List of Publications with Abstracts

Orion Sky Lawlor
Department of Computer Science
University of Alaska at Fairbanks
olawlor@acm.org

- 2009 “Message Passing for GPGPU Clusters: cudaMPI” [**?**]

  We present and analyze two new communication libraries, cudaMPI and glMPI, that provide an MPI-like message passing interface to communicate data stored on the graphics cards of a distributed-memory parallel computer. These libraries can help applications that perform general purpose computations on these networked GPU clusters. We explore how to efficiently support both point-to-point and collective communication for either contiguous or noncontiguous data on modern graphics cards. Our software design is informed by a detailed analysis of the actual performance of modern graphics hardware, for which we develop and test a simple but useful performance model.

- 2007 “MPIglut: Powerwall Programming Made Easier” [1]

  A powerwall is an array of separate screens that work together to provide a single unified display. Powerwalls are often driven by a small cluster, which requires parallel software to organize and synchronize the distributed rendering process. This paper describes MPIglut, our powerwall-friendly implementation of the popular sequential GLUT OpenGL 3D programming interface. MPIglut internally communicates using MPI to provide a single coherent display even across a distributed-memory parallel machine. Uniquely, MPIglut is source-code compatible with ordinary sequential GLUT code while providing high performance.

- 2006 “Interpolation-Friendly Soft Shadow Maps” [2]

  We present Penumbra Limit Maps, a technique for extracting soft shadows from an modified shadow map. The shadow representation used by our method has excellent interpolation properties, allowing the shadow boundary to be rendered with sub shadowmap-pixel accuracy, which partially mitigates the resolution problems common to shadow map methods. Unlike similar shadow map methods, our method includes both inner and outer penumbrae, and is in fact physically correct for the simplest case of a straight object edge and infinitely distant extended light source. At object corners or where multiple object edges overlap, our method is no longer physically exact, but still gives plausible results. Finally, we show the method can be implemented naturally and efficiently on programmable graphics hardware.

- 2006 “Multiple Flows of Control in Migratable Parallel Programs” [3]

Many important parallel applications require multiple flows of control to run on a single processor. In this paper, we present a study of four flow-of-control mechanisms: processes, kernel threads, user-level threads and event-driven objects. Through experiments, we demonstrate the practical performance and limitations of these techniques on a variety of platforms. We also examine migration of these flows-of-control with focus on thread migration, which is critical for application-independent dynamic load balancing in parallel computing applications. Thread migration, however, is challenging due to the complexity of both user and system state involved. In this paper, we present several techniques to support migratable threads and compare the performance of these techniques.

- 2006 "ParFUM: A Parallel Framework for Unstructured Meshes for Scalable Dynamic Physics Applications" [4]

  Unstructured meshes are used in many engineering applications with irregular domains, from elastic deformation problems to crack propagation to fluid flow. Because of their complexity and dynamic behavior, the development of scalable parallel software for these applications is challenging. The Charm++ Parallel Framework for Unstructured Meshes allows one to write parallel programs that operate on unstructured meshes with only minimal knowledge of parallel computing, while making it possible to achieve excellent scalability even for complex applications. Charm++'s messagedriven model enables computation/communication overlap, while its runtime load balancing capabilities make it possible to react to the changes in computational load that occur in dynamic physics applications. The framework is highly flexible and has been enhanced with numerous capabilities for the manipulation of unstructured meshes, such as parallel mesh adaptivity and collision detection.

- 2006 "A system integration framework for coupled multiphysics simulations" [5]

  Multiphysics simulations are playing an increasingly important role in computational science and engineering for applications ranging from aircraft design to medical treatments. These simulations require integration of techniques and tools from multiple disciplines, and in turn demand new advanced technologies to integrate independently developed physics solvers eectively. In this paper, we describe some numerical, geometrical, and system software components required by such integration, with a concrete case study of detailed, three dimensional, parallel rocket simulations involving system level interactions among fluid, solid, and combustion, as well as subsystem-level interactions. We package these components into a software framework that provides state of the art, common-refinement based methods for transferring data between potentially nonmatching meshes, novel and robust face-offsetting methods for tracking Lagrangian surface meshes, as well as integrated support for parallel mesh optimization, remeshing, algebraic manipulations, performance monitoring, and high-level data management and I/O. From these general, reusable framework components we construct domain-specic building blocks to facilitate integration of parallel, multiphysics simulations from high level specifications that are easy to read and can also be visualized graphically. These reusable building blocks are integrated with independently developed physics codes to perform various multiphysics simulations.

