Метод поиска паттернов в лог файлах телекоммуникационных устройств для мониторинга их состояния

Н. А. Жукова
Санкт-Петербургский государственный электротехнический университет «ЛЭТИ» им. В.И. Ульянова (Ленина);
Санкт-Петербургский институт информатики и автоматизации Российской академии наук
nazhukova@mail.ru

И. А. Куликов
Санкт-Петербургский государственный электротехнический университет «ЛЭТИ» им. В.И. Ульянова (Ленина)
i.a.kulikov@gmail.com

Н. Ю. Уткин
Санкт-Петербургский государственный электротехнический университет «ЛЭТИ» им. В.И. Ульянова (Ленина)
utkin.kolya2012@yandex.ru

***Аннотация*. В докладе рассмотрен метод поиска паттернов в лог файлах телекоммуникационных устройств. Представлен обзор используемых на практике методов, показана значимость лог файлов, как источника оперативных данных, предоставляемых системами мониторинга. Результаты обработки лог файлов используются при построении и перестройки моделей телекоммуникационных сетей, что позволяет учитывать в них оперативные данные. По данным реальных лог файлов, полученных от устройств, применяемых в телекоммуникационных сетях, построены паттерны, характеризующие их поведение. Разработана программа на языке Phyton для поиска и анализа лог файлов, формирования пакетов оперативных данных и добавления их в модель телекоммуникационной сети.**

Ключевые слова: телекоммуникационная сеть, граф знаний, анализ лог файлов, паттерны

I. Паттерны состояний устройств по данным лог файлов и методы их поиска

Средства для анализа лог файлов предоставляют многие производители программного обеспечения (ПО), такие как Splunk [1], Sumo Logic [2], Loggly [3], LogEntries [4]. Их решения по обработки и анализу содержания лог файлов широко используются в различных информационных системах. К общим требованиям, предъявляемым к средствам анализа лог файлов, относятся следующие [5]:

- Отсутствие требований к настройкам, требующим участия эксперта предметной области. При изменении форматов сообщений в лог файлах или появлении новых, средство поиска паттернов не должно требовать дополнительных действий от администратора.
- Возможность обработки лог файлов с неоднородной структурой: средство поиска паттернов должно поддерживать форматы сообщений, формируемых различными системами. Каждая система может формировать сообщения, размещаемые в лог файлах в нескольких форматах.
- Возможность обработки больших объемов данных: многие системы, в частности, IoT-системы, генерируют миллионы сообщений каждый день. Производительность средств обработки лог файлов должна обеспечивать возможность их оперативной обработки.
- Масштабируемость: средство поиска паттернов должно поддерживать распределенную обработку содержащихся в лог файлах сообщений.

При мониторинге телекоммуникационных сетей (ТС) необходимо иметь возможность определять, путем анализа лог файлов, изменения состояний телекоммуникационных устройств и приложений, которые на них запускаются. На основе последовательностей появления паттернов возможно восстановить поведение запускаемых на устройствах приложений. Пример сообщений лог файла, по которым можно судить о состоянии приложений, запущенных на устройстве, приведен ниже:

Примеры записей лог файла:

[INF] 01.12.2016 08:36:17.650 [DOB2PE] [781:825 DOB2::AppThread] :2782 [I] ScriptLogger: [start] 37524 [DUMP][CH][CB] dcn:55#"TLCHD" fl:0x00000330 tribId:57391 url:"watchTv?locator=qam://sourceid=16123"

[INF] 01.12.2016 08:36:17.719 [DVBS_VIDEO] [615:900 QamMediaSource] :1999 QamMediaSource::StateWaitingSI::handle_si_ready:2520:[0x1017548]: SI record is ready, media_source: 17 source_id: 0x3efb frequency: 165000000Hz qam_mode: 16 program_Ready] with error [NoError]

Структура записей лог файла:

[LOG LEVEL] [DATE] [TIME] [APPLICATION ID] [COMPONENT ID] [LOGGED MESSAGE]

Сообщения в рассмотренных лог файлах имеют следующую структуру: *[LOG LEVEL] [APPLICATION ID] [COMPONENT ID] [LOGGED MESSAGE].* В поле *[LOGGED MESSAGE]* размещаются данные о состоянии устройства, в значительном числе случаев данные не являются полностью структурированными, что требует применения регулярных выражений (regexp) при поиске паттернов.

