производственные Р и В превращаются из замкнутых операций, не использующих информации извне, в открытые операции, для которых такая информация необходима. Специалисты, занятые на производстве, могут с помощью единой вычислительной системы запрашивать информацию извне. Я оказываю сильное

*) Имеется в виду возможность точного численного решения исходных уравнений. (Прим. перев.)

давление на Вашингтон по поводу создания единой вычислительной системы для научных работников.

Следующая тема — программное обеспечение. Проблема программного обеспечения привела в конце концов к застою в развитии как в больших лабораториях, так и на производстве. Я думаю, что это послужило главной причиной того, что идея конструирования суперкомпьютера типа «Крэй» возникла не сразу. Если бы эта идея была реализована темпами, характерными для 60-х годов, суперкомпьютеры были бы сейчас общепотребительными.

Проблема программного обеспечения выражается одним термином — ФОРТРАН. ФОРТРАН ограничивает возможности, поскольку существует предел сложности для программ, написанных на этом языке. У ФОРТРАНа есть две проблемы. Первая заключается в том, что на нем нельзя читать, а вторая — в том, что его нельзя модифицировать. Специалисты по ЭВМ жалуются на это давно, но их усилия не привели пока к осязаемым результатам. Сначала они изобрели АЛГОЛ, потом — PL/1. Потом PASCAL. Но насколько я понимаю, ни в одном из этих языков проблемы ФОРТРАНа не решены.

Существо проблемы иллюстрируется следующей аналогией: 60-страничная программа, написанная на ФОРТРАНе, содержит примерно такое же количество информации, как и учебник повышенного уровня. Чтобы переписать информацию, содержащуюся в учебнике, на ФОРТРАН, надо переставить в нем все слова. Если, например, графически изобразить реализацию большой программы на ФОРТРАНе, то график будет метаться по всей программе. Можно ли с этим что-нибудь поделать? Я попытался понять, какие идеи могли бы оказаться действительно полезными, и убежден, что такие идеи существуют и, будучи должным образом реализованы, помогут заменить ФОРТРАН. Эти идеи, однако, не используются для научных расчетов. Они используются в других областях — таких, как обработка данных или операционные системы, поскольку ученые заявили, что они не нуждаются ни в чем, кроме ФОРТРАНа, и специалисты по ЭВМ поверили им на слово. Кроме того, практическая реализация этих идей требует специалиста, который хорошо понимает, в чем именно нуждаются ученые, использующие ЭВМ. Этого специалисты по ЭВМ не знают.

Мне хочется высказать свои соображения по поводу того, как можно было бы применить некоторые идеи теории ЭВМ к программированию в научных целях. Задача состоит не в том, чтобы избавиться от ФОРТРАНа, а в том, чтобы сделать его языком машины, а не человека. При этом не имеет значения, что такой язык неудобочитаем, поскольку его не придется читать. Или не стоит!

Имея в виду тысячелетний опыт, накопленный человечеством, специалисты по ЭВМ ищут пути усовершенствования организации вычислений. В качестве примера рассмотрим учебник как хранилище информации, организованной так, что ее можно из него извлечь (т. е. прочесть). Предположим, что мы разрабатываем программу, организованную подобно учебнику. Тогда глава 1 содержит уравнения, которые мы хотим решать, глава 2 — описание численных методов, использованных для решения этих уравнений. В главе 3 приводятся граничные условия, а в главе 4 описывается структура выходных данных. Кроме того, необходимо оптимально организовать программу с учетом типа компьютера. В стандартной же программе написанной на ФОРТРАНе, операции из каждой главы используются многократно.

Как могла бы работать программа, составленная из независимых глав? Один из способов — определить в конце второй главы после кодирования методов вычисления специальное преобразование, «подставляющее» этот код в уравнения из первой главы. Такое преобразование означает, например, что знак интеграла заменяется определенным числовым символом.

Уже существуют системы, которые выполняют преобразования, близкие к описанному выше. Преимущество применения таких преобразований в том, что уравнения из первой главы являются частью окончательной программы, так что если изменить программу, не меняя уравнений, то она не будет работать. Уравнения должны быть жестко связаны с программой. Если уравнения записаны правильно и красиво, но не являются частью программы, то эти уравнения реализуются в программах старого типа, а не таких, какие мы обсуждаем.

