VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

MINISTERSTVO VNITRA ČESKÉ REPUBLIKY

Příjemci podpory:

Vysoké učení technické v Brně
Fakulta informačních technologií

Poskytovatel:

Ministerstvo vnitra ČR

***Integrovaná platforma pro zpracování digitálních dat z bezpečnostních incidentů* (TARZAN)**

Identifikační kód VI20172020062

Název předkládaného výsledku: dist-forensic-digital-data-repo

Typ výsledku dle UV č. 837/2017	Evidenční číslo (příjemce)	Rok vzniku
R software		**2018**
ISBN-ISSN	**Webový odkaz na výsledek**	**Kde a kdy publikováno**
	https://www.fit.vut.cz/research/product/585/	

Anotace k výsledku:

Distribuované úložiště pro digitální forenzní data s repozitářem dat a metadat, API pro dotazy a příchozí/odchozí data, indexováním, plug-in systémem pro dosud nepodporované typy dat atd. Projekt se skládá ze čtyř komponent: komunikační modul založený na Apache Kafka; module pro persistenci, tj. trvalé úložiště, založený na Apache Cassandra pro data a MongoDB pro metadata; modul distribuovaného repositáře (repozitářový server); a modul demo klienta repozitáře, který produkuje a do repositáže ukládá PCAP data.

Řešitelský tým: *Petr Matoušek (manažer a hlavní řešitel), Marek Rychlý (realizační tým)*

dist-forensic-digital-data-repo

Distributed Forensic Digital Data Repository

Technical Documentation

RNDr. Marek Rychlý, Ph.D.
Ing. Martin Josefik

Tarzan Project Documentation
Brno Univeristy of Technology, Faculty of Information Technology

Distributed Forensic Digital Data Repository
Technical Documentation

Marek Rychlý
Martin Josefík

Brno University of Technology
rychly@fit.vutbr.cz, xjosef00@stud.fit.vutbr.cz

Abstract. Distributed storage for digital forensic data with data/metadata repository, API for queries and incoming/outgoing data, indexing, plug-in system for yet unsupported data-types, etc. The project consists of four components: Communication module (a communication bus based on Apache Kafka); Persistence module (a persistent storage based on Apache Cassandra for data and MongoDB for metadata); DistributedRepository module (the repository server); and ProducerDemo module (a demo of the repository client – a producer of PCAP data). This document outlines technical aspects of the system.

1 Architecture

The system consists of several modules that are depicted in Fig. 1.

The distributed repository is storing data in a distributed filesystem based on HDFS, while metadata (e.g., indices) are stored in distributed databases MongoDB and Cassandra. MongoDB is utilized to store a structured data required by the distributed storage itself, while Cassandra is used to store metadata and extracted data required by clients (i.e., data and their indices).

To extract the data and obtain their metadata, the system utilizes Pcap4J library.

As a communication platform between the distributed repository and its clients, Kafka message broker is employed.

2 Implementation

A client can connect to the distributed repository via Kafka message queue as depicted in Fig. 2.

After the connection, the client can control the repository by commands sent via the message queue. The list of commands is available in the User Guide.

3 Acknowledgements

This work was supported by the Ministry of the Interior of the Czech Republic as a part of the project Integrated platform for analysis of digital data from security incidents VI20172020062.

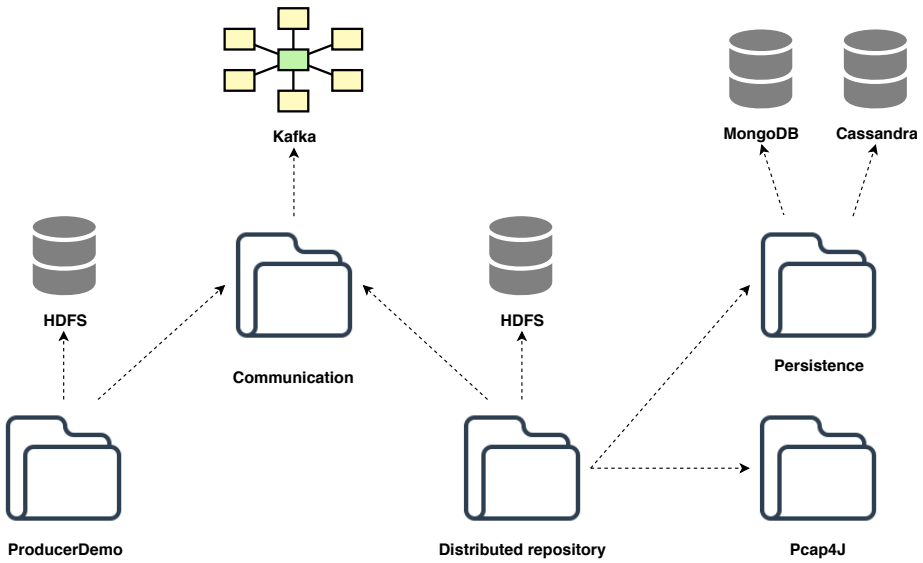

Fig. 1. The architecture of the system.

