Distributed Search on a Large Amount of Log Data

Fabrice Mourlin **Charif Mahmoudi** **Lahlou Guy Djiken**

Presented by:

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

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

UNIVERSITÉ PARIS-EST

IARIA

April 2021

The Seventh International Conference on Big Data, Small Data, Linked Data and Open Data, ALLDATA 2021 Portugal

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
- Importance of Log Data and Big Data
- Use Case for Monitoring with Artificial Intelligence
- Software Architecture and Tools
- Big Data Streaming and Results
- Conclusion and Future Work

Introduction

- ✓ Log and file system
- ✓ Challenge and tools in supervision
- ✓ Search for a pattern of behavior is more effective and prevention becomes better and more reactive
- ✓ Log analysis solutions incorporate additional data sources

Importance of Log Data and Big Data

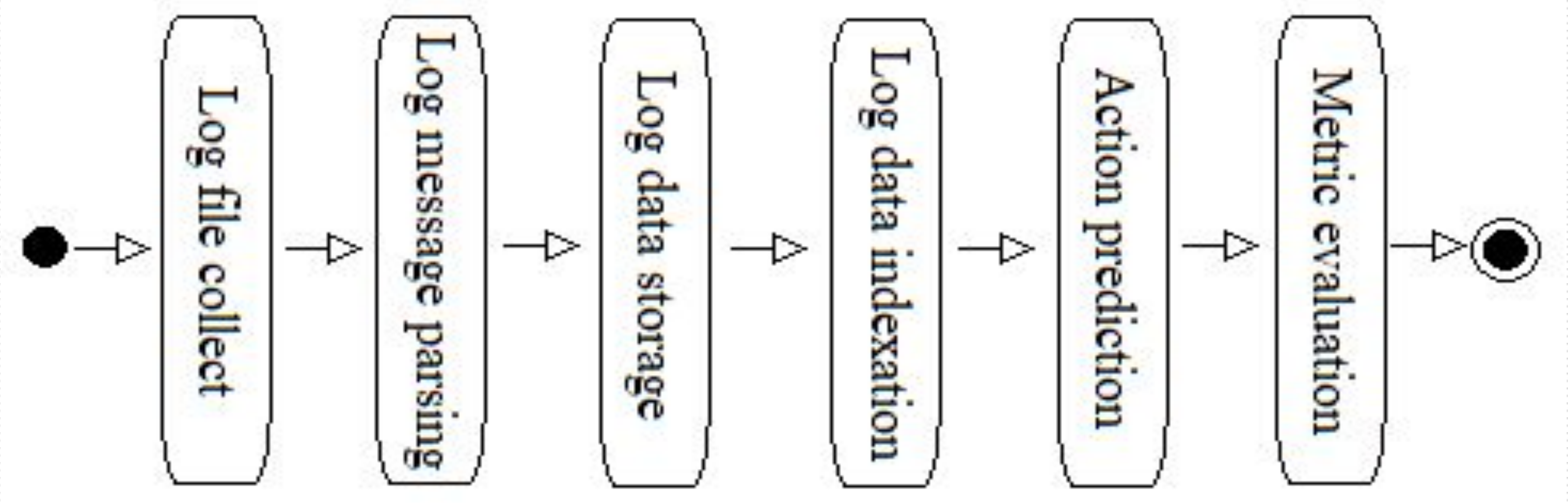
- ✓ Usage of logs
- ✓ Use of logging has been common practice in IT for many years
- ✓ Logs for intervention and prediction
- ✓ Structure of logs and impact on the analysis strategy

Approaches in the Litterature

- Publication and related works
  - Qiang Fu
  - W. Xu’s
  - Chinghway
  - Jakub Breier
  - …
- Techniques and platforms
  - Apache Hadoop
  - Data mining – K mean Model – Deep Learning
  - Loglens

Use Case for Monitoring with Artificial Intelligence

- Monitoring activities of information systems
- Application servers and data management servers
- Step for proceeding of monitoring

