

# РАЗРАБОТКА КОМПОНЕНТОВ НРС --- COMMUNITY CLOUD

Разработчики:

Беляев Николай НГТУ ФПМИ

Вайцель Сергей НГТУ ФПМИ

Волков Александр НГУ ФИТ

Куратор:

Городничев

Максим Александрович

ИВМиМГ СО РАН

# Введение

- HPC Community Cloud – инструментарий взаимодействия пользователя с объединенными HPC-системами
- Основное предназначение комплекса – упрощение и ускорение процессов создания параллельных программ и запуска их на кластере (кластерах)

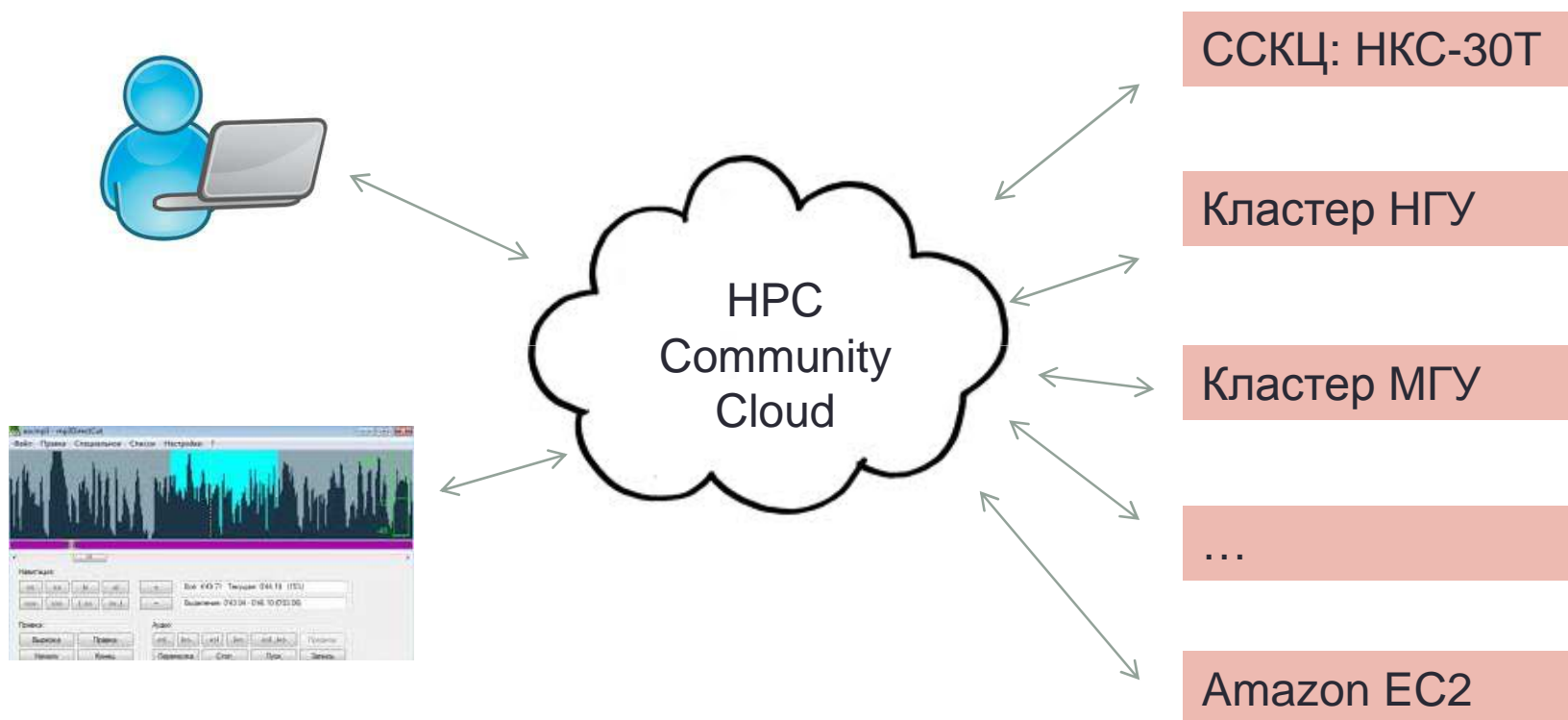
# Цель

- Предоставить простые способы запуска параллельных программ для исследователей и студентов
- Повысить уровень взаимодействия пользователя с высокопроизводительными системами

