УДК 004.75
DOI: 10.25559/SITITO.17.202101.731

Оригинальная статья

Применение стека технологий ELK для сбора и анализа системных журналов событий

Н. А. Балашов[1*], М. В. Балашова[2], С. Р. Книгин[2], Н. А. Кутовский[1]
[1] Международная межправительственная организация Объединенный институт ядерных исследований, г. Дубна, Российская Федерация
141980, Российская Федерация, Московская область, г. Дубна, ул. Жолио-Кюри, д. 6
[2] Государственное бюджетное образовательное учреждение высшего образования Московской области «Университет «Дубна», г. Дубна, Российская Федерация
141980, Российская Федерация, Московская обл., г. Дубна, ул. Университетская, д. 19
[*] balashov@jinr.ru

Аннотация
Современные научные исследования во многих областях часто требуют использования мощных вычислительных систем и сложных программных комплексов для эффективного решения исследовательских задач. Многие научные организации строят собственные вычислительные комплексы, одним из примеров которых является облачная инфраструктура Объединенного института ядерных исследований. В процессе эксплуатации подобных крупных вычислительных систем неизбежно возникают нештатные ситуации и сбои, разрешение которых в первую очередь опирается на анализ системных журналов событий. С ростом масштаба инфраструктуры и усложнением ее структуры процесс анализа журналов событий также усложняется и для его эффективной реализации в крупномасштабных инфраструктурах требуется внедрение дополнительных инструментов. В данной работе рассматривается опыт организации и внедрения системы централизованного сбора и анализа системных журналов событий облачной инфраструктуры ОИЯИ. В качестве основы для разрабатываемой системы был взят стек технологий *Elasticsearch*, *Logstash*, *Kibana* (*ELK*), широко применяющийся для решения схожих задач во многих других крупных научных вычислительных инфраструктурах и хорошо себя зарекомендовавший как для решения задач сбора и анализа журналов событий различных систем, так в ряде иных задач анализа слабоструктурированных и неструктурированных данных. На примере реализации механизма обеспечения отказоустойчивости управляющих узлов облачной инфраструктуры ОИЯИ показано, что современные системы могут иметь динамически изменяемую конфигурацию, приводящую к усложнению изучения журналов событий, и как с помощью разработанной системы можно упростить их анализ в подобных ситуациях.

Ключевые слова: облачные вычисления, отказоустойчивость, виртуализация, обработка данных.

Авторы заявляют об отсутствии конфликта интересов.

Для цитирования: Балашов, Н. А. Применение стека технологий ELK для сбора и анализа системных журналов событий / Н. А. Балашов, М. В. Балашова, С. Р. Книгин, Н. А. Кутовский. – DOI 10.25559/SITITO.17.202101.731 // Современные информационные технологии и ИТ-образование. – 2021. – Т. 17, № 1. – С. 61-68.

Original article

Using ELK Stack for Event Log Acquisition and Analysis

N. A. Balashov[a*], M. V. Balashova[b], S. R. Knigin[b], N. A. Kutovskiy[a]
[1] Joint Institute for Nuclear Research, Dubna, Russian Federation
6 Joliot-Curie St., Dubna 141980, Moscow region, Russian Federation
[2] Dubna State University, Dubna, Russian Federation
19 Universitetskaya St., Dubna 141980, Moscow region, Russian Federation
[*] balashov@jinr.ru

Abstract

Modern scientific research in many areas often requires the use of powerful computing systems and complex software systems to effectively solve research problems. Many scientific organizations build their own computing systems, an example of which is the cloud infrastructure of the Joint Institute for Nuclear Research. During the operation of such large computing systems, emergency situations and failures inevitably arise, the resolution of which is primarily based on the analysis of system event logs. As infrastructure grows in scale and complexity, event log analysis becomes a more complex process that requires additional tools to be used for its effective implementation in large-scale infrastructures. In this paper we share our experience in organizing and implementing a system for centralized collection and analysis of system event logs of the JINR cloud infrastructure. We chose the Elasticsearch, Logstash, Kibana (ELK) technology stack as the basis for the developed system, which is widely used to solve similar problems in many other large scientific computing infrastructures and has proven to be suitable both for solving problems of collecting and analyzing event logs of various systems, as well as a number of other problems in semi-structured and unstructured data analysis. On the example of the mechanism for ensuring the fault tolerance of the control nodes of the JINR cloud infrastructure we show that configurations of modern systems can have dynamic nature, complicating examination of system event logs, and how the developed system can be used to simplify their analysis in such situations.

Keywords: cloud computing, fault-tolerance, virtualization, data processing.

The authors declare no conflict of interest.

