GRID-сегмент КарНЦ РАН и основы разработки программ для GRID-платформы BOINC

Докладчик: Наталия Никитина

Грид

- Сеть вычислительных ресурсов

- Цель: объединить вычислительные мощности и возможности хранения данных географически распределенных компьютеров

Типы Грид-систем

- Объединение специально выделенных компьютеров (суперкомпьютеров)
- Объединение персональных компьютеров во время их простоя
  - Добровольные вычисления
  - Настольный Грид

Типы Грид-систем

- Добровольные вычисления:
  - Анонимное участие
  - Участники не несут ответственности
  - Участники контролируют выполнение заданий
  - «Избыточные вычисления»
  - Участники доверяют организаторам
  - Участники контролируют приложение
  - Большие и дешевые вычислительные мощности
  - Начисление «кредитов» за работу
  - Большое количество выполняющихся заданий в единицу времени

Типы Грид-систем

- Настольные Грид:
  - Неанонимное участие
  - Организаторы доверяют участникам
  - Ресурсы доступны постоянно
  - Участники не контролируют приложение
  - Ограничены локальной сетью организации
  - Большая скорость выполнения заданий

BOINC

(Berkeley Open Infrastructure for Network Computing)

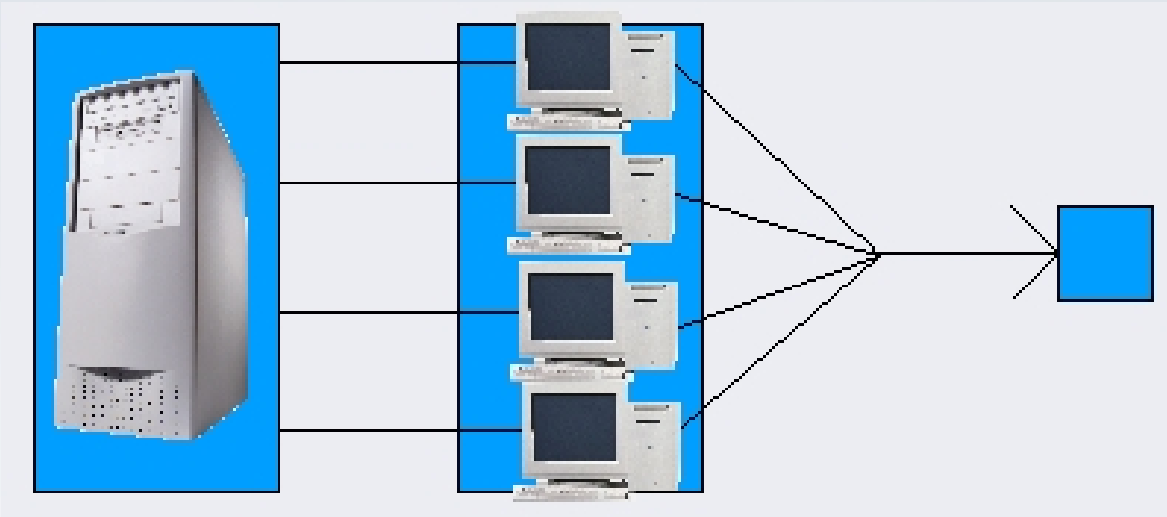
- SETI@home (1995 г.) — распределенный анализ сигналов радиотелескопа на компьютерах добровольцев (на пике нагрузки >600 TFlops)
- >500,000 активных компьютеров
- >3,900 TFlops

BOINC

(Berkeley Open Infrastructure for Network Computing)

- **Архитектура «клиент-сервер»**
- **Открытый исходный код** (GNU LGPL)
- **Многоплатформенность**
- **Гибкая прикладная оболочка**
- **Проверка результатов**
- **Безопасность**
- **Web-интерфейс**

Приложения для BOINC



- Анализ больших объемов данных, который можно выполнять по частям
- Многократный запуск программы на разных наборах входных данных
- Поиск перебором

