

Параллельное программирование

Ведущий практической части курса – Фатеев Алексей

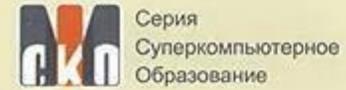
Сергеевич E-mail: fateev.as@mipt.ru

Организационные вопросы. Критерии оценки.

- **Обязательные задачи (без крайнего срока)**
- 0. Вводные задачи по изучаемым технологиям
- 1. Лабораторная работа по первой части семестра
- 2. Задача по второй части семестра
- 3. Контрольные работы (1 или 2, по желанию лектора)
- Оценка усредняется и ограничивается отл(8).
- *Дополнительные задачи (имеют крайний срок)*
- Баллы суммируются с оценкой, полученной в обязательной части.
- Итоговая оценка округляется (8.5 -> 9, 8.49 -> 8).

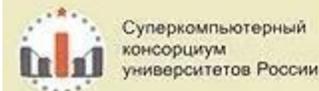
Теоретическая часть - лекции

- Численные методы, алгоритмы и программы.
- Введение в
- распараллеливание (Карпов
- Владимир Ефимович, Лобанов Алексей Иванович), 2014 год, Физматкнига,
- ISBN 978-5-89155-234-0
-



В. Е. Карпов
А. И. Лобанов

Численные методы,
алгоритмы и программы.
Введение в
распараллеливание



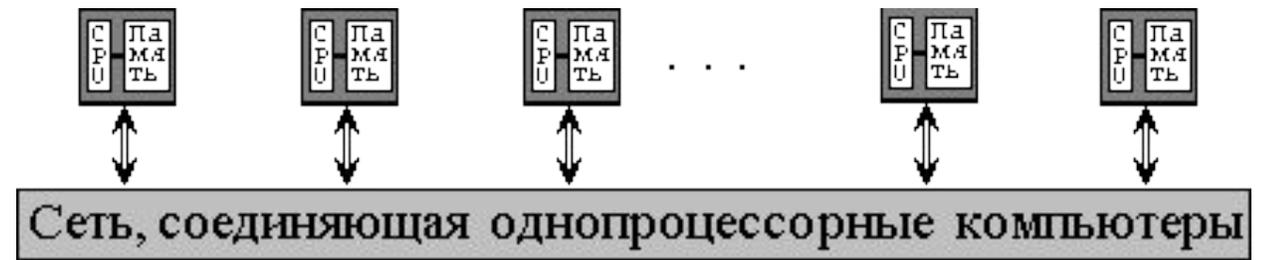
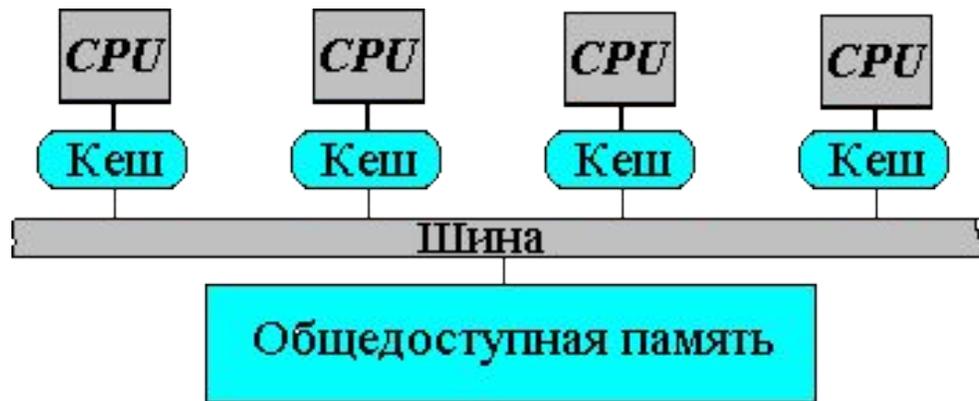
НОЦ СКТ МФТИ