For citation: Balashov N.A., Balashova M.V., Knigin S.R., Kutovskiy N.A. Using ELK Stack for Event Log Acquisition and Analysis. *Sovremennye informacionnye tehnologii i IT-obrazovanie* = Modern Information Technologies and IT-Education. 2021; 17(1):61-68. DOI: https://doi.org/10.25559/SITITO.17.202101.731

Введение

Современные достижения в области информационных технологий и их широкое распространение в различных сферах деятельности привело к взрывному росту обрабатываемых данных и возникновению понятия «больших данных» [1, 2]. Наряду с ростом объема данных, происходит и рост сложности задач, решаемых с помощью информационных технологий, что ведет не только к росту масштаба вычислительных инфраструктур, но и к усложнению их устройства. Так, например, достижения в области технологий виртуализации и значительный рост объемов вычислительных ресурсов привели к возникновению нового направления исследований в области технологий построения и применения вычислительных инфраструктур – облачных вычислений [3, 4].

Одной из основных идей облачных вычислений является введение дополнительного уровня абстракции над физическими ресурсами вычислительной инфраструктуры в виде виртуальных машин (ВМ). При этом ВМ могут как являться конечным продуктом облачной инфраструктуры, так и служить базой для построения сложных многокомпонентных информационно-вычислительных систем, которые сами могут являться облачными сервисами более высокого уровня. Очевидно, что подобное усложнение структуры вычислительных инфраструктур и сервисов приводит и к усложнению процесса управления ими.

Одной из актуальных проблем, возникающих при управлении облачными инфраструктурами, является обработка возникающих нештатных ситуаций и сбоев в работе системы [5-7]. Так, задачу поиска первоисточников возникающих неполадок значительно усложняет распределенная архитектура современных систем с большим количеством взаимосвязей между их компонентами. Для ее решения, как правило, используется анализ данных журналов событий и различных показателей функционирования системы, получаемых с помощью систем мониторинга. Актуальность данной проблемы также подтверждается и большим количеством исследований, направленных на изучение методов извлечения полезной информации из получаемых и, как правило, слабоструктурированных данных [8-11].

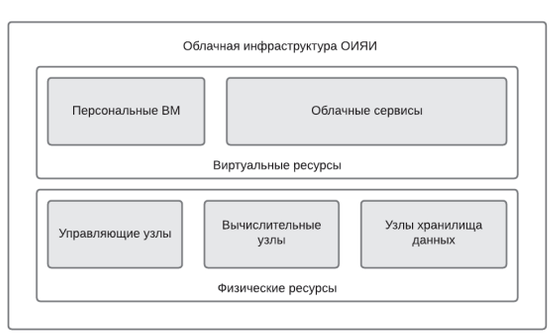
Другой актуальной проблемой является падение скорости извлечения данных из-за быстрого увеличения объемов получаемых данных. При возникновении нештатных ситуаций, скорость извлечения данных о функционировании системы напрямую влияет на скорость исправления возникающих сбоев и, соответственно, на общую стабильность работы системы. Для решения описанных проблем и задач разрабатываются программные продукты и комплексы, которые позволяют организовать сбор и хранение данных, а также предоставляют инструменты для их анализа. В данной работе рассматривается подобное техническое решение и аспекты его внедрения для анализа журналов событий облачной инфраструктуры Объединенного института ядерных исследований (ОИЯИ).

Облачная инфраструктура ОИЯИ

ОИЯИ принимает участие в большом количестве различных научных и образовательных проектов и экспериментов, которые нуждаются в надежной и масштабируемой информационной вычислительной инфраструктуре. На базе Лаборатории информационных технологий (ЛИТ) ОИЯИ была развернута облачная инфраструктура, базирующаяся на модели инфраструктура как сервис (англ. *Infrastructure as a Service, IaaS*). По данной модели конечным продуктом облачного сервиса являются виртуальные машины – виртуальные серверы с необходимой вычислительной мощностью. Облако ОИЯИ [12] построено на основе проекта с открытым исходным кодом *OpenNebula*, в состав которого входят средства для развертывания виртуального окружения, управления хранилищами данных, контроля доступа и мониторинга.

Сервис используется как для предоставления персональных ВМ индивидуальным пользователям, так и как основа для ряда многоузловых информационно-вычислительных комплексов, например, таких, как виртуальный вычислительный кластер *HTCondor* и сервис интерактивных вычислений *JupyterHub*. Также облако ОИЯИ является частью интегрированной облачной инфраструктуры стран-участниц ОИЯИ [13, 14], состоящей из отдельных облачных инфраструктур ее участников. Подобная интеграция дает возможность использования всех отдельных облачных вычислительных инфраструктур участников через единый интерфейс запуска вычислительных задач.

