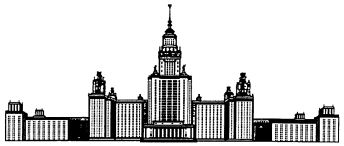


Московский Государственный Университет им. М.В.Ломоносова
Факультет Вычислительной Математики и Кибернетики

Анализ эффективности масштабируемых подходов к решению задач с преобладанием ввода-вывода

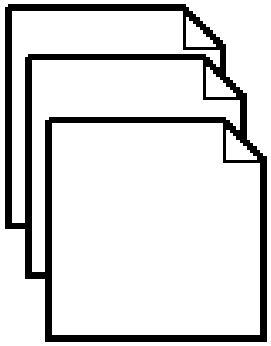
Андреев Д.Ю. ВЦ РАН, МГУ ВМК
Джосан О. В. МГУ ВМК

PaVT2011, Пермь – 29.03.2011

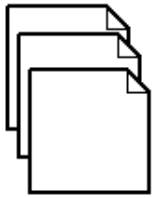
Обращая внимание на ввод-вывод

- **+ Хранилище данных с возможностью параллельного доступа к файлам (GPFS, PVFS, Lustre, и т. д.)**
- **- Коммуникационная среда вычислительной системы неоднорода (производительность коммуникационной среды).**
- **+ Обработка больших объёмов данных.**
- **- Данные превышают размер оперативной памяти, загрузка данных из внешних источников данных.**
- **+ Множество параллельно работающих устройств.**
- **- Количество вычислений на единицу данных имеет значение.**

