Федеральное агентство связи (Россвязь)

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Сибирский государственный университет телекоммуникаций и информатики»

(СибГУТИ)

09.03.01.Б1.Б.14 – АВС

№ кода и наименование направления подготовки

**РАСЧЁТНО-ГРАФИЧЕСКОЕ ЗАДАНИЕ**

по дисциплине «Архитектура вычислительных систем»

Вариант № 2

Выполнил:

студент гр. ИП-411 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ / Выдрин А.Э. /

подпись

Проверил:

доцент кафедры ВС

к.т.н. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ / А.В. Ефимов /

ОЦЕНКА, подпись

Новосибирск 2016

ОГЛАВЛЕНИЕ

[1. ОТВЕТ НА ПЕРВЫЙ ВОПРОС 3](#_Toc468310170)

[1.1. ЗАДАНИЕ 1 3](#_Toc468310171)

[1.2. ОТВЕТ 3](#_Toc468310172)

[1.2.1 Сравнительный анализ вычислительных систем с архитектурами MISD и MIMD. 3](#_Toc468310173)

[1.2.2 Пример функциональных структур промышленных ВС.](#_Toc468310174) 6

[2. ОТВЕТ НА ВТОРОЙ ВОПРОС](#_Toc468310176) 9

[2.1. ЗАДАНИЕ 2](#_Toc468310177) 9

[2.2. ОТВЕТ](#_Toc468310177) ……………………………………………………………………...10

2.2.1 Распределение данных по вычислителям ВС……………..…….…...10

2.2.2 Блок-схема p -алгоритма умножения матриц………………….……12

2.2.3 Рассчитаем максимум накладных расходов………………………....13

[3. СПИСОК ИСТОЧНИКОВ](#_Toc468310176) 14

# ОТВЕТ НА ПЕРВЫЙ ВОПРОС.

## ЗАДАНИЕ 1

Выполнить сравнительный анализ вычислительных систем с архитектурами MISD и MIMD. Привести примеры функциональных структур промышленных ВС.

## ОТВЕТ

**1.2.1 Сравнительный анализ вычислительных систем с архитектурами MISD и MIMD**

* Архитектуры MISD и MIMD относятся к вычислительным системам – то есть к композиции аппаратурно-программных средств, предназначенных для параллельной обработки данных.
* Обе архитектуры имеют “множественность” потоков команд. Множественность характеризуется количеством одновременно реализуемых потоков команд.
* Анализируя MIMD и MISD архитектуры, можно выделить, что в архитектуре MIMD допускается обработка несколькими потоками команд несколько потоков данных, в то время как в MISD допускается обработка несколькими потоками команд только одного потока данных, то есть MIMD компьютер в сравнении с MISD может параллельно выполнять совершенно разные программы.
* В MISD архитектуре информация на выходе одного процессора является входной информацией для другого процессора, а в MIMD каждый процессор может работать со своим собственным потоком данных, который недоступен для других процессоров.
* ЭВМ с архитектурой MISD насчитывается довольно мало по сравнению с MIMD архитектурой в связи с тем, что MIMD зачастую больше подходят для общих методик параллельных данных так как они обеспечивают лучшее масштабирование и использование вычислительных ресурсов.
* Архитектура MIMD дает большую гибкость по сравнению с MISD так как

может работать как однопользовательская система, обеспечивая высокопроизводительную обработку данных для одной прикладной задачи, так и многопрограммная машина, выполняющая множество задач параллельно, и как некоторая комбинация этих возможностей.

* По типу ВС MISD архитектура бывает конвейерной:

В такой системе реализуется принцип конвейерной (магистральной) обработки, который основан на разбиении всего процесса на последовательно выполняемые этапы. Конвейеры оперируют c векторами данных, которые являются одномерными массивами или одномерными упорядоченными совокупностями элементов данных одного типа. Высокий уровень быстродействия был достигнут в конвейерных ВС за счет мультиконвейерности (параллельной работы множества конвейеров) и конвейеризации на микроуровне (на уровне фаз выполнения арифметическиx операций). Конвейерные вычислительные системы (ВС) относились к числу самых популярных высокопроизводительных средств обработки информации в 70-х и 80-х годах 20 века. Они обеспечивали быстродействие порядка 10 опер./с.

