

# Web-портал для проведения виртуальных экспериментов в распределенных вычислительных средах\*

Е.А. Захаров

Южно-Уральский государственный университет

Системы суперкомпьютерного моделирования, как правило, требуют соблюдения определенного, зачастую, далеко не тривиального технологического цикла, что ограничивает круг потенциальных пользователей системы. Для снижения влияния этого фактора и решения проблем интеграции систем компьютерного проектирования в вычислительной среде была предложена технология DiVTB, позволяющая инкапсулировать процесс постановки и решения определенного класса задач. В результате взаимодействие пользователя с вычислительной системой, необходимое для постановки задачи и получения результатов ее моделирования, сводится к уточнению ряда параметров из фиксированных диапазонов. В статье рассматриваются архитектурные и алгоритмические особенности реализации компонента DiVTB Portal.

## 1. Введение

На **Рис. 1** изображена архитектура системы DiVTB (Distributed Virtual Test Bed - Распределенный Виртуальный Испытательный Стенд, РаВИС) [1, 2]. Одним из немаловажных компонентов этой системы является интерфейс пользователя, предоставляемый ему для взаимодействия с вычислительной средой и распределенными ресурсами. В качестве приложения, играющего роль такого интерфейса, ранее была начата разработка компонента DiVTB Portal [3].

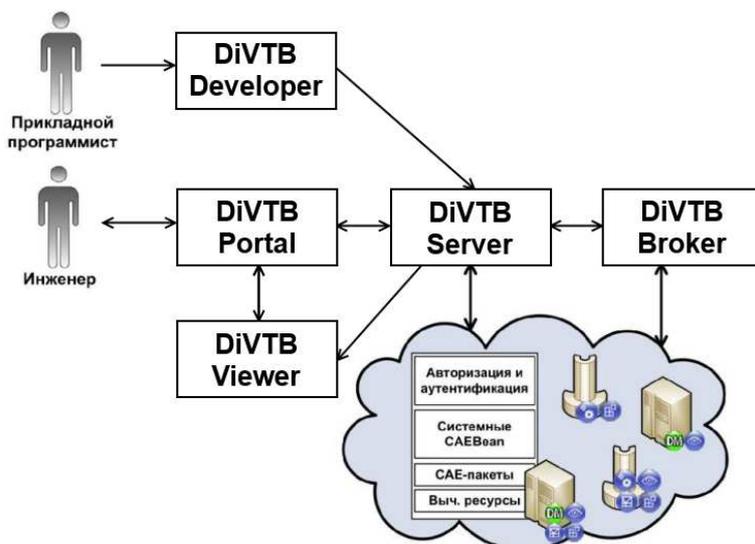


Рис. 1. Архитектура системы DiVTB

Компонент DiVTB Portal тесно взаимодействует с DiVTB Server [4], который отвечает за исполнение задач, а так же хранение данных, включая исходные файлы доступных для исполнения испытательных стендов.

Исполнение задач сервером DiVTB происходит на основе ресурсов, предоставляемых системой DiVTB Broker [6, 7]. Данная система производит сбор, анализ и распределение нагрузки в вычислительной среде между доступными вычислительными ресурсами.

Исходные файлы доступных для исполнения испытательных стендов формируются еще одним компонентом системы - DiVTB Developer [5]. Данный компонент предоставляет разра-

\* Исследование выполнено при финансовой поддержке РФФИ в рамках научного проекта № 11-07-00478-а и Министерства образования и науки РФ (государственное задание 8.3786.2011).

ботчику интерфейс для формирования файлов проектов, включая, как разработку интерфейса испытательного стенда задач для DiVTB Portal, так и разработку описания процесса исполнения задачи для DiVTB Server.

Как правило, результаты работы инженерных пакетов представляют собой не малые объемы данных, что затрудняет их передачу конечному пользователю системы, а так же требуют дальнейшей обработки для проведения анализа. Все это предъявляет дополнительные требования к производительности вычислительных устройств пользователей. Данную задачу решает последний компонент системы DiVTB - сервис интерактивной визуализации DiVTB Viewer [8].

## 2. Архитектура DiVTB Portal

*Инженер*, взаимодействуя с web-интерфейсом DiVTB Portal (Рис. 2), получает возможность просматривать список доступных ему испытательных стендов, задавать параметры для конкретной задачи и запускать ее на исполнение. Web-интерфейс отвечает за аутентификацию пользователей системы и определение прав доступа к ее компонентам.

