



Vorhabensbeschreibung zum studentischen Forschungsvorhaben  
im Rahmen des Software Campus

## RHINO

### Efficient Management of Very Large Distributed State for Scalable Stream Processing



<b>Antragssteller:</b>	Technische Universität Berlin
<b>Ansprechpartner:</b>	Bonaventura Del Monte  Technische Universität Berlin Database Systems and Information Management Group Raum EN 741 Einsteinufer 17 10587 Berlin bonaventura.delmonte@tu-berlin.de / +49 30 314 22784
<b>Betreuer (Forschungspartner):</b>	Prof. Dr. Volker Markl, volker.markl@tu-berlin.de
<b>Betreuer (Industriepartner):</b>	Dr. Radu Tudoran, Huawei Technologies, radu.tudoran@huawei.com
<b>Beginn des Mikroprojectes:</b>	1. Januar 2019
<b>Laufzeit des Projektes:</b>	24 Monate

# Contents

<b>1</b>	<b>Zusammenfassung</b>	<b>3</b>
<b>2</b>	<b>Project Definition and Motivation</b>	<b>5</b>
2.1	Focus and Goal of the Project . . . . .	5
2.2	Research and Technical Objectives . . . . .	5
2.3	Relation to Funding Programs . . . . .	6
<b>3</b>	<b>Background and Related Work</b>	<b>8</b>
3.1	Background . . . . .	8
3.2	Current Research Overview . . . . .	8
<b>4</b>	<b>Project Consortium</b>	<b>10</b>
4.1	The Database Systems and Information Management Research Group (DIMA) . . . . .	10
4.2	Huawei Technologies Co. Ltd . . . . .	10
4.3	Cooperation between DIMA and Huawei . . . . .	11
<b>5</b>	<b>Detailed Workplan</b>	<b>12</b>
5.1	Work Packages description . . . . .	12
5.1.1	Work package 1 - Requirements Analysis and Use Cases Definition . . . . .	12
5.1.2	Work package 2 - Design of Enhanced State Management Techniques . . . . .	13
5.1.3	Work package 3 - First System Prototype . . . . .	13
5.1.4	Work package 4 - Final System Prototype . . . . .	14
5.1.5	Work package 5 - Continuous Benchmarking . . . . .	14
5.1.6	Work package 6 - Project Management . . . . .	15
5.2	Project Schedule and Milestones . . . . .	15
5.3	Financial plan . . . . .	16
<b>6</b>	<b>Exploitation Plan</b>	<b>19</b>
6.1	Business Exploitation . . . . .	19
6.2	Research Exploitation . . . . .	19
6.3	Further Business and Research Exploitation . . . . .	20
<b>7</b>	<b>Verwertungsplan</b>	<b>21</b>
7.1	Wirtschaftliche Erfolgsaussichten . . . . .	21
7.2	Wissenschaftlich-technische Erfolgsaussichten . . . . .	21
7.3	Wissenschaftliche und wirtschaftliche Anschlussfähigkeit . . . . .	22
<b>8</b>	<b>Appendix: Attachments</b>	<b>25</b>
8.1	High-Performance Workstation . . . . .	25
8.2	Monitor . . . . .	26
8.3	Cloud Instances . . . . .	26

# 1 Zusammenfassung

Das Internet der Dinge, Industrie 4.0 und Big-Data-Anwendungen produzieren kontinuierlich große Mengen komplexer Daten. Dies führt zu vielschichtigen technischen Herausforderungen aufgrund der Notwendigkeit von Datenverarbeitungssystemen, nahtlos mit Datenströmen (Streams) in jeder betriebswirtschaftlichen Geschäftslogik umgehen zu müssen, wodurch letztendlich Mehrwert entsteht. Einer aktuellen Studie zufolge ist der Big-Data-Markt im Jahr 2018 42 Milliarden Dollar wert und wird bis 2027 103 Milliarden Dollar wert sein.

Bestehende Stream-Processing-Systeme (Stream Processing Engines, SPEs) bieten eine schnelle zustandsbehaftete Verarbeitung von Datenströmen mit geringer Latenz und hohem Durchsatz trotz Schwankungen in der Datenrate. Zustandsbehaftete Verarbeitung profitiert von bedarfsorientierter Ressourcenelastizität, Lastverteilung und Fehlertoleranz.

Sowohl die Forschung [1; 2; 3; 4] als auch die Industrie [5; 6; 7] unterstützen Ressourcenelastizität für zustandsbehaftete Verarbeitung und Fehlertoleranz bei verteilten oder partiell verteilten großen Zustandsmengen. In diesem Fall kann der Zustand mehrere Hundert Gigabyte groß sein.

Viele Streaming-Anwendungen erfordern eine zustandsbehaftete Verarbeitung und erzeugen eine große Zustandsmenge, die SPEs an ihre Grenzen bringt. Anwendungsbeispiele sind die Datenanalyse-Systeme hinter populären Multimedia-Diensten und Online-Marktplätzen. Diese Systeme führen eine komplexe Ereignisverarbeitung über Datenströme in Echtzeit aus. Zum Beispiel analysieren Multimediadienste und Online-Marktplätze das Benutzerverhalten, um neue Inhalte oder Produkte durch maschinelle Lernmethoden wie kollaborative Filterung zu empfehlen. Die Zustandsgröße in diesen Anwendungen wächst mit der Anzahl der Benutzer und deren Interaktionen mit der Anwendung (z.B. bewertete Artikel, Käufe) und kann auf mehreren Terabyte anwachsen. Die Analyse erfordert häufig verschiedene Zugriffsmuster auf die zugrunde liegende Modelle (z.B. das maschinelle Lernmodell), die als Zustand gespeichert werden. Daher benötigen wir einen global lesbaren und schreibbaren Zustand, um diese Algorithmen lückenlos zu unterstützen. Gegenwärtige SPEs können die Rechenressourcen nicht effizient in Bezug auf die Größe des Zustands nutzen. Dies gilt nicht nur für Intra-Cluster-Instanzen, sondern auch für Inter-Cluster-Instanzen, z.B. die Migration der SPE in Verarbeitungsumgebungen oder zu günstigeren "Pay-as-you-go"-Instanzen. Die oben genannten industriellen Herausforderungen motivieren das Ziel unseres Projektes: Eine Stream-Verarbeitung mit geringer Latenz und hohem Durchsatz unter der Bedingung, dass Operatoren einen sehr großen Zustand handhaben. Zu diesem Zweck konzentrieren wir uns auf Managementtechniken, die eine feinkörnige Fehlertoleranz, On-Demand-Ressourcenskalisierung und Lastverteilung bei einem sehr großen verteilten Zustand ermöglichen.

Unser Ziel wirft folgende Forschungsfragen auf:

1. Wie können feingranulare Lastenverteilung, Fehlertoleranz und Ressourcenelastizität mit niedriger Latenz bereitgestellt werden, wenn Operatoren eine große Zustandsmenge besitzen?
2. Wie können "Exactly Once"-Verarbeitungsgarantien der laufenden Abfragen beibehalten werden?
3. Wie können wir sicherstellen, dass die Robustheit der Anfrageverarbeitung des zugrunde liegenden Systems nicht beeinträchtigt werden?

Um Fehlertoleranz, Ressourcenelastizität und dynamischem Lastenausgleich zu gewährleisten, muss der Zustand übertragen werden. Dies führt zu einer Latenz, die proportional zur Zustandsgröße ist. Eine "Exactly-Once" Stream-Verarbeitung erfordert einen konsistenten Zustand, d.h. die Ergebnisse müssen so genau sein, als ob kein Fehler aufgetreten wäre oder das SPE keine Neuskalierung oder Neuverteilung des Zustands durchgeführt hätte. Außerdem muss das SPE trotz dieser Operationen kontinuierlich Ereignisse verarbeiten. Nach unserem besten Wissen gibt es kein System, das über eine robuste Zustandsverwaltung verfügt, um einen sehr großen, verteilten Zustand effizient zu handhaben. Viele Autoren untersuchten dieses Problem, indem sie ihren Geltungsbereich auf verteilten oder partiell verteilten Zustand [1; 6; 7] bzw. und auf kleine Zustandsmengen [8; 2; 3] einschränken. Außerdem erlauben heutzutage eingesetzte SPEs (z.B. Apache Flink) dem Endbenutzer nicht, die Streaming-Topologie nahtlos zu ändern, um eine Ressourcenskalisierung oder einen Lastausgleich durchzuführen. In diesen Fällen muss der Endbenutzer sicherstellen, dass eintreffende Ereignisse zwischengespeichert werden, und die Topologie stoppen und neu

starten. Dies führt zu sehr hohen Ausfallzeiten und Wartezeiten bei der Verarbeitung von Abfragen. Das Ziel dieses Projekts ist es, Techniken zu entwickeln, um große Zustandsmengen mit minimalen Auswirkungen auf die Leistung der Anfrageverarbeitung zu bewegen. Das Verschieben großer Zustände zwischen verschiedenen Operatorinstanzen auf einmal ist aufgrund der Netzwerkübertragung teuer, insbesondere wenn das System bereits während des normalen Betriebs überlastet ist. Unsere vorgeschlagene Lösung ist ein inkrementeller Migrationsmechanismus mit geringer Latenz, der feingranulare Teile des Zustands ("state shards") mit Hilfe periodischer *inkrementeller Sicherungspunkte* und *Replikatgruppen* verschiebt und die laufende Topologie nicht stoppt. Jedes Replikat wird durch *inkrementelle Sicherungspunkte* aktualisiert, die für den primären Operator generiert wurden.

