



ПАРАЛЛЕЛЬНЫЕ АЛГОРИТМЫ

Занятие 1. Технология MPI



Составитель: Герасимов А.С.

Учебный кластер МФТИ

head.vdi.mipt.ru

remote.vdi.mipt.ru:52960

ssh login@head.vdi.mipt.ru

- Узлы: 1 головной (head) и 7 вычислительных
- Узлы идентичны: 4 ядра, 15 ГБ ОЗУ
- Система очередей – Torque/PBS

Пример PBS-задачи

job.sh

```
#!/bin/bash
```

```
#PBS -l walltime=00:10:00,nodes=7:ppn=1
```

```
#PBS -N job_name
```

```
#PBS -q batch
```

```
uname -n
```

Запуск задачи

```
qsub job.sh
```

Выход задачи:

- `<job_name>.o<ID>` – выход stdout
- `<job_name>.e<ID>` – выход stderr

Ограничения:

- 5 заданий / пользователя
- 10 минут выполнения
- 1 ГБ памяти

Просмотр текущих задач в очереди

qstat

```
[kolya@head mpi]$ qstat
```

Job id	Name	User	Time Use	S	Queue
-----	-----	-----	-----	-	-----
25.localhost	my_job	kolya	0	R	batch
26.localhost	my_job	kolya	0	R	batch
27.localhost	my_job	kolya	0	R	batch
28.localhost	my_job	kolya	0	R	batch
29.localhost	my_job	kolya	0	R	batch

Удаление задачи

qdel <ID>

MPI (Message Passing Interface)

- Библиотека функций, предназначенная для поддержки работы параллельных процессов.
- Базовый механизм связи между процессами – передача и приём сообщений.
- Ориентирован на системы с распределенной памятью
- Состав сообщений:
 - *отправитель — ранг (номер в группе) отправителя;*
 - *получатель — ранг получателя;*
 - *признак(и);*
 - *коммуникатор — код группы процессов.*
- Блокирующие / неблокирующие передачи

Общие процедуры MPI. Инициализация

```
int MPI_Init (int* argc, char*** argv)
```



- MPI_SUCCESS
- код ошибки



Аргументы функции main()

```
int MPI_Finalize (void)
```

Общие процедуры MPI. Инициализация

Основа программы

```
#include "mpi.h"
```

```
int main(int argc, char** argv) {
```

```
    MPI_Init(&argc, &argv);
```

```
    ...
```

```
    MPI_Finalize();
```

```
}
```


Общие процедуры MPI. Размер группы

```
int MPI_Comm_size  
    (MPI_Comm comm, int* size)
```




- Коммуникатор группы
- MPI_COMM_WORLD



(OUT) Размер группы

Общие процедуры MPI. Ранг процесса

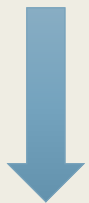
```
int MPI_Comm_rank  
    (MPI_Comm comm, int* rank)
```

- 
- Коммуникатор группы
 - MPI_COMM_WORLD

(OUT) Номер процесса в группе [0; size-1]

Общие процедуры MPI. Подсчет времени

```
double MPI_Wtime (void)
```



Некоторое время в секундах

Общие процедуры MPI. Оценка ускорения

Закон Амдала

$$a \leq \frac{1}{(1 - p) + \frac{p}{n}}$$

a – оценка ускорения

p – распараллеливаемая часть программы (доля общего времени выполнения)

n – количество процессов

Компиляция программы

```
mpicc superhot.c -o hot
```

Запуск программы

```
mpirun -np <thread_num> ./hot
```

job.sh

```
#!/bin/bash
```

```
#PBS -l walltime=00:01:00,nodes=1:ppn=3
```

```
#PBS -N my_job
```

```
#PBS -q batch
```

```
cd $PBS_O_WORKDIR
```

```
mpirun --hostfile $PBS_NODEFILE -np 3 ./hot
```

Общие процедуры MPI. Задача 1

- Составить и запустить программу «Hello, world!»
- Вывести размер своего коммутатора и своего процесса :D