

ОТОБРАЖЕНИЕ ПАРАЛЛЕЛЬНЫХ БЛОЧНЫХ МЕТОДОВ РЕШЕНИЯ ЗАДАЧИ КОШИ НА MIMD СТРУКТУРЫ С ТОПОЛОГИЕЙ КВАЗИМАТРИЦА

Фельдман Л.П., Дмитриева О.А.

Кафедра ПМИ ДонНТУ

feldman@r5.dgtu.donetsk.ua, dmitriv@r5.dgtu.donetsk.ua

Abstract

Feldman L., Dmitrieva O. Reflection of parallel block methods for decision of Cauchy task on MIMD structures with quasimatrix topology. The realization methods of block algorithms on multiprocessors calculable type systems MBC-100/1000 are worked up. Herewith a calculations model, that are founded on steps sequence, each of which breaks down under well-ordered phases: local calculations, communications, barrier synchronization, is used. Two parallelisation method, that are attached to realization multisteps multipoints of block methods, «fine-grained» and «coarse-grained», are offered. Estimations of parallelism of offered methods: acceleration and effectiveness for diverse types of solved systems of usual differential equalizations, are got.

Введение

Все чаще альтернативой традиционным суперкомпьютерным системам становятся вычислительные системы, создаваемые из массово выпускаемых компонентов. При решении многих прикладных задач такие вычислительные системы с небольшим или средним числом вычислительных модулей (до 128-256) показывают производительность, не уступающую или даже превосходящую производительность традиционных суперкомпьютеров как с распределенной, так и с разделяемой памятью [1,2]. Такие вычислительные системы обладают рядом преимуществ, среди которых следует указать более низкую стоимость, короткий цикл разработки и возможность оперативно использовать наиболее эффективные вычислительные и коммуникационные компоненты из имеющихся на рынке во время создания системы.

Наиболее типичными представителями систем, создаваемых из массово выпускаемых компонентов, являются многопроцессорные вычислительные системы MBC-100/1000. В этих системах массовое распараллеливание вычислений достигается за счет использования быстродействующей микропроцессорной базы. Массово параллельные масштабируемые системы MBC предназначены для решения прикладных задач, требующих большого объема вычислений и обработки данных. Использование таких систем позволяет решать сложные вычислительные задачи качественно нового уровня для различных областей применений, причем подтверждена возможность эффективного распараллеливания компьютерной обработки и целесообразность дальнейшего развертывания работ этого направления [2].

Для построения самой вычислительной системы используется топология квазиматрица, которая позволяет реализовать наращиваемые по производительности многопроцессорные вычислительные системы без изменения топологии связей между процессорными модулями, образующими вычислительное поле. При этом важно, что максимальная длина пути передачи информации между процессорными элементами в

системе МВС не превышает длину пути в системах топологии гиперкуб, что свидетельствует о ее оптимальности. Преимуществом системы с архитектурой квазиматрица по сравнению с гиперкубом является ее наращиваемость без изменения конструкции самого модуля (вычислительные системы в топологии гиперкуб, в отличие от МВС, не являются масштабируемыми).

Еще одним преимуществом МВС является реализация двухуровневого функционирования, т.е. разделение производства вычислений (верхний уровень) и осуществления межпроцессорных обменов (нижний уровень). Это дает возможность без изменения архитектуры системы производить замену типов как вычислительного процессора, так и коммуникационного на более производительные (по мере их появления) с сохранением преемственности по отношению к созданному программному и прикладному математическому обеспечению.

Отображение решения СОДУ многошаговым многоточечным блочным методом на топологию квазиматрица

В [3] было показано, что «узким местом» решения задачи Коши для систем ОДУ

$$\frac{dx}{dt} = f(t, x), \quad x(t_0) = x_0 \quad (1)$$

многоточечным блочным методом

$$\frac{u_{n,i} - u_{n,0}}{i\tau} = \sum_{j=1}^k b_{i,j} F_{n-1,j} + \sum_{j=1}^k a_{i,j} F_{n,j}, \quad i = \overline{1, k}, \quad n = 1, 2, \dots, N, \quad (2)$$

где $F_{n,j} = f(t_n + j\tau, u(t_n + j\tau))$

с помощью следующих итерационных формул

$$u_{n,i,0} = u_{n,0} + i\tau \sum_{j=1}^k c_{i,j} F_{n-1,j}, \quad i = \overline{1, k}, \quad n = 1, 2, \dots, N, \\ u_{n,i,s+1} = u_{n,0} + i\tau \left(\sum_{j=1}^k b_{i,j} F_{n-1,j} + \sum_{j=1}^k a_{i,j} F_{n,j,s} \right), \quad i = \overline{1, k}, \quad s = \overline{0, k-1}, \quad n = 1, 2, \dots, N \quad (3)$$

