

тупными и недорогими, что позволяет применять данную систему в офисных и домашних приложениях.

### Литература

1. *Reed T. R., Du Buf J. M. H.* A review of recent texture segmentation and feature extraction techniques. – CVGIP: Image Understanding, v. 57, 3, 1993, 359 – 372.
2. *Chui, C. K.* 1992 An Introduction to Wavelets. Academic Press.
3. *Змитрович А.И.* Интеллектуальные информационные системы. – Мн.: НТООО «ТетраСистемс», 1997.

## МОДЕЛИРОВАНИЕ РАБОТЫ SPMD ПРОГРАММ В ЛОКАЛЬНЫХ СЕТЯХ

### А. Е. Верхотуров

Согласно классификации Флинна [1] имеются два основных класса параллельных ЭВМ.

- SIMD (Single Instruction - Multiple Data) - одиночный поток команд и множественный поток данных. В таких ЭВМ выполняется единственная программа, но каждая команда обрабатывает массив данных. Это соответствует векторной форме параллелизма.

- MIMD (Multiple Instruction - Multiple Data) - множественный поток команд и множественный поток данных. В таких ЭВМ одновременно и независимо друг от друга выполняется несколько программных ветвей, обменивающихся данными. Такие системы обычно называют многопроцессорными. Кластерные компьютеры относятся к этому классу.

С появлением стандарта параллельного программирования MPI и реализации MPICH для кластеров на основе сетей ПЭВМ, стала актуальной задача эффективного использования аппаратных средств вычислительной сети параллельными программами. Параллельная MPI-программа запускается на узлах вычислительной сети, в виде процесса операционной системы, стандарт MPI предоставляет функции для обмена сообщениями между процессами MPI.

Различают два вида параллелизма для кластеров: параллелизм по данным и параллелизм по операциям. Параллелизм по операциям означает выполнение различных действий на различных узлах вычислительной сети. Параллелизм по данным, называемый также геометрическим, означает, что разные процессы выполняют одни и те же операции, но с разными данными.

Параллелизм по данным удобен при программировании задач с обработкой больших массивов данных тем, что параллельная программа создается и отлаживается теми же средствами, что и последовательная. Та-

кой подход в программировании называется SPMD (Single Program Multiple Data), и применяется в большинстве вычислительных задач.

Термин SPMD относится не к архитектуре компьютеров, а к модели распараллеливания программ и не является расширением систематики Флинна. SPMD обычно относится к MPP (т.е. MIMD) - системам и означает, что несколько копий одной программы параллельно выполняются в разных процессорных узлах с разными данными.

Обработываемые массивы данных распределяются по процессам параллельной программы, каждый процесс запущен на отдельном процессоре кластера. Тогда работа процесса будет представлять собой следующую схему:

- вычисления со своей частью данных,
- обмен данными.

В случае итерационного алгоритма эти действия будут происходить на каждой итерации, плюс проверка условия выхода из цикла.

Моделирование работы SPMD программ позволит изучить факторы, понижающие эффективность и разработать пути увеличения масштабируемости программ.

Самый простой способ оценить время работы программы на параллельной ЭВМ – записать его в виде закона Амдала [2].

$$R = \frac{W \cdot t}{\left( W_{\tilde{n}\tilde{e}} + \frac{W - W_{\tilde{n}\tilde{e}}}{n} \right) \cdot t + w_c \cdot t_c}$$

где  $W$  – общее число операций,  $W_{ск}$  – число скалярных операций,  $n$  – число процессоров,  $t$  – время выполнения одной операции,  $w_c$  – количество операций обмена,  $t_c$  – время выполнения одного обмена.

Время  $T(L)$ , необходимое на передачу сообщения длины  $L$ , определяется следующим образом:

$$T(L) = s + L / R,$$

где  $T$  – полное время передачи сообщения,  $s$  – латентность,  $L$  – длина сообщения,  $R$  – пропускная способность канала связи [3].

Таким образом, время работы параллельной программы на кластере является функцией от числа тактов процессора, необходимых для выполнения операций в программы, удельного веса скалярных операций алгоритма, числа процессоров кластера и их тактовой частоты, а также свойств сети: пропускной способности и латентности.

Результаты формульного моделирования представлены на рис. 1. В качестве примера была выбрана параллельная программа решения систем линейных алгебраических уравнения методом Якоби [4]. Формула

была написана для одной итерации метода. Результаты формульного моделирования очень близки к экспериментальным данным.

Однако совпадение с экспериментом не наблюдается в случаях, если число процессоров велико, либо мал размер матрицы. Объяснить причины этого позволяет имитационное моделирование процесса передачи данных по сети.

Имитационная модель реализована на C++. Объектом рассмотрения является процесс параллельной программы, который в определенный момент времени может обрабатывать данные, передавать их, либо ожидать получения данных. Время, необходимое для счета и передачи сообщений считается так же, как и в формульной модели, но для каждого процесса. Главное отличие имитационной модели заключается в том, что учитываются простои процессов из-за разбалансировки загрузки и ожидания при получении сообщений по сети.

В модели учитывалась схема передачи данных в каждой итерации. В нашем примере это коллективная операция `MPI_Allgatherv` (рис. 3).

