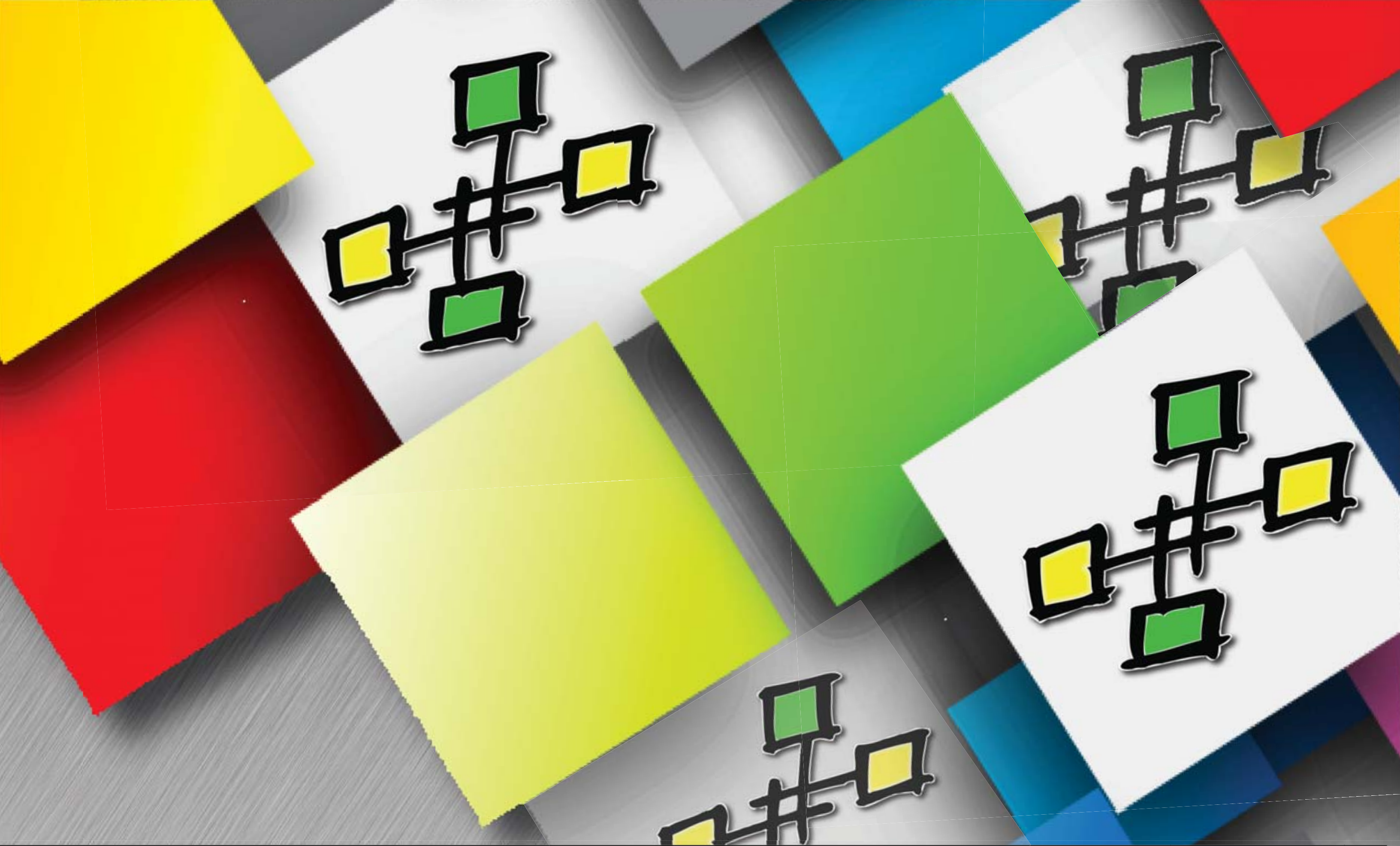


OpenSHMEM 2014

The Future of OpenSHMEM and Related Technologies

Workshop Program & Abstracts

March 04-06, 2014

Register at:

http://www.csm.ornl.gov/workshops/openshmem2013/

OpenSHMEM 2014

Program Chairs

Oscar Hernandez
Oak Ridge National Laboratory
PO BOX 2008 MS-6164
Oak Ridge, TN 37831-6164

Steve Poole
Oak Ridge National Laboratory
PO BOX 2008 MS-6406
Oak Ridge, TN 37831-6406

Pavel Shamis
Oak Ridge National Laboratory
PO BOX 2008 MS6164
Oak Ridge, TN 37831-6164

Workshop Organizers

Jennifer Goodpasture
Oak Ridge National Laboratory
PO BOX 2008 MS-6406
Oak Ridge, TN 37831-6406

Lora Wolfe
Oak Ridge National Laboratory
PO BOX 2008 MS-6164
Oak Ridge, TN 37831-6164

Program Committee

Barbara Chapman
University of Houston

Steve Poole
Oak Ridge National Laboratory

Tony Curtis
University of Houston

Barney Maccabe
Oak Ridge National Laboratory

Nick Park
Department of Defense

Duncan Poole
NVIDIA, Corporation

Sameer Shende
University of Oregon

Wolfang Nagel
TU-Dresden

Duncan Roweth
Cray, Inc.