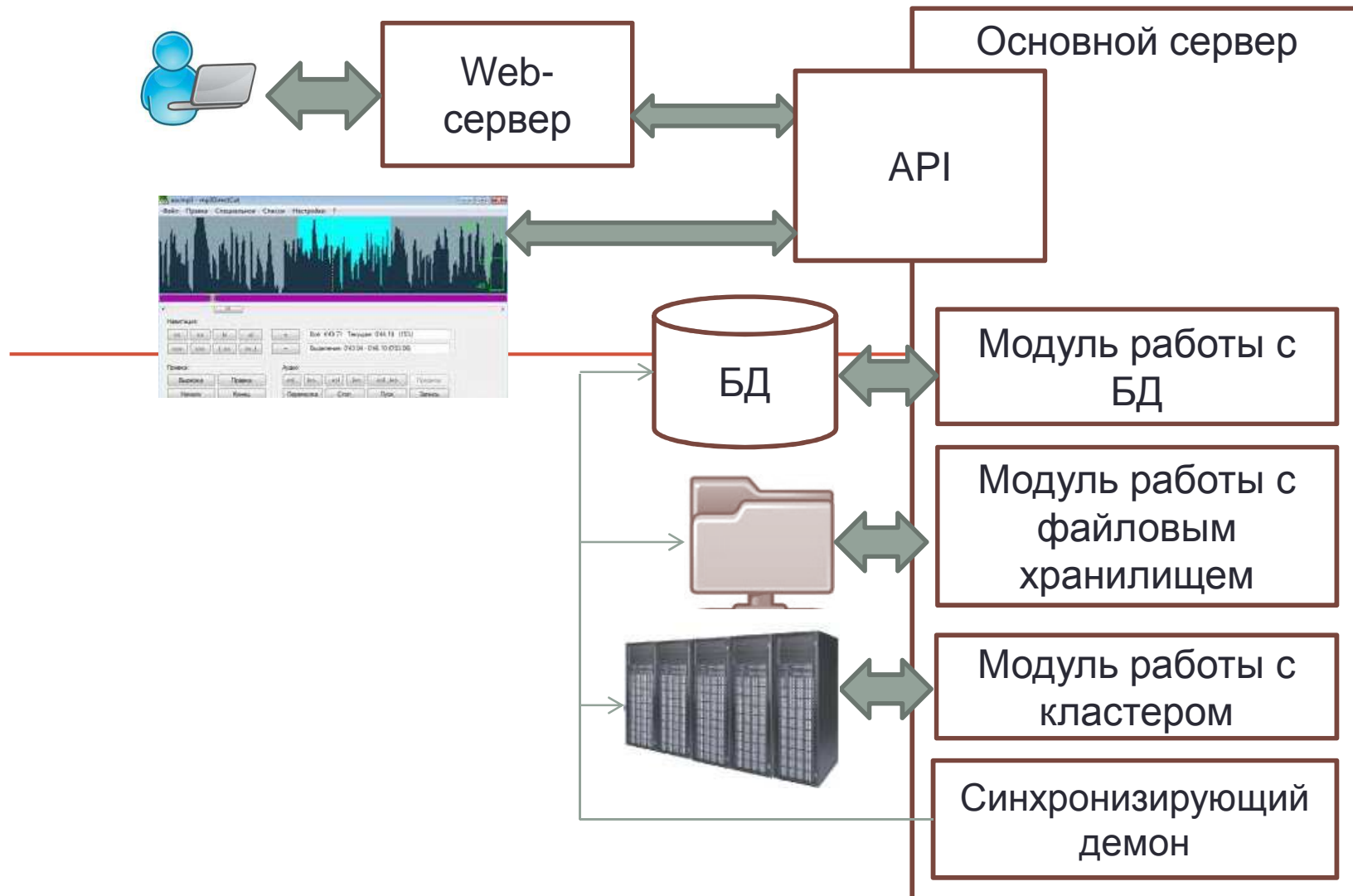
# Задачи

- Разработать инструментарий взаимодействия пользователей и внешних программных систем с высокопроизводительным оборудованием
- Обеспечить быстрое создание параллельных программ для определенных классов задач на основе фреймворков, встраиваемых в HPCSS
- Разработать визуализацию результатов вычислений
- Обеспечить отказоустойчивость при выполнении параллельных программ на суперкомпьютере

