

## ОСОБЕННОСТИ ОПТИМИЗАЦИИ РАСПРЕДЕЛЕНИЯ ВЫЧИСЛИТЕЛЬНОЙ НАГРУЗКИ В ЗАДАЧАХ ПАРАЛЛЕЛЬНОЙ ОБРАБОТКИ ИНФОРМАЦИИ И ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ

А. В. Бухановский, С. В. Иванов (Санкт-Петербург)

### 1. Введение

Повышение требований к производительности современных информационных и интеллектуальных систем, применяемых в морских исследованиях, приводит к необходимости использования многопроцессорных вычислительных комплексов [1, 2]. Для обработки и усвоения информации в таких системах необходимо применять специализированные параллельные алгоритмы и соответствующее программное обеспечение. Это часто требует разработки новых подходов к хорошо известным методам обработки и синтеза данных, возможный параллелизм которых не всегда очевиден [3]. Однако, даже для алгоритмов, теоретически допускающих полное распараллеливание, практическое ускорение будет далеко не всегда удовлетворительным по причине дисбаланса нагрузки вычислительных узлов вследствие системных и (или) архитектурных особенностей вычислительного комплекса.

Менее всего проблемы дисбаланса нагрузки сказываются для SMP и NUMA систем с достаточно небольшим количеством процессоров, тем не менее – обладающих высокой рыночной стоимостью. В то же время, для кластеров и MPP систем проблема оптимального распределения нагрузки более критична, особенно в системах, узлы которых объединены более, чем двумя маршрутизаторами [4].

Следует отметить, что современные разработчики параллельного программного обеспечения уделяют проблемам балансировки нагрузки достаточно большое значение. Большинство современных промышленных суперкомпьютеров имеют встроенные системы балансировки нагрузки, такие как LL – Loadleveler или CSM – Cluster System Management (IBM), однако они являются высокоуровневыми решениями и работают на уровне приложений. В этом же ключе функционирует и универсальная система динамического распределения нагрузки DYNAMITE [5].

Следует отметить, что сама задача балансировки нагрузки в сложных системах также лежит в сфере приложений искусственного интеллекта [6], и представляет самостоятельный интерес. Однако, учитывая, что для создания сложных информационных и интеллектуальных систем, оперирующих с большими объемами информации (в частности, характерных для морских исследований [2]), необходимо учитывать специфические особенности алгоритмов и обрабатываемых данных, разработка методов оптимизации нагрузки должна быть тесно связана с предметной областью исследований. Данная работа посвящена основным причинам, из-за которых может возникнуть дисбаланс нагрузки при статистических вычислениях, а также – практическим технологиям их устранения.

### 2. Проблемы распределения вычислительной нагрузки в задачах обработки информации и имитационного моделирования

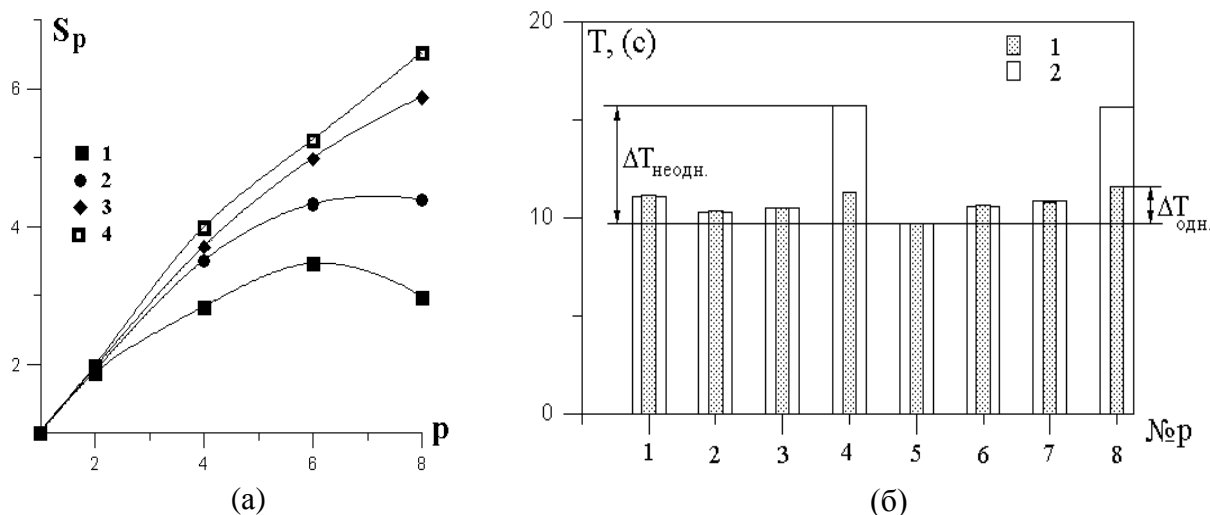
Для того, чтобы формализовать задачу распределения нагрузки в многопроцессорной системе, рассмотрим теоретическую модель BSP-программы [7]. Даже в том случае, если программа теоретически допускает полное распараллеливание на  $p$  процессоров, это еще не гарантирует эффективной работы алгоритма на конкретной вычислительной платформе, как за счет кэширования данных, так (и это главное) – с размещением их на локальных узлах. С этим и связаны основные причины возникновения дисбаланса нагрузки.