Gary Grider
Los Alamos National Laboratory

Manjunath Venkata
Oak Ridge National Laboratory

Gilad Shainer
Mellanox Technologies

Matt Baker
Oak Ridge National Laboratory

Laura Carrington
San Diego Supercomputer Center

Monika ten Bruggencate
Cray, Inc.

George Bosilca
University of Tennessee

Gregory Koenig
Oak Ridge National Laboratory

Josh Lothian
Oak Ridge National Laboratory

Chung-Hsing Hsu
Oak Ridge National Laboratory

"THE FUTURE OF OPENSHMEM AND RELATED TECHNOLOGIES"

OpenSHMEM is a modern derivative of the SGI SHMEM API originally developed by Cray Research, Inc. for efficient programming of large-scale systems. Because of its strong following among users, the Extreme Scale Systems Center at Oak Ridge National Laboratory, together with the University of Houston, led the effort to standardize the API with input from the vendors and user community. In 2012, version 1.0 was released and opened for comments or further revisions. The goal of OpenSHMEM is to make sure OpenSHMEM implementations that are portable across multiple vendors, including SGI, Cray, IBM, Hewlett-Packard, Intel, and Mellanox Technologies.

OpenSHMEM is a partitioned global address space (PGAS) one-sided communications library that is capable of decoupling the data transfer from the synchronization of the communication source and target. It uses remote memory access (RMA) for one-sided communication mechanism that allows data to be transferred from one processing element (PE) memory space to another (remote) PE memory space. The RMA operation is described entirely by one PE (the active side) without the direct intervention of the other PE (the passive side), minimizing the overhead of communication. Irregular communication patterns with small/medium sized data transfers, in which data source and data target are not known a priori often benefit from a one-sided communication library such as OpenSHMEM.

The OpenSHMEM API provides concise and powerful library calls for communicating and processing data. The API has a limited set of communication calls that can provide opportunities for overlapping communication with computation. The OpenSHMEM API provides calls for data communication, group synchronizations, data collection and reduction operations, distributed locks, and process and data accessibility checking.

THE OPENSHMEM 2014 WORKSHOP

The OpenSHMEM workshop is an annual event dedicated to the promotion and advancement of parallel programming with the OpenSHMEM programming interface and to helping shape its future direction. It is the premier venue to discuss and present the latest developments, implementation technology, tools, trends, recent research ideas and results related to OpenSHMEM and its use in applications. As we move to Exascale, there are several areas in OpenSHMEM that we need to address to ensure that OpenSHMEM will work on future systems, including better scalability, resilience, I/O, multi-threading support, power/energy efficiency, locality, etc.

This year's workshop will emphasize the future direction of OpenSHMEM and related technologies, tools and frameworks. We will have several invited speakers and 14 papers from the industry and academia, which will talk about the current state of OpenSHMEM and its future in terms of the future hardware trends, and ideas for future extensions for OpenSHMEM. On the last day of the workshop we will have a panel session where we will summarize the results of the workshop and propose roadmap for the future of OpenSHMEM, with input from the community.

KEYNOTE

PRESENTED BY

DR. VIVEK SARKAR, E.D. BUTCHER CHAIR IN ENGINEERING, RICE UNIVERSITY

MARCH 5 AT 11:50 A.M. - 1:00 P.M.
LUNCH PROVIDED

Vivek Sarkar conducts research in multiple aspects of parallel software including programming languages, program analysis, compiler optimizations and runtimes for parallel and high performance computer systems. He currently leads the Habanero Multicore Software Research project at Rice University, and serves as Associate Director of the NSF Expeditions project on the Center for Domain-Specific Computing. Prior to joining Rice in July 2007, Vivek was Senior Manager of Programming Technologies at IBM Research. His responsibilities at IBM included leading IBM's research efforts in programming model, tools, and productivity in the PERCS project during 2002- 2007 as part of the DARPA High Productivity Computing System program. His past projects include the X10 programming language, the Jikes Research Virtual Machine for the Java language, the MIT RAW multicore project, the ASTI optimizer used in IBM's XL Fortran product compilers, the PTRAN automatic parallelization system, and profile-directed partitioning and scheduling of Sisal programs. Vivek holds a B.Tech. degree from the Indian Institute of Technology, Kanpur, an M.S. degree from University of Wisconsin-Madison, and a Ph.D. from Stanford University. He became a member of the IBM Academy of Technology in 1995, the E.D. Butcher Chair in Engineering at Rice University in 2007, and was inducted as an ACM Fellow in 2008. Vivek has been serving as a member of the US Department of Energy's Advanced Scientific Computing Advisory Committee (ASCAC) since 2009. He has also become the chair of the Computer Science Department at Rice University since July 2013.