Облако ОИЯИ активно развивается. В 2015 году вычислительный ресурс облака ОИЯИ уже насчитывал 200 ядер ЦП и 400 ГБ оперативной памяти. К 2016 году цифры выросли до значений 330 ядер ЦП и 840 ГБ оперативной памяти. По состоянию на 2019 год, количество ядер ЦП возросло до 1564, а объем оперативной памяти до 8.54 ТБ. На текущий момент вычислительные мощности составляют 5000 ядер ЦП и 56 ТБ ОЗУ. На данный момент облако состоит из 205 серверов, которые условно можно отнести к трем типам: управляющие, вычислительные и узлы хранения. 184 сервера выделены под виртуальные машины пользователей. Высокая динамика роста вычислительных ресурсов влечет за собой увеличение числа виртуальных серверов, что, в свою очередь, значительно увеличивает потенциальный объем обрабатываемых данных. Хотя структура облака и является относительно простой на физическом уровне, на уровне использующих ее сервисов структура усложняется и приобретает динамический характер, в частности, за счет их масштабируемости. Упрощенная схема облачной инфраструктуры ОИЯИ представлена на рисунке 1.



Р и с. 1. Упрощенная схема облачной инфраструктуры ОИЯИ
F i g. 1. Simplified diagram of the JINR cloud infrastructure

Задача сбора и анализа журналов событий становится все более и более сложной из-за роста масштабов облака ОИЯИ, а также внедрения новых сервисов, построенных на его основе, и интеграции со сторонними вычислительными системами.

Требования к системе

Для обеспечения надежности и доступности сервисов требуется оперативно реагировать на отклоняющиеся от нормы события, возникающие в кластере облачной инфраструктуры. Критические ошибки и предупреждения могут повлечь за собой сбой функционирования системы, что может привести к потере пользовательских данных, данных экспериментов. Не исключены и угрозы информационной безопасности. Для выявления всех аномальных явлений необходимо проводить непрерывный системный сбор и анализ лог-файлов инфраструктуры, что невозможно без использования сторонних специализированных инструментов. Для решения этой задачи необходимо внедрить в облако ОИЯИ специализированную систему сбора и анализа лог-файлов.

Основные требования к системе были сформулированы исходя из описанной выше специфики облачной инфраструктуры ОИЯИ, они включают:

- Возможность интеграции с системой единого входа ОИЯИ (*Single Sign-On*, *SSO*).
- Возможность масштабирования системы при увеличении нагрузки.
- Способность обрабатывать слабоструктурированные данные в различных форматах, в частности - файлы системных журналов событий.
- Предоставление механизма защиты передаваемых по сети данных в процессе их сбора.
- Наличие графического пользовательского интерфейса для облегчения взаимодействия операторов с системой.
- Возможность установки собственного экземпляра системы на собственных вычислительных мощностях.
- Распространение по лицензии с открытым исходным кодом для обеспечения возможности свободной доработки системы, исходя из локальных особенностей.

Реализация системы

Основные компоненты

При реализации системы была задействована совокупность инструментов по сбору, анализу, агрегации и фильтрации данных, также известная под названием *ELK Stack*:

- *Elasticsearch* – распределенная система поиска и аналитики, которая позволяет хранить, анализировать и искать информацию из больших объемов данных. В своем ядре содержит *Lucene* – библиотеку для полнотекстового высокопроизводительного поиска.
- *Logstash* – конвейер обработки данных, который позволяет фильтровать и преобразовывать принимаемые данные и передавать полученные результаты одному или нескольким получателям.
- *Kibana* – веб-интерфейс для визуализации данных.

Elasticsearch применяется для решения большого спектра задач, начиная от обычного поиска слов в тексте [15], заканчивая сбором и анализом метрик производительности [16], геоданных [17], различных системных данных [18] и даже машинным обучением [19]. Рассматриваемый стек технологий хорошо себя зарекомендовал как в индустрии, так и в научных и образовательных организациях [20-25].

Elasticsearch Stack – программный продукт с поддержкой кластерной архитектуры, что позволит горизонтально масштабировать сервис при быстро растущем облаке.

ELK Stack является системой с открытым исходным кодом под лицензией *Apache* 2.0, что позволяет бесплатно использовать его продукты, в то же время у компании *Elastic* есть коммерческий продукт *X-Pack* под лицензией *Elastic License*, что существенно усложняет работу с продуктами *Elastic*. Часть функционала *X-Pack* доступна по пробной версии, но основные и довольно важные функциональные возможности распространяются только под коммерческой лицензией:

- *Index Lifecycle Management* – позволяет управлять состоянием индексов (индекс - специализированная структура и механизм управления данными в *Elasticsearch*).
- *Infrastructure and Logs UI* – плагин для визуализации, фильтрации и просмотра лог-файлов.
- *Kibana multi-tenancy* – инструменты для разграничения прав доступа пользователей к различным объектам системы (индексам, панелям визуализации данных и т.д.), необходимые для реализации многопользовательского использования системы.

