FAKULTÄT FÜR !NFORMATIK
Faculty of Informatics

Diplomarbeitspräsentation

Distributed Systems Group

A Big Data Analytics Framework for Evaluating Automated Elastic Scalability of the SMACK-Stack

Benedikt Wedenik

Masterstudium:
Software Engineering & Internet Computing

Technische Universität Wien
Institut für Informationssysteme
Arbeitsbereich: Distributed Systems Group
Betreuer: Univ.Prof. Mag.rer.soc.oec. Dr.rer.soc.oec.
Schahram Dustdar

Motivation & Research Challenges

In the last years the demand for information availability and shorter response times has been increasing. Today's business requirements are changing: Waiting hours or even days for the result of a query is not acceptable anymore in many sectors. The response needs to be immediate, or the query is discarded [1]. This is why "Fast Data", as an approach to solve those problems, increases its popularity, as being "big data, but fast" [2].

- **Deploying large scale applications**
  Requires multiple instances of different technologies to be deployed in a defined sequence to fulfill subsequent dependencies, often including manual steps.
- **Initial setup**
  The decision of how to configure the instances of an application is a non-trivial task, as there are almost infinite combination possibilities and the impact can be drastic.
- **Monitoring**
  Considering just RAM, CPU and disk usage is in most cases insufficient, as a deeper understanding of the used frameworks is required, which introduces a new layer of complexity.
- **Scaling when needed**
  Understanding what's going on in a cluster and reacting accordingly is crucial for the success of any large scale application.

Contributions

To solve the problems mentioned above, the framework illustrated in Figure 1 has been developed in the course of this thesis, whereas the single components can be described as follows:

- **Automated Scaling Tool for SMACK**
  The scaling tool evaluates the collected metrics from the REST service and scales the individual parts of the SMACK stack up or down.
- **REST Service Collecting Monitoring Information**
  This is the service which collects all the extracted metrics and compiles them into a useful format. In addition, there is the possibility to generate plots at runtime.
- **JMX Extraction Tool**
  This tool is designed to automatically extract interesting metrics from SMACK components via JMX and sending them to a central service, in this case the REST monitoring service.
- **Framework to Easily Launch SMACK in AWS**
  With the help of this framework it takes just a few command line calls to launch and deploy the whole SMACK stack in the cloud.
- **Deployment Blueprints**
  Those reference architecture and configuration recommendations help to launch the SMACK stack and getting most out of the available resource.