HYBRID PROGRAMMING CHALLENGES FOR EXTREME SCALE SOFTWARE

KEYNOTE

It is widely recognized that computer systems in the next decade will be qualitatively different from current and past computer systems. Specifically, they will be built using homogeneous and heterogeneous many-core processors with 100's of cores per chip, their performance will be driven by parallelism (million-way parallelism just for a departmental server), and constrained by energy and data movement. They will also be subject to frequent faults and failures. Unlike previous generations of hardware evolution, these Extreme Scale systems will have a profound impact on future software. The software challenges are further compounded by the need to support new workloads and application domains that have traditionally not had to worry about parallel computing in the past.

In general, a holistic redesign of the entire software stack is needed to address the programmability and performance requirements of Extreme Scale systems. This redesign will need to span programming models, languages, compilers, runtime systems, and system software. A major challenge in this redesign arises from the fact that current programming systems have their roots in execution models that focused on homogeneous models of parallelism e.g., OpenMP's roots are in SMP parallelism, MPI and SHMEM's roots are in cluster parallelism, and CUDA and OpenCL's roots are in GPU parallelism. This in turn leads to the "hybrid programming" challenge for application developers, as they are forced to explore approaches to combine two or all three of these models in the same application. Despite some early experiences and attempts by some of the programming systems to broaden their scope (e.g., addition of accelerator pragmas to OpenMP), hybrid programming remains an open problem and a major obstacle for application enablement on future systems.

In this talk, we summarize experiences with hybrid programming in the Habanero Multicore Software Research project [1] which targets a wide range of homogeneous and heterogeneous manycore processors in both single-node and cluster configurations. We focus on key primitives in the Habanero execution model that simplify hybrid programming, while also enabling a unified runtime system for heterogeneous hardware. Some of these primitives are also being adopted by the new Open Community Runtime (OCR) open source project [2]. These primitives have been validated in a range of applications, including medical imaging applications studied in the NSF Expeditions Center for Domain-Specific Computing (CDSC) [3].

Background material for this talk will be drawn in part from the DARPA Exascale Software Study report [4] led by the speaker. This talk will also draw from a recent (March 2013) study led by the speaker on Synergistic Challenges in Data-Intensive Science and Exascale Computing [5] for the US Department of Energy's Office of Science. We would like to acknowledge the contributions of all participants in both studies, as well as the contributions of all members of the Habanero, OCR, and CDSC projects.

REFERENCES:
[1] Habanero Multicore Software Research project. http://habanero.rice.edu.
[2] Open Community Runtime (OCR) open source project. https://01.org/projects/open-community-runtime.
[3] Center for Domain-Specific Computing (CDSC). http://cdsc.ucla.edu.
[4] DARPA Exascale Software Study report, September 2009. http://users.ece.gatech.edu/~mrichard/ExascaleComputingStudyReports/ECS_reports.htm.
[5] DOE report on Synergistic Challenges in Data-Intensive Science and Exascale Computing, March 2013. http://science.energy.gov/~/media/ascr/ascac/pdf/reports/2013/ASCAC_Data_Intensive_Computing_report_final.pdf.

DAY 1

TUESDAY 04 MARCH

8:00 AM **Registration Opens**

8:30 AM **Morning Tutorials**

OpenSHMEM/UCCS Tutorial

Held in Governor Calvert Ballroom East, located in the Governor Calvert House

Tutorial led by *Tony Curtis, Swaroop Pophale, Aaron Welch, University of Houston*

OpenSHMEM is a one-sided communication library API aimed at standardizing several vendor implementations of SHMEM. In this tutorial, we present an introductory course on the use of OpenSHMEM, its current state and the community's future plans. We will show how to use OpenSHMEM to add parallelism to programs via an exploration of its core features, to port sequential applications to run at scale while improving the program performance, and discuss how to migrate existing applications that use message passing techniques to equivalent OpenSHMEM programs that run more efficiently. Tips for porting programs using other existing flavors of SHMEM to portable OpenSHMEM programs will be given. The second part of the tutorial will focus on the plans for OpenSHMEM development, including a look at new PGAS run-time software called UCCS. UCCS is designed to sit underneath PGAS user-oriented libraries and languages such as OpenSHMEM, UPC, CAF and Chapel.

