Bo Zhao

Rudower Chaussee 25, Institut für Informatik, Humboldt-Universität zu Berlin
12489, Berlin – Germany
hu-berlin.de/bo_zhao
Phone: +49 (0) 30 2093 3070 | Email: bo.zhao@hu-berlin.de

Research Interest

My research interests include **complex event processing**, **data stream management system**, **large-scale distributed data processing**, **real-time data analysis** and **code optimization**.

Education

Humboldt-Universität zu Berlin — **Berlin, Germany**
PhD candindate in Computer Science — *02/2016–05/2021(expected)*
Topic: State Management for Efficient Complex Event Processing

Xi'an Jiaotong University — **Xi'an, China**
M.S. in Computer Science, GPA: 3.78/4 — *09/2012–07/2015*
Ranking: Top 1%, 2nd among 150 students

Wuhan Institute of Technology — **Wuhan, China**
B.S. in Network Engineering, Major GPA: 91.06/100 — *09/2008–07/2012*
Ranking: Top 1%, 2nd among 500 students

Visiting Research

University of Queensland — **Brisbane, Australia**
Visiting Researcher in Computer Science — *05/2017–06/2017*
Data Stream Processing

RWTH-AACHEN University — **Aachen, Germany**
Visiting M.S. Student in Computer Science, High Performance Computing — *09/2013–02/2015*

Honors & Awards

2017–2018: IEEE ICDE Student Travel Grant

2015–2016: EDBT Summer School Travel Grant

2014–2015: Outstanding Graduate, ACM SIGPLAN Travel Grant, ACM SIGMICRO Travel Grant

2012–2013: China National Scholarship(top 0.2%), Creative-Master Scholarship, Excellent Master Student

2011–2012: Excellent Graduation Thesis, Outstanding Graduate

2010–2011: China National Scholarship(top 0.2%), Top Grade Scholarship, Pacemaker to Merit Student, Advanced Individual in Social Practice

2009–2010: China National Scholarship(top 0.2%), Top Grade Scholarship, Pacemaker to Merit Student

2008–2009: Top Grade Scholarship, Pacemaker to Merit Student, Outstanding League Member

Interships & Jobs

06/2019–09/2019: Software Development Engineer Intern at Amazon, AWS Redshift, Berlin, Germany

02/2016–present : Research Assistant in Humboldt-Universität zu Berlin, Berlin, Germany

09/2015–12/2015: Research Assistant in Technische Universität Darmstadt, Darmstadt, Germany

10/2013–02/2015: Student assistant in German Research School for Simulation Sciences, Aachen, Germany

06/2010–09/2010: Tarena technology company, Wuhan, China

06/2009–09/2009: China United Network Communications Group Co.Ltd, Wuhan, China

Publications

Under review: 1×SIGMOD'21, 1×IEEE Transactions on Power Systems.

- Bo Zhao, Nguyen Quoc Viet Hung, Matthias Weidlich: **Load Shedding for Complex Event Processing: Input-based and State-based Techniques**, *In Proc. of the 36th IEEE International Conference on Data Engineering (ICDE), Dallas, TX, USA, IEEE, April 2020*

- Gururaghav Raman, Jimmy Chih-Hsien Peng, Bo Zhao, Matthias Weidlich: **Dynamic Decision Making for Demand Response through Adaptive Event Stream Monitoring**, *In Proc. of the IEEE Power & Energy Society General Meeting (PESGM), Atlanta, GA, USA. IEEE, August 2019.*

- Bo Zhao: **Complex Event Processing under Constrained Resources by State-based Load Shedding**, *In Proc. of the 34th IEEE International Conference on Data Engineering (ICDE), Paris, France, IEEE, April 2018*

- Bo Zhao, Zhen Li, Ali Jannesari, Felix Wolf, Weiguo Wu: **Dependence-Based Code Transformation for Coarse-Grained Parallelism**, *In Proc. of the International Workshop on Code Optimisation for Multi and Many Cores (COSMIC'15) held in conjunction with CGO 2015, San Francisco Bay Area, CA, USA, ACM, February 2015*