## 2 Project Definition and Motivation

Internet of Things, Industry 4.0, and Big Data application continuously produce large amounts of complex data at high-rate. This introduces a multifaceted technical challenge due to the need of data processing systems that must seamlessly handle data streams on any operational business logic, which eventually produces value. Big Data and in particular Stream Data Processing are rapidly growing markets; recent studies expect them to be worth 103 and 50 billions of American dollars by 2027, respectively. Existing Stream Processing Engines (SPEs) offer fast stateful processing of data streams with low latency and high throughput despite fluctuations in the data rate. Stateful processing benefits from on-demand resource elasticity, load balancing, and fault tolerance. Currently, both research [1; 2; 3; 4] and industry [5; 6; 7] address resource elasticity for stateful processing while assuring fault tolerance in case of partitioned or partially distributed large state. Here, large state means hundreds of gigabytes. Many streaming applications require stateful processing and generate large state that pushes SPEs to their limits. Example of applications are the data analytics stack behind popular multimedia services and online marketplaces. These stacks perform complex event processing on live streams, e.g., multimedia services and online marketplaces analyze users behaviour to recommend new contents or items through Machine Learning methods such as collaborative filtering [9]. The size of the state in these applications scales with the number of users and their interactions with the application (e.g., rated items, purchases) and can grow to terabyte sizes. Analytics often require diverse access patterns to their underlying model (e.g., machine learning model), which is stored as state. Therefore, we need globally readable and writable state to seamlessly support those algorithms. Current SPEs fails to efficiently use the computing resources respecting the size of the state. This does not only apply to intra-cluster instances but also inter-cluster ones, e.g., migrating the SPE among operational environments or to cheaper “pay-as-you-go” instances.

### 2.1 Focus and Goal of the Project

Motivated by above industrial challenges, the goal of our project is to achieve stream processing with low latency and high throughput when operators handle very large state. To this end, we focus on management techniques that enable fine-grained fault-tolerance, on-demand resource scaling, and load balancing in the presence of very large distributed state. Our goal leads to the following research questions:

1. How to provide fine-grained load-balancing, fault-tolerance, and resource elasticity when operators hold large state with low-latency?
2. How to preserve exactly-once processing guarantees of the running queries?
3. How to not hinder the robustness of query processing capabilities of the underlying system?

Guaranteeing fault-tolerance, resource elasticity, and dynamic load balancing requires state transfer, which in turn introduces latency proportional to its size. Exactly-once stream processing requires consistent state, i.e., results must be as accurate as if no failure happened or the SPE did not perform any rescaling or rebalancing operation on the state. Besides, a SPE must continuously process stream tuples despite any of those operations. To the best of our knowledge, there is no system that fully features robust state management to efficiently handle very large, distributed state. Many authors investigated this problem by constraining their scope to partitioned or partially distributed state [1; 6; 7] and to smaller size [8; 2; 3]. Furthermore, nowadays production-ready SPEs (e.g., Apache Flink) do not allow the end-user to seamlessly alter the streaming topology to perform resource scaling or load-balancing. In those cases, the end-user has to ensure that upstream backup is in place and stop and restart the topology. This results in very high downtime and query processing latency.

### 2.2 Research and Technical Objectives

The objective of this project is to develop state management techniques to move large operator states with minimal impact on the performance of query processing. Migrating large states between operator instances in one shot is expensive due to network transfer, especially if the system is already overloaded

during its regular operation. Our key idea is to incrementally maintain a replica group for each fine-grained state unit over different work units. Each replica is updated through *incremental checkpoints* generated on the primary operator. In addition, intrinsic issues of migration pose new challenges, e.g., data consistency, tuples rerouting, physical shards handling, and network transfer cost.

Our proposed solution is a low-latency incremental migration mechanism that moves fine-grained state shards by using periodic *incremental checkpoints* and *replica groups*. An incremental checkpoint is a periodic snapshot of a state shards that involves only modified values. A replica group is a set of computing instances holding a copy of a portion of the state. Our migration mechanism moves large operator states with low impact on the system performance and without stopping the streaming topology. Although incremental migration reduces the transfer overhead, we also provide a placement scheme for primary state shards and replica groups that minimizes transfer cost. Our solutions are summarized as follows:

1. a communication-efficient replication protocol that keeps a replica group consistent with the changes in the state of the primary operator
2. an optimal primary state shards and replica groups placement for decreasing migration cost
3. a hand-over protocol that migrates the processing between two work units with minimal latency.

### 2.3 Relation to Funding Programs

Internet of Things, Industry 4.0, and Big Data applications continuously produce large amounts of complex, high-rate data. This introduces a multifaceted technical challenge due to the need of data processing systems that must seamlessly handle data streams on any operational business logic, which eventually produces value. The goal of this project is to improve the performance of data processing systems in order to maximize the exploitable value of the data. The project is thus a great fit in overall digital program of the Federal Republic of Germany. In particular, the scope of this project is aligned with the following programs of the Federal Ministry of Education and Research (Bundesministerium für Bildung und Forschung - BMBF) and of the Cabinet of the Federal Republic of Germany (Bundesregierung):

1. *Neue Hightech-Strategie*: the new High-Tech Strategy (Neue Hightech-Strategie) of the Cabinet of Germany [10] has a dedicated programm for digital innovation (Digitalen Wirtschaft und Gesellschaft). This project is heavily related to three of the following sub-programs: Big Data [11], Machine Learning, Cloud Computing [12], Smart Services, and Industry 4.0 [13]. Indeed, our project aims to improve the performance of Big Data processing engines running in a distributed environment, e.g., cloud instances. An engine enhanced with the findings our project can easily bring extra value to Industry 4.0 and Smart Service scenarios as our solution allows for timely analysis (e.g., through Machine Learning methods) of high-rate, high-volume streams of data.
2. *Zukunftsprojekt Industrie 4.0*: This is another program funded by the BMBF whose goal is to improve industrial processes through the application of novel digital technologies, thus strengthening the industrial leadership of the Federal Republic of Germany. The scope of our project is again aligned with the overall agenda of the BMBF as the prototype proposed as solution of our research problem could be used to improve the data analytics stack used for smart factories and smart services.
3. *Software Campus*: The Software Campus initiative is part of the activities of EIT Digital (formerly EIT ICT Labs), which is one of the European digital innovation accelerator promoted by the European Institute of Innovation & Technology (EIT).

### The Software Campus Initiative

The goal of Software Campus is to foster the next generation of German IT executives. It aims to combine cutting-edge scientific research and hands-on project management activities. The participants receive project management training and develop their own project idea with the support of academic and

industrial partners. As a result, the participants learn several technical skills as well soft skills, which are highly valued in the ICT Industry. Ultimately, Software Campus paves the way towards IT-Management positions for its participants.

### 3 Background and Related Work

In this section, we provide several background information regarding Big Data processing (see Section 3.1). We provide a brief overview of the systems that are currently used in the Big Data stack. In particular, we describe data stream processing because it is at the core of this research proposal. Then, we provide an overview of the current state-of-the-art of state management management techniques for streaming processing.

#### 3.1 Background

During the last ten years, the end-to-end processing, management, and analysis of Big Data has become one of the most prominent topic in Computer Science. Both research and industry have pulled together to build efficient methods to deal with Big Data. From this tight collaboration, several Open-Source frameworks have become the standard de-facto in Big Data processing. The first of those is Apache Hadoop, which allows for Big Data processing on commodity hardware by leveraging a distributed file system (called Hadoop Distributed File System [14]) and the Map-Reduce processing paradigm [15]. Apache Hadoop enabled a two-stages, disk-based processing pipeline on terabytes of batch data. However, Apache Hadoop became shortly obsolete as a novel generation of Big Data processing frameworks emerged, offering several out-of-the-shelf features, e.g., in-memory processing, larger processing pipelines, DBMS-style operators (e.g., join), iterative processing, stream data processing. Apache Spark [7], Apache Flink [5], Apache Kafka [16; 17], and Apache Storm [18] are the next-generation of Big Data frameworks. Apache Spark and Apache Flink offer batch and streaming processing capabilities, whereas Apache Kafka and Apache Storm are mainly designed for stream processing. In the scope of this project, our main focus is on stream data processing, in particular we target distributed dataflow engines with extended support to stateful stream processing. In particular, we choose as starting building block Apache Flink for two reasons. First, Apache Flink has a carefully designed state management component that allows for diverse stateful stream processing paradigms [19]. Second, Apache Flink was initially developed at the Technical University of Berlin. However, Apache Flink does not enable fine-grained load-balancing, resource elasticity, and fault-tolerance for single logical operators. Indeed, Apache Flink allows for those features by stopping and restarting the whole streaming topology, which results in higher latency and longer system downtime.

