УДК 519.68
DOI: 10.15827/0236-235X.128.565-572
Дата подачи статьи: 19.04.19
2019. Т. 32. № 4. С. 565–572

Интеллектуальный сбор информации из распределенных источников

М.С. Ефимова [1], *аспирант, maria.efimova@hotmail.com*

[1] *Санкт-Петербургский государственный электротехнический университет «ЛЭТИ» им. В.И. Ульянова (Ленина), г. Санкт-Петербург, 197376, Россия*

В статье рассмотрена задача сбора данных из распределенных источников на примере анализа разнородной распределенной финансовой информации, сделаны анализ и сравнение существующих подходов к сбору информации. Большинство из них для решения проблемы предполагают сбор данных в единое хранилище с последующим их анализом, однако это вызывает задержку от момента генерации данных до момента применения к ним методов анализа, связанную с необходимостью передачи от источника к месту хранения. В результате существенно снижается оперативность принятия решений и увеличивается трафик в сети. Кроме того, сбор данных от всех источников может привести к значительным расходам в случае, если доступ к некоторым из них платный или ограничен тарифным планом. Рассмотренные подходы предполагают включение хранилищ данных, средств ETL (излечения, трансформации и загрузки), лямбда-архитектуры, облачных вычислений, туманных вычислений, а также анализ распределенных данных на основе модели акторов. Однако выявлено, что они не учитывают стоимость и приоритеты источников данных и не позволяют обращаться к ним динамически, следовательно, не удовлетворяют всем условиям поставленной задачи.

В статье предложен и описан метод интеллектуального сбора информации с динамическим обращением к источникам данных в зависимости от текущей необходимости, стоимости и приоритета источников. Разработанный метод позволяет сократить трафик в сети, ускорить процесс анализа данных и снизить стоимость обращения к источникам данных.

Ключевые слова*: интеллектуальный анализ, распределенные источники, разнородные данные, анализ финансовой информации, Интернет вещей.*

В настоящее время растет число систем, осуществляющих анализ данных, которые поступают из разных источников (статические и динамические) и могут иметь разный формат, разную стоимость доступа, обновляться с разной периодичностью и т.п.

Примером подобных систем являются системы Интернета вещей. В настоящее время они объединяют в себе множество устройств, порождающих потоки данных. По прогнозам Gartner, Inc. (научно-исследовательская и консультационная корпорация), к 2020 году в Интернете вещей будет около 26 миллиардов устройств [1], каждое из которых является источником данных.

Еще одним примером распределенных источников являются ресурсы финансовой информации. Принятие решений на финансовых рынках требует анализа информации из различных источников. Это помогает в большей степени оценить ситуацию на рынке и увеличить точность принятия решения. Данными, используемыми для принятия решений, могут быть информация об изменении цен на финансовые инструменты, аналитические отчеты, новостные публикации, исторические экономические данные и т.д.

Подобного рода системы, кроме источников информации, могут содержать статические хранилища справочной информации, классификаторы, справочники, ранее обработанную статистику и др.

В последнее время проводится много исследований, посвященных интеллектуализации такого рода систем. Под интеллектуализацией часто понимают применение методов машинного обучения к данным, получаемым от источников для извлечения новых знаний, а также использование полученных знаний для управления источниками, в том числе и в автоматическом режиме.

Таким образом, интеллектуализация позволяет улучшить качество результатов принимаемых решений за счет использования современных методов анализа данных, получаемых от источников, и повысить оперативность управления источниками за счет автоматического принятия решений.

Однако обе задачи интеллектуализации сталкиваются с проблемой обработки данных

из многочисленных источников, с большими объемами информации, большой разновидностью типов данных и высокой скоростью их генерации. В последнее время к таким данным применяется термин «большие данные».

Большинство существующих подходов для решения этой проблемы предлагают сбор данных в единое хранилище с их последующим анализом. Однако такой подход предполагает задержку от момента генерации данных до момента применения к ним методов анализа, связанную с необходимостью передачи данных от источника к месту хранения. Это существенно снижает оперативность принятия решений, а также увеличивает трафик в сети. Кроме того, сбор данных от всех источников может привести к значительным расходам в случае, если доступ к некоторым из них является платным или ограничен тарифным планом.

