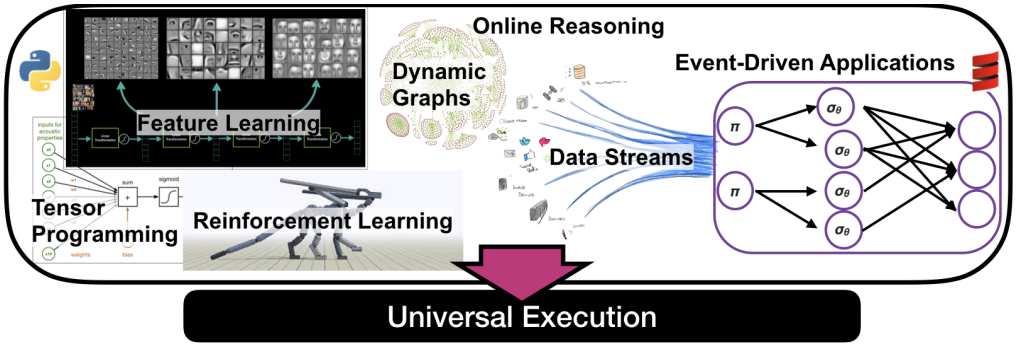
CONTINUOUS DEEP ANALYTICS

Prof. Seif Haridi

Midterm Report



Executive Summary

The Continuous Deep Analytics SSF project addresses a set of core challenges in distributed computing systems for declarative data management and AI in support of critical decision making processes.

> **Research Statement:** *Modern end-to-end data pipelines are highly complex and unoptimised. They combine code from different frontends (e.g., SQL, Beam, Keras), declared in different programming languages (e.g., Python, Scala) and execute across many backend runtimes (e.g., Spark, Flink, Tensorflow). Data and intermediate results take a long and slow path through excessive materialization and conversions down to different partially supported hardware accelerators. This hinders the prospect of reliable actuation and data-driven critical decision making. The Continuous Deep Analytics (CDA) project aims to shape the next-generation systems for scalable, data-driven applications and pipelines for critical decision making. Our work aims to combine state of the art mechanisms in compiler and database technology together with hardware-accelerated machine learning and modern methods in reliable online ML.*

Our areas of focus are defined as follows:

Uncertainty, Dynamicity and Interpretability in Learning Systems: Contemporary ML algorithms and methods produce models that output point predictions, without proper uncertainty quantification, while assuming that training and test instances are drawn from the same (static) underlying distribution. The project will apply and further develop recent frameworks to obtain guarantees on the prediction error and well-calibrated probability distributions, respectively, and employ these frameworks for detecting and adapting to changes in the underlying distributions. The project will also investigate auxiliary techniques for explaining and complementing predictions of popular black-box models (e.g., CNNs, GCNs,) and strive to provide human-understandable knowledge for critical decision making.

Declarative Data Programming: Aiming for simple and clear user-facing data programming models suitable for a seamless declaration of dynamic ML and event-based applications while enabling optimisation and unified execution through exposing special properties in computation, state and time via novel type systems.

Compilation and Code Generation for Continuous Deep Analytics: In particular the efficient lifting of necessary model specific abstractions in intermediate languages (IRs) that allow multiple levels of optimisation. In addition, we aim to exploit the latest advances in tools for optimization and code generation (e.g., Google's MLIR - part of the LLVM library) on hardware accelerators (GPUs, TPUs, Multicore CPUs) in support of various types of data such as tensors (matrices and vectors), streams and relational tables.

Distributed Runtime Support: Including middleware and programming models for building decentralised, high performance dataflow applications as well as proof-of-concept implementations of novel distributed systems that adapt to heterogeneous hardware and are resilient to failures while dealing with mixed batch and streaming workloads and dynamic processing requirements on shared resources for long-running executions.

1 Project Background

The CDA project aims to provide the foundations of continuous deep analytics from the problem domain to the specification down to code generation and distributed execution of diverse tasks on heterogeneous computing platforms. This section discusses all the interdisciplinary areas, the specific problems and milestones that we aim to achieve throughout the course of the project and beyond.

1.1 MOTIVATION, AND LONG-TERM VISION

CDA is a five-year project financed by the Swedish Foundation of Strategic Research. The project began in Q3 2017 by a group of ML and computer system researchers at KTH Royal Institute of Technology and RISE Digital Systems (former Swedish Institute of Computer Science) in Stockholm, Sweden. The project has a concrete vision: to lay the foundations of Continuous Deep Analytics, a new computing approach that enables the creation and actuation of live interpretable models of the world. Since its conception, the CDA project has enabled the formation of a unique research group consisting of active "data science" practitioners, theoretical ML researchers, programming language experts and distributed computing researchers under a unified purpose.

The need for continuous deep analytics is becoming increasingly evident in the last years. Large-scale machine learning methods, a prime focus of research and industry, have evolved and been optimised to scale the training of black box models to large amounts of input data in a nearly linear fashion with respect to accuracy. As an example, GPT-3 by OpenAI is the current landmark in language prediction "black-box" models providing a capacity of 175 billion machine learning parameters. At the same time, less effort has been spent in addressing core limitations of large-scale machine learning such as lack of support for interpretability, programmability, and seamless integration with critical decision making processes that are important at societal scale. In practice, many hard challenges surface when practitioners in data engineering and data analytics leave behind model prototyping and testing and attempt to create reliable continuous services and applications that automate and enrich time-critical processes. Examples of such services include anomaly detection pipelines, adaptive recommender systems, time-series forecasting, real-time monitoring and control (e.g., traffic, power plants, remote sensing) in smart cities and more generally applications that are 1) continuous and 2) sensitive to change in real-world trends (i.e., concept drift) and can be 3) responsible for executing parts of corresponding business logic. Supporting such application requirements using existing technologies today requires complex engineering and rare domain expertise since all data science, programming and configuration tasks have to be addressed in combination. To that end, we identify three core challenges that motivate the use of Continuous Deep Analytics systems to program, compile and run continuous applications that are constantly adaptive to data changes and can overtake critical decision making tasks.

1.1.1 The Declarative Data Programming Challenge

Individual data processing frameworks used today serve the sole purpose of providing a set of operations on specific types of data. Typical examples include Tensorflow which specialises on linear algebra / matrix transformations, Flink/Beam which specialise in event-based stateful logic, Spark which supports scalable operations and ad-hoc queries on DataFrames (relational data) and Ray that specialises on simulations and reinforcement learning task programming. While a combination of different frameworks can in part fulfil the creation of continuous deep analytical workloads, there is no programming model that can integrate seamlessly all these types of workloads under a single program. This poses a hard requirement in creating critical services: the combination of rare expertise in engineering, machine learning and other domains which could be alleviated if a set of high level abstractions were available within a single domain specific language (DSL) on top. This would also further allow for common efficient compilation and code generation for heterogeneous hardware which is not feasible today using composite, multi-framework solutions.

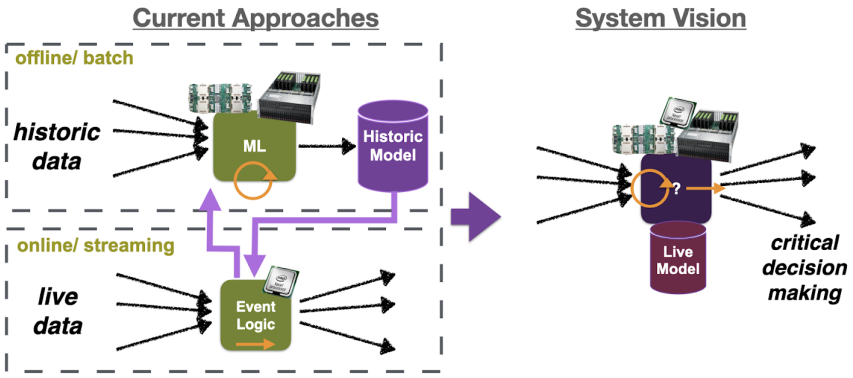


Figure 1: Approaches to Critical Decision Making.