#### 3.2 Current Research Overview

Castro et al. address the problem of scaling up and recovering stateful operators in a cloud environment through a set of primitives for state management that enables scaling up and recovery of stateful operators [1]. Their experiments include operators with small states and they confirmed that larger state has a higher recovery time. In a second work, the same authors propose a new abstraction over large mutable state, called stateful dataflow graph, which manages partitioned or partial distributed state [2]. Our aim is to fill the gap in this area by providing a mechanism that both scales out and recovers a long-running system with very large distributed state. ChronoStream is a system that seamlessly migrates and executes tasks [3], whose authors believe to have achieved costless migration thank to a locality-sensitive data placement scheme, delta checkpointing, and a lightweight transactional migration protocol. Although their experiments look promising, we argue transactional migration may be avoided by using two different protocols (one for state migration and one for the hand-over) and delta checkpoints adds synchronization issues.

Ding et al. deal with finding the optimal task assignment that minimizes the costs for state migration and satisfies load balancing constraints [8]. To this end, they introduce a live and progressive migration mechanism with negligible and controllable delay. They come to a different conclusion w.r.t. ChronoStream, because they also argue that synchronization issues may affect results correctness while performing a migration. The solution of Ding et al. performs multiple mini-migrations progressively: each mini-migration migrates a number of tasks smaller than a given threshold [8]. On the other hand, their experiments do not cover large state migration and it is unclear how the system could perform in such task. Furthermore, both ChronoStream and Ding et al. consider partitioned state.

Nasir et al. present *partial key grouping* as a solution to handle load imbalance caused by skewness in the keys distribution of input streams [20; 21]. The main idea is to keep track of the number of items in each parallel instance of an operator and route a new item to the instance with smaller load. Items with the same key are routed to different parallel instances of the same operator. An improvement to the solution is to determine the “hottest” keys in the stream and assign more workers to those keys. However, they assumed the operator state has the associative property, thus merging intermediate partitioned sub-states is possible with an extra aggregate operation. Splitting the state of a given key, indeed, mitigates its growth on one working unit, yet aggregating large state will require some potentially expensive network transfers. Our aim is to propose a load balancing approach that avoids such partial aggregations.

Gedik et al. propose transparent auto-parallelization for stream processing through a migration mechanism [22]. However, we argue that their approach does not consider distributed large state and it is totally decoupled from fault-tolerance. Katsipoulakis et al. introduce few stream partitioning algorithms for stateful operators to mitigate data skewness [23], however, they do not consider dynamic topology changes and state migration.

Many SPEs have effectively implemented state management techniques (e.g., Apache Flink [5; 6; 19], Apache Spark [7], SEEP [2], Naiad [4]). In particular, Apache Flink features a technique that asynchronously checkpoints the global states to minimize the latency of a snapshot [24; 19]. However, those systems do not feature any technique to enable fine-grained load-balancing, resource elasticity, and fault-tolerance for single logic operators. Indeed, those techniques are supported only by stopping and restarting the streaming topologies, which leads to higher latency and longer downtime of the system.

## 4 Project Consortium

### 4.1 The Database Systems and Information Management Research Group (DIMA)

The Database Systems and Information Management Research Group (DIMA) under the direction of Prof. Dr. Volker Markl conducts research in the areas of information modeling, business intelligence, query processing, query optimization, impact of new hardware architectures on information management, and applications. While having a strong focus on system building and validating research in practical scenarios and use-cases, the group aims at exploring and providing fundamental and theoretically sound solutions to current major research challenges. Currently, the DIMA group consists of 14 Ph.D. students and several Post-Doctorate researchers, project managers, and administrative employees as well as dozens of student assistants. The DIMA group has a very close cooperation with the Intelligent Analytics for Massive Data (IAM) research unit at the German Research Center for Artificial Intelligence (DFKI GmbH) and the Berlin Big Data Center (BBDC). DFKI is a research center that is actively involved in numerous organizations representing and continuously advancing Germany as an excellent location for cutting-edge research and technology. Instead, BBDC is a research project funded by the German Ministry for Education. These three groups share the same research vision, i.e., to help bridge the data analytics talent gap through research and the development (R&D) of novel technologies, systems, and methods along the entire data value chain, including data acquisition, information extraction and integration, model building, and interactive exploration. IAM, DIMA, and BBDC are managed by Prof. Dr. Volker Markl, a full professor of Computer Science at the Technische Universität Berlin. Notable accomplishments of the DIMA group include Stratosphere, a cross-institutional collaborative project, which resulted in the development of a big data analytics platform that is today known as Apache Flink. The DIMA group collaborates with large international companies, e.g., IBM, Oracle, Amazon, SAP, Huawei, Deutsche Telekom. The DIMA group helps creating startups, such as Parstream, Internet Memory Research, and data Artisans. The DIMA group is highly visible in the research community. Between 2008 and 2017, DIMA research was published at top venues, including VLDB (11 publications), SIGMOD (5 publications), ICDE (3 publications), and EDBT (2 publications). The DIMA group employs Postdoctoral Researchers, Research Associates pursuing PhD degrees, and Research and Teaching Assistants pursuing Bachelor's and Master's degrees. Students collaborate in an open, research-driven, entrepreneurially minded, international environment with a clear focus on creating, innovating, prototyping, and validating leading edge research in practical settings with our scientific and industry partners.

Prof. Dr. Volker Markl will serve as scientific advisor of the project. He is a Full Professor and Chair of the Database Systems and Information Management Group at the Technische Universität Berlin (TUB) and an Adjunct Full Professor at the University of Toronto. He is the Director of the Intelligent Analytics for Massive Data Research Group at DFKI and Director of the Berlin Big Data Center. In addition, he serves as the Secretary of the VLDB Endowment. His current research interests include new hardware architectures for information management, scalable processing and optimization of declarative data analysis programs, and scalable data science. As of today, he has presented over 200 invited talks in numerous industrial settings, major conferences, and research institutions worldwide. Furthermore, he has authored and published over 100 research papers at world-class scientific venues. Between 2010-2016, he was Speaker and Principal Investigator of the Stratosphere Research Unit funded by the German Research Foundation (DFG), which resulted in numerous top-tier publications, as well as the Apache Flink big data analytics system. In 2014, he was named one of Germany's leading digital minds (Digitale Köpfe) by the German Informatics Society. Prior to joining TUB, he was a Research Staff Member and Project Leader at the IBM Almaden Research Center in San Jose, California.

### 4.2 Huawei Technologies Co. Ltd

Huawei Technologies Co. Ltd, hereinafter Huawei, is one of the largest private provider of Information and Communications Technology (ICT) infrastructures and smart devices. Huawei has its headquarters in Shenzhen (China) but it operates in more than 170 countries and regions, providing integrated solutions

across four key domains, i.e., telecom networks, IT, smart devices, and cloud services. Huawei R&D departments are very active in the areas of cloud infrastructures, Internet of Things, Industry 4.0, and autonomous vehicles [25] and one of its main research center is in Munich [25], which is completely dedicated to research of those topics. This is also a result of the tight cooperation between Germany and China in the areas of Internet of Things and Industry 4.0 [26]. At the moment, Huawei is building a cloud platform to power its processing infrastructure to handle a broader range of use-cases, e.g., Industry 4.0, Internet of Things, Internet of Vehicles, e-commerce, and healthcare. At the core of the data analytics stack, Huawei is using Apache Flink as stateful stream processing engine to perform online analysis of stream data [27].

### 4.3 Cooperation between DIMA and Huawei

In the scope of this project, the cooperation between the DIMA Research Group and Huawei is twofold and involves a scientific and technical exchange during the development of the project as well as a project management training. Huawei and the DIMA group share the same interest for scalable stream processing and they already collaborate in two projects in this area. The first project is called ADAM and is also funded via Software Campus and it aims to enable approximate stream processing on modern hardware (e.g., Field Programmable Gate Arrays). The second project running between DIMA and Huawei has also a strong focus on approximate stream processing, however, it targets distributed execution engines (e.g., Apache Flink). This project will continue to promote the scientific cooperation between the DIMA research group and Huawei, strengthening the quality of DIMA research and providing Huawei with cutting-edge research findings that can help in improving their data processing systems. In particular, Huawei is running a very large scale data analytics pipeline called CloudStream<sup>1</sup>, which would benefit from the fine-grained state management techniques that we aim to investigate in the scope of this research project. Therefore, we consider Huawei as the ideal industrial partner for the RHINO project given also the research portfolio of the DIMA research group. On the grounds of the previous joint cooperation and high level of expertise of the partners, there exist very high chances of success for this project.