- 2005 "Performance Degradation in the Presence of Subnormal Floating-Point Values" [6]

  Operating system interference in parallel programs can cause tremendous performance degradation. This paper discusses the interference caused by the quiet generation of subnormal floating point values. We analyze the performance impact of subnormal values in a parallel simulation of a stress wave propagating through a three dimensional bar. The floating-point exception handling mechanisms of various parallel architectures and operating systems lead to widely differing performance for the same program. We show that a parallel program will exhibit greatly amplified performance degradation due to this interference. In addition we provide an simple example program that demonstrates underflow on a single processor. Finally we suggest a novel option for fixing these undesired slowdowns.

- 2005 "An Integration Framework for Simulations of Solid Rocket Motors" [7]

  Simulation of solid rocket motors requires coupling physical models and software tools from multiple disciplines, and in turn demands advanced techniques to integrate independently developed physics solvers effectively. In this paper, we overview some computer science components required for such integration. We package these components into a software framework that provides system support of high-level data management and performance monitoring, as well as computational services such as novel and robust algorithms for tracking Lagrangian surface meshes, parallel mesh optimization, and data transfer between nonmatching meshes. From these reusable framework components we construct domain-specic building blocks to facilitate integration of parallel, multiphysics simulations from high-level specications. Through examples, we demonstrate the exibility of our framework and its components.

- 2004 "Impostors for Parallel Interactive Computer Graphics" [8]

  We demonstrate an interactive parallel rendering system based on the impostors technique. Impostors increase the latency tolerance of an interactive rendering system, which allows us to use the power of a parallel machine even at high resolutions and framerates. Impostors also decrease the required rendering bandwidth, which makes possible the interactive use of a variety of advanced rendering techniques. These techniques are demonstrated by the interactive high-quality rendering of very large detailed models on large distributed memory parallel machines.

- 2004 "Debugging Support for Charm++" [9]

  This paper describes a parallel debugger and the related debugging support implemented for CHARM++, a data driven parallel programming language. Because we build extensive debugging support into the parallel runtime system, applications can be debugged at a very high level.

- 2004 "Performance Modeling and Programming Environments for Petaflops Computers and the Blue Gene Machine" [10]

  We present a performance modeling and programming environment for petaflops computers and the Blue Gene machine. It consists of a parallel simulator, BigSim, for predicting performance of machines with a very large number of processors, and BigNetSim, an ongoing effort to incorporate a pluggable module of a detailed contention based network model. It provides the ability to make performance predictions for machines such as BlueGene/L. We also explore the programming environments for several planned applications on the machines including Finite Element Method (FEM) simulation.

- 2003 "Bounding Iterated Function Systems using Convex Optimization" [11]

  We present an algorithm to construct a tight bounding polyhedron for a recursive procedural model. We first use an iterated function system (IFS) to represent the extent of the procedural model. Then we present a novel algorithm that expresses the IFS-bounding problem as a set of linear constraints on a linear objective function, which can then be solved via standard techniques for linear convex optimization. As such, our algorithm is guaranteed to find the recursively optimal bounding polyhedron, if it exists. Finally, we demonstrate examples of this algorithm on two and three dimensional recursive procedural models.

- 2003 "Supporting dynamic parallel object arrays" [12]

  We present efficient support for generalized arrays of parallel data driven objects. Array elements are regular C++ objects, and are scattered across the parallel machine. An individual element is addressed by its "index", which can be an arbitrary object rather than a simple integer. For example, an array index can be a series of numbers, supporting multidimensional sparse arrays; a bit vector, supporting collections of quadtree nodes; or a string. Methods can be invoked on any individual array element from any processor, and the elements can participate in reductions and broadcasts. Individual elements can be created or deleted dynamically at any time. Most importantly, the elements can migrate from processor to processor at any time. The paper discusses support for message delivery and collective operations in face of such dynamic behavior. The migration capabilities of array elements have proven extremely useful, for example, in implementing flexible load balancing strategies and for exploiting workstation clusters adaptively. We present the design, an implementation, and performance results.