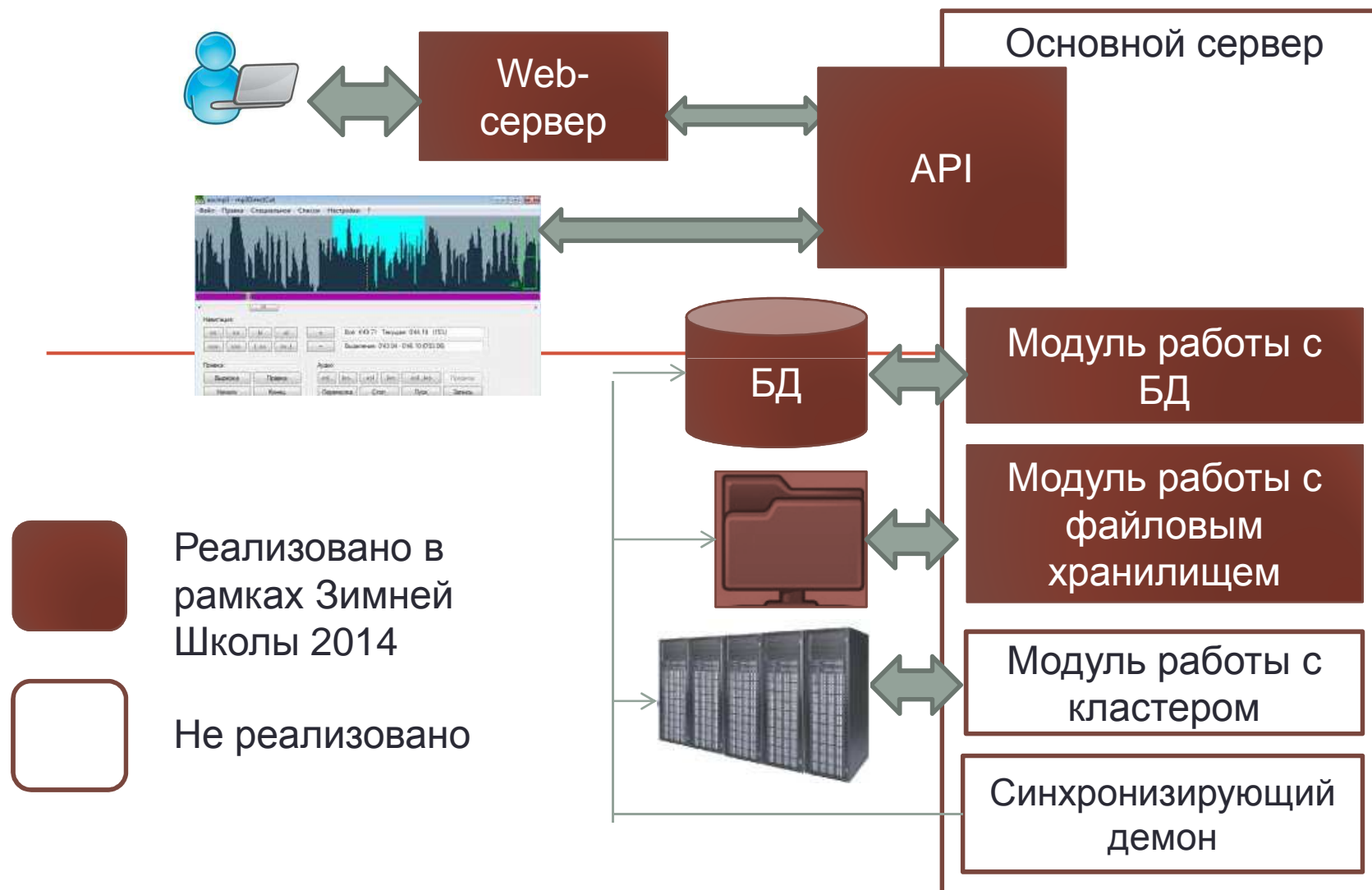
# Идея



# Архитектура системы HPC Community Cloud



# Архитектура системы HPC Community Cloud



# API для внешних систем

- *Запуск экспериментов*
- post\_newexperiment (JSONconfiguration)
- get\_experimentstate (experimentName, userName)
- get\_experimentresults (experimentName, userName)
  
- *Работа с файловой системой*
- get\_filelist (userName, projectName)
- get\_file (userName, projectName, type, fileName)
- post\_modifyfile (userName, projectName, type, fileName)
- post\_newfile (userName, projectName, type, fileName)
- get\_deletefile (userName, projectName, type, fileName)