Весной 2019 года *Amazon* выпустил *Open Distro for Elasticsearch*[1] под свободной лицензией *Apache* 2.0, который включает в себя свободные реализации многих возможностей из *X-Pack*. Во-первых, набор функций безопасности:

- *Node-to-node encryption* – шифрование трафика между узлами *Elasticsearch* кластера.
- *HTTP basic authentication* – метод аутентификации, который включает в себя имя пользователя и пароль как часть *HTTP*-запроса.
- *Role-based access control* – управление доступом на основе ролей.
- *Kibana multi-tenancy* – инструменты разграничения прав доступа.
- *Alerting* – система для отправки уведомлений.

Во-вторых, *SQL*, что предоставляет возможность писать запросы на языке *SQL*, а не на предметно-ориентированном языке запросов *Elasticsearch* (*DLS*).

В-третьих, *Performance Analyzer – REST API*, позволяющий получать различные метрики производительности кластера.

В-четвертых, *Index Management* – позволяет управлять индексами.

Используемый в описываемых работах дистрибутив *Open Distro for Elasticsearch* является полностью свободным и открытым программным решением, основная цель которого – обеспечение дальнейшего развития инновационных программных решений с открытым исходным кодом и свободное

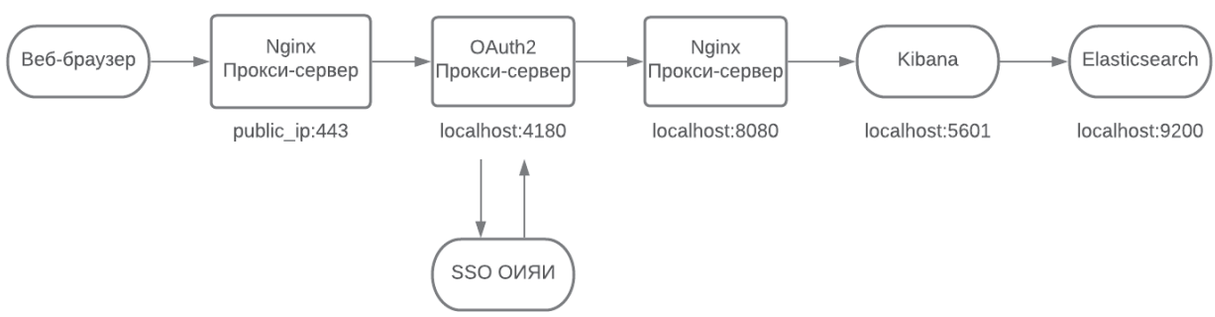
[1] Open Distro for Elasticsearch [Электронный ресурс]. URL: https://opendistro.github.io/for-elasticsearch (дата обращения: 02.03.2021).

распространение полнофункционального дистрибутива системы.

Интеграция с SSO ОИЯИ

Одним из важнейших критериев выбора системы являлась возможность ее интеграции с системой *SSO* ОИЯИ, которая является собственной реализацией популярного протокола аутентификации *OAuth2*[2]. Хотя выбранный стек технологий и не предоставляет прямой поддержки протокола *OAuth2*, он имеет механизм подключения внешних систем аутентификации через проксирующий сервер. Функциями прокси-сервера являются переадресация пользователя на страницу входа в системе *SSO* ОИЯИ и передача необходимых пользовательских данных сервисам *ELK*.

Для реализации описанного механизма был разработан собственный прокси-сервер на основе проекта *OAuth2 Proxy*[3], осуществляющий перенаправление пользовательских запросов на вход в систему *SSO* ОИЯИ и получения от нее пользовательских данных в случае успешной аутентификации. Для формирования корректного *HTTP*-запроса на вход в *Kibana* (из получаемых *OAuth2 Proxy* данных о пользователе) был использован широко известный веб-сервер *Nginx*. Также *Nginx* был использован и для установления безопасных соединений между веб-браузером пользователя, системой *ELK* и *SSO* ОИЯИ. Схема работы этого механизма проиллюстрирована на рисунке 2.



Р и с. 2. Схема интеграции технологии *SSO* в систему анализа данных

F i g. 2. Scheme for integrating SSO technology into a data analysis system

Защита сетевых соединений

Одним из требований к разрабатываемой системе было предоставление механизма защиты передаваемых по сети данных в процессе их сбора. В системе можно выделить три категории сетевых соединений, которые необходимо защитить:

- связь веб-браузера клиента с веб-интерфейсом *Kibana*;
- связи между компонентами *ELK*, находящихся на разных сетевых узлах;
- связи между узлами, с которых осуществляется сбор данных, и подсистемами *ELK*.

Защита первой категории соединений была обеспечена за счет шифрования трафика веб-сервером *Nginx* при реализации интеграции системы с *SSO* ОИЯИ.