на параллельных вычислительных структурах типа SIMD является невозможность одновременного вычисления значений правых частей, поскольку, в общем случае, правые части уравнений различны, а в SIMD структурах все процессоры одновременно могут выполнять только однотипные операции. С указанной проблемой хорошо справляются вычислительные архитектуры типа MIMD, но, как обычно бывает, новые преимущества обязательно влекут за собой новые недостатки. Для рассматриваемого случая к недостаткам отнесем необходимость учета не только вычислительных аспектов, но и декомпозицию заданий на процессы, синхронизацию их работы, возрастание времени, затрачиваемого на передачу информации между процессорами. Представим модель вычислений последовательностью шагов, каждый из которых состоит из трех упорядоченных фаз [4]: локальных вычислений, коммуникаций, барьерной синхронизации (рис. 1).

Выберем размерность блока k пропорциональной размерности вычислительного модуля: 2, 4, 8, 16 и т.д. Это требование не является обязательным, но, поскольку, речь идет о вычислительной архитектуре с топологией «квазиматрица», задачи будут хорошо масштабироваться, если размерности блока будут пропорциональны степени 2.

Если количество процессорных элементов равно p , то p/k^2 - количество уравнений системы, вычисление значений неизвестных для которых будет осуществляться параллельно. Тогда $m/p/k^2$ - количество «тактов», за которые можно рассчитать всю систему для блока из k точек. В идеале размерность процессорного массива $m*k^2$. Для вычисления начального приближения на процессорном поле размерностью $k \times k$ потребуется

$$T_{k \times k_нач_итер} = m * (2t_{умн} + 2(k-1)t_{сдв} + \log_2 k * t_{сл}) + \frac{m}{k} t_f.$$

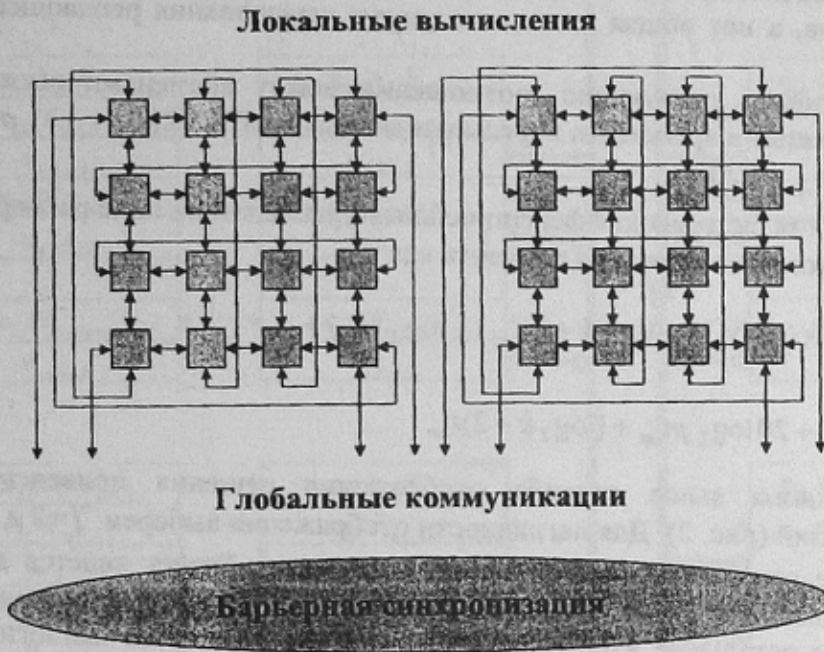


Рисунок 1 - Модель вычислений, представленная последовательностью супершагов

Общее время вычисления правых частей уменьшится в k раз, поскольку MIMD системы позволяют одновременно выполнять разнородные операции. Для полного расчета блока размерностью k точек система из m дифференциальных уравнений потребует

$$T_{k \times k} = k * [m * (2t_{умн} + 2(k-1)t_{сдв} + \log_2 k * t_{сл}) + \frac{m}{k} t_f] + m * (t_{умн} + (k-1)t_{сдв} + \log_2 k * t_{сл})$$