1.1.2 The Distributed Continuous System Execution Challenge

An integral part of a data processing framework is its execution engine. Typically, computation is distributed and scheduled to run on multiple compute nodes by each runtime in parallel while aiming for a specific workload such as “stream execution” on live data or “bulk iterative execution” on historic data employed e.g., by streaming systems or batch processing systems respectively. In addition, each runtime provides its own unique set of processing guarantees, fault tolerance and support for hardware accelerators according to its needs. The execution of data-driven continuous applications requires a combination of different runtime technologies to incorporate offline deep analytics (e.g., ML model training) and online preprocessing and inference (e.g., continuous stream computation). This leads to data pipelines consisting of fragmented codebases that run separately across system “silos”, with no shared utilisation of hardware, in turn causing excessive materialisation of intermediate results (e.g., large ML models communicated through big files). Therefore, modern data pipelines are complex and dominated by unnecessary data synchronisation latencies that make critical decision making impossible.

As of today, the creation of a distributed execution engine that can deal with mixed workloads and heterogeneous hardware remains an open research problem of its own. Nevertheless, it is a crucial requirement for continuous deep analytics to be able to integrate different executions under the same runtime.

1.1.3 The Reliable Online Learning Challenge

As machine learning is increasingly used not only for decision support, but also automated decision making, trust in the resulting decisions or recommendations becomes vital. Consequently, how to make machine learning solutions reliable is today a key question addressed by researchers from many disciplines. Being able to explain the logic behind the predictions of a machine learning model is widely considered to be one of the cornerstones for enabling trust. However interpretable models that fulfil this criterion, can only rarely be used as direct replacements for the strongest black-box models. Instead, attention has shifted towards approaches to bridging the gap between (strong) black-box models and (weak) interpretable models, which allow predictions of the black-box models to be explained using (local or global) interpretable approximations. Another cornerstone for enabling trust is that the uncertainty of the output of the machine learning models is properly quantified, for example, that the output class probability distributions and confidence intervals are well-calibrated. Finally, the learning system should also be able to detect shifts in the underlying data distributions over time, and adapt accordingly.

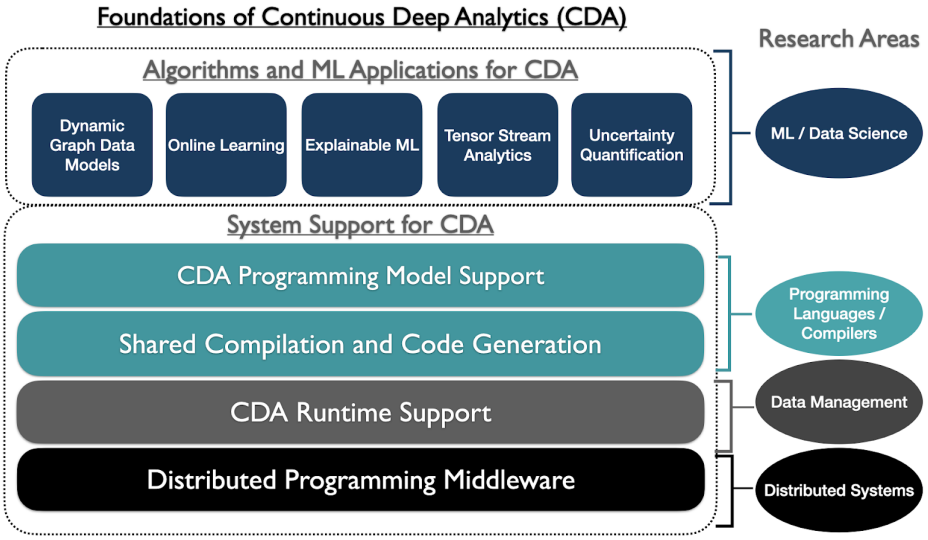


Figure 2: An Overview of Goals and Objectives for Continuous Deep Analytics.

1.2 CONCRETE GOALS AND OBJECTIVES

The CDA project goals constitute all aspects that define continuous deep analytics, from the conceptual models and algorithms to concrete system specifications from the programming model down to the execution of data computing tasks. As shown in figure 2, all objectives are organised hierarchically to provide the necessary requirements of CDA bottom-up from the system execution to the programming abstractions and applications that are going to be expressed in the proposed system implementation. The objectives of each of the underlying areas and domains are further summarised below.

1.2.1 Algorithms and ML Applications for CDA

Current ML and data engineering methods are tailored to fit a particular model of computation. Thus, a core objective of the project is to identify emerging problems that cannot be supported by existing methods and instead require new computational models or a combination of existing models. Among the problem domains investigated in the project is the field of dynamic graphs. Graph data is used today for analysing complex social networks, creating recommender systems, analysing traffic networks in large cities, 5G telecommunication performance monitoring and modelling of medical data (e.g., chemical interactions, medicine side-effects, classification of cancerous cells etc.). More recently the emerging field of “Graph Embedding Models” has proven to be a promising direction for composing highly-accurate predictive models out of massive graphs e.g., via the use of Convolutional Neural Networks (CNNs). However, the applicability of such accurate models is currently limited for critical decision making due to continuous concept drift in the data that invalidates their accuracy, coupled by slow training times. From its foundation, CDA will tackle this problem by laying down a formal specification of dynamic graph embeddings and the necessary system requirements to support it. The objectives related to the Reliable Online Learning Challenge include: online learning, i.e., efficient learning from streaming data, uncertainty quantification, i.e., the uncertainty of the predictions should be properly quantified (well-calibrated), concept drift, i.e., the learning system should be able to detect and adapt to changes in the underlying distributions, and explainable machine learning, i.e., the logic behind the predictions should be understandable by humans.

1.2.2 Programming Model and Multimodel Compilation for CDA

A core objective of CDA is to define a unified programming model for continuous and deep analytics. This entails data types and operators that involve data transformations, serving and querying (e.g., relational and stream operators) as well as model creation and reasoning (e.g., tensors, uncertainty quantification). Furthermore, to deal with the dynamic properties of data the target programming model should provide build-in support for concept drift detection and adaptation and reliable discretisation (e.g., stream windows) while making runtime concerns transparent to the user. In addition to language/model

specification challenges, a sophisticated compiler infrastructure is necessary to allow optimal code generation across operators with distinct workload and semantic differences, yet, with the common need to undertake their combined execution on shared hardware. A promising direction is the use of “Intermediate Representations” (IRs) as a common ground while exploiting active open source libraries for code compilation such as LLVM and MLIR, used internally in Tensorflow, among other systems.

1.2.3 Runtime Support for CDA

CDA poses new requirements in data management and runtime execution, questioning the purpose of current data processing technologies and opening new directions in the design of computational systems. At one end systems that are designed for continuous execution are incapable of supporting heavy ML workloads such as the bulk iterative workflows of Tensorflow. While at the other end, systems that specialise in batch workloads fail to meet the real-time processing requirements of critical applications. The suitable hardware across these domains of computation also differs significantly with GPUs and TPUs being at the focus of ML systems and cache-centric multicore CPUs being the dedicated hardware of continuous processing systems (e.g., Apache Flink) . Therefore, in CDA it is crucial to explore a new runtime design that can exploit heterogeneous hardware for mixed workloads and to seek novel approaches to share active application state across different representations. The latest advances in embedded databases such as Log-Structure-Merge storage and adaptive indexing are among others a promising direction in CDA’s vision runtime.