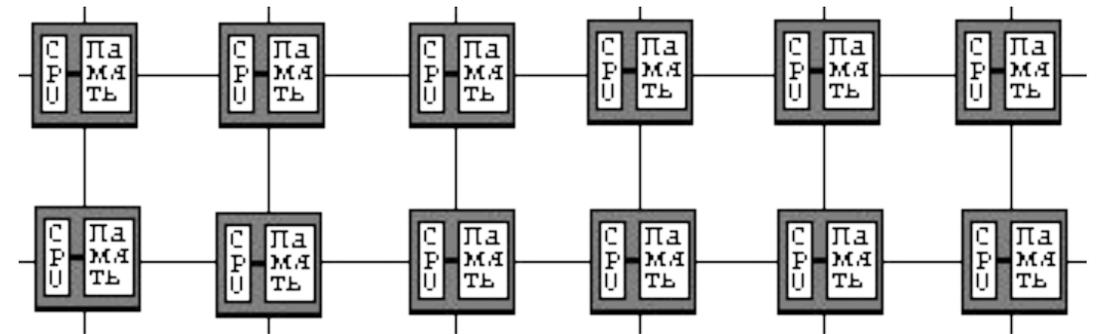
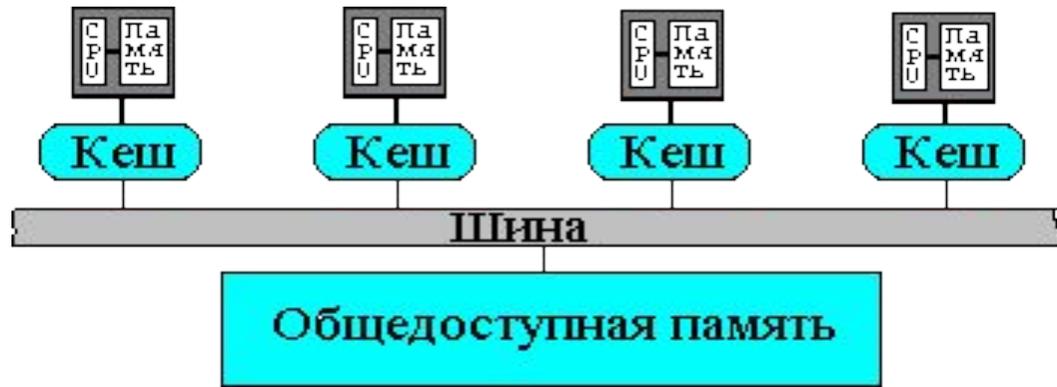
Организационные вопросы. Доступ к учебному кластеру.

- - Доступен персональный логин каждому студенту
- - Возможен удалённый доступ по протоколу ssh с широкого круга устройств
- - Логин предназначен исключительно для выполнения задач по курсу «Параллельное программирование»
- - Возможно выполнение учебных задач на собственном оборудовании, при наличии там минимум 4 физических процессорных ядер. Ответственность за настройку и работоспособность оборудования в таком случае студент берёт на себя