При этом окончательный результат анализа данных, например, текущее состояние пациентов, аварийные предупреждения, изменения в котировках акций и т.д., важнее сбора всех данных из всех источников. Как правило, такие результаты основаны на определенной выборке из всех временных рядов (например, выборка для определенного периода времени, когда произошло интересующее событие). Следовательно, для решения непосредственно самих практических задач не требуется пересылать все данные из всех источников в единый центр обработки данных.

Постановка задачи

Рассмотрим задачу сбора информации из распределенных источников на примере анализа финансовых рынков, для решения которой требуется анализ биржевых котировок, курсов валют, новостных лент и т.п. (рис. 1).

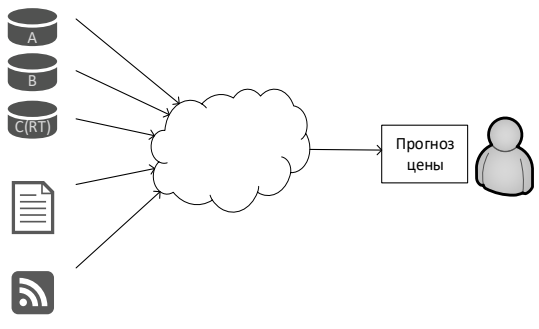

Рис. 1. Задача

Fig. 1. A Problem

Информация в таких источниках может изменяться как с некоторой периодичностью, так и постоянно. Данные от них могут поступать как по событию, так и по запросу. Количество источников и объем поступающей от них информации также могут быть разными. Примерами таких источников являются следующие.

- Провайдеры финансовой информации (например, Bloomberg, Reuters, Google Finance, Yahoo! Finance и др.), предоставляющие различные сведения:
  – полный исторический ценовой временной ряд (статический, ежедневный);
  – цена закрытия инструмента на одну конкретную дату (статическая, дневная);
  – самая последняя цена в течение дня (в режиме реального времени).
- Характеристики финансовых инструментов (статические источники информации).
- Реальные и исторические временные ряды связанных финансовых инструментов.
- Данные об объемах транзакции (поле в Bloomberg) – общий объем торговли за инструмент за один день.

Кроме того, для анализа могут использоваться дополнительные источники: новостные ленты, информационные сайты государственных органов и/или крупных корпораций и т.п.

В связи с распределенным характером источников система должна агрегировать данные из различных источников с разными форматами. При принятии решений также очень важна своевременность информации: задержки в системе могут привести к финансовым потерям. Следовательно, разрабатываемая система должна ускорять запросы и обработку данных и работать с источниками, обновляющимися в реальном времени. Обращения к некоторым источникам могут быть дорогостоящими, соответственно, разрабатываемое решение должно управлять информационными потоками для снижения стоимости.

Сбор информации от всех источников с последующим объединением и анализом может занимать много времени и серьезно увеличивать сетевой трафик. Альтернативным решением является сбор информации от источников по необходимости, которая определяется результатами анализа доступных частичных данных. Нередко прогноз может быть выполнен на основе данных из небольшого подмножества доступных источников. Источники данных могут быть классифицированы по приоритету. При определении приоритета источника учи-

тываются вид данных, стоимость обращения к источнику, а также значимость данных, содержащихся в нем для выполнения определенного вида прогнозирования.

Таким образом, для решения данной задачи система сбора и анализа должна удовлетворять следующим требованиям:

– агрегировать информацию из разных источников;

– работать с данными, меняющимися в реальном времени;

– работать с данными разных форматов;

– управлять информационными потоками (использовать источники по необходимости);

– обнаруживать ошибки как в исторических, так и в периодически меняющихся данных.

