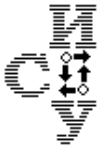
Многопроцессорные вычислительные системы
2008. №1(15)



УДК 681.5.015.63-192

© 2008 г. **Д.А. Назаров**
(Институт автоматики и процессов управления ДВО РАН, Владивосток)

# ИСПОЛЬЗОВАНИЕ РАСПРЕДЕЛЕННЫХ ВЫЧИСЛЕНИЙ ПРИ ПОСТРОЕНИИ ОБЛАСТИ РАБОТОСПОСОБНОСТИ[1]

Рассматриваются теоретические аспекты реализации параллельного алгоритма построения области работоспособности в распределенной на локальной сети вычислительной системе. Предложен алгоритм декомпозиции данных при параллельном построении области работоспособности на узлах распределенной вычислительной системы.

## Введение

Проектирование технических устройств и систем связано с необходимостью выбора номинальных значений параметров их элементов (внутренних параметров), а также допусков на эти параметры [1]. Для решения таких задач нужно определить множество значений внутренних параметров, при которых объект проектирования будет находиться в работоспособном состоянии, или, как обычно говорят, область работоспособности.

Нахождение области работоспособности относится к числу задач высокой вычислительной сложности. Это связано как с высокой размерностью пространства поиска (большим числом внутренних параметров), так и с необходимостью многократно обращаться к модели объекта проектирования, связывающей выходные параметры с параметрами элементов. Трудоемкость нахождения области работоспособности часто ограничивает возможность оптимального параметрического синтеза сравнительно простыми устройствами.

В последние годы отмечается интенсивное развитие концепции параллельных и распределенных вычислений. Такие технологии вычислений позволяют выполнять сложные и, в основном, множественные расчеты, одновременно используя различные вычислительные мощности (процессоры) [2].

Применяя параллельные расчеты преимущественно на кластерах с несколькими процессорами. При вычислениях такого типа зачастую немаловажным фактором скорости расчетов является обмен данными между вычислительными процессами. Взаимодействие между вычислительными процессами выполняется обычно на уровне системных процессов.

[1] Работа выполнена при поддержке грантов ДВО РАН 06-III-А 03-070, 06-I-П15-054 (программа ОЭММПУ РАН №15).

Распределенные вычисления ориентированы на использование свободных ресурсов машин, подключенных к компьютерным сетям. Этими ресурсами могут быть, например, дисковые пространства, программные средства, однако самый распространенный по использованию ресурс – вычислительные мощности машин, которые используются в обычном режиме работы лишь на несколько процентов, а то и вовсе простаивают [2]. Применение распределенной модели расчетов часто обусловлено как значительно большей распространенностью локальных вычислительных сетей (ЛВС), чем многопроцессорных кластеров, так и дороговизной последних.

Целью работы является модификация параллельного алгоритма построения области работоспособности [3] с целью реализации его в распределенной вычислительной системе. Предлагается алгоритм декомпозиции данных, отличающийся от алгоритма, описанного в работе [3], тем, что разбиение данных производится не на геометрическом представлении многомерной сетки, а на одномерном массиве состояний, что позволяет с большей простотой подойти к вопросу балансировки нагрузок между вычислительными узлами различной вычислительной мощности.

## Построение области работоспособности на основе сеточного представления

Задача параметрического синтеза технических устройств и систем [1] состоит в выборе номинальных значений внутренних параметров $\mathbf{x}_{ном} = (x_{1ном}, x_{2ном}, \ldots, x_{пном})$ исследуемого устройства, обеспечивающих максимум вероятности нахождения внутренних параметров внутри области работоспособности в течение заданного интервала времени:

$$\mathbf{x}_{ном} = \arg\max P(\mathbf{X}(\mathbf{x}_{ном}, t) \in D_X, \forall t \in [0, T]), \quad (1)$$

где $\mathbf{X}(\mathbf{x}_{ном}, t)$ – случайный процесс изменения параметров; $D_x$ – область работоспособности; $T$ – заданное время эксплуатации устройства.

