



9. The Tntnet project C++ dynamite for the web / <http://www.tntnet.org/> - Загл. с экрана.

10. High-speed C++ MVC Framework for Web Application <http://www.treefrogframework.org/> - Загл. с экрана.

1. RFC 3875, «The Common Gateway Interface (CGI) Version 1.1» [Текст], October 2004

11. RFC 2616, «Hypertext Transfer Protocol -- HTTP/1.1» [Текст], June 1999

Л.В. Логанова

ВЫБОР МАТЕМАТИЧЕСКОЙ МОДЕЛИ ПАРАЛЛЕЛЬНОГО АЛГОРИТМА С УЧЕТОМ ОСОБЕННОСТЕЙ АРХИТЕКТУРЫ СУПЕРКОМПЬЮТЕРА СЕРГЕЙ КОРОЛЁВ

(Самарский государственный аэрокосмический университет)

Введение. При решении задач математического моделирования основной интерес исследователей, прежде всего, связан с разработкой или выбором эффективного численного метода. Однако, зачастую при выборе численного метода, в тени остаются вопросы, связанные с архитектурой аппаратных средств и программным обеспечением, используемым для его реализации. Традиционно эти вопросы в литературе рассматривались отдельно [1-4]. Исключение составляют, пожалуй [5-6].

Однако в последние годы появились работы, в которых совместно рассматриваются математические основы параллельных численных методов и параллельных вычислительных систем (ВС). Для параллельных ВС особенно актуальной становится проблема согласованности алгоритмов со структурой таких систем. Эффективная реализация алгоритма на выбранной параллельной вычислительной системе может быть не обеспечена в результате недостаточной загрузки устройств, а также недостаточного учета особенностей коммуникационной сети, соединяющей их. Вместе с тем, еще 70-е годы прошлого столетия академиком Г.И. Марчуком было сформировано соответствующее направление исследований, получившее название «отображение проблем вычислительной математики на архитектуру вычислительных систем». Данное направление включает в себя: структурный анализ алгоритмов, изучение математических особенностей архитектур вычислительных систем, автоматизацию процессов распараллеливания программ, разработку и исследование параллельных численных методов [7].

В настоящее время с активным внедрением в повседневную практику GP GPU интерес к данной тематике многократно возрос. При этом для совместного исследования параллельных алгоритмов и ВС необходимо, чтобы средства их описания были унифицированы. Для построения математической модели алгоритма, реализуемого на суперкомпьютере «Сергей Королев», приведем несколько нотаций представления математических моделей алгоритмов и ВС.



1 Концепция Воеводина В. В. и Воеводина Вл. В.

В работах [8-9] авторы в качестве средств описания ВС и алгоритмов предлагают использование графов. Сравнение свойств систем и алгоритмов на уровне графов приводит к операциям сравнения графов, например, на изоморфизм, гомоморфизм.

Для построения графа алгоритма предлагается множеству операций поставить во взаимно однозначное соответствие множество точек. Соединить дугой точки (вершины), соответствующие операциям, для которых аргумент одной из них является результатом выполнения другой. Выполнив разметку графа особым образом, построить строгую параллельную форму [8-9]. Из возможных параллельных форм выбрать максимальную.

ВС в данной концепции представляется совокупностью функциональных устройств (ФУ), обмен информацией между которыми осуществляется коммуникационной сетью [8-9]. Математическое описание ВС может быть также представлено графом. При этом в качестве базовой модели может быть взят граф алгоритма, в котором каждая вершина отождествляется с ФУ, а дуги определяют связи между ними. Применяя операции простого гомоморфизма к графу базовой модели, получим модель ВС. Данный прием позволяет увеличить загруженность функциональных устройств, обеспечивая этим рост их производительности. При этом возрастает и ускорение. Определение загруженности ВС как взвешенной суммы загруженностей отдельных устройств:

$$p = \sum_{i=1}^s \alpha_i p_i, \quad \alpha_i = \pi_i / (\sum_{j=1}^s \pi_j)$$
 позволяет сделать вывод о том, за счет какого устройства ее можно повысить. Максимально возможное ускорение параллельного алгоритма (N – количество операций, m – высота параллельной формы) равно N/m достигается при использовании устройств минимальным количеством равным ширине алгоритма (q).

