Big Data for Monitoring Mobile Applications

Fabrice MOURLIN **Lahlou Guy DJIKEN** **Laurent NEL**

Presented by:

Lahlou Guy DJIKEN from **University of Douala**, **Cameroon**

Email: **gdjiken@fs-univ-douala.cm**

UNIVERSITÉ PARIS-EST

April 2022

The Eighth International Conference on Big Data, Small Data, Linked Data and Open Data, ALLDATA 2022 Barcelona, Spain

Home Page

- Guy Lahlou Djiken is a lecturer and researcher in the Applied Computing Laboratory of University of Douala and the Laboratory of Algorithm, Complexity and Logic (LACL);
- He is interested in mobility in communication systems more precisely in Communication Networks and Services, Big Data and Artificial Intelligence, Paravirtualization and IoT;
- The interoperability of the above fields of interest is at the core of this current research;
- One of the axes of exploration is the impact of paravirtualization in order to accelerate the inference of programs based on Artificial Intelligence given the lack of network infrastructures.

Outline

- Introduction
- Log analysis and Reporting of Artificial Intelligence
- Use case for the anomaly prediction
- Software architecture and Data streaming
- Artificial Intelligence Model and Results
- Conclusion and Future Work

Introduction

- ✓ Supervision of mobile application
- ✓ Log messages and log analysis
- ✓ Anomaly prediction and difficulties associated
- ✓ Standardization of formats during the data collection
- ✓ Mobile applications and using for the taking picture of experiments in biology

