Московский физико-технический институт
(государственный университет)
Физтех-школа прикладной математики и информатики
Кафедра алгоритмов и технологий программирования

Система Hjudge

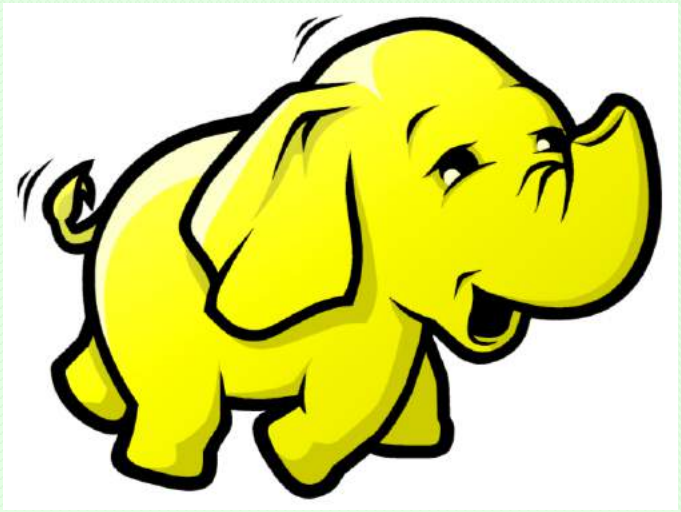
Как автоматизировать проверку заданий при изучении работы с большими данными

О.Н. Ивченко
А.А. Драль
М.А. Ройтберг

Большие данные

Экосистема Hadoop 2008

- Основные компоненты
  - Hadoop (2005)– фреймворк, реализующий парадигму MapReduce
    - включает распределенную файловую систему HDFS
  - Spark – фреймворк, обеспечивающий более быструю, по сравнению с Hadoop, обработку данных
    - ленивая обработка
    - обработка в реальном времени
  - Hive – SQL-обёртка над Hadoop,
  - HBase – БД NoSQL-типа, обеспечивает хранение больших разреженных массивов данных



Экосистема Hadoop

- Где используется

Описание

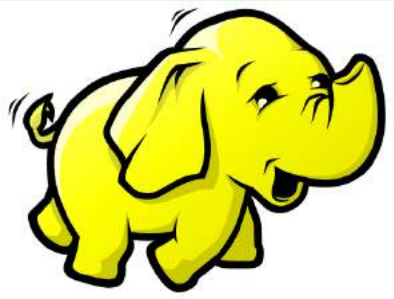
- Хранение и Обработка Больших Объемов Данных
  - http://atp-fivt.org/kurs-xobod/okurs/
- Читается с весны 2015 г.
- Направлен на работу с данными, для обработки которых нужно **> 1 машины**
- Предмет рассмотрения – экосистема Hadoop
- Практическая часть – программистские задачи по 4 основным модулям:
  - MapReduce (Hadoop Java API, Hadoop Streaming)
  - Hive
  - Spark
  - HBase

Примеры задач

- ***Задача 1:*** Посчитать число вхождений слова (term) в каждую статью (term frequency, tf). Слова очистить от знаков пунктуации.

  Данные: статьи википедии (id <пробел> text)

  Формат результата: term, article id, tf



- ***Задача 2:*** Посчитать богатство языка (число уникальных слов) каждого персонажа и число реплик.

  Данные: текст «Горе от ума» (персонаж <пробел> строка реплики)

  Формат результата: персонаж, число уник. слов, число реплик

- ***Задача 3:*** Вывести TOP10 наиболее посещаемых страниц за указанный интервал времени. На входе - интервал времени, на выходе - список наиболее популярных страниц.

  Данные: история посещений пользователями страниц.

Примеры задач

- ***Задача 4:*** Вывести суммарное распределение количества посещений по часам (для каждого часа в сутках вывести количество посещений, пришедшее в этот час).

  Данные: история посещений пользователями страниц.

- ***Задача 5:*** Требуется написать пользовательскую функцию Hive. На вход ей подаётся адрес сети и адрес маски (именно в таком порядке). На выходе получаем множество возможных IP-адресов в этой сети.

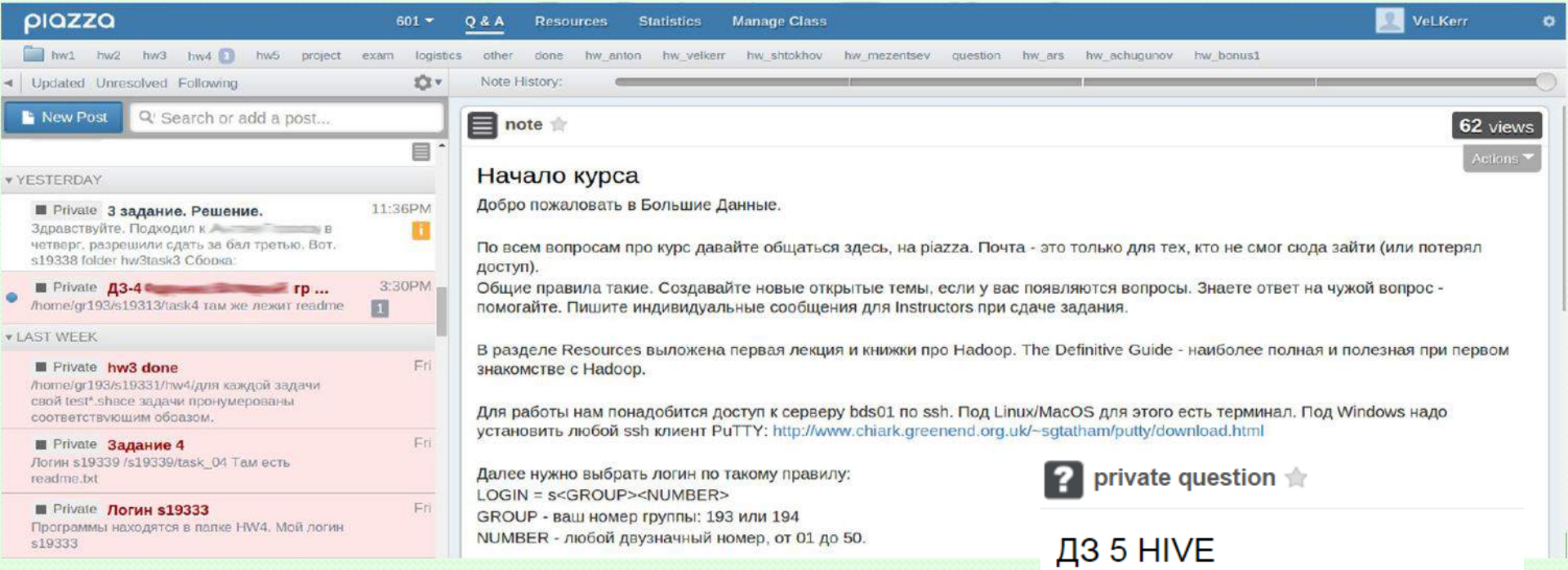
  Данные: история посещений пользователями страниц.