1.2.4 Middleware Support for Reliable and Efficient Distributed Programming

Writing correct and efficient distributed software is a challenging task at the best of times, yet the solutions to many common issues can be—and often are—abstracted away into programming languages and frameworks that are specifically designed for this environment. However, existing languages and frameworks, such as Erlang or Akka, have traditionally focused on abstractions aimed at relatively high-level application development and in the process often compromised on the side of convenience over performance. When it comes to efficient distributed data processing, however, such a compromise is not acceptable; in fact convenience should come second to performance in almost all decisions in a data analytics runtime. Achieving high performance in such a framework requires abstractions that empower runtime developers, for example to use the best communication strategy for each individual algorithm in a system, rather than being forced into a one-size-fits-all solution as is commonly provided by existing Actor systems. Finding these empowering abstractions and combining them into a distributed programming middleware underlying the actual analytics runtime is an important goal of the CDA project, as it allows our platform to be efficient, correct, and maintainable in the long run.

1.3 THE START-UP PROCESS

The project has been executed according to initial plans with no important deviations in terms of vision, results and resource management. As planned, the first two years of the project were used to train and develop the skills of newly hired doctoral students as well as integrating work and transferring knowledge from experienced doctoral students that were close to graduation in the group. This initial work has been fundamental to the project, with contributions ranging from core software libraries, demonstration and vision papers to high-impact publications in top conferences. From the beginning of the third year onwards more resources were used to boost the development of our new “CDA Platform” and its counterparts, including middleware development, interns collaborating on planning and development, workshops and additions of more senior members to the group.

1.4 THE BASIC ORGANISATION, LEADERSHIP, RESEARCH ENVIRONMENTS, RELATION TO OTHER GRANTS, ETC.

The project composition consists of a series of work-packages organised based on the concrete area of their respective objectives. Work Packages 3 and 4 that are centred around algorithms and ML applications have been coalesced into a single work-package. The main project coordination is conducted by Prof. Seif Haridi while the work-package Principal Investigators (PIs) are shown below:

WP1 Dataflow Runtime with Elasticity in Time, Resources and Uncertainty (Christian Schulte, *Paris*

Carbone since 2019).

WP2 Declarative Programming Models (Seif Haridi)

WP3 (previously WP3+WP4) Algorithms and ML Applications for CDA (Daniel Gillblad, Anders Holst, *Henrik Boström since 2019*)

Related Grants

- EU-horizon 2020 StreamLine project (Dec. 2015 to Nov. 2018)
- Swedish Governmental Agency for Innovation Systems (VINNOVA), Predictive Models with Interpretability and Concept Drift Analytics (Oct. 2018 to March 2021)
- Swedish Governmental Agency for Innovation Systems (VINNOVA), Resultaten i staten (Oct. 2019 to Sep. 2021)

1.5 STEERING GROUP

Several changes in the steering group of the project have occured from its conception due to change of affiliation and retirement/loss. Professor Seif Haridi holds the overall management of the project as it was set in its initial proposal and vision. Professor Christian Schulte who was originally responsible for WP1 passed away in March 2020 while Professor Henrik Boström (KTH) effectively replaced Anders Holst (RISE) and Daniel Gillblad (RISE) in 2019 as the responsible for the ML research work in WP3-WP4. Furthermore, Lars Kroll (RISE) and Paris Carbone (RISE) became senior contributors to the project following their PhD graduation at KTH in Jan-2020 and Nov-2018 respectively. The latter also maintains an Assistant Professor position at KTH and is currently leading the efforts of the CDA project at RISE. None of the aforementioned changes in the steering group has affected the progress of the project.

2 The Research of CDA

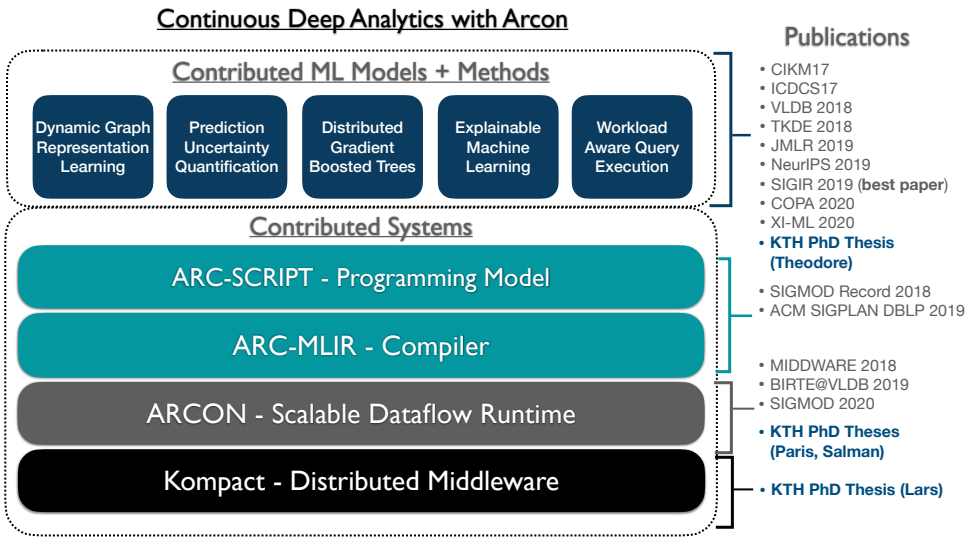


Figure 3: An Overview of the Research of CDA.

Research in CDA has been driven by the vision of designing and creating a complete system from the ground up that can enable continuous and deep data analytics. Figure 3 highlights most major research achievements in the CDA project published at top A+ venues while following the same layered architecture of objectives presented in Figure 2. Alongside individual achievements major efforts were also spent into putting all contributions together to compose Arcon the first known open source software platform that can achieve continuous and deep data analytics, that is a distinctive challenge and contribution on its own. The rest of this section summarises our published research work as well as the corresponding system software libraries that have been released for open source use as part of the **Arcon Continuous Deep Analytics Framework**.

2.1 SCIENTIFIC RESULTS

2.1.1 Contributions in Machine Learning Models and Methods

Prediction Uncertainty Quantification

The purpose of this part of the project is to investigate core challenges in reliable machine learning to form the core system requirements for online ML and continuous deep analytics. In this line of work, we have so far looked into the following directions: i) defining and quantifying model uncertainty and ii) online learning—algorithms and requirements. This has resulted in a study on uncertainty estimation of online random forests published in JMLR, the top machine learning journal (Vasiloudis et al, JMLR, 2019). These methods are aimed at quantifying the uncertainty of predictions in domains where mistakes are costly, like the medical and financial domains, and decisions need to be made under a tight time and computational budget. The main results from that study were also presented at the Systems for ML workshop at NeurIPS (Vasiloudis et al, NeurIPS, 2019). The project has also contributed with a novel approach for uncertainty quantification, called Mondrian Conformal Regressors (Boström & Johansson, 2020), which overcomes two weaknesses of the original conformal regression approach; i) predicted intervals may be several times larger or smaller than any previously observed error, and ii) the variance of the sizes of the predicted intervals is not positively correlated to the available information on the relative difficulty (prediction quality) of the predicted instances. In contrast, Mondrian conformal regressors can never produce intervals that are larger than twice the largest observed error. Moreover, a large-scale empirical investigation was conducted, showing that Mondrian conformal regressors have the desired property that the variance of the size of the predicted intervals is positively correlated with the accu-

racy of the function that is used to estimate prediction quality. In (Werner et al, 2020), a large-scale investigation of conformal predictive systems (CPSs) is presented. Such systems output probability distributions for real-valued labels of test examples, rather than point predictions (as output by regular regression models) or confidence intervals (as output by conformal regressors). The results show that by using either variance or the k-nearest-neighbour method for estimating prediction quality, a significant increase in performance, as measured by the continuous ranked probability score, can be obtained for regression forests, compared to omitting the quality estimation. The results furthermore show that the use of out-of-bag examples for calibration is competitive with the most effective way of splitting training data into a proper training set and a calibration set, without requiring tuning of the calibration set size. In (Karlsson et al, 2020), inductive conformal prediction was applied to a dataset of laboratory-generated aerosol particles, i.e., small airborne particles suspended in air affecting the climate and human health, consisting of ten particle subclasses that were grouped into four parent classes. Different types of particles come from different sources and impact the environment in different ways, which is why a reliable particle classification is of interest. The performance of the inductive conformal predictor (ICP) was evaluated on particle subclasses that were not included in training or calibration and was shown to give accurate predictions in some cases, namely if the unknown particle is similar to the known ones in the parent class. The precision of the underlying model was not high enough to reject all unknown particles for any subclass at the chosen significance levels, but the ICP managed to reject them at a higher rate if they were sufficiently different from the training and calibration samples.

