Henggang Cui

Contact Information	2220A, Collaborative Innovation Center, 5000 Forbes Ave, Pittsburgh, PA, 15213	(651)403-3366 hengganc@andrew.cmu.edu

Research Interests

System Support for Large-Scale Machine Learning

Education

Carnegie Mellon University, Pittsburgh, PA.

Ph.D., Electrical and Computer Engineering, *Expected:* May, 2017

- Advisor: Greg Ganger
- Thesis Title: Exploiting Application Characteristics for Efficient System Support for Data-Parallel Machine Learning

Tsinghua University, Beijing, China

B.S., Electronic Information Science and Technology, July 2012

Research Experience

Research Assistant Aug 2012 to present

Parallel Data Lab, Carnegie Mellon University

Advisor: Greg Ganger

GeePS: Specialized Parameter Server for Deep Learning on GPUs. Deep learning tasks are often performed on GPUs, but scaling GPU applications to multiple GPU machines is challenging, because of the limited GPU memory capacity and expensive data movement overheads between GPU and CPU memory. GeePS is a parameter server that is specialized for GPU applications (especially deep learning). It allows the application to access parameter data directly through GPU memory, hiding data movement latencies from the application. GeePS also provides GPU memory management support that allows the training of deep neural networks that do not fit in GPU memory, by efficiently swapping data to/from CPU memory in the background. GeePS, as a parameter server system, can be used to scale many single-machine GPU-based ML systems, such as Caffe. Our experiments show that our GeePS-supported Caffe scales almost linearly from the original single-machine Caffe (13× more training throughput with 16 machines), and is able to efficiently support neural networks that do not fit in GPU memory.
Work published at EuroSys'16.
GeePS open-sourced at `https://github.com/cuihenggang/geeps`

IterStore: Efficient Parameter Server for Iterative ML. Many ML algorithms, including SGD, PageRank, and Gibbs Sampling, have the *iterativeness* property that the same (or nearly the same) sequence of parameter data accesses is applied to the parameter server every iteration. This repeating access sequence can be exploited to improve the parameter server throughput. We designed an efficient method to collect the repeating access sequence from the application, as well as five parameter server optimizations using the collected information, including prefetching, contiguous marshalling-free data storage, locality and numa-aware data placement, and specialized caching policy. Our experiments show that these optimizations greatly reduce the total run time of our application benchmarks by up to 50×.
Work published at SoCC'14.

Stale Synchronous Parallel (SSP): Trading Data Freshness for Speed. SSP is a flexible consistency model for synchronizing the progress of parallel ML workers. It is a middle ground between the traditional BSP approach (where the workers wait at barriers) and the Asynchrony approach (where the workers never wait).

SSP allows each worker to be a bounded number of iterations ahead of the slowest one, thus making data staleness a tunable parameter and allowing us to explicitly trade data freshness for speed. Our experiments show that SSP helps reduce the convergence time of many ML benchmarks, such as matrix factorization, LDA, and PageRank.
Work published at NIPS'13 and ATC'14.

MLtuner: Automatic ML tuning. ML tasks often require the selection and tuning of many *training tunables*—such as the learning rate, the mini-batch size, and the data staleness bound—that have a significant impact on the performance. We designed a MLtuner system that is able to automatically tune settings for those training tunables. It can be linked with existing ML systems, such as parameter servers. MLtuner uses efficient snapshotting and optimization-guided online trial-and-error to find good initial tunable settings as well as to re-tune settings during execution. Our experiments with four real ML tasks, including deep learning and matrix factorization, show that MLtuner automatically enables performance within 40–178% of having oracle knowledge of the best tunable settings, and outperforms oracle when no single set of settings are best for the entire execution.
Work under submission.

Research Intern May 2014 to Aug 2014
System Group, HP Labs
Mentor: Kimberly Keeton

Aperture: Ingest-Time Transformation for Big Time Series Data. Aperture is a database system for big time series data analytics, and is able to support interactive analytical queries with sub-second latencies on both recent data and historical data. Our approach is to apply *transformations* to the data at ingest time, so that future queries can be answered using the transformed data with much lower latencies. We have implemented several representative use cases in our system. For example, wavelet transformation can be used to compactly represent numerical time-series data, and analytical queries, such as correlation search, can be directly answered from the wavelet coefficients, thus with much lower latency. Count-min sketching can be used to represent massive counting information, such as occurrence counts of IP addresses. Ingest-time transformation can also be used to do anomaly detection. Our experiments show that our approach can speed up many analytical query benchmarks by over an order of magnitude, with minimal impact on ingest throughput.
Work published at SoCC'15.

Conference Talks

GeePS: Scalable Deep Learning on Distributed GPUs with a GPU-Specialized Parameter Server
In *EuroSys 2016*, London, UK Apr 2016

Using Data Transformations for Low-latency Time Series Analysis
In *SoCC 2015*, Kohala Coast, HI Aug 2015

Exploiting Iterative-ness for Parallel ML Computations
In *SoCC 2014*, Seattle, WA Nov 2014

Exploiting Bounded Staleness to Speed Up Big Data Analytics
In *ATC 2014*, Philadelphia, PA Jun 2014

LazyTable: Distributed Machine Learning with the Stale Synchronous Parallel Model
In *SOSP 2013 WIP Talk*, Farmington, PA Nov 2013

Publications

1. Aaron Harlap, **Henggang Cui**, Wei Dai, Jinliang Wei, Gregory R. Ganger, Phillip B. Gibbons, Garth A. Gibson, and Eric P. Xing.
   Addressing the Straggler Problem for Iterative Convergent Parallel ML.
   In *ACM Symposium on Cloud Computing (SoCC'16)*, 2016.