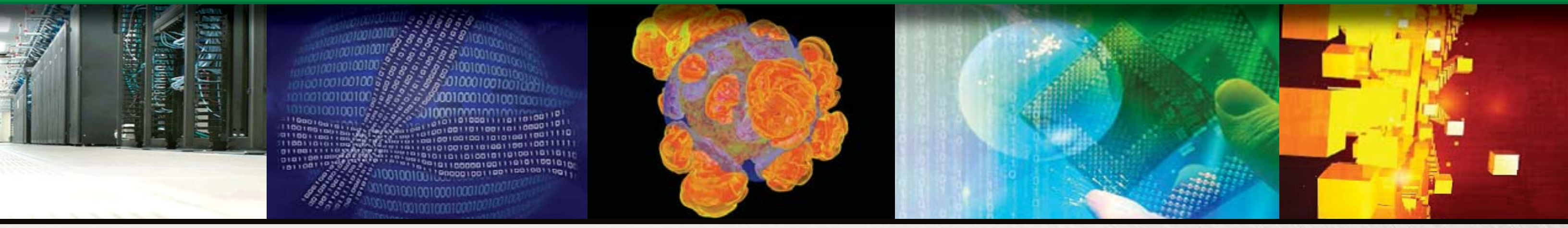
Accelerator Programming with OpenACC and OpenSHMEM Tutorial

Held in Governor Calvert Ballroom Center, located in the Governor Calvert House

Tutorial led by *Jean-Charles Vasnier, CAPS Enterprise*

This tutorial has been designed for those who are interested in porting their OpenSHMEM applications to a hardware accelerator, such as a GPU, using OpenACC. Following a mixture of lectures and demonstrations, we will explore the basic steps to port an application on the GPU. First, attendees will learn how to port a kernel on the GPU using directives. Then we see how to improve the overall performance of the application by reducing the data transfers between the host and the accelerators and by tuning the kernel.

Note: Breaks will be taken at the discretion of the instructor, and throughout the tutorial.
A working lunch is provided at 11:30 AM



1:00 PM Afternoon Tutorials

OpenSHMEM Tools Tutorial

Held in Governor Calvert Ballroom East, located in the Governor Calvert House

Tutorial led by *Nick Forrington, Allinea, Ltd.; Oscar Hernandez, Oak Ridge National Laboratory; Sameer Shende, ParaTools, Inc.; Frank Winkler, Technische Universität Dresden*

This tutorial will focus on the state-of-the-art of tools available for OpenSHMEM including a tutorial on program analysis, performance and debugging tools currently available for OpenSHMEM. We will also discuss the future roadmap to provide an integrated tools environment for OpenSHMEM. The tools that we will cover are: the OpenSHMEM Analyzer, TAU Performance Analysis tools, Vampir Tracing Tools, and DDT Debugger for OpenSHMEM.

TAU is a performance tool that provides portable profiling and tracing for OpenSHMEM applications. This tutorial provides hands-on exercises on how this tool integrates with OpenSHMEM.

Vampir is tool-set for performance analysis that traces events and identifies problems in HPC applications. It is the most scalable tracing analysis tool that can scale up-to several hundred thousand processes. It consists of the run-time measurement system VampirTrace and the visualization tools Vampir and VampirServer. In this tutorial, we will present how to use Vampir to trace OpenSHMEM applications at scale.

The DDT portion of the tutorial will cover the fundamentals of debugging multi-process OpenSHMEM programs with the Allinea DDT parallel debugging tool, and will include an introduction to the DDT user interface and how to start programs, as well as how to track down crashes and compare variables across processes.

The OpenSHMEM Analyzer is a compiler-based tool that can help users detect errors and provide useful analyses about their OpenSHMEM applications. In this tutorial we will show how the tool can be used to detect incorrect use of variables in OpenSHMEM calls, out-of-bounds checks for symmetric data, checks for incorrect initialization of pointers to non-symmetric data, and symmetric data alias information.

Verbs Programming Tutorial

Held in Governor Calvert Ballroom Center, located in the Governor Calvert

Tutorial led by *Dotan Barak, Mellanox Technologies*

This tutorial provides a basic overview of the InfiniBand technology and explain its advantages as a networking technology. Among others, this tutorial covers the following topics: various InfiniBand hardware and software components; explain how to utilize the InfiniBand technology for best performance; review the verbs API which is required for programming over InfiniBand; and finally it will provide several tips and tricks on verbs programming.

DAY 2

WEDNESDAY 05 MARCH

8:00 AM Registration Desk Opens (Atrium in the Governor Calvert House)

8:30 AM Welcome and Introductions (Working Breakfast)

Steve Poole, Oak Ridge National Laboratory

8:45 AM Future Technologies for InfiniBand

Presented by Richard Graham, Mellanox Technologies

The talk will provide a description of Mellanox's OpenSHMEM architecture, implementation, and benchmark results. It will also discuss specification issues, and suggestions for modifications to the specification.

