DOI: 10.5862/JCSTCS/3
УДК 004.415

Л.Ю. Лабошин, А.А. Лукашин, В.С. Заборовский

ПРИМЕНЕНИЕ ТЕХНОЛОГИИ MAPREDUCE ДЛЯ КОНТРОЛЯ ДОСТУПА К ИНФОРМАЦИОННЫМ РЕСУРСАМ В КОРПОРАТИВНЫХ СЕТЯХ

L.Yu. Laboshin, A.A. Lukashin, V.S. Zaborovsky

APPLYING MAPREDUCE AND NETWORK TRAFFIC ANALYSIS TO CONTROL ACCESS TO INFORMATION RESOURCES

Обработка информации в настоящее время является одной из наиболее актуальных задач. С ростом и развитием информационных и телекоммуникационных технологий выросли и объемы передаваемой информации по сети Интернет. Одновременно с обработкой информации встает вопрос ее защиты.

Предложен подход к построению распределенной вычислительной системы, осуществляющей обработку снимков сетевого трафика за приемлемое время и обеспечивающей близкий к линейному рост производительности при наращивании вычислительных мощностей.

АНАЛИЗ СЕТЕВОГО ТРАФИКА; БОЛЬШИЕ ДАННЫЕ; MAPREDUCE; HADOOP.

Nowadays information security is an important issue. Network traffic analysis is widely used by Internet Service Providers to evaluate network performance, to collect statistics and to detect vulnerabilities. To analyze traffic traces collected from a large network it is required a computer system where both storage and computing resources can be easily scaled out to handle and process multi-Terabyte files. Cloud computing platforms and cluster file systems could provide resizable compute and storage capacity. The MapReduce programming model developed by Google in 2004 allows processing huge amounts of data in distributed manner by defining the map and reduce functions. The given paper proposes a cloud-computing framework based on a MapReduce approach for fast internet traffic analytics.

DEEP PACKET INSPECTION; BIGDATA; MAPREDUCE; HADOOP.

Обработка информации – одна из наиболее актуальных задач. С лавинообразным ростом и развитием информационных и телекоммуникационных технологий увеличиваются и объемы передаваемой информации по сети Интернет. Одновременно с обработкой информации встает вопрос ее защиты [1]. Информационная безопасность современных вычислительных систем является одной из приоритетных задач, сформулированных правительством РФ. В соответствии с приказом Минкомсвязи, который обязует интернет-провайдеров предоставлять снимки сетевого трафика за последние 12 часов с 1 июня 2014 г., важность оперативной обработки таких снимков является неоспоримой. В связи с вышесказанным, оперативный анализ и обработка сетевого трафика – очень важная и актуальная проблема, требующая решения.

Задача обработки сверхбольших снимков трафика относится к задачам класса «больших данных». При пропускной способности сетевого канала в 10 Гбит/с за 12 ч будет передано в одну сторону более 50 ТБ информации. Как правило, магистральный провайдер располагает каналами в сотни гигабит. Но уже при десятигигабитном канале требуется обработка порядка сотни терабайт. Обработка такого объема информации относится к задаче больших данных и не может быть произведена одним сервером за приемлемое время.

Предлагаемым в данной статье решением проблемы является организация об-

лачной среды, вычислительные ресурсы которой могут масштабироваться до тысяч виртуальных серверов, и создание на ее базе распределенной вычислительной системы, осуществляющей обработку снимков сетевого трафика за приемлемое время и обеспечивающей близкий к линейному рост производительности при наращивании вычислительных мощностей.

Перспективный путь повышения эффективности использования вычислительных ресурсов и уменьшения времени решения задач обработки больших данных – совместное применение технологий облачных вычислений, гибридных вычислительных архитектур и функциональных средств программирования.