где добавочное время $m * (t_{умн} + (k-1)t_{сдв} + \log_2 k * t_{сл})$ обусловлено необходимостью вычисления значений $\sum_{j=1}^k b_{i,j} F_{n-1,q,j}$, $q = \overline{1, m}$, которые будут использоваться для расчета следующего блока. В общем случае, если размер процессорного поля может быть выражен зависимостью $\gamma * k * k \leq 128$, где для каждой фиксированной размерности блока γ может принимать значения из некоторого интервала (например, при $k=4$ $\gamma = \{1, 2, 4, 8\}$), т.е., если будут использоваться структурные модули

размерностью 32, 64 или 128 процессорных элементов, можно сократить общее количество арифметических операций в γ раз. С ростом размерности блока k диапазон значений γ уменьшается, и наоборот, при уменьшении размерности блока расширяется диапазон допустимых значений γ . Тогда для одной итерации

$$\frac{mk^2}{p}(2t_{умн} + (k-1)t_{сдв} + (\log_2 k + 1)t_{сл}) + \log_2 pt_{сдв} + \frac{mk}{p}t_f$$

Таким образом, использование всего множества процессорных элементов, повлечет за собой возрастание времен обменов «все-всем», т. е. увеличится время, которое будет необходимо на осуществление сдвигов. Но из зависимости видно, что это возрастание незначительно и не окажет сколь-нибудь серьезного влияния на длительность расчетов, а вот общая выгода от такого наращивания решающего поля очевидна.

Если использовать следующие соотношения между временами выполнения арифметических операций и временами пересылки данных в МВС [2]

$$t_{умн} \approx t_{сл} \approx t_{оп}, t_{сдв} \approx 20t_{оп} \quad (4)$$

то общее время решения системы дифференциальных уравнений на поле размерностью $\gamma * k * k \leq 128$ для одного блока, можно выразить как

$$T_{k \times k} \approx k \left[\frac{mk^2}{p}(20(k-1)t_{оп} + (\log_2 k + 3)t_{оп}) + \log_2 pt_{оп} + \frac{mk}{p}t_f \right] + 20 \log_2 pt_{оп} + (\log_2 k + 2)t_{оп} \quad (5)$$

Для описанного выше способа отображения решения приведем схему расположения значений (рис. 2). Для наглядности отображения выберем $\gamma=2$ и $k=4$, т.е. под решающее поле отводится 32 процессорных элемента. Расчет ведется для n -го блока. Размерность системы пропорциональна размерности блока. Расчет ведется для одной итерации, все остальные итерационные значения получаются аналогично, т.е. для расчета всех итерационных значений процесс достаточно повторить k раз. Размерность блока k может быть произвольной, но для реальных задач не рекомендуется выбирать k больше 8 из-за слишком резкого уменьшения шага интегрирования [5].

Тогда реально влияющими на длительность обработки задачи временами будут оставаться

$$T_{k \times k} \approx \frac{20mk^4}{p}t_{оп} + \frac{mk^2}{p}t_f \quad (6)$$

Если при реализации многошаговых блочных методов на SIMD структурах основное время расчетов определялось трудоемкостью последовательного вычисления правых частей системы обыкновенных дифференциальных уравнений, то теперь эту проблему удалось решить (появилась возможность проводить расчет правых частей параллельно), но тут же возникла проблема потерь времени на осуществление сдвигов.

Если каким-то образом можно будет сократить время обменов, то характеристики параллелизма: ускорение и эффективность, полученные при решении исследуемой задачи на МВС-1000 будут оптимальными.