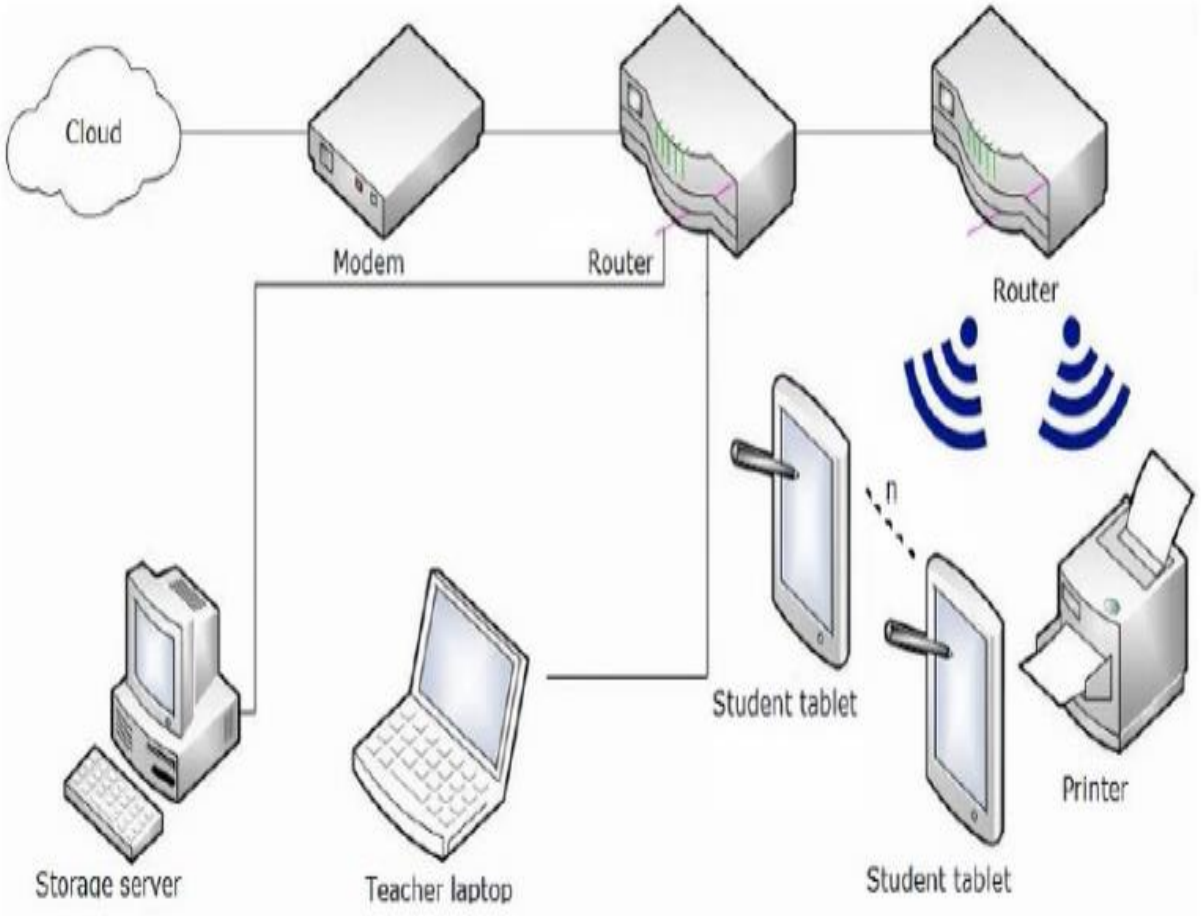
Log analysis and reporting of AI

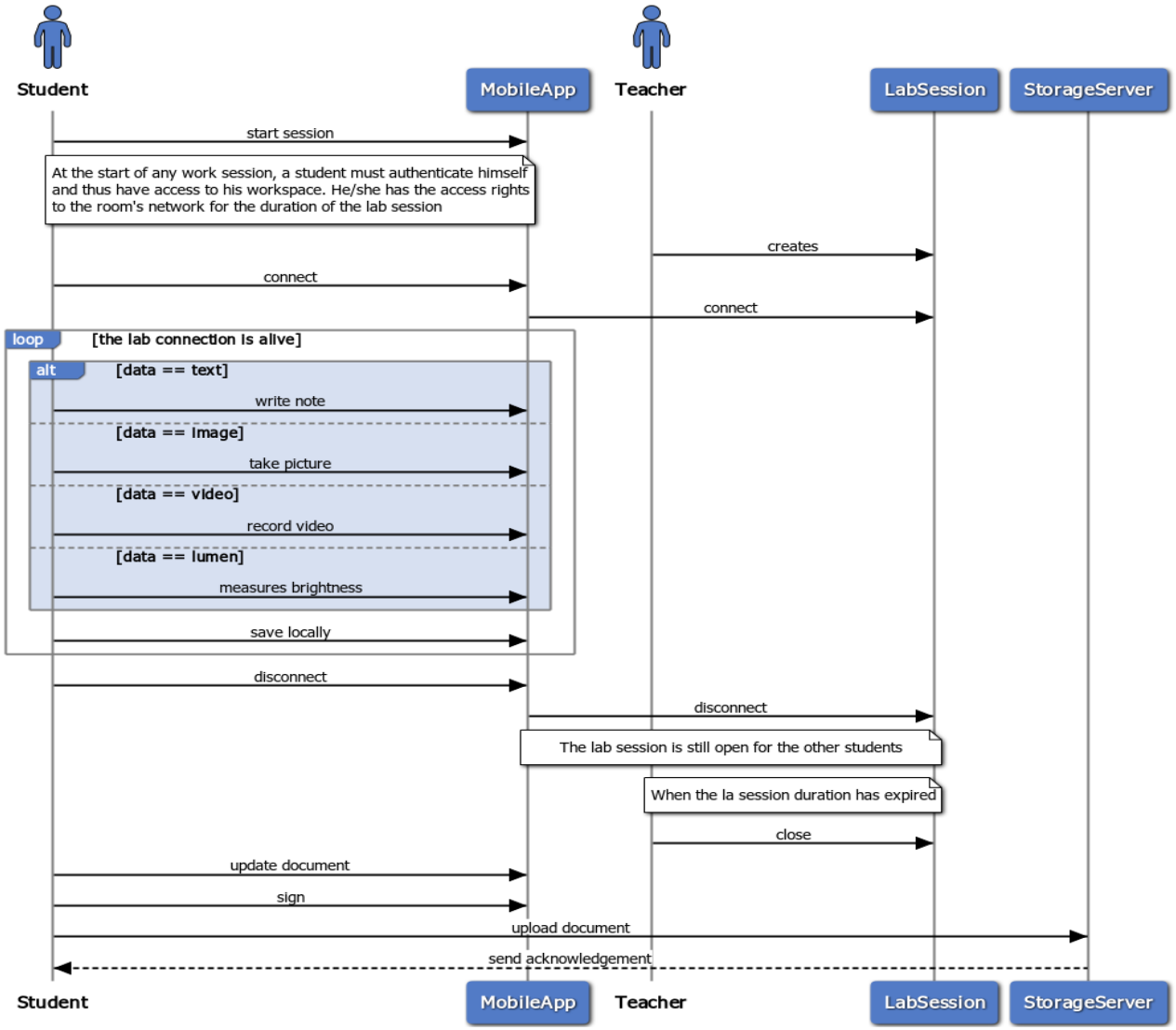
- ✓ Software monitoring and log analysis of distributed systems
- ✓ Development of Machine Learning and impacts on the use of logs
- ✓ Use of predictive machine learning models
  - Wei Luo and rulebook for AI model development
  - Henderson and Stevens for reporting
  - Dos Santos and reproductible analysis approach

Use case for the anomaly prediction

- ✓ Context description: analysis of an experiment in a laboratory
- ✓ Principes and nominal scenario during an experiment in a lab room
  - Student
  - MobileApp
  - Teacher
  - LabSession
  - StorageServer