9:35 AM The Evolution of the NVIDIA Compute Device Memory Model

Presented by Donald Becker and Duncan Poole, Nvidia Corporation

This talk will discuss the evolution of the NVIDIA compute device memory model from isolated address spaces on CPUs and compute devices towards a distributed universally addressable memory model. Leveraging commodity products has led to a series of design tradeoffs in the existing complex memory organization. We will discuss some of these limitations, and the steps NVIDIA envisions for simplifying the view from a large-system programmer's point of view. This must be accomplished while retaining the efficiency and performance required across a broad range of markets. While neither of the authors have a crystal ball, we can have a practical discussion of near term design options which might be addressed in OpenSHMEM.

10:30 AM OpenSHMEM on Portals

Presented by Keith Underwood, Intel Corporation

SHMEM originated in the context of a very specific hardware platform. Over the years, various SHMEM implementations have added features and/or tweaked semantics to match the capabilities of a different hardware platform. OpenSHMEM emerged to standardize those features and semantics, but retains characteristics that are heavily influenced by the platform of its birth. Portals 4 was designed to address the needs of both MPI and PGAS usage models. In the process, it focused on exposing building blocks that could be provided by hardware and minimizing the total software overhead. This presentation examines some of those features and how they influenced the design of Portals 4 and the resulting implications for hardware. Areas where modern hardware and software environments pose challenges are also discussed. Finally, there is a discussion of some aspects of the OpenSHMEM stack that could be evolved to improve its match to what hardware can provide.

11:50 AM **KEYNOTE:** Hybrid Programming Challenges for Extreme Scale Software (Working Lunch)

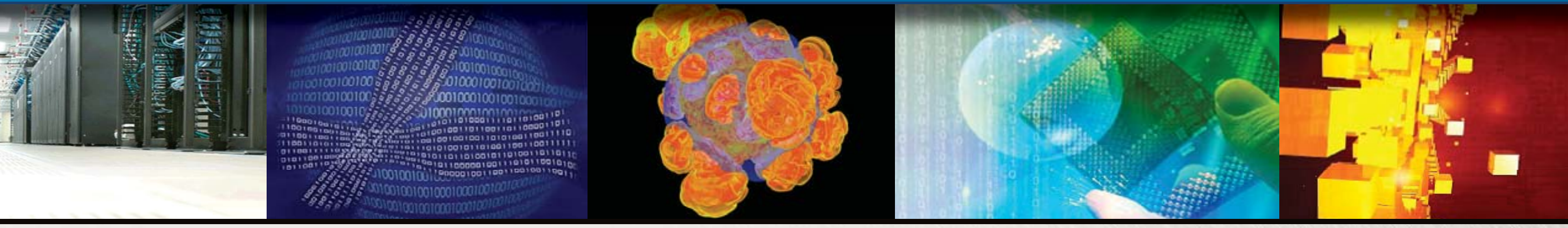
Presented by Vivek Sarkar, Rice University

In this talk, we summarize experiences with hybrid programming in the Habanero Multicore Software Research project [1] which targets a wide range of homogeneous and heterogeneous manycore processors in both single-node and cluster configurations. We focus on key primitives in the Habanero execution model that simplify hybrid programming, while also enabling a unified runtime system for heterogeneous hardware. Some of these primitives are also being adopted by the new Open Community Runtime (OCR) open source project [2]. These primitives have been validated in a range of applications, including medical imaging applications studied in the NSF Expeditions Center for Domain-Specific Computing (CDSC) [3].

Background material for this talk will be drawn in part from the DARPA Exascale Software Study report [4] led by the speaker. This talk will also draw from a recent (March 2013) study led by the speaker on Synergistic Challenges in Data-Intensive Science and Exascale Computing [5] for the US Department of Energy's Office of Science. We would like to acknowledge the contributions of all participants in both studies, as well as the contributions of all members of the Habanero, OCR, and CDSC projects.

REFERENCES:
[1] Habanero Multicore Software Research project. http://habanero.rice.edu.
[2] Open Community Runtime (OCR) open source project. https://01.org/projects/open-community-runtime.
[3] Center for Domain-Specific Computing (CDSC). http://cdsc.ucla.edu.
[4] DARPA Exascale Software Study report, September 2009. http://users.ece.gatech.edu/~mrichard/ExascaleComputingStudyReports/ECS_reports.htm.
[5] DOE report on Synergistic Challenges in Data-Intensive Science and Exascale Computing, March 2013. http://science.energy.gov/~/media/ascr/ascac/pdf/reports/2013/ASCAC_Data_Intensive_Computing_report_final.pdf.

Note: All presentations, with the exception of the Keynote, are 45 minutes with breaks between talks.