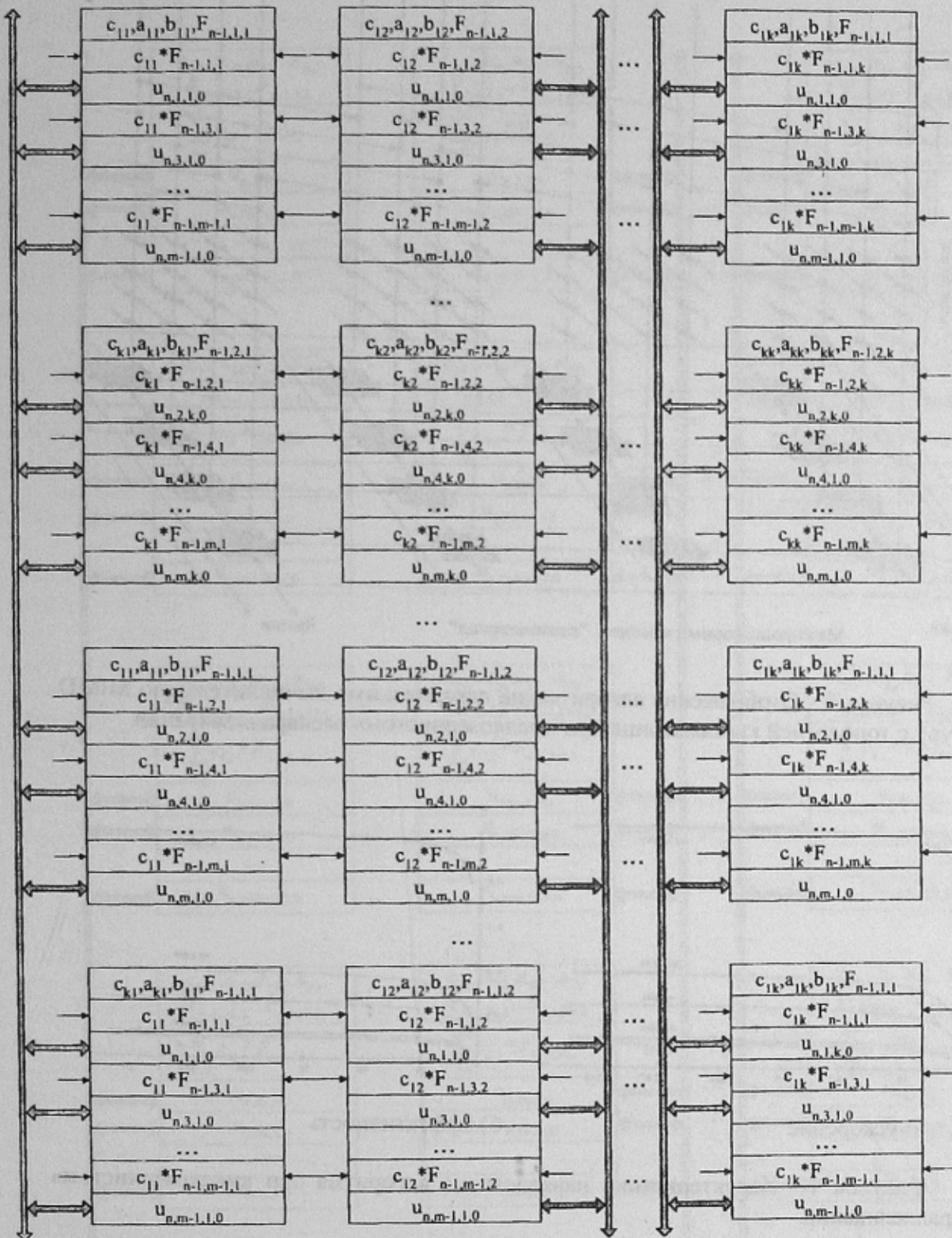


Рисунок 2 - Схема расположения значений по процессорам системы MBC-1000 при «мелкозернистом» распараллеливании