**Алгоритмический дисбаланс нагрузки.** Несомненно, что нарушение теоретической балансировки нагрузки, вызванное обычно особенностями применяемых алгоритмов, например, содержащих в себе вычисляемое логическое условие, часто оказывается основной причиной дисбаланса. В частности, это особенно проявляется для различного рода статистических итерационных алгоритмов (например, реализующих задачу параметрического оценивания методом максимального правдоподобия в виде системы нелинейных уравнений, или вычисление естественного базиса итерационными методами решения проблемы собственных значений). Однако для большинства статистических вычислений алгоритмы сводятся к линейным типовым операциям с однородными массивами данных, которые можно с достаточно высокой точностью разделить на равные фрагменты, т. е. воспользоваться принципом декомпозиции по ансамблю [8], т. е. условие теоретической балансировки автоматически выполняется.

**Коммуникационный дисбаланс нагрузки.** При параллельных статистических вычислениях необходимо передавать большие объемы данных (например, элементов выборки) с узла ввода-вывода на другие вычислительные узлы. При этом эффективность такого алгоритма зависит от величины  $T_{comp}/T_{comm}$  (отношения времени параллельных вычислений к времени, потребному для передачи данных на узлы). В качестве примера на рис. 1 приведены оценки производительности параллельного статистического алгоритма моделирования ансамбля многомерной гауссовой случайной величины методом Монте-Карло по большим выборкам различного объема. На рис. 1(а) изображен график зависимости ускорения  $S_p$  от числа процессоров, а на рис. 1(б) – время полной загрузки процессоров в ходе вычислений  $T=T_{comp}+T_{comm}$ . Все расчеты выполнены на восьмипроцессорном кластере «Паритет» [9] Института высокопроизводительных вычислений и информационных систем.

Как видно из рис. 1(а), лишь для достаточно больших объемов данных достигается значительная эффективность (например,  $S_p=6.5$  для объема выборки  $N=20000$ ); при этом при увеличении числа  $p$  происходит резкое падение производительности. Это может быть обусловлено как слишком высокими затратами на передачу данных  $T_{comm}$  (закон Амдаля [10]), так и потерей ускорения за счет дисбаланса загрузки узлов. В частности, из рис. 1(б) видно, что даже для однородной системы на 8 процессорах дисбаланс может составлять более 10%.

В ряде случаев это приводит к ситуации, когда использование многопроцессорной системы дает обратный эффект, и параллельная программа начинает работать медленнее своего последовательного аналога. Эта ситуация с точки зрения дисбаланса весьма характерна – время последовательных операций на подготовку и рассылку данных значительно превышает не только время их параллельной обработки, но и время работы последовательного алгоритма.



**Рис. 1. Оценки производительности параллельного алгоритма статистического моделирования МСВ (зерностью  $m=20$ )**

(а): Ускорение параллельных вычислений (однородная система):

Объем выборки: 1- $N=2000$ , 2- $N=4000$ , 3- $N=10000$ , 4- $N=20000$ .

(б): Время полной загрузки процессоров ( $N=20000$ ):

1 – однородная система, 2 – неоднородная система.

**Аппаратный дисбаланс нагрузки.** В том случае, если вычислительный комплекс объединяет ресурсы, неоднородные как с аппаратной, так и с программной точки зрения, имеет место естественный дисбаланс нагрузки. При этом на физически однородных многопользовательских вычислительных системах такая ситуация порождается при одновременной работе, например, двух параллельных приложений, использующих один вычислительный узел. Их результирующее ускорение может значительно упасть. В качестве примера на рис. 1(б) приведен график загрузки процессоров, на кластере, где на одном из вычислительных узлов (процессоры № 4 и № 8) запущена тестовая задача с равным приоритетом. В этом случае дисбаланс составляет уже около 50%, и ускорение вычислений резко падает ( $S_p=4.3$ ). Данный эффект будет сказываться тем сильнее, чем больше процессоров задействовано при решении задачи. Важно отметить, что проблема множественного доступа носит во многом спорадический характер, что, очевидно, усложняет методы ее решения.