Условия работоспособности задаются обычно в виде допусков на выходные параметры системы

$$\mathbf{y}_{\min} \le \mathbf{y}(t) \le \mathbf{y}_{\max}. \quad (2)$$

Математическая модель системы $\mathbf{y}(\mathbf{x}(t))$ определяет зависимость вектора выходных параметров $\mathbf{y}(t) = (y_1(t), y_2(t), \ldots, y_m(t))$ от вектора внутренних параметров $\mathbf{x}(t) = (x_1(t), x_2(t), \ldots, x_n(t))$ в виде непрерывных зависимостей

$$y_i(t) = y_i(\mathbf{x}(t)), \quad i = \overline{1,m}. \quad (3)$$

Зависимости (3) обычно задаются неявно, в алгоритмической форме, в виде численного решения систем уравнений [1, 4 – 6], которые описывают функционирование системы.

Множество точек пространства внутренних параметров, в которых выполняются условия работоспособности (2), называется областью работоспособности $D_x$.

$$D_x = \left\{\mathbf{x} \in R^n : \mathbf{y}_{\min} \le \mathbf{y}(\mathbf{x}) \le \mathbf{y}_{\max}\right\}. \quad (4)$$

Как правило, информация о форме, конфигурации области $D_x$ неизвестна. Построение этой области позволит избежать трудоемких расчетов модели систе-

мы (3) путем проверки принадлежности реализации вектора внутренних параметров области работоспособности $\mathbf{x} \in D_x$, а также позволит решить задачу параметрического синтеза путем выбора оптимального вектора параметров с использованием детерминированных критериев, – например, запаса работоспособности [4].

На значения внутренних параметров обычно накладываются ограничения (допуски производственные, расчетные и пр.)

$$x_{i\min} \le x_i \le x_{i\max}, i = \overline{1,n}. \quad (5)$$

В $n$-мерном пространстве внутренних параметров в ортогональной системе координат соотношения (5) образуют брус (параллелепипед) допусков $B_d$ [5, 6].

$$B_d = \{\mathbf{x} \in R^n \mid x_{i\min} \le x_i \le x_{i\max}, i = \overline{1,n}\}. \quad (6)$$

Описанный около области $D_x$ брус $B_o$ является ортогональным параллелепипедом с гранями, параллельными соответствующим граням бруса допусков $B_d$ в пространстве внутренних параметров. Описанный брус $B_o$ касается своими гранями границ области работоспособности. Построение параллелепипеда $B_o$ является первым шагом к представлению области работоспособности и в некоторых случаях позволяет существенно уменьшить пространство поиска [5, 6].

Следующий шаг к представлению области работоспособности $D_x$ – наложение $n$-мерной сетки на описанный параллелепипед $B_o$. Сетка может быть наложена не только на параллелепипед $B_o$, но и на брус допусков $B_d$ , однако в первом случае, как уже было отмечено, область поиска может быть в несколько раз меньше. Наложение $n$-мерной сетки осуществляется квантованием области значений каждого $i$-го параметра следующим образом. Каждая $i$-я координатная ось либо внутри отрезка допусков (5) бруса допусков $B_d$, либо внутри описанного бруса $B_o$ разбивается на $l_i$ равномерных отрезков – «квантов»[7]. Таким образом, можно сказать, что описанный брус $B_o$ или брус допусков $B_d$ представлен в виде множества непересекающихся параллелепипедов.

$$B_0 = \bigcup_{k_1=1}^{l_1} \bigcup_{k_2=1}^{l_2} \ldots \bigcup_{k_n=1}^{l_n} B_{k_1,k_2,\ldots k_n}, \quad (7)$$

где $l_i$ – количество квантов на координатной оси $i$-го параметра.

На рис. 1 схематично проиллюстрировано разбиение бруса на множество параллелепипедов (подбрусов).

В геометрическом центре каждого подбруса $B_{k1,k2,\ldots,kn}$ выбирается точка-«представитель». Если в данной точке условие работоспособности (2) выполнено, то считаем, что в каждой точке внутри $B_{k1,k2,\ldots,kn}$ оно выполняется. Имеем два подмножества подбрусов, в одном из которых условие работоспособности выполняется, а в другом – нет. Первое подмножество условно назовем подмножеством «хороших» подбрусов, второе – подмножеством «плохих» подбрусов. Таким образом, получено представление области работоспособности $D_x$ в виде множества непересекающихся параллелепипедов $B_{k1,k2,\ldots,kn}$, которое назовем сеточным представлением области работоспособности. В работах [5, 6] для обозначения этого объекта используется термин «матричное представление» (по аналогии с методом матричных испытаний, изложенным в [7]).