* MIMD архитектура по типу ВС делится на:

1. *ВС с распределенной памятью (Distributed memory):*

Процессор может обращаться к локальной памяти, может посылать и получать сообщения, передаваемые по сети, соединяющей процессоры. Сообщения используются для осуществления связи между процессорами или, что эквивалентно, для чтения и записи удаленных блоков памяти.

1. *Компьютеры с общей (разделяемой) памятью (True shared memory):*

Все процессоры совместно обращаются к общей памяти, обычно, через шину или иерархию шин. Частота обращений к общей памяти может быть уменьшена за счет сохранения копий часто используемых данных в кэш-памяти, связанной с каждым процессором. Доступ к этому кэш-памяти намного быстрее, чем непосредственно доступ к общей памяти.

*3)Компьютеры с виртуальной общей (разделяемой) памятью (Virtual shared memory) :*

Каждый процессор имеет собственную локальную память и может обращаться к локальной памяти других процессоров, используя "глобальный адрес". Если "глобальный адрес" указывает не на локальную память, то доступ к памяти реализуется с помощью сообщений, пересылаемых по коммуникационной сети.

В качестве **вывода** хотелось бы сказать, что и MIMD, и MISD архитектуры имеют свои преимущества и недостатки.

MIMD-системы характеризуются невысокой функциональной сложностью, дают больше возможностей, в системе легко осуществляется реконфигурация структуры путем добавления или удаления функциональных модулей, *но* при этом ограниченность производительности системы пропускной способностью общей шины и ухудшение общей производительности системы при ее расширении путем добавления модулей, а также отказ общей шины приводит к выходу из строя всей системы.

MISD-системы позволяют сравнивать результаты вычислений в целях обнаружения отказов, что гарантирует отказоустойчивость; они могут стать полезными для разработки новых концепций в теории и практике построения вычислительных систем, *но* при этом вычислительных машин такого класса практически нет и трудно привести пример их успешной реализации; также они дорогие и низко производительные по отношению к MIMD-системам и имеют более высокое время выполнения задач.

**1.2.2 Пример функциональной структуры промышленной ВС**



Рис. 1.2.1

Компьютер CRAY T3D - это массивно-параллельный **MIMD** компьютер с распределенной памятью, объединяющий от 32 до 2048 процессоров. Распределенность памяти означает то, что каждый процессор имеет непосредственный доступ только к своей локальной памяти, а доступ к данным, расположенным в памяти других процессоров, выполняется другими, более сложными способами.

CRAY T3D подключается к хост-компьютеру (главному или ведущему), роль которого, в частности, может исполнять CRAY Y-MP C90. Вся предварительная обработка и подготовка программ, выполняемых на CRAY T3D, проходит на хосте (например, компиляция). Связь хост-машины и T3D идет через высокоскоростной канал передачи данных с производительностью 200 Mбайт/с

Массивно-параллельный компьютер CRAY T3D работает на тактовой частоте 150MHz и имеет в своем составе три основные компоненты: сеть межпроцессорного взаимодействия (или по-другому коммуникационную сеть), вычислительные узлы и узлы ввода/вывода.

*Вычислительный узел* состоит из двух процессорных элементов (ПЭ), сетевого интерфейса контроллера блочных передач. Оба процессорных элемента, входящие в состав вычислительного узла, идентичны и могут работать независимо друг от друга.