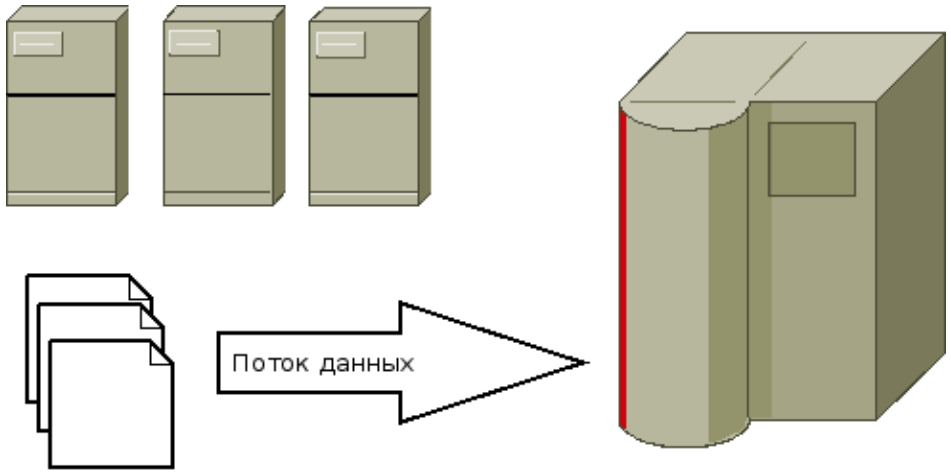
Задачи с потоком данных



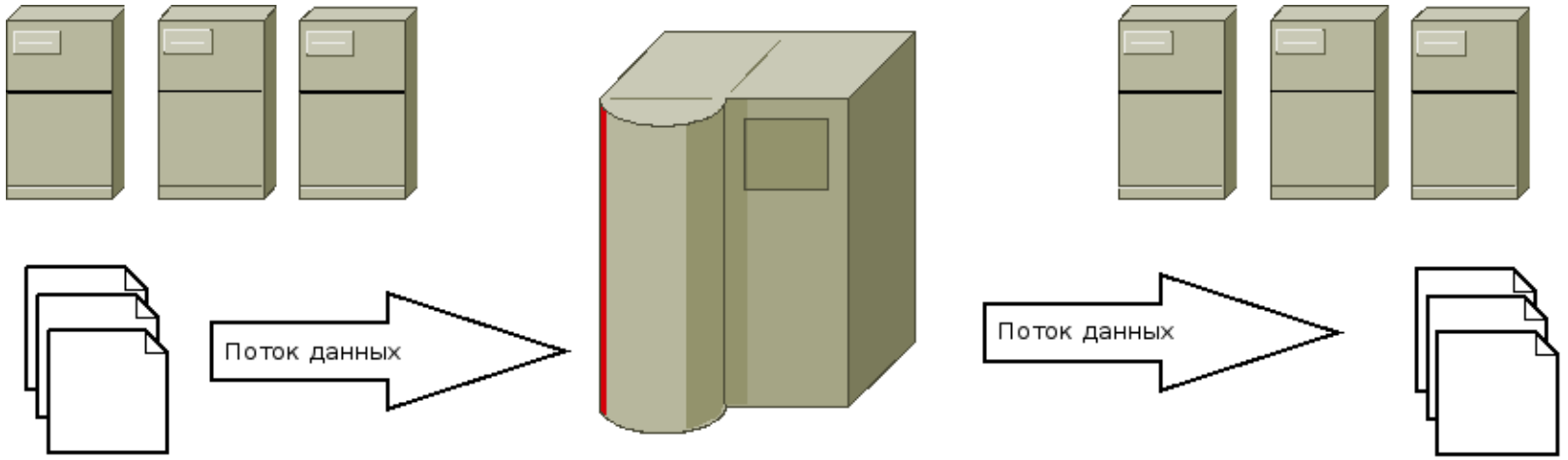
Задачи с потоком данных



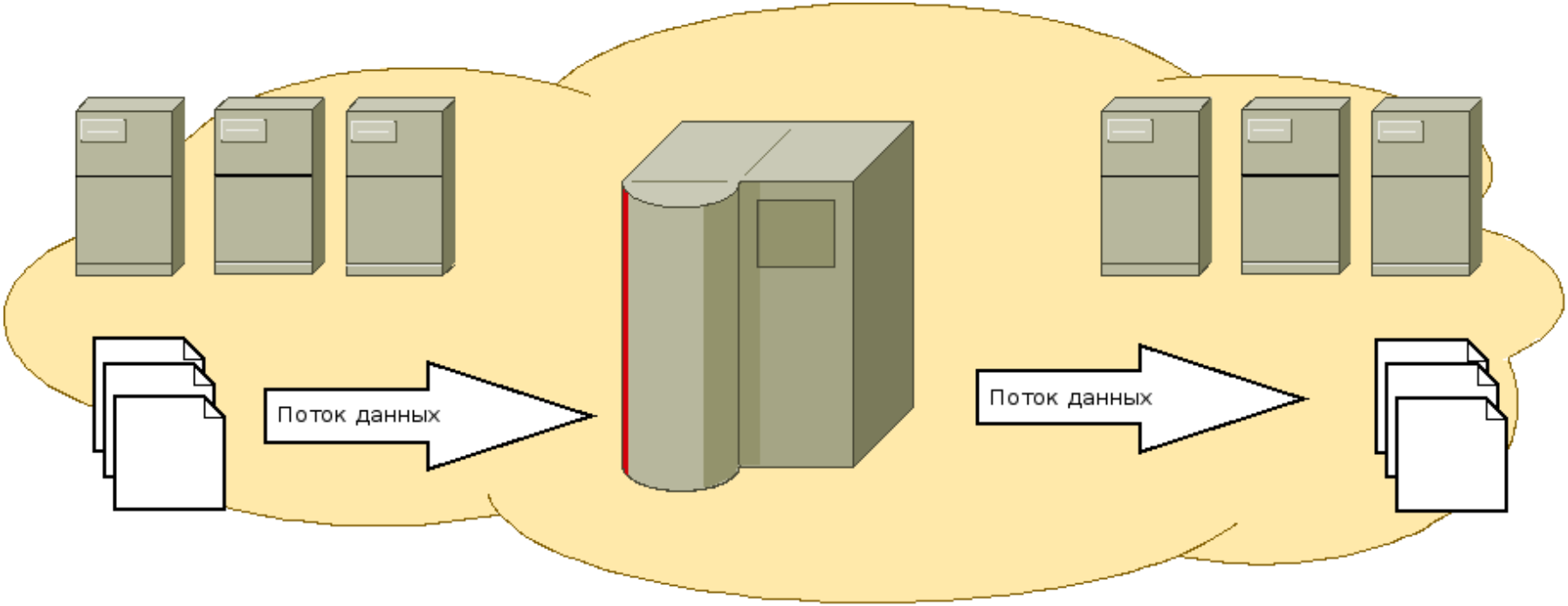
Задачи с потоком данных



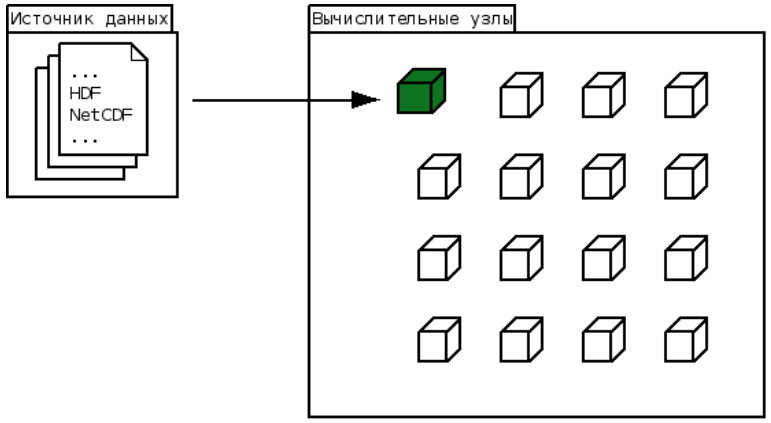
Задачи с потоком данных



Задачи с потоком данных

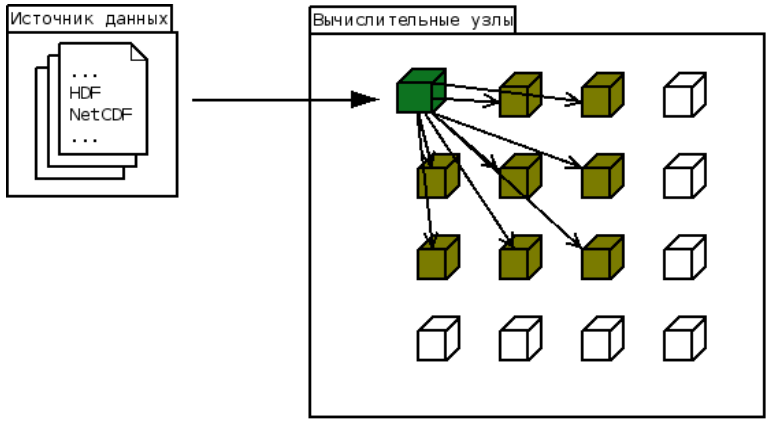


