

УДК 004.021

## **АНАЛИЗ ЭФФЕКТИВНОСТИ АЛГОРИТМА MAPREDUCE ПРИ РЕШЕНИИ ЗАДАЧИ ВЫЧИСЛЕНИЯ СУММАРНОГО ВРЕМЕНИ, ЗАТРАЧЕННОГО СИСТЕМОЙ TRACKOR ONEVIZON, НА РАБОТУ С ПОЛЬЗОВАТЕЛЕМ ЗА МЕСЯЦ С ПОМОЩЬЮ APACHE HADOOP**

Диденко А.А., магистрант гр. ПИМ-171, II курс

Научный руководитель: Пимонов А.Г., д.т.н., профессор

Кузбасский государственный технический университет имени Т.Ф. Горбачева  
г. Кемерово

**Введение.** В современном обществе все чаще возникают проблемы, связанные с анализом большого объема данных, таких как логи транзакций, геоданных, статистики, а также использования интернет-сайтов и много другого. Для того чтобы решить данные проблемы компания Google разработала и выложила в открытый доступ алгоритм распределенных вычислений MapReduce [1], который лег в основу свободно распространяемого проекта Apache Software Foundation, который состоит из набора библиотек, утилит, а также фреймворка для разработки и выполнения распределённых программ, которые могут работать на кластерах, состоящих из сотен и тысяч узлов [2]. В данной работе представлены анализ и демонстрация работы данного алгоритма на примере решения задачи вычисления суммарного времени, затраченного системой Trackor Onevizon, на работу с пользователем за месяц с помощью Apache Hadoop на кластере, поднятом в виртуальной машине.

**Trackor Onevizon.** Платформа Trackor OneVizion представляет из себя веб-приложение, которое решает проблему управления сложноорганизованной информацией, состоящей из различного типа данных. Данная платформа интегрирует управление информацией, управление документами, прогнозирование, управление задачами, а также картографирование Google с целью предоставления полнофункционального программного решения для управления бизнесом [3]. Все изменения данной системы логируются в одном из восьми логов: ошибки помещаются в error\_log, действия пользователей системы в usage\_log, http запросы в и из системы в rule\_http\_call\_log, уведомления в notif\_processing\_log, интеграции с другими системами в integration\_log, изменения конфигурации системы в comp\_object\_log, информация о импорте и экспорте компонентов системы в bpl\_comp\_log, изменения данных в audit\_log. В ходе данной статьи будут рассмотрены только usage\_log, потому что только он потребуются для решения задачи, описанной в следующем пункте. Структура данного лога представлена на рис. 1. Далее будут перечислены колонки, представляющие интерес для решения задачи:

1. USAGE\_LOG\_ID – первичный ключ;

2. ACTION\_ID – ид действия, совершённого пользователем, например загрузка грида, определенного тракор типа, вход и выход из системы, открытие формы редактирования трактора;
3. BROWSER\_RUNTIME – время обработки запроса браузером пользователя в секундах;
4. DB\_RUNTIME – время обработки запроса базой данных в секундах;
5. IIS\_FINISH – время завершения действия, используется для определения даты и времени событий;
6. IIS\_RUNTIME – время генерации грида в секундах;
7. PAGE\_NAME – имя открываемой страницы;
8. RESPONSE\_DELIVERY\_TIME – время на отправку данных клиенту из базы данных в секундах;
9. UN – имя пользователя;
10. USER\_ID – ид пользователя;
11. USER\_RUNTIME – время с того момента, как данные были отправлены пользователю, и до момента, когда они были отображены в браузере в секундах.