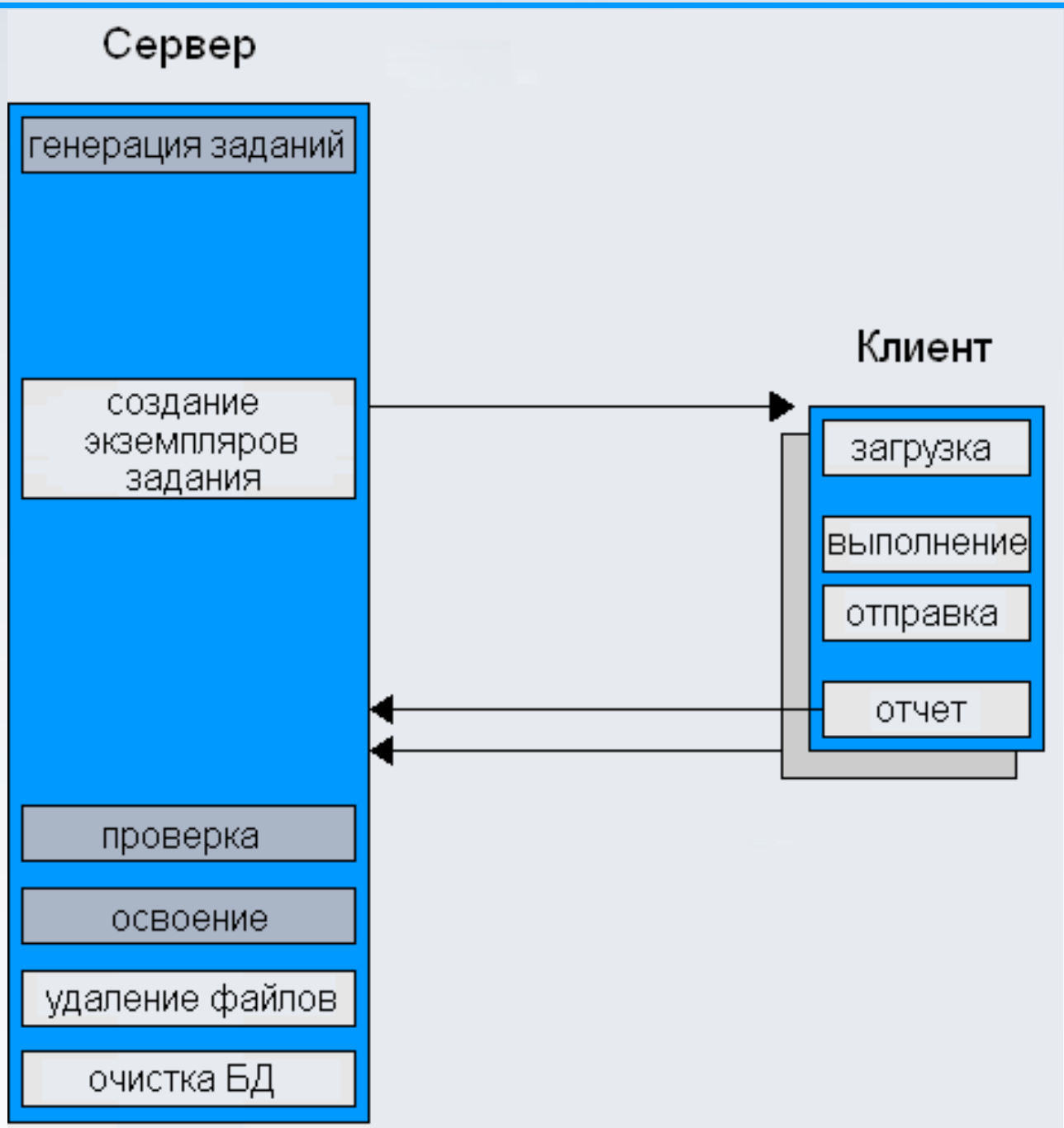
Основные понятия

- **Проект = Приложения + База данных + Веб-сайт + Сервер**
- **Приложение**
- **Платформа**
- **Версия приложения**
- **Рабочее задание**
  - Оценка требований к ресурсам (объем ОП, дискового пространства, число операций с плавающей точкой и др.)
  - Параметры выполнения (количество экземпляров задания, минимальное количество результатов и др.)
- **Результат**
- **Учетная запись**

Архитектура BOINC

- Ядро клиента
- Научное приложение
- BOINC-менеджер
- Программа экранной заставки

Жизненный цикл рабочего задания



Группы функций API

- **Базовые функции**
  - инициализация и завершение обращений к API, работа с файлами, создание контрольных точек, определение критических секций и др.
- **Управление последовательностью подзаданий**
- **Работа с графикой для рисования заставки**
- **Обмен сообщениями с сервером в процессе выполнения задания**
- **Отправка на сервер промежуточных результатов**
- **Получение и изменение состояния приложения**
- **Получение диагностических сообщений**