Для обеспечения шифрования всех внутренних соединений, попадающих в оставшиеся две категории, в *ELK* предусмотрен собственный механизм, для реализации которого требуется наличие удостоверяющего центра (УЦ). Задачей УЦ является подтверждение подлинности ключей шифрования с помощью сертификатов электронной подписи.

Подобный механизм защиты имеет широкую область применения, и в крупных вычислительных инфраструктурах, подобных облачной инфраструктуре ОИЯИ, единый УЦ может быть использован в разных компонентах инфраструктуры. В частности, в рассматриваемой инфраструктуре уже был ранее создан УЦ для защиты соединений в системе управления конфигурациями *Puppet*[4], использующейся для конфигурирования облачных узлов. Таким образом, уже на этапе первоначальной настройки новые узлы получают собственные сертификаты, подписанные локальным УЦ, которые впоследствии используются для шифрования всех внутренних соединений. Подобная интеграция позволила избежать дублирования функционала для решения схожих задач в разных частях инфраструктуры.

Применение Logstash

Одним из примеров, иллюстрирующих динамическое изменение конфигурации облачной инфраструктуры, является механизм обеспечения отказоустойчивости управляющих узлов облака ОИЯИ. Он основан на консенсусном алгоритме *Raft*[5], который гарантирует согласованность серверов относительно друг друга. Это достигается тем, что в кластере из 3 управляющих узлов, путем голосования, выбирается специальный узел,

[2] *Hardt D.* The OAuth 2.0 Authorization Framework. RFC 6749. October 2012. DOI: 10.17487/RFC6749 [Электронный ресурс]. URL: https://www.rfc-editor.org/info/rfc6749 (дата обращения: 02.03.2021).

[3] *Speed J.* OAuth2 Proxy [Электронный ресурс]. URL: https://github.com/oauth2-proxy/oauth2-proxy (дата обращения: 02.03.2021).

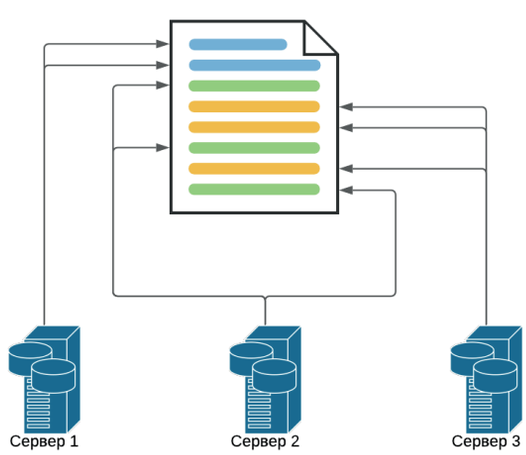
[4] Puppet: Powerful infrastructure automation and delivery [Электронный ресурс]. URL: https://puppet.com (дата обращения: 02.03.2021).

[5] OpenNebula [Электронный ресурс]. URL: https://docs.opennebula.io/5.12/advanced_components/ha/frontend_ha_setup.html#raft-overview (дата обращения: 02.03.2021).

называемый «лидером», который будет обеспечивать функциональность системы. Лидер периодически посылает сигналы остальным узлам, которые называются «последователи», чтобы сохранить свой статус лидерства. Если управляющему узлу кластера не удалось отправить сигнал, то выбираются кандидаты и начинаются новые выборы лидера. При каждой модификации системы, прежде чем сделать запись о смене состояния системы в базе данных (БД), лидер записывает в БД последовательность операций, которые необходимо выполнить для смены состояния системы, и реплицирует эти записи на последователей. Данная операция увеличивает задержку выполнения операций при работе с БД, но гарантирует сохранение целостности состояния системы и обеспечивает сохранение работоспособности кластера при выходе из строя лидирующего узла.

Из описания алгоритма *Raft* видно, что управляющие узлы в каждый отдельно взятый момент времени будут иметь различные роли, которые со временем могут измениться (например, в случае возникновения сбоя в системе). При этом каждый узел ведет собственные системные журналы событий, но из-за возможной смены ролей узлов отдельные части содержимого одного и того же лог-файла на отдельно взятом узле могут соответствовать разным ролям, что затрудняет анализ журналов событий.