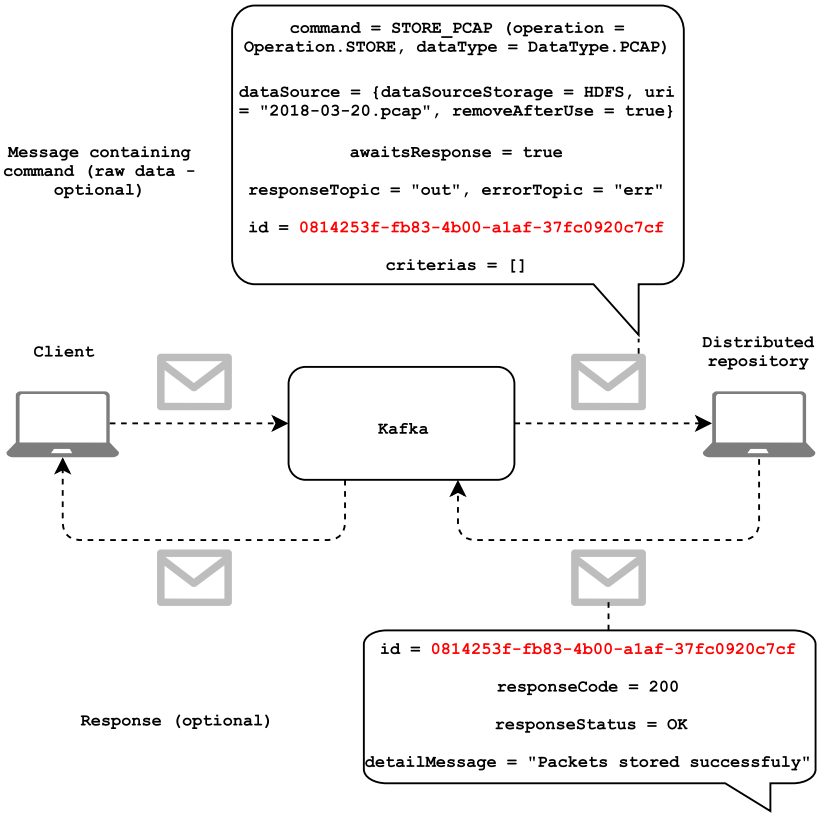

Fig. 2. The interaction of a client and the system via Kafka message queues.

dist-forensic-digital-data-repo

Distributed Forensic Digital Data Repository

Installation Guide

RNDr. Marek Rychlý, Ph.D.
Ing. Martin Josefik

Tarzan Project Documentation
Brno Univeristy of Technology, Faculty of Information Technology

Distributed Forensic Digital Data Repository
Installation Guide

Marek Rychlý
Martin Josefík

Brno University of Technology
rychly@fit.vutbr.cz, xjosef00@stud.fit.vutbr.cz

Abstract. Distributed storage for digital forensic data with data/metadata repository, API for queries and incoming/outgoing data, indexing, plug-in system for yet unsupported data-types, etc. The project consists of four components: Communication module (a communication bus based on Apache Kafka); Persistence module (a persistent storage based on Apache Cassandra for data and MongoDB for metadata); DistributedRepository module (the repository server); and ProducerDemo module (a demo of the repository client – a producer of PCAP data). This document provides an installation guide to the system.

1 Requirements

Docker is required to be installed and running.

2 Infrastructure

To pull and build required Docker images and start the Docker containers infrastructure based on the images, run the following statements:

```
cd dist-forensic-digital-data-repo/docker/Environment
docker-compose pull
docker-compose build
docker-compose up
```

3 Application

To build and install the application modules, run the following statements:

```
cd dist-forensic-digital-data-repo/app
for I in Communication Persistence
    DistributedRepository ProducerDemo; do
  cd ${I}
  ./install.sh || break
  cd -
done
```

To run the application the Distributed Repository, you need to:

```
cd dist-forensic-digital-data-repo
cd app/DistributedRepository
./run.sh
```

4 Test

To run the Producer Demo which is demonstration the usage of the Distributed Repository:

```
cd dist-forensic-digital-data-repo/app/ProducerDemo
./run.sh
```

5 Acknowledgements

This work was supported by the Ministry of the Interior of the Czech Republic as a part of the project Integrated platform for analysis of digital data from security incidents VI20172020062.

dist-forensic-digital-data-repo

Distributed Forensic Digital Data Repository

User Guide

RNDr. Marek Rychlý, Ph.D.
Ing. Martin Josefik

Tarzan Project Documentation
Brno Univeristy of Technology, Faculty of Information Technology