Очевидно, что для ряда алгоритмов, ориентированных на специализированные вычислительные системы с небольшим количеством вычислительных узлов, можно *a priori* построить статический план оптимальной нагрузки вычислительных узлов. Однако в общем случае единственным способом решения подобных проблем может стать только динамическая балансировка нагрузки.

### 3. Технологии динамической балансировки нагрузки в задачах обработки данных и имитационного моделирования

Рассмотрим некоторые технологии динамической балансировки нагрузки, которые позволяют оптимизировать ускорение для ряда типовых алгоритмов обработки и моделирования информации, в частности, статистического оценивания случайных величин и функций, а также – стохастического воспроизведения ансамблей реализаций случайных величин, временных рядов и случайных полей методом Монте-Карло [11]. Для неоднородных вычислительных сред, или в случае частичной или полной загрузки одного или нескольких вычислительных узлов однородной вычислительной среды, од-

ним из наиболее разумных видится алгоритм разделения исходных данных или участков моделирования на блоки, внутри которых динамически вводится дробная неравномерность, соответствующая частоте  $v_k = L_k^{-1}$  моделирования или обработки данных вычислительным узлом. В случае изменения загрузочной конъюнктуры, очередной блок должен динамически скорректировать перераспределение данных между узлами, что должно привести к оптимизации загрузки всех вычислительных узлов суперкомпьютера и, как следствие, к уменьшению суммарного времени вычислений.

В простейшем случае, для декомпозиции по ансамблю, каждому блоку ставится в соответствие оптимальный план загрузки процессоров, которые характеризуются набором коэффициентов  $q_k$ ,  $\sum_{k=1}^p q_k = 1$ , определяющих долю выборочных данных, обрабатываемых на каждом процессоре. На каждом шаге  $m$  вычислительной процедуры производится уточнение коэффициентов  $q_k^* = \frac{v'_k}{v'_1 + \dots + v'_p}$ , где  $v'_k = (L_k + T_{comm}^{(k)})^{-1}$  – полное время загрузки каждого процессора на предыдущем шаге (включая время передачи данных на узел). Следует отметить, что такая технология является достаточно эффективной. Например, для однородной вычислительной системы с большим количеством процессоров, свободной от других приложений, теоретически она уже при  $m=2$  позволяет достичь существенного увеличения производительности за счёт уменьшения коммуникационного дисбаланса. Для систем, загрузка процессоров которых меняется во времени, устойчивых оценок коэффициентов  $q_k^*$  получить, по-видимому, не удастся.

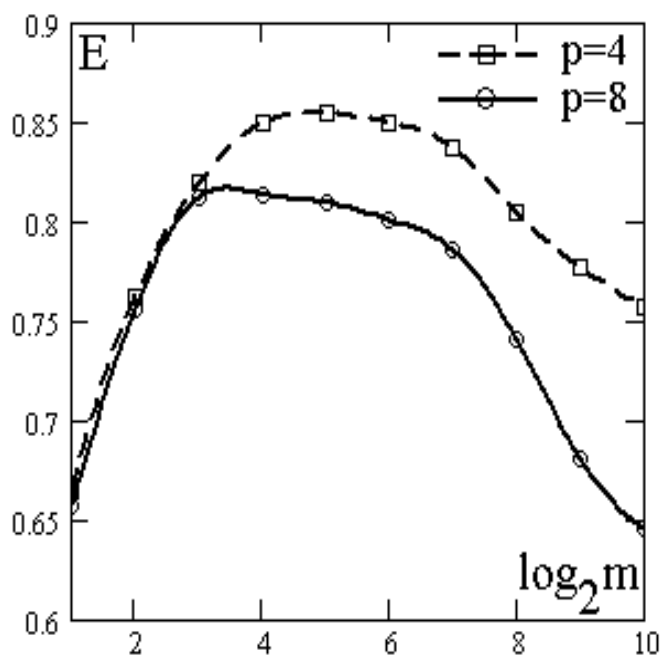


Рис. 2. Зависимость эффективности  $E$  параллельного алгоритма от числа разбиений  $m$  в процедуре динамической балансировки нагрузки

Следует отметить, что производительность алгоритма балансировки существенно зависит от числа разбиений  $m$ . В частности, на рис. 2 приведены зависимости эффективности параллельного алгоритма  $E = S_p/p$  от  $m$  для задачи оценивания многомерной случайной величины (см. рис. 1), для вычислений, проводимых на 4 и 8 процессо-

рах кластера «Паритет». Из рис. видно, что уже при  $m=8\div 16$  эффективность (а, следовательно, ускорение) становится максимальным (причем процедура балансировки позволила увеличить ускорение более, чем на 24% для  $p=8$ , и около 30% – для  $p=4$  при исходном дисбалансе порядка 45%, что является результатом, близким к оптимальному). При этом видно, что при значительном увеличении  $m$  происходит падение производительности.