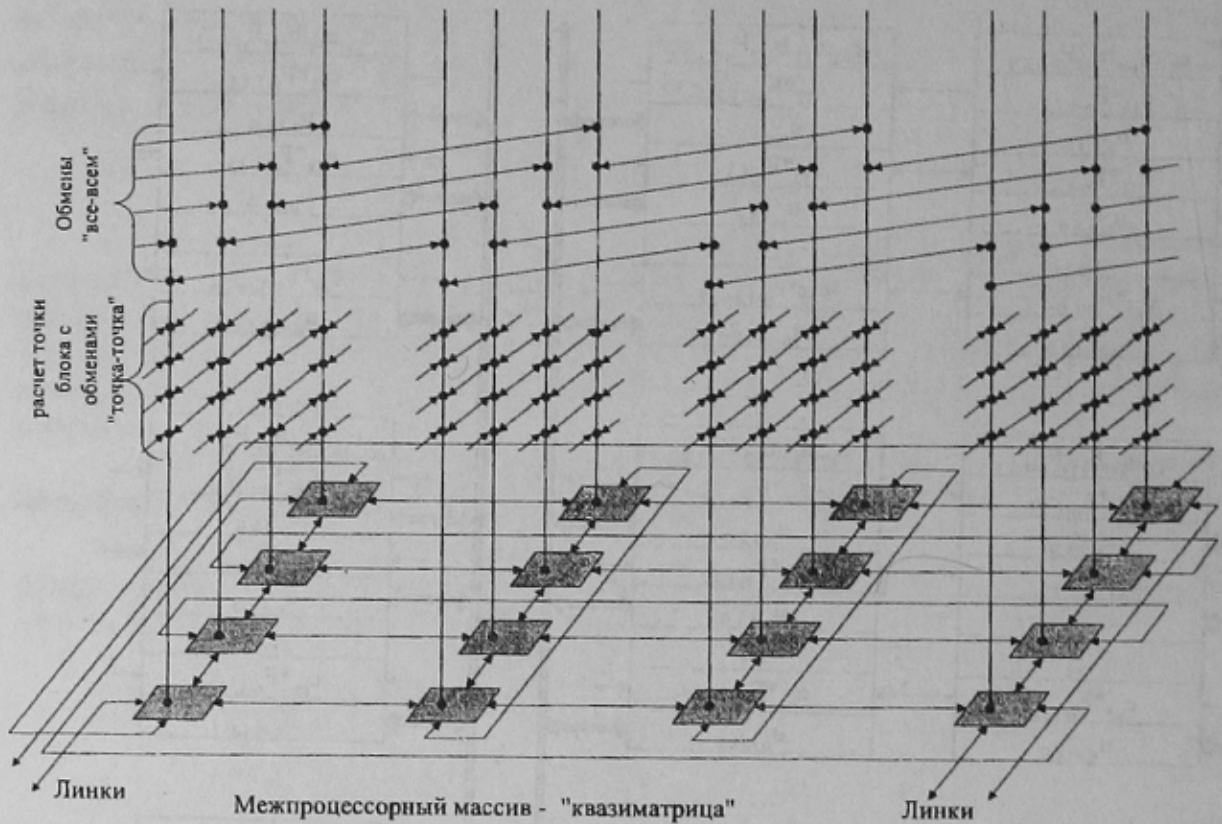


Рисунок 3 - Отображение алгоритма на параллельную вычислительную MIMD структуру с топологией квазиматрица при «мелкозернистом» распараллеливании

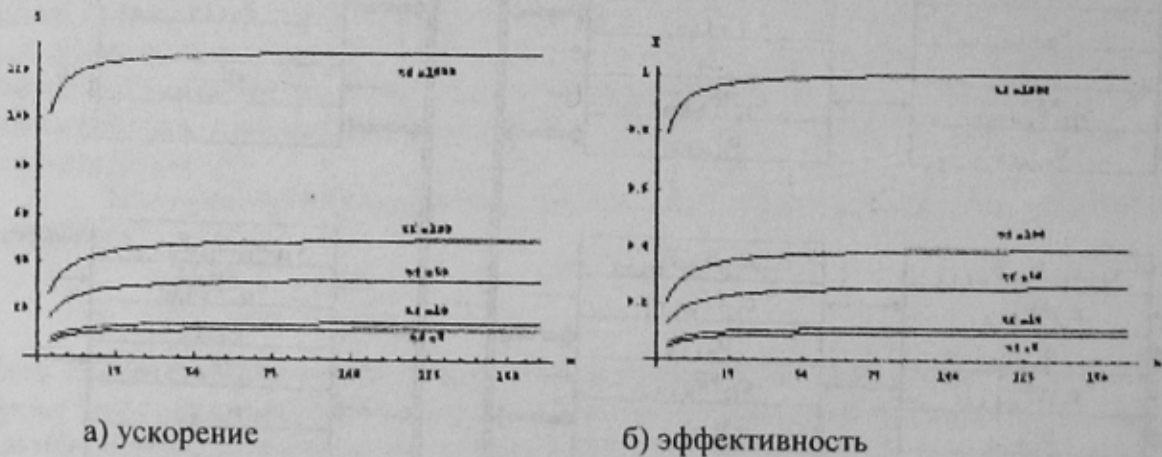


Рисунок 4 - Характеристики параллелизма алгоритма при «мелкозернистом» распараллеливании

Располагать данные необходимо таким образом, чтобы сокращалось количество обменов «точка-точка». Назначим теперь каждый процессорный элемент для расчета одной точки блока (рис.5), т.е. попытаемся «продлить» время работы каждого процессорного элемента в автономном режиме. Таким расположением

