Research Statement

Future Scalable and Sustainable Data-Intensive Systems

Bo Zhao
bo.zhao@aalto.fi

I conduct research on efficient ***data-intensive systems*** that translate ***data*** into ***value*** for decision making. The scope of my research spans across multiple subfields, from scalable reinforcement learning (RL) systems to distributed data stream management systems, as well as compilation-based optimisation techniques. My long-term goal is to explore and understand the fundamental connections between data management and modern machine learning (ML) systems to make decision-making more transparent, robust and efficient.

As data is collected at unprecedented rates for timely analysis, the *model-centric* paradigm of ML is shifting towards a ***data-centric*** and ***system-centric*** paradigm. The recent breakthroughs in large ML models (*e.g.* GPT 4, PaLM, Gato, Metaformer and ChatGPT) and the remarkable outcomes of RL in complex real-world settings (*e.g.* AlphaGo, AlphaStar, AlphaFold and AlphaCode) have shown that ***scalable*** data and knowledge management is critical to obtain state-of-the-art performance for complex tasks such as information search, image processing, text understanding, health care and robotics.

My research is based on formulating research problems, building real systems and a strong analytical mechanism of performance models to attack key bottlenecks in the ML ifecycle. In principle, I strive towards high impact research, often in collaboration with other researchers from leading industry labs and academia. To provide further insights, I will elaborate on my recent and ongoing research, followed by future plans (detailed future plans are available on request).

1 Recent and Ongoing Research

I have led and contributed to multiple projects of large-scale data-intensive systems at Queen Mary University of London, Imperial College London, Humboldt-Universität zu Berlin, and Amazon Web Services. My focus is on optimisation of three layers for the ML lifecycle, including the scalable ML layer [1, 2], the data stream management layer [3, 4, 5] to integrate ML input pipelines and external domain knowledge, and the low-level compilation-based optimisation layer [6, 7, 8] for resource-aware ML, as well as their applications in real-world settings [9, 10].

1.1 Dataflow-Oriented Scalable and Energy-Efficient Reinforcement Learning Systems [ATC'23]

Reinforcement learning solves decision-making problems in which an agent continuously learns to act in an unknown environment. Training a large number of agents is resource-intensive and must scale to large GPU or TPU clusters while achieving energy efficiency. Yet, current distributed RL systems hardcode a single strategy to parallelise and distribute an RL algorithm based on its algorithmic structure and only permit the acceleration of specific parts of the computation (e.g. policy deep neural network updates) on GPU/TPU workers. Fundamentally, existing systems lack abstractions to decouple RL algorithms from their execution.

To tackle this challenge, during my postdoc at Imperial College London and my Assistant Professorship at Queen Mary University of London I have designed a flexible distributed RL system, *MSRL* [1], based on a new abstraction of *fragmented dataflow graphs* which offer flexibility of how RL training is parallelised and distributed. MSRL maps an RL algorithm to parallel computational *fragments*. Fragments are executed on different devices by translating them to low-level intermediate dataflow representations such as computational graphs of ML engines (PyTorch, TensorFlow, MindSpore), CUDA implementations or CPU threads. A *distribution policy* governs how fragments are mapped to devices, without changing algorithm implementations. MSRL subsumes distribution strategies of existing systems, while scaling RL training to many GPU workers (*e.g.* 128 V100 GPUs). The distribution strategies provide a spectrum for trade-off between high performance (*e.g.* mapping fragments to high-end GPUs, A100) and energy efficiency (*e.g.* mapping fragments to Raspberry Pi devices). My work has been integrated into MindSpore, a leading industry ML framework, under the name of *MindSpore Reinforcement* [2].