Существующие подходы к анализу данных из множества источников

Задача сбора информации из разных источников не является новой. В конце прошлого века и начале настоящего широкое распространение получила концепция построения хранилищ данных (DataWarehouse). Эта концепция была предложена в 1992 г. Б. Инмоном. Он определил хранилище данных как предметно-ориентированный, интегрированный, неизменчивый, поддерживающий хронологию набор данных, организованный для поддержки принятия решений [2]. В соответствии с ней данные из OLTP-систем собираются в хранилища данных для последующего анализа. Интеграция данных выполняется периодически из стационарных, как правило, реляционных источников.

Выделяют следующие типы архитектур хранилищ данных [2]:

– физическое хранилище данных, содержащее консолидированные данные, извлеченные из нескольких операционных источников [3];

– витрины данных, предоставляющие данные, интересные для конкретного пользователя или группы пользователей;

– виртуальное хранилище данных, предоставляющее конечным пользователям прямой доступ к нескольким источникам [4].

Характеристики архитектур хранилищ сравниваются в таблице 1.

Наиболее функциональными являются физические хранилища данных и витрины данных. Однако в отличие от виртуального хранилища данных они не обеспечивают доступ к текущей информации, что существенно снижает актуальность и оперативность результатов анализа.

Таблица 1

Сравнение архитектур хранилищ данных

Table 1

Comparison of data warehouse architecture types

Характеристика	Хранилище данных		
	Физическое	Витрины данных	Виртуальное
Предметно-ориентированный	+	+	+
Данные интегрированы	+	+	-
Накапливающиеся данные	+	+	-
Исторические данные включены	+	+	-
Текущие данные включены	-	-	+
Данные агрегированы	+	+	-
Детализированные данные	+	+	+

В качестве средств для переноса данных в физические хранилища используются ETL-инструменты [5]. Стандартные ETL-инструменты извлекают информацию из источников данных, очищают, объединяют и преобразуют ее в формат, поддерживаемый хранилищами данных, а затем загружают в них преобразованную информацию.

В настоящее время в десятку популярных ETL-инструментов входят следующие системы с открытым кодом:

– Pentaho Data Integration (PDI) (также известен как Kettle) – ETL-средство комплекса Pentaho [6];

– CloverETL – фреймворк для интеграции, преобразования, очистки и унификации данных в приложениях, БД и хранилищах [7];

– Talend Open Studio (TOS) – инструмент ETL для интеграции данных [8].

ETL-инструменты могут извлекать информацию разного формата из разных источников. Для этого могут использоваться соответствующие расширения для ETL-инструментов. Однако они не работают с источниками, обновляющимися с некоторой периодичностью, в том числе в реальном времени, и не имеют средств динамического подключения новых источников.

Позднее, с увеличением вариативности форматов источников данных, ростом объемов информации и частотой ее обновления встала проблема сбора и обработки больших данных. Для этого в настоящее время широко используются технологии noSQL, потоковой обработки [9] и системы, построенные на базе лямбда-архитектуры (lambda architecture (LA) [10], которая объединяет эти и другие технологии.

LA используется для одновременной обработки потоковых данных, поступающих в реальном времени, и пакетных данных. LA – это парадигма, определенная Натаном Марцем, которая обеспечивает основы для построения распределенных систем в реальном времени гибким и отказоустойчивым способом. Структура LA имеет три уровня.

Пакетный уровень – архив необработанных исторических данных. Этот уровень поддерживает пакеты данных и управляет их передачей.

Сервисный уровень индексирует пакеты и обрабатывает результаты вычислений, которые выполняются на уровне пакетов. Результаты немного задерживаются во времени из-за индексации и обработки входящей информации.

Потоковый уровень отвечает за обработку данных, поступающих в режиме реального времени. Это набор хранилищ данных, где они находятся в очереди в потоковом или рабочем режиме. На этом уровне разница в релевантности данных компенсируется, а информация с коротким жизненным циклом добавляется в конкретные представления в реальном времени (чтобы избежать дублирования данных). Эти представления обрабатывают свои запросы параллельно с уровнем обслуживания.