При параллельной передаче данных достаточно не более  $\{\log_2(n)\}$  операций парного обмена от каждого процесса, где  $n$  – число процессов,  $\{x\}$  – наименьшее из целых, превышающих  $x$ . [1] Однако параллельная передача возможна только для свитча, в случае использования концентратора доступ к среде передачи данных процессы получают последовательно [3]. Согласно результатам моделирования, в результате перераспределения загрузки процессоров можно добиться более эффективного

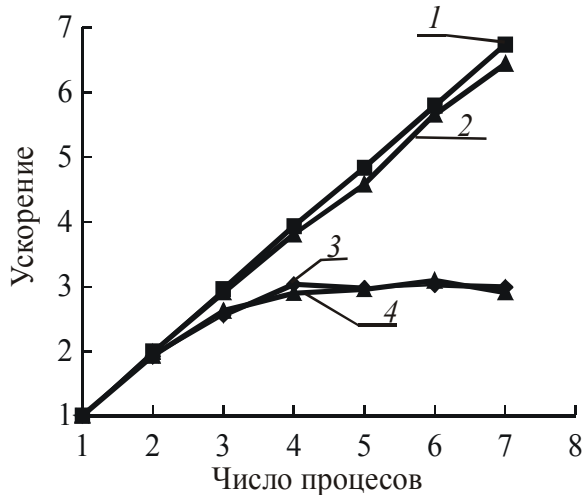


Рис. 1. Результаты формульного моделирования 1 – формула для свитча 2–эксперимент на свитче, 3 – формула для хаба, 4 – эксперимент на хабе

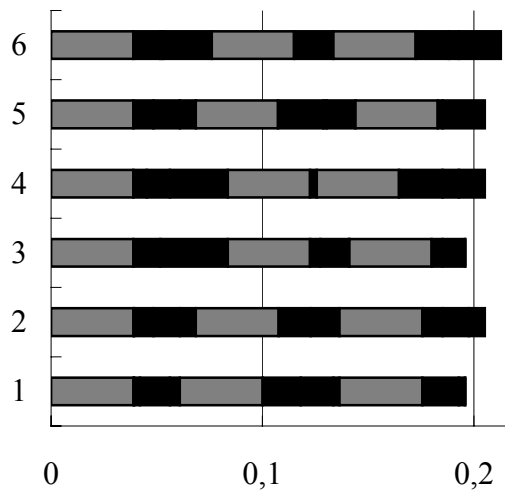


Рис. 2. Схема работы параллельной программы по результатам имитационного моделирования:

серым цветом показано время счета, черным – ожидания процессов при передаче данных

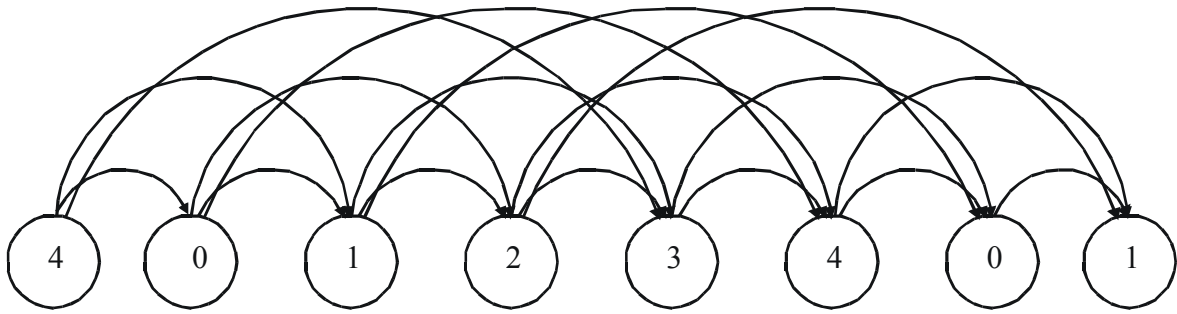


Рис. 3. Схема операции коллективной передачи данных MPI\_Allgather(v) для пяти процессов MPI

использования вычислительных ресурсов за счет неявного регламентирования порядка обращений к среде передач данных. Для примера с методом Якоби на матрицах порядка 1000 время, потраченное на передачу данных, удалось сократить на 20% для пяти процессов.

Получены формульная и имитационная модель для программы с параллелизмом по данным на примере метода простой итерации Якоби для СЛАУ. На модели показано, что в случае реализации совмещения обменов с вычислениями возможно увеличение эффективности программ за счет перераспределения нагрузки с учетом реализации операций обмена.

#### Литература

1. Шпаковский Г. И., Серикова Н. В. Программирование для многопроцессорных систем в стандарте MPI. – Мн.: БГУ, 2002. – 323 с.
2. Шпаковский Г. И. Организация параллельных ЭВМ и суперскалярных процессоров: Учеб. пособие. Мн.: Белгосуниверситет, 1996. 284 с.
3. Олифер В. Г., Олифер Н. А. Компьютерные сети. Принципы, технологии, протоколы: Учебник для вузов. 2-е изд. – СПб.: Питер, 2003. – 864 с.
4. Верхотуров А. Е. Параллельная реализация методов решения систем линейных алгебраических уравнений. // Сборник статей VII Республиканской научной конференции студентов и аспирантов Беларуси (НИРС-2002), Витебск, 2002.

## ИСПОЛЬЗОВАНИЕ ВЕЙВЛЕТ-ПРЕОБРАЗОВАНИЙ ДЛЯ ОБНАРУЖЕНИЯ ПАТОЛОГИЙ РЕЧЕВОГО АППАРАТА

С. В. Демидчик

### 1. ВВЕДЕНИЕ

Для диагностики речевых заболеваний широко используется акустический анализ. По сравнению с другими данный метод обладает рядом существенных преимуществ. Во-первых, это неинвазивный метод обследования. Во-вторых, метод позволяет обнаружить заболевания на ранних