Применение модели MapReduce в задачах анализа сетевого трафика

Существует большое количество инструментов, таких как, например, анализатор сетевых протоколов wireshark и CoralReef, позволяющих производить мониторинг и анализ сетевого трафика. Однако большинство таких инструментов ориентировано на использование на одном высокопроизводительном сервере, что делает невозможным их использование для обработки больших объемов трафика, полученных на высокоскоростных каналах связи. Yeonhee Lee [8, 9] предложена система анализа сетевого трафика на базе Apache Hadoop, ограниченная только сбором статистической информации. Библиотека RIPE не использует возможности параллельного чтения и записи в распределенную файловую систему, что серьезно сказывается на общей производительности системы.

Модель обработки данных MapReduce, разработанная компанией Google в 2004 г., позволяет параллельно обрабатывать данные большого объема путем задания функций Map и Reduce [2]. Вычислительные элементы, реализующие эти функции, называются, соответственно, маппер (mapper) и редьюсер (reducer). Одновременно над одной задачей может работать большое количество мапперов и редьюсеров, распределенных по узлам кластера. Данные для обработки с помощью MapReduce должны быть представлены в формате ключ – значение $<k; v>$.

Весь объем входных данных разбивается на фрагменты определенного размера или блоки, каждый такой блок поступает на вход одному из мапперов. Входящая пара $<k_{in};v_{in}>$ преобразуется в промежуточную пару $<k_{int};v_{int}>$. Затем промежуточные данные, полученные со всех мапперов, группируются по ключу k_{int} и поступают на вход редьюсерам в виде $<k_{int};list\ v^{i}_{int}>$. Таким образом, значения, соответствующие одному ключу, попадают в один редьюсер. После окончательной обработки на выходе редьюсера получаем пары $<k_{out};v_{out}>$, которые записываются в выходной файл. Схематично работа MapReduce-задачи представлена на рис. 1.

В современных системах обработки больших данных, таких как Apache Hadoop, хранение обрабатываемых данных осуществляется в распределенных файловых системах, например, Hadoop File System (HDFS). При этом обеспечивается очень важный принцип локальности обработки данных: процессы mapper и reducer запускаются на тех узлах кластера, где находятся обрабатываемые данные, что сокращает время доступа к распределенному хранилищу.

Предлагаемым путем решения задачи параллельного анализа сетевого трафика в распределенной облачной среде является применение парадигмы MapReduce совместно с функциональными технологиями программирования, основанными на неизменяемых структурах данных, что позволяет распараллелить обработку информации без блокировок и операций синхронизации.

Для анализа снимка сетевого трафика

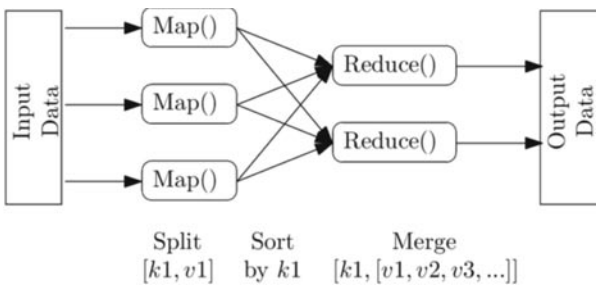

Рис. 1. Функции Map и Reduce

необходимо перенести файлы с трафиком в параллельную файловую систему и запустить операции Map/Reduce. При этом масштабирование скорости обработки снимка будет реализовано за счет наращивания количества узлов кластера.

Производительность каждой отдельно взятой задачи зависит от характеристик кластера, на котором она выполняется. Распределение задач между ядрами CPU или вычислителями на базе GPU представляется нетривиальной задачей. Масштабируемая платформа анализа данных Spark, которая включает в себя примитивы для вычислений в оперативной памяти и, следовательно, обладает некоторыми преимуществами в части производительности по отношению к подходу Hadoop, основанному на кластерной схеме хранения данных [7]. Spark реализован на мультипарадигменном языке программирования Scala и поддерживает этот язык, который обеспечивает уникальную среду для обработки данных.