Данный подход является наиболее подходящим при создании специализированных ВС, ориентированных на заданный алгоритм. При рассмотрении существующих ВС, даже если система срабатывает и реализует алгоритм, его реализация может характеризоваться слишком большим временем, низкой загруженностью ФУ и т.д. К тому же нет необходимости в рассмотрении готовых современных ВС на уровне ФУ, т.к. последние строятся в соответствии со стандартами, определяемыми их современным развитием. Например, использование в современных кластерных вычислительных системах типичных топологий, организацию узлов в виде SMP и т.д. Так из описания, приведенного на сайте СГАУ, суперкомпьютер «Сергей Королев» имеет многоядерные узлы, а также узлы, оснащенные несколькими GPU. Поэтому обратимся к другой нотации.

2 Концепция Гергеля В.П.

Рассмотрим математическую модель вычислительного кластера, описанную в работе [10]. Произвольную кластерную ВС также предлагается описывать с помощью ориентированного графа, учитывая статические (например,



$c(p_i)$ – количество вычислительных ядер, $m(p_i)$ – объемы оперативной памяти, $d(p_i)$ – объемы дисковой памяти в килобайтах, $s(p_i)$ – относительную производительность) и динамические (например, $u_i(t)$ – загруженность его вычислительных ядер, $m_i(t)$ – объем доступной оперативной и $d_i(t)$ – дисковой памяти в момент времени $t \in [0; +\infty)$) характеристики, пропускную способность каждой сетевой связи в байтах в секунду (b). Приводится оценка времени передачи сообщения от одного сетевого устройства к другому.

Модель алгоритма предлагается представлять в виде графа $G = (V, R)$, где $V = \{1, \dots, |V|\}$, V – множество вершин графа, представляющих выполняемые операции, R – множество дуг графа, отражающих информационные зависимости между операциями и отмеченные весами, показывающими интенсивность информационного обмена [11]. При построении параллельного алгоритма предлагается учитывать следующие принципы: граф должен быть минимального диаметра (d – длина максимального пути графа); количество процессоров определяется величиной $p \sim T_1/T_\infty$, при этом $\forall p \Rightarrow T_p < T_\infty + T_1/p$, $p \geq T_1/T_\infty \Rightarrow T_p \leq 2T_\infty$ (T_1 – время вычислений на 1 устройстве, T_p – время реализации параллельного алгоритма, а $T_\infty = \min_{p \geq 1} T_p$). Вывод о масштабируемости параллельного алгоритма следует из формулы эффективности: $E_p = T_1 / (T_1 + T_0) = 1 / (1 + T_0/T_1)$, T_0 – время накладных расходов.

Таким образом, представленная математическая модель ВС наиболее полно соответствует состоянию современных архитектур ВС. Однако, ориентируясь на конкретную модель кластера, при оценивании коммуникационной трудоемкости целесообразно предварительно воспользоваться моделью, предложенной Хокни [12] ввиду ее простоты. При описании модели алгоритма развивается подход, рассмотренный в предыдущей концепции. Ориентируясь на суперкомпьютер «Сергей Королев», следует рассмотреть модели с учетом особенностей GPU.

Концепция Хорошевского В.Г.

В [13] математическая модель ВС ($S = \langle C, G, A(P(D)) \rangle$), рассматривается как совокупность множества вычислителей (C) с типичной структурой (G : линия, кольцо и т.д.) и алгоритмом функционирования ($A(P(D))$) при реализации параллельной программы P обработки данных (D). При этом $D = \bigcup_{i=0}^{N-1} D_i$, в

общем случае $\bigcap_{i=0}^{N-1} D_i \neq \emptyset$, $P = \bigcup_{i=0}^{N-1} P_i$, $\bigcap_{i=0}^{N-1} P_i = \emptyset$ (P_i – i ветвь программы). В качестве рекомендации к распараллеливанию сложных задач, предлагается применение крупноблочного распараллеливания, когда каждому вычислителю (C_i) предписывается выполнение i ветви программы (P_i). В качестве конструктивного приема крупноблочного распараллеливания задач предлагается распараллеливание по циклам.



Т.о. в данной нотации предлагается рассматривать ВС как совокупность устройств, функционирующих в соответствии с заданным алгоритмом в отличие [8-11]. Модель ВС учитывает структуру коллектива вычислителей в отличие от [8-9], но не содержит определений характеристик ее функционирования в отличие от [10]. Модель параллельного алгоритма предполагается в виде графа, однако, не содержит оценки времени его выполнения в отличие от [8-9, 11].

Заключение Таким образом, для построения и исследования математической модели параллельного алгоритма, с минимальным временем реализации на суперкомпьютере «Сергей Королев» предполагается выполнение следующих шагов: построение графа алгоритма согласно [8-9, 11]; его оптимизация с помощью операций простого гомоморфизма. В качестве модели ВС воспользуемся представлением из работы [10]. Оценивание загруженности устройств будем выполнять с учетом обеспечения максимального ускорения. Для оценки коммуникационной трудоемкости, воспользуемся моделью Хокни. Анализ масштабируемости алгоритма будем проводить в соответствии с [11]. Для оценки ускорения алгоритма воспользуемся законом Амдала.