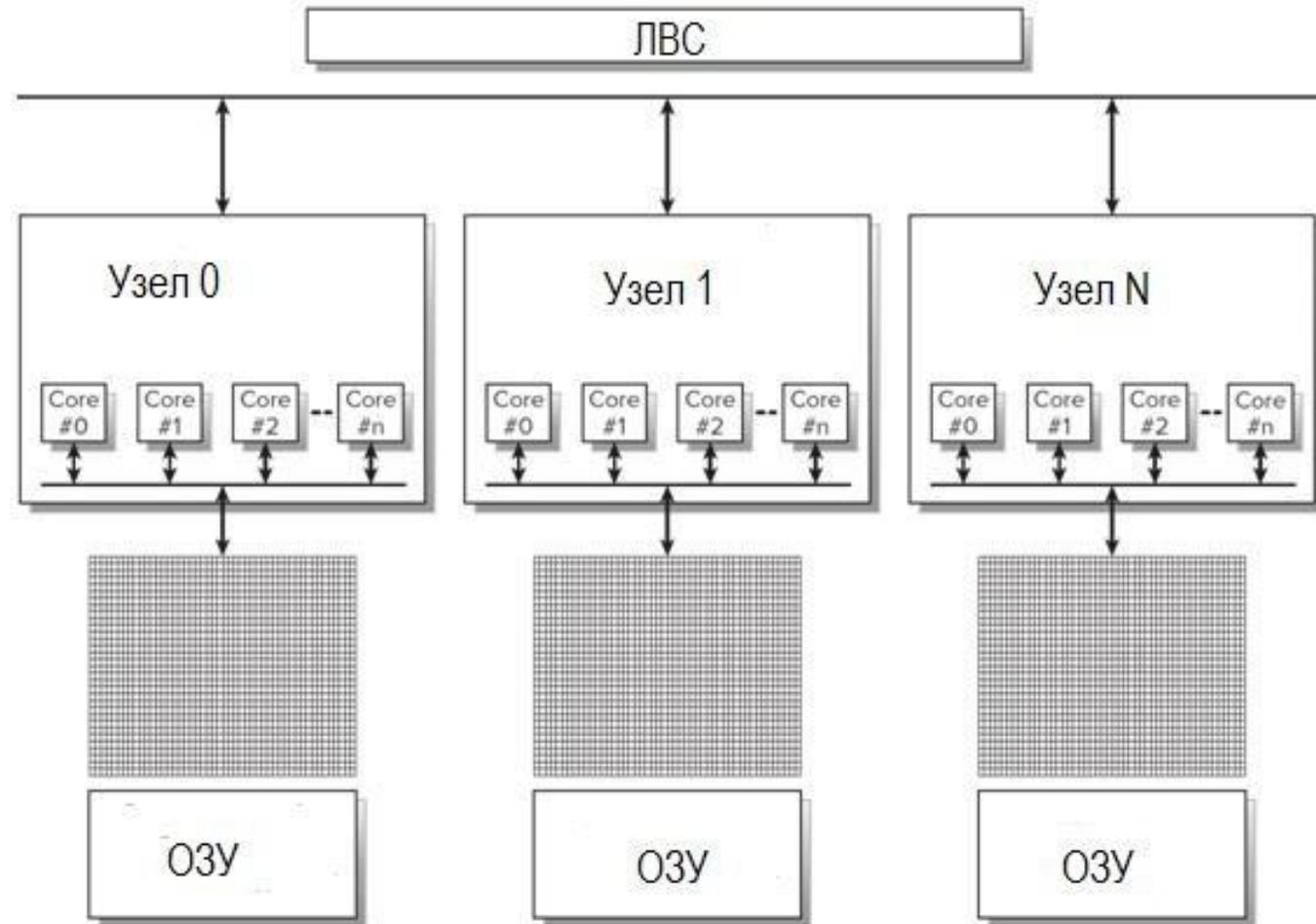
Классические системы с распределенной и общей памятью



Примеры архитектурной «экзотики»



Классическая архитектура суперкомпьютера кластерного типа (без GPU)



Высокопроизводительные вычисления (HPC, High Performance Computing)