В создании систем, предназначенных для обслуживания научных исследований, должны участвовать ученые, так что я организовал в Корнелле проект GIBBS и буду его руководителем. Все остальные участники проекта будут специалистами по ЭВМ. Я надеюсь на то, что появятся и другие проекты такого рода, поскольку любая отдельно взятая разработка имеет высокую вероятность провалиться, как это произошло с операционными системами.

И, наконец, существует проблема машинного обеспечения. Каким образом можно достичь необходимой для исследования теоретических и, конечно, экспериментальных вопросов мощности компьютеров? Основным направлением в развитии элементной базы является не увеличение быстродействия, а удешевление деталей. Для получения максимальной мощности необходимо иметь все большее и большее число параллельно работающих компьютеров, а не один компьютер, работающий все быстрее и быстрее.

В промышленных кругах все больше распространяется убеждение в необходимости продумать применение параллельных процессоров на всех уровнях. Существует уникальная возможность для ученых включиться в процесс создания таких систем на ранней стадии их разработки, когда для того, чтобы такая система работала, необходим специалист с докторской степенью по физике. Ученые станут тогда «копилкой» возникающих проблем. Когда в промышленности увидят, какие трудности возникают у ученых, появится возможность анализа их причин и нового развития. Что необходимо промышленности, так это сделать нечто такое, чем человек со степенью доктора физики может управлять спросонья. В этом случае управлять такой системой сможет и любой прохожий. Чтобы проиллюстрировать работу различных систем с параллельно включенными процессорами, обратимся опять к аналогии, именно, к обслуживанию пассажиров в аэропортах. Первая возможность, не слишком привлекательная — иметь за стойкой одного служащего. Если он не может справиться со своими обязанностями, разрабатывается проект служащего с четырьмя руками!

Вторым направлением является создание системы с параллельными процессорами — это так называемый векторный суперкомпьютер. В этом случае мы имеем дело с несколькими служащими за стойкой и очередью пассажиров. Первый пассажир подходит к первому служащему, получает билет и направляется ко второму служащему, который проставляет в билете имя и фамилию пассажира. Третий служащий отмечает в билете место назначения. Описанная система аналогична сборочному конвейеру.

Такая форма организации хороша, пока пассажиры нуждаются только в оформлении билетов. Но что произойдет, если один из них захочет выяснить, где затерялся его багаж? Если это произойдет, конвейер остановится на то время, которое один из служащих затратит на поиски исчезнувшего багажа, и только когда он закончит, конвейер вновь сможет оформлять билеты.

Некоторые типы суперкомпьютеров обладают именно этим недостатком. Они чрезвычайно быстро осуществляют операции умножения и деления, но сложные вычисления с переадресацией проделываются гораздо медленнее (в специально отведенной для этого системе).

Еще один способ организации вычислений впервые был реализован в «Illiac-4» и теперь применяется в нескольких коммерческих образцах. При этой системе пассажиры выстраиваются в линию вдоль стойки, за

которой находятся служащие, и ждут указаний служащего с мегафоном. Он командует в мегафон: «Пассажирам подойти к стойке», — и они подходят. Затем он командует: «Достать билеты!» — и затем выкрикивает в мегафон инструкцию по их заполнению. Этот способ, так же как и конвейерный, очень хорош для заполнения билетов, но страдает тем же недостатком. Как только один из пассажиров начнет поиски исчезнувшего багажа, один из служащих будет заниматься этим вопросом, подчиняясь указаниям клерка с мегафоном.

В реальных ЭВМ процессоры упорядочены так, что информация от процессора поступает только ближайшему соседу. Это наиболее часто используемая форма связи процессоров, обеспечивающая наибольшую быстроту передачи информации. Для регулярных проблем типа расчетов по Монте-Карло в решеточных калибровочных теориях такая организация вычислений чрезвычайно эффективна. Но для расчета явлений типа распространения поверхностей разрыва в динамике жидкости вычисления, производимые в разных процессорах, могут сильно отличаться.