Методика обработки снимков сетевого трафика для полного анализа передаваемых данных

Существует множество форматов для сохранения снимков сетевого трафика, однако одним из самых распространенных форматов является PCAP (Packet CAPture) [3]. Библиотека libpcap является платформонезависимой библиотекой с открытым исходным кодом (версия для Windows носит название winpcap). PCAP используется в качестве основного формата такими программными средствами, как tcpdump, wireshark, ngrep, wnort, nmap, и является де факто стандартом для захвата и анализа сетевых пакетов данных. Libpcap – двоичный файл, состоящий из глобального заголовка, позволяющего его идентифицировать, и записей для каждого захваченного пакета (рис. 2). Для хранения и анализа трафика в разрабатываемой системе было решено использовать формат libpcap.

Программы, построенные на программной модели MapReduce, в большинстве своем используются для обработки больших объемов текстовых файлов, таких как лог-файлы или веб-страницы, поэтому основной формат входных файлов текстовый. Каждая строка интерпретируется как пара $<k_{in};v_{in}>$: ключ – это смещение от начала файла, значение – содержимое строки. Двоичный формат PCAP не имеет отметок между пакетами, таких как символ перевода строки в текстовых документах. Так как исходный файл разделяется на блоки фиксированной длины, пакетная запись в файле часто находится в двух соседних блоках. Длина записи также варьируется от пакета к пакету, что осложняет их выявление в пределах блока распределенной файловой системы. Следовательно, для корректной работы системы (рис. 3) необходимо разработать надежный алгоритм определения начала пакетной записи в блоке.

Размер кадра в сети Ethernet обычно находится в диапазоне 64–1518 байт (за исключением Jumbo фреймов, размер которых может достигать нескольких килобайт). В заголовке пакетной записи PCAP файла присутствуют два двухбайтовых поля: первое содержит длину пакетной записи, второе – длину пакета в сети. При стандартных настройках захвата сетевого трафика максимальная длина пакетной записи 65 535 байт, и длина пакета в сети никогда не превышает этого значения. Следовательно, в соответствующих полях пакетной записи будут находиться равные значения.

В данной статье предложен эвристический алгоритм, в котором эта особенность используется как своеобразная метка начала пакетной записи. Обработка блока входных данных начинается с поиска двух равных, идущих подряд, двухбайтовых полей, содержащих значение, соответствующее допустимому размеру кадра в сети Ethernet. Когда такие поля найдены, определяется предполагаемое место начала пакетной записи и производится дополнительная проверка его корректности, например, путем сравнения