2. **Henggang Cui**, Hao Zhang, Gregory R. Ganger, Phillip B. Gibbons, and Eric P. Xing.
   GeePS: Scalable Deep Learning on Distributed GPUs with a GPU-Specialized Parameter Server.
   In *ACM European Conference on Computer Systems (EuroSys'16)*, 2016.

3. **Henggang Cui**, Kimberly Keeton, Indrajit Roy, Krishnamurthy Viswanathan, and Gregory R. Ganger.
   Using Data Transformations for Low-latency Time Series Analysis.
   In *ACM Symposium on Cloud Computing (SoCC'15)*, 2015.

4. Jinliang Wei, Wei Dai, Aurick Qiao, Qirong Ho, **Henggang Cui**, Gregory R. Ganger, Phillip B. Gibbons, Garth A. Gibson, Eric P. Xing.
   Managed Communication and Consistency for Fast Data-Parallel Iterative Analytics.
   In *ACM Symposium on Cloud Computing (SoCC'15)*, 2015.

5. **Henggang Cui**, Alexey Tumanov, Jinliang Wei, Lianghong Xu, Wei Dai, Jesse Haber-Kucharsky, Qirong Ho, Gregory R. Ganger, Phillip B. Gibbons, Garth A. Gibson, and Eric P. Xing.
   Exploiting Iterative-ness for Parallel ML Computations.
   In *ACM Symposium on Cloud Computing (SoCC'14)*, 2014.

6. **Henggang Cui**, James Cipar, Qirong Ho, Jin Kyu Kim, Seunghak Lee, Abhimanu Kumar, Jinliang Wei, Wei Dai, Gregory R. Ganger, Phillip B. Gibbons, Garth A. Gibson, and Eric P. Xing.
   Exploiting Bounded Staleness to Speed Up Big Data Analytics.
   In *USENIX Annual Technical Conference (ATC'14)*, 2014.

7. Qirong Ho, James Cipar, **Henggang Cui**, Jin Kyu Kim, Seunghak Lee, Phillip B. Gibbons, Garth A. Gibson, Gregory R. Ganger, and Eric P. Xing.
   More Effective Distributed ML via a Stale Synchronous Parallel Parameter Server.
   In *Neural Information Processing Systems (NIPS'13)*, 2013.

8. **Henggang Cui**, Danielle Rasooly, Moises R. N. Ribeiro, and Leonid Kazovsky.
   Optically Cross-Braced Hypercube: A Reconfigurable Physical Layer for Interconnects and Server-Centric Datacenters.
   In *Optical Fiber Communication Conference and Exposition (OFC/NFOEC)*, 2012.

9. Dan Li, **Henggang Cui**, Yan Hu, Yong Xia, and Xin Wang.
   Scalable Data Center Multicast using Multi-class Bloom Filter.
   In *19th IEEE International Conference on Network Protocols (ICNP'11)*, 2011.

Awards

Student Awards — Tsinghua University

Outstanding Graduate Award of Tsinghua University	Jul 2012
Outstanding Graduate Award of Beijing, China	Jul 2012
Outstanding Diploma Thesis Award of Tsinghua University	Jul 2012
National Scholarship of China	Nov 2011
National Scholarship of China	Nov 2010

Travel Awards

EuroSys 2016, London, UK	Apr 2016
SoCC 2014, Seattle, WA	Nov 2014
ATC 2014, Philadelphia, PA	Jun 2014
SOSP 2013, Farmington, PA	Nov 2013

Courses

Carnegie Mellon University (GPA 3.93/4)

18749 Building Reliable Distributed Systems, Fall 2015. Course project:
Elastic machine learning on temporarily available/affordable resources

15712 Advanced Topics in Operating Systems, Fall 2014. Course project:
Straggler mitigation in parallel machine learning

10780 Graduate Artificial Intelligence, Spring 2014. Course project:
An online method for adaptive data staleness tuning

15799b Advanced Topics in Database Systems, Fall 2013. Course project:
Iterativeness-aware optimization for big data analytics

18746 Storage Systems, Spring 2013. Course project:
Cloudfs, a hybrid file system integrating solid-state devices and cloud storage

15750 Graduate Algorithms, Spring 2013.

15799a Advanced Topics in OS Engineering, Fall 2012. Course project:
1. Pebbles, implementing a Unix-like kernel from scratch
2. PebPeb, paravirtualization of Pebbles kernel

10701 Machine Learning, Fall 2012. Course project:
Implementation and comparison of parallel EM on GraphLab, Spark, and Piccolo

Teaching Experience

Teaching Assistant

18746/15746 Storage Systems — Spring 2015 and Fall 2016

Skills

Computer Programming
C++, Python, C, CUDA, Java, Matlab, Shell, Perl, Scala

Software Systems
Git, Linux, LaTeX, SVN

Big data Systems
Caffe, GraphLab, MPI, Spark, HBase, OpenTSDB, Hadoop