Министерство информационных технологий и связи РФ

Сибирский Государственный Университет Телекоммуникаций и Информатики

Кафедра Вычислительных Систем

**Расчетно-графическое задание**

**по дисциплине: «Архитектура Вычислительных Систем»**

13 вариант

Выполнил:

студент гр. П-81

Руднев Д.А

Проверил:

Хорошевский В.Г.

Новосибирск, 2010г

Вариант 13.

**Задание:**

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

2. Произвести численный расчет и построить графики для функций надежности  и готовности *s*(i,*t*) ЭВМ, обладающей следующими техническими параметрами:

* средним временем безотказной работы =106 ч.
* интенсивностью восстановления *µ*=10 1/ч.

**Задание №1.**

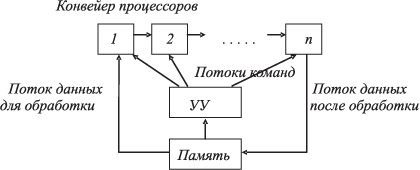
**Сравнительный анализ вычислительных систем с архитектурами MISD и MIMD**

Архитектуры MISD, MIMD относятся к вычислительным системам. В этих архитектурах имеет место “множественность” потоков команд. Множественность характеризуется количеством одновременно реализуемых потоков команд. Архитектура MIMD реализует также “множественность” потоков данных, в отличие от архитектуры MISD, для которой характерен “одиночный” поток данных.

**MISD** (Multiple Instruction stream/ Single Data stream) представляет собой системы с множественным потоком команд и одиночным потоком данных.

Данная структура содержит цепочку последовательно соединенных процессоров так, что информация на выходе одного процессора является входной информацией для другого процессора.

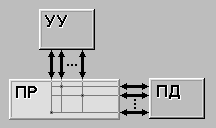
Принцип обработки показан на следующем рисунке:



С помощью множественного потока команд, поступающего в процессоры, решение задачи для заданного потока данных как бы развертывается последовательно в конвейерной цепочке этих процессоров. По этой причине системы со структурой типа MISD принято называть конвейерными.

Конвейерный процессор является простейшим вариантом конвейерной ВС. В самом деле, конкретная пара операндов в конвейере последовательно обрабатывается всеми элементарными блоками обработки. С другой стороны все эти блоки работают параллельно, но над своими операндами данных.

**MIMD** (Multipe Instruction stream / Multiple Data stream) - множественный поток команд и множественный поток данных. Базовой моделью вычислений в этом случае является совокупность независимых процессов, эпизодически обращающихся к разделяемым данным. В такой системе каждый процессорный элемент выполняет свою программу достаточно независимо от других процессорных элементов. Класс [MIMD](http://zeit-geist.narod.ru/page.file/class_3mimd.htm) чрезвычайно широк, поскольку включает в себя всевозможные мультипроцессорные системы: Cm\*, C.mmp, CRAY Y-MP, Denelcor HEP,BBN Butterfly, Intel Paragon, CRAY T3D и многие другие. Архитектура MIMD дает большую гибкость: при наличии адекватной поддержки со стороны аппаратных средств и программного обеспечения MIMD может работать как однопользовательская система, обеспечивая высокопроизводительную обработку данных для одной прикладной задачи, как многопрограммная машина, выполняющая множество задач параллельно, и как некоторая комбинация этих возможностей. К тому же архитектура MIMD может использовать все преимущества современной микропроцессорной технологии на основе строгого учета соотношения стоимость/производительность. В действительности практически все современные многопроцессорные системы строятся на тех же микропроцессорах, которые можно найти в персональных компьютерах, рабочих станциях и небольших однопроцессорных серверах.



MIMD компьютер имеет N процессоров, независимо исполняющих N потоков команд и обрабатывающих N потоков данных. Каждый процессор функционирует под управлением собственного потока команд, то есть MIMD компьютер может параллельно выполнять совершенно разные программы.

MIMD архитектуры далее классифицируются в зависимости от физической организации памяти. Исходя из организации памяти, различают следующие типы параллельных архитектур:

*• Компьютеры с распределенной памятью*

Процессор может обращаться к локальной памяти, может посылать и получать сообщения, передаваемые по сети, соединяющей процессоры.