Системы с LA интегрируют гетерогенные компоненты, которые передают данные от одного компонента к другому (в пределах каждого из уровней). Такими компонентами могут быть разные средства: распределенной обработки информации, потоковой обработки, средства анализа и др. Примером является набор средств от организации Apache Software Foundation:

– Apache Hadoop для обработки больших объемов данных на уровне пакетов;

– Apache Spark Streaming для обработки потоков данных в режиме реального времени на уровне скорости;

– Apache Spark SQL и Apache Spark MLib для создания запросов и анализа на уровне сервиса.

Для LA- и ETL-инструментов предполагается, что данные собираются на сервисном уровне для дальнейшей обработки. Необходимость переноса всех данных из источников в централизованное хранилище порождает большой сетевой трафик и практически исключает возможность использования беспроводных каналов связи с ограниченной пропускной способностью. Кроме того, как и в случае с ETL, системы с LA не предполагают масштабирования источников информации и их динамического подключения. Однако в отличие от ETL LA позволяет работать с данными в реальном времени.

В системах Интернета решений для анализа данных и решения проблем вычислительных ресурсов используются технологии облачных вычислений [11]. Облако предоставляет масштабируемое хранилище, вычислительные ресурсы и другие инструменты для создания аналитических сервисов. Однако при таком подходе сохраняются перечисленные недостатки. Для их устранения компанией Cisco предложена концепция туманных вычислений [12]. Она расширяет облачные вычисления, перемещая часть вычислений ближе к источникам. Туманные вычисления полностью решают или снижают влияние ряда распространенных проблем распределенных систем:

– высокая задержка в сети;

– масштабирование источников информации;

– трудности, связанные с подвижностью конечных точек;

– высокая стоимость полосы пропускания;

– большая географическая распределенность систем.

Несмотря на достоинства и популярность концепции туманных вычислений, отсутствуют готовые решения для ее реализации. Это объясняется как молодостью данной концепции, так и высоким уровнем абстракции.

Одним из решений, соответствующих концепции туманных вычислений, является анализ распределенных данных на основе акторов [13]. Он может использоваться как для облачных, так и для туманных вычислений. Предложенный подход позволяет разбивать алгоритмы интеллектуального анализа данных на «чистые» функции и выполнять их на распределенных источниках.

Алгоритм интеллектуального анализа данных представляется в виде последовательности вызовов функций. Для их параллельного вы-

полнения добавлена функция, которая позволяет распараллеливать алгоритмы. Для выполнения в распределенной среде алгоритм интеллектуального анализа данных, разбитый на функции, был сопоставлен с моделью акторов. Таким образом, алгоритм для анализа распределенных данных представлен в виде набора акторов, которые обмениваются сообщениями с основным актором. Акторы переносят часть вычислений на источники, что повышает производительность анализа и уменьшает сетевой трафик между источниками и облаком. Однако этот подход имеет некоторые ограничения: он не позволяет устанавливать приоритеты источников данных и запрашивать данные в соответствии с их приоритетами. Кроме того, не учитывается стоимость запросов из источников данных.

В таблице 2 приведен сравнительный анализ рассмотренных подходов к сбору данных из распределенных источников в соответствии с требованиями к системе анализа. В результате можно сделать вывод, что ни одна из описанных технологий не позволяет решить задачу интеллектуального сбора информации для анализа.

Интеллектуальный сбор информации для анализа

Анализ данных из источников, запрашиваемых по необходимости, может быть альтернативой сбору всех данных перед выполнением анализа. Этот подход включает следующие ключевые шаги:

– выбор базовых источников данных; первоначальный анализ проводится на основе информации, поступающей из этих источников данных;

– определение приоритета запроса для каждого источника данных; может быть выполнено на основе различных критериев (например, стоимость получения информации и/или время обработки и сложность);

– запрос в другие источники данных в зависимости от требований, если информация из базовых источников данных недостаточна для предварительного анализа и/или вероятность того, что результаты верны, является низкой (например, в данных обнаруживаются выбросы и/или всплески);

– обработка данных и оценка результатов, выполняемые отдельно от основной задачи (например, прогнозирование цены инструмента) и, если возможно, вблизи источника данных (предпочтительно в узле, где расположены данные, или в его локальной сети).