Рис. 2. Структура файла PCAP

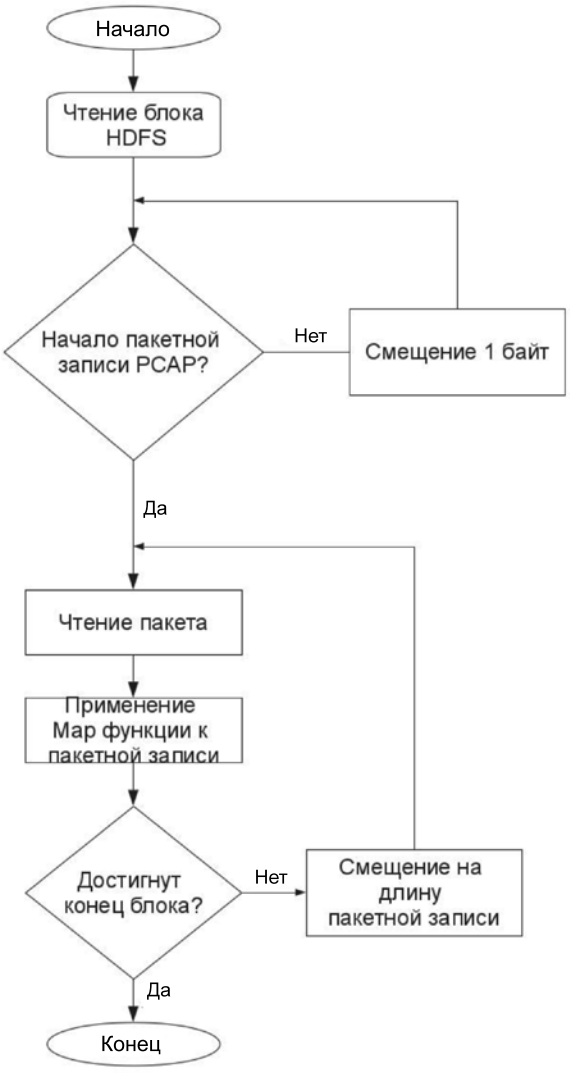

Рис. 3. Процедура обработки блока файловой системы

поля Ethertype с допустимыми значениями (IEEE 802.3) [4]. Подобная проверка выполняется для каждой обнаруженной пакетной записи, что исключает ложное обнаружение начала кадра. После определения первой пакетной записи поиск следующей осуществляется схожим образом с того места, где закончилась предыдущая (рис. 4).

Разработанный алгоритм дает возможность разбить снимок сетевого трафика на блоки, которые содержат только полные сессии (TCP и UDP), что позволяет осуществить анализ всего контекста сетевого соединения.

Предложенный подход позволяет решить задачи, которые можно разбить на два класса.

1. Анализ статистических данных сетевого трафика на базе MapReduce. Используя описанный выше метод чтения пакетных записей в пределах блока с помощью задания простых функций подсчета, становится возможным извлечение статистической информации из файла снимка сетевого трафика, вплоть до транспортного уровня модели OSI. Результаты анализа на данном уровне предоставляют следующую информацию:

общее количество трафика между отдельными подсетями (байтов, пакетов, соединений);

общее количество локального трафика;

выявление трафика, нарушающего политику доступа.

Предложенный подход позволяет решить следующие практические задачи:

- выявление наличия вирусного трафика в сети (аномально большое количество входящего и исходящего трафика);
- выявление DDoS атак (аномально большое количество трафика, сгенерированного почтовыми или DNS серверами);
- доступ к запрещенным IP адресам.

2. Полный анализ содержимого сетевого трафика. Полный анализ пакетов сетевого трафика (deep packet inspection) позволяет сопоставлять поток трафика с базой известных паттернов, известных как сигнатуры. Сигнатуры являются представлением потенциально вредоносного содержимого в виде строки символов или определенной последовательности байтов. В отличие от рассмотренных выше задач подсчета, которые могут быть выполнены на IP уровне, такой анализ не может быть проведен для отдельно взятого блока в файловой системе, а требует построения отдельных потоков данных, формируемых сетевыми приложениями в рамках информационного взаимодействия. Такие потоки, называемые *виртуальными соединениями*, как некоторая абстракция, существуют параллельно друг от друга, при этом не имеют между собой разделяемых ресурсов, что позволяет осуществлять их параллельную обработку [5].

Для построения виртуального соединения, при обработке каждого блока с помощью mapper для каждого пакета задается