USAGE_LOG					
COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
ACTION_ID	NUMBER	Yes	null	3	Action, possible values are "grid", "logon"
ACTION_STATUS_ID	NUMBER	Yes	null	32	0 - SUCCESS, 1 - FAIL (user request fails in case of DB or web server error)
BROWSER_RUNTIME	NUMBER	Yes	null	28	Time spent on client side (executing of js code and parsing data)
CLIENT_IP	VARCHAR2(39)	Yes	null	22	IP address of a client
COLS_COUNT	NUMBER	Yes	null	35	
DB_RUNTIME	NUMBER	Yes	null	15	Time DB spent for executing query and return result to IIS
ERROR_MESSAGE	VARCHAR2(1000)	Yes	null	33	
FILTER_NAME	VARCHAR2(200)	Yes	null	17	Filter name
FILTER_USER_ID	NUMBER	Yes	null	16	Filter owner
GATHER_SQL_STATS	NUMBER(1,0)	Yes	null	38	When 1 sql exec plan will be collected for user grid load action based on usage_log_id comment inside the SQL
GRID_PAGE_ID	NUMBER	Yes	null	31	
HIGHLIGHT_MODE	NUMBER	Yes	null	25	View Options Highlight Mode
IIS_FINISH	DATE	Yes	null	6	Grid generation finish time
IIS_RUNTIME	NUMBER	Yes	null	7	Grid generation time
IIS_START	DATE	Yes	current_date	5	Grid generation start time
IS_EDIT_MODE	NUMBER	Yes	null	26	Is Edit Mode On
LOGIN_AS_BY_USER_ID	NUMBER	Yes	null	37	
NOT_LIMITED_SQL	CLOB	Yes	null	30	SQL without grid paging limit
PAGE_NAME	VARCHAR2(255)	Yes	null	4	Name VisionEPM page
PROCESS_ID	NUMBER	Yes	null	36	Process id from Excel Report
PROGRAM_ID	NUMBER	No	null	14	Foreign key to "V_PROGRAM" table
RESPONSE_DELIVERY_TIME	NUMBER	Yes	null	27	Time spent on grid data transfer to client
ROWS_COUNT	NUMBER	Yes	null	34	Total Number of Rows in a Dataset
SERVER_IP	VARCHAR2(39)	Yes	null	21	IP address of web server
SQL	CLOB	Yes	null	11	SQL query used to generate data grid
TMP_REQUEST_TOKEN	VARCHAR2(16)	Yes	null	29	Token for tracking status of request on client
UN	VARCHAR2(50)	Yes	null	2	User name
URL	VARCHAR2(2000)	Yes	null	10	Page URL
USAGE_LOG_BROWSER_ID	NUMBER	Yes	null	20	Foreign key to "usage_log_new3_BROWSER"
USAGE_LOG_ID	NUMBER	Yes	null	1	Primary key
USER_ID	NUMBER	Yes	null	9	Foreign key to "USERS" table
USER_RUNTIME	NUMBER	Yes	null	8	Time since data sent to client and shown in browser
VIEW_NAME	VARCHAR2(200)	Yes	null	19	View Options name
VIEW_USER_ID	NUMBER	Yes	null	18	View Options owner
WS_REQUEST	CLOB	Yes	null	23	Body of WebService request for "Web Service" action
XITOR_TYPE_ID	NUMBER	Yes	null	12	Foreign key to "XITOR_TYPE" table
XML_SIZE	NUMBER	Yes	null	24	Size of XML/CSV file sent back to a client
ZONE_ID	NUMBER	Yes	null	13	Foreign key to "V_ZONE" table

Рис. 1. Структура USAGE\_LOG

**Постановка задачи.** Есть лог из 3.1 млн. транзакций, выгруженный из системы и представленный в формате: user\_id \t un \t action\_type \t page\_name \t iis\_finish \t iis\_runtime \t user\_runtime \t db\_runtime \t response\_delivery\_time \t browser\_runtime \n. Пример записей из лога представлен на рис. 2.

Необходимо вычислить суммарное время, затраченное системой Traskor Onevizon, на работу с пользователем за месяц.