Пример описанного подхода показан на рисунке 2. Источники данных *A* и *B* определены как базовые. Они могут содержать, например, временные ряды исторических дневных цен закрытия (*A*) и цены закрытия на одну конкретную дату (*B*). Эта информация объединяется для создания единого временного ряда изменения цены. Затем выполняется анализ для определения точности временных рядов (например,

Таблица 2

Сравнительный анализ подходов к сбору данных из распределенных источников

Table 2

Comparison of existing approaches to distributed data collection

Характеристика	ХД	ETL	LA	Облако	Туман	Анализ распределенных данных
Извлечение информации из нескольких источников	+	+	+	+	+	+
Работа с данными в разных форматах	-	+	+	+	+	+
Агрегирование информации из нескольких источников	+	+	+	+	+	+
Работа с данными в реальном времени	-	-	+	+	+	+
Выполнение в распределенной среде	-	+	+	+	+	+
Выполнение альтернативных веток обработки данных (блок «если»)	-	+	-	-	-	-
Выполнение динамических запросов к источникам данных	-	-	-	-	-	-
Выполнение анализа данных	-	-	+	+	+	+

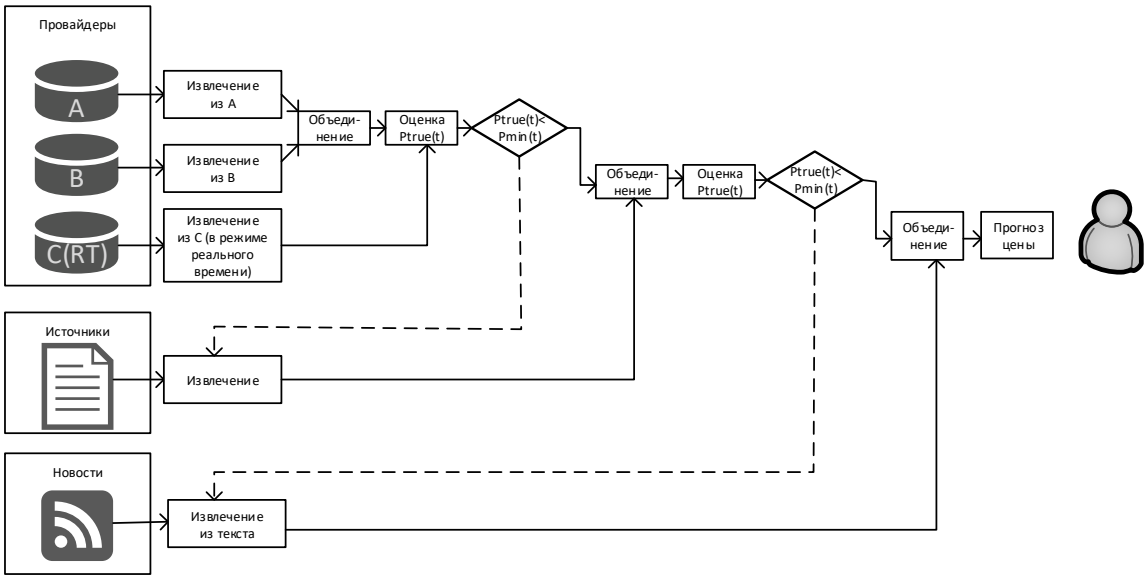

Рис. 2. Интеллектуальный сбор данных из распределенных источников

Fig. 2. Smart data collection from distributed data sources

если в данных есть выбросы и/или всплески). Если точность низкая, выполняется запрос к другим источникам данных, например, к источнику данных (*C*), который содержит самую последнюю цену. Если на момент запроса информация недоступна или ее недостаточно для повышения точности данных, выполняется запрос к следующему источнику данных в соответствии с определенным приоритетом. Примером такого источника данных может быть лента новостей, которая требует более сложных методов для анализа текста (методы NLP, субъективный анализ информации, анализ настроений и т.д.).

