

Серафима Вручтель

Data Scientist  **циан**



@vruchtel_s



vruchtel.serafima@yandex.ru

Параллельные и распределённые вычисления

Параллельные и распределённые вычисления

Параллельные

Распределённые

Параллельные и распределённые вычисления

Параллельные

- многократное ускорение
- высокопроизводительные машины

с точки зрения разработчика:
**набор взаимодействующих
процессов**

Распределённые

Параллельные и распределённые вычисления

Параллельные

- многократное ускорение
- высокопроизводительные машины

с точки зрения разработчика:
набор взаимодействующих процессов

Распределённые

- большие объёмы данных
- обычные машины
- отказоустойчивость

с точки зрения разработчика:
одно распределённое вычислительное устройство

MPI

Message Passing Interface

Коммуникационный протокол, независимый от языка программирования, который используется для программирования параллельных компьютеров (кластеры и суперкомпьютеры).

1991 - начало разработки MPI

ноябрь 1993 - проект стандарта MPI представлен на конференции "Supercomputing'93"

июнь 1994 - выпуск версии MPI 1.0

...

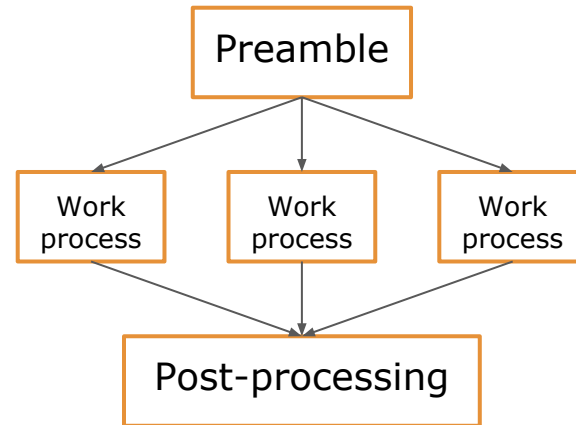
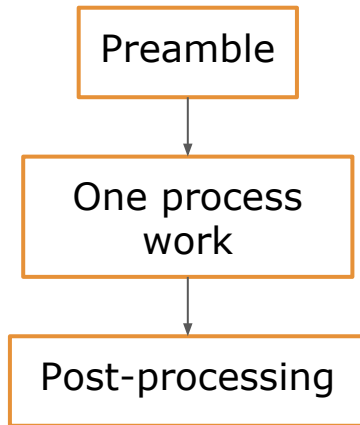
2012 - MPI 3.0

Интерфейс стал де-факто стандартом для связи процессов при работе **параллельной программы на системе с распределённой памятью**.

Терминология

Процесс - компьютерная программа в стадии своего выполнения.

Компьютерная программа является пассивным набором инструкций.
Процесс является активным выполнением этих инструкций.



Hello World на MPI

MPI Call	Parameters	Purpose
MPI_Init	(int *argc, char ***argv) Передача всем процессам аргументов main	Инициализация параллельной части приложения
MPI_Comm_size	(MPI_Comm comm, int *size) Искомое значение оказывается в переменной size	Определение общего числа запущенных параллельных процессов
MPI_Comm_rank	(MPI_Comm comm, int *rank) Искомое значение оказывается в переменной rank	Определение номера запущенного процесса (от 0 до size - 1)
MPI_Finalize	(void)	Завершение параллельной части приложения

Кластер МФТИ

Зайти на кластер: `ssh USER@calc.cod.phystech.edu`

Перед компиляцией: `module add mpi/openmpi4-x86_64`

Компиляция

C: `mpicc filename.c`

C++: `mpic++ filename.cpp`

По умолчанию
исполняемый файл
имеет имя a.out

Запуск

`mpirun -np num_of_processes ./a.out` num_of_processes -
это число

Send - Receive

int MPI_Send(void* buf, int count, MPI_Datatype datatype, int dest, int msgtag, MPI_Comm comm)

- *buf* - адрес начала буфера послыки сообщения
- *count* - число передаваемых элементов в сообщении
- *datatype* - тип передаваемых элементов
- *dest* - номер процесса-получателя
- *msgtag* - идентификатор сообщения
- *comm* - идентификатор коммуникатора

int MPI_Recv(void* buf, int count, MPI_Datatype datatype, int source, int msgtag, MPI_Comm comm, MPI_Status *status)

- *OUT buf* - адрес начала буфера приема сообщения
- *count* - максимальное число элементов в принимаемом сообщении
- *datatype* - тип элементов принимаемого сообщения
- *source* - номер процесса-отправителя
- *msgtag* - идентификатор принимаемого сообщения
- *comm* - идентификатор коммуникатора
- *OUT status* - параметры принятого сообщения

Время

$$S = \frac{T_1}{T_p}$$

Ускорение

T_1

Время работы алгоритма на одном процессоре (ядре)

T_p

Время работы алгоритма на **p** процессорах (ядрах)

double MPI_Wtime(void)

Функция возвращает астрономическое время в секундах (вещественное число), прошедшее с некоторого момента в прошлом. Гарантируется, что этот момент не будет изменен за время существования процесса.

Запуск задач на кластере

Ресурс-менеджер slurm

Запуск скрипта на кластере

sbatch --comment="Your task comment" run.sh

Результат:

> Submitted batch job 871175

В файле **slurm-871175.out** будет результат запуска

Запуск задач на кластере

Запуск задачи на 8 процессах

```
sbatch -n 8 --comment="Your task comment" run.sh
```

Аналогично, но по 4 процесса на каждой ноде

```
sbatch -n 8 --ntasks-per-node=4 --comment="Your task comment" run.sh
```

Конфиг sbatch

> run_sbatch_config.sh

```
#!/bin/bash
#
#SBATCH --ntasks=8
#SBATCH --cpus-per-task=1
#SBATCH --partition=RT
#SBATCH --job-name=example
#SBATCH --comment="Run mpi from config"
#SBATCH --output=out.txt
#SBATCH --error=error.txt
mpiexec ./a.out
```

Запуск: **sbatch run_sbatch_config.sh**