Line	User	Action	Grid	Date	Value 1	Value 2	Value 3	Value 4	Value 5
3111613	100164596	rfishkin	Grid	20-MAR-19	2.290	2.586	1.888	0.196	0.100
3111614	100164596	rfishkin	Grid	20-MAR-19	2.029	2.436	1.667	0.249	0.158
3111615	100164596	rfishkin	Grid	20-MAR-19	0.101	0.300	0.009	0.173	0.026
3111616	100164596	rfishkin	Grid	20-MAR-19	0.066	0.208	0.006	0.128	0.014
3111617	100164596	rfishkin	Grid	20-MAR-19	2.389	2.804	1.997	0.269	0.146
3111618	100170756	erobinson	Logon	20-MAR-19	0.000	0.000	0.000	0.000	0.000
3111619	100170756	erobinson	Grid	20-MAR-19	0.138	0.640	0.071	0.492	0.010
3111620	100170756	erobinson	Grid	20-MAR-19	4.562	4.903	3.702	0.237	0.104
3111621	100165476	dhasselman	Grid	20-MAR-19	0.242	0.486	0.164	0.178	0.066
3111622	100165476	dhasselman	Grid	20-MAR-19	0.308	0.542	0.001	0.186	0.048
3111623	100165476	dhasselman	Grid	20-MAR-19	0.242	0.000	0.133	0.000	0.000
3111624	100165476	dhasselman	Grid	20-MAR-19	0.226	0.000	0.112	0.000	0.000
3111625	100165476	dhasselman	Grid	20-MAR-19	0.803	0.987	0.609	0.142	0.042
3111626	100165476	dhasselman	Grid	20-MAR-19	0.169	0.305	0.000	0.103	0.033
3111627	100165476	dhasselman	Grid	20-MAR-19	0.078	0.192	0.000	0.094	0.020
3111628	100165476	dhasselman	Grid	20-MAR-19	6.753	7.450	6.575	0.610	0.087
3111629	100165476	dhasselman	Grid	20-MAR-19	0.245	0.000	0.122	0.000	0.000
3111630	100165476	dhasselman	Grid	20-MAR-19	0.239	0.000	0.094	0.000	0.000
3111631	100165476	dhasselman	Grid	20-MAR-19	0.248	0.000	0.126	0.000	0.000
3111632	100165476	dhasselman	Grid	20-MAR-19	0.194	0.000	0.115	0.000	0.000
3111633	100165476	dhasselman	Grid	20-MAR-19	5.337	8.263	1.851	1.765	1.161
3111634	100165476	dhasselman	Grid	20-MAR-19	2.903	4.602	1.574	0.740	0.959
3111635	100170596	bcarroll	Grid	20-MAR-19	3.404	4.823	1.874	0.807	0.612
3111636	100164915	narledge	Grid	21-MAR-19	0.000	0.000	0.000	2.128	1.729
3111637	100169976	tranderson	Grid	21-MAR-19	2.735	6.545	1.443	2.868	0.942
3111638	100169976	tranderson	Grid	21-MAR-19	0.840	2.277	0.340	0.763	0.674
3111639	100170616	hnetties	Grid	21-MAR-19	0.746	1.065	0.229	0.367	0.052
3111640	100163934	mtran	Logon	22-MAR-19	0.000	0.000	0.000	0.000	0.000
3111641	100164755	rkasser	Grid	22-MAR-19	0.000	0.000	7.286	1.204	1.633
3111642	100170956	csipes	Grid	22-MAR-19	1.361	0.000	0.000	0.000	0.000
3111643	100170956	csipes	Grid	22-MAR-19	1.325	1.907	1.278	0.572	0.010
3111644	100166954	ejenkins	Grid	22-MAR-19	1.062	0.000	0.000	0.000	0.000
3111645	100166954	ejenkins	Grid	22-MAR-19	0.478	0.811	0.382	0.323	0.010
3111646	100166954	ejenkins	Grid	22-MAR-19	0.000	0.000	0.000	3.233	2.825
3111647	100164456	adatt	Logon	22-MAR-19	0.000	0.000	0.000	0.000	0.000
3111648	100164145	trunion	Grid	22-MAR-19	4.729	6.878	3.810	1.224	0.925
3111649	100164145	trunion	Grid	22-MAR-19	6.604	7.875	5.975	0.678	0.593
3111650	100164145	trunion	Grid	22-MAR-19	6.430	6.722	5.347	0.169	0.123
3111651	100164145	trunion	Grid	22-MAR-19	2.702	2.838	2.212	0.089	0.047
3111652	100164145	trunion	Grid	22-MAR-19	6.591	8.268	5.708	0.868	0.809
3111653	100167754	aderosia	Grid	22-MAR-19	2.228	3.909	0.982	0.463	1.218
3111654	100164199	beaton	Grid	22-MAR-19	3.981	6.420	2.190	1.447	0.992
3111655	100164199	beaton	Grid	22-MAR-19	2.369	3.742	0.461	0.505	0.868
3111656	100163934	mtran	Grid	22-MAR-19	0.585	0.692	0.384	0.078	0.029
3111657	100163934	mtran	Grid	22-MAR-19	0.190	0.310	0.027	0.093	0.027
3111658	100164239	jelmore	Grid	22-MAR-19	2.514	3.816	1.364	0.185	1.117
3111659	100164239	jelmore	Grid	22-MAR-19	1.510	3.674	0.677	0.572	1.552
3111660	100164239	jelmore	Grid	22-MAR-19	1.564	2.900	0.910	0.244	1.092
3111661	100164596	rfishkin	Grid	22-MAR-19	0.459	0.821	0.373	0.277	0.095
3111662									