1:00PM Cray's OpenSHMEM activities & their proposal for thread-safe SHMEM extensions

Presented by Monika ten Bruggencate, Software Engineer at Cray, Inc.

This talk will give an overview of Cray's OpenSHMEM on-going activities and their planned support for thread-safety for Cray SHMEM on Cray XE and XC systems.

1:50PM MPI + X (OpenSHMEM?)

Presented by Michael Raymond, SGI Corporation

As the number of compute elements on a node increase, the HPC world has decided that the dominant programming model should be MPI between nodes and X within a node, where X might be OpenMP, pthreads, UPC, etc. What about OpenSHMEM? This talk will explore the implications of using OpenSHMEM as X, including the benefits and the weaknesses.

2:40PM Unified Common Communication Substrate (UCCS)

Presented by Pavel Shamis, Oak Ridge National Lab & Thomas Herault, University of Tennessee

Universal Common Communication Substrate (UCCS) is a low-level communication substrate that exposes high-performance communication primitives, while providing network interoperability. It is intended to support multiple upper layer protocols (ULPs) or programming models including SHMEM, UPC, Titanium, Co-Array Fortran, Global Arrays, MPI, GASNet, and File I/O. It provides various communication operations including one-sided and two-sided point-to-point, collectives, and remote atomic operations. In addition to operations for ULPs, it provides an out-of-band communication channel required typically required to wire-up communication libraries.

3:35PM Future Technologies for AMD

Presented by Vinod Tipparaju, Advanced Micro Devices, Inc.

This talk introduces HSA and discusses how HSA simplifies the use of accelerators by supporting unified programming models. HSA enhances support for symmetric memory in the context of submitting work to the accelerators. This talk will discuss HSAs support for asynchronous functions, function closures and lambda functions which enables support for various programming models and languages.

4:30PM IBM OpenSHMEM Implementation over the Parallel Active Message Interface (PAMI)

Presented by Alan Benner, IBM Systems and Technology Group

For the DARPA HPCS project, IBM created a highly flexible communications protocol called the Parallel Active Message Interface (PAMI). It combines the advantages and features of Blue Gene's Deep Computing Message Framework (DCMF) and IBM Parallel Environment's Low-Level Application Programming Interface (LAPI). It also serves as a common communications layer for various IBM message passing API's, such as PEMPI and MPICH2, as well as several PGAS programming models, including UPC, X10, and OpenSHMEM. PAMI provides flexibility for protocols by providing an implementation for different IBM hardware platforms, such as IBM Blue Gene, Power Systems, and System x. IBM OpenSHMEM is one of the communications programming models that is implemented over PAMI. In this talk, I will present the background and basics of PAMI, how the OpenSHMEM function is neatly mapped to its PAMI counterpart, and a high level description of the design concepts.

5:30PM HIPATIA Birds Of A Feather Session

Presented by Josh Lothian, Jonathan Schrock, & Sarah Powers, Oak Ridge National Laboratory

HIPATIA (High Performance Adaptive Integrated Linear Algebra Benchmark) is a next-generation benchmark that is easily extensible while providing access to power metrics and CPU counters. Unlike many of the more popular benchmarks today, HIPATIA's initial focus is on solving sparse matrices within the integer domain using GMP. In addition to sparse, integer matrices, HIPATIA will be configurable for computation on real, complex, or fixed-point values, in dense or sparse matrix formats. We intend HIPATIA to adapt to many different usage scenarios that are not currently well represented in existing benchmarks. We will discuss current progress of HIPATIA development, as well as future development plans.

DAY 3

THURSDAY 06 MARCH

8:00 AM OpenSHMEM Implementations and Evaluation Session

Designing a High Performance OpenSHMEM Implementation using Universal Common Communication Substrate as a Communication Middleware

Presented by Pavel Shamis, Oak Ridge National Laboratory

OpenSHMEM is an effort to standardize the well-known SHMEM parallel programming library. The project aims to produce an open-source and portable SHMEM API and is led by ORNL and UH. In this paper, we optimize the current OpenSHMEM reference implementation, based on GASNet, to achieve higher performance characteristics. To achieve these desired performance characteristics, we have redesigned an important component of the OpenSHMEM implementation, the network layer, to leverage a low-level communication library designed for implementing parallel programming models called UCCS. In particular, UCCS provides an interface and semantics such as native atomic operations and remote memory operations to better support PGAS programming models, including OpenSHMEM. Through the use of microbenchmarks, we evaluate this new OpenSHMEM implementation on various network metrics, including the latency of point-to-point and collective operations. Furthermore, we compare the performance of our OpenSHMEM implementation with the state-of-the-art SGI SHMEM. Our results show that the atomic operations of our OpenSHMEM implementation outperform SGI's SHMEM implementation by 3%. Its RMA operations outperform both SGI's SHMEM and the original OpenSHMEM reference implementation by as much as 18% and 12% for gets, and as much as 83% and 53% for puts.