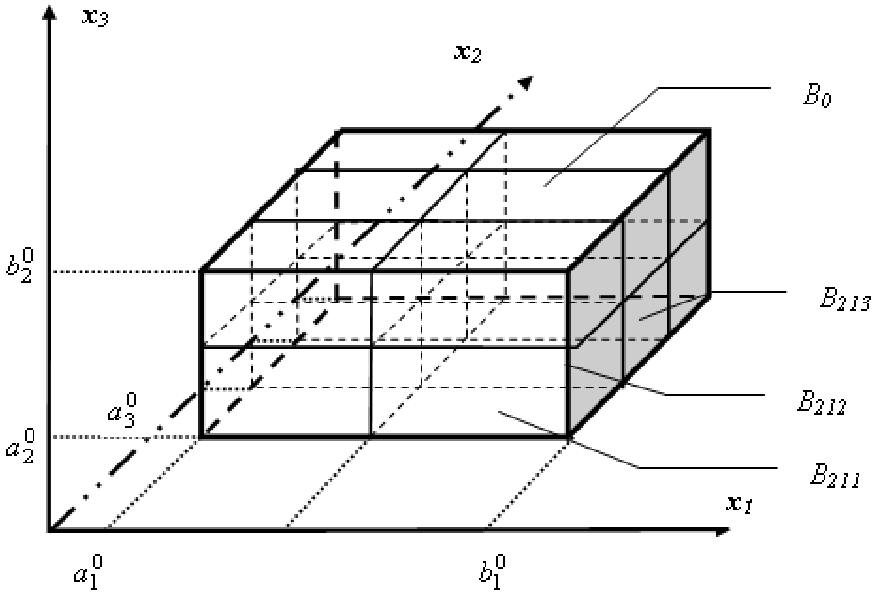



*Рис. 1.* Разбиение бруса на множество подбрусов на примере трехмерного пространства внутренних параметров.

Для хранения и использования информации, описывающей сеточное представление области работоспособности, необходимо знать координаты границ описанного параллелепипеда $B_o$ или бруса допусков $B_d$, количество квантов $l_i$ на каждой $i$-й координатной оси. Имея эту информацию, можно вычислить координаты границ подбрусов и их точек-«представителей» – реализации векторов внутренних параметров на сетке [5].

Кроме геометрических данных сетки, необходимо также хранить информацию о состоянии элементов этой сетки. Состояния подбрусов («плохой» или «хороший») описывается множеством $A$. Элементы этого множества могут принимать два значения: 0 или 1. Нуль означает, что соответствующий подбрус относится к подмножеству «плохих», единица – к подмножеству «хороших». Количество элементов в этом массиве равно произведению количества квантов каждого из параметров:

$$R = \prod_{i=1}^{n} l_i \ . \qquad (8)$$

Формулы взаимнооднозначного соответствия индекса $p$ каждого элемента массива $A[R]$ и набора $n$ индексов $(k_1,k_2,\ldots,k_n)$ подбруса $B_{k1,k2,\ldots,kn}$ (как элементов $n$-мерной сетки), а также соответствующие алгоритмы обсуждаются в работах [5, 6].

Таким образом, «хорошие» элементы массива *A[R]* представляют модель области работоспособности $D_x$. Построение такой модели области требует полного перебора точек-представителей всех подбрусов, вычисления выходных параметров (3) в этих точках и проверки условий работоспособности (2). Точность аппроксимации области $D_x$ сеточным представлением определяется количеством квантов $l_i$ на каждой $i$-й координатной оси ($i = 1,\ldots,n$) [5]. Поэтому чем выше детализация сеточного представления, тем больше расчетов модели (3) необходимо провести, что при высокой сложности модели требует больших вычислительных затрат.

### Распределение расчетов по вычислительным узлам

В рассматриваемой распределенной вычислительной системе для построения сеточного представления области работоспособности используются два основных типа узлов: диспетчер (сервер); вычислительный узел (клиент).

Компьютеры, входящие в вычислительную сеть, должны быть соединены локальной сетью по протоколу TCP/IP, также допускается подключение компьютеров через сеть Internet (протокол TCP/IP). Из указанных типов узлов обязательно присутствие обоих, однако диспетчер должен быть только один, а вычислительных узлов может быть несколько. Допустимым является также и использование одного вычислительного узла и диспетчера. Однако такая схема вычислительной системы может быть целесообразной в случае, если клиент имеет более мощные вычислительные ресурсы, чем диспетчер, и время вычисления на этом узле в сумме со временем передачи данных по сети меньше времени, которое будет потрачено на такую же задачу на машине-диспетчере. Вычислительный узел-клиент одновременно может выполнять задание, полученное только от одного сервера-диспетчера.