Ни один из упомянутых способов, однако, не кажется подходящим для детального анализа экспериментальных данных. Недавно появилась новая, более многообещающая система с параллельными процессорами, однако единственный существующий образец стоит очень дорого и пока не слишком выгоден коммерчески. В ней использованы те же принципы, какие используются при наиболее эффективной продаже авиационных билетов. Очередной пассажир подходит к любому из освободившихся служащих. Каждый из служащих может выполнять любые операции независимо от остальных. Существенной особенностью этой системы является возможность одновременного расчета по многим подпрограммам. Любой из процессоров способен длительное время работать автономно, прежде чем он свяжется с остальной системой. Это обстоятельство имеет огромное значение, поскольку при изучении систем с параллельными процессорами ученые неизменно сталкивались с тем, что время, которое затрачивается на передачу сообщения из одного процессора в другой, оказывается намного больше, чем этого хотелось бы.

Что необходимо для системы с параллельными процессорами, так это сокращение числа контактов процессоров друг с другом. Прежде чем возникнет необходимость в обмене информацией, каждый из процессоров должен выполнить как можно больший объем работы.

Заканчивая, я хотел бы рассказать о варианте системы с параллельными процессорами, разработанном в Нью-Йоркском университете. Этот компьютер пока еще не реализован практически. Были проверены только основные концепции. Пусть за стойкой находятся 4000 клерков. Очередь из пассажиров может быть похожа на бутылочное горлышко, и они не смогут достаточно быстро добраться до служащих. Суть нового подхода в том, что когда один из клерков заканчивает обслуживание пассажира, он выходит из-за стойки и направляется к очереди. Если два клерка сталкиваются на пути к очереди, один из них останавливается, а второй приглашает для оформления билетов сразу двух пассажиров — к себе и к своему товарищу.

Если служащих 4000, то тот, кто подходит к очереди, приглашает в среднем от 32 до 64 пассажиров и направляет их к своим коллегам.

«Ультракомпьютер» имеет 4000 процессоров, 4000 запоминающих устройств и сеть соединительных проводов и узлов, обеспечивающих возможность связи любого процессора с любым запоминающим устройством. Если два процессора запрашивают одну ячейку, их сигналы встретятся в одном из узлов, в котором формируется единый сигнал. Задержки не произойдет даже в том случае, если все 4000 процессоров обратятся к одному запоминающему устройству, например, когда все они должны работать по одной и той же подпрограмме. Вместо того чтобы добиваться выполнения

этой подпрограммы, они пытаются перейти к работе над следующей подпрограммой. Для этих целей конструкторы ультракомпьютера изобрели операции, которые они назвали ВЫЗВАТЬ и СЛЮЖИТЬ *). Когда поступают команды на вызов индекса подпрограммы и его изменение на единицу смешивается, они становятся сигналом на вызов этого индекса и замену его двумя. Когда результат выполнения программы приходит обратно, эти индексы направляются в разные процессоры, так что каждый процессор оперирует с «индивидуальным» индексом.

Если говорить о далекой перспективе, то я считаю этот ультракомпьютер наиболее подходящей системой с параллельными процессорами, которую можно применять в научных исследованиях.

Исследования в области искусственного интеллекта, воспроизведение человеческой речи и т. д. лежат несколько в другой плоскости. Здесь предпочитают говорить о применении параллельных процессоров в структурах «древесного» типа **). Эта форма организации проста, ибо она двумерна и соответствует духу проблемы. Однако то, что хорошо для проблем вроде проблемы искусственного интеллекта, не всегда полезно для решения физических задач, где опять возникает проблема «бутылочного горлышка» при попытке организовать связь между всеми процессорами.

Нет никаких сомнений в том, что в ближайшее время появится большее количество систем такого рода, и, быть может, все то, о чем я говорил, будет реализовано на практике.

*) Англ. fetch и add. (Прим. перев.)

**) Имеются в виду процессы, описываемые планарными «древесными» графами. (Прим. перев.)