Huawei will proactively support this project with a mentor who will provide technical expertise. During the project time frame, there will be 5 face-to-face meetings between the Huawei mentor and the project manager. The goal of these meetings is twofold. The project manager will provide updates regarding the work packages to track the status of the project. The Huawei mentor will provide feedback on the diverse aspects of the project and will help shaping the system requirements, the use cases, the software architecture as well as the validation scenario and KPIs that will be used to continuously assess the intermediate and final prototypes.

---

<sup>1</sup><https://www.huaweicloud.com/>

## 5 Detailed Workplan

### 5.1 Work Packages description

This project is divided in six work packages and it spans on 24 months. In the following, we give a brief overview of each work package, which we describe in details in the next part of this section.

In Work Package 1 (WP1), we plan to investigate and define with the support of Huawei the function requirements as well as the industrial use cases that need analytics on stream data, which deals with high data-rates and generate very large state. In WP1, we also plan to define the validation scenario and its related Key Performance Indicators (KPIs) of our final system prototype. In Work Package 2 (WP2), we plan to design novel state management techniques based on our previous research [28] and tailored to Apache Flink, i.e., the most suitable processing framework for use cases involving stateful stream processing. In Work Package 3 (WP3), we plan to develop a first prototype of our state management system, directly in the engine of Apache Flink. In Work Package 4 (WP4), we plan to enhance our first prototype thanks to the feedback of Work Package 5. In Work Package 5 (WP5), we plan to thoroughly benchmark our system prototype, providing input to WP4 in order to continuously improve the quality of our final system. In Work Package 6 (WP6), we plan to carry out all the management-related aspects of the project, e.g., the management of human resources and the reporting of milestones, deliverables, and scientific results.

The leadership of the project in all the work packages is assigned to Bonaventura Del Monte, who is appointed as Project Manager with the support of the mentors from Huawei and DFKI. To successfully complete our project, two student assistants and one researcher are required because of the large codebase of the targeted framework, the technical depth, and the scale of the validation scenario. The two students assistant will be hired for the whole 24 months of the project, whereas a researcher will be hired for 3 PMs. The two student assistant are needed to carry out the software engineering related tasks of the project, which are identified by the project manager, the mentors, and the help of the junior researcher. The requirement for the research is a Master's degree in Computer Science or Computer Engineering. This is necessary because the design and implementation of the critical components of the system need a higher level of expertise.

The efforts of the project manager and mentors are not always explicitly stated because they are not part of the project proposal.

#### 5.1.1 Work package 1 - Requirements Analysis and Use Cases Definition

##### Description

In WP1, we aim to investigate and define with the support of Huawei the function requirements as well as the industrial use cases that need analytics on stream data, which deals with high data-rates and generate very large state. In WP1, we also plan to define the validation scenario and its related KPIs of our final system prototype. In particular, we focus on those use cases that require stateful stream processing and generate terabytes of state. We target those use cases because they will benefit the most from enhanced state management techniques. An early definition of the validation scenario along with related KPIs is also beneficial because it provides an assessment framework for the successful completion of the project.

##### Related deliverable

- **D1.1** Description of Functional Requirement, Use Cases, and Validation Scenario.

##### Time Schedule and Effort

Time Schedule	Researcher	Student Assistant	Project Manager
M1 to M4 (4 months)	0 PM	4 PM	2 PM

WP1 is scheduled in the first four months of the project. WP1 is essential for the successful completion of the project because it defines the use cases in which our prototype could be used as well as a validation scenario to thoroughly assess the capabilities of our solution. With the support of Huawei, the project

manager is tasked to gather the requirements and the use cases as well as defining the validation scenario and its KPIs. The two student assistant are instead tasked with learning the internals of Apache Flink, if needed. In case they already have a good understanding of the internals of Apache Flink, they will help the project manager with the validation scenarios and related KPIs.

### 5.1.2 Work package 2 - Design of Enhanced State Management Techniques

#### Description

The goal of WP2 is twofold. First, we aim to define novel state management techniques to provide fine-grained resource elasticity, load-balancing, and fault-tolerance when the stream processing engine handles very large operator state. Those techniques consists of distributed protocols for incremental data migration and consistent handover among worker units. They are agnostic to the underlying processing engine and an early stage definition is available in our previous research [28]. Finally, we tailor our novel techniques to the selected dataflow engine, i.e., Apache Flink. We choose Apache Flink as it is the most suitable engine for those type of techniques. Furthermore, Huawei uses Apache Flink in several production environments. It is worthwhile mentioning that the state management components must be tightly coupled with the processing engine. If we decided to create an external system to manage the state, we would lose performance [19].

#### Related deliverable

- **D2.1** Design of novel state management techniques for fine-grained resource elasticity, load-balancing, and fault-tolerance in the presence of very large operator state.

#### Time Schedule and Effort

Time Schedule	Researcher	Student Assistant	Project Manager
M5 to M8 (4 months)	1 PM	4 PM	1 PM

This WP has a duration of 4 months, it starts in M5 and ends in M8. This WP may start earlier than M4, if WP1 is completed earlier than expected. Thank to his knowledge of distributed stream processing and state management techniques, the project manager spends 1 PM to guide the researcher and the two student assistants in designing new techniques (e.g., distributed protocols) to handle very large operator state in stream processing engines. The junior researcher works for 1 PMs, whereas the two student assistants for overall 4 PMs.

### 5.1.3 Work package 3 - First System Prototype

#### Description

In WP3, we combine the findings of WP1 and WP2 to build a first prototype of a system. We will build the prototype in the engine of Apache Flink. The goal is to extend the capabilities of Apache Flink to support fine-grained fault-tolerance, resource elasticity, and load balancing features in the presence of very large distributed state, while preserving exactly-once processing guarantees. The outcome of WP3 is a proof-of-concept that will be the input of WP5, which has the goal of assessing the initial performance of our novel state management techniques.

#### Related deliverable

- **D3.1** First System Prototype.

#### Time Schedule and Effort

Time Schedule	Researcher	Student Assistant	Project Manager
M9 to M15 (7 months)	1 PM	10 PM	1 PM

WP3 is scheduled on a 7 months time frame. WP3 starts in M9 and ends in M15. The project manager will coordinate the team (1 PM), which consists in a researcher (1 PM) and two student assistants (10 PM). The researcher will define the software interfaces as well as the overall skeleton of the software components. The task of the two student assistants is to implement the interface and complete the skeleton w.r.t. the algorithmic designed, defined in WP2. Minimal functional tests (e.g., unit tests) of the prototype will be also implemented. We leverage the expertise of the junior researcher to define and implement the critical components of the system.

#### 5.1.4 Work package 4 - Final System Prototype

##### Description

By using the continuous feedback of WP5, we incrementally develop the final system prototype. We solve eventual bottlenecks and anomalous behaviours of the first prototype of the system and we ensure consistent and correct stream data processing of the system. This WP is run in parallel with WP5 and the two WPs are tightly coupled to benefit from the mutual continuous benchmarking and continuous delivery of improved versions of the system.

##### Related deliverable

- **D4.1** Final System Prototype.

##### Time Schedule and Effort

Time Schedule	Researcher	Student Assistant	Project Manager
M19 to M24 (6 months)	1 PM	16 PM	1 PM

WP4 is scheduled on a 6 months time frame. WP4 starts in M19 and ends in M24. The project manager will coordinate the team (1 PM), which consists in a researcher (1 PM) and two student assistants (16 PMs). The researcher and the two student assistants have to task of completing the implementation of the system and removing the bottlenecks of the system, which will be identified in WP5. We leverage the expertise of the junior researcher to define and implement the critical components of the system.

#### 5.1.5 Work package 5 - Continuous Benchmarking

##### Description

The goal of this WP is to continuously benchmark the capabilities of our prototype w.r.t. the KPIs and validation scenarios defined in WP4. This WP starts by assessing the prototype developed in WP3. Then, it provides continuous feedback to WP4 in order to resolve eventual performance bottleneck or anomalous behaviours. Hence, the main goal of WP5 is to help delivering a final, high-quality prototype through a thorough performance drill down. We intend to benchmark our prototype on a multi-node (or multi-container) infrastructure, thus, simulating a real-world scenario. The multi-node infrastructure is owned and managed by the DIMA group. Further experiments will be carried on real-world cloud instance to improve the validity of our approach. However, the setup and the execution of distributed experiments will be completely carried out by the team members of the project. This WP runs in parallel with WP5 and the two WPs are tightly coupled to benefit from the mutual continuous benchmarking and continuous delivery of improved versions of the system.