Для принятия решения о необходимости подключения дополнительного источника могут использоваться модели, построенные алгоритмами Data Mining. Например, при появлении выбросов во временных рядах основного потока для принятия решения о необходимости проверки, ошибка это или нет, можно применить модель, выявляющую такие выбросы. Модель может быть построена по ранее собранным данным и возникающим ситуациям, разрешаемым экспертами. Кроме того, при наличии обратной связи в процессе получения данных от аналитика она может обучаться, тем самым адаптируясь к текущим условиям.

Заключение

В статье описана задача сбора информации для анализа из распределенных источников. При неравнозначности таких источников (с точки зрения периодичности обновления информации, достоверности, стоимости и т.п.) возникает необходимость их динамического подключения к анализу. В настоящее время ни один из подходов, используемых для анализа данных, не отвечает всем требованиям такой задачи. Для решения предложен подход, определяющий базовые источники и в процессе выполнения анализа подключающий новые источники по мере необходимости в каждый момент времени. Для принятия решения о подключении нового источника предложено использовать методы интеллектуального анализа.

Литература

1. Gartner Says the Internet of Things Installed Base Will Grow to 26 Billion Units By 2020. 2013. URL: http://www.gartner.com/newsroom/id/2636073 (дата обращения: 15.04.2019).

2. Chandra P., Gupta M.K. Comprehensive survey on data warehousing research. Intern. J. of Inform. Technology, 2018, no. 10, pp. 217–224. DOI: 10.1007/s41870-017-0067-y.

3. Scabora L.C., Brito J.J., Ciferri R.R., Ciferri C.D.D.A. Physical data warehouse design on NoSQL databases – OLAP Query Processing over HBase. Proc. 18th Intern. Conf. SCITEPRESS, 2016, pp. 111–118. DOI: 10.5220/0005815901110118.

4. Khan F.A., Ahmad A., Imran M., Alharbi M., Jan B. Efficient data access and performance improvement model for virtual data warehouse. Sustainable cities and society, 2017, vol. 35, pp. 232–240. DOI: 10.1016/j.scs.2017.08.003.

5. Kimball R., Caserta J. The Data Warehouse ETL Toolkit. Wiley Publ., 2004, 526 p.

6. Pentaho Data Integration. URL: http://www.pentaho.com/product/data-integration (дата обращения: 15.04.2019).

7. Clover ETL. URL: http://www.cloveretl.com/products (дата обращения: 15.04.2019).

8. Talend Open Studio. URL: http://www.talend.com/products/talend-open-studio (дата обращения: 15.04.2019).

9. Aggarwal C.C. Data streams: models and algorithms. Springer Science & Business Media, 2007, 353 p. DOI: 10.1007/978-0-387-47534-9.

10. Marz N., Warren J. Big Data: Principles and Best Practices of Scalable Real-Time Data Systems. NY, Manning Publ., 2015, 330 p.

11. Gubbi J., Buyya R., Marusic S., Palaniswami M. Internet of Things (IoT): A vision, architectural elements, and future directions. Future Generation Computer Systems, 2013, no. 29, pp. 1645–1660.

12. Bonomi F., Milito R., Zhu J., Addepalli S. Fog computing and its role in the internet of things. Proc. MCC, Helsinki, Finland, 2012, pp. 13–16.

13. Kholod I., Petuhov I., Kapustin N. Creation of data mining cloud service on the actor model. Internet of Things, Smart Spaces, and Next Generation Networks and Systems. Springer, 2015, pp. 585–598.