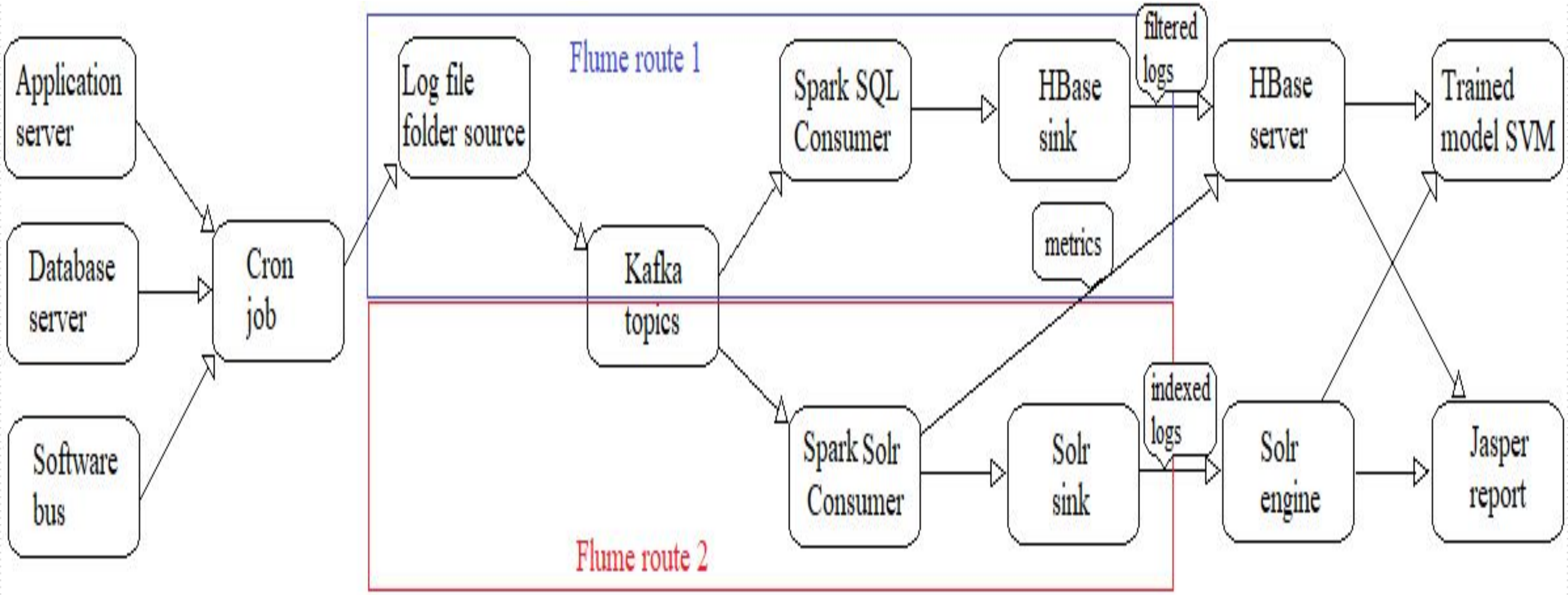


Software Architecture and Tools

- Managing the sequencing of tasks on the analysis platform
- Hadoop ecosystem and set of software to process huge data sets
  - Distributed analytical frameworks
  - Zookeeper clustering tool
  - Hadoop Distributed File System (HDFS)
- Solr Framework 8.1 and different components

Big Data Workflow for Log Analysis

- This architecture shows two Flume routes
  - Spark SQL with Hbase sink that transites in a Kafka channel
  - Spark Solr with Solr sink that transites in topic



Big Data Platform

- HBase is a highly reliable data store, supporting disaster recovery and cross-datacenter replication
- Solr Cloud is the indexing and search engine
- Jasper Report tool allows us to build report from data automatically and regularly

Configuration (1/2)

- Via operating system:
  - □ Define specific configuration scripts for routing log files to the log file folder
  - □ Utilization the entries in cron tables
- Via event streaming-tools:
  - □ The Flume and Kafka tools are both tools
  - □ Kafka API for control on the management of messages associated
- Via persistent storage:
  - □ Reduce the number in separate HFiles
  - □ Load time and reduce disk consumption

Configuration (2/2)

- Via indexing engine:
  - Apache Solr is an open source search engine
  - Solr index can be considered as an equivalent of a SQL table
  - Solr installation is distributed on our Big Data
  - Define the structure of the documents that are indexed into Solr
  - Introduce our own parsing strategy via class programming

Component architecture (1/2)

- Based on Spark Framework version 2.4.7
  - □ Spark SQL □ Hbase □ filtered logs
  - □ Spark Solr □ Solr □ indexed logs
- Based on Spring Data
  - □ SolrCloud on the cluster through the same Zookeeper agents
  - □ Spark Solr consumer uses the Spring Data and SolrJ library
  - □ Configuration provides a given analyzer for each Solr