Загрузка данных в оперативную память



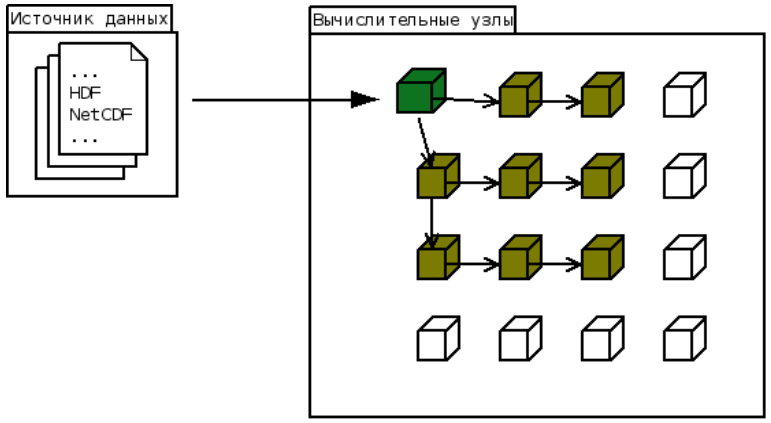
- **Нагрузка на один вычислительный узел и один коммуникационный канал;**
- **Ожидание данных множеством процессоров;**
- **Активное использование коммуникационной среды передачи данных между процессорами во время перераспределить данных;**