II. Метод поиска паттернов в лог файлах с применением программного средства Logstash

В качестве программного средства для анализа лог файлов был использован Logstash [6]. Logstash представляет собой решение, обеспечивающее пересылку сообщений со множеством возможностей по обработке пересылаемых сообщений. Он получает данные из нескольких источников одновременно, преобразует их, а затем отправляет потребителю информации. Каждое событие обрабатывается трехступенчатым конвейером: Вход → Фильтрация → Выход. Использование плагинов позволяет обрабатывать различные потоки, поступающие на вход программного средства. Logstash поддерживает различные способы получения данных, включая загрузку из файлов в формате CSV, передачу с помощью сокетов TCP / UDP, по HTTP API, получение данных от Elasticsearch и многие другие. При обработке сообщений в них могут вноситься изменения – удалены необязательные данные или добавлены дополнительные. Logstash предоставляет плагины для выполнения разнообразных операций с данными, в том числе, фильтрации данных. Например, с помощью фильтра Grok можно извлекать данные из строковых полей по шаблонам, фильтр Ruby позволяет выполнять собственный код Ruby для обработки событий, данные о которых размещены в сообщениях. Logstash поддерживает множество вариантов представления данных. Как правило, данные о событиях, зарегистрированных в сообщениях, передаются в Elasticsearch, также в Logstash имеется возможность сохранения данных в файлы формата CSV, JSON, в базу данных SQL и пр. [8].

Для конфигурирования Logstash необходимо сопоставить содержание лог файлов со сценариями, описывающими поведение приложений, установленных на анализируемых устройствах. Сценарии поведения описываются в виде последовательности событий, происходящих на устройствах. Правило определяется в виде подстроки или регулярного выражения, поиск которого выполняется в записи, размещенном в поле [LOGGED MESSAGE]. Конфигурация программного средства Logstash приведена в табл. 1.

ТАБЛИЦА I Конфигурация программного средства Logstash

№ п/п	Компоненты записи лог файла	Правило анализа лог файла	Произошедшее событие
1	[LOG LEVEL]	Правило 1	Событие 1
2	[APPLICATION ID]	Правило 2	Событие 2
3	[COMPONENT ID]	Правило 3	Событие 3
4	[LOGGED MESSAGE]	Правило 4	Событие 4

В результате использования настроенного таким образом Logstash возможно получать сценарии поведения выбранных приложений на устройстве. При наличии описаний шаблонов поведения приложений, определяющих последовательности событий при нормальной работе приложений и при возникновении ошибочных ситуаций, возможно делать обоснованные выводы о состоянии устройств и определять необходимые управляющие воздействия.

III. Пример поиска паттернов в лог файлах абонентских устройств оператора кабельного ТВ

В качестве анализируемого устройства рассматривался ресивер цифрового телевидения (Set-Top Box, STB) [7], в качестве анализируемого приложения – WhatchTV (просмотр каналов вещательного ТВ), в качестве отслеживаемого сценария – переключение просматриваемого абонентом канала.

Ниже приведены записи из лога файла устройства, отражающие события успешного переключения между каналами:

[INF] 01.12.2016 08:36:17.615 [DOB2PE] [781:825 DOB2::AppThread] :2734 [I] ScriptLogger: [start] 37489 [watchTv] doOnFocus cmd:-1 {tuner:1, state:STATE_INIT, errId:0} popup:{cnt:0, isShow:0}, cb:{isShow:1, state:0/0, vis:0 [I] Video: switch_to[0x81bf58]: url(qam://sourceid=16123)

[INF] 01.12.2016 08:36:17.650 [DOB2PE] [781:825 DOB2::AppThread] :2782 [I] ScriptLogger: [start] 37524 [DUMP][CH][CB] dcn:55#"TLCHD" fl:0x00000330 tribId:57391 url:"watchTv?locator=qam://sourceid=16123"

[INF] 01.12.2016 08:36:17.719 [DVBS_VIDEO] [615:900 QamMediaSource] :1999 QamMediaSource::StateWaitingSI::handle_si_ready:2520:[0x1017548]: SI record is ready, media_source: 17 source_id: 0x3efb frequency: 165000000Hz qam_mode: 16 program_Ready] with error [NoError]

[INF] 01.12.2016 08:36:18.007 [DVBS_VIDEO] [615:900 QamMediaSource] :2024 QamMediaSource::StateWaitingPat::handle_pat_ready:2636:[0x1017558]: waiting for PMT, media_source: 17 program_number: 3

[INF] 01.12.2016 08:36:18.070 [DVBS_CAS] [615:900 QamMediaSource] :2030 CasSessionImpl::reinit:414:[0x163bc20]: Search for CA descriptors for ca_system_id: 0x96b

[INF] 01.12.2016 08:36:18.070 [DVBS_CAS] [615:900 QamMediaSource] :2031 CasSessionImpl::find_program_level_ca_descr:330: Found supported Program CA descriptor ca_system_id: 0x96b, ecm_pid: 0x217

Результаты анализа представленного фрагмента лог файла представлены в табл. 2.