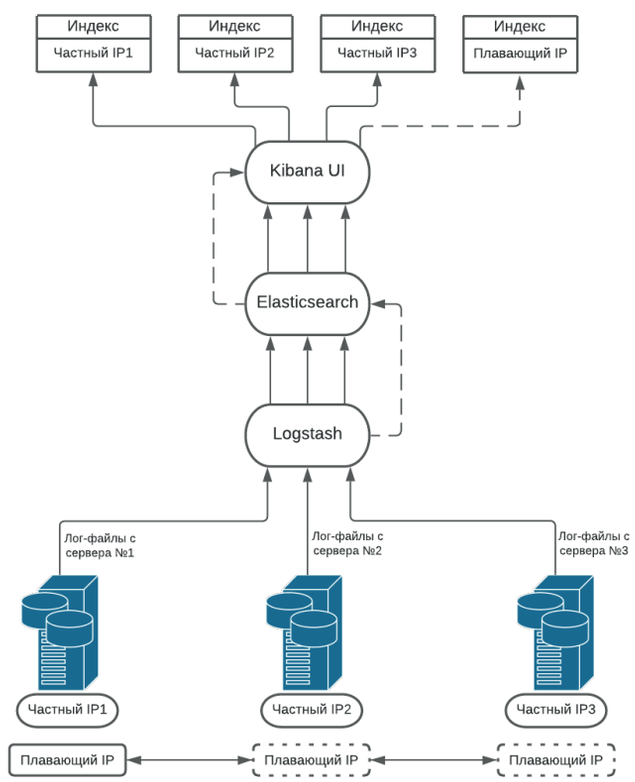
Алгоритм *Raft* предусматривает наличие у лидера дополнительного «плавающего» *ip*-адреса, который при выходе из строя текущего лидера переходит к новому. Так как у каждого из трех узлов всегда имеется свой собственный *ip*-адрес, плавающий *ip*-адрес можно использовать как индикатор наличия у узла роли «лидер». Таким образом, упростить анализ журнала событий можно, организовав запись системных журналов не в три индекса *Elasticsearch*, а в четыре, соответствующих трем постоянным и одному плавающему *ip*-адресу. В данной организации четвертый индекс всегда будет содержать только записи журналов (что проиллюстрировано на рисунке 3), соответствующие лидирующему узлу, и при этом сохранится возможность независимого просмотра журналов всех трех узлов.



Р и с. 3. Схема формирования единого журнала лидирующего узла. Цветом выделены блоки записей журнала событий, поступающие с соответствующих серверов.

F i g. 3. The formation scheme of a single journal of the leading node. The color shows the blocks of event log records coming from the corresponding servers.

Для реализации описанной выше задачи был использован функционал подсистемы *Logstash*. С помощью встроенных фильтров *Logstash* извлекает *ip*-адреса управляющих узлов, сравнивает их с заданным шаблоном и отправляет данные в соответствующие индексы, что позволяет просматривать записи журналов независимо как со всех трех узлов, так и записи только лидирующего узла. Данная реализация проиллюстрирована на рисунке 4.



Р и с. 4. Схема пополнения индексов данными, содержащими записи журналов событий управляющих узлов

F i g. 4. Scheme for replenishing indexes with data containing records of the event logs of control node

Заключение

Анализ системных журналов является одной из важнейших задач для поддержания стабильной работы любой сложной программной системы, и с ростом масштабов и сложности современных информационно-вычислительных систем данная задача становится все более трудновыполнимой. В данной работе был описан опыт внедрения стека технологий *ELK* в качестве системы сбора и анализа системных журналов событий облачной инфраструктуры ОИЯИ. Данный опыт показывает, что, используя данный стек технологий, можно довольно быстро и достаточно просто построить систему сбора и анализа системных журналов событий для осуществления централизованного мониторинга *IT*-инфраструктуры. В дальнейшем планируется расширить область применения системы на все

облачные сервисы ОИЯИ, а также оценить возможность её применения и для сбора журналов работы пользовательских вычислительных задач, выполняемых на кластере *HTCondor*. Использование *ELK* в качестве системы анализа системных журналов событий – лишь частный случай в текущей практике, данный инструмент может быть востребован и в других исследовательских проектах ОИЯИ.

References

[1] Nawsher Khan et al. Big Data: Survey, Technologies, Opportunities, and Challenges. *The Scientific World Journal*. 2014; 2014:712826. (In Eng.) DOI: https://doi.org/10.1155/2014/712826

[2] Oussous A., Benjelloun F.-Z., Lahcen A.A., Belfkih S. Big Data technologies: A survey. *Journal of King Saud University – Computer and Information Sciences*. 2018; 30(4):431-448. (In Eng.) DOI: https://doi.org/10.1016/j.jksuci.2017.06.001

[3] Armbrust M. et al. A view of cloud computing. *Communications of the ACM*. 2010; 53(4):50-58. (In Eng.) DOI: https://doi.org/10.1145/1721654.1721672

[4] Varghese B., Buyya R. Next generation cloud computing: New trends and research directions. *Future Generation Computer Systems*. 2018; 79(3):849-861. (In Eng.) DOI: https://doi.org/10.1016/j.future.2017.09.020

[5] Prathiba S., Sowvarnica S. Survey of failures and fault tolerance in cloud. *2017 2nd International Conference on Computing and Communications Technologies (ICCCT)*. Chennai, India, IEEE; 2017. p. 169-172. (In Eng.) DOI: https://doi.org/10.1109/ICCCT2.2017.7972271