Это, по-видимому, связано с уменьшением зернистости отдельных вычислительных блоков за счет накладных расходов на организацию пересылок данных.

Следует отметить, что задача динамической балансировки нагрузки с целью оптимизации ускорения параллельного алгоритма  $S_p$  в общем случае должна включать в себя не только распределение нагрузки между всеми имеющимися процессорами, но и выбор их оптимального числа (в соответствии с особенностями задачи, см. рис. 1(а)). Для этого может быть использован алгоритм, основанный на оценках времени выполнения задачи на всем объеме данных с привлечением информации о теоретической сложности алгоритма, после чего часть процессоров должна выполнять либо минимальную работу, либо игнорироваться вовсе.

Несомненно, окончательное решение этих проблем может быть достигнуто лишь путем объединения алгоритмов в единую интеллектуальную технологию, позволяющую системе гибко определять условия своего функционирования. Это направление требует дальнейших исследований и использования более изощренных методов балансировки, например [12].

#### 4. Выводы

Таким образом, использование интеллектуальных методов динамического распределения нагрузки в задачах обработки и моделирования информации позволяет:

- I. обеспечивать эффективное функционирование информационных и интеллектуальных систем на базе многопроцессорной техники, в т. ч. – в реальном времени;
- II. упростить перенос программного обеспечения на многопроцессорные системы различной архитектуры, и повысить степень его надежности;
- III. снизить затраты на разработку, отладку и оптимизацию вычислительного программного обеспечения для многопроцессорных систем.

Тем не менее, рассмотренный в данной работе подход ориентирован на наиболее простые принципы создания параллельных алгоритмов, основанных на декомпозиции по ансамблю или по индексирующей переменной (т. е., имеющих линейную сложность). Для более сложных методов, например, использующих принцип перемешивания [3], этот вопрос требует дальнейших исследований.

#### Литература

1. **Нечаев Ю.И.** Искусственный интеллект: концепции и приложения. СПб., изд. Центр СПбГМТУ, 2000.
2. Интеллектуальные системы в морских исследованиях и технологиях /ред. **Ю.И. Нечаев.** СПб, Изд. СПбГМТУ, 2001
3. **Бухановский А.В., Иванов С.В.** Параллельная обработка данных в информационно-управляющих системах. Труды Конференции УИТ'03, С-Петербург, 2-4 апреля 2003, с. 64–68.
4. **Богданов А.В., Бухановский А.В.** Интегрированный комплекс разработки приложений для кластерных систем семейства «СКИФ». Труды X Всероссийской конференции ТЕЛЕМАТИКА'2003, СПб, 14-17 апреля, 2003, с. 294–295.
5. Dynamite-Technical Description  
([www.hoise.com/dynamite/dynproduct/description.html](http://www.hoise.com/dynamite/dynproduct/description.html))

6. **Бобровский С.** Перспективы и тенденции развития искусственного интеллекта. PC Week/RE №32, 2001 г., с. 32.
7. **Gerbessiotis A.V.** Architecture independent parallel algorithm design: theory vs practice. Future Generation Computer Systems, 18, 2002, pp. 573–593.
8. **Boukhanovsky A.V. Ivanov S.V.** Stochastic simulation of inhomogeneous metocean fields. Part III: High-performance parallel algorithms. Proc. of ICCS'03, Lecture Notes in Computer Sciences, 2657, Springer, 2003 (in press).
9. Институт высокопроизводительных вычислений и информационных систем. 2003 (на [www.csa.ru](http://www.csa.ru))
10. **Ортега Дж.** Введение в параллельные и векторные методы решения линейных систем. М. Мир, 1991
11. **Бухановский А.В.** Высокопроизводительные технологии стохастического моделирования внешних воздействий в морских интеллектуальных системах. Труды Конференции МОРИНТЕХ'2003 (этот том).
12. **Rotaru T., Nageli H.H.** Heterogeneous dynamic load balancing with a scheme based on the Laplasian polynomial. Lecture Notes in Computer Sciences, 2328, 2001, pp. 107–114.