Implementing OpenSHMEM using MPI-3 One-sided Communication

Presented by Jeff Hammond; Sayan Ghosh, University of Houston

This paper reports the design and implementation of OpenSHMEM over MPI using new one-sided communication features in MPI-3, which include not only new functions (e.g. remote atomics) but also a new memory model that is consistent with that of SHMEM. We use a new, non-collective MPI communicator creation routine to allow SHMEM collectives to use their MPI counterparts. Finally, we leverage MPI shared memory windows within a node, which allows direct (load-store) access. Performance evaluations are conducted for shared-memory and InfiniBand conduits using microbenchmarks.

A Comprehensive Performance Evaluation of OpenSHMEM Libraries on InfiniBand Clusters

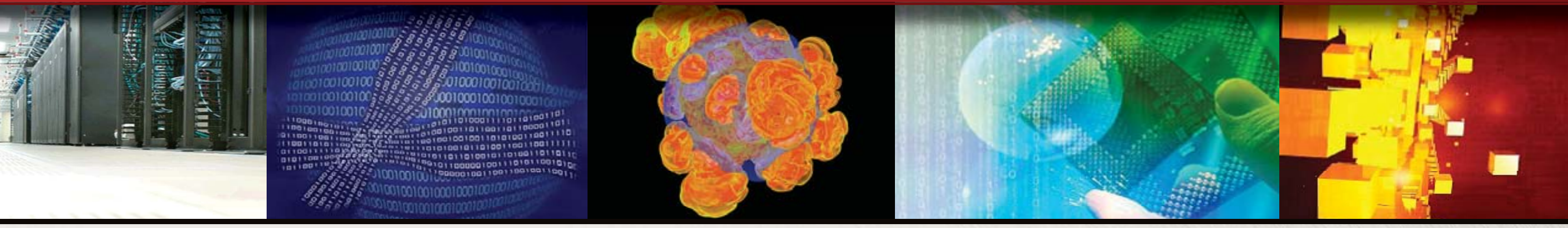
Presented by Jithin Jose, Ohio State University

OpenSHMEM is an open standard that brings together several long-standing vendor-specific SHMEM implementations and allows applications to use SHMEM in a platform-independent fashion. Several implementations of OpenSHMEM have become available on clusters interconnected by InfiniBand networks, which has gradually become the de facto high performance network interconnect standard. In this paper, we present a detailed comparison and analysis of the performance of different OpenSHMEM implementations, using micro-benchmarks and application kernels. This study, done on TACC Stampede system using up to 4,096 cores, provides a useful guide for application developers to understand and contrast various implementations and to select the one that works best for their applications.

Analyzing the Energy and Power Consumption of Remote Memory Accesses in the OpenSHMEM Model

Presented by Siddhartha Jana, University of Houston

PGAS models like OpenSHMEM provide interfaces to explicitly initiate one-sided remote memory accesses among processes. In addition, the model also provides synchronizing barriers to ensure a consistent view of the distributed memory at different phases of an application. The incorrect use of such interfaces affects the scalability achievable while using a parallel programming model. This study aims at understanding the effects of these constructs on the energy and power consumption behavior of OpenSHMEM applications. Our experiments show that the cost incurred in terms of the total energy and power consumed depends on multiple factors across the software and hardware stack. We conclude that there is a significant impact on the power consumed by the CPU and DRAM due to multiple factors including the design of the data transfer patterns within an application, the design of the communication protocols within a middleware, the architectural constraints laid by the interconnect solutions, and the levels of memory hierarchy within a compute node. This work motivates treating energy and power consumption as important factors while designing for current and future distributed systems.



Benchmarking Parallel Performance on Many-Core Processors

Presented by Bryant Lam, University of Florida

With the emergence of many-core processor architectures onto the HPC scene, concerns arise regarding the performance and productivity of numerous existing parallel-programming tools, models, and languages. As these devices begin augmenting conventional distributed cluster systems in an evolving age of heterogeneous supercomputing, proper evaluation and profiling of many-core processors must occur in order to understand their performance and architectural strengths with existing parallel-programming environments and HPC applications. This paper presents and evaluates the comparative performance between two many-core processors, the Tilera TILE-Gx8036 and the Intel Xeon Phi 5110P, in the context of their applications performance with the SHMEM and OpenMP parallel-programming environments. Several applications written or provided in SHMEM and OpenMP are evaluated in order to analyze the scalability of existing tools and libraries on these many-core platforms. Our results show that SHMEM and OpenMP parallel applications scale well on the TILE-Gx and Xeon Phi, but heavily depend on optimized libraries and instrumentation.