Рис. 2. Лог действий пользователя

**Решение данной задачи** стандартными методами, предусмотренными в системе Traskor Onevizon, а именно SQL запросом к базе данных, показанном на рис. 3, заняло 7 минут и 54 секунд, это приемлемый результат для такого объема записей, однако его можно улучшить, используя MapReduce.

Test SQL:

```

1 select un, sum(nvl(iis_runtime, 0) + nvl(user_runtime, 0) + nvl(db_runtime, 0) + nvl(response_delivery_time, 0) + nvl(browser_runtime, 0))
2 from usage_log
3 where iis_finish >= to_date('01.01.2019', 'MM.DD.YYYY') and iis_finish <= to_date('01.31.2019', 'MM.DD.YYYY')
4 group by un

```

☐ No Data  Success

Рис. 3. Решение задачи стандартными методами

Теперь данную задачу, но уже используя алгоритм MapReduce. Данный алгоритм состоит из нескольких шагов [4]. Первый шаг – это применение функции Map к каждому элементу исходного лога. Функция возвращает либо ничего, либо создает объект, состоящий из пары значений Key/Value, в нашем случае Key – un / Value – sum(iis\_runtime + user\_runtime + db\_runtime + response\_delivery\_time + browser\_runtime). Примеры таких объектов: [adidenko/2.483], [adidenko/12.238], [rfishkin/12.493] и так далее. Следующим шагом алгоритм сортирует все объекты, полученные на шаге map, с помощью метода shuffle and sort, а также создает новые экземпляры объектов, где все значения (Value) будут сгруппированы по ключу. Примерами таких

объектов служат: [adidenko: 2.483, 12.238, ...], [rfishkin: 12.493, ...] и так далее. Заключительным шагом будет выполнение функции Reduce для каждого из сгенерированных объектов. С ее помощью будут просуммированы все элементы Value объекта, и результаты ее работы возвращены в результирующую коллекцию в формате Key: Result. Все три шага изображены на рис. 4. Примеры результирующих объектов: [adidenko: 14.721], [rfishkin: 12.493] и так далее.