- ❑ В настоящее время – с использованием параллельных вычислений (массово-параллельные системы)
- ❑ Суперкомпьютер – есть ряд определений данного термина, один из – «компьютер, с мощностью, намного (т.е. более, чем на пару порядков) превосходящей вычислительную мощность типовой рабочей станции данного времени»
- ❑ Рейтинг и архитектуру самых мощных суперкомпьютеров можно изучить на www.top500.org
- ❑ На данный момент (2023) часть суперкомпьютеров – классическая кластерная архитектура, часть – с добавлением GPU
- ❑ Суперкомпьютер кластерного типа – подвид систем с распределёнными вычислениями
- ❑ В распределённых вычислениях в общем смысле этого слова есть ряд качеств, которые не всегда присущи суперкомпьютерам. Например, это качества, связанные с работой в условиях высокой вероятности отказа узлов, несимметричности архитектуры, не всегда предсказуемой пропускной способности сети, объединяющей узлы, динамически меняющейся в процессе работы конфигурации и т.д.
- ❑ Среди top500 на начало 2023 – около 47% - промышленное применение, 24% - исследовательские, около 17% - академические, остальное – правительственные, секретное применение и т.д. (распределение по количеству систем)

MPI - Message Passing Interface

- дословно – интерфейс передачи сообщений – программный интерфейс (API) для передачи сообщений, однако стандарт описывает спецификацию, у которой есть целый ряд реализаций (самые известные – MPICH, Open MPI, Intel MPI, Mvarich и т.д.)
- <https://www.mpi-forum.org/>
- эволюционирует с начала 1990-х годов, стал де-факто стандартом в области HPC (высокопроизводительных вычислений, в том числе суперкомпьютеров)
- основные языки – C/C++ и Fortran
- есть менее популярные варианты для .NET, JAVA, Julia, MATLAB, Ocaml, Python, R, есть попытки внедрить в Boost C++ Libraries

Как это работает?

- - Подключаем библиотеки MPI (`#include <mpi.h>`)
- (описание типов, функций, стандартный набор)
- Начало программы
- Инициализация MPI библиотек (`MPI_Init(&argc,&argv);`)
-
- Работа программы
- Обмены данными (`MPI_Send`, `MPI_Recv` и т.д.)
-
- Завершение работы MPI (`MPI_Finalize();`)

Завершение работы программы

Распределенная память - особенности

У каждого процессора своё адресное пространство
Для того, чтобы данные попали от одного процессора к другому,
программисту следует явно произвести передачу и приём данных

+ Отличная масштабируемость

- Обычно большие задержки
- Определённые сложности при создании программ
- Менее эффективное использование ресурсов

Коммуникаторы и группы

- Для определения, какие процессы могут взаимодействовать друг с другом, существуют специальные объекты – группы и коммуникаторы
- Функции MPI чаще всего требуют в качестве параметра коммуникатор – отдельный контекст обмена процессов некоторой группы
- Изначально есть MPI_COMM_WORLD, MPI_COMM_NULL, MPI_COMM_SELF
- Возможно создавать новые коммуникаторы

Нумерация процессов

- У процесса внутри коммутатора есть уникальный номер (rank, task id) – целое число, нумерация последовательная начиная с 0
- Также у коммутатора есть размер – общее число входящих в него исполнителей (процессов)
- Используется в качестве идентификации адресата при обмене данными (передаче и приёме сообщений)
- Часто в алгоритмах выделяется нулевой исполнитель (так как в общем случае заведомо неизвестно число исполнителей)

Hello World!

```
#include <mpi.h>
#include <stdio.h>

int main(int argc, char *argv[]){
    int commsize, my_rank;
    MPI_Init(&argc, &argv);
    MPI_Comm_size(MPI_COMM_WORLD, &commsize);
    MPI_Comm_rank(MPI_COMM_WORLD, &my_rank);
    printf ("Communicator size= %d My rank= %d\n", commsize, my_rank);
    MPI_Finalize();
}
```

Компиляция и запуск программы

- Надстройка над стандартным компилятором
`mpicc -o hello hello.c`
Вызывает стандартный компилятор с большим числом флагов
- Для запуска используется отдельная утилита `mpirun`
`mpirun -np 4 ./hello`
- На суперкомпьютерах обычно используется в пакетном режиме совместно с системой очередей
- Программы для суперкомпьютеров обычно не используют интерактивный ввод, данные передаются в виде аргументов или конфигурационных файлов и т.д.

Ссылка на материалы занятий