Основные функции API

- **Инициализация и завершение обращений к API**
  - *int boinc_init()*
  - *int boinc_init_parallel()*
  - *int boinc_finish (int status)*
- **Работа с файлами**
  - *boinc_resolve_filename(char *logical_name, char *physical_name, int len)*
  - *boinc_fopen(char* physical_name, char* mode)*
- **Создание «контрольных точек»**
  - *int boinc_time_to_checkpoint()*
  - *void boinc_checkpoint_completed()*
- **Определение текущей стадии выполнения рабочего задания**
  - *void boinc_fraction_done(double fraction_done)*

Взаимодействие приложения с клиентом

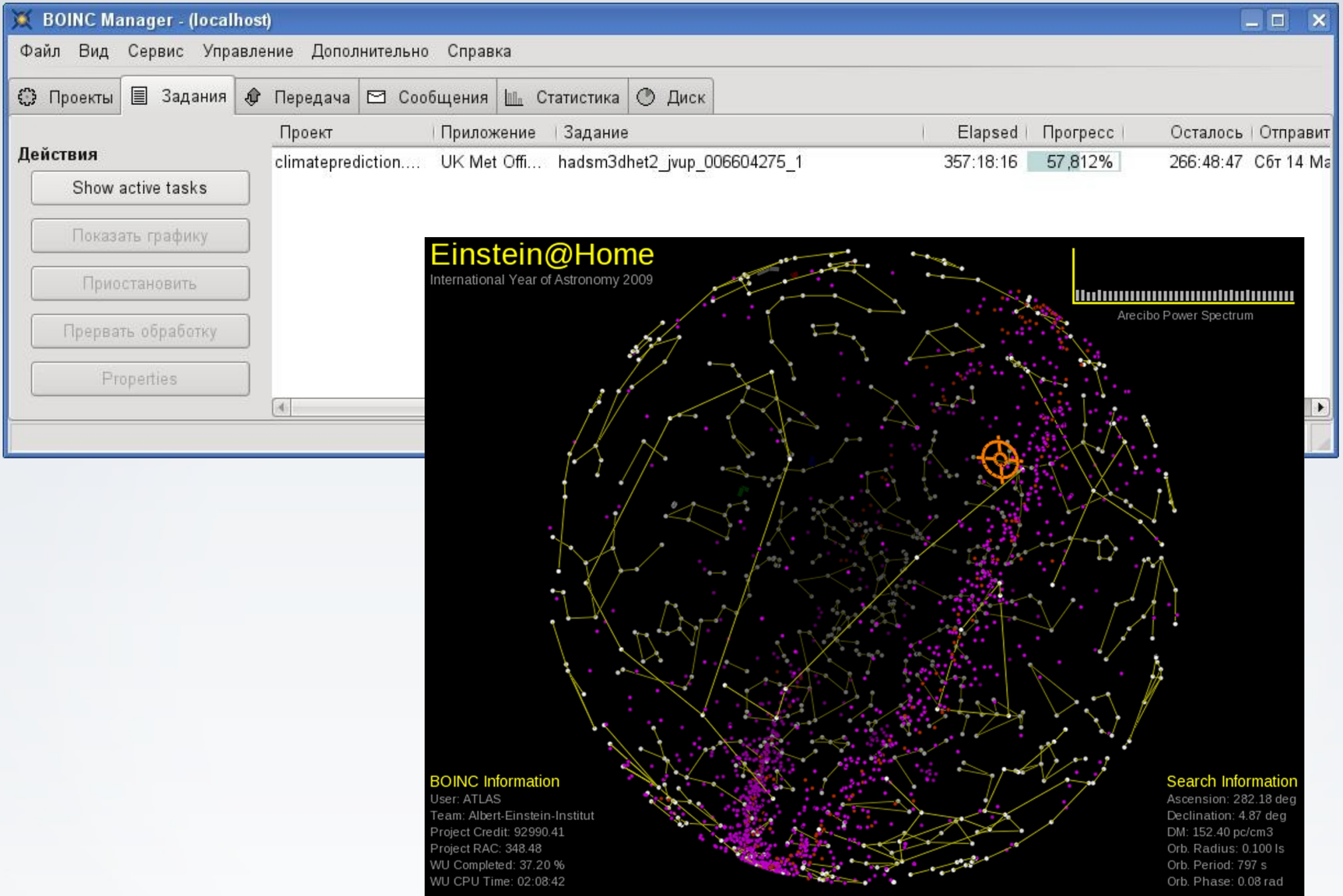
Клиент

- Команды управления выполнением рабочего задания
- Логические имена файлов

- Текущая стадия выполнения рабочего задания
- Использованное процессорное время
- Контрольные точки
- Успешное завершение рабочего задания