- Bo Zhao, Ali Jannesari: **Dependence-Based Parallel Code Generation Using Intel CnC**, *In Proc. of the 24th International Conference on Parallel Architectures and Compilation Techniques (PACT'15),San Francisco Bay Area, CA, USA, October 2015* (ACM SRC poster)

- Zhen Li, Bo Zhao, Ali Jannesari, Felix Wolf: **Beyond Data Parallelism: Identifying Parallel Tasks in Sequential Programs**, *In Proc. of the 15th International Conference on Algorithms and Architectures for Parallel Processing (ICA3PP'15), Springer, November 2015*

- Song Liu, Bo Zhao, Qing Jiang, Weiguo Wu: **A Semi-Automatic Coarse-Grained Parallelization Approach for Loop Optimization And Irregular Code Sections**(in Chinese). *Chinese Journal of Computers, 2017*

- Song Liu, Weiguo Wu, Bo Zhao, Qing Jiang: **Loop Tiling for Optimization of Locality and Parallelism**(in Chinese). *Journal of Computer Research and Development, 2015*

Talks

April 2020: The 36th IEEE International Conference on Data Engineering (*ICDE'20*), Dallas, TX, USA;

April 2018: The 34th IEEE International Conference on Data Engineering (*ICDE'18*), Paris, France;

September 2015: The Seventh Annual Concurrent Collections Workshop (*collocated with LCPC'15*), Raleigh, NC, USA;

September 2015: The 44th International Conference on Parallel Processing (*ICPP'15*), Beijing, China;

February 2015: The 2nd International Workshop on Code Optimisation for Multi and Many Cores (*COSMIC'15*), San Francisco Bay Area, CA, USA;

September 2014: The Sixth Annual Concurrent Collections Workshop (*collocated with LCPC'14*), Intel Corporation Hillsboro, Oregon, USA; Programming Contest: Parsec Apps

Projects

State Manangement for Efficient Complex Event Processing **Humboldt-Universität zu Berlin**

02/2016–present

This is my ongoing PhD project. I focus on optimized state management for complex event processing (CEP). CEP evaluates queries over streams of events for real-time detection of user-specified patterns which correlate event data within certain time windows. Some complex queries detect sequence patterns or perform sophisticated mathematical algorithms on top of value predicates. Such queries are stateful and therefore, the CEP engine needs to maintain a set of partial results. When combined with kleene closure operators, the size of partial results grows exponentially in the number of processed events. High input rates of streams amplify such a problem. This makes low latency data analysis challenging. What's worse, when integrating with remote data sources, CEP's performance deteriorates even further by data transmission latency. Massive parallelization is not trivial because of irregular data dependences among partial results and events. To tackle this problem, I propose optimizations of load shedding and efficient remote data fetching and caching techniques. To this end, I built a prototype CEP engine in C++ around 7K lines of code and implemented the load shedding optimization and remote data fetching and caching techniques. For development efficiency, I use Python library for offline machine learning such as training classifiers for load shedding decisions. Whereas, online classifiers are implemented in C++ for low latency analysis. Initial results have been published in ICDE'18 PhD Symposium. More detailed evaluations have been published in ICDE'20. The source code is publicly available (`https://github.com/zbjob/AthenaCEP`). The other work of remote data fetching and caching optimizations are under review for SIGMOD'21. I also cooperated with the Department of Electronic Engineering, National University of Singapore, to apply our complex event processing optimizations to smart grid management using Esper stream engine(`https://github.com/zbjob/SmartGrid`). The result is published in a IEEE PES-GM'19 conference paper. More comprehensive evaluations are presented in a journal article under review for IEEE Transactions on Power Systems.

