

Дальнейшее развитие систем MRPII связано с их перерастанием в системы нового класса – "Планирование ресурсов предприятия" (Enterprise Resource Planning – ERP). Системы этого класса ориентированы на работу с финансовой информацией для решения задач управления большими корпорациями с разнесенными территориально ресурсами. Сюда включается всё то, что необходимо для получения ресурсов, изготовления продукции, её транспортировки и расчётов по заказам клиентов. Помимо перечисленных функциональных требований, к системам ERP предъявляются и новые требования по применению графики, использованию реляционных баз данных, CASE-технологий для их развития, архитектуры вычислительных систем типа "клиент-сервер" и реализации их как открытых систем.

Обработка потока документов является базовым принципом формального отображения бизнес-процессов предприятия в современных ERP-системах. Этот принцип позволяет не только формально и функционально полно "записывать" конкретную бизнес-транзакцию, но и отслеживать состояние экономического процесса. Данные потока документов служат базой для учёта, анализа и планирования.

В докладе предложена программная реализация адаптации модуля SD ERP-системы SAP R/3. В модуле используется определённый вид обработки заказов на приобретение товаров и услуг клиентом. Суть решения заключается в использовании множества связанных документов для генерации рабочего процесса по продаже и распределению товаров и услуг. Система начинается с обработки предпродажных запросов и заканчивает выписыванием счета на отгруженную продукцию и изменением данных в главной бухгалтерии. Документы в системе – это записанные транзакции, которые представляют собой бизнес-события. Торговые документы состоят из заголовка и некоторого количества позиций. В свою очередь позиции могут быть разделены на несколько партий поставки. Если документ содержит ссылку на предшествующий документ, основная часть его данных копируется из предшествующего документа. Исходными данными для формирования документов служат также записи, содержащие информацию о клиентах, материалах, ценах и т.д.

В планировании производства можно выделить следующие области: расчет прогнозных значений, укрупненное планирование сбыта и производства, планирование производства, планирование потребностей в материалах, планирование распределения ресурсов, долгосрочное планирование, укрупненное и точное планирование производственных мощностей.

В управлении производством можно выделить следующие способы: управление производством на базе производственных заказов, обработка КАНБАН на базе коллекторов затрат, серийное производство с календарным планированием норм выработки и коллектором затрат, управление технологическими заказами для промышленности с непрерывным циклом производства, проектное изготовление на базе сетевых графиков.

ПАРАЛЛЕЛЬНАЯ РЕАЛИЗАЦИЯ МЕТОДОВ РЕШЕНИЯ СИСТЕМ ЛИНЕЙНЫХ АЛГЕБРАИЧЕСКИХ УРАВНЕНИЙ

А.Е. Верхотуров

Научные руководители - Г.И. Шпаковский,

Н.В. Серикова

Белорусский государственный университет

Методы решения систем линейных алгебраических уравнений (СЛАУ) можно разделить на два класса: прямые и итерационные. Недостаток прямых методов в том, что при наличии ошибок округления, они не позволяют достичь высокой точности [1]. Итерационные методы требуют больших вычислительных затрат. Для более быстрого решения таких задач можно использовать параллельные вычисления [2]. Были созданы и исследованы параллельные алгоритмы для методов простой итерации Якоби и Гаусса-Зейделя. Программы написаны на языке C, согласно стандарту MPI, что обеспечивает их переносимость [3].

Матрица A со столбцом свободных членов b распределяется по процессам построчно, что для данных методов наиболее оптимально. Здесь под процессом понимается узел распределённой системы, участвующий в вычислениях. Кратность размера матрицы числу процессов не требуется.

Итерационная формула метода Якоби:

$$x_i^{k+1} = \frac{1}{a_{ii}} \left(b_i - \sum_{\substack{j=1 \\ j \neq i}}^n a_{ij} x_j^k \right)$$

где x_i^k - i -й элемент вектора на k -й итерации.

Для каждого i (фактически для каждой строки матрицы) вычисления могут происходить параллельно, то есть независимо от других строк. Процессы вычисляют свою часть вектора \vec{x} , после каждой итерации происходит обмен данными - каждый процесс посылает всем свою часть вектора \vec{x} , все процессы принимают новый вектор. В MPI это осуществляется с помощью операции коллективного обмена

```
MPI_Allgatherv(&X[first], size, MPI_DOUBLE, X, sendcounts, displs, MPI_DOUBLE, MPI_COMM_WORLD);
```

где X - рассылаемый вектор, $first$ - номер строки, с которой начинается часть матрицы, содержащейся в каждом процессе

Эта операция требует синхронизации процессов, и, так как она выполняется на каждой итерации, ускорение вычислений не зависит от количества итераций. Кроме того, после каждой итерации необходимо оценить погрешность. Погрешность вычисляется как норма разности векторов на текущей и предыдущей итерации только в процессе с номером 0. Этот процесс должен передать всем остальным процессам информацию о том, нужно ли выполнять следующую итерацию, или вычисления закончены. Для этого используется операция коллективного обмена MPI:

```
MPI_Bcast(&bResult, 1, MPI_INT, 0, MPI_COMM_WORLD),
```

Передаётся одно значение типа `int`, находящееся по адресу `&bResult` от процесса с номером 0 всем остальным процессам в переменную с адресом `&bResult`.