ТАБЛИЦА II Паттерны переключения каналов на STB

№ п/п	Паттерны	Состояние устройства
1.	[LOG LEVEL] = [INF] [APPLICATION ID] = [DOB2PE] [LOGGED MESSAGE] **Ключевая строка:** url(qam://sourceid=16123) **Регулярное выражение для поиска:** [\w\d\[\]\._\:]+ url\(qam:\/\/sourceid=[0-9]{0,6};[\D]+=\d;\)\n	Приложение “Watch TV” получило фокус и URL локатора QAM канала с идентификатором “16123”

№ п/п	Паттерны	Состояние устройства
2.	[LOG LEVEL] = [INF] [APPLICATION ID] = [DOB2PE] [LOGGED MESSAGE] **Ключевая строка:** …[DUMP][CH][CB] dcn:55#"TLCHD …url:"watchTv?locator=qam://sourceid=16123 … **Регулярное выражение для поиска:** [\w\d\[\]\._\:\"\#]+ \[DUMP\]\[CH\]\[\w\d\[\]\._\:\"\#]+ url:"watchTv\?locator=qam:\/\/ sourceid=\w{0,6}"	Данные локатора QAM канала прочитаны; dcn = 55 – номер канала в сетке вещания, callsign = TLCHD – уникальное сокращенное наименование канала
3.	[LOG LEVEL] = [INF] [APPLICATION ID] = [DVBS_VIDEO] [LOGGED MESSAGE] **Ключевая строка:** … source_id: 0x3efb frequency: 165000000Hz qam_mode: 16 program_Ready … **Регулярное выражение для поиска:** [\S]+ frequency: *\d+ *[Hz]*, qam_mode: \d+, program_number: [\S]+	QAM тюнер инициализирован и настроен на канал со следующими параметрами: *source_id: 0x3efb* *frequency: 165000000Hz* *qam_mode: 16*
4.	[LOG LEVEL] = [INF] [APPLICATION ID] = [DVBS_CAS] [LOGGED MESSAGE] **Ключевая строка:** … Found supported Program CA descriptor ca_system_id: 0x96b, ecm_pid: 0x217 … **Регулярное выражение для поиска:** [\w\d\[\]\._\:]+ Found supported [\w]+ ca_system_id: \w+, ecm_pid: \w+	CAS (Conditional Access System) подтвердила доступ к потоку данных вызываемого канала для устройства.

Согласно данным, приведенным в табл. 2, был настроен Logstash таким образом, чтобы лог файлы устройства преобразовывались в CSV файл со следующей структурой: TIMESTAMP – метка времени события, DEVICE_ID – MAC адрес устройства, CHANNEL_ID – идентификатор ТВ канала, на который происходит переключение, APP_STATE – идентификатор события, соответствующий порядковому номеру строки из табл. 2:

TIMESTAMP,DEVICE_ID,CHANNEL_ID,APP_STATE

11.11.2020 20:02:29.046, 403DEC73B458,TLCHD,1

11.11.2020 20:15:02.112, 403DEC73B458, TLCHD,2

11.11.2020 20:17:13.032, 403DEC73B458, TLCHD,3

11.11.2020 20:28:39.254, 403DEC73B458, TLCHD,1

11.11.2020 20:28:41.002, 403DEC73B458, TLCHD,2

11.11.2020 20:34:01.146, 403DEC73B458, TLCHD,3…

Данные о полученных сценариях поведения приложений были переданы для последующего использования в систему мониторинга телекоммуникационной сети, построенную на базе графа знаний (ГЗ) [9].