Lightweight Profiling for AWS Redsfhit Queries **Amazon Web Services**

06/2019–09/2019

I was a software development intern at AWS Redshift team. During this intership, I developed a light-weight performance analysis tool to trace and analyze detailed runtime query execution behaviours of the cloud-based distributed database, Redshift, deployed on AWS. The tool is able to target the performance bottleneck and pave the way for further improvements. However, profiling at extremely fine granular incurs unacceptably high computational overhead. My goal was is close this gap. The performance analysis tool is based on eBPF (extended Berkeley Packet Filter) and its front end BCC (BPF Compiler Collection). In order to reduce the computational overhead, the eBPF code monitors statistics in kernel space and store them to BPF maps. In user space, I fetch the BPF maps via BCC APIs (through Python scripts) and perform sophisticated analysis asynchronously. To further reduce the computational overhead, I employ the approximation technique, sampling, to monitor execution time for different code sections. I integrate this profiler into the code generator of the query plan. Therefore, the probes have been automatically inserted to the generated C++ code from SQL. When the queries are executed, the profiling is automatically done. I evaluated the performance analysis tool against TPC-DS benchmark at 3TB and 10TB scale on AWS 5-node DC2.8xlarge instances clusters. The tool is able to reduce the computational overhead to 1.0% and obtain accurate profiling information. I also found some insights for performance improvements on real-world business workloads in AWS. For instance, one specific query of a Redshift customer has been improved by 1.75 $\times$ faster.

Auto-Parallelization on Multicores **RWTH-Aachen University**

10/2013–02/2015

I was a member of the Multicore Programming Research group in German Research School for Simulation Sciences, RWTH-AACHEN University, where I did my master thesis. I proposed and developed a novel auto-parallelization framework, which integrates data-dependence profiling, coarse-grained task parallelism extraction and source-to-source transformation. Our group identified sections of code called computational units (CUs), which follow a read-compute-write pattern and can be used as building blocks for forming parallel tasks. First, I used our own profiling tool to analyze the data and control dependence among the source code sections and then generated the dependence graph of CUs. Then I merged CUs in the CU graph and generated a more coarse-grained task graph. Second, I used LLVM front-end Clang to parse the source code, extract the AST, traverse the AST to locate DOALL loops and the code sections targeted by the task graph and then transformed the sequential source code to parallel Intel Threading Building Blocks (TBB) code and Intel Concurrent Collections (CnC) code. My work has been published in two conference papers. Project link: `https://www.informatik.tu-darmstadt.de/parallel/research_3/discopop.en.jsp`

Loop Optimization on Multicore System **Xi'an Jiaotong University**

09/2012–06/2015

This project was supported by the National Natural Science Foundation of China. A PhD student and I mainly focused on parallel loop optimizations on shared memory architectures, especially NUMA machines. We put forward a loop tiling strategy for imperfectly-nested loop nests and leveraged machine learning technologies to select the appropriate tile size to keep the cache hot. We successfully exploited fine-grained parallelism and data locality and tested our approach on several benchmarks with speedups up to 200 times. We published our results in two journal papers.

MCU Smart Model Car **Wuhan Institute of Technology**

09/2010–06/2011

This team consisted of a mechanical group, a hardware group and a software group. These groups respectively designed structural models, MCU motherboard and other hardware and embedded software control systems. The model car was controlled through sensors and the embedded software system. As a member of the software team, I was responsible for the designing, implementing and optimizing the steer control code and analyzing the data collected by the sensors. We cooperated to design and implement the whole model car and participated in the national competitions.

Languages

English : Fluent

German: Beginner

Chinese: Native speaker

References

Name	Affiliation	Email address
Prof. Matthias Weidlich	Humboldt-Universität zu Berlin	matthias.weidlich@hu-berlin.de
Dr. Nguyen Quoc Viet Hung	Griffith University	quocviethung.nguyen@griffith.edu.au
Prof. Felix Wolf	Technische Universität Darmstadt	wolf@cs.tu-darmstadt.de
Dr. Ali Jannesari	University of California, Berkeley	jannesari@eecs.berkeley.edu
Dr. Jimmy Chih-Hsien PENG	National University of Singapore	jpeng@nus.edu.sg