- 2003 "Adaptive MPI" [13]

  "Adaptive MPI", or AMPI, implements virtual MPI processors, several of which may reside on a single physical processor. This virtualization allows MPI applications to use an automatic migration-based load balancer, automatically overlap computation and communication, and provides several other benefits. In this paper, we present the design of and recent work on AMPI, its low-level and application performance, and some of the advanced capabilities enabled by virtualization.

- 2002 "A Voxel-Based Parallel Collision Detection Algorithm" [14]

  Two physical objects cannot occupy the same space at the same time. Simulated physical objects do not naturally obey this constraint. Instead, we must detect when two objects have collided–we must perform collision detection. This work presents a simple voxel-based collision detection algorithm, an efficient parallel implementation of the algorithm, and performance results.

- 2001 "A Grid-Based Parallel Collision Detection Algorithm" [15]

  In this work, we present a scalable high-level parallel solution to a large subclass of collision detection problems. Our approach is to divide space into a sparse grid of regular axis-aligned voxels distributed across the parallel machine. Objects are then sent to all the voxels they intersect. Once all the objects have arrived, each voxel becomes a self-contained subproblem, which is then solved using standard serial collision detection approaches. This voxel-based approach efficiently and naturally separates many objects that cannot ever collide, by placing them in separate voxels. Simultaneously, voxels bring together adjacent objects that may intersect.

- 2001 “Supporting Dynamic Parallel Object Arrays” [16]

  We present efficient support for generalized arrays of parallel data driven objects. The ”array elements” are scattered across a parallel machine. Each array element is an object that can be thought of as a virtual processor. The individual elements are addressed by their ”index”, which can be an arbitrary object rather than a simple integer. For example, it can be a series of numbers, supporting multidimensional sparse arrays; a bit vector, supporting collections of quadtree nodes; or a string. Messages can be sent to any individual array element from any processor, and the elements can participate in reductions and broadcasts. Individual elements can be created or deleted dynamically at any time. Most importantly, the elements can migrate from processor to processor at any time. The paper discusses support for message delivery and collective operations in face of such dynamic behavior. The migration capabilities of array elements have proven extremely useful, for example, in implementing flexible load balancing strategies and for exploiting workstation clusters adaptively.

- 1999 “Room Capacity Analysis Using a Pair of Evacuation Models” [17]

  We present two models for determining the amount of time it takes a given number of people to evacuate a given room. A room’s maximum capacity can be derived from this by imposing a maximum evacuation time. The maximum evacuation time must take into account factors such as the fire resistance of the room and should be calculated, for example, by the Fire Marshall. We developed a graph-based network flow simulation. People are modeled as a compressible fluid which flows toward and out the exit. This model assumes people’s interaction properties, based on industry research. We also developed a discrete particle simulation. In this model, people are modeled as disks that attempt to reach the exits. In this model, people’s interaction properties emerge from local, per-person assumptions. In this paper, we develop and analyze both models. We then compare and evaluate the models’ outputs, and finally analyze the capacity of a local dining hall, gymnasium, lecture hall, and swimming pool.

- 1999 “Automated DEM Production using ESA Tandem Mission Data for the Caribou-Poker Creek LTER Watershed, Alaska” [18]

  The Alaska SAR Facility has developed an automated procedure for production of a digital elevation model (DEM) from ESA tandem mission data. This software processes from ASF Computer Compatible Signal Data (CCSD) or Level Zero (raw) products to a map-projected, ground-range 30m DEM. Several advanced techniques have been integrated into the procedures to improve accuracy and to allow full automation. This spring, a test site at the Caribou-Poker Long Term Ecological Research (LTER) watershed was analyzed in collaboration with Larry Hinzman (Principal Investigator) and the Arctic Region Supercomputing Center. An accuracy assessment of the results for four ESA tandem pairs showed average differences ranging from approximately 4 to 8 meters in elevation when compared with field measurements using differential GPS. The final mosaic had an average difference of 4.68 meters in elevation. Point target analysis of existing 3x6 and 2x3 arc-second USGS digital elevation data showed an average difference of 19.97 and 10.04 meters in elevation, respectively. These advances in accuracy are due to use of precision timing and orbital data in an interferometric SAR processor using an average Doppler, precise baseline refinement, and direct ground rectification. The Alaska SAR Facility continues to refine its interferometric SAR processor in support of NASA-approved users.