Для загрузки данных в ГЗ, CSV файл был преобразован в формат RDF/XML:

```
<rdf:Description rdf:about = 'http://127.0.0.1/Event_1/'>
<my:request_timestamp rdf:datatype = 'http://www.w3.org/2001/XMLSchema#datetime'>2020-11-11T20:02:29.046</my:request_timestamp>
<my:has_req_type>STATE_CHANGE</my:has_req_type>
<my:has_state>1</my:has_state>
<my:request_detailes>
<rdf:Description>
<rdf:type>:statement</rdf:type>
<rdf:predicat>:new_state</rdf:predicat>
<rdf:subject>Watch_TV</rdf:subject>
<rdf:object><rdf:Description rdf:about = 'http://127.0.0.1/TLCHD/'></rdf:Description></rdf:object>
</rdf:Description>
</my:request_detailes>
<my:uses_dev>
<rdf:Description rdf:about = 'http://127.0.0.1/Device_1/' >
</rdf:Description>
</my:uses_dev>
</rdf:Description>

<rdf:Description rdf:about = 'http://127.0.0.1/Event_2/'>
<my:request_timestamp rdf:datatype = 'http://www.w3.org/2001/XMLSchema#datetime'>2020-11-11T20:15:02.112</my:request_timestamp>
<my:has_req_type>STATE_CHANGE</my:has_req_type>
<my:has_state>2</my:has_state>
<my:request_detailes>
<rdf:Description>
<rdf:type>:statement</rdf:type>
<rdf:predicat>:new_state</rdf:predicat>
<rdf:subject>Watch_TV</rdf:subject>
<rdf:object><rdf:Description rdf:about = 'http://127.0.0.1/TLCHD/'></rdf:Description></rdf:object>
</rdf:Description>
</my:request_detailes>
<my:uses_dev>
<rdf:Description rdf:about = 'http://127.0.0.1/Device_1/' >
</rdf:Description>
</my:uses_dev>
</rdf:Description>
…
```

Данные о сценариях поведения приложений далее могут быть получены из ГЗ путем выполнения SPARQL запросов. Пример запроса и ответа для получения сценариев поведения приложения переключения каналов приведен ниже:

SPARQL запрос.

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX my: <http://127.0.0.1/bg/ont/test1#>
SELECT *
WHERE
{
  ?Event_ID my:uses_dev <http://127.0.0.1/Device_A0722CEEC888/> .
  ?Event_ID my:request_timestamp ?timestamp .
  ?Event_ID my:has_state ?y
 }
```

Ответ (фрагмент):

Event_ID	timestamp	State
<http://127.0.0.1/Event_0/>	11.11.2020T20:02:32.279	1

<http://127.0.0.1/Event_1/>	11.11.2020T20:02:32.467	2
<http://127.0.0.1/Event_10/>	11.11.2020T20:03:03.664	1
<http://127.0.0.1/Event_11/>	11.11.2020T20:03:08.332	2
<http://127.0.0.1/Event_12/>	11.11.2020T20:03:09.784	1
<http://127.0.0.1/Event_13/>	11.11.2020T20:03:15.888	1
<http://127.0.0.1/Event_14/>	11.11.2020T20:03:22.036	1
<http://127.0.0.1/Event_15/>	11.11.2020T20:03:25.175	2

Файлы настройки Logstash, исходные лог файлы, CSV, RDF/XML файлы доступны в открытом репозитории на GitHub [10].

Выводы

В статье рассмотрен метод поиска паттернов в лог файлах телекоммуникационных устройств для анализа их состояния. Предложенный метод предусматривает использование высокопроизводительного решения для анализа и обработки лог файлов в режиме реального времени Logstash и загрузку найденных сценариев поведения приложений в граф знаний, позволяющий использовать их в дальнейшем при решении задач мониторинга. Представлен практический пример для сети оператора кабельного ТВ. Приведенный пример показал, что предложенный метод может использоваться для решения практических задач.

Список литературы

[1] Splunk. http://www.splunk.com/

[2] Sumo Logic. https://www.sumologic.com/

[3] Log Management Explained. https://www.loggly.com/log-management-explained/

[4] LogEntries. https://logentries.com/doc/

[5] Hamooni, H. et al. “LogMine: Fast Pattern Recognition for Log Analytics” Proceedings of the 25th ACM International on Conference on Information and Knowledge Management (2016): n. pag.

[6] Logstash. https://www.elastic.co/logstash

[7] M. Bajer, “Building an IoT Data Hub with Elasticsearch, Logstash and Kibana”, 2017 5th International Conference on Future Internet of Things and Cloud Workshops (FiCloudW), Prague, 2017, pp. 63-68, doi: 10.1109/FiCloudW.2017.101.

[8] STB. https://en.wikipedia.org/wiki/Set-top_box

[9] Kulikov I., Wohlgenannt G., Shichkina Y., Zhukova N. (2020) An Analytical Computing Infrastructure for Monitoring Dynamic Networks Based on Knowledge Graphs. In: Gervasi O. et al. (eds) Computational Science and Its Applications – ICCSA 2020. ICCSA 2020. Lecture Notes in Computer Science, vol 12254. Springer, Cham. https://doi.org/10.1007/978-3-030-58817-5_15

[10] GitHub relository link. https://github.com/kulikovia/SCM-2021