*Менеджер задач* включает в себя функционал, как запуска заданий на исполнение, так и управления уже существующими задачами пользователя, независимо от их текущего состояния.

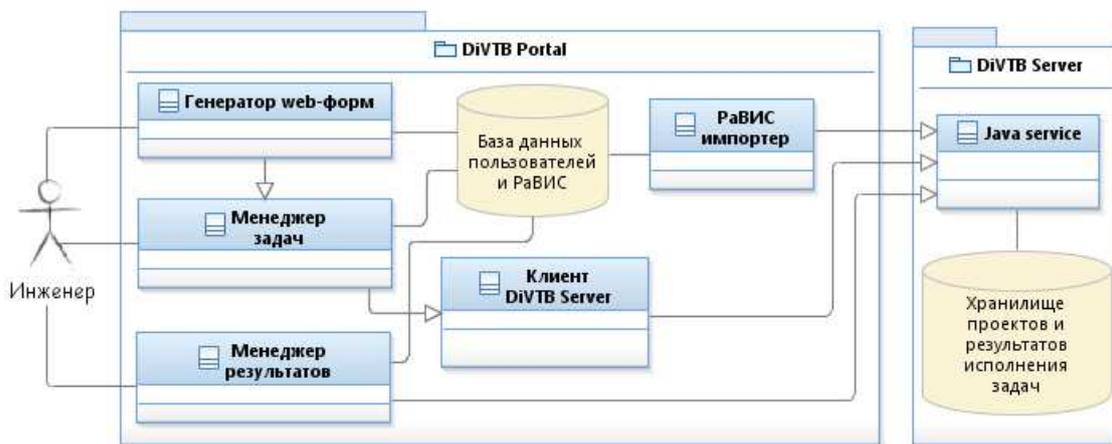


Рис. 2. Архитектура DiVTB Portal

*Генератор web-форм* отвечает за хранение проблемных оболочек DiVTB и автоматическую генерацию форм, необходимых для постановки задач. Сформированные задачи в дальнейшем могут быть запущены на исполнение с помощью менеджера задач.

*Клиент DiVTB Server* содержит в себе основы реализации взаимодействия с DiVTB Server и предоставляет всем остальным компонентам web-портала, которым, для реализации своего функционала, необходимо взаимодействие с сервером, удобный интерфейс.

*Менеджер результатов* исполнения задач, позволяет пользователю DiVTB Portal получить как результаты решения задачи в том виде, в котором их предоставляет конкретный пакет, так и информацию о ходе самого процесса проведения расчетов. *Импортер РаВИС* (распределенных виртуальных испытательных стендов) позволяет обновлять и поддерживать в актуальном состоянии список доступных распределенных виртуальных испытательных стендов, хранящихся на DiVTB Server.

## 3. Взаимодействие web-портала с компонентом DiVTB Server

Согласно архитектуре системы DiVTB (Рис. 1), разрабатываемый web-портал осуществляет взаимодействие с двумя компонентами: DiVTB Server и DiVTB Viewer. В текущей реализации системы DiVTB, DiVTB Server предоставляет web-порталу доступ к следующему функционалу:

- Получение оболочек поддерживаемых распределенных виртуальных испытательных стендов;

- Формирование задач на основе конкретного испытательного стенда, путем задания различных параметров;
- Запуск и отслеживание состояния задач пользователей DiVTB Portal;
- Получение результатов исполнения задачи в виде zip-архива, содержащего файлы, соответствующие используемым программным пакетам.