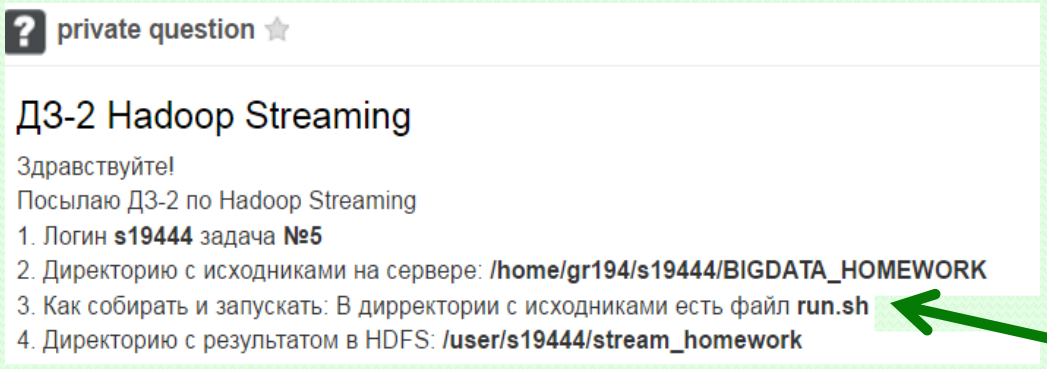
Данные в задачах

- имеют объём до нескольких десятков Гб
  - ✓ (нельзя быстро обработать на 1 машине)
- читаются из файла в HDFS
- записываются в HDFS или выводятся в консоль

3. Проверка заданий

Сдача задания студентом: PIAZZA





Процесс проверки заданий

Этапы проверки одного задания:

1. Сборка на Hadoop-кластере
2. Запуск программы
3. Мониторинг процесса работы
4. Проверка результата
5. Code review

Процесс проверки заданий

Этапы проверки одного задания:

1. Сборка на Hadoop-кластере (~ 5 с)
2. Запуск программы (~ 10 с)
3. Мониторинг процесса работы (~ 20 с)
4. Проверка результата (~ 90 с)
5. Code review (~ 120 с)

4. Инструменты автоматизации

1. Сборка на Hadoop-кластере

4. Инструменты автоматизации

1. Сборка на Hadoop-кластере

2. Запуск программы

(самописный bash-скрипт)

4. Инструменты автоматизации

1. Сборка на Hadoop-кластере

2. Запуск программы

(самописный bash-скрипт)

3. Мониторинг процесса работы
   - для Hadoop: MrBench (можно подавать на вход свой Jar):

```
$ hadoop jar /usr/lib/hadoop-mapreduce/hadoop-mapreduce-client-jobclient.jar
mrbench –help

MRBenchmark.0.0.2
Usage: mrbench ... [-jar <local path to job jar file containing Mapper and Reducer
implementations, default is current jar file>] ...
```

   - для Hive, Spark и HBase подобных инструментов не найдено

4. Инструменты автоматизации

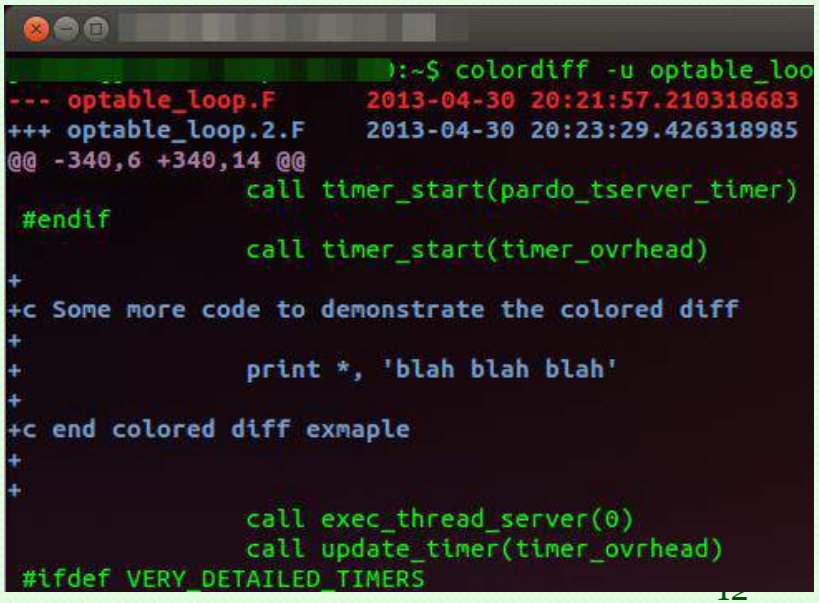
1. Сборка на Hadoop-кластере

2. Запуск программы

(самописный bash-скрипт)

3. Мониторинг процесса работы
   - для Hadoop: MrBench (можно подавать на вход свой Jar)

4. Проверка результата
   - diff (colordiff)



4. Инструменты автоматизации

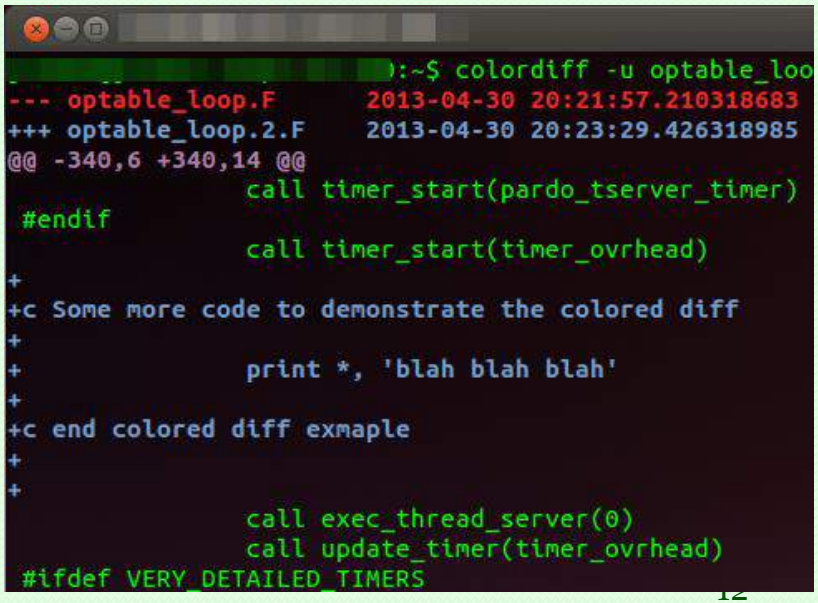
1. Сборка на Hadoop-кластере

2. Запуск программы

(самописный bash-скрипт)