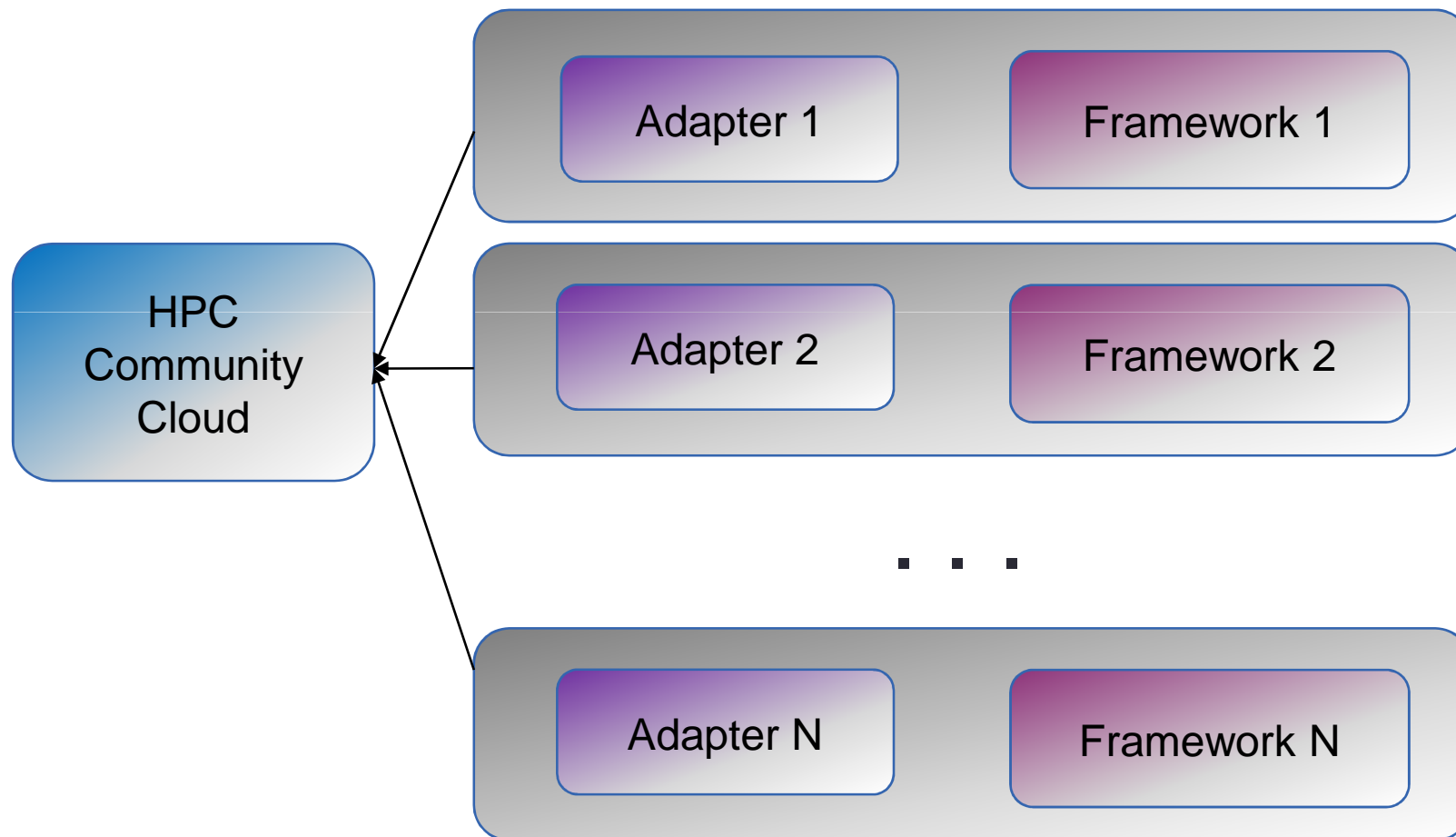
# Online-среда разработчика

The screenshot displays a web browser window titled "HPC Community Cloud". The address bar shows "hpc.sccc.ru". The page header includes "User" and "Setting" menus. On the left side, there is a sidebar with a "JobName" input field and several menu items: "SOURCE", "RESOURCE", "RESULT", "StdErr.txt", "StdOut.txt", "error.log", "MAKE", "CONFIGURATION", and "UPLOAD FILES". The main area contains a code editor with the following C++ code:

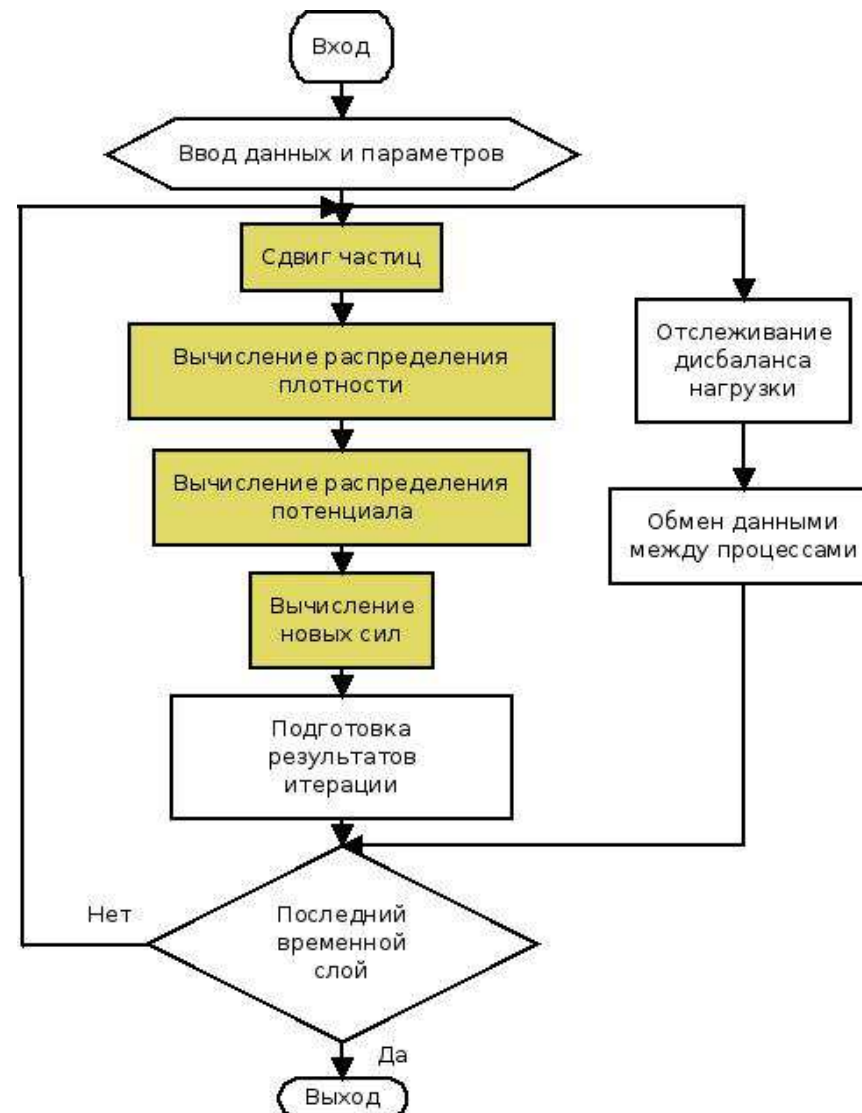
```
13 int rank, size;
14 MPI_Init(&argc, &argv);
15 MPI_Comm_size(MPI_COMM_WORLD, &size);|
16 MPI_Comm_rank(MPI_COMM_WORLD, &rank);
17
18 std::cerr << "Rank: " << rank << ", size : " << size << "\n";
19
20 int sizeX = 102, sizeY = 102;
21 double domainSizeX = 1.0, domainSizeY = 1.0;
22 double hx = domainSizeX/(sizeX-1);
23 double hy = domainSizeY/(sizeY-1);
24
25 int localSizeX = (sizeX-2)/size+2;
26 int localStartX = rank*((localSizeX-2)*rank+1)+0;
27
28 double* field1 = new double[localSizeX*sizeY];
29 double* field2 = new double[localSizeX*sizeY];
30
31 //fill-in the border values
32 for(int ix = 1; ix<sizeX-1; ix++)
33     field1[0*localSizeX+ix] = field2[0*localSizeX+ix] = u(ix+localStartX, 0);
34 for(int ix = 1; ix<sizeX-1; ix++)
35     field1[(sizeY-1)*localSizeX+ix] = field2[(sizeY-1)*localSizeX+ix] = u((ix+localStartX)*hx, 1.0);
36 if(0 == rank)
37     for(int iy = 0; iy<sizeY; iy++)
38         field1[iy*localSizeX+0] = field2[iy*localSizeX+0] = u(0, iy*hy);
39 if(size-1 == rank)
40     for(int iy = 0; iy<sizeY; iy++)
41         field1[iy*localSizeX+localSizeX] = field2[iy*localSizeX+localSizeX] =
42 u(localStartX+localSizeX, iy*hy);
43 std::cerr << "Init done\n";
44
45 MPI_Finalize();
46 }
```

On the right side of the code editor, there are three buttons: "SAVE", "MAKE", and "RUN".