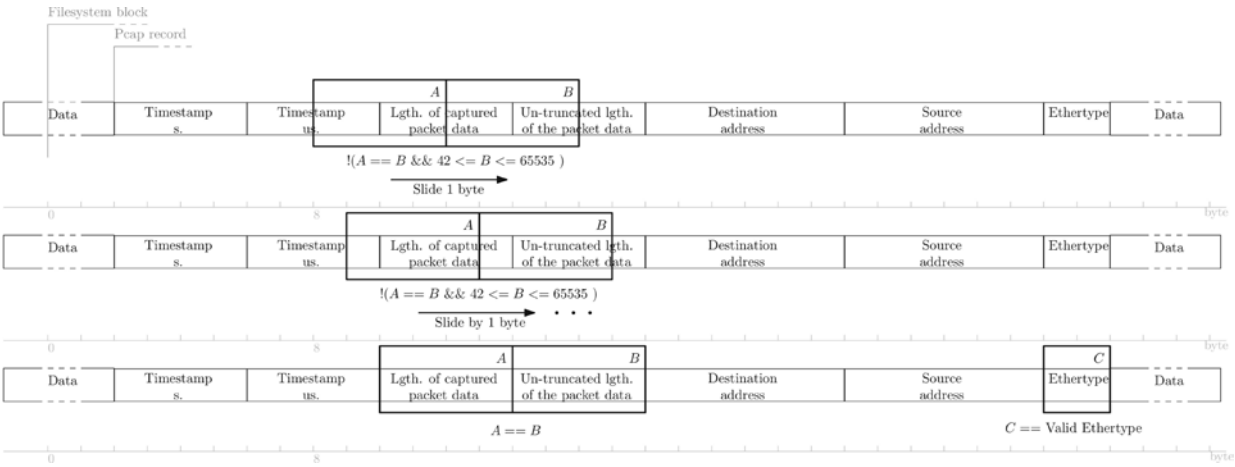

Рис. 4. Алгоритм выявления начала пакетной записи в блоке распределенной файловой системы

ключ, построенный на основании некоторой коммутативной операции над хешами адресами отправителя и получателя, такой что *F*(*src*, *dst*, *port*)==*F*(*dst*, *src*, *port*). На основании значения этого ключа на стадии reduce собирается выходной файл, содержащий в себе одно независимое виртуальное соединение. Над такими файлами может быть произведен детальный анализ соответствующего протокола, например, заголовков и содержимого HTTP – трафика.

Архитектура системы анализа сетевого трафика в магистральных сетях

Предлагаемый в статье подход можно применить для анализа сетевого трафика в магистральных сетях крупных провайдеров. Нами были разработаны следующие требования к системе анализа трафика:

- индексация трафика, сгенерированного сетью за 30 мин, должна занимать 15–20 мин работы кластера (время обработки может быть уменьшено за счет масштабирования системы);
- время поиска по проиндексированным данным суточного трафика за 5–30 мин в зависимости от типа поискового запроса;
- оперативное хранилище сетевого трафика за последние 12 ч с емкостью от 150 ТБ;
- возможность подключения долговременного хранилища для сохраненных снимков сетевого трафика.

Для реализации разработанных требований необходимо осуществлять оперативную индексацию данных и ведение учета обработанных снимков в локальной базе данных. Кроме того, система должна предусматривать очистку хранилища, если в снимках не было найдено важной информации. Структура системы представлена на рис. 5.

Трафик поступает с сетевого оборудования (например, за счет зеркалирования трафика в коммутаторах) в подсистему загрузки снимков в хранилище, которая аккумулирует данные за временное окно и загружает файл на обработку и хранение в параллельную файловую систему. Кластер Map/Reduce осуществляет анализ и индексацию содержимого сетевого трафика. Управление системой осуществляется через веб-интерфейс, предоставляющий возможность администрирования комплекса (в т. ч. масштабирования), задания поисковых запросов и визуализации результатов поиска.

Необходимо отметить, что существуют системы для обработки больших данных, поставляемые компаниями IBM и Teradata. Однако стоимость таких систем высока, для анализа трафика требуется их значительная доработка. Рассматриваемая в статье система имеет следующие преимущества перед существующими аналогами:

- масштабирование производительности и объема хранилища путем простого добавления серверов в кластер;
- применение открытых программных разработок;
- возможность установки на разные ап-