Network and sequence diagram





Software architecture (1/3)

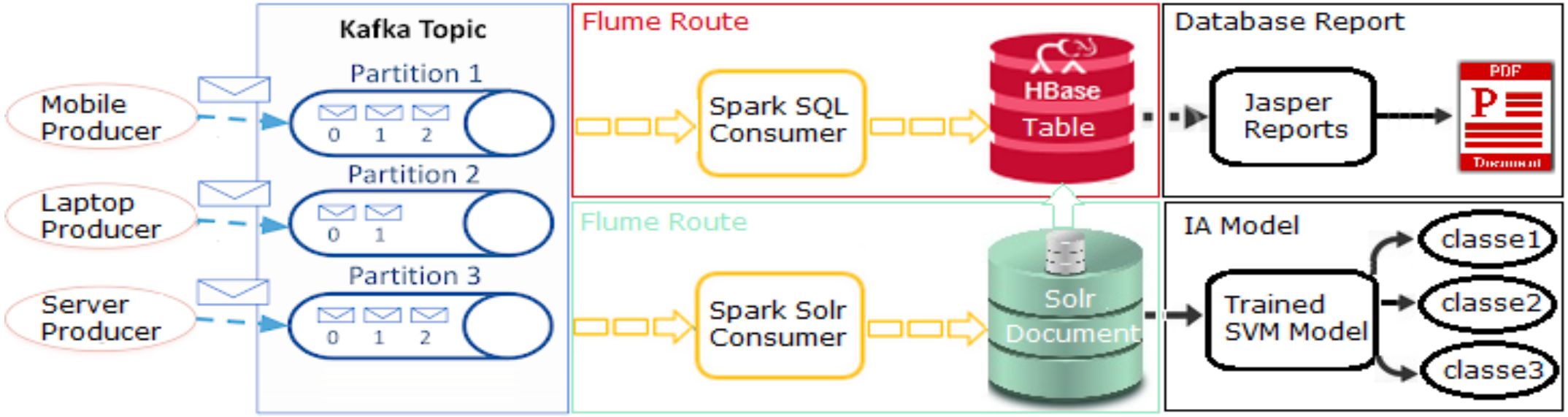
- ✓ The analysis of logs is more complex
  - the arrival of log data continuously
  - the need to impose a data schema to index the information
  - refine the search for information and the detection of anomalies
- ✓ Our goal is to collect and cross-reference information from the various sources
- ✓Thus, it is essential to monitor the events related to the management of the laboratory sessions.
- ✓In addition, any event related to an information capture or modification is useful.

Software architecture (2/3)

- ✓ The server application part is deployed on the storage server
- ✓ This part is developed with the Spring Boot library
- ✓ We use intensively the Spring configuration for the logs, but also for the persistence aspects
- ✓ The database is Postgresql version 10
- ✓ We focuse on describing our Big Data workflow from collection to building our AI model

Software architecture (3/3)

- ✓ Big Data architecture : Data collection and Big Data analysis
- ✓ Layer components of our project:
  - Kafka: kafka for topics partition
  - Flume: FlumeRoute (Spark SQL and Spark Solr) Consumer
  - Database: Jasper Reports
  - IA Model: SVM model by classifiers



Data streaming (1/3)

- ✓ Spark SQL Consumer : Kafka receiver class → DStream
- ✓ Spark SQL Consumer component aims to apply name conventions and a common structure
- ✓ We have defined a mapping between HBase and Spark tables, called Table Catalog
  - *The row key definition implies the creation of a specific key generator in our component.*
  - *The mapping between table column in Spark and the column family and column qualifier in HBase needs a declarative name convention*
- ✓ The HBase sink exploits the parallelism on the set of Region servers

Data streaming (2/3)

- ✓ The pipeline is built with Apache Spark and Apache Spark Solr connector
- ✓ Spark framework is used for distributed in memory compute, transform and ingest to build the pipeline
- ✓ The Apache Solr collection, which plays the role of a SQL table, is configured with shards
- ✓ The definition of shard is based on the number of partitions and the replicas rate for fault tolerance ability
- ✓ The Spark executors run a task, which transforms and enriches each log message

Data streaming (3/3)