Загрузка данных в оперативную память



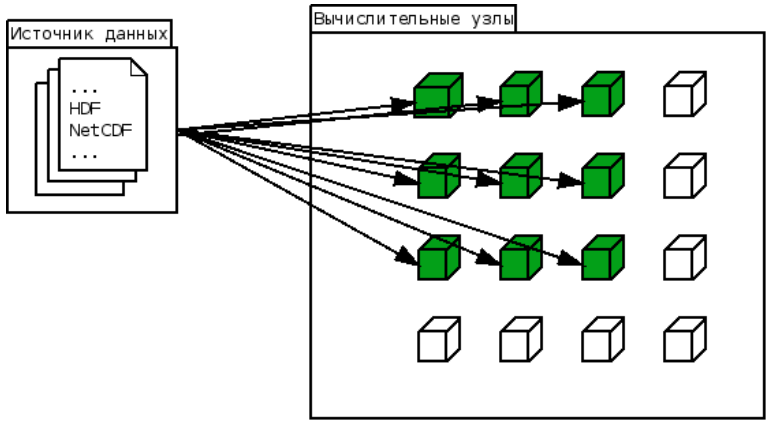
- **Нагрузка на один вычислительный узел и один коммуникационный канал;**
- **Ожидание данных множеством процессоров;**
- **Активное использование коммуникационной среды передачи данных между процессорами во время перераспределить данных;**

Загрузка данных в оперативную память



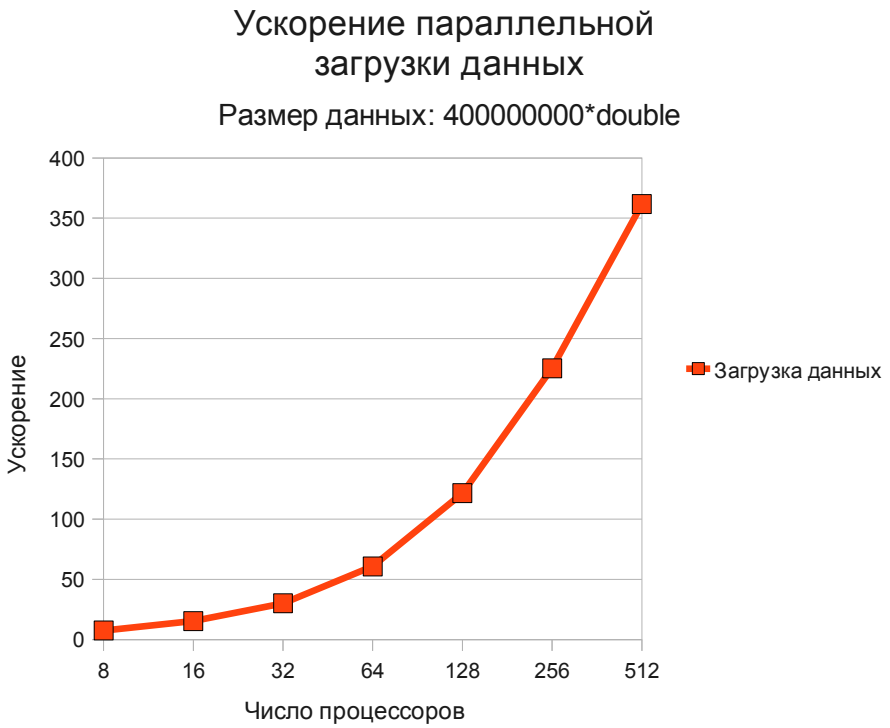
- **Нагрузка на один вычислительный узел и один коммуникационный канал;**
- **Ожидание данных множеством процессоров;**
- **Активное использование коммуникационной среды передачи данных между процессорами во время перераспределить данных;**