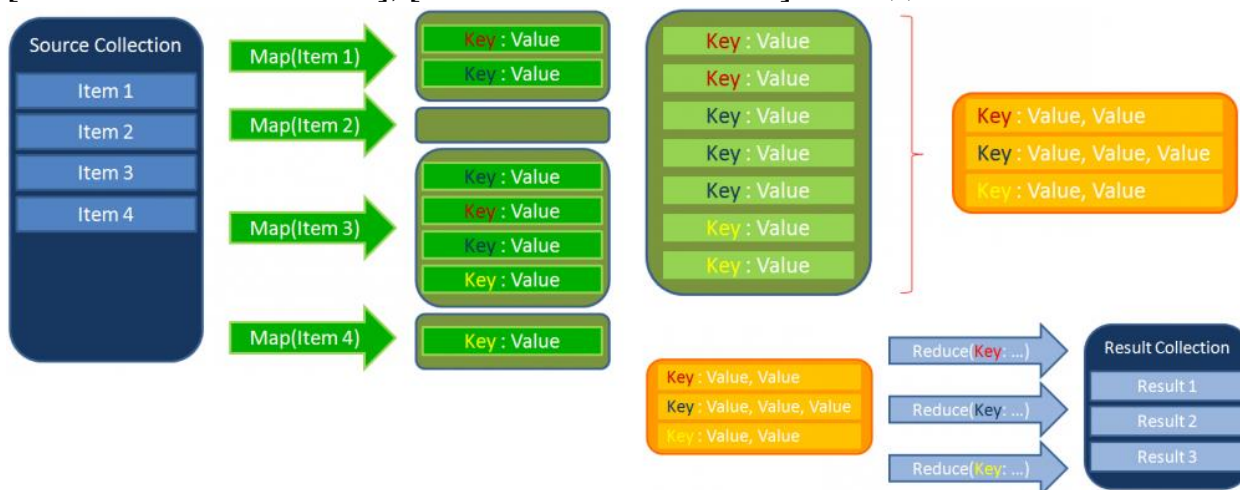


Рис. 4. Алгоритм MapReduce

Для реализации данного алгоритма был выбран язык программирования Python, а также был создан виртуальный кластер в программе Oracle VM VirtualBox.

Для решения данной задачи был написан код Reduce и 3 различных вариантов функции Map, а именно:

1) Mapper\_datetime, где происходит вычленение необходимых строк по дате iis\_finish и суммирование 5 значений iis\_runtime, user\_runtime, db\_runtime, response\_delivery\_time, browser\_runtime;

2) Mapper\_alt\_string, где определение строк проходит по части строки iis\_finish, а также суммирование 5 значений iis\_runtime, user\_runtime, db\_runtime, response\_delivery\_time, browser\_runtime;

3) Mapper\_2\_phases\_1 и Mapper\_2\_phases\_2, где процесс разбит на 2 фазы а) поиска строк с нужными датами iis\_finish и б) суммирования 5 значений iis\_runtime, user\_runtime, db\_runtime, response\_delivery\_time, browser\_runtime.

Код данных функций представлен на рис. 5, 6, 7, 8. Время выполнения и результаты решения задачи всеми 3 способами представлены на рис. 9.

```

reducer.py x test_data(1).txt x mapper_total_runtime.py x mapper_iis_runtime_date.py x
#!/usr/bin/python

import sys

valuesTotal = 0
oldKey = None

# Loop around the data
# It will be in the format key\tval
# Where key is user name or category of user log action, val is the time in seconds

for line in sys.stdin:
    data_mapped = line.strip().split("\t")
    if len(data_mapped) != 2:
        # Something has gone wrong. Skip this line.
        continue

    thisKey, thisValue = data_mapped

    if oldKey and oldKey != thisKey:
        print oldKey, "\t", valuesTotal
        oldKey = thisKey;
        valuesTotal = 0

    oldKey = thisKey
    valuesTotal += float(thisValue)

if oldKey != None:
    print oldKey, "\t", valuesTotal
  
```

*Рис. 5. Reducer.py*

```

#!/usr/bin/python

# Format of each line is:
# user_id, un, action_type, page_name, date, iis_runtime, user_runtime, db_runtime, response_delivery_time,
# browser_runtime
#
# We want elements 2 (user name) and 6, 7, 8, 9, 10 (iis_runtime, user_runtime, db_runtime,
# response_delivery_time, browser_runtime) and 5 (date)
# We need to write them out to standard output, separated by a tab

import sys
from datetime import datetime

for line in sys.stdin:
    data = line.strip().split("\t")
    if len(data) == 10:
        user_id, un, action_type, page_name, date, iis_runtime, user_runtime, db_runtime,
        response_delivery_time, browser_runtime = data
        if len(date) > 0:
            datec = datetime.strptime(date, '%d-%b-%y')
            datemin = datetime.strptime('01-JAN-19', '%d-%b-%y')
            datemax = datetime.strptime('01-FEB-19', '%d-%b-%y')
            total_runtime = float(iis_runtime) + float(user_runtime) + float(db_runtime) + float
            (response_delivery_time) + float(browser_runtime)
            if datemin <= datec < datemax:
                print "{0}\t{1}".format(un, total_runtime)
  