### 3.1 Импорт доступных испытательных стендов в DiVTB Portal

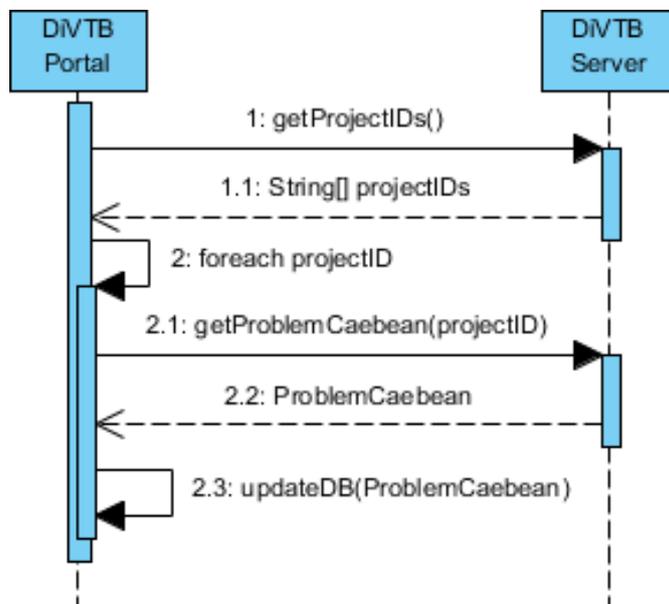
С течением времени список испытательных стендов, доступных в системе DiVTB, меняется. Это обусловлено появлением новых, изменением существующих или удалением уже не используемых стендов. Удаление испытательного стенда или изменение его параметров могут привести к потере данных ранее запущенных пользователями задач. Отсюда вытекает необходимость поддержания локальной базы данных для DiVTB Portal, хранящей различные версии когда-либо использовавшихся испытательных стендов.

Все доступные для исполнения испытательные стенды размещаются в хранилище проектов DiVTB Server. Каждый испытательный стенд, имеет свой уникальный идентификатор uid, опираясь на который DiVTB Portal может определить наличие или отсутствие его в своей локальной базе данных. Для поддержания базы данных доступных распределенных виртуальных испытательных стендов системы DiVTB в web-портал был добавлен компонент осуществляющий их импорт из хранилища DiVTB Server. Для достижения данной цели были использованы методы компонента DiVTB Server, приведенные на **Рис. 3**.

```
String[] getProjectIDs();
ProblemCaebean getProblemCaebean(String caebeanUid);
```

**Рис. 3.** Методы компонента DiVTB Server, используемые при реализации подсистемы импорта испытательных стендов в DiVTB Portal

При осуществлении импорта испытательных стендов происходит получение идентификаторов всех доступных проектов с помощью метода getProjectIDs. Далее для каждого идентификатора проверяется его наличие или отсутствие в базе данных DiVTB Portal (**Рис. 4**).



**Рис. 4.** Процесс обновления базы данных доступных испытательных стендов DiVTB Portal

В случае отсутствия проекта в базе данных у DiVTB Server запрашивается проблемная оболочка для данного проекта вызовом метода getProblemCaebean. Полученный испытательный стенд размещается в базе данных DiVTB Portal, после чего его можно сделать доступным для выбранных пользователей системы.

В случае, когда проект присутствует в базе данных DiVTB Portal, но отсутствует в базе данных DiVTB Server, данный проект переводится в состояние "не активен". Данное состояние означает невозможность для пользователей портала запускать задачи, основанные на данном испытательном стенде, при этом сохраняется возможность просмотра параметров запуска и результатов решенных ранее задач.