Component Architecture (2/2)

- Based on SolrJ library
  - Built our schema based on our data types
  - Implement new data classes for the new field types
  - Standardize the values present in the logs coming from different servers
  - Utilisation of TokenFilter and TokenFilterFactory
- Based on Spark-MLlib
- Based on Jasper Report library

Big Data Streaming and Results

Big Data Streaming

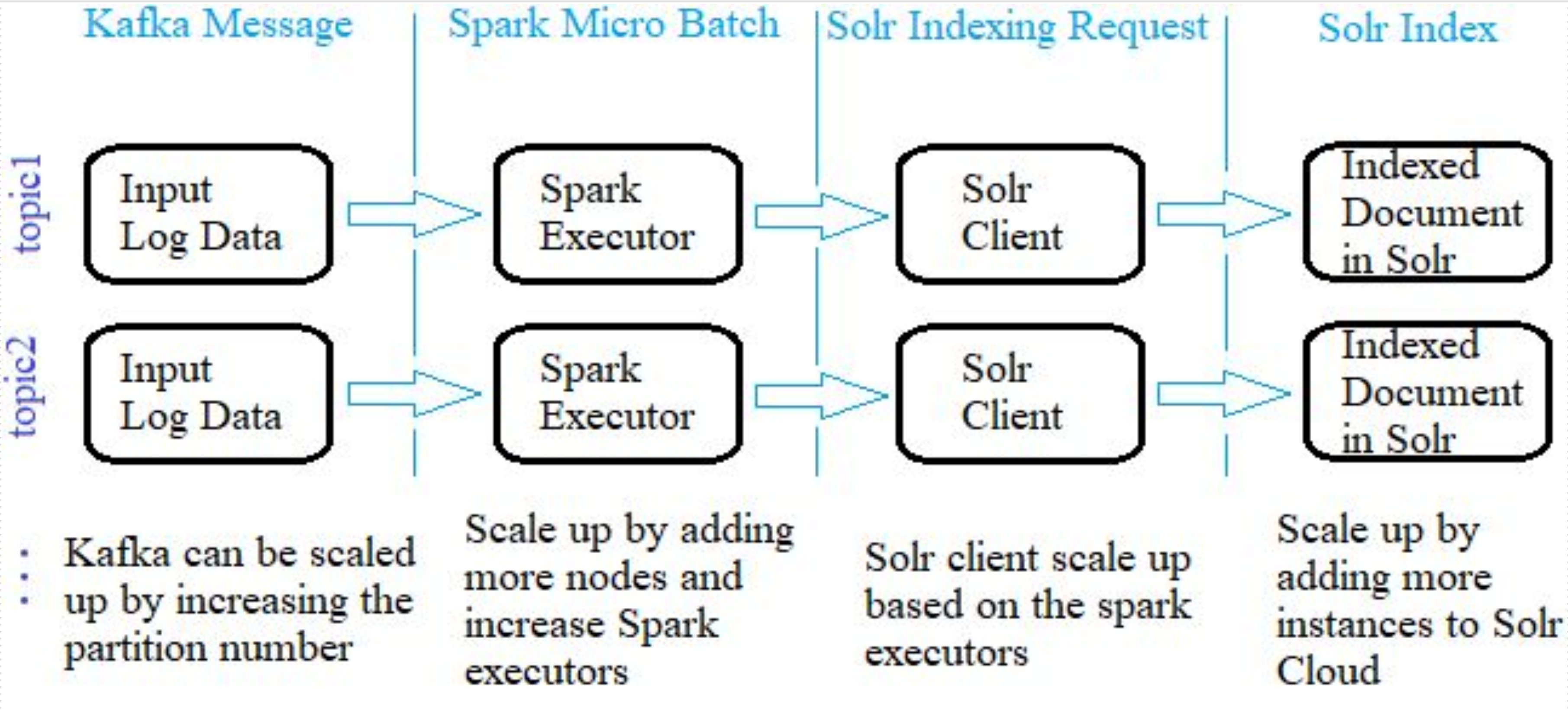
- Apache Kafka as queue system for our log
- Spark streaming is a real-time processing tool that runs on top of the Spark Engine
- The scheduler exploits all the computation resources of ou cluster
- Spark context sends all the tasks for the executors to run
  - □ Filtered log strategy
  - □ Index construction and query