Итерационная формула метода Гаусса-Зейделя:

$$x_i^{k+1} = \frac{1}{a_{ii}} \left(b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{k+1} - \sum_{j=i+1}^n a_{ij} x_j^k \right); i = \overline{1, n}; k = 0, 1, 2, \dots$$

Метод Гаусса-Зейделя в таком виде скалярный, поэтому идея Зейделя использовать на k итерации часть уже вычисленных значений k итерации используется только в рамках одного процесса, что уменьшает сходимость и может увеличить число итераций по сравнению с последовательной реализацией. Для некоторых задач время решения с применением распределённых вычислений может увеличиться.

Ведутся дальнейшие исследования по возможной модификации алгоритма с использованием неблокирующих операций обмена.

Для тестирования использовался кластер из четырёх ПЭВМ под управлением Windows 2000, соединённых Fast Ethernet 100Мб/с. Для этого на каждую ПЭВМ был установлен MPICH 1.2.1 - реализация MPI для Windows NT.

Тестирование происходило на пятидиагональных матрицах 160x160, 500x500, 1100x1100, 2852x2852, построенных для реальных задач. Полученное ускорение представлено на рис. 1. Ускорение для метода Гаусса-Зейделя на матрицах размерности до 500x500 незначительно превышает единицу, а для размерностей 200x200 - меньше единицы. Это означает, что использование вычислительных сетей, на которых производились эксперименты, увеличивали время решения задачи, а не уменьшали, и их использование неэффективно. Однако для размерностей, превышающих 1000x1000 получен выигрыш во времени. При увеличении размерности матриц отношение операций пересылок к полезным операциям уменьшается. Однако существует ограничение на размер матриц из-за конечного размера оперативной памяти в узлах.

Ускорение определено как отношение времени решения задачи на одной ЭВМ ко времени решения на вычислительной сети и, несмотря на то, что для метода Якоби ускорение выше, чем для метода Гаусса-Зейделя, время решения задач различалось примерно в 2 раза в пользу Гаусса-Зейделя, так как этот метод требует меньшее число итераций. Выбор метода и параметров вычислительной системы должны основываться на параметрах решаемой задачи.

Ускорение метода Якоби					Ускорение метода Гаусса-Зейделя				
Размер матрицы	1	Число процессов			Размер матрицы	1	Число процессов		
		2	3	4			2	3	4
160x160	1	0,893	0,817	0,662	160x160	1	0,619	0,461	0,384
500x500	1	1,651	1,953	2,198	500x500	1	1,053	1,201	1,323
1100x1100	1	1,586	2,039	2,554	1100x1100	1	1,610	2,014	2,330
2852x2852	1	1,676	2,712	3,188	2852x2852	1	1,382	2,678	3,058

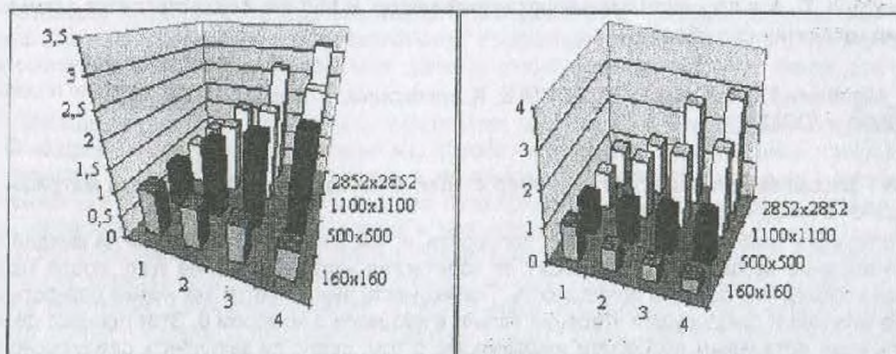


Рис 1 - Ускорение для методов Якоби и Гаусса-Зейделя для вычислительной сети из двух, трёх и четырёх ПЭВМ.

Литература.

1. Мулярчик С. Г. Численные методы: Конспект лекций. – Мн., БГУ, 2001 -127 с
2. Г. И. Шпаковский. Организация параллельных ЭВМ и суперскалярных процессоров: Учеб. Пособие. – Мн.: Беллосуниверситет, 1996. – 284 с.ил
3. W Gropp, E Lusk, and A.Skjei um Using MPI: Portable Parallel Programming with the Message-Passing Interface Second edition, published in 1999 by MIT Press, 371 pages.

**ПРИМЕНЕНИЕ ГРАФИЧЕСКИХ ФАЙЛОВ AUTOCAD ДЛЯ
ВВОДА ДАННЫХ В ОБЪЕКТНО-ОРИЕНТИРОВАННОМ
САПР**

А.В. Масилевич

Научный руководитель – Е.И. Махаринский
УО «Витебский государственный технологический университет»

В системах САПР (инженерных и научно-технических) часто возникает необходимость во вводе данных, описывающих геометрию деталей, инструментов, макетов, траекторий движения. В современных условиях при проведении проектных работ, эти данные ещё до работы с САПР проектировщик сохраняет в электронном виде, построив чертёж или схему в САД-системе (AutoCAD, T-FLEX, Solid Works)

Предлагается использовать такую электронную документацию в качестве входных данных прикладных САПР, избавляясь тем самым от необходимости построения сложных графических интерфейсов.

САПР следует передать файл, содержащий изображение из которого можно получить всю необходимую информацию. Главными требованиями к формату такого файла будут являться: векторное представление графики, возможность быстрого получения необходимой информации