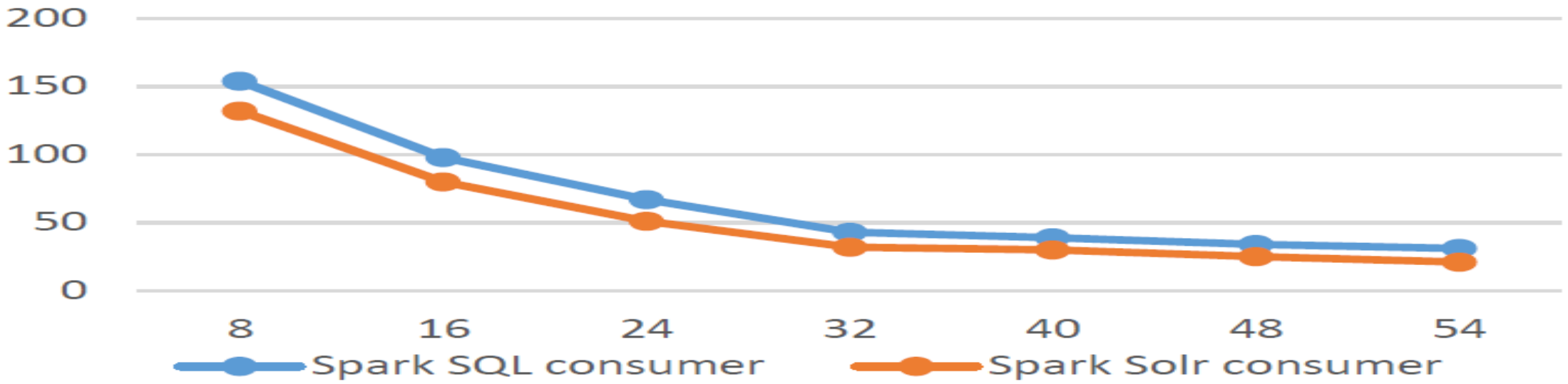
- ✓ Solr Cloud as a data source Spark is used when we create our ML model
- ✓ We send requests from Spark ML classes and read results from Solr with the use of Solr Resilient Distributed Dataset (SolrRDD class).
- ✓ With Spark SQL, we expose the results as SQL tables in the Spark session
- ✓ These data frames are the base of our ML model construction.
- ✓ The metrics called Term Factor (TF) and Inverse Document Frequency (IDF) are key features for the ML model.

AI Model and Results (1/3)

- ✓The concepts behind SVM algorithm are relatively simple
- ✓The classifier separates data points using a hyperplane with the largest amount of margin
- ✓In our working context, the margin between log patterns is a suitable discriminant
- ✓SVM offers very high accuracy compared to other classifiers such as logistic regression, and trees

AI Model and Results (2/3)

- ✓At runtime for our data set based on a unique log format, the cost of Spark SQL consumer decreases when the partitioning of dataset increases
- ✓We have to oversize the partitions and the gains are much less interesting



AI Model and Results (3/3)

- ✓ The analytical expression of the features precision, recall of retrieved log messages that are relevant to the find
- ✓ Our results for four classes are within acceptable ranges of values for the use of the model to be accepted

Class number	**Metrics**		
	Precision by label	***Recall by label***	***F1 score by label***
0.000000	0.815846	0.890100	0.896616
1.000000	0.911000	0.981000	0.991000
2.000000	0.854461	0.714857	0.851481
3.000000	0.852446	0.7589148	0.833129

Conclusion and Future Work (1/2)

- ✓ Our approach on log analysis and maintenance task prediction
  - We showed how an index engine is crucial for a suitable query engine
  - We have developed specific plugin for customizing the field types of our documents, but also for filtering the information from the log message
  - We have stressed the key role of our Spark components, one per data source
- ✓ We observed that our approach supported a large volume of logs
- ✓ From the filtered logs, we presented the construction of our SVM model based on work from the Center for Pattern Recognition and Data Analytics
- ✓ Our study also shows the limits that we want to push back, such as the management of log patterns

Conclusion and Future Work (2/2)

- ✓The indexing process based on a custom schema
- ✓We think that the use of DisMax query parser could be more suitable in log requests where messages are simple structured sentences
- ✓We want to extract dynamically the log format instead of the use of a static definition
- ✓We think also about malicious messages, which can perturb the indexing process and introduce bad request in our prediction step
- ✓The challenge needs to manage a set of malicious patterns and the quarantine of some message sequences

References

1. A. Oliner, A. Ganapathi, and W. Xu, “Advances and challenges in log analysis,” Communications of the ACM, 2012, 2nd ed., vol. 55, pp. 55-61
2. A. Juvonen, T. Sipola, and T. Hämäläinen, “Online anomaly detection using dimensionality reduction techniques for HTTP log analysis,” Computer Networks 91, pp. 46-56, November 2015.
3. M. Assefi, E. Behravesh, G. Liu, and A. P. Tafti, “Big data machine learning using apache Spark MLlib,” 2017 IEEE International Conference on Big Data (Big Data), 2017, pp. 3492-3498
4. T. D. Nguyen, V. Nguyen, T. Le, and D. Phung, “Distributed data augmented support vector machine on Spark,” 23rd International Conference on Pattern Recognition (ICPR), 2016, IEEE.

Thank you for your attention

UNIVERSITÉ PARIS-EST