[6] Garraghan P., Townend P., Xu J. An Empirical Failure-Analysis of a Large-Scale Cloud Computing Environment. *2014 IEEE 15th International Symposium on High-Assurance Systems Engineering*. Miami Beach, FL, USA, IEEE; 2014. p. 113-120. (In Eng.) DOI: https://doi.org/10.1109/HASE.2014.24

[7] Kochhar D., Jabanjalin H. An approach for fault tolerance in cloud computing using machine learning technique. *International Journal of Pure and Applied Mathematics*. 2017; 117(22):345-351. (In Eng.) DOI: https://doi.org/10.13140/RG.2.2.31419.67366

[8] He S. et al. A Survey on Automated Log Analysis for Reliability Engineering. arXiv:2009.07237. 2020. Available at: https://arxiv.org/abs/2009.07237 (accessed 02.03.2021). (In Eng.)

[9] Du M., Li F. Spell: Online streaming parsing of large unstructured system logs. *IEEE Transactions on Knowledge and Data Engineering*. 2019; 31(11):2213-2227. (In Eng.) DOI: https://doi.org/10.1109/TKDE.2018.2875442

[10] Wei D. et al. Research on unstructured text data mining and fault classification based on RNN-LSTM with malfunction inspection report. *Energies*. 2017; 10(3):406. (In Eng.) DOI: https://doi.org/10.3390/en10030406

[11] Zhang Q., Cao J., Sui Y. Development of a research platform for BEPC II accelerator fault diagnosis. *Radiation Detection Technology and Methods*. 2020; 4(3):269-276. (In Eng.) DOI: https://doi.org/10.1007/s41605-020-00180-2

[12] Balashov N.A., Baranov A.V., Kutovskiy N.A., Makhalkin A.N., Mazhitova Ye.M., Pelevanyuk I.S., Semenov R.N. Present Status and Main Directions of the JINR Cloud Development. *CEUR Workshop Proceedings: Proc. of 27th International Symposium NEC-2019 (Budva, Montenegro)*. 2019; 2507:185-189. Available at: http://ceur-ws.org/Vol-2507/185-189-paper-32.pdf (accessed 02.03.2021). (In Eng.)

[13] Balashov N. et al. Creating a Unified Educational Environment for Training IT Specialists of Organizations of the JINR Member States in the Field of Cloud Technologies. In: Sukhomlin V., Zubareva E. (Eds.) *Modern Information Technology and IT Education. SITITO 2018. Communications in Computer and Information Science*, vol. 1201. Springer, Cham; 2020. p. 149-162. (In Eng.) DOI: https://doi.org/10.1007/978-3-030-46895-8_12

[14] Balashov N.A. et al. Cloud integration within the Dirac interware. *CEUR Workshop Proceedings: Proc. of 27th International Symposium NEC-2019 (Budva, Montenegro)*. 2019; 2507:256-260. Available at: http://ceur-ws.org/Vol-2507/256-260-paper-45.pdf (accessed 02.03.2021). (In Eng.)

[15] Barbaresi A., Tinoco A.R. Using Elasticsearch for Linguistic Analysis of Tweets in Time and Space. *LREC 2018*. Miyazaki, Japan; 2018. p. 14-19. Available at: https://hal.archives-ouvertes.fr/hal-01798706 (accessed 02.03.2021). (In Eng.)

[16] Betke E., Kunkel J. Real-Time I/O-Monitoring of HPC Applications with SIOX, Elasticsearch, Grafana and FUSE. In: Kunkel J., Yokota R., Taufer M., Shalf J. (Eds.) *High Performance Computing. ISC High Performance 2017. Lecture Notes in Computer Science*, vol. 10524. Springer, Cham; 2017. p. 174-186. (In Eng.) DOI: https://doi.org/10.1007/978-3-319-67630-2_15

[17] Psaila G., Fosci P. J-CO: A Platform-Independent Framework for Managing Geo-Referenced JSON Data Sets. *Electronics*. 2021; 10(5):621. (In Eng.) DOI: https://doi.org/10.3390/electronics10050621

[18] Bajer M. Building an IoT Data Hub with Elasticsearch, Logstash and Kibana. *2017 5th International Conference on Future Internet of Things and Cloud Workshops (FiCloudW)*. Prague, Czech Republic, IEEE; 2017. p. 63-68. (In Eng.) DOI: https://doi.org/10.1109/FiCloudW.2017.101

[19] Negoita O., Carabas M. Enhanced Security Using Elasticsearch and Machine Learning. In: Arai K., Kapoor S., Bhatia R. (Eds.) *Intelligent Computing. SAI 2020. Advances in Intelligent Systems and Computing*, vol. 1230. Springer, Cham; 2020. P. 244-254. (In Eng.) DOI: https://doi.org/10.1007/978-3-030-52243-8_19