3. Мониторинг процесса работы
   - для Hadoop: MrBench (можно подавать на вход свой Jar)

4. Проверка результата
   - diff (colordiff)



- *в задачах нет однозначного решения => diff не всегда информативен*
- *данных очень много*

4. Инструменты автоматизации

1. Сборка на Hadoop-кластере

2. Запуск программы

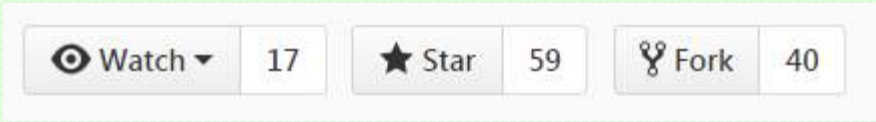
(самописный bash-скрипт)

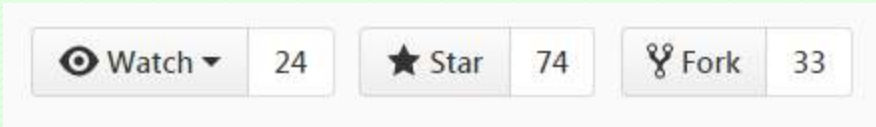
3. Мониторинг процесса работы
   - для Hadoop: MrBench (можно подавать на вход свой Jar)
4. Проверка результата
   - diff (colordiff)
5. Code review

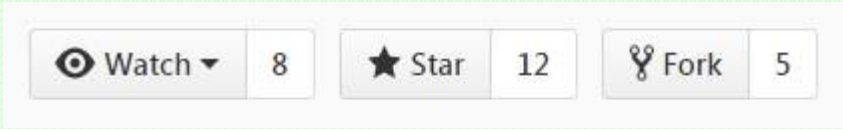
4. Инструменты автоматизации

- Модульное тестирование
  - MapReduce: MRUnit,
  - Hive:
    - Hive_test,
    - HiveRunner,
    - HiveQLUnit
  - Spark
    - Spark testing base

Для каждого сервиса существуют свои инструменты модульного тестирования. Они различаются в использовании => **их тяжело привести к общему интерфейсу**.







популярность библиотек на GitHub

5. HJudge

- Написана на Python 2.7 с использованием библиотеки PyDoop.

