Dr. Michael E. Bauer

Contact Information

630 Boren Avenue N
Apartment 911
Seattle, WA 98109

phone: (443)-465-3699
e-mail: mike@lightsighter.org
website: http://lightsighter.org

Research

High performance parallel programming systems for distributed heterogeneous architectures.
Warp-specialized GPU programming techniques.
Python for high performance data analytics and scientific computing.

Education

Stanford University, Stanford, CA, USA **2008-2014**
Ph.D., Computer Science (advised by Dr. Alex Aiken)
Thesis: *Legion: Programming Distributed Heterogeneous Architectures with Logical Regions*
M.S., Computer Science, June 2011

Duke University, Durham, NC, USA **2004-2008**
Magna Cum Laude, GPA: 3.865/4.0
B.S.E. Electrical Computer Engineering with Honors (advised by Dr. Daniel Sorin)
Thesis: *Proving the Completeness of Error Detection Mechanisms in Simple Core Multiprocessors*
B.S. Mathematics with High Honors (advised by Dr. Anne Catllá)
Thesis: *Faraday Waves Arising from Square Wave Forcing of the Damped Mathieu Equation*
B.S. Computer Science

Professional Experience

NVIDIA Research, Santa Clara, CA, USA
Principal Research Scientist **2014-Present**
Design and implementation of high performance runtime systems for supercomputers.

NVIDIA Research, Santa Clara, CA, USA
Part-Time Contractor **2011-2012**
Implementation, testing, and evaluation of the CudaDMA library on the Kepler GPU architecture. Experiments with a reverse time migration application leveraging CudaDMA used to extrapolate the potential benefits of incorporating hardware DMA engines on GPUs for efficient data movement.

NVIDIA Research, Santa Clara, CA, USA
Summer Intern **2010**
Design and evaluation of scalable software cache coherence protocols for future generation GPUs.

Refereed Conference Publications

Index Launches: Scalable, Flexible Representations of Parallel Task Groups,
Rupanshu Soi, Michael Bauer, Sean Treichler, Manolis Papadakis, Wonchan Lee,
Patrick McCormick, Alex Aiken, Elliott Slaughter
Supercomputing (**SC**), 2021.

Scaling Implicit Parallelism via Dynamic Control Replication,
Michael Bauer, Wonchan Lee, Elliott Slaughter, Zhihao Jia, Mario Di Renzo, Manolis Papadakis,
Galen Shipman, Pat McCormick, Michael Garland, Alex Aiken
Principles and Practices of Parallel Programming (**PPoPP**), 2021.

Legate NumPy: Accelerated and Distributed Array Computing,
Michael Bauer, Michael Garland
Supercomputing (**SC**), 2019.

**DNS of Autoignition of Diesel Surrogate Fuel in a Turbulent Jet at
High Pressure with S3D-Legion on Titan/Summit**,
Jacqueline Chen, Elliott Slaughter, Wonchan Lee, Aditya Konduri, Hemanth Kolla, Alex Aiken,
Sean Treichler, Michael Bauer, Giulio Borghesi
International Conference on Numerical Combustion (**NC**), 2019.

Dynamic Tracing: Just-In-Time Specialization of Task Graphs for Dynamic Task-Based Runtimes,
Wonchan Lee, Elliott Slaughter, Michael Bauer, Sean Treichler, Todd Warszawski,
Michael Garland, Alex Aiken
Supercomputing (**SC**), 2018.

Integrating External Resources with a Task-Based Programming Model,
Zhihao Jia, Sean Treichler, Galen Shipman, Michael Bauer, Noah Watkins, Carlos Maltzahn,
Patrick McCormick, Alex Aiken
International Conference on High Performance Computing (**HiPC**), 2017.

Control Replication: Compiling Implicit Parallelism to Efficient SPMD with Logical Regions,
Elliott Slaughter, Wonchan Lee, Sean Treichler, Wen Zhang, Michael Bauer, Galen Shipman,
Patrick McCormick, Alex Aiken
Supercomputing (**SC**), 2016.

Dependent Partitioning,
Sean Treichler, Michael Bauer, Rahul Sharma, Elliott Slaughter, Alex Aiken
Object Oriented Programming, Systems, Languages, and Applications (**OOPSLA**), 2016.

Regent: A High-Productivity Programming Language for HPC with Logical Regions,
Elliott Slaughter, Wonchan Lee, Sean Treichler, Michael Bauer, Alex Aiken
Supercomputing (**SC**), 2015.