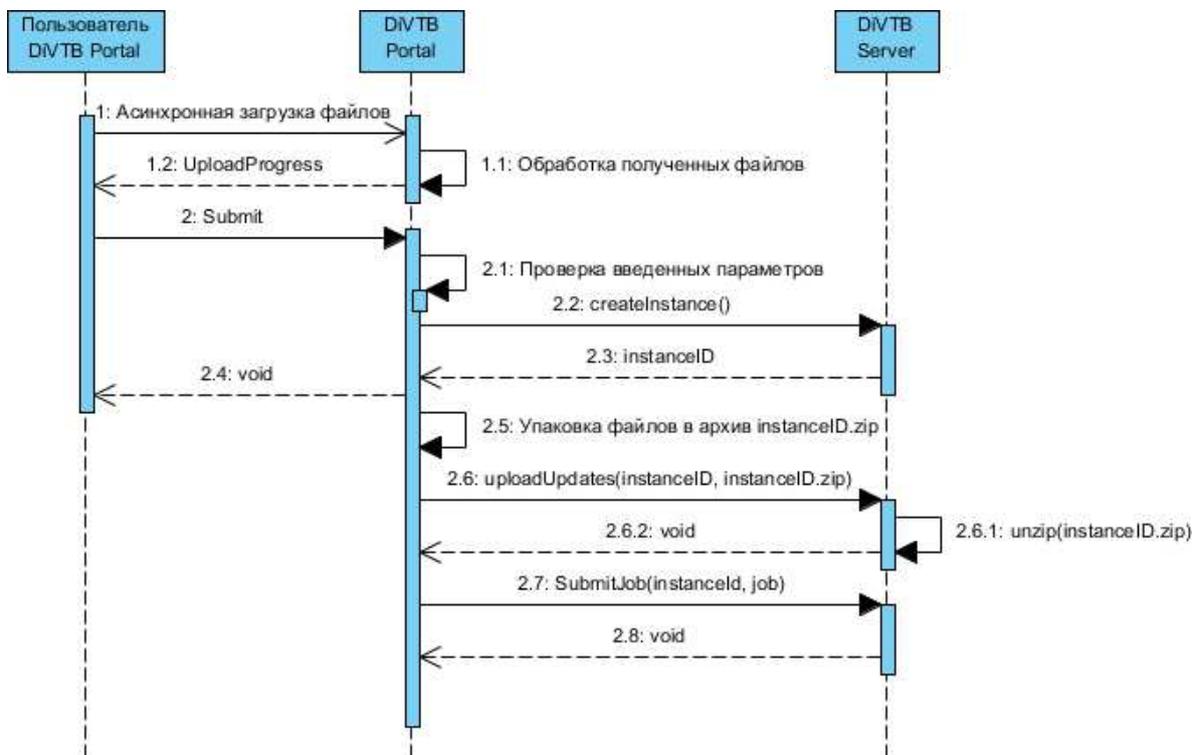
### 3.2 Формирование и запуск задачи

На основе выбранного пользователем испытательного стенда может быть сформирована и запущена на исполнение задача. Для реализации данной возможности были использованы методы, предоставляемые компонентом DiVTB Server, приведенные на **Рис. 5**.

```
String createInstance();
void uploadUpdates(String instanceUid,
                  Base64Binary zipArchive);
void submitJob(String instanceUid, ProblemCaebean job);
```

**Рис. 5.** Методы компонента DiVTB Server, использованные при реализации функции формирования и запуска задач порталом

Для формирования задачи необходимо создать instance в DiVTB Server, в рамках которого осуществляется вся дальнейшая работа. Сперва вызывается метод createInstance, который создает новый instance и возвращает присвоенный ему уникальный идентификатор uid. Полученный идентификатор, для удобства, используется порталом в качестве идентификатора задачи. Затем пользователь DiVTB Portal задает параметры испытательного стенда с помощью автоматически сгенерированной формы. Данные параметры дополняют испытательный стенд, тем самым формируя задачу, которая будет запущена в рамках созданного ранее instance вызовом метода submitJob.



**Рис. 6.** Процесс передачи файлов в качестве параметров испытательного стенда

В процессе взаимодействия с пользователями системы DiVTB выяснилось, что для постановки некоторых задач требуется возможность передачи файлов ресурсам системы DiVTB в качестве входных параметров. Данный функционал был реализован в виде дополнительного действия по передаче zip-архива, содержащего входные файлы. На **Рис. 6** изображен процесс

передачи файлов в качестве параметров испытательного стенда. На этапе задания параметров пользователь выбирает файлы необходимые для загрузки в качестве входных параметров испытательного стенда. Данные файлы асинхронно загружаются в DiVTB Portal, где временно хранятся до момента запуска задачи на исполнение (Рис. 7).

File name:  [Select file](#)  
Значение по умолчанию: GA\_OB\_DH.DAT

File name:  X  
Значение по умолчанию: GA\_OB\_DH.DAT

Рис. 7. Интерфейс пользователя DiVTB Portal при асинхронной загрузке файлов

В момент запуска все исходные параметры, включая файлы, проходят процесс проверки на корректность. После этого файлы, при наличии, упаковываются в zip-архив. Далее для отправки архива с файлами DiVTB Portal вызывает метод `uploadUpdates`, который загружает и распаковывает его в "рабочей директории" задачи на DiVTB Server. В свою очередь, при запуске задачи параметры, имеющие тип "File", будут содержать имена загруженных пользователем файлов.

### 3.3 Мониторинг и управление задачами пользователя

Все задачи в системе DiVTB могут находиться в одном из пяти состояний:

- NOT\_STARTED - начальное состояние, соответствующее еще не запущенной задаче;
- RUNNING - промежуточное состояние, присваивающееся задаче непосредственно в момент вызова метода `submitJob`;
- HELD - промежуточное состояние, присваивающееся задачам, исполнение которых временно приостановлено по каким-либо причинам;
- SUCCESSFULL - конечное состояние, соответствующее успешному завершению исполнения задачи;
- FAILED - конечное состояние, соответствующее экстренному завершению исполнения задачи в следствие возникновения каких-либо ошибок.