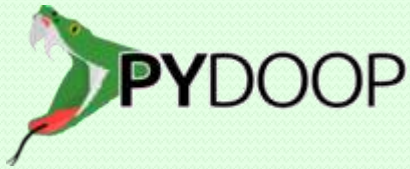
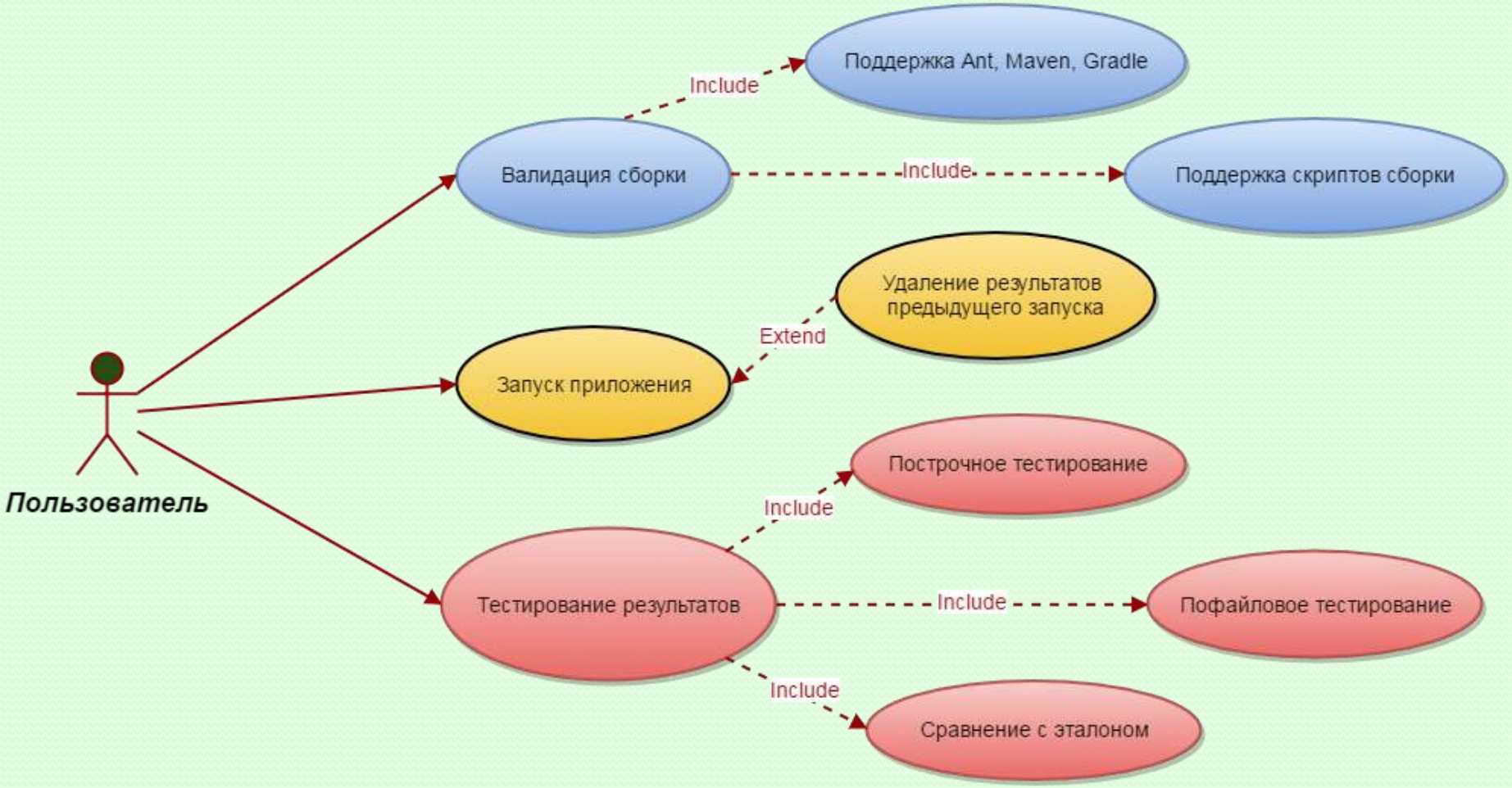
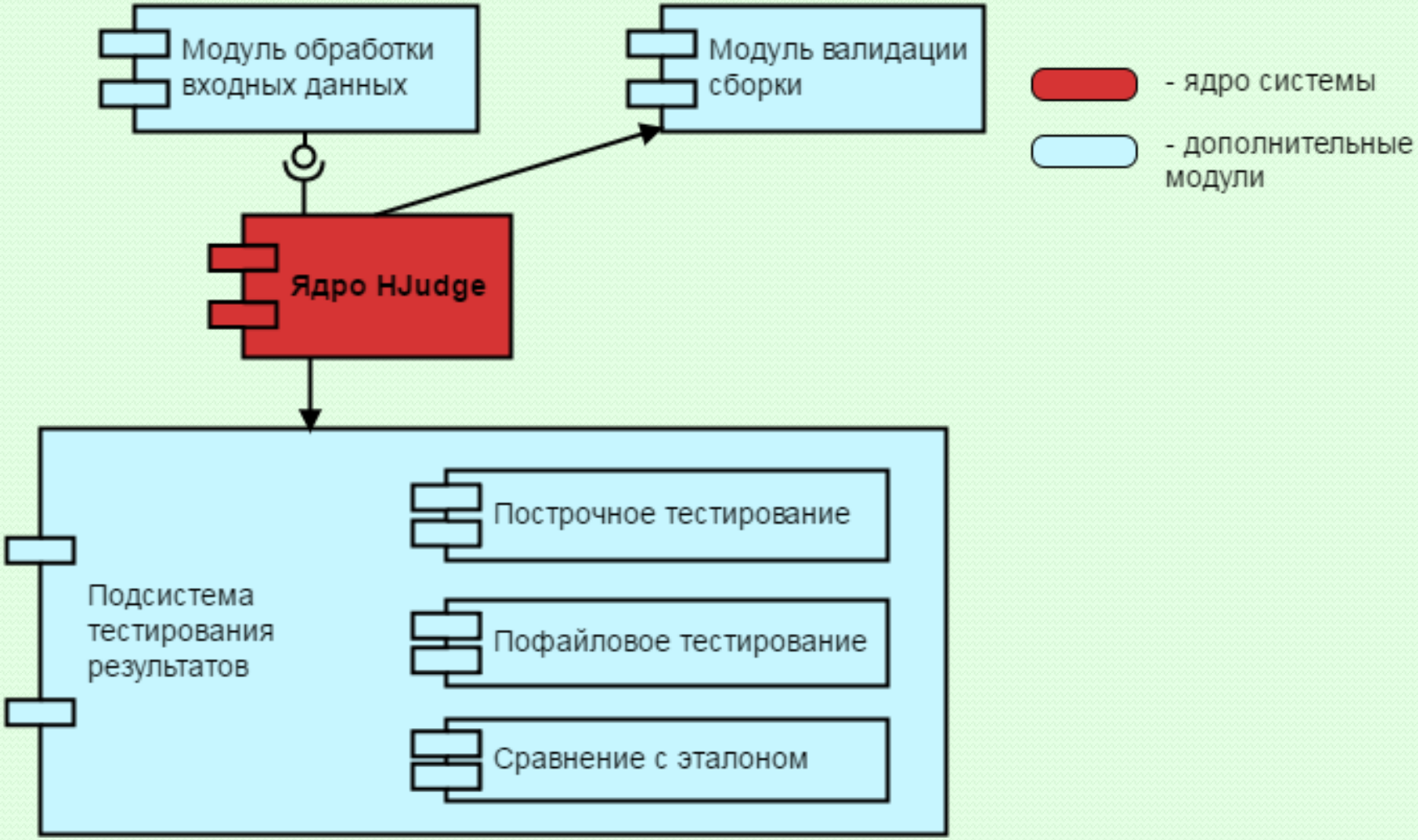


Схема работы



Архитектура

- Схема компонентов HJudge



Архитектура

- Назначение модулей

Модуль	Назначение
Модуль обработки входных данных	Получает данные из аргументов CLI и парсит
Модуль валидации сборки	Контролирует сборку тестируемого приложения
Модуль проверки результатов	Проверяет выход приложения
Ядро системы	• Запускает тестируемое приложение • Передаёт управление от модуля к модулю • Ведёт учёт проверенных приложений