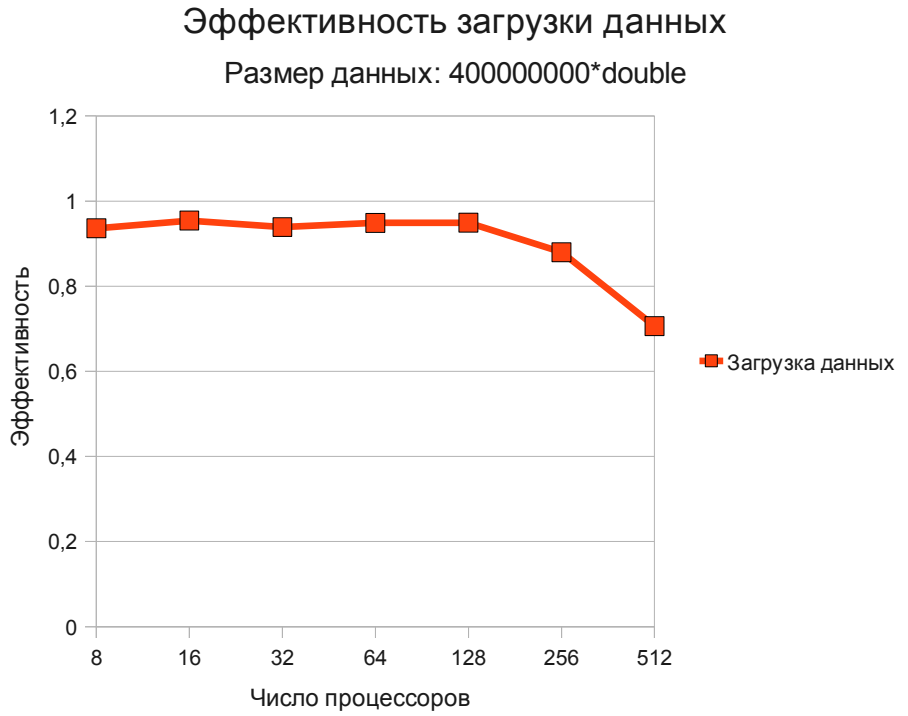
Загрузка данных в оперативную память



- **Высокая нагрузка на систему хранения данных и её коммуникационные каналы.**
- **Все процессоры заняты загрузкой данных;**
- **Возможен случайный доступ к данным во внешней памяти.**