Distributed Forensic Digital Data Repository
User Guide

Marek Rychlý
Martin Josefík

Brno University of Technology
rychly@fit.vutbr.cz, xjosef00@stud.fit.vutbr.cz

Abstract. Distributed storage for digital forensic data with data/metadata repository, API for queries and incoming/outgoing data, indexing, plug-in system for yet unsupported data-types, etc. The project consists of four components: Communication module (a communication bus based on Apache Kafka); Persistence module (a persistent storage based on Apache Cassandra for data and MongoDB for metadata); DistributedRepository module (the repository server); and ProducerDemo module (a demo of the repository client – a producer of PCAP data). This document provides a user guide to the system.

1 Commands

To control the Distributed Repository, clients can use a Kafka message queue published by the repository.

1.1 Request

Each message sent by a client to the repository must consist of:

- command – Specifies the type of command which is one of STORE PCAP and LOAD PCAP. The command also encapsulates the type of operation, which can be SAVE and LOAD, as well as the type of data such as PCAP, Packet, BINARY, LOG, etc.
- id – Mandatory parameter, specifies a unique message ID so that the client can identify the already processed orders.
- awaitsResponse – Two-state value if the client/sender of the message expects a response from the repository. If this parameter is specified as True, another "responseTopic" parameter must also be specified.
- responseTopic – Specifies the name of the queue (topic) in which the response on the client side is expected.
- errorTopic – An error may occur on the distributed storage side, such as database unavailability, command processing error, and more. This parameter is used to specify the queue to which error messages will be sent. This is a required parameter.

– dataSource – Source data to be stored can be sent in two ways. The first is to send data in binary form directly via the Kafka system together with the command. This method can be used for data that is not too large because the data is loaded into RAM. Such an approach is not suitable for large volumes of data. Therefore, there is a second way, and that is to pass data over a distributed HDFS file system. The "dataSource" parameter contains the name of the repository in the form of an enumeration constant of Kafka or HDFS, the path to the file if it is HDFS, and also the "removeAfterUse" attribute to remove the file after use. It can also be used for data reading commands to transfer the result, via HDFS or Kafka.
– criterias – Represents a list of criteria for querying. This parameter should be filled in only when sending a read command.

1.2 Reply

After processing a request by the Distributed repository, the system sends a reply, if requested (see the "awaitsResponse" parameter of the client's request above). The reply consists of:

– id – This is a unique ID copied from the request message. After receiving the response, the client will know to which request the response belongs.
– responseTopic – The name of the output queue to which the response is sent.
– responseCode – The response return code symbolizing how the operation turned out. Analogous with HTTP return codes, possible values are: OK (200), BAD REQUEST (400), UNSUPPORTED MEDIA TYPE (415), INTERNAL SERVER ERROR (500).
– status – Reserved parameter for response status.
– detailMessage – If an error occurs during the processing of the request, the answer can be sent why the error occurred or its cause.

2 Demonstration Example

To develop a client application, which uses the Kafka message queue to control the repository, see the Producer Demo application.

3 Acknowledgements

This work was supported by the Ministry of the Interior of the Czech Republic as a part of the project Integrated platform for analysis of digital data from security incidents VI20172020062.

dist-forensic-digital-data-repo

Distributed Forensic Digital Data Repository

Reference Documentation

RNDr. Marek Rychlý, Ph.D.
Ing. Martin Josefik

Tarzan Project Documentation
Brno Univeristy of Technology, Faculty of Information Technology

Distributed Forensic Digital Data Repository
Reference Documentation

Marek Rychlý
Martin Josefík

Brno University of Technology
rychly@fit.vutbr.cz, xjosef00@stud.fit.vutbr.cz

Abstract. Distributed storage for digital forensic data with data/metadata repository, API for queries and incoming/outgoing data, indexing, plug-in system for yet unsupported data-types, etc. The project consists of four components: Communication module (a communication bus based on Apache Kafka); Persistence module (a persistent storage based on Apache Cassandra for data and MongoDB for metadata); DistributedRepository module (the repository server); and ProducerDemo module (a demo of the repository client – a producer of PCAP data). This document introduces a reference documentation of the system.

1 Technical Report

For the reference documentation, see Master's Thesis ([1], in Czech) which serves as a technical report of the original project and which is still valid, however, no longer updated.

2 Acknowledgements

This work was supported by the Ministry of the Interior of the Czech Republic as a part of the project Integrated platform for analysis of digital data from security incidents VI20172020062.

References

1. Josefík, M.: *Distributed Forensic Digital Data Repository* . Master's thesis, Brno University of Technology, Faculty of Information Technology, 2018.
URL https://www.fit.vut.cz/study/thesis/20639/