- Процесс Code review на данный момент не автоматизируется

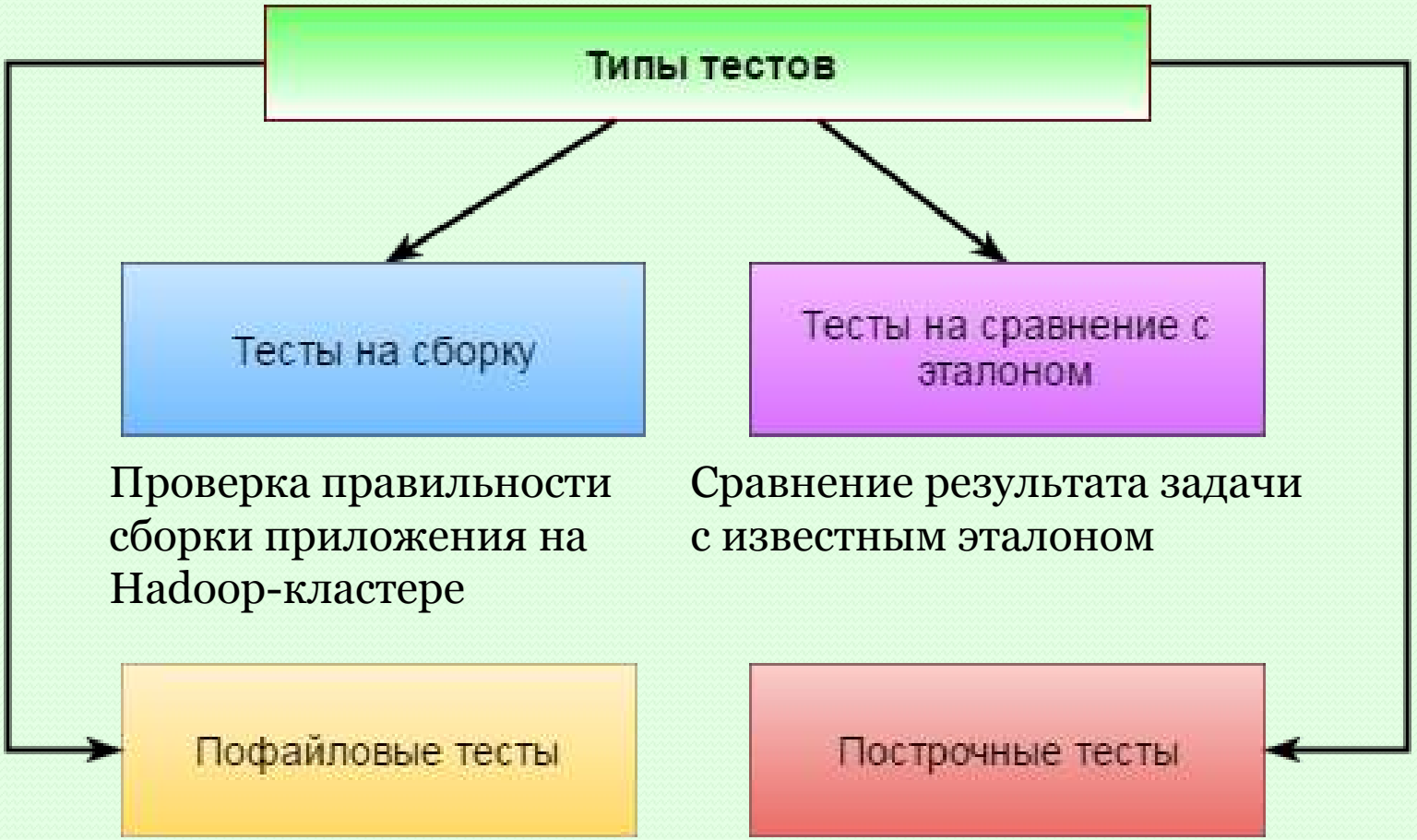
Архитектура

- Назначение модулей

Модуль	Назначение
Модуль обработки входных данных	Получает данные из аргументов CLI и парсит
Модуль валидации сборки	Контролирует сборку тестируемого приложения *(можно встраивать свои тесты)*
Модуль проверки результатов	Проверяет выход приложения *(можно встраивать свои тесты)*
Ядро системы	• Запускает тестируемое приложение • Передаёт управление от модуля к модулю • Ведёт учёт проверенных приложений

- Процесс Code review на данный момент не автоматизируется
- Пользовательские тесты – это обычные Python-функции, встраивающиеся в специальный класс

База тестов



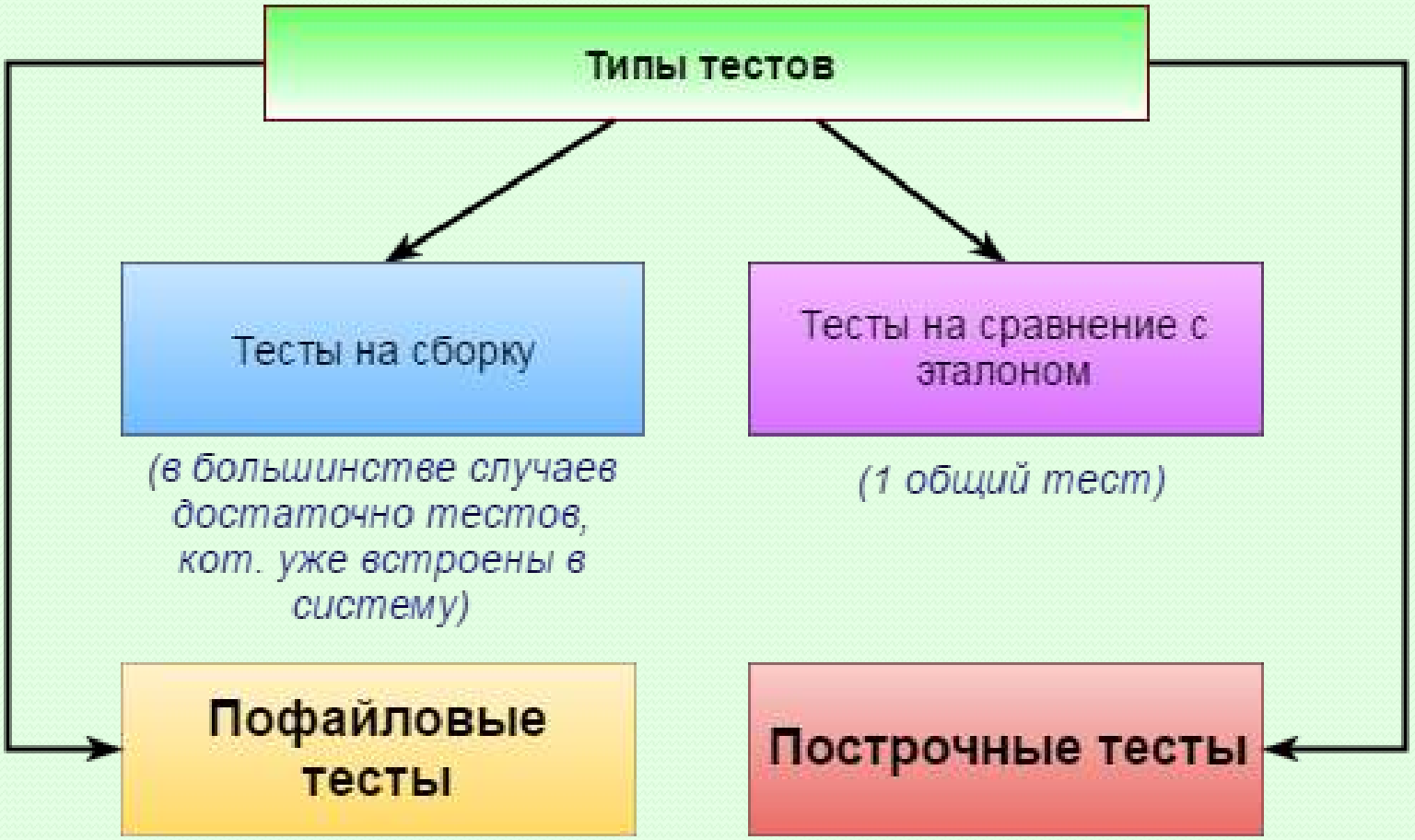
Проверка правильности сборки приложения на Hadoop-кластере