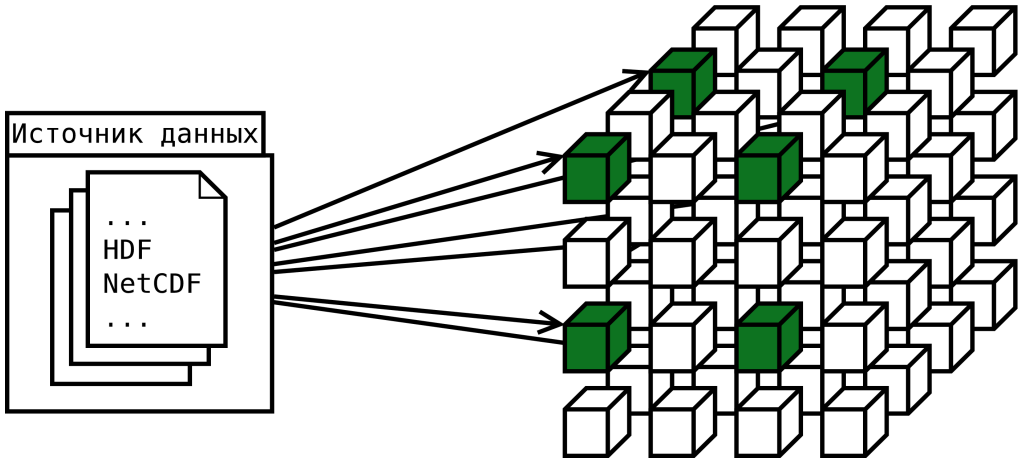
Загрузка данных в оперативную память





Ломоносов (T500)

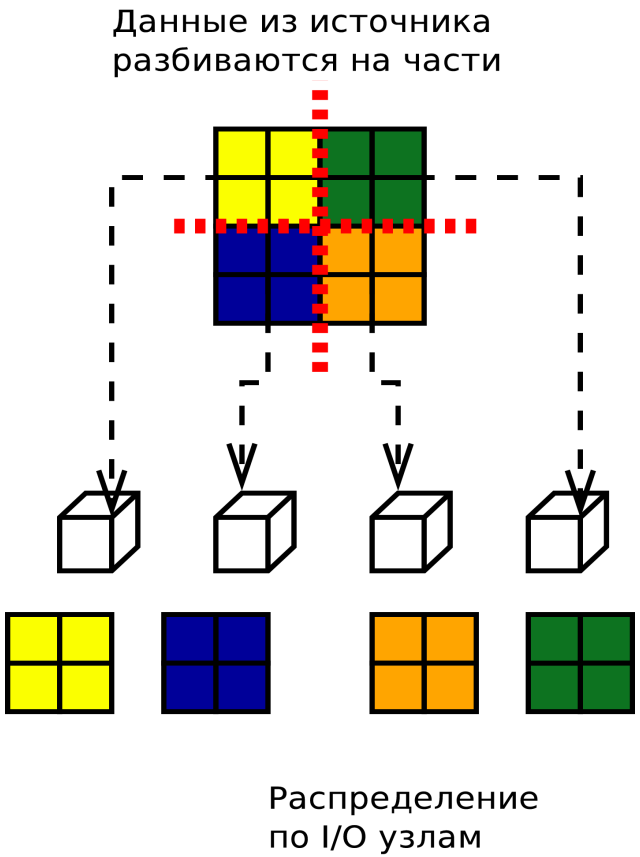
Загрузка/выгрузка данных



Вычислительные узлы

Узлы, выделенные для I/O операций

Загрузка/выгрузка данных

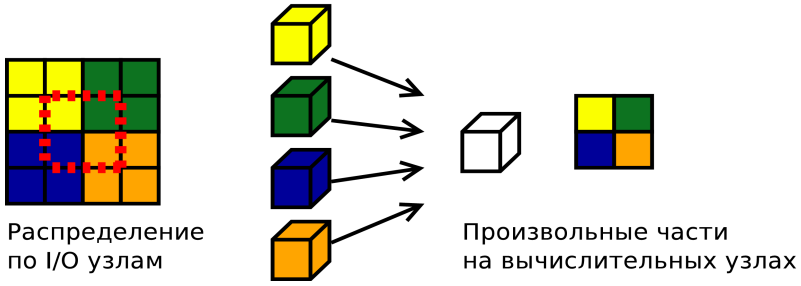


- **Параллельная загрузка данных на узлы отвечающие за ввод-вывод**

Взаимодействие вычислительных узлов и I/O узлов

- **I/O узлам позволяется работать независимо от остальных:**
  - **I/O узлы могут продолжать загрузку во вычислений на остальных узла**
  - **Вычислительные узлы не знают о распределении данных по I/O узлам, но знают размер запрашиваемых данных**