Scaling Out Algorithms for Online ML

Following a collaboration with Amazon AWS (Seattle), one focus of the project has been on the scalability of distributed gradient boosting trees. Methods that allow scaling along both the data and feature dimension (block-distribution), improving the scalability characteristics, were proposed and through proper use of sparsity are able to reduce the communication cost of the algorithm by orders of magnitude for highly sparse data. The corresponding paper (Vasiloudis et al, SIGIR, 2019) received the best short paper award at SIGIR, which is the top conference on Information Retrieval. These papers were also included in the Ph.D. thesis *Scalable Machine Learning through Approximation and Distributed Computing* (Vasiloudis, 2019).

Explainable Machine Learning

In (Boström et al, 2020), techniques for explaining forecasting models for multivariate time-series are investigated. Various approaches to explaining predictions of black box models have been proposed in the past, including model-agnostic techniques that measure feature importance (or effect) by presenting modified test instances to the underlying black-box model. These modifications typically rely on choosing feature values from the complete range of observed values. However, when applying machine learning algorithms to the task of forecasting from multivariate time-series, we suggest that the temporal aspect should be taken into account when analysing the feature effects. In (Boström et al, 2020), we proposed a modification of individual conditional expectation (ICE) plots, called ICE-T plots, which displays the prediction change for temporally ordered feature values, and demonstrated, in collaboration with the Swedish National Financial Management Authority, its use through a case study on predicting the Swedish gross domestic product (GDP) based on a comprehensive set of indicator and prognostic variables.

Dynamic Graph Representation Learning

The CDA project fuelled a novel research direction in ML for complex data that led to several contributions, industrial and research collaborations and newly submitted publications. As mentioned in section 1.2.1 the creation of graph models that encapsulate deep properties in large networks (e.g., medicine, road traffic, financial and social networks) is a promising yet challenging direction for predictive analytics. During the initial course of the project we explored methods for graph summarization (top-k densest subgraphs - CIKM17) and distributed graph partitioning (node-cut partitioning on dense weighted graphs - ICDCS17). Furthermore, we investigated the use of Convolutional Neural Networks for creating deep graph embeddings also known as Graph Convolutional Networks (GCNs). More particularly we studied the challenges related to the creation and maintenance of GCNs on unbounded data and their effectiveness in supporting real-time processing problems for complex data such as anomaly detection, feature prediction and stream data partitioning. Among other other research results (under

submission) and three novel MSc theses this line of work aided Swedbank in a collaboration that aimed to identify fraudulent transactions in real-time. More importantly, it served as a baseline application that guided the design of the Arcon system from its programming model to execution requirements.

2.1.2 Contributions in Computer Systems Research

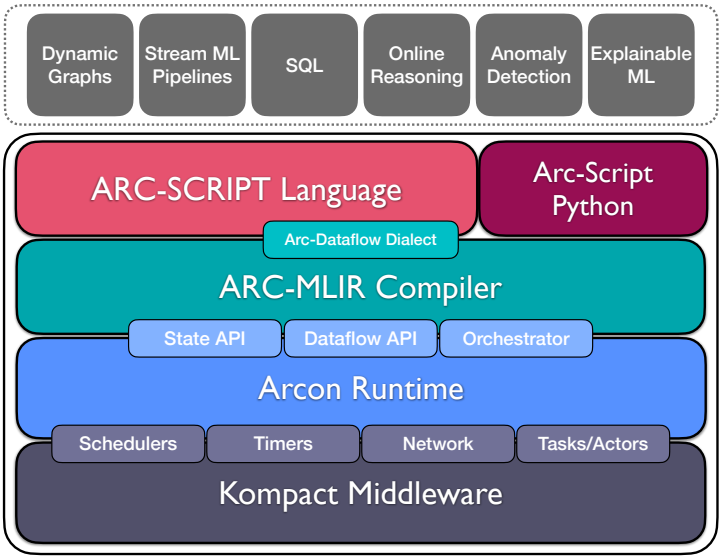


Figure 4: The Arcon System Software Stack.

The project contributions in systems research have touched open problems in compiler technology as well as declarative models and data management. Figure 4 depicts the main components of the system which aims to offer a unified way to program and execute data pipelines using 1) fundamentally diverse data types that have been traditionally supported by different workloads (tensors, relational tables, streams), 2) different frontend languages (e.g., Scala, Python) interchangeably and 3) support heterogeneous hardware such as GPUs, multicore CPUs, FPGAs etc.. To achieve this we separate our model (Arc-Script), compilation unit (Arc-MLIR) and execution runtime (Arcon Runtime) through the use of flexible intermediate representations (Arc Dialect) that serve the purpose to capture the essence of data transformations in a hardware-agnostic way while allowing a level of logical optimisation before code generation. Furthermore, the Arcon Runtime can execute and scale out dataflow pipelines of computational tasks that are generated by the ARC-MLIR compiler while also exposing consistent access to the state of running applications for exploratory data analysis.
Arcon builds upon our group's previous expertise and history in scalable system design such as Apache Flink, Hops, Kompics and Distributed Oz. In essence, Arcon is a programming system with high performance, agility and transparency for building high performance continuous applications. The most critical libraries of Arcon such as schedulers, timers and network channels are implemented in Kompact, a Rust language-based middleware that we have developed in the group to support high-performance computing at scale.
Below we summarise the components and research of Arcon from a top-down dependency perspective including *Arc-Script* (programming model), the *Arc Intermediate Representation* and its LLVM-based compiler, the *Arcon Dataflow Runtime Engine* and the *Kompact* distributed Middleware.

Arc-Script, the Arc Intermediate Representation and MLIR Compiler

Arc-script is a model and DSL in the making that targets the user-facing interface to the Arcon Runtime. It stands in the space between a general-purpose language and a specialised language or model such as DataFrames or SQL. Essentially the goal is to support the common ground between streams, relations, arrays/tensors and graphs which has not been exploited yet (depicted by '???' in Figure 5). Arc-Script aims tight integration with TensorFlow to support training and serving of machine learning models with high performance. Programs in Arc-Script are first translated into Arc, an Intermediate Representation (IR) we designed to enable the cross-optimization of batch and stream computation. The first version of Arc was based on Weld, an IR for batch operations (collaboration with Stanford

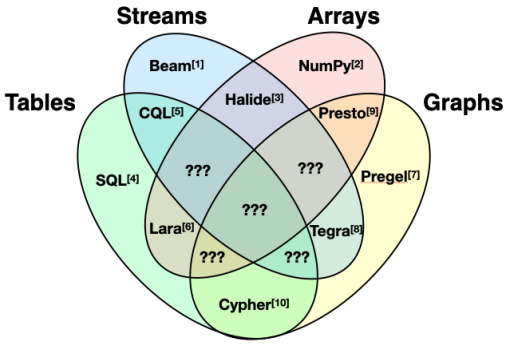


Figure 5: Open Domains in Data Programming.