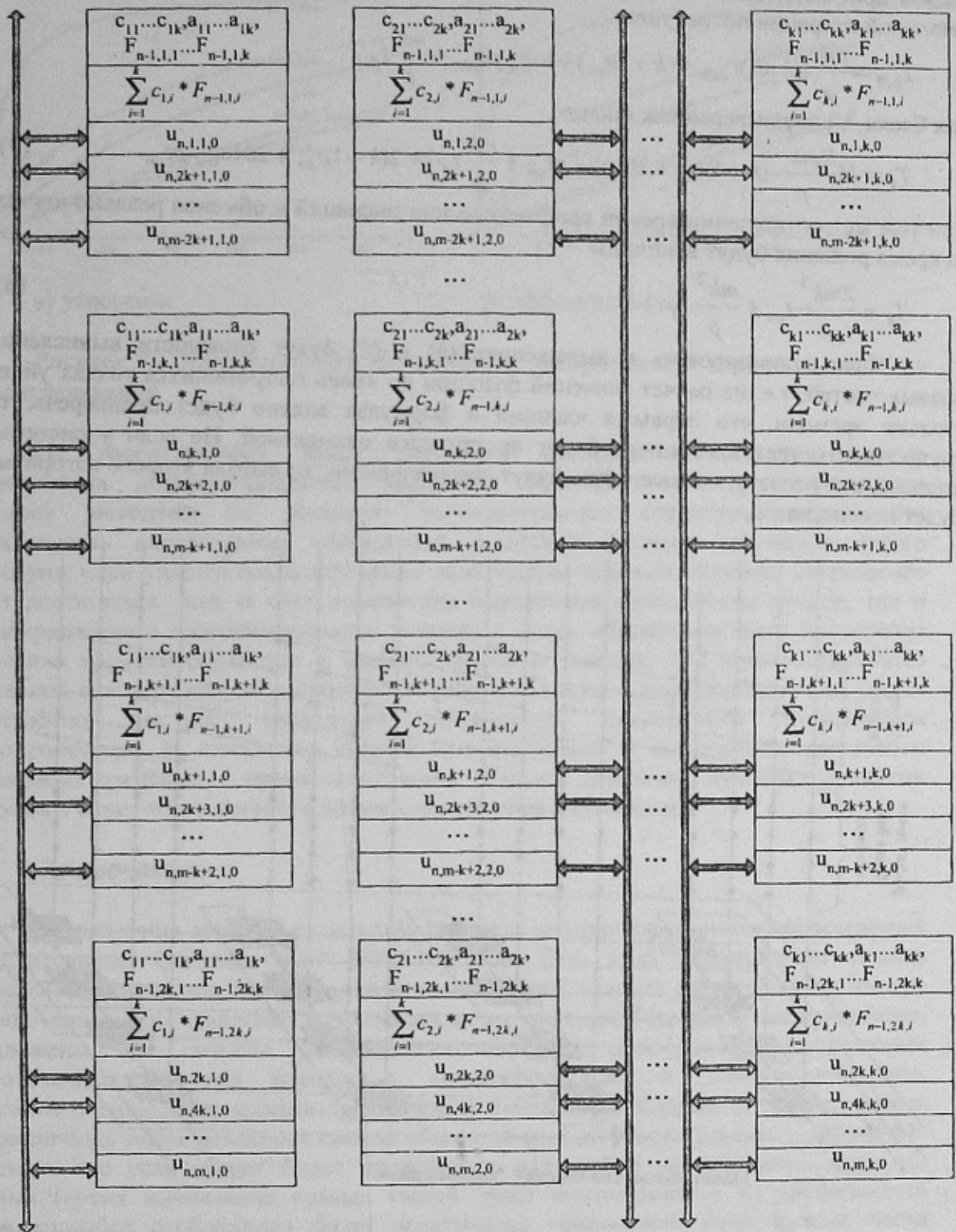


Рисунок 5 - Схема расположения значений по процессорам системы МВС-1000 при «крупнозернистом» распараллеливании

значений в процессорных элементах можно избавиться от целой группы обменов, которые в предыдущем алгоритме необходимо было осуществлять практически после каждой арифметической операции. При таком расположении данных на вычисление начальной итерации потребуется

$$T_{k,0} = \frac{mk}{p}((k+1)t_{умн} + (k+1)t_{сл}) + \log_2 pt_{сдв} + \frac{mk}{p}t_f.$$

Для блока, в силу выведенных оценок,

$$T_k \approx k\left[\frac{2mk}{p}(k+1)t_{он} + 20 \log_2 pt_{он} + \frac{mk}{p}t_f\right] + 2(k+1)t_{он} + 20 \log_2 pt_{он}. \tag{7}$$

При том же соотношении времен арифметических операций и обменов реально влиять на время решения будут величины

$$T_k \approx \frac{2mk^3}{p}t_{он} + \frac{mk^2}{p}t_f. \tag{8}$$

Если доминировать в выражениях (6) и (8) будут сложности вычисления правых частей, т.е. на расчет значений функции во вновь получившихся точках уйдет столько времени, что первыми членами в формулах можно будет пренебречь, то трудоемкость этих алгоритмов будет практически одинаковой. Но если у системы, подлежащей расчету, правые части будут тривиальными, то выгода второго алгоритма будет неоспоримой.