1.2 Efficient Stateful Data Stream Management and Knowledge Integration for ML Systems [ICDE'18, PES-GM'19, ICDE'20, SIGMOD'21, Applied Energy'22]

Data stream processing is critical for ***real-time intelligence***. It enables complex *analytical queries* and *ML lifecycle* with *low-latency*. On the one hand, distributed ML training requires efficient data stream management that ingests input data into the ML pipeline [11], computes and communicates updates across different partitions of a large ML model, and detects concept drifts for continual learning. On the other hand, integrating human knowledge into ML models significantly reduces the required training data and makes ML training more efficient, robust, explainable, and trustworthy. To this end, domain experts usually express the high-level knowledge as complex analytical queries (*e.g.* rules-based knowledge representation, pattern detection and mining) over data streams from different sources and applications (*e.g.* Internet of things). Such processing is *stateful* and therefore, the system needs to maintain partially-computed results, which grow exponentially in the number of processed data items. High input rates of streams amplify this issue, making low-latency data analysis challenging. To further complicate matters, stream engines need to fetch data from remote sources (*e.g.* integrating external domain-specific knowledge, or data privacy regulations such as GDPR[1]) which increases the data transmission latency by orders of magnitude and therefore, deteriorates overall performance.

Throughout my PhD at Humboldt-Universität zu Berlin, I have addressed this challenge for analytical queries (ML training or inference is incorporated as query operators) using optimised state management techniques. First, I have designed the *AthenaCEP* framework [3, 4] for best-effort query evaluation by *hybrid load shedding* that discards both input events and partial results based on a cost model. AthenaCEP carefully selects the candidates to drop in order to satisfy a latency bound while striving for a minimal loss in result quality. Second, to efficiently integrate remote data and external knowledge in ML, I have built the *EIRES* framework [5], which decouples the fetching of remote data from its actual use in query evaluation by *caching*, *prefetching* and *lazy evaluation* techniques. A cost model is proposed to determine when to fetch which remote data items and how long to retain them in the local cache. These frameworks, together with RL, have been applied to real-world settings of optimised smart grid management [9, 10], public transportation monitoring [3], and bushfire detection using satellite images [5]. Since their publication, several top-tier conference papers have used these frameworks as baselines (VLDB'21 [12], VLDB'22 [13]), and as the stream engine to detect rumours from massive streaming data (VLDBJ'22 [14]).

1.3 Compilation-Based Optimisation for Resource-Aware ML and Data Management Systems [FGCS'23, COSMIC'15, ICA3PP'15]

Recently, we have been witnessing ML models and analytical queries being compiled for efficient execution on heterogeneous hardware accelerators (*e.g.* TPU, GPU and FPGA), or devices with restricted computing power and limited energy supply (*e.g.* battery-powered edge devices). For the ML side, automated compilers (*e.g.* JAX, XLA and TVM) have been proposed for general deep learning workloads. Yet, they are not designed for the complex control and data flow of RL algorithms. For analytical queries, compilation-based query optimisation approaches have been studied for database systems over static data. However, few of them target dynamic data streams and efficient knowledge extraction.

I have conducted research on automatic parallel code generation using LLVM. Specifically, I proposed a framework that integrates data-dependence analysis, parallelism extraction and code transformation [6, 7, 8]. These techniques have been incorporated in MSRL [1] and EIRES [5] for efficient parallel execution. The representation of *fragments* in MSRL [1] are generated from dataflow analysis and parallelism extraction of the algorithm implementation. The SQL-like stream queries of EIRES [5] are compiled into intermediate representation to reuse the shared

[1] https://gdpr-info.eu/

subqueries and query predicates, which are generated as C++ code for efficient execution. To target the system's performance bottlenecks at runtime, I have designed a light-weight profiler that monitors resource utilisation of the distributed cloud-based data warehouse, Amazon Redshift [2], in real time.

2 Future Research

My vision is to build fully automated *holistic* data-intensive systems that integrate the ML layer, the data management layer, and the compilation-based optimisation layer. That is to answer the question ***"how to co-design multiple layers of the software stack to improve scalability, performance, and energy efficiency of machine learning systems"***. This requires end-to-end optimisation from high-level semantics (*e.g.* data constraints, domain knowledge, data sketching and approximation) to low-level optimisation techniques (*e.g.* code generation, memory management and task scheduling).