University and Matei Zaharia) and published in 2018 at the prestigious ACM SIGPLAN DBPL. Currently, Arc is contributing to the MLIR project (Multi-Level Intermediate Representation), an active effort for supporting intermediate languages and common compiler infrastructure for data analytics. MLIR allows expressing computations across different problem domains. It was initially contributed by Google to the LLVM ecosystem. An MLIR extension is called a "dialect" and multiple dialects in MLIR can coexist in the same program. Using MLIR further allows us to reuse functionality and optimizations provided by, for example the TensorFlow dialect, and their optimisations. The preliminary work on Arc-Script already introduces significant novelties to the space of language support for data analytics in addition to our MLIR dataflow dialect (Arc) for continuous deep analytics. Arc-Script is currently in active development and undergoing paper submissions to top-tier conferences and journals.

The Arcon Dataflow Runtime

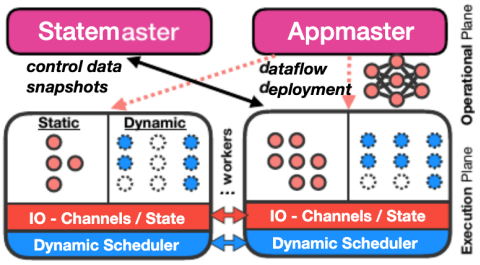


Figure 2: Runtime Overview

Figure 6: Hyrbid Execution in Arcon.

Arcon is a data processing runtime built from the ground up to support hybrid batch and stream pipelines. At its current state it provides support for stream operators such as stream windows and a novel state management layer using the latest advances in embedded database technology. Already at its early stage, Arcon has attracted the attention of data engineering groups as well as top academic communities due to its novel design and promising performance. Our group presented Arcon at BIRTE the premier real time-analytics venue at VLDB in Los Angeles 2019 as well as the research track of Flink Forward 2019, the most popular event in stream processing technology. Arcon has also gained visibility in the Swedish ICT at the Data Science Summit 2019, CASTOR events organized by KTH for the Swedish industry as well as the Chaos Engineering conference in Stockholm in 2019. Industrial partners such as King and Ericsson have also expressed interest into putting Arcon into practice and experimenting with its capabilities once the first production-ready release is out. Arcon can run code generated by Arc and orchestrate it to utilise scalable state and computational resources available in a cluster. The core idea of Arcon is to eliminate the need to i) use different frameworks to execute hybrid analytical workloads such as an ML training and serving pipeline with event-based business logic, and ii) provide a foundation for task- and data-parallel execution that will unlock new possibilities in the way we express data applications. Potential next features of Arcon are a stream feature store for ML programs (online feature engineering + ad-hoc data analysis), stateful serverless programs, a framework for online stream ML etc. Arcon makes use of a typed actor programming Rust middleware we have also developed in-house at RISE and KTH which is called Kompact.

Below we summarise a few highlights among currently available features in Arcon.

Workload-Aware State Management: Unlike existing dataflow engines such as Apache Flink which are pre-configured with a single embedded database such as RocksDB (log-structured merge-tree design), Arcon allows for an active use of any state backing infrastructure (e.g., indexed state in B+ Trees)

which can also be adaptively configured while the application is running based on its current workload similarly to modern database systems.
Scalable Time Triggers: Time operations such as session window triggers are common in continuous dataflow applications. In comparison to conventional stream processors Arcon uses the "Time Wheels" data structure as a novel way to keep track of millions of time events quickly and efficiently. This part of the Arcon library in addition to its high-performance networking are implemented using the Kompact Framework.
Native Execution Speeds: Event-based logic in Arcon can reach an order of magnitude better performance (throughput/latency) according to the NEXMark benchmarks, compared to state of the art dataflow systems due to its Rust-based implementation and code generation (through Arc) that allows optimisation capabilities at the instruction level and low-level memory management.

Kompact

Kompact is a programming framework that implements a novel hybrid message-passing programming model, combining the strengths of the Actor model and the Kompics component model. It does so by providing developers with the choice of three different messaging semantics at a fine grained level, so that every algorithm can utilise the best available guarantees and avoid inefficiencies arising from semantic mismatches. Additionally, Kompact's embedding in the Rust language produces efficient, statically typed, native code, produced and optimised for a variety of target platforms by the popular LLVM compiler backend.

At a high level, Kompact provides a light-weight process abstraction with exclusive internal state—called a component—that can communicate with other components by exchanging discrete pieces of information, called messages or events. As in other message-passing models, such as the Actor model or the Kompics component model, this design allows services written in Kompact to scale with load. It does so by efficiently utilising modern multi-core architectures, without introducing bottlenecks such as lock contention, which are common in shared-memory programming models. However, Kompact's reliance on static typing and the benefits of Rust's powerful memory management and compiler optimisations, have allowed it to show performance benefits over other state-of-the-art message-passing frameworks of up to 27×.

Not only does the component-based design of Kompact avoid scalability bottlenecks, it also provides the means for modular composition of abstractions and subservices into larger services and subsystems. This approach allows the Arcon runtime, which is built on top of Kompact, to remain both maintainable in the long term and flexible enough to allow future research on different subsystems by simply exchanging them as desired. Additionally, modularity allows testing and verification of the code at different levels of abstraction. This benefit, together with the compile-time guarantees of static typing, helps us to develop and maintain correct implementations of the set of reliable distributed services that make up unified stream and batch runtime, such as Arcon.

Scalable File Systems

The project also contributed to the design of scalable distributed storage/file systems that now runs as a managed system in cloud infrastructures including multi-datacenter clouds. The main work is on the HopsFS distributed file system for storing and managing large data sets. In particular one work in the project is to extend HopsFS to efficiently manage and store files of different varying sizes. Traditional cloud-based storage systems only efficiently manage files consisting of large block sizes. This is not optimal as a backend to ML training systems that for example in image recognition training where an image is a moderately small file. The work results in publications at ACM Middleware 2018, and a Ph.D. thesis by Salman Niazi. The work is integrated in HopsFS currently part of product portfolio of the start-up LogicalClocks[1].

2.2 PARTICIPATING RESEARCHERS

The CDA project funds a large group of researchers in computer systems, programming languages and machine learning. The full list is as follows:

[1] https://logicalclocks.com

Senior Researchers

- Seif Haridi, Professor at KTH — Chief Scientist at RISE
- Henrik Boström, Professor at KTH
- Christian Schulte, Former[2] Professor at KTH
- Sarunas Girdzijauskas, Associate Professor at KTH
- Daniel Gillblad, Lab Manager at RISE
- Paris Carbone, Senior Researcher at RISE | Assistant Professor at KTH
- Lars Kroll, Senior Researcher at RISE
- Frej Drejhammar, Senior Research Engineer at RISE

Completed Ph.D. Students

- Lars Kroll: Compile-time Safety and Runtime Performance in Programming Frameworks for Distributed Systems (2020)
- Theodore Vasiloudis: Scalable Machine Learning through Approximation and Distributed Computing (2019)
- Paris Carbone: Scalable and Reliable Data Stream Processing (2018)
- Salman Niazi: Scaling Distributed Hierarchical File Systems Using NewSQL Databases (2018)

Current Ph.D. Students

- Klas Segeljakt: Language Support for Continuous Deep Analytics
- Max Meldrum: In Support of Hybrid Analytics on Modern Hardware

2.3 LIST OF PUBLICATIONS

Journal Publications

- T Vasiloudis, GDF Morales, H Boström – Quantifying Uncertainty in Online Regression Forests, **Journal of Machine Learning Research** 20 (155), pp. 1-35, **2019**
- Z Abbas, V Kalavri, P Carbone, V Vlassov – Streaming graph partitioning: an experimental study, **PVLDB 2018**
- V Kalavri, V Vlassov, S Haridi – High-level programming abstractions for distributed graph processing, **IEEE Transactions of Knowledge in Data Engineering** (TKDE Journal) **2018**