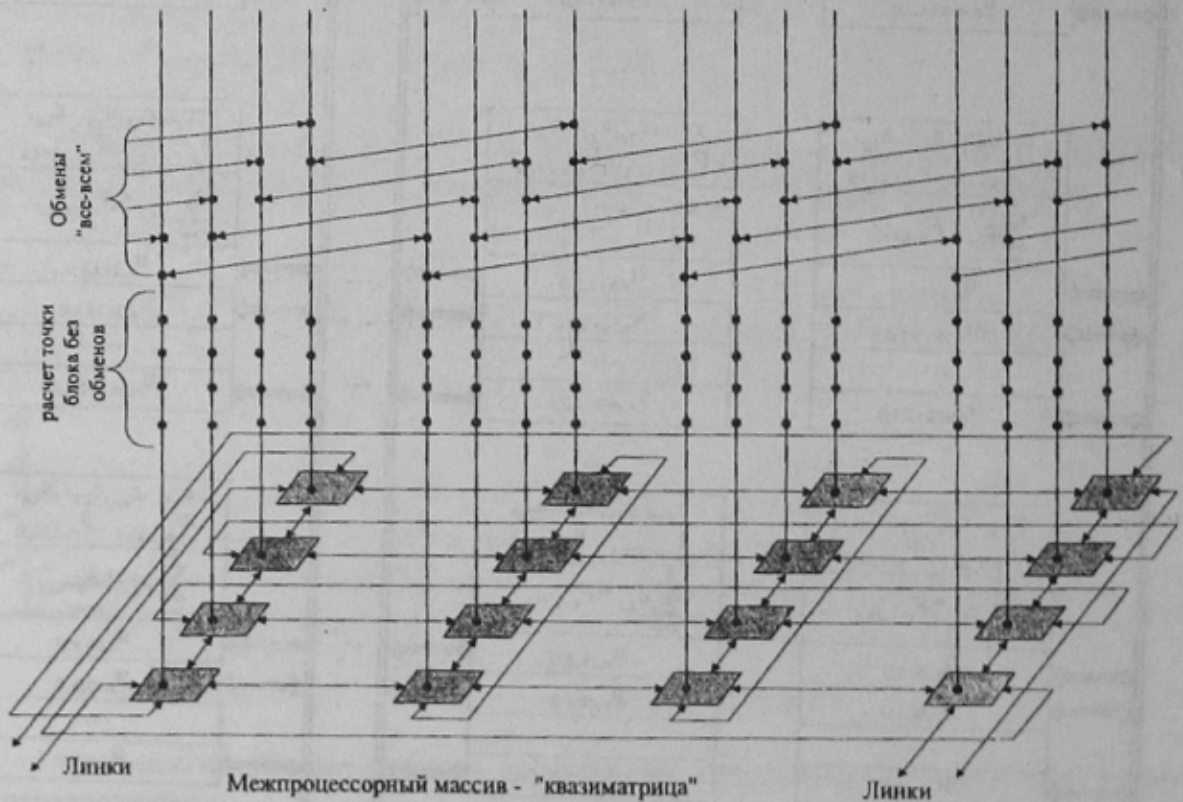


Рисунок 6 - Отображение алгоритма на параллельную вычислительную MIMD структуру с топологией квазиматрица при «крупнозернистом» распараллеливании