```

*Рис. 6. Mapper\_datetime.py*

```
#!/usr/bin/python

# Format of each line is:
# user_id, un, action_type, page_name, date, iis_runtime, user_runtime, db_runtime, response_delivery_time,
# browser_runtime
#
# We want elements 2 (user name) and 6, 7, 8, 9, 10 (iis_runtime, user_runtime, db_runtime,
# response_delivery_time, browser_runtime) and 5 (date)
# We need to write them out to standard output, separated by a tab

import sys

for line in sys.stdin:
    data = line.strip().split("\t")
    if len(data) == 10:
        user_id, un, action_type, page_name, date, iis_runtime, user_runtime, db_runtime,
        response_delivery_time, browser_runtime = data
        if len(date) > 0 and date[3:] == 'JAN-19':
            total_runtime = float(iis_runtime) + float(user_runtime) + float(db_runtime) + float
            (response_delivery_time) + float(browser_runtime)
            print "{0}\t{1}".format(un, total_runtime)
```

*Рис. 7. Mapper\_alt\_string.py*

```
#!/usr/bin/python

# Format of each line is:
# user_id, un, action_type, page_name, date, iis_runtime, user_runtime, db_runtime, response_delivery_time,
# browser_runtime
#
# We want elements 2 (user name) and 6, 7, 8, 9, 10 (iis_runtime, user_runtime, db_runtime,
# response_delivery_time, browser_runtime) and 5 (date)
# We need to write them out to standard output, separated by a tab

import sys

for line in sys.stdin:
    data = line.strip().split("\t")
    if len(data) == 10:
        user_id, un, action_type, page_name, date, iis_runtime, user_runtime, db_runtime,
        response_delivery_time, browser_runtime = data
        if len(date) > 0 and date[3:] == 'JAN-19':
            print "{0}\t{1}\t{2}\t{3}\t{4}\t{5}".format(un, iis_runtime, user_runtime, db_runtime,
            response_delivery_time, browser_runtime)

#!/usr/bin/python

# Format of each line is:
# un, iis_runtime, user_runtime, db_runtime, response_delivery_time, browser_runtime
#
# We want all elements
# We need to write them out to standard output, separated by a tab

import sys

for line in sys.stdin:
    data = line.strip().split("\t")
    if len(data) == 6:
        un, iis_runtime, user_runtime, db_runtime, response_delivery_time, browser_runtime = data
        total_runtime = float(iis_runtime) + float(user_runtime) + float(db_runtime) + float
        (response_delivery_time) + float(browser_runtime)
        print "{0}\t{1}".format(un, total_runtime)
```