Software & Systems Received 19.04.19
DOI: 10.15827/0236-235X.128.565-572 2019, vol. 32, no. 4, pp. 565–572

Smart data collection from distributed data sources

M.S. Efimova [1], *Postgraduate Student, maria.efimova@hotmail.com*

[1] *St. Petersburg Electrotechnical University "LETI", St. Petersburg, 197376, Russian Federation*

Abstract. The paper describes collecting and analysing data from distributed data sources using an example of analysing heterogeneous distributed financial information, analyzes and compares existing approaches to information collection and analysis. Most of the existing approaches that solve this problem require all data to be collected in a single repository to perform analysis on that data. However, such methods imply a delay from the moment when the data is generated until the moment when the analysis methods are applied to it due to the need to transfer the data from the source to the storage location. This significantly reduces the decision-making efficiency and increases network traffic. In addition, collecting data from all sources can lead to significant costs if access to some of the sources is not free or is limited by a tariff plan.

The considered approaches include data warehouses, ETL tools (extraction, transformation and loading), lambda architectures, cloud computing, fog computing, distributed data analysis based on the actor model. It has been concluded that these approaches do not take into account the cost and priorities of data sources and do not allow accessing them dynamically. Therefore, they do not meet all the requirements.

The paper proposes and describes a method of smart information collection with dynamic reference to data sources depending on current need, cost and source priority. The proposed method allows to reduce network traffic, speed up data analysis and reduce the costs associated with accessing data sources.

Keywords: smart data analysis, distributed data sources, heterogeneous data, financial analysis, Internet of Things.

References

1. *Gartner Says the Internet of Things Installed Base will Grow to 26 Billion Units By 2020*. 2013. Available at: http://www.gartner.com/newsroom/id/2636073 (accessed April 15, 2019).

2. Chandra P., Gupta M.K. Comprehensive survey on data warehousing research. *Intern. J. of Inform. Technology*. 2018, no. 10, pp. 217–224. DOI: 10.1007/s41870-017-0067-y.

3. Scabora L.C., Brito J.J., Ciferri R.R., Ciferri C.D.D.A. Physical data warehouse design on NoSQL databases – OLAP Query Processing over HBase. *Proc. 18th Intern. Conf. SCITEPRESS*. 2016, pp. 111–118. DOI: 10.5220/0005815901110118.

4. Khan F.A., Ahmad A., Imran M., Alharbi M., Jan B. Efficient data access and performance improvement model for virtual data warehouse. *Sustainable cities and society*. 2017, vol. 35, pp. 232–240. DOI: 10.1016/j.scs.2017.08.003.

5. Kimball R., Caserta J. *The Data Warehouse ETL Toolkit*. Wiley Publ., 2004, 526 p.

6. *Pentaho Data Integration*. Available at: http://www.pentaho.com/product/data-integration (accessed April 15, 2019).

7. *Clover ETL*. Available at: http://www.cloveretl.com/products (accessed April 15, 2019).

8. *Talend Open Studio*. Available at: http://www.talend.com/products/talend-open-studio (accessed April 15, 2019).

9. Aggarwal C.C. *Data Streams: Models and Algorithms*. Springer Science & Business Media, 2007, 353 p. DOI: 10.1007/978-0-387-47534-9.

10. Marz N., Warren J. *Big Data: Principles and Best Practices of Scalable Real-Time Data Systems*. NY, Manning Publ., 2015, 330 p.

11. Gubbi J., Buyya R., Marusic S., Palaniswami M. Internet of Things (IoT): A vision, architectural elements, and future directions. *Future Generation Computer Systems*. 2013, no. 29, pp. 1645–1660.

12. Bonomi F., Milito R., Zhu J., Addepalli S. Fog computing and its role in the internet of things. *Proc. MCC*. Helsinki, Finland, 2012, pp. 13–16.

13. Kholod I., Petuhov I., Kapustin N. Creation of data mining cloud service on the actor model. *Internet of Things, Smart Spaces, and Next Generation Networks and Systems*. Springer, 2015, pp. 585–598.

Для цитирования

Ефимова М.С. Интеллектуальный сбор информации из распределенных источников // Программные продукты и системы. 2019. Т. 32. № 4. С. 565–572. DOI: 10.15827/0236-235X.128.565-572.

For citation

Efimova M.S. Smart data collection from distributed data sources. *Software & Systems*. 2019, vol. 32, no. 4, pp. 565–572 (in Russ.). DOI: 10.15827/0236-235X.128.565-572.