Filtered Log Strategy

- Asynchronous reading
- Normalized form
- Structured data storage
  - The row key definition
    - Implies the creation of specific key generator in our component
  - Mapping between
    - table column in spark and the column family
    - Column qualifier in Hbase needs a declarative name convention

Index Construction and Query

- Index pipeline
- Query process



- Kafka can be scaled up by increasing the partition number

Scale up by adding more nodes and increase Spark executors

Solr client scale up based on the spark executors

Scale up by adding more instances to Solr Cloud

Index Construction and Query

- Data features
  - □ Architecture measurement
  - □ Model measurement

The analytical expression of the features precision, recall of retrieved log messages that are relevant to the find:

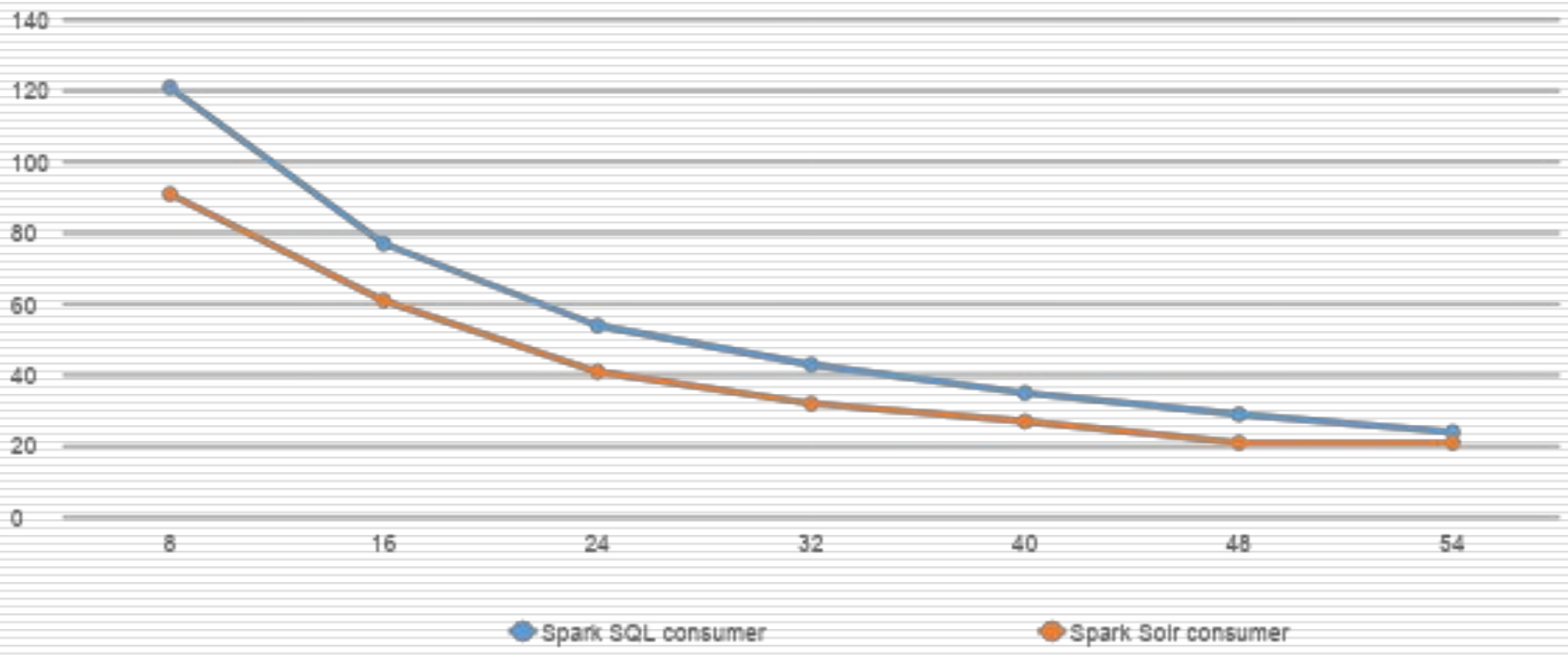
$$precision = \frac{|\{relevant\, \mathrm{log}\, messages\} \cap \{retrieved\ log\ messages\}|}{|\{retrieved\, \mathrm{log}\, messages\}|}$$

$$recall = \frac{|\{relevant\, \mathrm{log}\, messages\} \cap \{retrieved\ log\ messages\}|}{|\{relevant\, \mathrm{log}\, messages\}|}$$

$$F_\beta = (1 + \beta^2) * \frac{precision * recall}{(\beta^2 * precision) + recall}$$