Сравнение результата задачи с известным эталоном

Проверка корректности ***файлов*** в HDFS независимо друг от друга

Проверка корректностьи ***строк*** в файлах результата независимо друг от друга

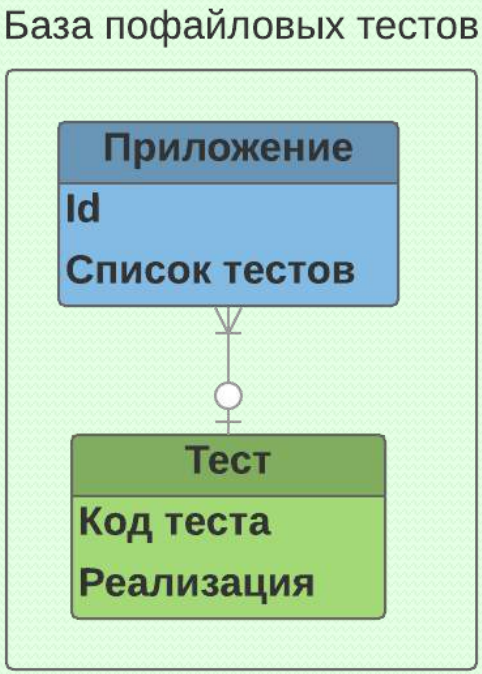
База тестов



- В основном, нужно встраивать только построчные и пофайловые тесты

База тестов

- Пофайловые тесты



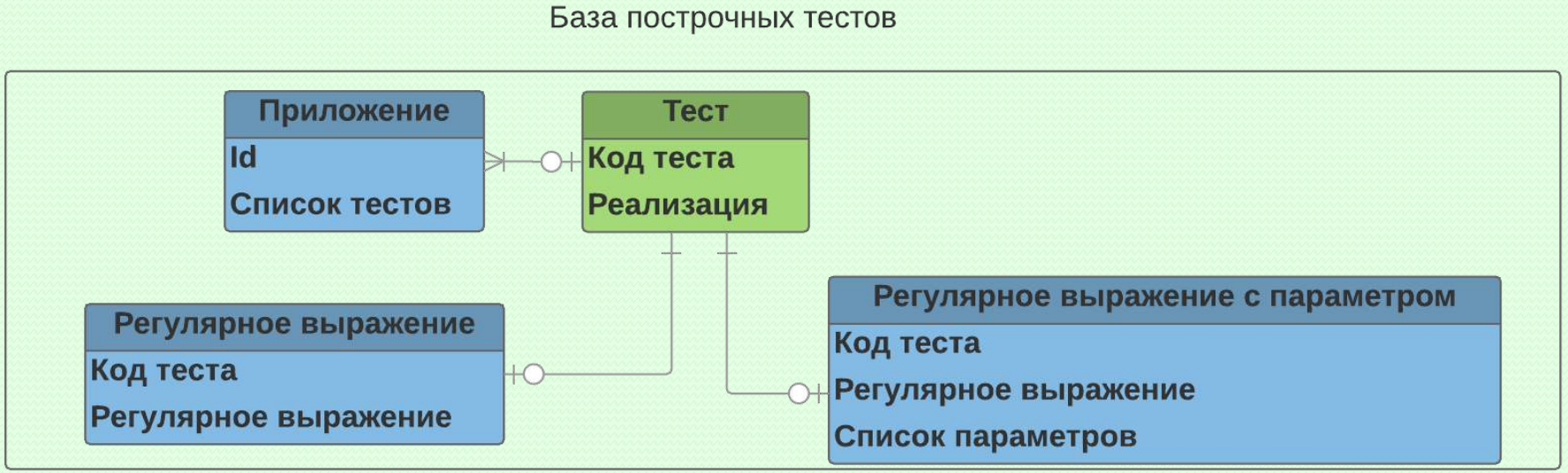

```
101    mrb2,mrb3_1,mrb8
102    mrb2,mrb3_1,mrb8
103    mrb2,mrb3_1,mrb8
104    mrb2,mrb8
105    mrb2,mrb3_1,mrb8
106    mrb2,mrb8
107    mrb2,mrb8
108    mrb7
109    mrb2,mrb3_1,mrb8
```

```
    def mra1_2(self):
        return self._mra1(5000, 50)

    def mrb12(self):
        for num, line in enumerate(self.hdfs_file):
            if num:
                return True
        return False
```

База тестов

- Построчные тесты




```
1  mrb1_1    ^\S+\s\d+$
2  mrb1_2    ^1(\.0+)?$|^0\.(?!0+$)\d+$
3  mrb1_3    ^[\S\s]+(?:\s\d+){3}\.?\d*$
4  mrb1_4    ^\S+(?:\s\d+\.?\d*){2}$
5  mrb1_5    ^(?:\S+\s){2}\d+$
6  mrb1_6    ^(?:\S{18}[,\s]){,4}\S{18}$
7  mrb1_7    ^\S+\s\d+\.?\d*(?:e-)?\d{,2}$
```

Текстовый файл

Python-класс

Отчётность по тестированию

1. Логи тестирования в реальном времени
2. Отчёт по окончании тестирования приложения