- 1998 "Automated Digital Elevation Model (DEM) Production Using ERS SAR Tandem Pairs" [19]

- 1998 "Repeat-Pass Satellite Interferometric Tools Available at the Alaska SAR Facility" [20]

- 1997 "Pursuit-Evasion Games in the Late Cretaceous" [21]

  We have been asked by a group of paleontologists to model the hunting and evasion strategies of the predator Velociraptor mongoliensis as, alone or with a friend, he pursues the prey species Thescelosaurus neglectus; both extinct dinosaurs. The predator is 6/5ths faster than the prey, but has a 3-times larger turning radius at top speed. The chase continues for only 15 seconds, whereupon the predator must stop because of a lactic acid buildup in his muscles.

- 1997 "Towards Operational Application of Satellite SAR Images in Hydrological Studies" [22]

- 1997 "Generation of Fine Resolution DEM at Test Areas in Alaska Using ERS Tandem Pairs and Precise Orbital Data" [23]

  The ASF Science division has released the world's first free end-to-end interferometric digital elevation model (DEM) generation system. This software, which processes from raw signal data through to a map-projected, ground-range 20m DEM, is completely automated. Preliminary comparison with differential gobal positioning system (GPS) indicates that over a 100km swath, horizontal position errors are less than 120m, and comparison with the 2x3 arc second United States Geological Survey (USGS) DEM indicates an average vertical error of 7m, 25m RMS. This result was obtained from an ERS tandem pair over Delta Junction, Alaska. These advances in accuracy are due to use of precision timing and orbital data in an interferometric SAR processor using an average doppler, precise baseline refinement, and direct ground rectification. The computationally intensive nature of these algorithms was minimized through the creation of a parallel SAR processor and a linearized ground rectification procedure.

References

[1] Orion Sky Lawlor. Message passing for gpgpu clusters: cudampi. In *IEEE Cluster PPAC Workshop*, 2009.

[2] Orion Sky Lawlor, Matthew Page, and Jon Genetti. MPIglut: Powerwall programming made easier. *Journal of WSCG*, pages 130–137, 2008.

[3] Orion Sky Lawlor. Interpolation-friendly soft shadow maps. In *Proceedings of Computer Graphics and Virtual Reality '06*, pages 111–117. CSREA Press, 2006.

[4] Gengbin Zheng, Orion Sky Lawlor, and Laxmikant V. Kale. Multiple flows of control in migratable parallel programs. In *Proceedings of 8th Workshop on High Performance Scientific and Engineering Computing (HPSEC-06)*. IEEE Press, August 2006.

[5] Orion Sky Lawlor, Sayantan Chakravorty, Terry L. Wilmarth, Nilesh Choudhury, Isaac Dooley, Gengbin Zheng, and Laxmikant V. Kale. ParFUM: A parallel framework for unstructured meshes for scalable dynamic physics applications. *Engineering With Computers*, 22(3):215–235, 2006.

[6] Xiangmin Jiao, Gengbin Zheng, Phillip A. Alexander, Michael T. Campbell, Orion S. Lawlor, John Norris, Andreas Haselbacher, and Michael T. Heath. A system integration framework for coupled multiphysics simulations. *Engineering with Computers*, 22(3–4):293–309, 2006.

[7] Orion Sky Lawlor, Hari Govind, Isaac Dooley, Michael Breitenfeld, and Laxmikant Kale. Performance degradation in the presence of subnormal floating-point values. In *Proceedings of the International Workshop on Operating System Interference in High Performance Applications*, September 2005.