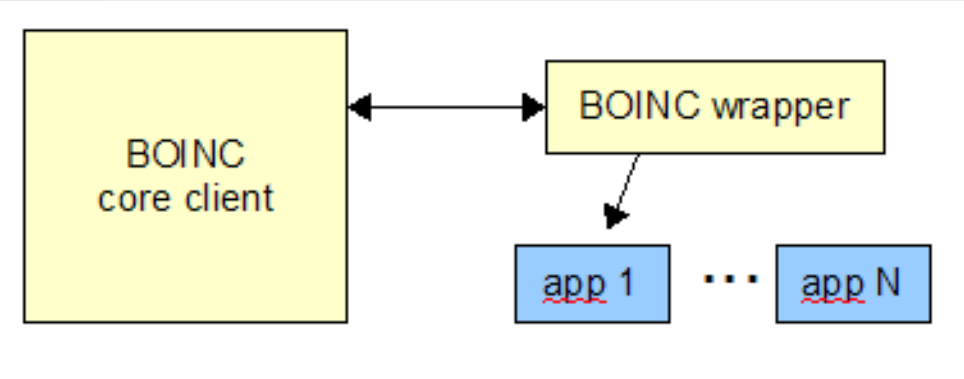
Приложение

Взаимодействие приложения с клиентом



Типы приложений

- Программа—«обертка» (Boinc Wrapper)




```
<job_desc>
     <task>
     <application>worker</application>
     [ <stdin_filename>stdin_file</stdin_filename> ]
     [ <stdout_filename>stdout_file</stdout_filename> ]
     [ <stderr_filename>stderr_file</stderr_filename> ]
     [ <command_line>--foo bar</command_line> ]
     [ <weight>X</weight> ]
     [ <checkpoint_filename>filename</checkpoint_filename> ]
     [ <fraction_done_filename>filename</fraction_done_filename> ]
   </task>
     [ ... ]
</job_desc>
```

Типы приложений

Модификация исходного кода

```
  retval = boinc_init();

  boinc_resolve_filename(INPUT_FILENAME, input_path,
                         sizeof(input_path));

  infile = boinc_fopen(input_path, "r");

  file_size(input_path, fsize);

  boinc_resolve_filename(OUTPUT_FILENAME,
                         output_path, sizeof(output_path));

  boinc_resolve_filename(CHECKPOINT_FILE,
chkpt_path,
                         sizeof(chkpt_path));

 state = boinc_fopen(chkpt_path, "r");
  if (state) {
     n = fscanf(state, "%d", &nchars);
     fclose(state);
  }
  if (state && n==1) {
     fseek(infile, nchars, SEEK_SET);
     boinc_truncate(output_path, nchars);
     retval = out.open(output_path, "ab");
  }

else {
     retval = out.open(output_path, "wb");
}

  for (int i=0; ; i++) {
     c = fgetc(infile);

     if (c == EOF) break;
     c = toupper(c);
     out._putchar(c);
     nchars++;

      if (boinc_time_to_checkpoint()) {
        retval = do_checkpoint(out, nchars);
        boinc_checkpoint_completed();
      }

     fd = nchars/fsize;
     boinc_fraction_done(fd); // fd*100%
  }

  boinc_fraction_done(1);      // 100 %
  boinc_finish(0);
```

Типы приложений

- Генерация новых рабочих заданий на основе полученных результатов
  - выполнение пакета рабочих заданий
  - освоение результатов пакета
  - генерация следующего пакета заданий

Поддержка приложений

- Модификация исходного кода
  - C, C++, Fortran и др. языки
- Использование программы-«обертки»
  - Приложение, состоящее из нескольких исполняемых файлов и/или скриптов командной оболочки
  - Java, Python, Lisp, Perl и др. языки
- Приложения, использующие GPU
  - CUDA, OpenCL
- Многопоточные приложения
  - OpenMP
  - MPI (в разработке)
  - Titanium, Cilk и др. языки
  - Многопоточные библиотеки

BOINC-Грид КарНЦ РАН

- Запуск готовых приложений
- Консультации по модификации приложений
- Разработка приложений