[20] Aimar A. et al. MONIT: Monitoring the CERN Data Centres and the WLCG Infrastructure. *EPJ Web of Conferences*. 2019; 214:08031. (In Eng.) DOI: https://doi.org/10.1051/epjconf/201921408031

[21] Herner K. et al. Advances in grid computing for the fabric for frontier experiments project at Fermilab. *Journal of Physics: Conference Series*. 2017; 898(5):052026. (In Eng.) DOI: https://doi.org/10.1088/1742-6596/898/5/052026

[22] Robles-Gómez A. et al. Using Kibana and ElasticSearch for the Recommendation of Job Offers to Students. *CEUR Workshop Proceedings.* 2017; 1925:93-99. Available at: http://

ceur-ws.org/Vol-1925/paper09.pdf (accessed 02.03.2021). (In Eng.)
[23] Han L., Zhu L. Design and Implementation of Elasticsearch for Media Data. *2020 International Conference on Computer Engineering and Application (ICCEA).* Guangzhou, China; 2020. p. 137-140. (In Eng.) DOI: https://doi.org/10.1109/ICCEA50009.2020.00036
[24] Andreeva J., Boehm M., Gaidioz B. et al. Experiment Dashboard for Monitoring Computing Activities of the LHC Virtual Organizations. *Journal of Grid Computing*. 2010; 8(2):323-339. (In Eng.) DOI: https://doi.org/10.1007/s10723-010-9148-x
[25] Kuc R., Rogozinski M. Mastering Elasticsearch. 2nd ed. Packt Publishing Ltd., Birmingham; 2015. (In Eng.)

Поступила 02.03.2021; одобрена после рецензирования 30.03.2021; принята к публикации 05.04.2021.
Submitted 02.03.2021; approved after reviewing 30.03.2021; accepted for publication 05.04.2021.

Об авторах:

Балашов Никита Александрович, инженер-программист Лаборатории информационных технологий, Международная межправительственная организация Объединенный институт ядерных исследований (141980, Российская Федерация, Московская область, г. Дубна, ул. Жолио-Кюри, д. 6), **ORCID:** http://orcid.org/0000-0002-3646-0522, balashov@jinr.ru
Балашова Марина Владимировна, старший преподаватель кафедры системного анализа и управления, Институт системного анализа и управления, Государственное бюджетное образовательное учреждение высшего образования Московской области «Университет «Дубна» (141980, Российская Федерация, Московская обл., г. Дубна, ул. Университетская, д. 19), **ORCID:** http://orcid.org/0000-0002-7110-5008, balashova.m.v@yandex.ru
Книгин Сергей Романович, студент кафедры системного анализа и управления, Институт системного анализа и управления, Государственное бюджетное образовательное учреждение высшего образования Московской области «Университет «Дубна» (141980, Российская Федерация, Московская обл., г. Дубна, ул. Университетская, д. 19), **ORCID:** http://orcid.org/0000-0003-3676-8014, knigin.sr@gmail.com
Кутовский Николай Александрович, старший научный сотрудник Лаборатории информационных технологий, Международная межправительственная организация Объединенный институт ядерных исследований (141980, Российская Федерация, Московская область, г. Дубна, ул. Жолио-Кюри, д. 6), кандидат физико-математических наук, **ORCID:** http://orcid.org/0000-0002-2920-8775, kut@jinr.ru

Все авторы прочитали и одобрили окончательный вариант рукописи.

About the authors:

Nikita A. Balashov, Software engineer of the Laboratory of Information Technologies, Joint Institute for Nuclear Research (6 Joliot-Curie St., Dubna 141980, Moscow region, Russian Federation), **ORCID:** http://orcid.org/0000-0002-3646-0522, balashov@jinr.ru
Marina V. Balashova, Senior lecturer at the Department of System Analysis and Management, Institute of System Analysis and Management, Dubna State University (19 Universitetskaya St., Dubna 141980, Moscow region, Russian Federation), **ORCID:** http://orcid.org/0000-0002-7110-5008, balashova.m.v@yandex.ru
Sergey R. Knigin, Student at the Department of System Analysis and Management, Institute of System Analysis and Management, Dubna State University (19 Universitetskaya St., Dubna 141980, Moscow region, Russian Federation), **ORCID:** http://orcid.org/0000-0003-3676-8014, knigin.sr@gmail.com
Nikolay A. Kutovskiy, Senior Researcher of the Laboratory of Information Technologies, Joint Institute for Nuclear Research (6 Joliot-Curie St., Dubna 141980, Moscow region, Russian Federation), Ph.D. (Phys.-Math.), **ORCID:** http://orcid.org/0000-0002-2920-8775, kut@jinr.ru

All authors have read and approved the final manuscript.