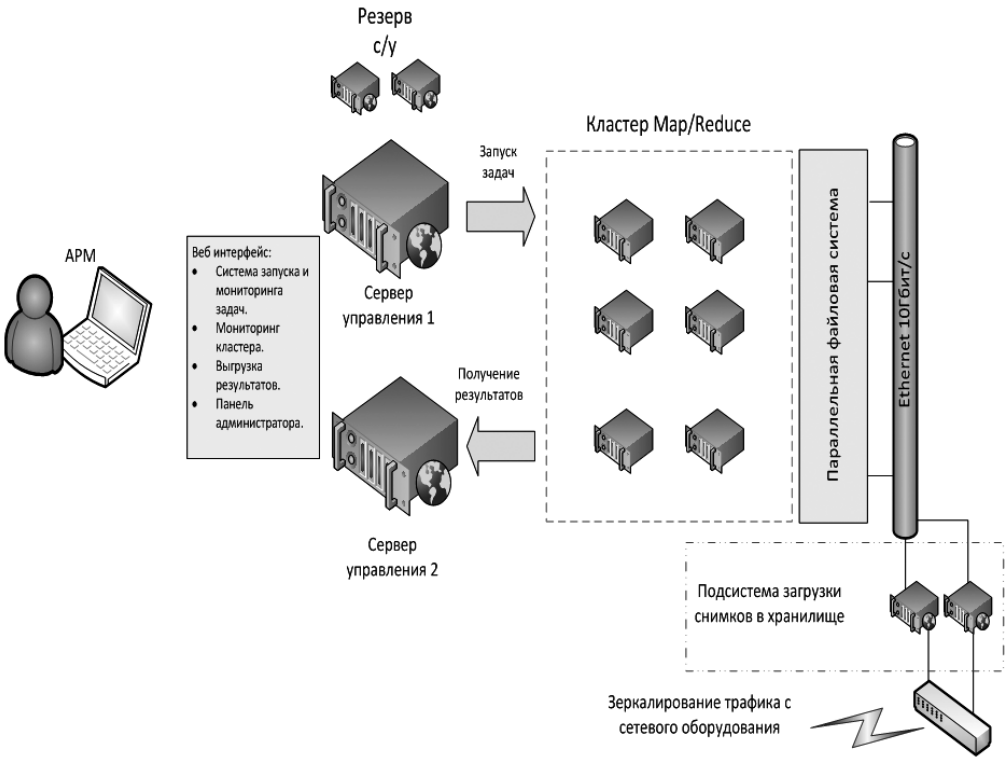

Рис. 5. Структура системы распределенного анализа снимков сетевого трафика

паратные платформы, что позволяет обеспечить использование сертифицированного оборудования без «закладок»;

• более низкая стоимость комплекса, даже с учетом затрат на разработку ПО.

Для масштабирования производительности системы анализа снимков сетевого трафика применяется облачная платформа типа инфраструктуры как сервис «Пилигрим» [6].

В статье предложен подход к анализу снимков сетевого трафика, основанный на парадигме MapReduce. В отличие от существующих решений, предполагающих произведение операций анализа на одном высокопроизводительном сервере, такой подход позволяет обеспечить автоматическое распараллеливание и хранение данных на внутренних дисках узлов кластера и может использоваться для построения вычислительной платформы, ресурсы которой могут масштабироваться в зависимости от объема входных данных для обработки сверхбольших файлов архивов сетевого трафика современных вычислительных сетей. Распределенная файловая система позволяет хранить данные большого объема на дисках серверов стандартной архитектуры, без необходимости установки дорогостоящей системы хранения.

СПИСОК ЛИТЕРАТУРЫ

1. **Федотов Н.Н.** Форензика – компьютерная криминалистика. М.: Юридический мир, 2007. С. 37–50.

2. **Dean J., Ghemawat S.** MapReduce: simplified data processing on large clusters //Communications of the ACM. 2008. Vol. 51. No. 1. Pp. 107–113.

3. **Garcia L.M.** Tcpdump and Libpcap [Электронный ресурс]/ URL: http://www. tcpdump. org/ (Дата обращения 26.08.2010).

4. **Eastlake D., Romascanu D.** IANA IEEE 802 Numbers [Электронный ресурс]/ URL: http:// www. iana. org/assignments/ieee-802-numbers/ ieee-802-numbers.xhtml/ (Дата обращения 26.08.2010).

5. **Заборовский В.С. и др.** Архитектура системы разграничения доступа к ресурсам гетерогенной вычислительной среды на основе контроля виртуальных соединений //Вестник

УГАТУ. 2013. Т. 15. № 5(45). С. 170–174.

6. **Заборовский В., Лукашин А.** Высокопроизводительная защищенная облачная среда // Открытые системы. СУБД. 2013. № 6. С. 10–13. [Электронный ресурс]/ URL: http://www.osp.ru/os/2013/06/13036845 (Дата обращения 12.11.2014).

7. **Kumawat T., Sharma P.K., Verma D., Joshi K., Kumawat V.** Implementation of spark cluster technique with scala // International Journal of Scientific and Research Publications. 2012. No. 2(11).