##### Related deliverable

- **D5.1** Benchmark of the first prototype
- **D5.2** Benchmark of the second prototype

### Time Schedule and Effort

Time Schedule	Researcher	Student Assistant	Project Manager
M15 to M24 (9 months)	0 PM	14 PM	1 PM

WP5 is scheduled on a 9 months time frame. WP4 starts in M15 and ends in M24. The project manager will coordinate the team (1 PM), which consists in two student assistants (14 PM). The two student assistants are tasked with implementing a validation scenario and run distributed experiments. The expertise of the project manager and the mentors is leveraged to define a solid benchmarking framework of the system (e.g., environment configuration, correct measurement of KPIs, correct performance drill down of the system).

#### 5.1.6 Work package 6 - Project Management

##### Description

The goal of WP6 is to ensure the achievement of objectives of the project in terms of scientific quality, timely delivery, and contribution to the expected impact of the project. WP6 aims to achieve an efficient end-to-end progress monitoring, timely and detailed reporting to Huawei (and to Software Campus organizers - if necessary), and internal coordination of the other work packages and the management of the team members. The coordination with Huawei as well as with the scientific advisor are important tasks of WP6 as they will be providing high valuable feedback, which is essential to successfully complete the overall project. The documentation of the project is the ultimate outcome of this WP6 and it will be carried out through 7 deliverables and 4 milestones (see Section 5.2).

##### Related deliverable

- **D6.1** Report of project result.

### Time Schedule and Effort

Time Schedule	Researcher	Student Assistant	Project Manager
M1 to M24 (24 months)	0 PM	0 PM	2 PM

WP6 is scheduled on a 24 months time frame (i.e., for the whole project duration). It starts in M1 and ends in M24. The project manager is tasked with the end-to-end management of the project for a total of 2PM of effort.

## 5.2 Project Schedule and Milestones

The project is scheduled on a 24 months time frame and it will start on January, 1 2019. The project manager is Bonaventura Del Monte who will coordinate with the mentors from Huawei and TUBDIMA as well as the human resources planned for this project (see Section 5.3). Overall, we plan five face-to-face meetings or workshops with Huawei (see Section 5.3). Furthermore, we plan to schedule videoconferences (or teleconferences) to proactively solve problems that might happen during the project, to provide extra information, and to request extra feedback. We plan four milestones along the project timeline, which are shown in Figure 1.

The four milestones are described as follows:

- **MS1.** Document containing the description of requirements, use cases, validation scenario, and KPIs
- **MS2.** Document containing the system design of novel state management techniques for stream data processing
- **MS3.** First system prototype
- **MS4.** Final system prototype

	Project Effort in Person Months				First Semester						Second Semester						Third Semester						Fourth Semester					
	Total	Researcher	Student Assistant	Project Manager	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
WP1	6	0	4	2				X																				
WP2	6	1	4	1								X																
WP3	12	1	10	1																X								
WP4	18	1	16	1																								X
WP5	15	0	14	1																								
WP6	2	0	0	2																								
<b>Total</b>	<b>59</b>	<b>3</b>	<b>48</b>	<b>8</b>	<b>Total Milestones: 4</b>																							

Figure 1: Gantt Diagramm of the Project.

### 5.3 Financial plan

Cost Entry	Cost Description	Position	Amount
<b>Personalkosten</b>	TVL-13 Stelle (3 Months)	0812	16.557,78 €
	Student Assistant (48 Months, 80h/month)	0822	53.107,20 €
	<b>Subtotal</b>		<b>69.664,98 €</b>
<b>Travels</b>	Software Campus Training and Events	0846	1.650,00 €
	Meetings w/ the Industrial Partner	0846	875,00 €
	Bahncard 50 (2 Years)	0846	510,00 €
	Conferences	0846	12.215,00 €
	<b>Subtotal</b>		<b>15.250,00 €</b>
<b>Equipment</b>	High-performance Workstation (32GB RAM)	0850	3727,00 €
	Accessories	0843	335,00 €
	GCP Instances	0833	10.371,58 €
	Scientific Material	0843	300,00 €
	Presentation Material	0843	150,00 €
	<b>Subtotal</b>		<b>14.883,58 €</b>
		<b>Total</b>	<b>99.798,56 €</b>

Table 1: Costs of the Project

#### 0812 - Human Resources Cost - Personalkosten

**Resercher.** To carry out part of the tasks of WP2-5, it is necessary to hire a researcher in possession of a MS.c in Computer Science. The expertise of the hired research will be used to design and implement the critical component of the system prototype. The researcher will be hired with a TVL-13 contract for 3 months.

#### 0822 - Human Resources Cost - Personalkosten

**Student Assistant.** To successfully complete all the work of WP1-5, it is necessary to hire 2 student assistants that will help the Researcher and the Project Manager in all the tasks of this project. In particular, they will help defining use cases and validation scenario, implementing the system prototype, and carrying out diverse benchmarks. Each student assistant will be hired for 24 months with a 80h/month contract.

#### 0846 - Travel Cost - Reisekosten

**Software Campus Training and Events.** All the participants of the Software Campus program have to take part to six training events. The goal of those training events is to improve the soft-skills of the

Conference	Venue	Fee	Travel	Hotel	Allowance	Length	Total
VLDB	Los Angeles, USA	750 €	1.200 €	1.644 €	336 €	6 Days	<b>3.930 €</b>
SIGMOD	Amsterdam, Netherlands	983 €	260 €	714 €	276 €	6 Days	<b>2.233 €</b>
ICDE	Macau, China	1.010 €	1.200 €	468 €	300 €	6 Days	<b>2.978 €</b>
OSDI	unknown (international)	640 €	1.300 €	828 €	306 €	6 Days	<b>3.074 €</b>
							<b>12.215 €</b>

Table 2: Costs for Conference Participation.

participants, in particular the ones needed for project management. Those six training events will take place during the first year of the software campus and their venue will be announced after the beginning of the project. For this reason, we calculate 350,00 € for each travel, which will be needed to cover travel and accommodation expenses. Each travel is supposed to last 1 or 1.5 business days. There are no expenses for the events at the Software Campus headquarters in Berlin because the Project Manager lives and works in Berlin. The overall cost for those training events is 2.100,00 €.

**Meeting with Huawei.** To better coordinate the project as well to discuss and evaluate the results of the project, it is very important to organize face-to-face meetings with the industrial partner. Therefore, we plan to organize five face-to-face meetings, i.e., a kickoff meeting at the beginning of the project as well as four status update meeting approximately every 6 months during the project time frame. For those five meetings, we we calculate 250,00 € for each travel, which will be needed to cover travel and accommodation expenses. Each travel is supposed to last 1 or 1.5 business days. The overall cost for those travels is 1.250,00 €.

**Conferences.** This project deals with important and currently relevant scientific topics, in particular the Internet of Things, Big Data and Industry 4.0. Those topics are a challenge for the realm of data processing and we therefore expect to publish our findings as articles and present them in conferences. TO ensure a very high-impact for our project, we aim at leading international conferences in the scope of data processing and databases. The CORE ranking provides an internationally recognized classification of conferences. The most important conferences of this field are rated A\* and A. The publication at A\* or A conferences allow for discussing about scientific results with internationally renowned scientists as well as establishing contacts with the biggest players in industry and research. We also include a national conference to present our findings in front of a local audience. Within the time frame of the project, we plan to submit our findings to few of these events. This includes presentations at conferences and workshops, submitted papers, demonstrations or invited lectures. The publication at A\* and A conferences is preceded by a long review process with strong competition. Therefore, it is not known at which conference our contributions will be accepted. We point out that the top conferences are ACM SIGMOD (Special Interest Group on Management of Data), VLDB (Very Large Databases), USENIX OSDI (Symposium on Operating Systems Design and Implementation), and IEEE ICDE (International Conference on Data Engineering). Attending and presenting the findings of our project at these venues is a great honour and a great opportunity because they give us the chance to discuss about our research with internationally renowned scientists as well as highly skilled engineers, working at top universities and companies. This also ensure high recognition to our project and it could foster future collaboration with academic and industrial partners. Furthermore, this will further improve the reputation of the Software Campus program. Because of the high impact of those conferences, research articles published at those venues are very likely to be further recognized and cited. Therefore, it is paramount to allocate a suitable budget to attend those conferences, which we summarize in Table 2. The overall expenses for conference-related travels is 12.215,00 €. **Travel expenses reduction.** Since the majority of the travels scheduled for this project are inside Germany, the project manager plans to travel by train. To reduce the travel expense, we plan to buy one Bahncard to get a discount on each train ticket. In Table 3, we report the cost of 24 train travels up to 75 € with and without Bahncard. Considering 11 travels inside

	W/o BahnCard	BahnCard 25	BahnCard 50	BahnCard 100
Cost for 24 travels up to 75 €	1.800 €	1.350 €	900 €	0 €
BahnCard Cost (2 Years)		124 €	510 €	8.540 €
Total	1.800 €	1.474 €	1.410 €	8.540 €

Table 3: Cost for Train travels.

