ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

МОСКОВСКИЙ ИНСТИТУТ ЭЛЕКТРОНИКИ И МАТЕМАТИКИ

Домашняя работа
на тему:

Разработка программного кластера по технологии MapReduce

Студент гр. СВС101 _____ Собко С.С.

Преподаватель _____ Байбикова Т.Н.

Москва 2015

1 Постановка задачи

1.1 Описание алгоритма

MapReduce – это программная модель и ассоциированная с ней реализация для обработки и генерации больших объемов данных с использованием параллельного распределенного кластерного алгоритма.
Простейшая реализация алгоритма состоит из операции map, которая является функцией высшего порядка и применяет заданную функцию к каждому элементу входного списка и операции reduce, которая является функцией высшего порядка и производит свертку результирующего списка, приводя его к единственному атомарному значению.
Операция map может выполняться для любой части списка независимо и может быть распределена между нодами кластера. Операция reduce выполняется после получения всех результатов выполнения операций map на всех нодах кластера.

1.2 Описание задачи

Необходимо реализовать имплементацию алгоритма MapReduce на JVM-стеке.
Для реализации выбран язык программирования Clojure, включающий в себя все достоинства языка программирования LISP по обработке списков и построенный на стеке технологий виртуальной машины Java.

2 Реализация

2.1 Исходный код

В листинге 1 представлен исходный код описания проекта, включающий в себя описание проекта, зависимости и подключаемые пространства имен приложения.

Listing 1 — project.clj

```
(defproject smapreduce "0.0.1-SNAPSHOT"
  :description "Simple MapReduce implementation"
  :dependencies [[org.clojure/clojure "1.7.0"]
                 [org.clojure/tools.nrepl "0.2.11"]]
  :javac-options ["-target" "1.6" "-source" "1.6" "-Xlint:-options"]
  :aot [smapreduce.core
        smapreduce.server]
  :main smapreduce.core)
```

В листинге 2 описывается функция round-robin-map, распределяющая список значений full-list поочередно по списку нод distribute-list.

Listing 2 — src/smapreduce/utils.clj

```
(ns smapreduce.utils
  (:gen-class))

(defn round-robin-map
  [full-list distribute-list]
  (let [grouped-values
        (group-by first (mapv
                         #(list %1 %2)
                         (cycle distribute-list)
                         full-list))]
    (map (fn [map-args]
           (let [distributed-item (first map-args)
                 full-list-for-item (map second (second map-args))]
             [distributed-item full-list-for-item]))
         grouped-values)))
```

Листинг 3 описывает исходный код пространства имен smapreduce.core, который позволяет запустить приложение с аргументами map или reduce.

Listing 3 — src/smapreduce/core.clj

```
(ns smapreduce.core
  (:gen-class)
  (:require [smapreduce.client :as c]
            [smapreduce.server :as s]))

(defn run-map-or-reduce
  ([command & args]
    (println "Command:" command "for Arguments:" args)
    (case command
      "map" (s/repl-server (read-string (first args)))
      "reduce" (let
        [[reduce-function map-function list-values & worker-ports]
        (vec (map read-string args))]
                (let [full-list (c/get-full-list
                                 map-function
                                 list-values
                                 worker-ports)]
                  (println "Full list is" full-list)
                  (let [reduced-value (reduce (eval reduce-function) full-list)]
                    (println "Reduced value is" reduced-value))))))

  ([]
```

```
    (println "Use map port or reduce port arguments")))

(defn -main
  [& args]
  (apply run-map-or-reduce args))
```

В листинге 4 показан исходный код пространства имен smapreduce.server, позволяющий открыть сервер nREPL (встроенный интерпретатор Clojure, доступный с использованием сетевого соединения).

Listing 4 — src/smapreduce/server.clj

```
(ns smapreduce.server
  (:gen-class)
  (:require [clojure.tools.nrepl.server :as server]
            [clojure.tools.nrepl.middleware.pr-values :as pv]))

(defn repl-server
  [port]
  (println "Created REPL server on port" port)
  (server/start-server :port port))
```

Листинг 5 описывает пространство имен smapreduce.client, реализующее функцию map-client для соединения с сервером nREPL и функцию get-full-list, конкатенирующую списки, возвращаемые с каждой из нод кластера функцией map-client в результирующий список.

Listing 5 — src/smapreduce/client.clj

```
(ns smapreduce.client
  (:gen-class)
  (:require [clojure.tools.nrepl :as repl]
            [smapreduce.utils :as u]))

(defn map-client
  [port map-function list-values]
  (let [code (str (list 'map map-function (cons 'quote (list list-values))))]
    (println "Evaluating" code "on" port)
    (with-open [conn (repl/connect :port port)]
      (-> (repl/client conn 1000)
        (repl/message {:op "eval" :code code})
        (repl/response-values)))))

(defn get-full-list
  [map-function list-values worker-ports]
  (let [rr-map
        (u/round-robin-map list-values worker-ports)]
    (reduce
      concat
      (map (fn [map-args]
             (let [[worker-port worker-values-list] map-args]
               (let [ret-val (first (map-client worker-port
                                                map-function
                                                worker-values-list))]
                 (println "For" worker-port
                          "values" worker-values-list
                          "Returned list is" ret-val)
                 ret-val)))
           rr-map))))
```

2.2 Запуск программы

Программное обеспечение запускается следующим образом.
Для запуска двух нод на портах 1088 и 1099 нужно выполнить следующие команды из корня проекта (до этого должен быть установлен Clojure и пакетный менеджер Leiningen):

```
# Map REPL instance on port 1088
$ lein run map 1088
# Map REPL instance on port 1099
$ lein run map 1099
```

Для сложения числа «3» с каждым значением списка и сверткой списка через оператор сложения с использованием нод на портах 1088 и 1099 необходимо выполнить следующую команду:

```
# MapReduce client
$ lein run reduce "+" "#(+ 3 %)" "(1 2 3 4)" 1088 1099
```

3 Выводы

В ходе выполнения домашней работы был изучен язык программирования Clojure, методы создания системы распределенных вычислений, парадигма MapReduce.
Полученные знания были применены на практике при разработке программного обеспечения sMapReduce, имплементирующий простейшую реализацию алгоритмы.

Программное обеспечение можно получить из системы контроля версий GitHub по следующему адресу: `https://github.com/MIEMHSE/smapreduce`.