Процесс вычисления выходных параметров (3) в точке-представителе каждого из подбрусов $B_{k1,k2,\ldots,kn}$ выполняется независимо от аналогичных процессов в точках-представителях других подбрусов. Поэтому распараллеливание процесса построения области работоспособности проводится по данным.

Удобство хранения сеточного представления (состояний элементов сетки) области работоспособности $D_x$ в виде одномерного массива $A[R]$ в данном случае заключается в возможности его разбиения как на равные, так и на пропорциональные вычислительным мощностям узлов части.

Пусть количество вычислительных узлов, готовых взять часть расчетов, равно $k$. Тогда в случае равных вычислительных мощностей клиентов массив $A[R]$ разбивается на $k$ равных частей (или с незначительным перевесом в некоторой части, так как возможен вариант, когда массив не разделится на $k$ частей без остатка). Массив $A[R]$ можно разбить на $k$ частей, пропорциональных вычислительным мощностям клиентов, если предварительно провести тест их производительности. Таким образом, декомпозиция данных на одномерном массиве является более простой и гибкой, чем геометрическая декомпозиция на $n$-мерной сетке.

На рис. 2 изображено разбиение массива $A[R]$ на равные фрагменты, которые обрабатываются одновременно узлами-клиентами распределенной вычислительной системы.

После разбиения массива $A[R]$ на $k$ частей для обработки на узлах распределенной системы формируется $k$ заданий (для каждого из узлов). Считаем, что клиентские части распределенной системы имеют в распоряжении программные средства расчета выходных параметров (3) и проверки условия работоспособности (2). Задание представляет собой пакет данных, содержащий следующую информацию:

векторы-ключи к данному сеточному представлению (границы описанного бруса $B_o$ или бруса допусков $B_d$ по каждому параметру и количество квантов $l_i$, на которые разбит интервал значений каждого $i$-го параметра внутри данного бруса);

индексы $M_a^j$ и $M_b^j$ соответственно начала и конца части массива *A[R]* для обработки в *j*-м задании (*j = 1, 2, ...,k*).

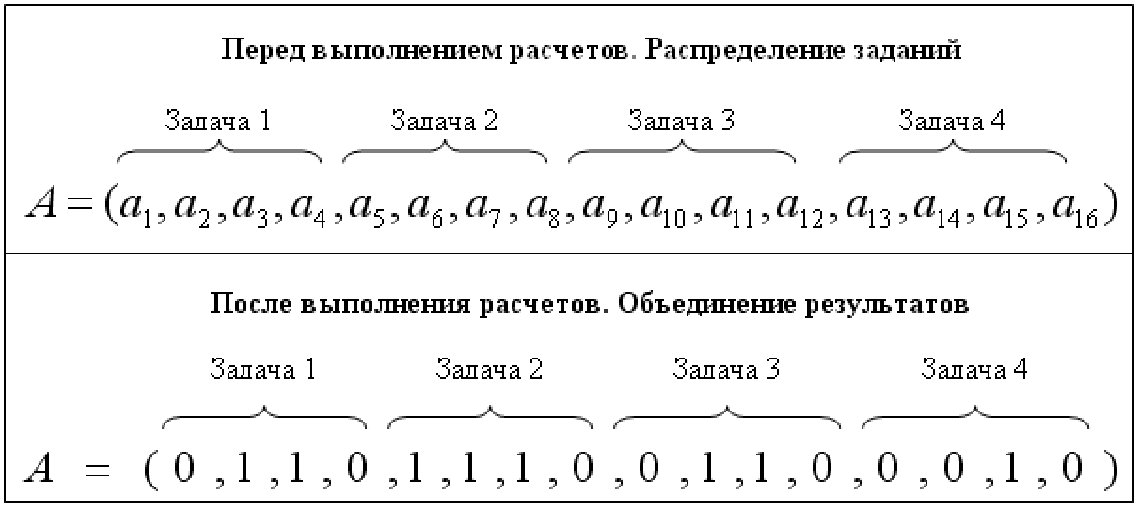



*Рис. 2*. Разбиение массива *A*[*R*] состояний для вычисления их значений на распределенной системе.