Литература

1. Самарский А.А. Введение в численные методы. – М.: Наука, 1987. – 269 с.
2. Мэтьюз, Джон, Г., Финк, Куртис, Д. Численные методы. Использование MATLAB, 3-е издание. : Пер. с англ. — М. : Издательский дом "Вильямс", 2001. — 720 с.
3. Максимов Н.В., Партыка Т.Л., Попов И.И. Архитектура ЭВМ и вычислительных систем; Учебник. – М.: ФОРУМ: ИНФРА-М, 2006
4. Каган Б.М. Электронные вычислительные машины и системы Издательство: М.: Энергоатомиздат Год издания: 1991, 592 с.
5. Голуб Дж., Ван Лоун Ч. Матричные вычисления М.: Мир, 1999, 548с.
6. Дж. Ортега Введение в параллельные и векторные методы решения линейных систем. Пер. с англ. –М: Мир, 1991. – 367 с.,ил.
7. Марчук Г. И., Котов В. Е. Проблемы вычислительной техники и фундаментальные исследования.- Автомат, и вычисл. техн., 1979, № 2,с. 3-14
8. Воеводин В. В. Математические модели и методы в параллельных процессах.—М.: Наука. Гл. ред. физ.-мат. лит., 1986.— 296 с.
9. Воеводин В.В., Воеводин Вл.В. Параллельные вычисления. – СПб.: БХВ-Петербург, 2002.- 608 с.:ил.
10. Гергель В.П., Полежаев П.Н. //Теоретические основы экспериментального исследования алгоритмов планирования задач для вычислительного кластера с помощью симулятора. Вестник ОГУ №9 (115)/сентябрь`2010
11. Гергель В.П. Теория и практика параллельных вычислений: учебное пособие /В.П. Гергель. — М.: Интернет-Университет Информационных Технологий; БИНОМ. Лаборатория знаний, 2007. – 423 с.: ил., табл. – (Основы информационных технологий).



12. Hockney R. The communication challenge for MPP: Intel Paragon and Meiko CS-2 // Parallel Computing. 1994. 20 (3). P. 389 – 398

13. Хорошевский В.Г. Архитектура вычислительных систем: Учеб. Пособие. – 2-е издание, перераб. И доп. – М.: Изд-во МГТУ им. Баумана, 2008, 520с.

В.В. Сазонов, М.А. Щербаков

ВОССТАНОВЛЕНИЕ ИЗОБРАЖЕНИЙ С ПОМОЩЬЮ НЕЛИНЕЙНОГО SVD-ФИЛЬТРА

(Пензенский государственный университет)

Введение

В общем случае решение задачи восстановления стационарного случайного сигнала или процесса на фоне случайных шумов и помех предполагает выбор одной из двух (как минимум) гипотез: определена или нет априорная модель исходного сигнала. Это принципиальный момент, во многом определяющий дальнейшее решение поставленной задачи.

В первом случае наблюдаемый сигнал целенаправленно подгоняется под выбранную модель, при этом помеха рассматривается как пассивный параметр оценки несоответствия между моделью и сигналом.

Принято считать, что наилучшее восстановление для класса стационарных гауссовых сигналов достигается линейной системой, а ее оптимальной реализацией является фильтр Винера-Колмогорова [1], который может быть представлен в следующем виде:

$$\hat{x}_{i,j} = \frac{\sigma_x^2}{\sigma_x^2 + \sigma_\eta^2} y_{i,j}, \quad (1)$$

где σ_x^2 - дисперсия полезного сигнала, σ_η^2 - дисперсия шума, $\hat{x}_{i,j}$ - восстановленный элемент исходного изображения, $y_{i,j}$ - элемент наблюдаемого изображения.

Соответственно, эффективность работы фильтра Винера в данных условиях во многом определяется точностью количественной оценки шумовой компоненты. Однако данный подход обладает определенными недостатками, в том числе:

- оптимальность достигается интегрально по всей совокупности анализируемого процесса;
- возможная неустойчивость алгоритмов фильтрации [1].

Эти недостатки хорошо известны и привели к появлению целого ряда достаточно успешных модификаций фильтра Винера, например, или в виде апертурных (масочных) алгоритмов, которые работают в окрестности локальных точек [1], или в виде робастных (устойчивых) алгоритмов [2].