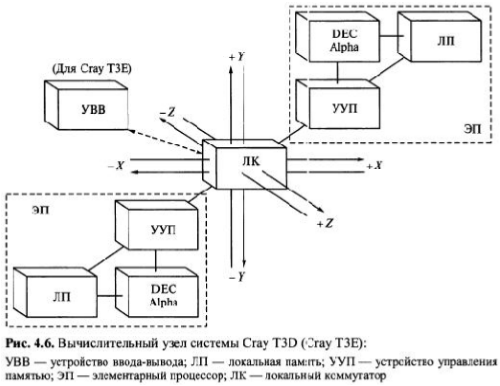


Рис. 1.2.2

*Процессорный элемент.* Каждый ПЭ содержит микропроцессор, локальную память и некоторые вспомогательные схемы.

*Микропроцессор* - это 64-х разрядный RISC (Reduced Instruction Set Computer) процессор ALPHA фирмы DEC, работающий на тактовой частоте 150 MHz. Микропроцессор имеет внутреннюю кэш-память команд и кэш-память данных, **MISD** архитектуру.

Объем *локальной памяти* ПЭ - 8 Mслов. Локальная память каждого процессорного элемента является частью физически распределенной, но логически разделяемой (или общей), памяти всего компьютера. В самом деле, память физически распределена, так как каждый ПЭ содержит свою локальную память. В тоже время, память разделяется всеми ПЭ, так как каждый ПЭ может обращаться к памяти любого другого ПЭ, не прерывая его работы.

Обращение к памяти другого ПЭ лишь в 6 раз медленнее, чем обращение к своей собственной локальной памяти.

*Сетевой интерфейс* формирует передачи перед посылкой через коммуникационную сеть другим вычислительным узлам или узлам ввода/вывода, а также принимает приходящие сообщения и распределяет их между двумя процессорными элементами узла.

*Контроллер блочных передач* - это контроллер асинхронного прямого доступа в память, который помогает перераспределять данные, расположенные в локальной памяти разных ПЭ компьютера CRAY T3D, без прерывания работы самих ПЭ.

*Коммуникационная сеть* обеспечивает передачу информации между вычислительными узлами и узлами ввода/вывода с максимальной скоростью в 140M байт/с. Сеть образует трехмерную решетку, соединяя сетевые маршрутизаторы узлов в направлениях X, Y, Z. Каждая элементарная связь между двумя узлами - это два однонаправленных канала передачи данных, что допускает одновременный обмен данными в противоположных направлениях.

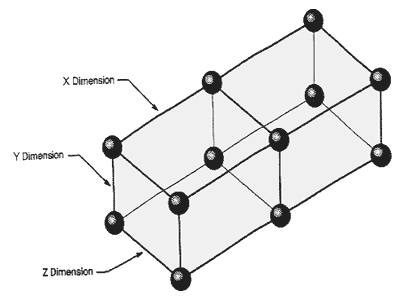


Рис. 1.2.2

Поскольку CRAY T3D - это компьютер с распределенной памятью, то взаимодействие процессоров, в основном, осуществляется посредством передачи сообщений друг другу. Отсюда два других замедляющих фактора - *время инициализации посылки сообщения* (латентность) и собственно *время передачи сообщения по сети*. Максимальная скорость передачи достигается на больших сообщениях, когда латентность, возникающая лишь в начале, не столь заметна на фоне непосредственно передачи данных.

*Возможность асинхронной посылки сообщений и вычислений*. Если или аппаратура, или программное обеспечение не поддерживают возможности проводить вычислений на фоне пересылок, то возникнут неизбежные накладные расходы, связанные с ожиданием полного завершения взаимодействия параллельных процессов.

Для достижения эффективной параллельной обработки необходимо добиться *равномерной загрузки всех процессоров*. Если равномерности нет, то часть процессоров неизбежно будет простаивать, ожидая остальных, хотя в этот момент они могли бы выполнять полезную работу. Иногда равномерность получается автоматически, например, при обработке прямоугольных матриц достаточно большого размера, однако уже при переходе к треугольным матрицам добиться хорошей равномерности не так просто.

