

## УЧЕБНО-ИССЛЕДОВАТЕЛЬСКАЯ СИСТЕМА ОБРАБОТКИ БОЛЬШИХ ДАННЫХ



**А.И. Демидчук**

Заведующий лабораторией  
высокопроизводительных  
вычислений БГУИР



**Д.Ю. Перцев**

Ассистент кафедры кафедрой  
электронных вычислительных  
машин БГУИР



**Д.И. Самаль**

Заведующий кафедрой элек-  
тронных вычислительных  
машин БГУИР, кандидат  
технических наук, доцент

Белорусский государственный университет информатики и радиоэлектроники, Республика Беларусь  
E-mail: samal@bsuir.by.

*Abstract.* Educational and research system to merge different frameworks to common interface is introduced. The description of class diagram is provided to apply for this purpose.

На сегодняшний день существует огромное число проектов по анализу и обработке в области Big Data (TensorFlow[1], Theano[2], множество проектов на Python, C++), каждый из которых имеет свои плюсы и минусы. Например, реализации на Python не поддерживают работу с графическим процессором, но включают большое число уже готовых к использованию алгоритмов. Проекты TensorFlow[1], Theano[2] наоборот включают ограниченное число уже готовых алгоритмов, но предоставляют интерфейс для авторских разработок, кроме того, поддерживается обработка вычислений на GPU.

Целью работы, проводимой в рамках научно-исследовательской лаборатории в БГУИР, является разработка учебно-исследовательской системы, позволяющей унифицировать интерфейс доступа к алгоритмам анализа данных, предоставить возможность динамического построения цепочки вызовов обработчиков, сбора и анализа статистики исполнения.

Общий вид программно-аппаратной платформы исполнения, представлен на рисунке 1.

В качестве клиента может выступать любой пользователь с персональным компьютером либо ноутбуком. Для упрощения организации доступа предполагается разработка специального сайта либо Eclipse-плагина, позволяющего в удобной форме сформировать последовательность операций, передать задание на кластер, получить результат и вернуть результат конечному пользователю.

Учебно-исследовательская система включает в себя следующие модули:

- интерфейс пользователя;
- сервисы;
- библиотека алгоритмов.

Клиент через сайт взаимодействует с модулем «Интерфейс пользователя». Его основной задачей является получение задания от пользователя и анализ полученных данных. При этом клиент вправе самостоятельно сформировать последовательность действий из допустимого набора либо указать задачу, которую требуется решить.

В случае, если пользователь указывает последовательность действий, задачей модуля «интерфейс пользователя» является последовательная передача управления сервисам для вы-

зова необходимых функций и формирование результата. Результат работы передается клиенту.