Germany, the BahnCard 50 is the right variant for reducing the expenses of our project. This results in lower cost for the travel listed above. The travel cost for Software Campus related events are reduced from 2.100,00 € to 1.650,00 €. The travel expenses needed to meet the industrial partner are reduced from 1.250,00 € to 875,00 €. This results in 825,00 € of savings.

### 0850 - Hardware

To develop our prototype system, it is necessary to use high-performance mobile workstations. The main reasons are the need of machines able to smoothly run the development environment (e.g., IDE, containers, Java Virtual Machine) and to help simulate the cluster execution, efficiently. To simulate the cluster execution, we need a machine able to run multi-threaded, data-intensive applications on multiple cores and fast disks in physical and virtualized environments. Therefore, we choose as high-performance mobile workstation, a machine equipped with 6 cores CPU, 32 GB of main memory, and 512 GB of Solid State Disk. The specifications of this workstations are available in the attachment Section 8.1.

### 0833 - Computing cost - Rechnerkosten

**Cloud Instances.** To properly benchmark our prototype, we need to run it on real-world production environments, i.e., a cluster of cloud instances. To this end, we select as cloud infrastructure the Google Cloud Platform. We emphasize that running, testing, and benchmarking our prototype on real-world production environments is very important because it ensures a high-impact for our project. Indeed, our prototype is a sophisticated distributed system that allows for stream data processing on scale-out infrastructures, thus running and benchmarking it on cloud instances is the ideal way to fully assess its capabilities. As the payment model of cloud instances is "pay-as-you-go", we plan to run our prototype on 20 *n1-standard-16* compute instances on the Google Cloud Platform. We select 15 instances to be equipped with 16 virtual cores, 60GB of main memory, 2x375 GB of SSD, and a 10Gbps Ethernet card. The remaining 5 instances are equipped with 16 virtual cores, 60GB of main memory, 1x375 GB of SSD, and a 10Gbps Ethernet card. Each instance will be used for 50 hours per month, for a total of 950 hours/month. The specifications of those cloud instances are available in the attachment Section 8.3.

### 0843 - Other expenses

**Scientific material.** During the time frame of the project, we expect to need specialized literature to improve the theoretical and technical know-how of the team members of the project, e.g., books or e-learning contents. Those resources could be used to learn a new programming language or novel statistical methods.

**Presentation Material.** During the project, we plan to participate to some conferences to present the outcome of our work (see Section 5.3). Each conference usually consists of two main events. Every author of an accepted paper gives a talk about the findings of the paper. Then, the author presents the work in a panel session by using a poster. Therefore, we allocate some budget for high-quality print of our posters.

**Accessories.** To improve the work efficiency of our team, we also require a monitor (see Section 8.2).

## 6 Exploitation Plan

### 6.1 Business Exploitation

Data-driven business strategies have become one of the most crucial points in the corporate agenda. They have transformed the way companies do business by redesigning core processes and making data a first-class citizen in the decision-making loop. With the aid of the right data, a company can easily make use of a broad range of techniques to increase revenue, cut cost, improve customer relationship, and reduce risk. However, how to use data and analytics and the deployment of the right technology architecture are the major blockers when implementing a data-driven business strategy. As of today, there are plenty of ready-to-use analytics and technology architectures that can be used to build a company’s data-driven strategy. Nevertheless, the volume, velocity, and complexity of the data add constraints on this choice. This has led to the development of several frameworks that can smoothly process both data-at-motion (streams) and data-at-rest (batch). Whereas several engines for batch and stream data are freely available, there is a shortage of data-processing engines that can handle complex, stateful analytics on fast data.

As a matter of facts, many streaming applications require stateful processing and generate large state that pushes SPEs to their limits. Example of applications are the data analytics stack behind popular multimedia services and online marketplaces. These stacks perform complex event processing on live streams, e.g., multimedia services and online marketplaces analyze users behaviour to recommend new contents or items. The size of the state in these applications scales with the number of users and their interactions with the application (e.g., rated items, purchases) and can grow to terabyte sizes. Current SPEs fails to efficiently use the computing resources with respect to the size of the state and do not provide efficient on-demand resource elasticity, load balancing, and fault tolerance. Furthermore, they fail to leverage inter-cluster instances, e.g., migrating the SPE among operational environments or to cheaper “pay-as-you-go” instances.

In this project, we target those applications as our use case and we intend to build a prototype that eases the handling of large distributed state in an SPE. This results in an SPE that can process stream with high throughput and low latency. This means that actionable results can be retrieved faster from the input data, leading to more profitable business decisions. Improving the operational efficiency of those SPEs also benefits the efficiency of the underlying hardware infrastructure, reducing operational cost. Moreover, our industrial partner Huawei offers “Cloud Stream Service”, a data stream processing platform as a service on cloud that is based on the same SPE that we aim to enhance, i.e., Apache Flink. For this reason, the findings of this project could help Huawei in improving the operational efficiency of their own data stream processing platform.

### 6.2 Research Exploitation

The timely analysis of complex, fast, and high-volume data through stateful analytics is not only an open challenge in the realm of Big Data, Cloud platforms, Internet of Things, and Industry 4.0 but also in academia. Therefore, we aim to present the findings of our project in conference and workshops and published as research articles. To guarantee a high impact of our research, we target the following top-level international conferences:

- “Very Large Databases” (VLDB)
- “Special Interest Group on Management of Data” (SIGMOD)
- “IEEE International Conference on Data Engineering” (ICDE)
- “USENIX Symposium on Operating Systems Design and Implementation” (OSDI)

Participating to those conference does not only provide visibility to the project but also gives us the opportunity to obtain feedback regarding the research findings of our project from people from academia and industry. This could also lead to potential new research collaborations and synergies with other top-notch universities, research institutes, and industrial partners.

Along with above research-related activities, we aim to improve the core competences of the DIMA research group in the realm of Big Data, Cloud platforms, Internet of Things, and Industry 4.0. These competences can help in educating the next generations of Master's students and Ph.D. students in our research group. Moreover, the theoretical and technical research outcomes of this project will increase the impact and soundness of the Ph.D. dissertation of the project manager, Bonaventura Del Monte. Finally, the strong research focus of this project will further shape the carrier prospective of all the team members of the project.

### **6.3 Further Business and Research Exploitation**

As streaming processing is an ongoing research topic in academia and industry, the research findings of this project can be used not only to promote joint research projects and collaboration by transferring knowledge and technology to academia and industry but also as basis for new research projects. The prototype that will be developed in this project could be further enhanced and integrated in a real-world platform. Furthermore, another Software Campus project lead by another Ph.D. student could use the outcome of our project as a starting point. Our prototype could also serve as reference in future academic or industrial project. Finally, the finding of the project could be used as a reference in advanced training University courses to further educate junior researchers.

## 7 Verwertungsplan

### 7.1 Wirtschaftliche Erfolgsaussichten

Datengesteuerte Geschäftsstrategien sind zu einem der wichtigsten Punkte auf der Agenda von Unternehmen geworden. Durch die verstärkte Nutzung von Daten in Entscheidungsprozessen kommt es zu einer Neugestaltung von Kernprozessen und Unternehmen verändern somit die Art und Weise in der Geschäfte getätigt werden: Unter Zuhilfenahme von relevanten Daten können Unternehmen eine breite Palette von Techniken nutzen um ihren Umsatz zu steigern, Kosten zu senken, Kundenbeziehung zu verbessern und Risiken zu reduzieren. Die Bereitstellung geeigneter IT Infrastruktur und die Auswahl relevanter Daten mithilfe dieser stellen hierbei die größten Hindernisse bei der Entwicklung von datengesteuerten Geschäftsstrategien dar. Obwohl es heute eine Vielzahl von gebrauchsfertigen Analyse- und Technologiearchitekturen zum Aufbau von datengetriebenen Strategien innerhalb von Unternehmen gibt, schränken das Volumen und die Rate der zu verarbeitenden Daten sowie deren Komplexität die Anzahl geeigneter Systeme ein. Dies hat zur Entwicklung diverser Frameworks geführt, die sowohl *data-at-motion* (Datenströme) als auch *data-at-rest* (Batch) verarbeiten können. Während viele Systeme für die Verarbeitung von Batch- und Stream-Daten frei verfügbar sind, gibt es einen Mangel an Datenverarbeitungssystemen (SPEs), die komplexe, zustandsbehaftete Analysen auf schnell eintreffenden Daten verarbeiten können.