Conference Publications

- P Carbone, M Fragkoulis, V Kalavri, A Katsifodimos – Beyond Analytics: the Evolution of Stream Processing Systems, **ACM SIGMOD 2020**
- H Boström, U Johansson – Mondrian Conformal Regressors, **COPA 2020**
- H Boström, P Höglund, S-O Junker, A-S Öberg, M Sparr – Explaining Multivariate Time Series Forecasts: an Application to Predicting the Swedish GDP, **XI-ML 2020**
- L Karlsson, H Boström, P Zieger – Classification of Aerosol Particles using Inductive Conformal Prediction, **COPA 2020**
- H Werner, L Carlsson, E Ahlberg, H Boström – Evaluating Different Approaches to Calibrating Conformal Predictive Systems, **COPA 2020**
- T Vasiloudis, H Cho, H Boström – Block-distributed Gradient Boosted Trees, **ACM SIGIR 2019**

[2] https://intra.kth.se/eecs/aktuellt-pa-eecs/nyheter/in-memory-of-christian-schulte-1.969552

- L Kroll, K Segeljakt, P Carbone, C Schulte, S Haridi. – Arc: an IR for batch and stream programming, **DBPL 2019**
- S Niazi, M Ronström. S Haridi, J Dowling – Size Matters: Improving the Performance of Small Files in Hadoop, **ACM Middleware 2018**
- Muhammad Anis Uddin Nasir, Aristides Gionis, Gianmarco De Francisci Morales, Sarunas Girdzijauskas – Fully Dynamic Algorithm for Top-k Densest Subgraphs. **CIKM 2017**: 1817-1826
- Kambiz Ghoorchian, Sarunas Girdzijauskas, Fatemeh Rahimian – DeGPar: Large Scale Topic Detection Using Node-Cut Partitioning on Dense Weighted Graphs. **ICDCS 2017**: 775-785

Workshop Papers

- M Meldrum, K Segeljakt, L Kroll, P Carbone, C Schulte, S Haridi – Arcon: Continuous and Deep Data Stream Analytics, **ACM BIRTE** at VLDB **2019**
- T Vasiloudis, H Cho, H Boström – Block-distributed Gradient Boosted Trees, Workshop on Systems for ML at NeurIPS **2019**

Ph.D. Dissertations

- Salman Niazi – Scaling Distributed Hierarchical File Systems Using NewSQL Databases, KTH Royal Institute of Technology 2018
- Paris Carbone – Scalable and Reliable Data Stream Processing, KTH Royal Institute of Technology 2018
- Theodore Vasiloudis – Scalable Machine Learning through Approximation and Distributed Computing, KTH Royal Institute of Technology 2019
- Lars Kroll – Compile-time Safety and Runtime Performance in Programming Frameworks for Distributed Systems, KTH Royal Institute of Technology 2020

Technical Reports

- M Fragkoulis, P Carbone, V Kalavri, A Katsifodimos – A Survey on the Evolution of Stream Processing Systems, **arXiv** preprint arXiv:2008.00842 **2020**
- S Sakr, T Rabl, M Hirzel, P Carbone, M Strohbach – Dagstuhl Seminar on Big Stream Processing, **SIGMOD** Record **2018**

Awards

Best Short Paper Award:

- T Vasiloudis, H Cho, H Boström – Block-distributed Gradient Boosted Trees, **ACM SIGIR 2019**

Posters

- P Carbone, L Kroll, K Segeljakt, M Meldrum, A Hasselberg, C Schulte, S Haridi – Continuous Deep Analytics (CDA)
- K Segeljakt, F Drejhammar – Using MLIR to implement a compiler for Arc, a language for Batch and Stream Programming

2.4 FUTURE RESEARCH PLANS

Our future plans in the project build on top of the CDA runtime and programming model foundations that were established in the first half of the project. The majority of the upcoming tasks in the project involve automating the optimisation and and execution of the Arcon system, adapting to workload and data changes and thus, improving its speed, reliability and accuracy over time which is termed necessary on long-running executions. Furthermore, we are planning to expand further the capabilities of the Arc-Script programming model with core programming support for dynamic ML and a Python interface.

Most importantly, we plan to demonstrate the capabilities of the system to deal with live changes in the data or hardware resources while providing realistic estimations of model error and uncertainty based on the methods we have investigated so far. Below we summarise the main plans within each individual area of the project.

2.4.1 Reliable Online ML

Algorithms developed in the first part of the project, e.g. for prediction uncertainty quantification, online learning and explainable machine learning, are to be implemented on top of the CDA runtime, primarily via Python APIs. The research on explainable machine learning will be extended to include also other formalisms for explaining predictions, such as rules, to approximate the reasoning of black box models, still with a focus on temporal dependencies. Techniques for efficiently querying the black-box models and processing large amounts of streaming data are here crucial for providing such explanations on-demand in an online manner. The frameworks for uncertainty quantification, in particular conformal and Venn prediction, will be adapted to concept drift detection. Strategies for effectively handling detected drifts will be explored and evaluated. Again, the implementation of these techniques will benefit significantly from exploiting the CDA runtime, allowing for continuous detection and adaptation at high data rates.

2.4.2 Programming Support for CDA

We plan to extend our work in Arc-Script and Arc-MLIR with new operations and a Python interface which is expected to bring more demonstrators to the project. We further plan for a tighter integration with our current demonstrator applications as well as the distributed runtime. More concretely, we are looking into expressing applications that involve preprocessing, training and serving of ML models over data streams of constantly changing characteristics. Among different directions, we have been looking into the expression and optimisation of continuous graph embedding pipelines in Arc-Script which can be used to predict missing properties of complex data in real-time. Furthermore, we seek for high confidence predictions which can be employed e.g., via the use of conformal prediction and uncertainty estimation models we have developed so far which can facilitate inference. To this end, we have started a close collaboration with research groups in the University of Edinburgh (Database Group) and Boston University (Computer Science Group). Finally, we plan to keep collaborating to the MLIR-LLVM ecosystem with new compilation model dialects and novel cross-domain optimisations for data analytics such as those that lay in the intersection of stream and ML operations, e.g. incremental training and array operations on stream windows. This is also a direction we have started investigating together with the Large-Scale Data and Systems group at Imperial College of London, targeting modern hardware accelerators such as GPUs and TPUs.

2.4.3 Runtime Support for CDA

Arcon and Kompact have become two independent active open source projects that we plan to further use as vehicles in our future research distributed computing. A set of research publications and journals are also currently being prepared for submission at top systems venues such as USENIX OSDI, MIDDLEWARE and SIGMOD. Below we summarise our upcoming plans on these projects:

Arcon: We plan to make Arcon a more independent system with its own Rust programming API for composing distributed stream and batch pipelines with distinct features. One upcoming core addition and open research topic we have started investigating is runtime optimisation and reconfiguration, allowing Arcon to tune itself, recompile and change its running tasks according to changes in the data workloads. Among other methods such as constraint solving, we plan to look into the promising direction of AutoML, a set of techniques that utilise ML-driven optimisation to tune systems. Furthermore, we plan to add transactional processing and state isolation guarantees in Arcon at the presence of tasks that communicate with external systems, as well as related implications of iterative computing and external state query support.

Kompact: As the most general-purpose software developed in the project, Kompact will be used as the main middleware to prototype and benchmark new distributed computing protocols designed in our group (e.g., on distributed consensus, integration with IoT and cloud services). In particular, there is

already ongoing work on implementations of the well-known Paxos and Raft algorithms, which we plan to publish in the future as both a reusable library and a paper comparing their respective performance in different real-world scenarios. Furthermore, we plan to add a set of distinct features in Kompact such as configurable network channels, flow control strategies and schedulers which will allow for more fine-grained control of performance-critical functions in Arcon or future computing systems that will be based on the message-passing middleware.