[8] Xiangmin Jiao, Gengbin Zheng, Orion Lawlor, Phil Alexander, Mike Campbell, Michael Heath, and Robert Fiedler. An integration framework for simulations of solid rocket motors. In *41st AIAA/ASME/SAE/ASEE Joint Propulsion Conference*, Tucson, Arizona, July 2005.

[9] Orion Sky Lawlor. *Impostors for Parallel Interactive Computer Graphics*. PhD thesis, University of Illinois at Urbana-Champaign, December 2004.

[10] Rashmi Jyothi, Orion Sky Lawlor, and L. V. Kale. Debugging support for Charm++. In *PADTAD Workshop for IPDPS 2004*, page 294. IEEE Press, April 2004.

[11] Gengbin Zheng, Terry Wilmarth, Orion Sky Lawlor, Laxmikant V. Kalé, Sarita Adve, David Padua, and Philippe Geubelle. Performance modeling and programming environments for petaflops computers and the Blue Gene machine. In *NSF Next Generation Systems Program Workshop, 18th International Parallel and Distributed Processing Symposium(IPDPS)*, Santa Fe, New Mexico, April 2004. IEEE Press.

[12] Orion Sky Lawlor and John Hart. Bounding iterated function systems using convex optimization. In *Proceedings of Pacific Graphics 2003*, pages 283–292, October 2003.

[13] Orion Sky Lawlor and L. V. Kalé. Supporting dynamic parallel object arrays. *Concurrency and Computation: Practice and Experience*, 15:371–393, 2003.

[14] Chao Huang, Orion Sky Lawlor, and L. V. Kalé. Adaptive MPI. In *Proceedings of the 16th International Workshop on Languages*

and Compilers for Parallel Computing (LCPC 03), College Station, Texas, October 2003.

[15] Orion Sky Lawlor and L. V. Kalé. A voxel-based parallel collision detection algorithm. In *Proceedings of the International Conference in Supercomputing*, pages 285–293. ACM Press, June 2002.

[16] Orion Sky Lawlor. A grid-based parallel collision detection algorithm. Master's thesis, Dept. of Computer Science, University of Illinois, May 2001.

[17] Orion Sky Lawlor and L. V. Kalé. Supporting dynamic parallel object arrays. In *Proceedings of ACM 2001 Java Grande/ISCOPE Conference*, pages 21–29, Stanford, CA, June 2001.

[18] Gregg A. Christopher, Orion Lawlor, and Jason Tedor. Room capacity analysis using a pair of evacuation models. *UMAP*, 20.3, 1999.

[19] Rick Guritz, Mike Ayers, Tom Logan, Shusun Li, and Orion Lawlor. Automated DEM production using ESA tandem mission data for the Caribou-Poker Creek LTER watershed, Alaska. In *Fringe '99 Meeting*, November 1999.

[20] Rick Guritz, Orion Lawlor, Tom Logan, Robert Fatland, Joanne Groves, Shusun Li, and Verne Kaupp. Automated Digital Elevation Model (DEM) production using ERS SAR tandem pairs. In *Proceedings of IGARSS '98*, 1998.

[21] Rick Guritz, Orion Lawlor, Tom Logan, Shusun Li, , and Verne Kaupp. Repeat-pass satellite interferometric tools available at the Alaska SAR Facility. In *American Geophysical Union 1998 Fall Meeting Transactions, EOS*, volume 79.45, pages 21–22, 1998.

[22] Gordon Bower, Orion Lawlor, and James Long. Pursuit-evasion games in the late Cretaceous. *UMAP*, 18.3, 1997.

[23] Shusun Li, Rick Guritz, Tom Logan, Z. Wang, and Orion Lawlor. Towards operational application of satellite SAR images in hydrological studies. In *American Geophysical Union EOS*, volume 78.45, 1997.

[24] Orion Lawlor, Tom Logan, Rick Guritz, Robert Fatland, Shusun Li, Z. Wang, and Coert Olmsted. Generation of fine resolution DEM at test areas in Alaska using ERS tandem pairs and precise orbital data. In *12th International Conference on Applied Geologic Remote Sensing*, November 1997.