**Введение в методы параллельного программирования**

* MPI: основные понятия и определения
* Введение в MPI
  + Инициализация и завершение MPI программ
  + Определение количества и ранга процессов
  + Прием и передача сообщений
  + Определение времени выполнения MPI программы
  + Коллективные операции передачи данных
* Пример: программа вычисления числа Pi
* Заключение

**Введение**

В вычислительных системах с распределенной памятью процессоры работают независимо друг от друга.

Для организации параллельных вычислений необходимо уметь:

* Распределять вычислительную нагрузку;
* Организовать информации

В MPI существует множества операций передачи данных:

* Обеспечиваются разные способы пересылки данных
* Реализованы практически все основные коммуникационные операции

Эти возможности являются наиболее сильной стороной MPI

MPI - это стандарт, которому должны удовлетворять средства организации передачи сообщении.

MPI- это программные средства, которые обеспечивают возможность передачи сообщениям стандарта MPI:

* программные средства должны быть организованы в виде библиотек программных модулей
* Должны быть доступны для наиболее широко используемых алгоритмических языков C и Fortran.

MPI позволяет существенно снизить остроту проблемы переносимости параллельных программ между разными комп системами.

MPI содействует повышению эффективности параллельных вычислений - практически для каждого типа вычислительных систем существуют реализации библиотек MPI.

MPI уменьшает сложность разработки параллельных программ. большая часть основных операций передачи данных предусматривается стандартом MPI. имеется большое количество библиотек параллельных методов, созданных с использованием MPI.

Под параллельной программой в рамках MPI понимается множество одновременно выполняемых процессов.

Процессы могут выполняться на разных процессорах. Вместе с этим на одном процессоре могут располагаться несколько процессов.

Каждый процесс параллельной программы порождается на основе копии одного и того же программного кода. Исходный программный код разрабатывается на алгоритмических языках С или Fortran с использованием библиотеки MPI. Количество процессов и число используемых процессоров определяется в момент запуска параллельной программы средствами среды исполнения MPI программ. Все процессы программы последовательно пронумерованы. Номер процесса именуется рангом процесса.

Под параллельной программой понимается такое множество одновременно выполняемых процессов. Процессы могут выполняться на разных процессорах вместе с этим, на одном процессоре могут располагаться несколько процессов.

В основу MPI положены четыре концепции:

* Тип операции передачи сообщений
* Тип данных, пересылаемых в сообщение
* Понятие коммутатора
* Понятие виртуальной топологии

Основу MPI составляют операции передачи сообщений.

* Среди предусмотренных в составе MPI функций различаются
* Парные операции между двумя процессами
* Коллективные коммуникационные действия для одновременного взаимодействия нескольких процессов.

Коммуникатор в MPI - специально создаваемый служебный объект, объединяющий в своем составе группу процессов и ряд дополнительных параметров:

* Парные операции передачи данных выполняется для процессов, принадлежащих одному и тому же коммуникатору.
* Коллективные операции применяются одновременно для всех процессов коммуникатора.
* Указание используемого коммутатора является обязательным для операций передачи данных в MPI.

В ходе вычислений могут создаваться новые и удаляться существующие коммуникаторы. Один и тот же процесс может принадлежать разным коммуникаторам. Все имеющиеся в параллельной программе процессы входят в состав создаваемого по умолчанию коммуникатор с идентификатором. При необходимости передачи данных между процессами из разных групп необходимо создавать глобальный коммуникатор.

**Понятие коммуникаторов**

* В ходе вычислений могут создаваться новые и удаляться существующие коммуникаторы
* Один и тот же процесс может принадлежать разным коммуникаторам
* Все имеющиеся в параллельной программе процессы входят в состав создаваемого по умолчанию коммуникатора с идентификатором MPI\_COMM\_WORLD.
* При необходимости передачи данных между процессами из разных групп необходимо создавать глобальный коммуникатор