Таким образом, задание, посылаемое диспетчером вычислительному узлу, имеет следующую структуру:

1. $x_{min\,1}$, $x_{min\,2}$, …, $x_{min\,n}$;
2. $x_{max\,1}$, $x_{max\,2}$, …, $x_{max\,n}$;
3. $l_1$, $l_2$, ..., $l_n$;
4. $M_a^j$;
5. $M_b^j$,

где *j* – номер задания (*j* = 1, 2, ...,*k*).

Вычислительный узел, получив задание, проводит расчет модели в точках-представителях подбрусов, однозначно соответствующих элементам массива *A*[*R*] с номера $M_a^j$ по $M_b^j$ . Как уже было сказано, точкой-представителем для каждого подбруса может быть, например, его геометрический центр, координаты которого являются центрами соответствующих квантов параметров. Границы каждого подбруса, а следовательно, и координаты его центра можно вычислить, зная индекс элемента массива *A*[*R*] и векторы-ключи к сеточному представлению [5, 6].

После проведения вычислений в рамках выполнения *j*-й задачи клиентское приложение отправляет диспетчеру обработанную часть сеточного представления в виде части массива *A*[*R*] состояний соответствующих подбрусов.

Диспетчер, принимая от приложений-клиентов обработанные части массива состояний *A*[*R*], объединяет их в единое представление искомой области работоспособности. Диспетчерское приложение хранит в оперативной памяти параметры каждого *j*-го задания и, в соответствии с индексами $M_a^j$ и $M_b^j$, вносит в массив *A*[*R*] данные той части, которые соответствуют *j*-й задаче.

Рассмотрим распределенный алгоритм построения области работоспособности $D_x$ в виде множества непересекающихся *n*-мерных параллелепипедов. Приведенный ниже алгоритм описывает работу распределенной системы в целом, т.е. работу обоих типов узлов.

*Алгоритм.*

*Шаг 1.* Запуск приложения-диспетчера. Ожидание подключений клиентских приложений.

*Шаг 2.* Запуск клиентского приложения. Установление связи с приложением-диспетчером, сообщение о состоянии готовности получить задание. Установка режима ожидания задания.

*Шаг 3.* Диспетчер фиксирует подключение клиента, запоминает его IP-адрес. Ожидание подключений других приложений-клиентов.

*Шаг 4.* После установки связи с нужным или возможным количеством $k$ клиентских приложений по решению оператора распределенной системы в диспетчерском приложении инициируется процесс формирования заданий и их последовательная отправка клиентским приложениям.

*Шаг 5.* Клиентское приложение, получив $j$-е задание, проводит вычисления выходных параметров (3) и инициализацию части массива (с номера $M_a^j$ по $M_b^j$) состояний $A[R]$ элементов сетки. Заполненная часть массива отправляется диспетчерскому приложению.

*Шаг 6.* Диспетчерское приложение, получив от клиента рассчитанную часть сеточного представления, записывает эти результаты в массив $A[R]$.

*Шаг 7.* Получив результаты от всех клиентских приложений, диспетчерское приложение объявляет об окончании решения задачи. Остановка.

Для обеспечения надежности проведения распределенных расчетов необходимо также учесть возможность аварийного отключения от сети одного или более узлов. Для этого требуется схема диспетчеризации заданий с учетом максимально допустимого времени расчета на каждом из клиентских приложений и перераспределении заданий, с которыми не справились одни узлы, тем клиентам, которые уже выполнили свои порции расчетов и готовы принять новые задания.

Результаты экспериментов показали, что около половины общего времени решения задачи составляет передача результатов вычислений по сети. Объем передаваемых данных можно сократить путем оптимизации их структуры. В силу особенности сеточного представления, а именно того, что элементы массива состояний $A[R]$ представляют собой числа 0 и 1, можно выполнить компрессию данных. Такое сжатие позволяет уменьшить объем передаваемой информации в 8 раз.

Компрессия выполняется упаковкой восьми байт, представляющих последовательность нулей и единиц, в 1 байт, также представляющий последовательность нулей и единиц, но в виде 8 бит, т.е. для передачи информации о состоянии одного элемента сетки («хороший» или «плохой») достаточно 1 бита информации, а значения восьми компонент массива состояний умещаются в 1 байт данных.

Однако сжатая схема хранения данных неудобна для анализа и визуализации [6], так как для доступа к отдельным битам потребуются дополнительные операции, – такие как наложение битовых масок, битовых сдвигов и т.п. Поэтому сжатие данных пригодно, в первую очередь, для передачи по сети либо для хранения на носителе.