Если один процессор должен вычислить некоторые данные, которые нужны другому процессору, и если второй процесс первым дойдет до точки приема соответствующего сообщения, то он с неизбежностью будет простаивать, ожидая передачи. Для того чтобы минимизировать *время ожидание прихода сообщения* первый процесс должен отправить требуемые данные как можно раньше, отложив независящую от них работу на потом, а второй процесс должен выполнить максимум работы, не требующей ожидаемой передачи, прежде, чем выходить на точку приема сообщения.

Чтобы не сложилось совсем плохого впечатления о массивно-параллельных компьютерах, надо заканчивать с негативными факторами, потому последний фактор - это *реальная производительность одного процессора*. Разные модели микропроцессоров могут поддерживать несколько уровней кэш-памяти, иметь специализированные функциональные устройства, регистровую структуру и т.п. Каждый микропроцессор, в конце концов, может иметь векторно-конвейерную архитектуру и в этом случае ему присущи практически все те факторы, которые мы обсуждали в лекции, посвященной особенностям программирования векторно-конвейерных компьютеров.

К сожалению, на работу каждой конкретной программы сказываются в той или иной мере все эти факторы одновременно, дополнительно усугубляя ситуацию с эффективностью параллельных программ. Однако в отличие от векторно-конвейерных компьютеров все изложенные здесь факторы, за исключением быть может последнего, могут снизить производительность не в десятки, а в сотни и даже тысячи раз по сравнению с пиковыми показателями производительности компьютера. Добиться на этих компьютерах, в принципе, можно многого, но усилий это может потребовать во многих случаях очень больших.

# ОТВЕТ НА ВТОРОЙ ВОПРОС

* 1. **ЗАДАНИЕ 2**

Построить блок-схему p -алгоритма умножения матриц:

A[1:Q;1:R], B[1:S;1:Q] обеспечивающего распределение в элементарных машинах ВС элементов результирующей матрицы по вертикальным полосам.

Отыскать максимум коэффициента ε накладных расходов при реализации

p -алгоритма на вычислительной системе, имеющей следующие параметры:

– разрядность *l =* 64;

– полосу пропускания канала между машинами 10 Гигабод *(Гигабит/с);*

– время выполнения операции сложения *tc* = 0,3 нс

– время выполнения операции умножения *ty* = 0,9 нс

A[1:Q;1:R] ** B[1:S;1:Q] =C[1:S;1:R]

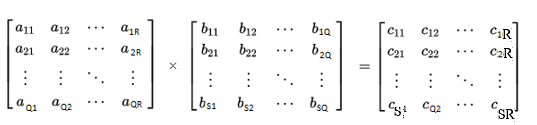
* 1. **ОТВЕТ**

**2.2.1 Распределение данных по вычислителям ВС**

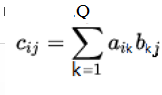
Произведение матриц Ψ[1:Q;1:R] ** Ω[1:S;1:Q] =C[1:S;1:R] можно представить в следующем виде:

a – элементы матрицы Ψ

b – элементы матрицы Ω



Элементы матрицы C будут вычисляться по формуле –

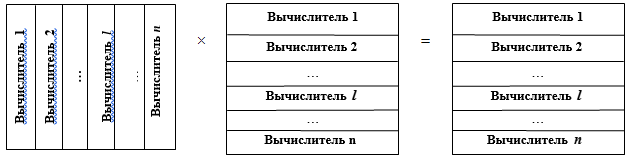
 ,

где i=1..S,j=1..R

Размеры матриц A и В достаточно большие, следовательно R>>n, S>>n, Q>>n, где n – кол-во вычислителей.

Алгоритм умножения матриц должен быть построен так, чтобы каждый вычислитель рассчитывал свой элемент матрицы C. Для построения алгоритма требуется распределить по вычислителям элементы матриц А и B, требуется однородное распределение, так как оно обеспечивает однородную нагрузку на вычислители. Для однородного распределения можно поступить так:

матрицу Ω нужно разделить на n одинаковых горизонтальных полос и каждая полоса размещается в своем вычислителе. Матрицу Ψ следует разделить на n вертикальных полос.



