Jinfeng Li
Computer Science and Engineering
The Chinese University of Hong Kong

Office: 116, Ho Sin Hang Engineering Building, The Chinese University of Hong Kong
Contact: jfli@cse.cuhk.edu.hk

Education Background

- **2014.9 – 2018.7** **Doctor of Philosophy (PH.D.)**
  Department of Computer Science and Engineering, The Chinese University of Hong Kong
  Supervisor: **James Cheng**
  Thesis committee members: **John C.S. Lui** and **Patrick P. C. Lee**

- **2010.9-2014.6** **Bachelor of Engineering (B.Eng.)**
  Faculty of Computer Science, Sun Yat-sen University
  GPA: 89.7/100 (ranking 2/212)
  Major GPA: 90.7/100 (ranking 1/212)

Research Interests

Distributed computing, Machine learning, Large-scale similarity search, Graph analysis

Publications

- **Jinfeng Li**, Xiao Yan, Jian Zhang, An Xu, James Cheng, Jie Liu, Kelvin Ng, and Ti-Chung Cheng.
  A General and Efficient Querying Method for Learning to Hash.
  To appear in Proceedings of the 37th ACM SIGMOD International Conference on Management of Data, 2018.
  (SIGMOD 2018)
  Homepage: **http://www.cse.cuhk.edu.hk/systems/hash/gqr/index.html**

- Yuzhen Huang, Tatiana Jin, Yidi Wu, Zhenkun Cai, Xiao Yan, Fan Yang, **Jinfeng Li**, Yuying Guo, James Cheng.
  FlexPS: Flexible Parallelism Control in Parameter Server Architecture.
  To appear in Proceedings of the VLDB Endowment, 2018.
  (VLDB 2018)

- **Jinfeng Li**, James Cheng, Fan Yang, Yuzhen Huang, Yunjian Zhao, Xiao Yan, Ruihao Zhao.
  LoSHa: A General Framework for Scalable Locality Sensitive Hashing.
  In Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval, 2017.
  (SIGIR 2017)

- Fan Yang, Fanhua Shang, Yuzhen Huang, James Cheng, **Jinfeng Li**, Yunjian Zhao, Ruihao Zhao.
  LFTF: A Framework for Efficient Tensor Analytics at Scale.
  In Proceedings of the VLDB Endowment, 2017. (PVLDB 2017)

- Fan Yang, Yuzhen Huang, Yunjian Zhao, **Jinfeng Li**, Guanxian Jiang, James Cheng.
  The Best of Both Worlds: Big Data Programming with Both Productivity and Performance.
  In Proceedings of the 36th ACM SIGMOD International Conference on Management of Data, 2017.

(SIGMOD Demo 2017)

- Fan Yang, **Jinfeng Li**, James Cheng**.**
  Husky: Towards a More Efficient and Expressive Distributed Computing Framework.
  In Proceedings of the VLDB Endowment, 2016. (PVLDB 2016)
  Homepage: **http://www.husky-project.com**

- **Jinfeng Li**, James Cheng, Yunjian Zhao, Fan Yang, Yuzhen Huang, Haipeng Chen, Ruihao zhao.
  A Comparison of General-Purpose Distributed Systems for Data Processing.
  In Proceedings of the 2016 IEEE International Conference on Big Data, 2016. (IEEE BigData 2016)

- Huanhuan Wu, Yuzhen Huang, James Cheng, **Jinfeng Li**, Yiping Ke.
  Reachability and Time-Based Path Queries in Temporal Graphs.
  In Proceedings of the 32nd IEEE International Conference on Data Engineering, 2016. (ICDE 2016)

Skills

- **Programming skills**: proficient in C++ and Linux, familiar with Java, Scala, C#, SQL and shell script
- **Tools:** Git, Vim, Gdb, CMake, Maven, Visual Studio, IntelliJ
- **Frameworks**:
  Machine learning: MLlib, GraphLab, Parameter Server (Petuum)
  Graph analysis: PowerGraph, GraphX, Pregel, Giraph, JanusGraph, Titan
  Batch-processing: Spark, Hadoop, Naiad, Flink, Husky
  Stream-processing: Spark Streaming, Naiad, Storm, MillWheel, ElasticSearch
  Storage: MySQL, HDFS, Flume, Kafka