На Рис. 8 приведены методы DiVTB Server, которые были использованы в DiVTB Portal для обновления информации о состоянии и работы с результатами уже завершенных задач пользователей.

```
String getStatus(String instanceUid);  
long getExecTime(String instanceUid);  
Base64Binary getResultsArch(String instanceUid);  
Base64Binary getWorkArch(String instanceUid);  
void removeResultDir(String instanceUid);  
void removeWorkDir(String instanceUid);
```

Рис. 8. Методы компонента DiVTB Server, использованные при реализации работы с задачами пользователя DiVTB Portal

При обращении пользователя портала к задачам, имеющим не конечное состояние, портал использует метод `getStatus` для получения текущего состояния каждой задачи. В случае изменения состояния с начального или промежуточного на конечное, помимо самого состояния, дополнительно запрашивается время исполнения задачи с помощью вызова метода `getExecTime`.

## Менеджер задач

<input type="checkbox"/> Имя задачи	Статус	Запущена	Завершена	Действия
<input type="checkbox"/> apim		2013-01-31 15:46:51	???	
<input type="checkbox"/> Magnetization of ultrathin film		2013-01-31 15:26:51	2013-01-31 15:27:42	
<input type="checkbox"/> mdsim-test		2012-12-26 15:05:37	2012-12-26 15:05:55	

Рис. 9. Интерфейс пользователя DiVTB Portal при получении результатов исполнения задач с различным конечным состоянием

При достижении задачей одного из конечных состояний, пользователю предоставляется возможность получить либо финальный результат, либо промежуточные итоги исполнения, в зависимости от факта успешности завершения (Рис. 9).

При удалении пользователем DiVTB Portal какой-либо задачи так же удаляются и результаты ее исполнения, хранящиеся в DiVTB Server. Конечный результат имеется только у задач, находящихся в состоянии SUCCESSFULL. Результаты этих задач будут удаляться вызовом метода `removeResultDir`, а во всех остальных случаях - вызовом метода `removeWorkDir`.

## 4. Интерфейс постановки задачи

При работе с интерфейсом DiVTB Portal пользователь может столкнуться с тремя различными способами задания параметров задач:

- Простое указание значения параметра (любые числа и строки);
- Выбор из ограниченного набора значений параметра;
- Передача файла в качестве параметра задачи.

Генератор web-форм DiVTB Portal позволяет пользователям задавать параметры испытательных стендов, путем указания их значений в текстовом поле ввода (Рис. 10). Параметры, в данном случае, могут иметь различный тип, в том числе: String, Int, Float, File и т.д. Указание о том, какой формат имеет параметр размещаются разработчиком интерфейса в комментариях.

### Параметры запуска:

Аргументы командной строки  
Справка: `scimq -h`  
Значение по умолчанию: `-h`

Рис. 10. Пример элемента интерфейса DiVTB Portal для задания значения параметра

При создании испытательного стенда в DiVTB Developer для любого параметра, за исключением параметров файлового типа, может быть ограничено множество их значений, с помощью перечисления всех возможных значений. При наличии такого перечисления поле ввода значения параметра будет представлено пользователю системы в виде элемента выбора значения из предложенного списка (Рис. 11).

Значение по умолчанию: 256

### Выполняемая операция:

Имя класса, реализующего операцию  
Имя Java класса, который реализует нужную операцию обработки изображений  
Значение по умолчанию: `ru.uran.imm.hadoopimage.client.Nope`

`ru.uran.imm.hadoopimage.client.EdgeDetection`  
`ru.uran.imm.hadoopimage.client.Resize`  
`ru.uran.imm.hadoopimage.client.Convert`  
`ru.uran.imm.hadoopimage.client.FFT`  
`ru.uran.imm.hadoopimage.client.MatrixFilter`  
`ru.uran.imm.hadoopimage.client.Nope`

Рис. 11. Пример элемента интерфейса DiVTB Portal для задания значения параметра, при наличии ограниченного набора его возможных значений

Процесс задания и передачи параметров файлового типа был рассмотрен ранее в разделе 3.2 Формирование и запуск задачи. На Рис. 7 изображен пример интерфейса портала до выбора файла пользователем и после его загрузки в хранилище файлов портала. Параметры типа File сочетают в себе 2 функции:

- обеспечивают загрузку выбранных файлов на вычислительный узел при запуске задачи;

- передают исполнителю задачи одноименный параметр типа String, в котором содержится имя загруженного пользователем файла.