• *Компьютеры с общей (разделяемой) памятью*

Все процессоры совместно обращаются к общей памяти, обычно, через шину или иерархию шин.

*• Компьютеры с виртуальной общей (разделяемой) памятью*

Общая память как таковая отсутствует. Каждый процессор имеет собственную локальную память и может обращаться к локальной памяти других процессоров, используя "глобальный адрес".

MIMD архитектуры с распределенной памятью классифицируются по пропускной способности коммутирующей сети. Например, в архитектуре, в которой пары из процессора и модуля памяти соединены сетью с топологий решетка, каждый процессор имеет одно и то же число подключений к сети вне зависимости от числа процессоров компьютера. С другой стороны в архитектуре, имеющей сеть с топологий гиперкуб, число соединений процессора с сетью является логарифмической функцией от числа процессоров.

Многопроцессорные системы за годы развития вычислительной техники претерпели ряд этапов своего развития. Исторически первой стала осваиваться технология SIMD. Однако в настоящее время наметился устойчивый интерес к архитектурам MIMD. Этот интерес главным образом определяется двумя факторами:

**1**.Архитектура MIMD дает большую гибкость: при наличии адекватной поддержки со стороны аппаратных средств и программного обеспечения MIMD может работать как однопользовательская система, обеспечивая высокопроизводительную обработку данных для одной прикладной задачи, как многопрограммная машина, выполняющая множество задач параллельно, и как некоторая комбинация этих возможностей.

**2**.Архитектура MIMD может использовать все преимущества современной микропроцессорной технологии на основе строгого учета соотношения стоимость/производительность. В действительности практически все современные многопроцессорные системы строятся на тех же микропроцессорах, которые можно найти в персональных компьютерах, рабочих станциях и небольших однопроцессорных серверах.

**Пример функциональных структур промышленных ВС**

Гетерогенный Процессор Элемента (HEP) был введен Denelcor в 1982 как первый в мире коммерческий компьютер MIMD. Система HEP, поскольку имя подразумевает, была соединена от многих гетерогенных компонентов - процессоры, модули памяти данных, и модули ввода / вывода. Компоненты были связаны через переключенную сеть.

Единственный процессор в системе HEP был довольно нетрадиционным; через "слово статуса программы очередь (PSW)," до пятидесяти процессов могли быть поддержаны в аппаратных средствах сразу. Восьмиэтапный трубопровод инструкции позволил инструкциям от восьми различных процессов продолжаться сразу. Фактически, только одной инструкции от данного процесса позволили присутствовать в трубопроводе в любом пункте вовремя. Поэтому, полная пропускная способность процессора 10 МИЛЛИОНОВ КОМАНД В СЕКУНДУ могла только быть достигнута, когда восемь или больше процессов были активными; никакой единственный процесс не мог достигнуть пропускной способности, больше чем 1.25 МИЛЛИОНА КОМАНД В СЕКУНДУ. Этот тип мультипронизывания обработки классифицирует HEP как процессор барреля.

Процессы были классифицированы или как на уровне пользователя или как на уровне наблюдателя. Процессы на уровне пользователя могли создать процессы на уровне наблюдателя, которые использовались, чтобы управлять процессами на уровне пользователя и выполнить ввод / вывод. Процессы того же самого класса были обязаны быть сгруппированными в одну из семи пользовательских задач и семи задач наблюдателя.

Каждый процессор, в дополнение к очереди PSW и трубопроводу инструкции, содержал память инструкции, 2 048 64-битовых регистров общего назначения и 4 096 постоянных регистров. Постоянные регистры были дифференцированы фактом, что только процессы наблюдателя могли изменить свое содержание. Интересно, сами процессоры не содержали памяти данных; вместо этого, модули памяти данных могли отдельно быть присоединены к переключенной сети.

**З адание №2.**

Произвести численный расчет и построить графики для функций надежности  и готовности *s*(i,*t*) ЭВМ, обладающей следующими техническими параметрами:

* средним временем безотказной работы =106 ч.
* интенсивностью восстановления *µ*=10 1/ч.