Отчётность по тестированию

1. Логи тестирования в реальном времени

```
instructor@bds02:~/tasks/s19441_task107_v2$ hjudge --run-cmd "bash run.sh" --out-dir /user/s19441/gore_ot
_uma_result/ --build maven --login s19441 --num 107
===== Testing task: 1 (HW: 107): STARTING =====
main.py ##### INFO     [2017-01-24 20:36:00,850]  You probably don't have permissions to write into out-d
ir.
main.py ##### INFO     [2017-01-24 20:36:00,850]  Out-dir will be changed to /user/instructor/s19441_task
107
main.py ##### WARNING  [2017-01-24 20:36:00,865]  Test MRB2 is specified, but not realized.
main.py ##### WARNING  [2017-01-24 20:36:00,865]  Test MRB8 is specified, but not realized.
main.py ##### WARNING  [2017-01-24 20:36:00,865]  There are no realized full tests for task  107.
main.py ##### WARNING  [2017-01-24 20:36:00,866]  There are no specified systems tests for task  107.
main.py ##### INFO     [2017-01-24 20:37:52,486]  Testing task: 1 (HW: 107): test MRB14 failed on line "5
-я княжна       0 4"!
main.py ##### INFO     [2017-01-24 20:37:52,486]  Testing task: 1 (HW: 107): test MRB6 failed on line "5-
я княжна       0 4"!
main.py ##### INFO     [2017-01-24 20:37:52,614]  Testing task: 1 (HW: 107): test MRB13 passed!
main.py ##### INFO     [2017-01-24 20:37:52,615]  Testing task: 1 (HW: 107): test MRB4_1 passed!
```

Нет прав на запись в выходную директорию, система её заменяет

Тесты, кот. есть в списке, но нет в класс с Python-функциями

Список пройденных тестов

Отчётность по тестированию

1. Логи тестирования в реальном времени

```
instructor@bds02:~/tasks/s19441_task107_v2$ hjudge --run-cmd "bash run.sh" --out-dir /user/s19441/gore_ot
_uma_result/ --build maven --login s19441 --num 107
===== Testing task: 1 (HW: 107): STARTING =====
main.py ##### INFO     [2017-01-24 20:36:00,850]  You probably don't have permissions to write into out-d
ir.
main.py ##### INFO     [2017-01-24 20:36:00,850]  Out-dir will be changed to /user/instructor/s19441_task
107
main.py ##### WARNING  [2017-01-24 20:36:00,865]  Test MRB2 is specified, but not realized.
main.py ##### WARNING  [2017-01-24 20:36:00,865]  Test MRB8 is specified, but not realized.
main.py ##### WARNING  [2017-01-24 20:36:00,865]  There are no realized full tests for task  107.
main.py ##### WARNING  [2017-01-24 20:36:00,866]  There are no specified systems tests for task  107.
main.py ##### INFO     [2017-01-24 20:37:52,486]  Testing task: 1 (HW: 107): test MRB14 failed on line "5
-я княжна       0 4"!
main.py ##### INFO     [2017-01-24 20:37:52,486]  Testing task: 1 (HW: 107): test MRB6 failed on line "5-
я княжна       0 4"!
main.py ##### INFO     [2017-01-24 20:37:52,614]  Testing task: 1 (HW: 107): test MRB13 passed!
main.py ##### INFO     [2017-01-24 20:37:52,615]  Testing task: 1 (HW: 107): test MRB4_1 passed!
```

Нет прав на запись в выходную директорию, система её заменяет

Тесты, кот. есть в списке, но нет в класс с Python-функциями

Список пройденных тестов

- Для экономии времени можно настроить систему так, что тестирование будет останавливаться при первом не пройденном тесте.

Отчётность по тестированию

2. Отчёт по окончании тестирования приложения

```
===== Testing task: 1 (HW: 107): SUMMARY =====
        Tests passed: mrb13 mrb4_1
        Tests failed: mrb14 mrb6
================================================
```

- Можно запустить на проверку сразу несколько приложений
  - Когда тестирование завершится, отчёт будет выдан по каждому

Установка

Особенности установки

- Имеет мало дополнительных зависимостей:
  1. Hadoop, Hive, Spark, HBase
  2. Python 2,7
  3. PyDoop 1.2
  4. PyPi Regex

Установка

Особенности установки

- Имеет мало дополнительных зависимостей:
  1. Hadoop, Hive, Spark, HBase
  2. Python 2,7
  3. PyDoop 1.2
  4. PyPi Regex //опционально

} Обычно установлены на Hadoop-кластере

Установка

Особенности установки

- Имеет мало дополнительных зависимостей:
  1. Hadoop, Hive, Spark, HBase
  2. Python 2,7
  3. PyDoop 1.2
  4. PyPi Regex //опционально

} Обычно установлены на Hadoop-кластере

- Установка через Pip
  - `pip install git+http://gitlab.vdi.mipt.ru/VeLKerr/hjudge.git`
- Проект пока не выложен в открытый доступ => нужна регистрация на gitlab.vdi.mipt.ru.

6. Апробация HJudge

Система использовалась на курсах:

- Хранение и обработка больших объёмов данных, весна 2016 г.
- Многопроцессорные вычислительные системы, осень 2016 г.

6. Апробация HJudge

- Кол-во ошибок, обнаруженных на каждом этапе

Тема ДЗ	Валидация сборки	Проверка запуска	Проверка результатов		
			Построчн.	Пофайл.	Полное
Hadoop Java API	3	11	26	31	2
Hadoop Streaming	1	5	21	28	0
Joins	2	7	25	13	1
Spark	4	32	211	114	102
Всего	10	55	273	186	105
%	1,6	8,7	43,4	29,6	16,7

6. Апробация HJudge

- Кол-во ошибок, обнаруженных на каждом этапе

Тема ДЗ	Валидация сборки	Проверка запуска	Проверка результатов		
			Построчн.	Пофайл.	Полное
Hadoop Java API	3	11	26	31	2
Hadoop Streaming	1	5	21	28	0
Joins	2	7	25	13	1
Spark	4	32	211	114	102
Всего	10	55	273	186	105
%	1,6	8,7	43,4	29,6	16,7

- Больше всего ошибок обнаруживается на проверке результатов
- На шаге проверки результатов больше всего ошибок отлавливается построчными и пофайловыми тестами

6. Апробация HJudge

- Замеры времени тестирования

Тема ДЗ	Время ручной проверки, с			Время проверки с HJudge, с		
	min	mean	max	min	mean	max
Hadoop Java API	50	100	150	8	16,5	25
Hadoop Streaming	50	80	110	3	5	7
Joins	40	120	200	5	8,5	12
Spark	2	24	46	1	2	3
В среднем	35,5	**81**	126,5	4,3	**8**	11,8