Internships

- **2017.7-2017.8 Senior Development Engineer, Alibaba Cloud, Alibaba Group**
  Supervised by Zhengping Qian (Group Manager, Big-Data Infrastructure Team, Alibaba)

  The team is building an integrated graph platform for both OLTP an OLAP workloads. During my internship, I analyzed different systems (Timely dataflow, Pregel, PowerGraph, GraphX, GraphChi and so on) and contributed ideas to the design choices of the new integrated graph platform that we were developing. I also set up distributed environments and deployed HBase, elasticsearch and JanusGraph (a graph database) for their daily graph storage and analytics. To load Terabyte-scale data to JanusGraph efficiently, I developed a load-balance distributed loader and reduced the loading time from days to hours. Meanwhile, I also designed and developed to be used in JanusGraph to accelerate query processing. My codes have been incorporated into their repository.

- **2014.7-2014.8 Data mining Engineer, Department of Data Science, Alibaba Group**
  Supervised by Wanli Min (Principal Data Scientist, Alibaba)

  I worked in the Department of Data Science, and joined a recommendation project that analyzed the features of users and found potential customers for different products. I developed algorithms on the Taobao MapReduce platform to process billions of user data.

Services

- **Teaching Assistant:** Introduction to Database Systems, Data Structures and Applications,
  Problem Solving by Programming

- **External Reviewer:**
  2017: ICDE, VLDB, BigData
  2016: SIGMOD, KDD, SOCC, ICDM, BigData
  2015: SIGMOD, ICDE, KDD, BigData, DASFAA, WAIM

Honors

- **2014.9 – 2018.8** Postgraduate scholarship, The Chinese University of Hong Kong
- **2016, 2017** Certificate of merit, CSE teaching assistant, The Chinese University of Hong Kong
- **2014** Outstanding graduate, Sun Yat-sen University (first-class honor)
- **2013** National scholarship, China
- **2011, 2012, 2013** First-class scholarship, Sun Yat-sen University (top 5%)
- **2013** First prize, National Challenge Cup Students' Competition, Guangdong District, China

Projects

I initiated or was a key team member of a number of projects in distributed computing, which include ***scalable similarity search***, ***general and efficient distributed computing*** and ***large-scale machine learning and data mining***. All of my research works are made open source.

Scalable similarity search

- **LoSHa: A General Framework for Scalable Locality Sensitive Hashing (SIGIR 2017)**
  Hashing algorithms are widely adopted to index similar items in high dimensional space for efficient nearest neighbor search. LoSHa is a distributed computing framework that reduces the development cost by designing a ***tailor-made, general programming interface*** and achieves ***high efficiency*** by exploring LSH-specific system implementation and optimizations. We evaluated LoSHa and show that the performance can be an order of magnitude faster, while the implementations on LoSHa are even more intuitive and require few lines of code.
  LoSHa is the first general-purpose system for hashing-based similarity search, which serves the purpose of information retrieval, recommendation, object detection and so on. LoSHa is faster than Spark by an order of magnitude, yet requires 10 times less lines of code to implement a hashing algorithm. I was the proposer and also the leader of the LoSHa project.

- **GQR: A General and Efficient Querying Method for Learning to Hash (SIGMOD 2018)**
  As an effective solution to the approximate nearest neighbors (ANN) search problem, learning to hash (L2H) is able to learn similarity-preserving hash functions tailored for a given dataset. However, existing L2H research mainly focuses on improving query performance by learning good hash functions, while Hamming ranking (HR) is used as the default querying method. We showed by analysis and experiments that Hamming distance, the similarity indicator used in HR, is too coarse-grained and thus limits the performance of query processing. We proposed a new fine-grained similarity indicator, ***quantization distance (QD),*** that provides more information about the similarity between a query and the items in a bucket. We then developed two efficient querying methods based on QD, which achieve significantly better query performance than HR. Our methods are general and can work with various L2H algorithms. Our results demonstrate that a simple and elegant querying method can produce performance gain equivalent to advanced and complicated hash function learning algorithms.
  While researchers are developing increasingly complicated learning algorithms for better query efficiency, GQR is proposed to improve the query processing without burdening the training process. GQR accelerates the performance of extensive L2H algorithms (e.g. PCAH, ITQ, ALSH, K-means Hashing, Online Hashing…) by an order of magnitude. We have also implemented distributed GQR with LoSHa. I was the proposer and also the leader of the GQR project.