Известно, что в нормальных условиях эксплуатации ЭВМ интенсивность отказа а функция надежности и математическое ожидание времени *безотказной работы* соответственно равны:

.

Подставляя известные нам данные получим следующую функцию для расчета надежности:

*r*(*t*)=exp(-1/\**t*)

*r*(*t*)=exp(-*t*/106);

Рассчитаем значения функции и построим график:

|  |  |
| --- | --- |
| **t, ч** |  |
| 1\*106 | 0,367879 |
| 2\*106 | 0,135335 |
| 3\*106 | 0,049787 |
| 4\*106 | 0,018316 |
| 5\*106 | 6,738\*10-3 |
| 6\*106 | 2,479\*10-3 |
| 7\*106 | 9,119\*10-4 |
| 8\*106 | 3,355\*10-4 |
| 9\*106 | 1,234\*10-4 |
| 10\*106 | 4,54\*10-5 |

Теперь рассчитаем значения функции готовности. Расчет будем производить по следующим формулам:



. ,

для начальных состояний ЭВМ , причем  соответствует состоянию отказа, а  – работоспособному состоянию машины, где *λ*=1/.

*s*(0,*t*)=10/(10+1/106)-10/(10+1/106)\*exp((-*t)*\*(10+1/106))=

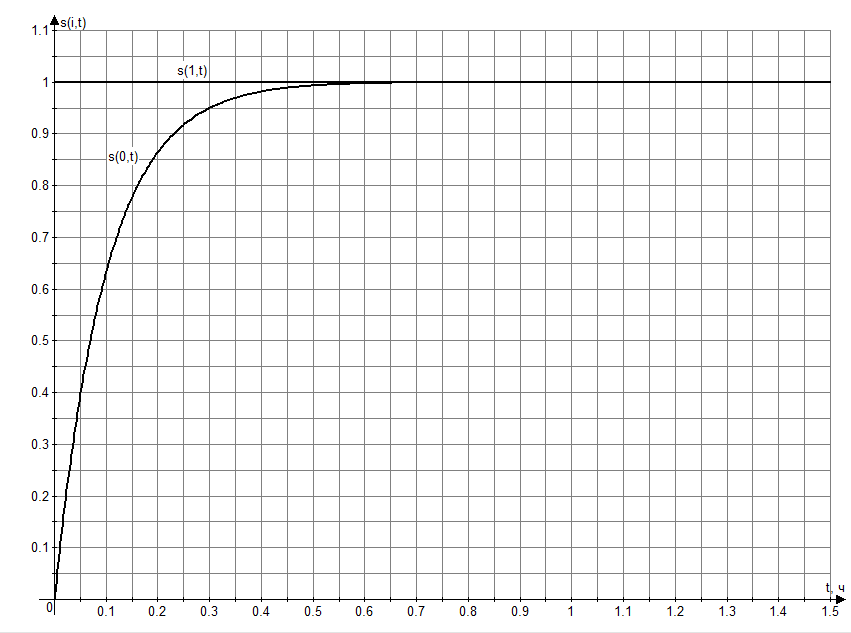
=0,9999999-0,9999999\*exp((-*t*)\*(10,000001));

*s*(1,*t*)=10/(10+1/106)-0.000001/(10+1/106)\*exp((-*t)*\*(10+1/106))=

=0,9999999-0.0000001\*exp((-*t*)\*(10.000001));

Рассчитаем значения функции и построим график:

|  |  |  |
| --- | --- | --- |
| **t, ч** | **S(1,t)** | **S(0,t)** |
| 0 | 1 | 0 |
| 0.01 | 1 | 0.0951626 |
| 0.02 | 1 | 0.181269 |
| 0.03 | 1 | 0.259182 |
| 0.04 | 1 | 0.32968 |
| 0.05 | 1 | 0.393469 |
| 0.06 | 1 | 0.451188 |
| 0.07 | 1 | 0.503415 |
| 0.08 | 1 | 0.550671 |
| 0.09 | 1 | 0.59343 |
| 0.1 | 1 | 0.63212 |
| 0.11 | 1 | 0.667129 |
| 0.12 | 1 | 0.698806 |
| 0.13 | 1 | 0.727468 |
| 0.14 | 1 | 0.753403 |
| 0.15 | 1 | 0.77687 |
| 0.16 | 1 | 0.798103 |
| 0.17 | 1 | 0.817316 |
| 0.18 | 1 | 0.834701 |
| 0.19 | 1 | 0.850431 |
| 0.2 | 1 | 0.864665 |
| 0.21 | 1 | 0.877544 |
| 0.22 | 1 | 0.889197 |
| 0.23 | 1 | 0.899741 |
| 0.24 | 1 | 0.909282 |
| 0.25 | 1 | 0.917915 |
| 0.26 | 1 | 0.925726 |
| 0.27 | 1 | 0.932794 |
| 0.28 | 1 | 0.93919 |
| 0.29 | 1 | 0.944977 |
| 0.3 | 1 | 0.950213 |
| 0.31 | 1 | 0.954951 |
| 0.32 | 1 | 0.959238 |
| 0.33 | 1 | 0.963117 |
| 0.34 | 1 | 0.966627 |
| 0.35 | 1 | 0.969803 |
| 0.36 | 1 | 0.972676 |
| 0.37 | 1 | 0.975276 |
| 0.38 | 1 | 0.977629 |
| 0.39 | 1 | 0.979758 |
| 0.4 | 1 | 0.981684 |
| 0.41 | 1 | 0.983427 |
| 0.42 | 1 | 0.985004 |
| 0.43 | 1 | 0.986431 |
| 0.44 | 1 | 0.987723 |
| 0.45 | 1 | 0.988891 |
| 0.46 | 1 | 0.989948 |
| 0.47 | 1 | 0.990905 |
| 0.48 | 1 | 0.99177 |
| 0.49 | 1 | 0.992553 |
| 0.5 | 1 | 0.993262 |
| 0.51 | 1 | 0.993903 |
| 0.52 | 1 | 0.994483 |
| 0.53 | 1 | 0.995008 |
| 0.54 | 1 | 0.995483 |
| 0.55 | 1 | 0.995913 |
| 0.56 | 1 | 0.996302 |
| 0.57 | 1 | 0.996654 |
| 0.58 | 1 | 0.996972 |
| 0.59 | 1 | 0.99726 |
| 0.6 | 1 | 0.997521 |
| 0.61 | 1 | 0.997757 |
| 0.62 | 1 | 0.99797 |
| 0.63 | 1 | 0.998164 |
| 0.64 | 1 | 0.998338 |
| 0.65 | 1 | 0.998496 |
| 0.66 | 1 | 0.99864 |
| 0.67 | 1 | 0.998769 |
| 0.68 | 1 | 0.998886 |
| 0.69 | 1 | 0.998992 |
| 0.7 | 1 | 0.999088 |
| 0.71 | 1 | 0.999175 |
| 0.72 | 1 | 0.999253 |
| 0.73 | 1 | 0.999324 |
| 0.74 | 1 | 0.999389 |
| 0.75 | 1 | 0.999447 |
| 0.76 | 1 | 0.999499 |
| 0.77 | 1 | 0.999547 |
| 0.78 | 1 | 0.99959 |
| 0.79 | 1 | 0.999629 |
| 0.8 | 1 | 0.999664 |
| 0.81 | 1 | 0.999696 |
| 0.82 | 1 | 0.999725 |
| 0.83 | 1 | 0.999751 |
| 0.839999 | 1 | 0.999775 |
| 0.849999 | 1 | 0.999796 |
| 0.859999 | 1 | 0.999816 |
| 0.869999 | 1 | 0.999833 |
| 0.879999 | 1 | 0.999849 |
| 0.889999 | 1 | 0.999863 |
| 0.899999 | 1 | 0.999876 |



Функция готовности может быть использована в качестве количественной характеристики стационарного режима работы ВС:

Список используемой литературы:

1. Хорошевский В.Г., Архитектура вычислительных систем. – М.: СибГУТИ, 2002.

2. Н. Дубова, Суперкомпьютеры. Открытые системы, 1995.