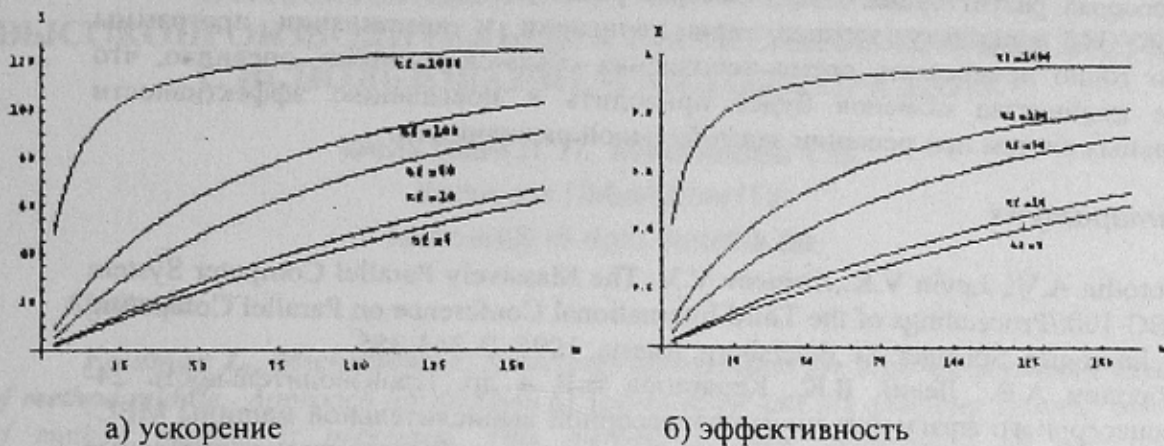


Рисунок 7 - Характеристики параллелизма алгоритма при «крупнозернистом» распараллеливании

Из рассмотренных выше вариантов отображения решения систем обыкновенных дифференциальных уравнений многошаговыми многоточечными блочными методами на реальную вычислительную структуру видно, что эффективность параллельных вычислений в MIMD системах достигает своего максимума, если удастся сократить время выполнения обменов. Причем, сокращение может достигаться как за счет повышения пропускной способности линков, так и целенаправленного программирования решаемых задач, обеспечивающего повышение отношения времени обработки к времени передачи данных. Эта задача становится актуальной еще и потому, что скорость обработки данных в микропроцессорах растет существенно быстрее увеличения пропускной способности интерфейсов микропроцессора. И, поскольку заранее при написании и компиляции программы невозможно предсказать время исполнения обменов данными, лучшим вариантом программы будет тот, у которого количество обменов минимально.

Заключение

Разработаны способы реализации блочных алгоритмов на многопроцессорных вычислительных системах типа МВС-100/1000. При этом использована модель вычислений, основанная на последовательности шагов, каждый из которых состоит из упорядоченных фаз: локальных вычислений, коммуникаций, барьерной синхронизации. Предложены два способа распараллеливания при реализации многошаговых многоточечных блочных методов – «мелкозернистый» и «крупнозернистый». Получены оценки параллелизма предлагаемых способов: ускорение и эффективность для различных типов решаемых систем обыкновенных дифференциальных уравнений. Показано, что если расчет будет проводиться для систем со сложными правыми частями (время вычисления правых частей будет доминировать), то трудоемкость обоих способов отображения будет практически одинаковой. Если правые части системы будут тривиальными, то использование второго способа предпочтительнее. За счет целенаправленного программирования (использования систолических алгоритмов умножения матриц, оптимального размещения данных по процессорам) в предложенных способах отображения значительно сокращено количество обменов. Эта

задача становится актуальной еще и потому, что скорость обработки данных в микропроцессорах растет существенно быстрее увеличения пропускной способности интерфейсов. И, поскольку заранее, при написании и компиляции программы невозможно точно предсказать время исполнения обменов данными, очевидно, что сокращение количества обменов будет приводить к повышению эффективности вычислительных систем при решении задач большой размерности.

Литература

1. Zabrodin A.V., Levin V.K., Korneev V.V. The Massively Parallel Computer System MBC-100//Proceedings of the Third International Conference on Parallel Computing Technologies, Springer. St. Petersburg, Russia, 1995. P. 341-355.
2. Забродин А.В., Левин В.К., Каратанов В.В и др. Производительность 24-процессорного фрагмента многопроцессорной вычислительной машины MBC-1000M.//Тезисы докладов конференции «Высокопроизводительные вычисления и их приложения». 30 октября - 2 ноября 2000 года, Черногловка – М.:МГУ, 2000. С.9-11.
3. Фельдман Л.П., Дмитриева О.А. Эффективные методы распараллеливания численного решения задачи Коши для обыкновенных дифференциальных уравнений.// Математическое моделирование, том 13, № 7, 2001. – С. 66-72.
4. Бергизияров П.К., Луговская Ю.П., Султанов В.Г. Особенности реализации типовых алгоритмических структур с массивным параллелизмом.//Тезисы докладов конференции «Высокопроизводительные вычисления и их приложения». 30 октября - 2 ноября 2000 года, Черногловка – М.: МГУ, 2000. С. 51-55.
5. Фельдман Л.П., Дмитриева О.А. Разработка и обоснование параллельных блочных методов решения обыкновенных дифференциальных уравнений на SIMD-структурах. //Научн. тр. Донецкого государственного технического университета. Серия: Проблемы моделирования и автоматизации проектирования динамических систем, выпуск 29. - Севастополь: «Вебер», 2001. - С. 70-79.