General and efficient distributed computing

- **Husky: Towards a More Efficient and Expressive Distributed Computing Framework (VLDB 2016)**

  Distributed systems such as Hadoop or Spark are widely deployed for massive data processing. They provide coarse-grained operators such as map and reduce to ease the development of distributed algorithms. However, these operators do not support point-to-point communication between two data objects, making it difficult to develop efficient algorithms for some workloads such as ***graph processing*** and ***machine learning***. Husky supports an ***object-oriented programming interface*** and allows ***fine-grained control on messaging between objects***. With these underlying designs, Husky is able to achieve good expressiveness that can implement various workloads. Besides, Husky is efficient that can achieve even better performance compared with domain-specific systems (e.g. Giraph, GraphLab and Parameter Server).

  Husky offers benefits of both existing general-purpose systems (i.e. Spark, Hadoop, Flink and Naiad) and domain-specific systems (i.e. GraphLab, Giraph, Pregel), and achieves a more general and efficient computing model. Husky is an order of magnitude faster than Spark on iterative workloads such as graph analytics and machine learning. I was one of the proposers of Husky and also gave a demo presentation in the 2017 SIGMOD conference in Chicago. I also supervised a group of master and undergraduate students to develop machine learning packages (e.g. PCA, KMeans, ALS, DBSCAN, TFIDF and etc) for Husky.

Large-scale machine learning and data mining

- **A Comparison of General-Purpose Distributed Systems for Data Processing (IEEE BigData 2016)**

  General-purpose distributed systems for data processing become popular in recent years due to the high demand from industry for big data analytics. In this project, we conducted an extensive performance study on four state-of-the-art general-purpose distributed computing systems. We selected ***Spark, Flink, Naiad*** and ***Husky*** in the comparison and studied their performance on three different types of workloads, i.e. non-iterative bulk workloads (***WordCount*** and ***TeraSort***), iterative graph workloads (***PageRank*** and ***SSSP***) and iterative machine learning workloads (**LR and ALS**). Our results reveal useful insights on the design and implementation, which help the improvement of existing systems and the development of better new systems.

  I was the leader and also the proposer of this project.

- **FlexPS: Flexible Parallelism Control in Parameter Server Architecture**

  Modern distributed machine learning systems such as Parameter Server and Petuum adopt parameter server (PS) architecture to coordinate the access of model parameters. These systems fix the degree of parallelism when processing a machine learning workload. However, we show that the best degree of parallelism varies at different execution stage. Fixing the parallelism limits the runtime performance. To support flexible parallelism control on the PS architecture, we propose FlexPS that makes a **multi-stage abstraction** such that workloads can **change the parallelism by need** at each stage. FlexPS further exploits optimizations such as stage scheduler, stage-aware consistency controller, flexible parameter access and direct model transfer to further improve the performance.

  I was a key member of this project.

- **LFTF: A Framework for Efficient Tensor Analytics at Scale (VLDB 2017)**

  Tensors are higher order generalizations of matrices modeling multi-aspect data, such as a set of purchase records with the schema (user_id, product_id, timestamp, feedback). Since tensor containing more information, tensor factorization becomes more and more popular in recommender systems. However, high computational cost hinders its applications to large-scale datasets. LFTF (Lock-Free Tensor Factorization) proposes a distributed framework that improves the efficiency and scalability of tensor factorization. LFTF exploits **asynchronous processing** and achieve a **lock-free execution** that significantly outperforms existing methods.

  LFTF is built upon Husky as a machine learning application. For faster training, we choose SGD as the training method. By carefully designed the data layout, LFTF can avoid conflict updates to the same parameter. I was a key member in developing the technique of lock-free asynchronous processing in this project.