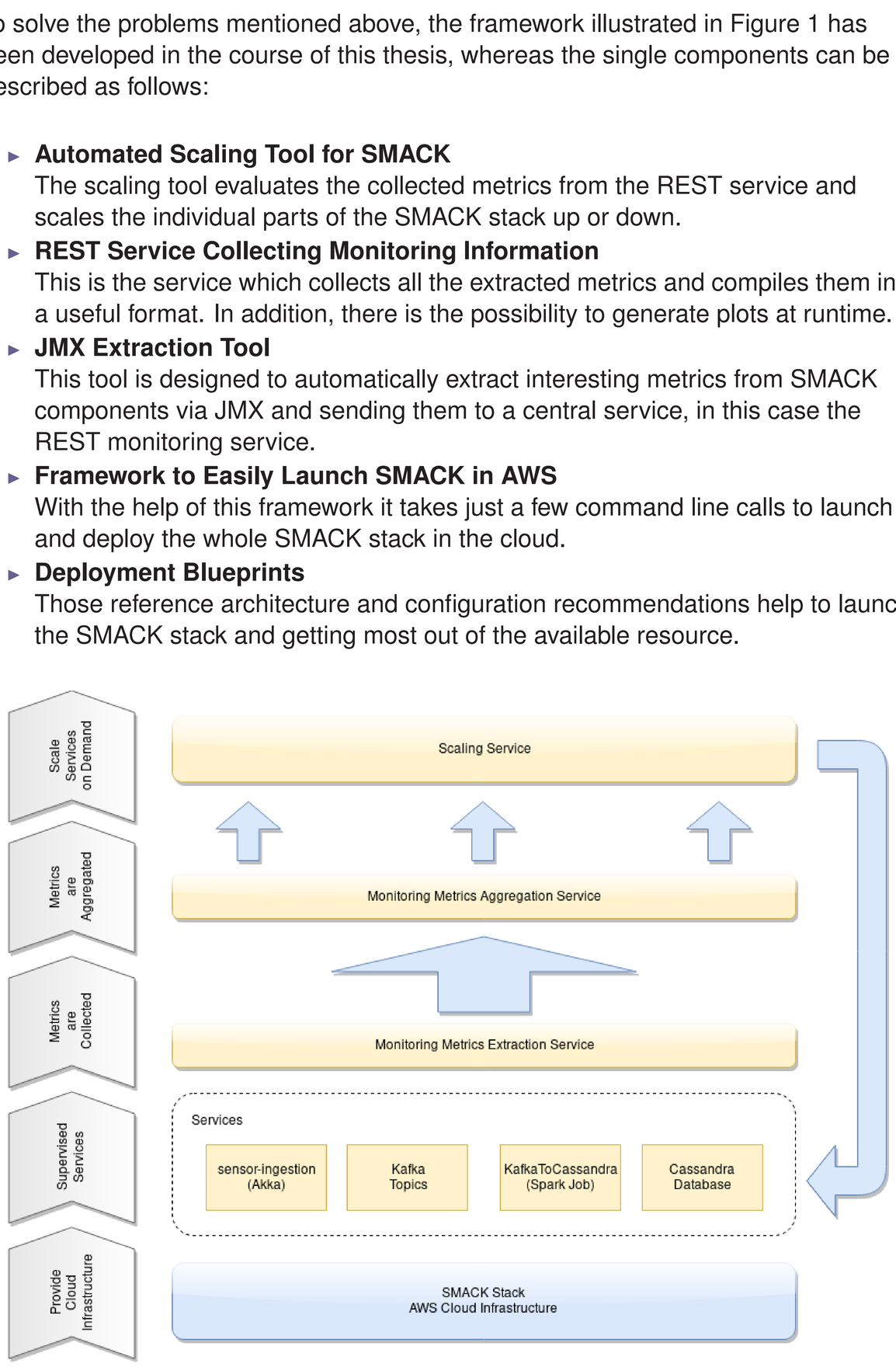

Figure 1: Framework Target Architecture

Background

The SMACK-Stack consists of five technologies combined to a lightning fast data pipeline for today's needs of big data applications, as shown in Figure 2.

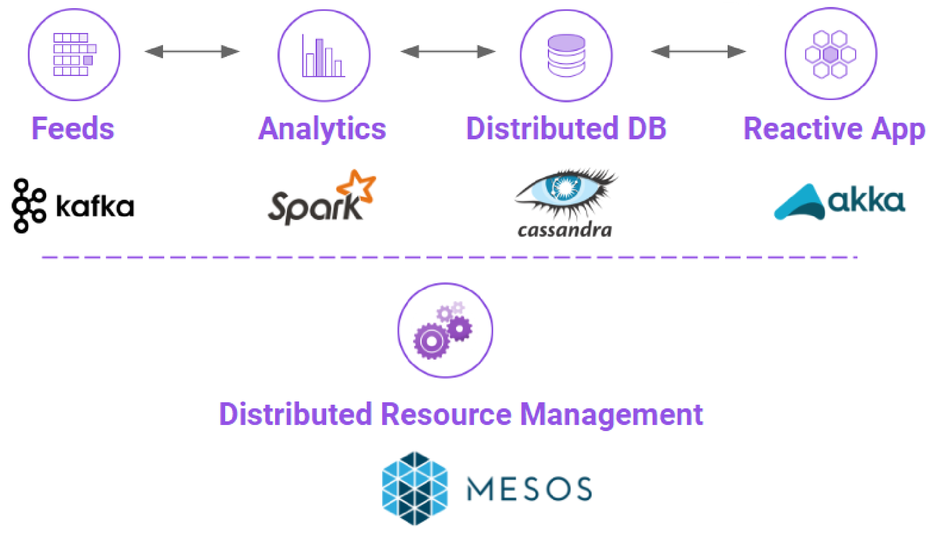

Figure 2: SMACK Stack Illustration, mesosphere.com

- Apache **S**park is the engine of the pipeline, providing batch-, as well as stream-processing power for large-scale data processing.
- **M**esos is a datacenter operating system with the aim to reduce complexity and ease the deployment and maintenance of large-scale distributed applications.
- Apache **A**kka can be seen as the model, providing the possibility to build powerful reactive distributed message-driven applications.
- Apache **C**assandra is a highly distributed database which is a hybrid between a column-oriented and a key-value DBMS, which is implemented avoiding a single point of failure.
- Apache **K**afka serves as publish-subscribe message broker, which is usually the ingestion point of the pipeline.

Results & Conclusion

As part of the contribution, two real world applications have been developed in order to provide a valid base for the evaluation of the framework. The *IoT Data Storage Application* is mainly I/O bound and represents applications which require high throughput. In case of the *Acceleration Prediction Application*, a prediction based on IoT data is performed, which is heavily computational intense.

To evaluate the actual impact of the framework, empirical experiments have been conducted. The setup comprises one run without the framework - i.e. unsupervised, which serves as a baseline, and a supervised one, in which the framework automatically scales the respective services. When analyzing the measured results, both applications benefit from the scaling service, where the following improvements can be highlighted:

- An **increase from 272 to 472 MBit/s** was measured when running the *IoT Data Storage Application*, as well as other **metrics like the time-in-mailbox of a message were improved**.
- The *Acceleration Prediction Application* benefited in form of **shorter message processing times** and an overall **faster task completion**.

References

[1] R. Estrada and I. Ruiz.
Big data, big challenges.
In *Big Data SMACK*, pages 3–16. Springer, 2016.

[2] G. Mishne, J. Dalton, Z. Li, A. Sharma, and J. Lin.
Fast data in the era of big data: Twitter's real-time related query suggestion architecture.
In *Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data*, pages 1147–1158. ACM, 2013.

Kontakt: benedikt.wedenik@alumni.tuwien.ac.at