Viele Streaming-Anwendungen erfordern Datenverarbeitungsprogramme die mit großen internen Zuständen arbeiten und bringen SPEs somit an die Grenze ihrer Leistungsfähigkeit. Beispiele hierfür sind Datenanalyse Architekturen, welche in populären Multimedia-Diensten und Online-Marktplätzen eingesetzt werden und die komplexe Verarbeitung von Ereignissen innerhalb von Datenströmen ermöglichen, wie z.B.: die Analyse von Multimedia Diensten und Online-Marktplätzen um deren Nutzern neue Inhalte und Produkte zu empfehlen. Die Größe des verwalteten Zustands in der Anwendungen skaliert hierbei mit der Anzahl der Nutzer und deren Interaktionen mit der Anwendung (z.B.: Anzahl der bewerteten Artikel und Einkäufe), und kann in Größenordnungen von bis zu Terabytes wachsen. Aktuelle SPEs können die zur Verfügung stehenden Rechenressourcen im Hinblick auf die Größe des Zustands nicht effizient nutzen und bieten keine bedarfsorientierte Elastizität, Lastenausgleich und Fehlertoleranz für diese Ressourcen. Darüber hinaus können sie Inter-Cluster-Instanzen nicht nutzen, d.h. SPEs können nicht zwischen Betriebsumgebungen oder günstigeren *Pay-as-you-go* -Instanzen migriert werden.

Innerhalb dieses Projekts, soll ein SPE Prototyp erstellt werden, der Datenströme mit hohem Durchsatz und geringer Latenz verarbeitet, die Handhabung von großen verteilten Zuständen erleichtert und somit die effiziente Verarbeitung von komplexen Datenflussprogrammen ermöglicht. Somit lassen sich schneller verwendbare Ergebnisse aus den Eingabedaten erzeugen, was wiederum zu profitablen Geschäftsentscheidungen führt. Die Verbesserung der Betriebseffizienz von SPEs bewirkt auch gleichzeitig die effizientere Nutzung zugrunde liegender Hardware Infrastruktur, wodurch Betriebskosten gesenkt werden. Das von uns als Grundlage angedachte SPE Apache Flink wird darüber hinaus auch von unserem Industriepartner Huawei in der Cloud als “platform-as-a-service” unter dem Namen “Cloud Stream Service” angeboten. Daher können die Ergebnisse dieses Projekts Huawei dabei helfen die betriebliche Effizienz ihrer Datenstromverarbeitungsplattform zu verbessern.

### 7.2 Wissenschaftlich-technische Erfolgsaussichten

Die zustandsbehaftete Analyse komplexer, schneller und großvolumiger Datenströme ist nicht nur in den Bereichen Big Data, Cloud-Plattformen, Internet der Dinge und Industrie 4.0 eine offene Herausforderung, sondern auch in der Wissenschaft. Daher planen wir die Ergebnisse unseres Projekts in Konferenzen und Workshops vorstellen und als Forschungsartikel veröffentlichen. Um eine hohe Reichweite unserer Forschungsergebnisse zu gewährleisten, haben wir uns für folgende internationale Top-Level-Konferenzen entschieden:

- “Very Large Databases” (VLDB)
- “ACM Special Interest Group on Management of Data” (SIGMOD)

- “IEEE International Conference on Data Engineering” (ICDE)
- “USENIX Symposium on Operating Systems Design and Implementation” (OSDI)

Die Teilnahme an oben genannten Konferenzen erhöht nicht nur die Sichtbarkeit des Projekts, sondern gibt uns auch die Möglichkeit, Rückmeldungen zu unseren Forschungsergebnissen von Forschern aus Wissenschaft und Industrie zu erhalten. Dies eröffnet auch die Möglichkeit zu neuen Forschungsk Kooperationen und Synergien mit anderen hochrangigen Universitäten, Forschungsinstituten und Industriepartnern. Neben den genannten forschungsbezogenen Bestrebungen wollen wir die Kernkompetenzen des und insbesondere der DIMA-Forschungsgruppe im Bereich Big Data, Cloud-Plattformen, Internet der Dinge und Industrie 4.0 verbessern. Diese Kompetenzen können bei der Ausbildung der nächsten Generationen von Master- und Ph.D. Studenten innerhalb unserer Forschungsgruppe helfen. Die theoretischen und technischen Forschungsergebnisse dieses Projekts werden hierbei auch die Aussagekräftigkeit und Wirkung der Dissertation des Projektmanagers, Bonaventura Del Monte, verstärken. Der Fokus auf Forschung dieses Projekts kann dabei auch eine positive Wirkung auf alle Teammitglieder des Projekts haben.

### **7.3 Wissenschaftliche und wirtschaftliche Anschlussfähigkeit**

Die Verarbeitung von Datenströmen ist ein aktuelles und gefragtes Forschungsthema in Wissenschaft und Industrie. Die in diesem Projekt erzielten Ergebnisse können nicht nur zur Förderung weiterer gemeinsamer Forschungsprojekte und Kooperationen genutzt werden, indem Wissen und Technologie in Wissenschaft und Industrie transferiert werden, sondern auch als Grundlage für neue Forschungsprojekte dienen. Der in diesem Projekt entwickelte Prototyp kann weiter verbessert und in eine reale Plattform integriert werden, als Ausgangspunkt für ein weiteres Software Campus Projekt verwendet werden und als Referenz für zukünftige akademische oder industrielle Projekte dienen. Weiter kann das Ergebnis des Projekts als Referenz in Fortbildungsveranstaltungen für die Weiterbildung von Nachwuchswissenschaftlern dienen.

## References

- [1] R. Castro Fernandez, M. Migliavacca, E. Kalyvianaki, and P. Pietzuch, “Integrating scale out and fault tolerance in stream processing using operator state management,” in *ACM SIGMOD*, 2013. [Online]. Available: <http://doi.acm.org/10.1145/2463676.2465282>
- [2] —, “Making state explicit for imperative big data processing,” in *USENIX ATC*, 2014. [Online]. Available: <http://dl.acm.org/citation.cfm?id=2643634.2643640>
- [3] Y. Wu and K. Tan, “Chronostream: Elastic stateful stream computation in the cloud,” in *IEEE ICDE*, 2015. [Online]. Available: <http://dx.doi.org/10.1109/ICDE.2015.7113328>
- [4] D. Murray, F. McSherry, R. Isaacs, M. Isard, P. Barham, and M. Abadi, “Naiad: A timely dataflow system,” in *ACM SOSP*, 2013. [Online]. Available: <http://doi.acm.org/10.1145/2517349.2522738>
- [5] A. Alexandrov, R. Bergmann, S. Ewen, J.-C. Freytag, F. Hueske, A. Heise, O. Kao, M. Leich, U. Leser, V. Markl, F. Naumann, M. Peters, A. Rheinländer, M. J. Sax, S. Schelter, M. Höger, K. Tzoumas, and D. Warneke, “The stratosphere platform for big data analytics,” *The VLDB Journal*, 2014. [Online]. Available: <http://dx.doi.org/10.1007/s00778-014-0357-y>
- [6] P. Carbone, A. Katsifodimos, S. Ewen, V. Markl, S. Haridi, and K. Tzoumas, “Apache flink<sup>TM</sup>: Stream and batch processing in a single engine,” *IEEE Data Eng. Bull.*, vol. 30, no. 40, 2015. [Online]. Available: <http://sites.computer.org/debull/A15dec/p28.pdf>
- [7] M. Zaharia, T. Das, H. Li, T. Hunter, S. Shenker, and I. Stoica, “Discretized streams: Fault-tolerant streaming computation at scale,” in *ACM SOSP*, 2013. [Online]. Available: <http://doi.acm.org/10.1145/2517349.2522737>
- [8] J. Ding, T. Fu, R. T. B. Ma, M. Winslett, Y. Yang, Z. Zhang, and H. Chao, “Optimal operator state migration for elastic data stream processing.” *CoRR*, vol. abs/1501.03619, 2015. [Online]. Available: <http://dblp.uni-trier.de/db/journals/corr/corr1501.html#DingFMWYZC15>
- [9] R. Sumbaly, J. Kreps, and S. Shah, “The big data ecosystem at linkedin,” in *ACM SIGMOD*, 2013. [Online]. Available: <http://doi.acm.org/10.1145/2463676.2463707>
- [10] o. Verfasserangabe, “Digitale wirtschaft und gesellschaft,” *Bundesregierung*, [https://www.bundesregierung.de/Webs/Breg/DE/Themen/Forschung/1HightechStrategie/1DigitaleWirtschaftGesellschaft/\\_node.html](https://www.bundesregierung.de/Webs/Breg/DE/Themen/Forschung/1HightechStrategie/1DigitaleWirtschaftGesellschaft/_node.html).
- [11] —, “Goldgräber im datenberg,” *Bundesregierung*, [https://www.bundesregierung.de/Webs/Breg/DE/Themen/Forschung/1HightechStrategie/1DigitaleWirtschaftGesellschaft/bigdata/\\_node.html](https://www.bundesregierung.de/Webs/Breg/DE/Themen/Forschung/1HightechStrategie/1DigitaleWirtschaftGesellschaft/bigdata/_node.html).
- [12] —, “Cloud computing,” *Bundesregierung*, [https://www.bundesregierung.de/Webs/Breg/DE/Themen/Forschung/1HightechStrategie/1DigitaleWirtschaftGesellschaft/cloud/\\_node.html](https://www.bundesregierung.de/Webs/Breg/DE/Themen/Forschung/1HightechStrategie/1DigitaleWirtschaftGesellschaft/cloud/_node.html).
- [13] —, “Die intelligente fabrik,” *Bundesregierung*, [https://www.bundesregierung.de/Webs/Breg/DE/Themen/Forschung/1HightechStrategie/1DigitaleWirtschaftGesellschaft/bigdata/\\_node.html](https://www.bundesregierung.de/Webs/Breg/DE/Themen/Forschung/1HightechStrategie/1DigitaleWirtschaftGesellschaft/bigdata/_node.html).
- [14] S. Ghemawat, H. Gobiuff, and S.-T. Leung, “The google file system,” in *Proceedings of the 19th ACM Symposium on Operating Systems Principles*, Bolton Landing, NY, 2003, pp. 20–43.
- [15] J. Dean and S. Ghemawat, “Mapreduce: Simplified data processing on large clusters,” in *Proceedings of the 6th Conference on Symposium on Operating Systems Design & Implementation - Volume 6*, ser. OSDI’04. Berkeley, CA, USA: USENIX Association, 2004. [Online]. Available: <http://dl.acm.org/citation.cfm?id=1251254.1251264>
- [16] G. Wang, J. Koshy, S. Subramanian, K. Paramasivam, M. Zadeh, N. Narkhede, J. Rao, J. Kreps, and J. Stein, “Building a replicated logging system with apache kafka,” *Proc. VLDB Endow.*, vol. 8, no. 12, pp. 1654–1655, Aug. 2015. [Online]. Available: <http://dx.doi.org/10.14778/2824032.2824063>