*Рис. 8. Mapper\_2\_phases\_1.py и Mapper\_2\_phases\_2.py*



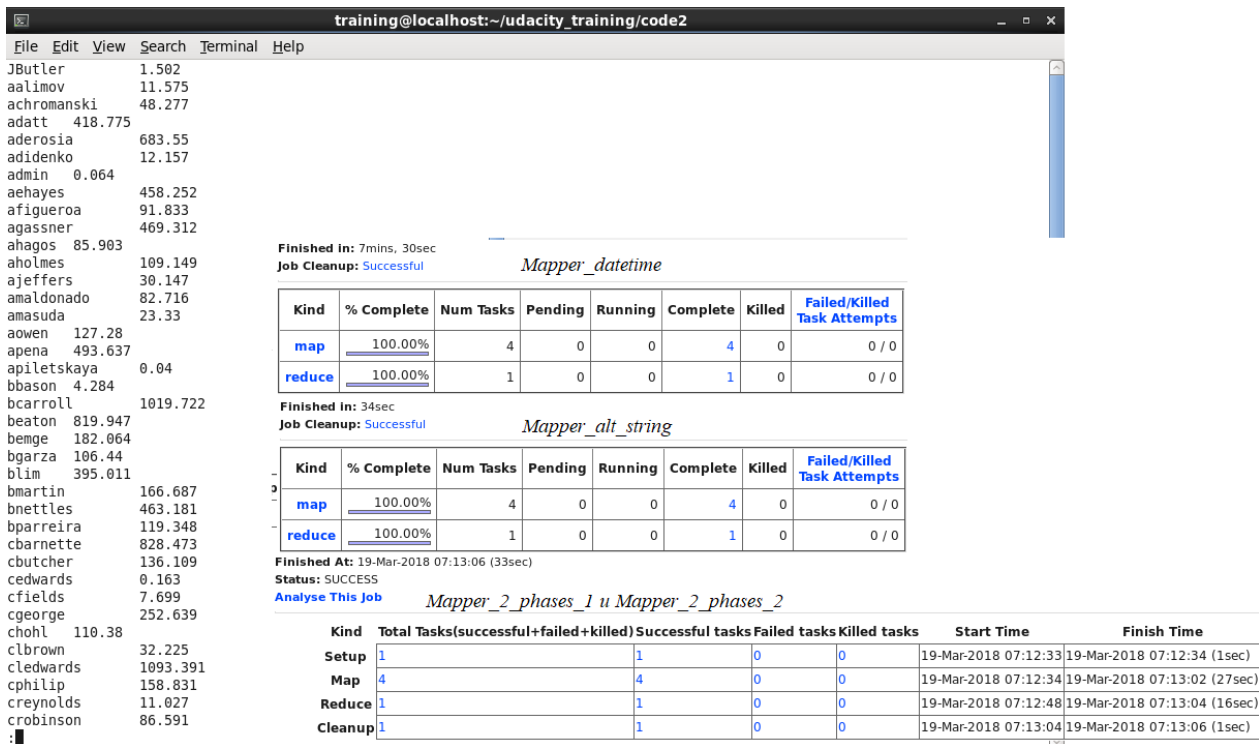


Рис. 9. Результаты решения задачи и время выполнения 3 различных подходов

Из статистики решения видно, что для решения каждого из вариантов задачи было создано четыре параллельных процесса map и один – reduce. Самым эффективным по времени выполнения маппером себя показал двухфазовый с 33 секундами, затем идет Mapper\_alt\_string с 34 секундами, и наименее эффективным себя показал Mapper\_datetime с 7 минутами и 30 секундами. Такие плохие показатели у последней функции вызваны конвертациями строк в даты, поэтому без острой необходимости такие операции лучше не использовать.

**Заключение.** В данной статье был рассмотрен алгоритм MapReduce на примере решения задачи вычисления суммарного времени, затраченного системой Trackor Onevizon, на работу с пользователем за месяц с помощью Apache Hadoop на кластере, поднятом в виртуальной машине. В результате было наглядно продемонстрировано, что для данного лога и конкретного компьютера алгоритм MapReduce в самой оптимальной конфигурации в 15 раз эффективнее по времени, чем стандартное решение, написанное на SQL, но при использовании в маппере функций конвертаций строк в даты время выполнения задачи с помощью алгоритма MapReduce сравнимо со стандартным решением.

**Список литературы:**

1. Википедия MapReduce [Электронный ресурс]. – Режим доступа: <https://ru.wikipedia.org/wiki/MapReduce>, свободный (дата обращения: 27.03.2019).
2. Википедия Hadoop [Электронный ресурс]. – Режим доступа: <https://ru.wikipedia.org/wiki/Hadoop>, свободный (дата обращения: 27.03.2019).
3. Графический компонент создания и редактирования пользовательских форм для платформы Onevizion [Электронный ресурс]. – Режим доступа: <http://science.kuzstu.ru/wp-content/Events/Conference/RM/2018/RM18/pages/Articles/31547-.pdf>, свободный (дата обращения: 27.03.2019).
4. Что такое MapReduce? // Статья блога о разработке, программировании на C#/.NET, и не только [Электронный ресурс]. – Режим доступа: <http://regfordev.blogspot.ru/2015/09/mapreduce.html#.WrQJDOjFKUm>, свободный (дата обращения: 27.03.2019).