Verification of Producer-Consumer Synchronization in GPU Programs,
Michael Bauer, Rahul Sharma, Alex Aiken
Programming Language Design and Implementation (**PLDI**), 2015.

Structure Slicing: Extending Logical Regions with Fields,
Michael Bauer, Sean Treichler, Elliott Slaughter, Alex Aiken
Supercomputing (**SC**), 2014.

Realm: An Event-Based Low-Level Runtime for Distributed Memory Architectures,
Sean Treichler, Michael Bauer, Alex Aiken
Parallel Architectures and Compilation Techniques (**PACT**), 2014.

Singe: Leveraging Warp Specialization for High Performance on GPUs,
Michael Bauer, Sean Treichler, Alex Aiken
Principles and Practices of Parallel Programming (**PPoPP**), 2014.

Language Support for Dynamic, Hierarchical Data Partitioning,
Sean Treichler, Michael Bauer, Alex Aiken
Object-Oriented Programming, Systems, and Languages (**OOPSLA**), 2013.

Legion: Expressing Locality and Independence with Logical Regions,
Michael Bauer, Sean Treichler, Elliott Slaughter, Alex Aiken
Supercomputing (**SC**), 2012.

CudaDMA: Optimizing GPU Memory Bandwidth via Warp Specialization,
Michael Bauer, Henry Cook, Brucek Khailany
Supercomputing (**SC**), 2011.

**Programming the Memory Hierarchy Revisited:
Supporting Irregular Parallelism in Sequoia**,
Michael Bauer, John Clark, Eric Schkufza, Alex Aiken
Principles and Practices of Parallel Programming (**PPoPP**), 2011.

Argus: Low-Cost, Comprehensive Error Detection in Simple Cores,
Albert Meixner, Michael Bauer, Daniel Sorin
International Symposium on Microarchitecture (**MICRO**), 2007.

**Reducing the Impact of Process Variability with Prefetching and
Criticality-Based Resource Allocation**,
Bogdan Romanescu, Michael Bauer, Daniel Sorin, Sule Ozev
Parallel Architectures and Compilation Techniques (**PACT**), 2007.

Invited Talks

Reprising the Zen of Python for High Performance Computing,
Keynote Address at Workshop on Python for High Performance and Scientific Computing, 2020.

Software Systems

Legion: `legion.stanford.edu`
Implemented the majority of the Legion runtime system for executing high performance applications on large distributed clusters with heterogeneous hardware including CPUs and GPUs. Constructed an event-driven implementation to support Legion's deferred execution model. Also designed algorithms for managing distributed meta-data structures across distinct address spaces.

CudaDMA: `lightsighter.github.io/CudaDMA/`
Primary developer of the CudaDMA library which supports high-performance data transfer patterns between on-chip and off-chip memories for NVIDIA GPUs. Used template meta-programming to statically compute optimal data transfer algorithms without requiring modifications to the CUDA compiler. Invented warp specialization as a way to overlap data transfers with computations and decouple transfer pattern abstractions from their implementations.

Sequoia++: `sequoia.stanford.edu`
Lead developer of an extension to the Sequoia language for supporting dynamic parallelism. Designed and implemented extensions to the Sequoia programming model, language, compiler, and runtime.

Technical Skills

Fluent Languages: C, C++, Java, Python, CUDA, MATLAB

Competent Languages: VHDL, Perl, Scala, Haskell, Shell Script, PHP, SQL, Javascript, Fortran

Programming Systems: MPI, GASNet, IBVerbs, uGNI, P-Threads, OpenMP, OpenACC, OpenCL, OpenGL, DirectX, GLSL, SSE, AVX, x86 assembly, Lex, Flex, Yacc, Bison, ASTGen, PBS, Torque, Hadoop, Map-Reduce, Software Transactional Memory, LaTeX(e.g. this CV)

Fellowships and Honors

NVIDIA Graduate Fellowship Recipient, 2010 and 2011

Pratt Research Fellow, Duke University Pratt School of Engineering, 2007

PRUV Research Fellow, Duke University Department of Mathematics, 2006

Mathematical Competition in Modeling: Outstanding Ranking (top <1%), 2007

Finisher: 2007 Boston Marathon 2:52:43 (446 out of 20,388)

Professional Service

Program Committee: IPDPS 2014, PPoPP 2015
External Review Committee: PPoPP 2014
External Reviewer: PLDI 2010, OOPSLA 2010, PLDI 2012, SC 2012, SC 2013, MICRO 2013