3 Strategic Relevance

3.1 PATENTS

The project's focus has so far been to conduct open-access research and open source contributions as well as the reinforcement of state-of-the-art software libraries used internationally such as LLVM-MLIR. Nevertheless, we expect a number of patents to be registered by the end of the project in order to protect intellectual property of European and Swedish ICT, e.g., in the context of the Logical Clocks or other spin-off companies that will be potentially established by the end of the project.

3.2 SPIN-OFF COMPANIES

Logical Clocks[3]**:** Seif Haridi and Salman Niazi are co-founders.

3.3 STRATEGIC SIGNIFICANCE

The CDA project is actively participating in the digitisation and integration of AI technologies in the Swedish ICT. RISE is one of the core drivers of innovation in the Swedish industry today and the CDA project has played a key role so far for RISE in establishing new channels of collaboration. Examples include a new official collaboration between RISE and the *TRATON group* (Scania, Volkswagen) which aims to use, among others, the software developed in CDA in the digital transformation of the automobile and transportation industry. Another collaboration with *King* was also initiated recently solely in terms of providing the mobile gaming industry quick access to the tools and systems (Arcon, Kompact) we have been developing in CDA. Finally, a series of workshops between RISE and Ericsson and a formal discussion has been initiated recently with the goal to help the telecom industry shift its focus towards automated and ML-driven services with CDA being considered as a promising new software to exploit at large scale across edge and cloud networks within the next 10 years.

The CDA project has fuelled ideas and new industrial collaborations within the KTH Digital Futures and CASTOR[4] networks. The latter is a newly established software research centre and professional network which aims to bring cutting-edge software and newly researched techniques to the Swedish industry. Our joint work with Swedbank on anomaly detection using continuous deep analytics was one of the latest results of that collaboration.

[3] `https://logicalclocks.com`

[4] `https://www.castor.kth.se/project/continuous-deep-analytics/`

4 The Graduate Training of the Project

The CDA project helped to initiate the formation of a large research group across RISE and KTH that included different activities such as third-cycle courses, weekly discussion groups, training hackathons, invited talks, student competitions and workshops. Alongside senior researchers, research engineers, graduate and undergraduate students, the CDA group aimed the establishment of deep knowledge in the latest advances in computer systems, machine learning and data management through the following activities:

Figure 7: MOOC Recording (Paris Carbone) at TUDelft.

- Intensive Course (3rd cycle) on the “Fundamentals of Database Systems” (P1-P4 2020).
- Weekly Research Paper Seminar & Discussion Series[5].
- Contributed Seminars and Design of KTH systems courses: ID2203 and ID2220.
- Student Hackathons: IR Dataflow Abstractions (November 2019) and Benchmarking Streaming Runtimes (February 2020).
- Ph.D. Completion/Defence of Paris Carbone, Salman Niazi, Theodore Vasiloudis and Lars Kroll.
- Group Research Visit to Imperial College London for Joint Workshop (Dec. 2018)
- Joint MOOC[6] implemented with TUDelft on “Taming Big Data Streams” with Paris Carbone from CDA contributing all video seminars and content on Stream State Management and Reliability (Figure 7).

[5] https://docs.google.com/document/d/1we83ntrvdhXZG1oXpWcFRBOYLxjpVFc5uRdKmX3nwS8/edit?usp=sharing
[6] https://online-learning.tudelft.nl/courses/taming-big-data-streams-real-time-data-processing-at-scale/

5 Collaborations

5.1 INTERNAL COLLABORATION

The CDA group helped creating a cluster of expertise across four distinct groups between KTH and RISE (ML@KTH led by Henrik Boström, Systems@KTH led by Seif Haridi and Christian Schulte, ML@RISE led by Daniel Gillblad and Systems@RISE led by Paris Carbone). Collaborations within CDA have been fruitful in the following ways:

- Helped speeding up knowledge transfer within joint discussions in weekly meetings
- Joint supervision of M.Sc. and Ph.D. theses
- Joint publications between RISE and KTH (e.g., work in Block-distributed Gradient Boosted Trees with Theodore Vasiloudis and Henrik Boström)
- Joint RISE+KTH workshops (e.g., PLDS, DISCAN)

5.2 CROSS-DISCIPLINE COLLABORATION

The fusion of ML and Systems experts across KTH and RISE created a new perspective and a series of research collaborations in the new trending field of Systems for ML (or ML for Systems). This would not have been possible without a close interaction achieved in the CDA project. Apart from knowledge transfer between the two groups, significant results were also achieved through joint supervision (e.g., Dr. Theodore Vasiloudis' Ph.D. work was supervised by Seif Haridi and Henrik Boström, who led the work in systems and ML respectively). Furthermore, several upcoming publications in conformal prediction systems and uncertainty estimation in Arcon will also be the result of joint research between Systems and ML researchers in the project.

5.3 INTERNATIONAL COLLABORATION

Several of the results of the project would not have been possible without the feedback and close collaboration with other research institutes internationally. Below we highlight a few collaborations that were established or reinforced within the scope of the CDA project.

Stanford University: Our work on intermediate representation (IR) support for batch and stream analytics began as a collaboration with Stanford University and Matei Zaharia's group (creator of Apache Spark). Via SSF's support, our researcher Lars Kroll from KTH spent two weeks at Matei's research group understanding and extending Weld, Stanford's novel IR for optimising data pipelines with stream capabilities. This later led to Arc, the CDA IR which is today part of the MLIR-LLVM ecosystem.
TU Delft: Several of the goals of CDA align with the vision of Web Information Systems (WIS) research group at TUDelft. The expertise at TUDelft has complemented ours with use cases and ideas in problems related to data management. Several results of our collaboration with Asterios Katsifodimos' group include a joint MOOC, a newly published survey (on Stream Processing Systems) and a Tutorial as SIGMOD 2020 on steam processing technology while a joint paper is also currently under submission on iterative dataflow execution.
Boston University: Our continued collaboration with Vasiliki Kalavri (currently Assistant Professor at BU) has led to significant results in the intersection of stream technology and graph analytics. Via a series of joint MSc projects co-supervised by our two groups we have managed to invent novel techniques for graph analytics including partitioning techniques (published at VLDB 2018) and graph embeddings (two papers under submission) which also serve today as core showcases of the Arcon system and its capabilities.
Imperial College London: The Large Scale Data and Systems group at ICL (Peter Pietzuch, Holger Pirk) has a long history and expertise in systems for hardware accelerated analytics. From the early beginnings we co-organised joint workshops and established new collaborations between PhD students of which many results have come to materialise today. A core example of this collaboration is the design of a low level representation for incremental analytics for which we also plan to get EU (ERC) funding in the upcoming months along joint publications.

TUBerlin: Our collaboration history with TUBerlin (DIMA group led by Volker Markl) has its roots in our joint efforts in the Stratosphere and Flink system (SSF E2E project) as well as EU projects and EIT programs that we jointly participated in. To this day we still maintain a close collaboration, especially in the prospect of combining our efforts between the Nebula Streams project (IoT streaming at TUBerlin) and CDA via the use of our work on IRs and code generation in the scope of sensor networks. In the latest results from the DIMA group (SIGMOD 2020) it is already mentioned that the Arc IR will be a potential candidate in the future to boost performance in the Nebula Streams platform.
University of Edinburgh: Our showcase application on online graph embeddings is done in collaboration with PhD student Massimo Perini from UoE under the co-supervision of Milos Nicolic's group that focuses on algorithms for stream analytics and ML. This work is planned to be submitted and published in 2021. In addition, our upcoming work on iterative dataflow models is also done under the co-supervision of Pramod Bhatotia from UoE.
Heriot-Watt University: Finally, we have been working with Artem Shinkarov from Heriot-Watt on integrating type-inference for array (tensor) programming into Arc and Arc-Script. This work should allow us to optimise and generate efficient implementations for algorithms over n-dimensional datasets, which are common, in particular, in scientific computing.