## Результаты эксперимента

Программная реализация рассмотренных алгоритмов представлена в виде первой экспериментальной версии распределенной на ЛВС системы.

В качестве примера была рассмотрена схема транзисторно-транзисторной логики (рис. 3). Область работоспособности строилась в пространстве 4 внутренних параметров – значений сопротивлений резисторов $R1$, $R2$, $R3$, $R4$ схемы.

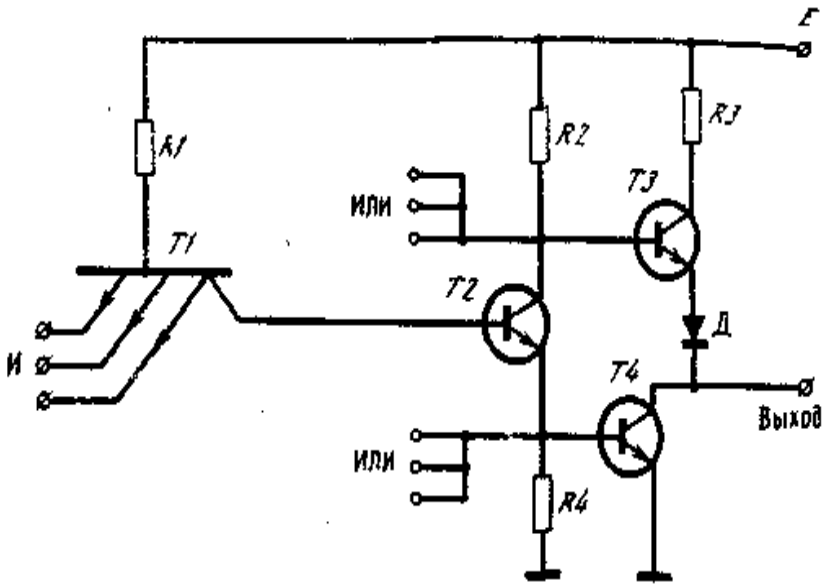



*Рис. 3*. Принципиальная схема транзисторно-транзисторной логики (ТТЛ).

При проведении численного эксперимента по построению области работоспособности для данной схемы были получены следующие результаты.

Было задействовано 5 машин локальной вычислительной сети (100 Мбит/с): диспетчерская машина (AMD Athlon XP 2400+ ~2ГГц); 4 вычислительных узла одинаковой с сервером конфигурации.

Все приложения разработаны для функционирования в операционной системе Microsoft® Windows. Однако имеется возможность подключения к распределенной системе вычислительного узла, работающего и в других операционных системах, – например, на платформе ОС семейства UNIX. В таком случае имеем гетерогенную распределенную вычислительную систему [2].

В рассматриваемом примере акцент делается на сравнении скорости решения задачи в автономном режиме, т.е. на одной машине (последовательно), со скоростью решения задачи с использованием распределенной системы (параллельно). Немаловажен и учет времени передачи данных по сети.

Серверное приложение (диспетчер) самостоятельно выполняло задачу по построению сеточного представления области $D_x$ в течение 57,5 сек. При проведении повторных экспериментов это время рознится незначительно.

В таблице приведено время, потраченное на выполнение работ по построению заданной части сеточного представления каждым вычислительным узлом.

| Машина №1 | Машина №2 | Машина №3 | Машина №4 |
|---|---|---|---|
| 12,453 сек. | 10,453 сек. | 14,704 сек. | 11,391 сек. |

Учитывая, что задание всем машинам было разослано практически одновременно (сотые доли секунды), все задействованные вычислительные узлы приступили к выполнению своей части работы одновременно. Диспетчерское приложение зафиксировало успешное окончание решения задачи приблизительно (округлено до целых) через 25 сек. после рассылки заданий на вычислительные узлы. Если учесть, что машина №2 справилась со своей задачей примерно через 10 сек., то остальные 15 сек. – это общее время передачи данных результатов по сети.

Всего без компрессии в проведенном эксперименте сервер должен был принять 40 960 000 байт – длина массива состояний *A[R]* элементов сеточного представления (по 80 квантов по каждому параметру – $80^4$ вариаций компонент векторов внутренних параметров). Сервер принимал данные в многопоточном режиме от всех клиентов одновременно.

Применение алгоритма 8-кратного сжатия передаваемых данных позволило снизить общее время передачи данных на сервер до 8 сек. По сравнению с данными, приведенными в таблице, операции по сжатию данных существенно не отразились на времени выполнения задач на вычислительных узлах.