# Фреймворки для HPCSS



# Particle-in-cell



# РIS-фреймворк

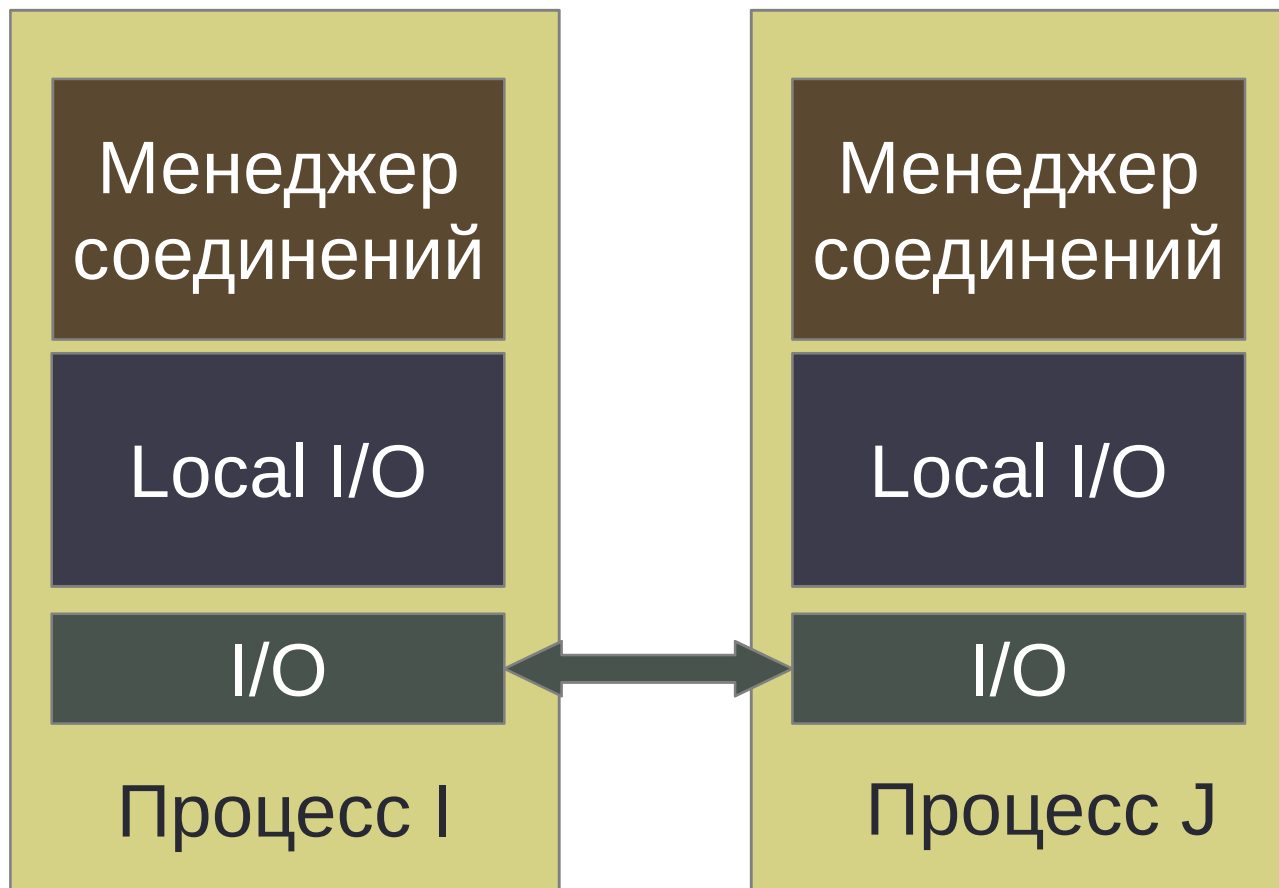
- Фреймворк содержит модули для создания MPI-программ, реализующих метод частиц-в-ячейках и адаптер
- Пользователь посредством Адаптера задает необходимые параметры и описывает вычислительные действия, необходимые для решения своей задачи

# Обеспечение отказоустойчивости

Основные идеи:

- Обеспечить возможность разработчика контролировать поведение параллельной программы в случае сбоя
- В случае сбоя одного или нескольких узлов мультимпьютера обеспечить перезапуск процесса вычислений на работоспособных узлах с частичным восстановлением состояния задачи

## Архитектура отказоустойчивой системы



# Результаты

- Разработана архитектура внутреннего сервера
- Переработан Web-инструментарий разработчика
- Разработан базовый API для внешних систем
- Разработан способ встраивания фреймворков в НРССС
- Получены результаты расчета тестовой задачи методом PIC на основе фреймворка
- Разработана архитектура системы обеспечения отказоустойчивости и начата разработка ее прототипа

# Спасибо за внимание!

Разработчики:

Беляев Николай НГТУ ФПМИ

Вайцель Сергей НГТУ ФПМИ

Волков Александр НГУ ФИТ

Куратор:

Городничев

Максим Александрович

ИВМиМГ СО РАН



# Роли в команде

Беляев Николай

обеспечение fault tolerance

Вайцель Сергей

Web-интерфейс со встроенными IDE и Dashboard

Волков Александр

фреймворк для PIC, адаптер фреймворков