5.4 INDUSTRIAL COLLABORATION

As mentioned in Section 3.3 we have disseminated our efforts in CDA to CASTOR and Digital Futures. This, in combination with strategic partnerships at RISE helped us reach data engineers and analysts in the industry and get exposed to real needs early in the project. As of today, there have been numerous industrial collaborations in the form of M.Sc. thesis co-supervision for the following projects:

- Reliable External State Management for Dataflow Backends – Co-supervised by **Logical Clocks** and co-funded by **Google Research** (Research Cloud Grant) – Hasseb Asif
- Externalising Dataflow State to NoSQL Databases – Co-supervised by **Logical Clocks** and co-funded by **Google Research** (Research Cloud Grant) – Sruthi Kumar
- Real-time Financial Anomaly Detection – Co-supervised by **Swedbank** – Anna Martignano
- Window Aggregation Algorithm Repository[7] - A cooperation with **IBM (Watson) Research**.

Planned Collaborations: A set of new Ph.D. theses are planned to be co-supervised and co-funded by **King** in order to improve complex data pipelines used for analysing social network and player activity data at King as well as their recommender system. Furthermore, two upcoming strategic collaborations between RISE and **Ericsson** and **TRATON group** are expected to lead to applications for CDA in the next half of the project. Despite the high interest to put Arcon and Kompact into use, the details of the collaboration are still under discussion at the time of writing.

[7] https://github.com/IBM/sliding-window-aggregators

6 Work Continuation

A description of the expected organisation of the activities within the project after the SSF funding expires. Which parts of the project do you consider your most valuable contributions to the total research system in Sweden? There are two types of results in the CDA project, 1) Prototypes and Methods and 2) Production-Ready Open-Source Systems. For the latter, we foresee the continuation of further development for at least three to five years after the CDA project ends, given the potential for applying for incubation, e.g., in the Apache Foundation as well as commercialisation. More concretely we plan to continue the development of the Kompact and further extend its capabilities towards edge computing and sensor networks as well as building higher-level services such as decentralised serverless capabilities. Arcon is already the first system of its kind when it comes to hybrid batch and stream analytics and it is expected to serve hundreds of industrial use cases in data analytics that aim for high performance. Thus, due to the development demands we plan to build a user group and an active development community that can take over our efforts and expand Arcon's ecosystem such as new connectors to external systems and message queues as well as code generators for new hardware. In terms of research, we have already started applying for new grants that can help us expand and extend over time our research and development efforts in Arcon and Kompact beyond the expiration of the CDA project, at least until 2025.

7 Budget of the Project

A total budget of 32 010 000 SEK has been reserved for the research and industrial exploitation of the project for its duration from 2017 until 2022. The following table summarises the budget consumed until 31/12/2019 which corresponds to nearly the first half of the project. The budget increases per year are due to new hires and senior-member promotions in the group. For example in 2018 we had one additional Ph.D. hire and one promotion, while in 2019 we had two graduations and one more Ph.D. hire.

Year	Budget Consumed
2017	1 942 121 SEK
2018	5 583 187 SEK
2019	4 418 319 SEK
Total	11 943 627 SEK

We reserved twice the estimated budget for the second half of the project (20 066 373 SEK), since according to our previous experience most research result exploitation and critical development efforts are typically required towards the end of the project, close to its maturity. To that end, we plan to expand further our research group by two additional graduate level Ph.D. researchers and one postdoctoral researcher by 2021.

Year	Estimated Budget
2020	6 100 000 SEK
2021	7 100 000 SEK
2022	6 800 000 SEK
Total	20 000 000 SEK

8 External Information and Activities

Alongside conference presentations we have also focused major efforts into communicating the benefits that the CDA project will bring to society through a series of workshops, tutorials and technical seminars in popular research and industrial venues. Below we summarised most of our efforts in more detail.

Organised Workshops

So far, the following two workshops were organised within the CDA project with several influential presenters and keynote speakers invited.

Workshop in Programming Languages and Distributed Systems (Mar. 2020)
Webpage: `https://plds.github.io/index.html`

Workshop in Distributed Computing & Analytics (Sep. 2018)
Webpage: `https://discan18.github.io/`

Organised Tutorials

SIGMOD Tutorial on Stream Processing Systems
`https://streaming-research.github.io/Tutorial-SIGMOD-2020/`

Tech Talks and Events

- Sruthi Kumar & Hasseb Asif – FlinkNDB: Skyrocketing Stateful Capabilities of Apache Flink, **Flink Forward Global Event** (Oct. 2020)
- Sruthi Kumar & Hasseb Asif – NEXMark-Beam: Your best companion for testing and benchmarking new core stream processing libraries, **Beam Summit** (Aug. 2020)
- Frej Drejhammar – Extending Clang and LLVM for Interpreter Profiling Perf-ection, **Euro-LLVM** (Apr. 2020)
- Klas Segeljakt – Euro-LLVM Experiences using LLVM to implement a custom language, **Euro-LLVM** (Apr. 2020)
- Klas Segeljakt – Arc: An MLIR dialect for Data Analytics, **PLDS** (Mar. 2020)
- Paris Carbone – Seamless Batch and Stream Computation on Heterogeneous Hardware with Arcon, **PLDS** (Mar. 2020)
- Paris Carbone – Reliable Stream Processing at Scale, **Chaos Engineering Workshop Stockholm** (Dec. 2019)
- Paris Carbone – Continuous Intelligence: Intersecting Event-Based Business Logic and ML, **Nordic Data Science Summit NDSML** (Oct. 2019)
- Paris Carbone – Continuous Intelligence through Computation Sharing, **CASTOR Software Days** (Oct. 2019)
- Massimo Perini – Deep Stream Dynamic Graph Analytics with Grapharis, **Flink Forward Berlin** 2019
- Klas Segeljakt, Max Meldrum – Introducing Arc: An IR for unified batch and streaming, **Flink Forward Berlin** (Oct. 2019)
- Paris Carbone – Stream Loops on Flink, **Flink Forward Berlin** (Nov. 2018)
- Paris Carbone – Asynchronous Epoch Commits for Fast and Consistent Stateful Streaming with Apache Flink, **LADIS**/**PODC** (July 2018).
- Paris Carbone – The Road to Continuous Deep Analytics, **RISE Open House** (Apr. 2018)

Project Web Material

Project Website: `https://cda-group.github.io/papers.html`
CDA Slack Group: `https://cda-kth-sics.slack.com`
Project Mail-list: `https://groups.google.com/g/cda-project`
Github Group: `https://github.com/cda-group`

9 SWOT Analysis

Strengths

- CDA itself defines a new and distinct problem domain for systems and ML alike with high significance and interesting challenges alike.
- Software design approach in CDA is decoupled to hardware and relevant advances which can be incorporated to the project fast.

Weaknesses

- Supporting $100\,\%$ existing functionality already present in the ecosystem of big data technologies in addition to CDA's novelties might add a strong development burden in the long-term.
- Research in "Systems for ML" is criticised by both ML and systems communities for its identity and that increases the challenge of publishing work in the right venue.

Opportunities

- New open source initiatives in compiler technology such as MLIR helps reduce the development overhead and boost usability of results in the project to a great extent.
- Increasing awareness of the limitations of "black box" models and transformers such as GPT-3 will fuel interest for reliable analytics which CDA aims to provide ahead of time.

Threats

- The continuous dependence on cloud-provided ML services could shift public focus from novel in-house data management such as CDA to general models under the centralised control of big powers in cloud computing such as Amazon, Google, OpenAI etc.
- It is challenging to keep talent and critical mass in Sweden given the potential opportunities and offers that are constantly given by big corporations in Silicon Valley and China.