8. **Lee Y., Kang W., Lee Y.** A hadoop-based packet trace processing tool // In TMA. 2011. Pp. 51–63.

9. **Lee Y., Lee Y.** Toward scalable internet traffic measurement and analysis with Hadoop // Computer Communication Review. 2013. No. 43(1). Pp. 5–13.

REFERENCES

1. **Fedotov N.N.** *Forenzika – kompyuternaya kriminalistika* [*Forensic Computer Forensics*]. Moscow: Yuridicheskiy mir Publ., 2007, Pp. 37–50. (rus)

2. **Dean J., Ghemawat S.** MapReduce: simplified data processing on large clusters, *Communications of the ACM*, 2008, Vol. 51, No. 1, Pp. 107-113.

3. **Garcia L.M.** *Tcpdump and Libpcap*. Available: http://www. tcpdump. org/ (Accessed 26.08.2010).

4. **Eastlake D., Romascanu D.** *IANA IEEE 802 Numbers*. Available: http://www. iana. org/assignments/ieee-802-numbers/ieee-802-numbers.xhtml/ (Accessed 26.08.2010).

5. **Zaborovskiy V.S. et al.** Arkhitektura sistemy razgranicheniya dostupa k resursam geterogennoy vychislitelnoy sredy na osnove kontrolya virtualnykh soyedineniy [The system architecture is restricting access to the resources of a heterogeneous computing environment based on the control of virtual connections]. *Vestnik UGATU* [*Herald UGATU*], 2013, Vol. 15, No. 5 (45), Pp. 170–174. (rus)

6. **Zaborovskiy V., Lukashin A.** Vysokoproizvoditelnaya zashchishchennaya oblachnaya sreda [High secure Cloud], *Otkrytyye sistemy. SUBD* [*Open Systems. DBMS*], 2013, No. 6, Pp. 10–13. Available: http://www.osp.ru/os/2013/06/13036845 (Accessed 12.11.2014). (rus)

7. **Kumawat T., Sharma P.K., Verma D., Joshi K., Kumawat V.** Implementation of spark cluster technique with scala, *International Journal of Scientific and Research Publications,* 2012, No. 2(11).

8. **Lee Y., Kang W., Lee Y.** A hadoop-based packet trace processing tool, *In TMA,* 2011, Pp. 51–63.

9. **Lee Y., Lee Y.** Toward scalable internet traffic measurement and analysis with hadoop, *Computer Communication Review,* 2013, No. 43(1), Pp. 5–13.

ЛАБОШИН Леонид Юрьевич – *аспирант кафедры телематики Санкт-Петербургского политехнического университета Петра Великого.*
195251, Россия, Санкт-Петербург, ул. Политехническая, д. 29.
E-mail: laboshinl@neva.ru

LABOSHIN Leonid Yu. *Peter the Great St. Petersburg Polytechnic University.*
195251, Politekhnicheskaya Str. 29, St. Petersburg, Russia.
E-mail: laboshinl@neva.ru

ЛУКАШИН Алексей Андреевич – *доцент кафедры телематики Санкт-Петербургского политехнического университета Петра Великого, кандидат технических наук.*
195251, Россия, Санкт-Петербург, ул. Политехническая, д. 29.
E-mail: lukash.spb.ru@gmail.com

LUKASHIN Alexey A. *Peter the Great St. Petersburg Polytechnic University.*
195251, Politekhnicheskaya Str. 29, St. Petersburg, Russia.
E-mail: lukash.spb.ru@gmail.com

ЗАБОРОВСКИЙ Владимир Сергеевич – *заведующий кафедрой телематики Санкт-Петербургского политехнического университета Петра Великого, доктор технических наук.*
195251, Россия, Санкт-Петербург, ул. Политехническая, д. 29.
E-mail: vlad@neva.ru

ZABOROVSKY Vladimir S. *Peter the Great St. Petersburg Polytechnic University.*
195251, Politekhnicheskaya Str. 29, St. Petersburg, Russia.
E-mail: vlad@neva.ru