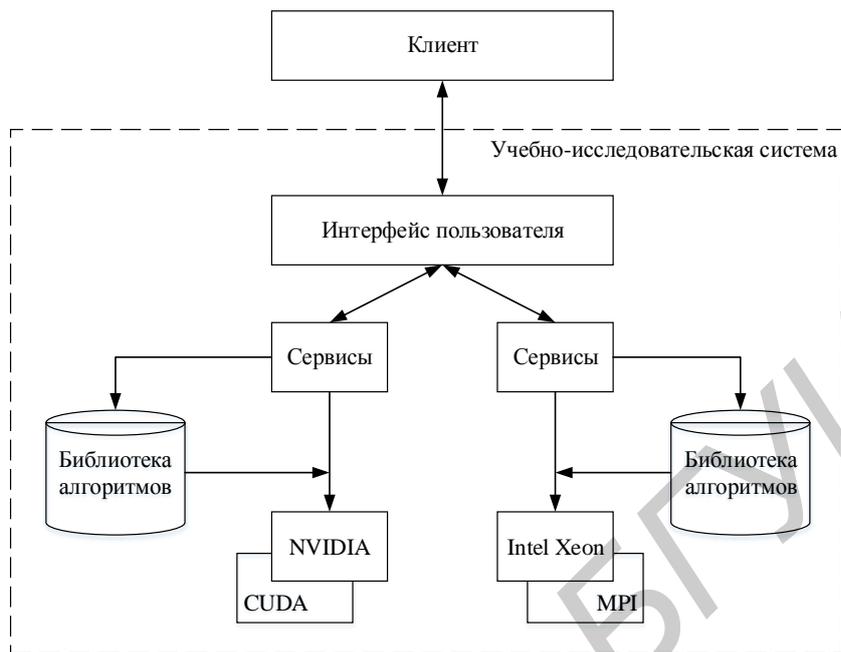


Рис. 1. Структурная схема программно-аппаратной платформы исполнения

Если пользователь указал решаемую задачу, но не сформулировал алгоритм решения, задачей модуля является формирование последовательности действий, после чего схема работы повторяет предыдущий вариант.

Обязательными требованиями к модулю «Интерфейс пользователя» являются:

- унификация интерфейса пользователя, т.к. система включает в себя множество различных компонентов с алгоритмами;

- должна быть возможность динамического обновления поддерживаемого набора функций и информирование об этом клиента. Наиболее оптимальным подходом в этом случае является применение клиент-серверной архитектуры на основе Web-технологий;

- предоставление доступа к уже готовым решениям задач (например, кластерный анализ данных), а также к отдельным алгоритмам, чтобы пользователь самостоятельно мог сформировать последовательность действий.

К ограничениям для данной модели можно отнести:

- минимальный набор поддерживаемых функций на начальном этапе. Это позволит протестировать и отладить систему, разработать оптимальную схему доступа;

- версия используемого программного обеспечения, на которой гарантируется работоспособность системы;

- информирование пользователя о поддерживаемом наборе функций предполагается осуществлять через XML-структуру, передаваемую по TCP протоколу.

В основе учебно-исследовательской системы располагается вычислительный кластер БГУИР, имеющий следующую конфигурацию:

- 1 вычислительный блок (7 модулей), каждый из которых включает следующие элементы:

- блейд-стойка: GPU SuperBlade SBI-7127RG;
- центральный процессор: Intel Xeon E5-2650 (2 шт.);
- ОЗУ: 32 Gb RAM стандарта DDR3;



–промежуточное звено, связывающее библиотеку алгоритмов с контекстом пользователя;

–модуль подключаемой библиотеки алгоритмов;

–модуль загрузки/сохранения данных.

Контекст пользователя включает базовый класс *UserContext*, через который осуществляется взаимодействие пользователя с библиотекой, и *ProfileType*, определяющий права пользователя.

При инициализации системы формируется все множество поддерживаемых алгоритмов через класс *ConcreteCommandFactory*. Задачей данного класса является:

– формирование статического списка команд *Command*, поддерживаемых разрабатываемой системой;

– формирование связи с необходимыми библиотеками *Framework* через *ConcreteCommandFactory*.

Каждая подключаемая библиотека представляет собой отдельный класс *Framework*, позволяющий подключить библиотеку, провести базовые настройки без участия пользователя, предоставить список поддерживаемых алгоритмов. Каждая подключаемая библиотека наследуется от *FrameworkImpl* для стандартизации доступа.

При запросе на обработку данных через *UserContext* выполняются следующие действия:

– обращение к *ConcreteCommandFactory* для получения ссылок на объекты *Command*, необходимые для исполнения в соответствии с указаниями пользователя;

– формирование объекта *Algorithm*, представляющего собой последовательность команд *Command* для исполнения.

По мере формирования списка команд и указания необходимых параметров через класс *Attribute* начинается обработка данных и формирование результата.

Модуль загрузки и сохранения данных *Data* предоставляет унифицированный интерфейс доступа для доступа к различным репозиториям с данными. В качестве источников данных предполагаются:

– база данных (класс *InDbData*);

– файл на жестком диске (класс *OnFsData*);

– ОЗУ (класс *InMemoryData*).

Представленная учебно-исследовательская система является универсальной, легко расширяемой и адаптируемой под необходимые условия работы.

#### Литература

[1]. TensorFlow [Electronic resource]. – Mode of access: <https://www.tensorflow.org/>. – Date of access: 01.03.2017.

[2]. Theano [Electronic resource]. – Mode of access: <http://deeplearning.net/software/theano/>. – Date of access: 01.03.2017.