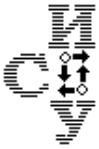
Общее время решения задачи параллельно на распределенной системе с использованием сжатия данных при передаче составило 18,5 сек., что в 3,1 раза быстрее, чем решение этой задачи последовательно (автономно на одной машине).

Из результатов эксперимента видно, насколько существенно влияние фактора объема передаваемых данных в общей скорости решения задачи с помощью технологии распределенных вычислений. Основная цель эксперимента состояла в уменьшении общего времени при построении сеточного представления области работоспособности с использованием распределенных вычислений.

## Заключение

Задача оптимального выбора параметров требует проведения множественных расчетов модели технического устройства при различных значениях внутренних параметров, многовариантного анализа, что предъявляет высокие требования к вычислительным мощностям. Набирающая в последнее время популярность концепция распределенных вычислений позволяет выполнять задачу сложных и множественных расчетов не только на специализированной технике (суперкомпьютеры), но и на обычных рабочих станциях, соединенных по сети. Выбор в пользу схемы распределенных на ЛВС расчетов часто аргументируется большей распространенностью локальных компьютерных сетей и дороговизной суперкомпьютеров.

В работе был предложен модифицированный параллельный алгоритм построения области работоспособности и исследованы проблемы реализации этого алгоритма в распределенной на ЛВС вычислительной системе. Приведен пример построения области работоспособности с помощью разработанной распределенной вычислительной системы. В результате эксперимента было достигнуто ускорение 0,32 (в 3,11 раза быстрее последовательного решения).



ЛИТЕРАТУРА

1. *Абрамов. О.В.* Параметрический синтез стохастических систем с учетом требований надежности. – М.: Наука, 1992.
2. Грид-технологии и вычисления в распределенной среде / Афанасьев А.П., Волошинов В.В., Посыпкин М.А., Сухорослов О.В., Хуторной Д.А. // Избранные доклады III Международной конференции «Параллельные вычисления и задачи управления» PACO'2006 / Ин-т пробл. упр. им. В.А. Трапезникова РАН, 2006. – С.5–16.
3. Параллельные алгоритмы построения области работоспособности / Абрамов О.В., Диго Г.Б., Диго Н.Б. Катуева Я.В. // Информатика и системы управления. – 2004. – № 2(8). – С.121–133.
4. *Абрамов О.В., Катуева Я.В., Назаров Д.А.* Оптимальный параметрический синтез по критерию запаса работоспособности. // Проблемы управления. – 2007. – № 6. – С.64-69.
5. *Катуева Я.В., Назаров Д.А.* Аппроксимация и построение областей работоспособности в задаче параметрического синтеза // Тр. междунар. симпозиума "Надежность и качество" / ПГУ. – Пенза, 2005. – С.130-134.
6. *Катуева Я.В., Назаров Д.А.* Алгоритмы анализа области работоспособности, заданной в матричной форме // Информатика и системы управления. – 2005. – №2(10). – С.118–128.
7. *Васильев Б.В., Козлов Б.А., Ткаченко Л.Г.* Надежность и эффективность радиоэлектронных устройств. – М.: Советское радио, 1964.

*Статья представлена к публикации членом редколлегии О.В. Абрамовым.*

УДК 681.3.062

© 2008 г. **И.А. Трещев**
(Комсомольский-на-Амуре государственный технический университет)

# ПОСТРОЕНИЕ МНОГОПОТОЧНЫХ ПРИЛОЖЕНИЙ ДЛЯ РАСПАРАЛЛЕЛИВАНИЯ АЛГОРИТМОВ ПЕРЕБОРА

Предлагается модификация общей схемы распараллеливания алгоритмов для решения задач, допускающих решение методом перебора с возвратом. Модифицированная схема применяется для построения многопоточных приложений на компьютерах с SMP-архитектурой. Приводятся результаты тестирования этого метода на ряде классических задач перебора.

## Введение

Метод перебора с возвратом предназначен для поиска конечных последовательностей элементов $(x_1, x_2,\ldots, x_n)$, связанных заданными отношениями. Этот метод формально был описан в пятидесятых годах прошлого столетия. Мы будем рассматривать подкласс класса задач, которые решаются методом перебора с возвратом. Опишем задачи этого подкласса. Более подробно они рассматриваются в учебном пособии [1].