## Испытательный стенд: Magnetization of ultrathin film

**Идентификационные параметры задачи:**

---

Название задачи  
 Название задачи, которое будет отображаться в менеджере задач   
 Значение по умолчанию: Magnetization of ultrathin film

---

**Необходимые ресурсы:**

---

Количество ядер  
 Необходимое количество процессорных ядер для постановки задачи  шт  
 Значение по умолчанию: 1

---

Объем ОЗУ  
 Объем оперативной памяти, необходимый для постановки задачи  МБ  
 Значение по умолчанию: 256

---

**Параметры задачи:**

---

Целое число, большее единицы  
 Число монослоев плёнки  число  
 Значение по умолчанию: 2

---

Дробное положительное число  
 Точность вычисления температуры Кюри  число  
 Значение по умолчанию: 0.1

---

Строка, задающая тип решателя  
 Решатель    
 Значение по умолчанию: scipy\_solve

---

Строка формата выходного рисунка  
 Формат выходного рисунка кривой намагниченности    
 Значение по умолчанию: PNG

---

Рис. 12. Пример интерфейса DiVTB Portal для задачи "Magnetization of ultrathin film"

На Рис. 12 приведен пример интерфейса испытательного стенда DiVTB Portal для задачи «Magnetization of ultrathin film».

## 5. Заключение

В данной работе были рассмотрены архитектурные особенности реализации компонента DiVTB Portal. Были освещены особенности взаимодействия разработанного портала с компонентом DiVTB Server.

Дальнейшее развитие DiVTB Portal будет направлено на совершенствование архитектуры, развитие функций администрирования портала и интеграцию с сервисом интерактивной визуализации DiVTB Viewer.

## Литература

1. Радченко Г.И. Распределенные виртуальные испытательные стенды: использование систем инженерного проектирования и анализа в распределенных вычислительных средах // Вестник ЮУрГУ. Серия "Математическое моделирование и программирование". 2011. № 37(254). Вып. 10. С. 108-121.
2. Radchenko G., Hudyakova E. A Service-Oriented Approach of Integration of Computer-Aided Engineering Systems in Distributed Computing Environments // UNICORE Summit 2012 Proceedings. Forschungszentrum Julich, 2012. P. 57-66.

3. Захаров Е.А. Веб портал для проведения виртуальных экспериментов в распределенных вычислительных средах // Научный сервис в сети Интернет: поиск новых решений: Труды Международной суперкомпьютерной конференции (17-22 сентября 2012 г., г. Новороссийск). М.: Изд-во МГУ, 2012. С. 257-260.
4. Савченко Д.И., Радченко Г.И. DiVTB Server: среда выполнения виртуальных экспериментов. См. настоящий сборник
5. Худякова Е.С., Радченко Г.И. Веб-система разработки и внедрения распределенных виртуальных испытательных стендов // Материалы XII Всероссийской конференции "Высокопроизводительные параллельные вычисления на кластерных системах" (Нижний Новгород, 26-28 ноября 2012 г.). Нижний Новгород: Изд-во Нижегородского государственного университета, 2012. С. 430-434.
6. Шамакина А.В. Брокер ресурсов для поддержки проблемно-ориентированных сред // Вестник ЮУрГУ. Серия "Вычислительная математика и информатика". 2012. № 46(305). Вып. 1. С. 88-98.
7. Shamakina A. Brokering Service for Supporting Problem-Oriented Grid Environments // UNICORE Summit 2012 Proceedings, Forschungszentrum Julich, 2012. P. 67-75.
8. Диков Д.А., Радченко Г.И. Интеграция сервиса интерактивной визуализации для распределенных виртуальных испытательных стендов с системой CAEBeans // Высокопроизводительные параллельные вычисления на кластерных системах. Материалы XII Всероссийской конференции (Нижний Новгород, 26-28 ноября 2012 г.). Нижний Новгород: Изд-во Нижегородского государственного университета, 2012. С. 134-138.