

Модели параллельного программирования

Цель лекции:

- **ознакомить магистрантов с основными моделями параллельного программирования, с вопросами эффективности использования процессоров**

Содержание лекции:

- передача сообщений,**
- параллелизм данных,**
- модель разделяемой памяти,**
- оценка степени параллелизма**

Передача сообщений:

**Особенность данной модели:
возникновение новых задач во
время выполнения программы
и возможность выполнения в
одном процессоре несколько
задач**

Передача сообщений:

**Во многих случаях
предварительно создаются
определенное количество
задач, и это количество
остаётся без изменения во
время выполнения программы**

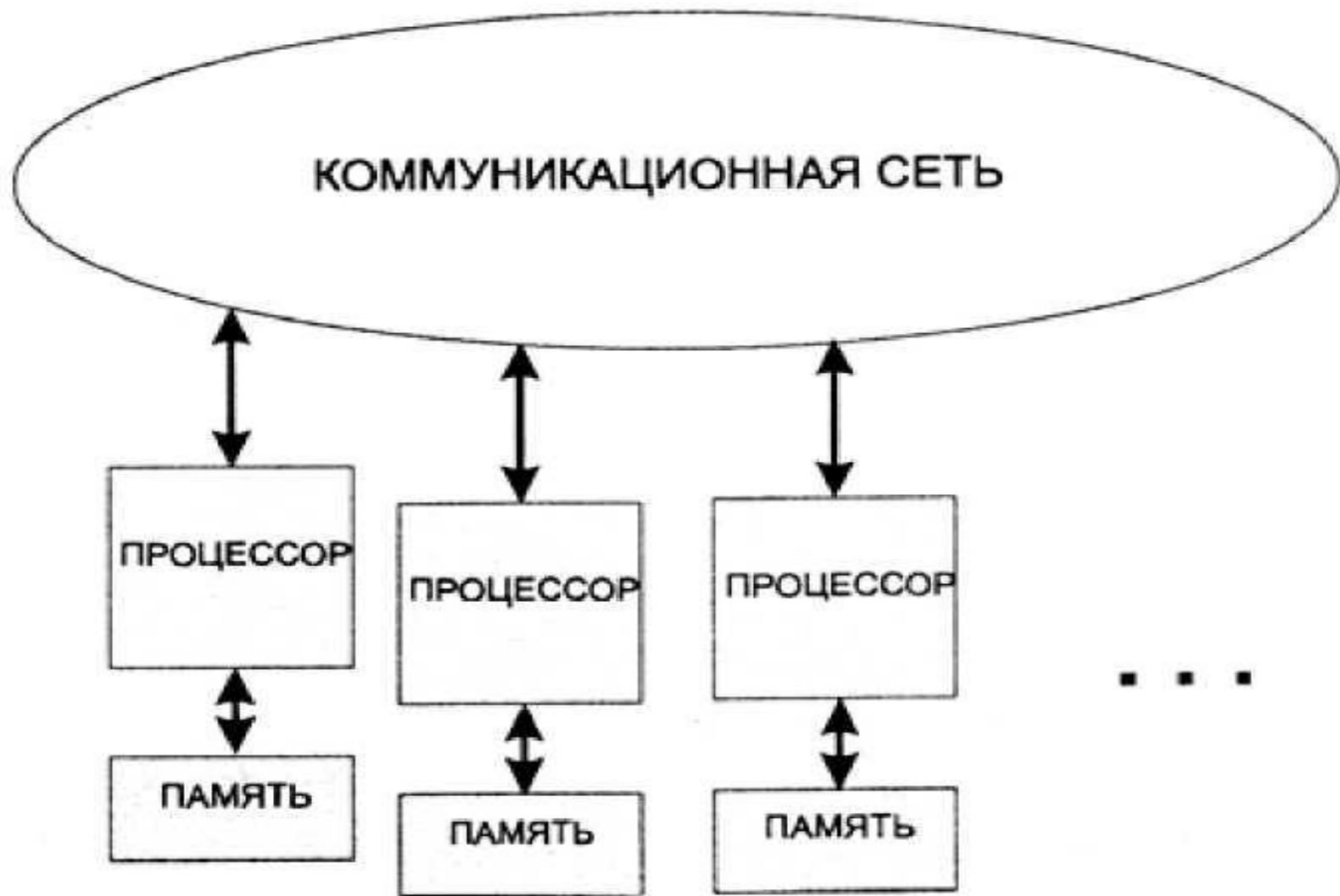
Передача сообщений:

**Single Program Multiple Data
SPMD - одна программа, массив данных). В этом случае каждая задача использует один и тот же программный код, а производит обработку разные данные.**

Передача сообщений:

Компьютер состоит из нескольких процессоров, каждый из которых снабжен своей собственной памятью. Компьютер, включенный в коммуникационную сеть, называется хост - машиной (*host*).

Передача сообщений: Схема модели



Передача сообщений:

- **Модель может быть реализована на системах:**
- **с распределенной памятью;**
- **с разделяемой памятью;**
- **на кластерах рабочих станций;**
- **на обычных однопроцессорных компьютерах.**

Параллелизм данных:

Основная идея этой модели программирования является применение одной операции сразу к нескольким элементам массива данных.

Параллелизм данных:

Основная идея этой модели программирования является применение одной операции сразу к нескольким элементам массива данных.

Фрагменты такого массива обрабатываются либо на векторном процессоре, либо на разных процессорах.

Параллелизм данных:

В данном случае векторизация или распараллеливание в основном выполняется во время трансляции программы.

Программист выполняет следующие функции:

- задавать транслятору опций векторной или параллельной оптимизации;**
- использовать специализированных языков параллельных вычислений**

Параллелизм данных:

Программист должен указать транслятору, как следует распределить данные между процессорами.

После трансляции генерируется SPMD-код, в него добавляются команды обмена данными в автоматическом режиме.

Модель разделяемой памяти:

Используется общая память; задачи имеют общее адресное пространство, каждая задача обращается к общей памяти и выполняет операции считывания /записи. Доступ к памяти управляется с помощью некоторых механизмов, в частности, как семафоры.

Модель разделяемой памяти:

Преимуществом данной модели является то, что здесь не требуется описывать обмен данными между задачами в явном виде. Такое положение упрощает работу программиста.

Структура алгоритма:

Любой алгоритм может иметь такую структуру, где встречаются фрагменты, которые могут быть выполнены одновременно на нескольких устройствах, а также фрагменты, которые выполняются последовательно на одном устройстве. Т.е. в программе могут быть как параллельная часть, так и последовательная часть.

Оценка степени параллелизма:

Время выполнения алгоритма на последовательном компьютере определяется следующей суммой :

$$t = t_s + t_p$$

где t_s - время выполнения последовательной части алгоритма;
 t_p - время выполнения параллельной части алгоритма.