- **Многомерные массивы**
- **MPI_Type_create_subarray(...)**
- **MPI_Put(...) для распределения**
- **MPI_Get(...) для сбора данных**



Тестирование подхода

- **Синтетическая тест-функция - задаётся количество сложений, вычитаний, умножений и делений на единицу данных.**
- **Количество узлов ввода-вывода.**
- **Размер данных.**

1. **Данные загружаются на узлы I/O**
2. **Данные перераспределяются по вычислительным узлам**
3. **Пока работает синтетическая тест-функция узлы I/O продолжают загрузку и перераспределение данных.**

Тестирование подхода

- **Наглядный пример:**
  - **1 сложение, 1 умножение, 4000000000 double ~ скалярное произведение векторов.**
  - **Из-за малого числа данных эффективность резко падает при 512 процессорах.**
  - **На 256 процессорах время выполнения счёта ~= 0.059**
  - **+ время загрузки данных на узлы ~= 0.014**

  - **При выделении 64 I/O процессора, вычисления ~= 0.067**
  - **Параллельно время загрузки и распределения ~= 0.054**

  - **При повторных операциях — экономия ~ 8% времени.**

Схема библиотеки

ParimprC_Init(); //инициализация библиотеки

ParimprC_ParimprCreateComm(...);
//создание MPI-коммуникатора библиотеки, выбор I/O узлов

ParimprC_DataLoad_[Type](...); //загрузка данных из источника

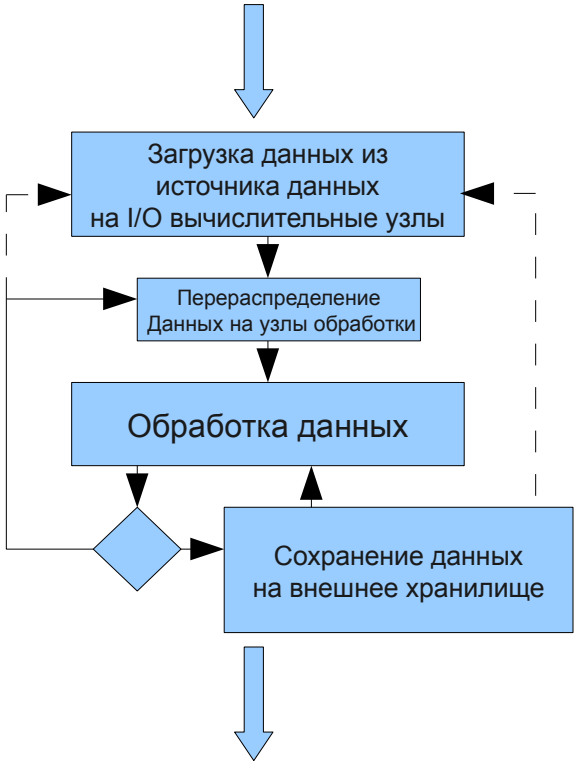
ParimprC_LayoutCreate(...);
//создание MPI-коммуникатора для обработки данных

ParimprC_DataPrepare(...);
//распределение данных между MPI-нитями обработчиками данных

<вызов обработчика данных>

Схема алгоритма

- **Выбираются вычислительные узлы, отвечающие за ввод-вывод;**
- **Загружаются данные из внешней памяти на выделенные узлы;**
- **Распределить данные на узлы обработки данных;**
- **Обработка данных;**
- **Выгрузка данных с узлов обработки.**



Предусмотренные возможности библиотеки

- **Произвольные источники данных:**
  - **Форматы файлов**
  - **Алгоритмы архивации/шифрования**
  - **Генераторы данных**
- **Алгоритмы выбора I/O узлов**
- **Алгоритмы автоматизированного выбора разбиения**
- **Работа в рамках выделенного коммуникатора (при условии глобальной инициализации библиотеки)**
- **Организация конвейера обработчиков**

Спасибо за внимание.

Андреев Дмитрий Юрьевич: **andreevd@cs.msu.su**