Ψ ** Ω = C

Первый вычислитель рассылает всем остальным вычислителям первый столбец из своей полосы матрицы A. После этого все вычислители производят вычисления своих элементов аij, используя элементы своей вертикальной полосы матрицы А. Затем первый вычислитель пересылает остальным второй столбец из первой полосы матрицы B, потом все вычислители производят соответствующие расчеты. После рассылки всех столбцов первого вычислителя матрицы B, рассылками будут заниматься последовательно второй, третий и n-ый вычислитель. В результате все элементы матрицы C будут рассчитаны, но окажутся распределенными между вычислителями по горизонтальным полосам.

Построение блока схемы р-алгоритма l-ой ветви параллельной программы, которая будет реализовываться на l-ом вычислителе, α – переменная, соответствующая номеру вычислителя.

**2.2.2 Блок-схема p-алгоритма умножения матриц**

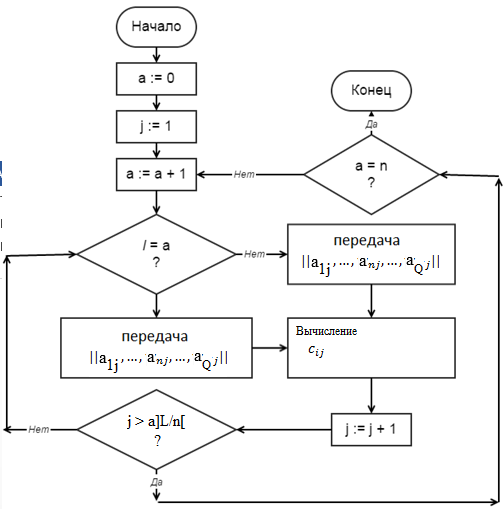


Рис. 2.1

α — номер передающего вычислителя;

{1, 2, ... , α -1, α + 1, ..., n } — номера принимающих вычислителей

]Q / n [ (l -1) < j ≤ ] Q / n [l;

l — номер ветви P-алгоритма

aij — элемент ij матрицы A.

bij — элемент ij матрицы B.

cij — элемент ij матрицы C.

Q – количество строк матрицы A и количество столбцов матрицы B;

R- количество столбцов матрицы A;

S – количество строк матрицы B;

**2.2.3 Рассчитаем максимум накладных расходов**

tn – время пересылки;

l – разрядность;

v – скорость обмена;

ρ – минимальный допустимый размер матриц, при котором еще целесообразно решение задачи на n вычислителях. Максимальные накладные расходы будут при ρ=1;

ty – время умножения;

tc – время сложения;

Для данной задачи:

l =64;

v =1010 бод;

ρ =1;

ty =0,9 нс;

*tc* = 0,3 нс;

tn = l/ ν=64/1010=6,4 нс;

Показатель накладных расходов, где t – время, которое расходуется на организацию и собственно реализацию обменов информации, T – время на счет, выполнения арифметических, логических и других операций. Оценим  для алгоритма умножения матриц.

Пусть tn – время пересылки одного слова (элемента матрицы); ty и tc – время выполнения операций умножения и сложения. Тогда эффективность параллельного алгоритма (р–программы) умножения матриц большого размера можно характеризовать показателями:

Ответ: максимум коэффициента ε накладных расходов при реализации р – алгоритма на вычислительной системе составляет .

# СПИСОК ИСТОЧНИКОВ

1. Хорошевский В.Г. Архитектура вычислительных систем. – М.: МГТУ им. Н.Э. Баумана, 2008. – 520 с.
2. Грушев А.В. [электронный ресурс] // Cray X-MP. URL: https://ru.wikipedia.org/wiki/Cray\_X-MP (дата обращения:17.12.16)
3. Бройдо В.Л. Вычислительные системы, сети и телекоммуникации. Учебник для вузов. 2-е изд. – СПб.: Питер,2004
4. Внуков С.Н. [электронный ресурс] // MISD и MIMD компьютеры. URL: http://studopedia.ru/2\_42751\_MISD-kompyuteri.html (дата обращения:16.12.16)