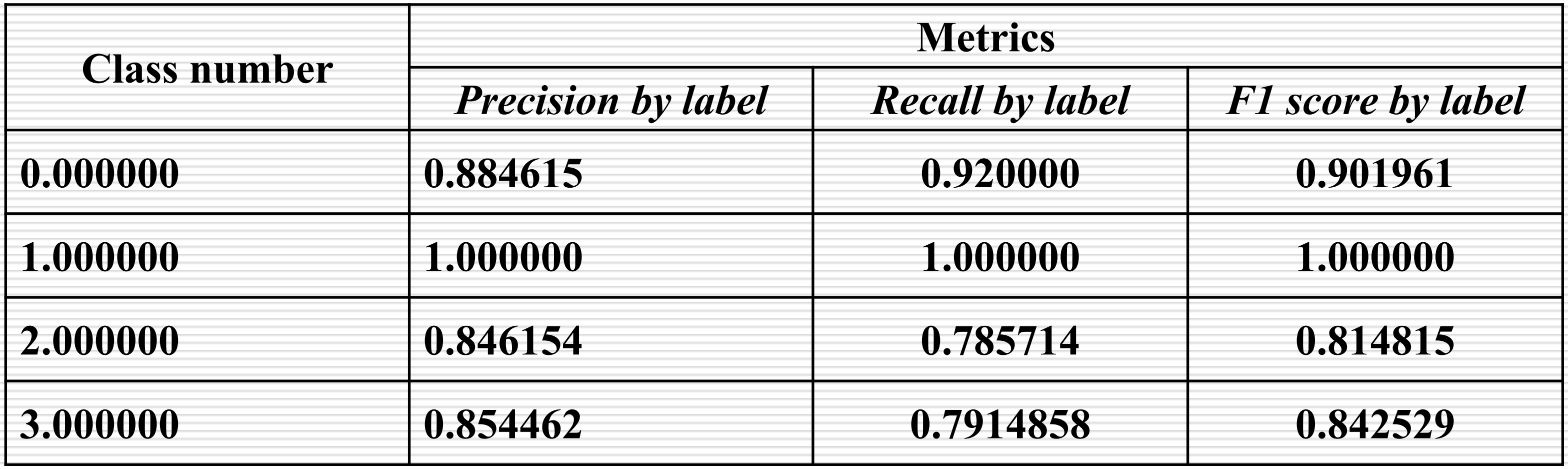
Index Construction and Query

- Data features
  - □ Model measurement



Results and Actions

- Model measurement



Class number	Metrics		
	Precision by label	*Recall by label*	*F1 score by label*
0.000000	0.884615	0.920000	0.901961
1.000000	1.000000	1.000000	1.000000
2.000000	0.846154	0.785714	0.814815
3.000000	0.854462	0.7914858	0.842529

- □ Weighted precision = 0.917402
- □ Weighted recall = 0.918033
- □ Weighted F1 score = 0.917318
- □ Weighted false positive rate = 0.043919

Conclusion and Future Work

- ✓ Our approach on log analysis and maintenance task prediction:
  - Index engine for a suitable query engine
  - Specific plugin for customizing the field type of our documents
  - Information filter from the log message
- ✓ Construction of our SVM model
- ✓ Utilization the AI model for classification log data

Future Work

- ✓ Improvement of the indexing process based on a custom schema
- ✓ Dynamic extraction of the log format instead of the use of static definition
- ✓ Management of malicious messages or patterns

References

1. J. Andersson and U. Schwickerath, “Anomaly Detection in the Elasticsearch Service,” CERN Openlab Summer Student Report, pp. 1-18, 2019.
2. Li, Tao et al., “FLAP : An end-to-end event log analysis platform for system management,” Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1547-1556, 2017.
3. B. Debnath, et al., “Loglens : A real-time log analysis system,” 2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS), IEEE, pp. 1052-1062, 2018.
4. K. Koitzsch, “Advanced Search Techniques with Hadoop, Lucene, and Solr,” Pro Hadoop Data Analytics, Apress, Berkeley, CA, pp. 91-136, 2017

Thank you for your attention