- [17] J. Kreps, “Kafka : a distributed messaging system for log processing,” 2011.
- [18] A. Toshniwal, S. Taneja, A. Shukla, K. Ramasamy, J. M. Patel, S. Kulkarni, J. Jackson, K. Gade, M. Fu, J. Donham, N. Bhagat, S. Mittal, and D. Ryaboy, “Storm@twitter,” in *Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data*, ser. SIGMOD '14. New York, NY, USA: ACM, 2014, pp. 147–156. [Online]. Available: <http://doi.acm.org/10.1145/2588555.2595641>
- [19] P. Carbone, S. Ewen, G. Fóra, S. Haridi, S. Richter, and K. Tzoumas, “State management in apache flink&reg:: Consistent stateful distributed stream processing,” *Proc. VLDB Endow.*, Aug. 2017. [Online]. Available: <https://doi.org/10.14778/3137765.3137777>
- [20] M. Nasir, G. Morales, D. García-Soriano, N. Kourtellis, and M. Serafini, “The power of both choices: Practical load balancing for distributed stream processing engines,” *CoRR*, vol. abs/1504.00788, 2015. [Online]. Available: <http://arxiv.org/abs/1504.00788>
- [21] —, “When two choices are not enough: Balancing at scale in distributed stream processing,” *CoRR*, vol. abs/1510.05714, 2015. [Online]. Available: <http://arxiv.org/abs/1510.05714>
- [22] B. Gedik, S. Schneider, M. Hirzel, and K.-L. Wu, “Elastic scaling for data stream processing,” *IEEE Trans. Parallel Distrib. Syst.*, 2014. [Online]. Available: <http://dx.doi.org/10.1109/TPDS.2013.295>
- [23] N. R. Katsipoulakis, A. Labrinidis, and P. K. Chrysanthis, “A holistic view of stream partitioning costs,” *Proc. VLDB Endow.*, vol. 10, no. 11, pp. 1286–1297, Aug. 2017. [Online]. Available: <https://doi.org/10.14778/3137628.3137639>
- [24] P. Carbone, G. Fóra, S. Ewen, S. Haridi, and K. Tzoumas, “Lightweight asynchronous snapshots for distributed dataflows,” *CoRR*, vol. abs/1506.08603, 2015. [Online]. Available: <http://arxiv.org/abs/1506.08603>
- [25] T. Qiming, “Are you ready for industry 4.0?” *ICT Insights*, vol. 13, 2015. [Online]. Available: [http://e.huawei.com/en/publications/global/ict\\_insights/201502251048/Special%20Report/201502251639](http://e.huawei.com/en/publications/global/ict_insights/201502251048/Special%20Report/201502251639)
- [26] o. Verfasserangabe, *Sino-German Cooperation on Industrie 4.0*. Bundesministerium für Bildung und Forschung, 2016, [http://www.plattform-i40.de/I40/Redaktion/DE/Downloads/Publikation-gesamt/deutschland-china-kooperation.pdf?\\_\\_blob=publicationFile&v=2](http://www.plattform-i40.de/I40/Redaktion/DE/Downloads/Publikation-gesamt/deutschland-china-kooperation.pdf?__blob=publicationFile&v=2).
- [27] “Flink real-time analysis in CloudStream Service of Huawei Cloud,” <https://data-artisans.com/flink-forward/resources/flink-real-time-analysis-in-cloudstream-service-of-huawei-cloud>.
- [28] B. Del Monte, “Efficient migration of very large distributed state for scalable stream processing,” 2017.


## 8 Appendix: Attachments

### 8.1 High-Performance Workstation

https://www.apple.com/de/shop/bag

Firefox Repo Misc Tech Uni Work GitHub Sign in to Overleaf Vim Cheat Sheet -... tmux shortcuts & ... Online regex teste... Cacao: Or

#### Artikel in deiner Einkaufstasche



**15" MacBook Pro - Space Grau** 3.519,00 €  **3.519,00 €**


Versand: 2-4 Werktage [Entfernen](#)  
Teilenummer: Z0V0

**Hardware**

- 2,2 GHz 6-Core Intel Core i7 Prozessor der 8. Generation (Turbo Boost bis zu 4,1 GHz)
- Retina Display mit True Tone
- Touch Bar und Touch ID
- Radeon Pro 555X mit 4 GB GDDR5 Grafikspeicher
- 32 GB 2400 MHz DDR4 Arbeitsspeicher
- 512 GB SSD Speicher
- Vier Thunderbolt 3 Anschlüsse
- Beleuchtete Tastatur - Englisch, USA
- Zubehörkit

[Geschenkoptionen einblenden](#)


Enthält Urheberrechtsabgaben in Höhe von 10,55 €



**Magic Mouse 2 - Silber** 89,00 €  **89,00 €**

Lieferung: Auf Lager [Entfernen](#)  
Teilenummer: MLA02Z/A

[Geschenkoptionen einblenden](#)



**Magic Keyboard - Englisch (International)** 119,00 €  **119,00 €**

Lieferung: Auf Lager [Entfernen](#)  
Teilenummer: MLA22Z/A

[Geschenkoptionen einblenden](#)

Finanzierung verfügbar.  
Voraussichtliche Monatsraten.

Zwischensumme	3.727,00 €
<b>Kostenfreie Lieferung</b>	0,00 €
<b>Gesamtsumme</b>	<b>3.727,00 €</b>

Enthält die Mehrwertsteuer in Höhe von 595,07 €

1x High-performance workstation with 32 GB of RAM, 6 cores, and 512 GB of SSD

## 8.2 Monitor

The screenshot shows the JACOB Elektronik GmbH website. The main product is a Dell UltraSharp U2518D LED-Monitor. The page includes a product image, a price of €335,00, and a 'Sofort lieferbar' (Immediately available) status. A 'Produktbeschreibung' (Product description) table is visible below the main content.

Produktbeschreibung	
Produktbezeichnung	Dell UltraSharp U2518D - LED-Monitor - 63.44 cm (25")
Gerätekategorie	LED-Beleuchtungsmonitor LCD-Monitor - 63.44 cm (25")
Leistungseigenschaften	USB 3.0-Hub
Bildschirmtyp	IPS
Schermessung	Breitbild - 16:9
Native Auflösung	QHD 2560 x 1440 bei 60 Hz
Pixelpitch	0.216 mm

1x 25 inches Monitor

## 8.3 Cloud Instances

The screenshot shows the Google Cloud Platform console. The 'Compute Engine' section is active, displaying the configuration for 20 instances. The configuration includes 15x instances with 750 total hours per month and 4x instances with 200 total hours per month. The VM class is 'regular' and the instance type is 'n1-standard-16'. The region is 'Frankfurt'. The total estimated component cost is EUR 8,267.73 per 1 year for the 15x instances and EUR 2,103.85 per 1 year for the 4x instances. The total estimated cost is EUR 10,371.58 per 1 year.

20x n1-standard-16 Google Cloud Platform Instances