I am more than happy to discuss and share my future research plan. Please contact me if you are interested. Thanks.

References

[1] H. Zhu*, **B. Zhao*** , G. Chen, W. Chen, Y. Chen, L. Shi, Y. Yang, P .Pietzuch, L. Chen (*equal contribution): "MSRL: Distributed reinforcement learning with dataflow fragments," in *ATC 2023*. [Online]. Available: https://zbjob.github.io/ATC23.pdf

[2] Mindspore reinforcement. [Online]. Available: https://github.com/mindspore-lab/mindrl

[3] **B. Zhao**, N. Q. V. Hung, and M. Weidlich, "Load shedding for complex event processing: Input-based and state-based techniques," in *ICDE 2020*. [Online]. Available: https://doi.org/10.1109/ICDE48307.2020.00099

[4] **B. Zhao**, "Complex event processing under constrained resources by state-based load shedding," in *ICDE 2018*. [Online]. Available: https://doi.org/10.1109/ICDE.2018.00218

[5] **B. Zhao**, H. van der Aa, T. T. Nguyen, Q. V. H. Nguyen, and M. Weidlich, "EIRES: efficient integration of remote data in event stream processing," in *SIGMOD 2021*. [Online]. Available: https://doi.org/10.1145/3448016.3457304

[6] **B. Zhao**, Z. Li, A. Jannesari, F. Wolf, and W. Wu, "Dependence-based code transformation for coarse-grained parallelism," in *International Workshop on Code Optimisation for Multi and Many Cores, COSMIC@CGO 2015*. [Online]. Available: https://doi.org/10.1145/2723772.2723777

[7] Z. Li, **B. Zhao**, A. Jannesari, and F. Wolf, "Beyond data parallelism: Identifying parallel tasks in sequential programs," in *ICA3PP 2015*. [Online]. Available: https://doi.org/10.1007/978-3-319-27140-8_39

[8] S. Liu, X. Wan, Z. Zhang, **B. Zhao**, W. Wu, "TurboStencil: You Only Compute Once for Stencil Computation," In *Future Generation Computer Systems*, 2023. Online]. Available: https://doi.org/10.1016/j.future.2023.04.019

[9] G. Raman, **B. Zhao**, J. C.-H. Peng, and M. Weidlich, "Adaptive incentive-based demand response with distributed non-compliance assessment," in *Applied Energy 2022*. [Online]. Available: https://doi.org/10.1016/j.apenergy.2022.119998

[10] G. Raman, J. C.-H. Peng, **B. Zhao**, and M. Weidlich, "Dynamic decision making for demand response through adaptive event stream monitoring," in *IEEE Power Energy Society General Meeting (PESGM) 2019*. [Online]. Available: https://doi.org/10.1109/PESGM40551.2019.8974095

[11] D. G. Murray, J. Simsa, A. Klimovic, and I. Indyk, "tf.data: A machine learning data processing framework," *VLDB*, 2021. [Online]. Available: http://www.vldb.org/pvldb/vol14/p2945-klimovic.pdf

[12] K. Chapnik, I. Kolchinsky, and A. Schuster, "DARLING: data-aware load shedding in complex event processing systems," *VLDB*, 2021. [Online]. Available: http://www.vldb.org/pvldb/vol15/p541-chapnik.pdf

[13] M. Bucchi, A. Grez, A. Quintana, C. Riveros, and S. Vansummeren, "CORE: a Complex Event Recognition Engine," *VLDB*, 2022. [Online]. Available: http://www.vldb.org/pvldb/vol15/p1951-riveros.pdf

[2] https://aws.amazon.com/redshift/

[14] TT. Nguyen, TT. Huynh, H. Yin, M. Weidlich, TT. Nguyen, TS. Mai, and QVH. Nguyen, “Detecting rumours with latency guarantees using massive streaming data,” *The VLDB Journal*, 2022. [Online]. Available: http://doi.org/10.1007/s00778-022-00750-4