6. Апробация HJudge

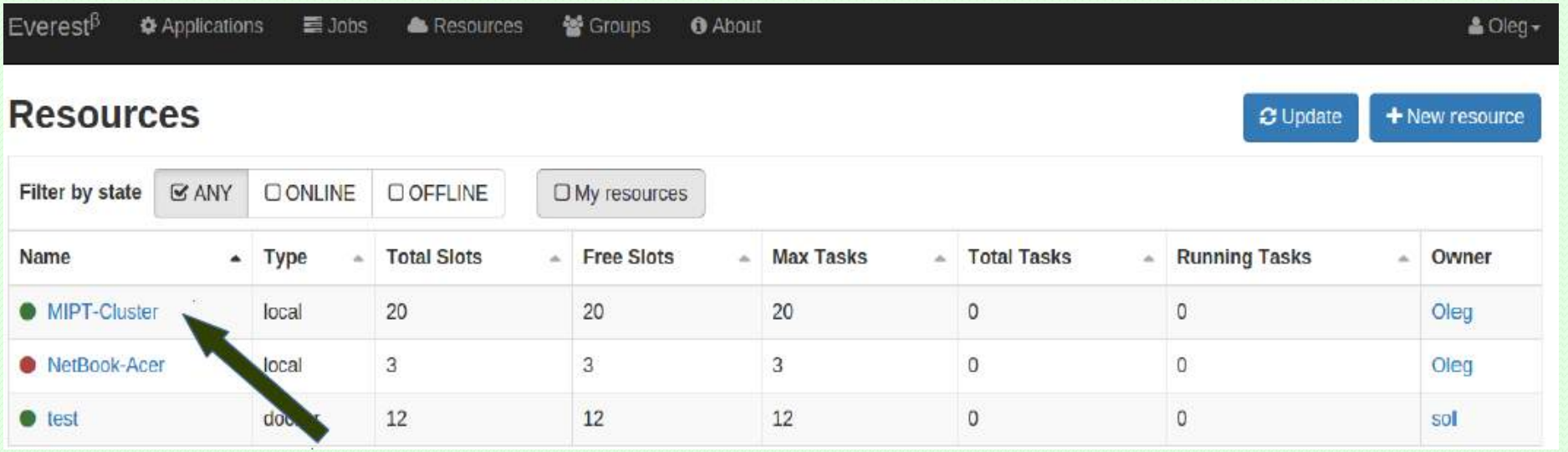
- Замеры времени тестирования

Тема ДЗ	Время ручной проверки, с			Время проверки с HJudge, с		
	min	mean	max	min	mean	max
Hadoop Java API	50	100	150	8	16,5	25
Hadoop Streaming	50	80	110	3	5	7
Joins	40	120	200	5	8,5	12
Spark	2	24	46	1	2	3
В среднем	35,5	**81**	126,5	4,3	**8**	11,8

- HJudge даёт выигрыш по времени **в 10 раз**

7. Планы по доработке

1. Интеграция с облачной платформой Everest
   - ❖ позволит пользоваться не только преподавателям, но и студентам, т.к. можно ограничить доступ к тестам



7. Планы по доработке

2. Интеграция с платформой Piazza
   - студенты уже пользуются платформой
   - при появлении нового сообщения с заданием, система будет получать из него данные автоматически
   - существует неофициальное Python API для взаимодействия с Piazza
     - http://hfaran.me/reverse-engineering-piazzas-api/

7. Планы по доработке

3. Возможность писать тестирующие функции не на Python

8. Вывод

1. Система HJudge реализована и продолжает развиваться
2. Система используется на курсах кафедры АТП и внедрена в учебный процесс

Федеральное государственное автономное образовательное
учреждение высшего образования
"Московский физико-технический институт
(государственный университет)"
Факультет инноваций и высоких технологий
Кафедра алгоритмов и технологий программирования

МФТИ

Институтский пер., 9, г. Долгопрудный, Московская обл., 141701
Тел.: +7 (495) 408-45-54. Факс: +7 (495) 408-68-69. E-mail: info@mipt.ru
ОКПО: 02068901 ОГРН: 1027739386135 ИНН/КПП: 5008006211/772701001
https://mipt.ru/

"УТВЕРЖДАЮ"
проректор по учебной работе и экономическому развитию
Зубцов Д.А.
"11" июля 2016 г.

АКТ

о внедрении результатов дипломной работы Ивченко О.Н.
"Тестирование распределенных приложений в экосистеме Hadoop",
представленной на соискание степени магистра при разработке НИР

Настоящим актом подтверждаем, что практические результаты дипломной работы Ивченко О.Н. "Тестирование распределенных приложений в экосистеме Hadoop", а именно программа для ЭВМ "HJudge" внедрена на кафедру АТП ФИВТ МФТИ. Программа для ЭВМ "HJudge" представляет собой систему тестирования приложений обработки больших данных и используется на кафедре в качестве проверяющей системы практических заданий по курсу "Хранение и обработка больших объёмов данных". Разработанная система позволяет автоматизировать тестирование сборки и запуска студенческих приложений, а также проверку их результатов.

В течение текущего семестра была проведена апробация HJudge на некотором множестве студенческих приложений. Апробация показала, что система позволяет экономить до 90% времени на проверку заданий, что позволяет улучшить качество контрольных заданий.

В настоящий момент подана заявка на государственную регистрацию программы на имя МФТИ. После осуществления государственной регистрации программы и получения свидетельства о государственной регистрации, программа будет поставлена на баланс МФТИ в качестве НМА.

Председатель комиссии: д.ф.-м.н. Ройтберг М.А.
Члены комиссии: д.ф.-м.н. Цитович И.И.
к.ф.-м.н. Горохов А.А.
Секретарь: Сокова Е.С.

Согласовано
Главный бухгалтер Т.П. Терещенко

8. Вывод

3. Система зарегистрирована в реестре программного обеспечения РФ.

РОССИЙСКАЯ ФЕДЕРАЦИЯ

СВИДЕТЕЛЬСТВО

о государственной регистрации программы для ЭВМ

№ 2016660616

HJudge

Правообладатель: *федеральное государственное автономное образовательное учреждение высшего образования «Московский физико-технический институт (государственный университет)» (RU)*

Авторы: *Ивченко Олег Николаевич (RU), Драль Алексей Александрович (RU)*

Заявка № 2016617861
Дата поступления 18 июля 2016 г.
Дата государственной регистрации в Реестре программ для ЭВМ *19 сентября 2016 г.*

Руководитель Федеральной службы по интеллектуальной собственности

Г.П. Ивлиев

Спасибо за внимание!

Вопросы?