

Проектирование коммуникаций и вычислений

Цель лекции:

- **с типами коммуникаций, используемых в параллельном программировании, рекомендациями по проектированию их, а также планированием вычислений.**

Содержание лекции:

**проектирование коммуникаций
и вычислений, основные типы
коммуникаций и способы их
организации**

Коммуникации делятся на следующие типы:

- **локальные коммуникации, когда каждая подзадача связана с небольшим набором других подзадач;**
- **глобальные коммуникации, когда каждая подзадача связана с большим числом других подзадач**

Коммуникации делятся на следующие типы:

- **структурированные коммуникации, когда каждая подзадача и связанные с ней другие подзадачи образуют регулярную структуру;**
- **неструктурированные коммуникации, когда подзадачи связаны произвольным графом;**

Коммуникации делятся на следующие типы:

- **статические коммуникации, когда схема коммуникаций не меняется с течением времени;**
- **динамические коммуникации, когда схема коммуникаций изменяется в процессе выполнения программы;**

Коммуникации делятся на следующие типы:

- **синхронные коммуникации, когда отправитель и получатель данных координируют обмен;**
- **асинхронные коммуникации, когда обмен данными не координируется.**

• Обмен сообщениями может быть реализован по-разному:

- с помощью потоков;**
- с помощью межпроцессорных коммуникаций (IPC-
Inter-Process Communication);**
- TCP-сокетов и других.**

- **Один из самых распространенных способов программирования коммуникаций является использование библиотек PVM (Parallel Virtual Machine) и MPI (Message Passing Interface), реализующих обмен сообщениями.**

Существуют другие способы организации коммуникаций:

- **RPC (Remote Procedure Control);**
- **CORBA (Common Object Request Broker Architecture);**
- **DCOM (Distributed Component Object Model).**

- **Способ RPC позволяет одному процессу вызывать процедуру из другого процесса, передавать ей параметры; если будет необходимость, то получать результаты выполнения**

- Способ CORBA определяет протокол взаимодействия между процессами, независимый от языка программирования и операционной системы. Для описания интерфейсов используется известный язык IDL (Interface Definition Language).

Укрупнение :

- **После декомпозиции и проектирования коммуникаций будет получен алгоритм, который не ориентирован на конкретную архитектуру вычислительной системы.**

Существуют другие способы организации коммуникаций:

- **RPC (Remote Procedure Control);**
- **CORBA (Common Object Request Broker Architecture);**
- **DCOM (Distributed Component Object Model).**

Основными требованиями к укрупнению подзадач являются следующие:

- снижение затрат на коммуникацию;**
- при укрупнении возможно дублирование вычисления или данные, но это не должно к потере производительности и масштабируемости программы;**

Основными требованиями к укрупнению подзадач являются следующие:

- трудоемкость результирующих задач должны быть примерно одинаковой;**
- должна сохраниться масштабируемость;**

Основными требованиями к укрупнению подзадач являются следующие:

- должна сохраниться возможность параллельного выполнения;**
- стоимость и трудоемкость разработки должны снижаться.**

Планирование вычислений.

- **Здесь должны быть определены процессоры, на которых будут обрабатываться подзадачи.**
- **Основным критерием эффективности в этой разработке является минимум времени выполнения программы.**

Планирование вычислений.

- **Ставится условие сбалансированной работы процессоров. Для этого существуют различные алгоритмы, применяемые в методах декомпозиции данных.**

Планирование вычислений.

- **Все эти методы предназначены для укрупнения мелкозернистых задач, чтобы в результате на один процессор приходилась одна крупноблочная задача.**

Программные средства высокопроизводительных вычислений.

- средства анализа и выявления параллелизма;**
- трансляторы;**
- операционные системы;.**

Программные средства высокопроизводительных вычислений.

- средства отладки;**
- средства оценки
производительности
программы и ее отдельных
частей.**

Программные средства высокопроизводительных вычислений.

- **Выполнение параллельной программы должна быть среда, включающая в себе операционную систему с поддержкой мультипроцессирования, многозадачности и многопоточности.**