Hybrid Programming using OpenSHMEM and OpenACC

Presented by Matthew Baker, Oak Ridge National Laboratory

With high performance systems exploiting multicore and accelerator-based architectures on a distributed shared memory system, heterogeneous hybrid programming models are the natural choice to exploit all the hardware made available on these systems. Previous efforts looking into hybrid models have primarily focused on using OpenMP directives (for shared memory programming) with MPI (for inter-node programming on a cluster), using OpenMP to spawn threads on a node and communication libraries like MPI to communicate across nodes. As accelerators get added into the mix, and there is better hardware support for PGAS languages/APIs, this means that new and unexplored heterogeneous hybrid models will be needed to effectively leverage the new hardware. In this paper we explore the use of OpenACC directives to program GPUs and the use of OpenSHMEM, a PGAS library for one-sided communication between nodes. We use the NAS-BT Multi-zone benchmark that was converted to use the OpenSHMEM library API for network communication between nodes and OpenACC to exploit accelerators that are present within a node. We evaluate the performance of the benchmark and discuss our experiences during the development of the OpenSHMEM+OpenACC hybrid program.

11:30 AM **OpenSHMEM Tools Session** (Working Lunch)

Profiling Non-Numeric OpenSHMEM Applications with the TAU Performance System

Presented by John Linford and Tyler Simon, ParaTools, Inc.

The recent development of a unified SHMEM framework, OpenSHMEM, has enabled further study in the porting and scaling of applications that can benefit from the SHMEM programming model. This paper focuses on non-numerical graph algorithms, which typically have a low FLOPS/byte ratio. An overview of the space and time complexity of Kruskal's and Prim's algorithms for generating a minimum spanning tree (MST) is presented, along with an implementation of Kruskal's algorithm that uses OpenSHMEM to generate the MST in parallel without intermediate communication. Additionally, a procedure for applying the TAU Performance System to OpenSHMEM applications to produce in depth performance profiles showing time spent in code regions, memory access patterns, and network load is presented. Performance evaluations from the Cray XK7 "Titan" system at Oak Ridge National Laboratory and a 48 core shared memory system at University of Maryland, Baltimore County are provided.

DAY 3

THURSDAY 06 MARCH

12:00 PM OpenSHMEM Tools Session (continued)

Towards Parallel Performance Analysis Tools for the OpenSHMEM Standard

Presented by Andreas Knüpfer, Technische Universität Dresden

This paper discusses theoretical and practical aspects when extending performance analysis tools to support the OpenSHMEM standard for parallel programming. The theoretical part covers the mapping of OpenSHMEM's communication primitives to a generic event record scheme that is compatible with a range of PGAS libraries. The visualization of the recorded events is included as well. The practical parts demonstrate an experimental extension for Cray-SHMEM in Vampir-Trace and Vampir and the first results with a parallel example application. Since Cray-SHMEM is similar to OpenSHMEM in many respects, this serves as a realistic preview. Finally, an outlook on a native support for OpenSHMEM is given together with some recommendations for future revisions of the OpenSHMEM standard from the perspective of performance tools.

Extending the OpenSHMEM Analyzer to Perform Synchronization and Multi-Valued Analysis

Presented by Swaroop Prophale, University of Houston

OpenSHMEM Analyzer (OSA) is a compiler-based tool that provides static analysis for OpenSHMEM programs. It was developed with the intention of providing feedback to the users about semantics errors due to incorrect use of the OpenSHMEM API in their programs, thus making development of OpenSHMEM applications an easier task for beginners as well as experienced programmers. In this paper we discuss the improvements to the OSA tool to perform parallel analysis to detect the collective synchronization structure of a program. Synchronization is a critical aspect of all programming models and in OpenSHMEM it is the responsibility of the programmer to introduce synchronization calls to ensure the completion of communication among processing elements (PEs) to prevent use of old/incorrect data, avoid deadlocks and ensure data race free execution and keeping in mind the semantics of OpenSHMEM library specification.

A Global View Programming Abstraction for Transitioning MPI Codes to PGAS Languages

Presented by Tiffany Mitnz, Oak Ridge National Laboratory

The multicore generation of scientific high performance computing has provided a platform for the realization of Exascale computing, and has also underscored the need for new paradigms in coding parallel applications. The current standard for writing parallel applications requires programmers to use languages designed for sequential execution. These languages have abstractions that only allow programmers to operate on the process centric local view of data. To provide suitable languages for parallel execution, many research efforts have designed languages based on the Partitioned Global Address Space (PGAS) programming model. Chapel is one of the more recent languages to be developed using this model. Chapel supports multithreaded execution with high-level abstractions for parallelism. With Chapel in mind, we have developed a set of directives that serve as intermediate expressions for transitioning scientific applications from languages designed for sequential execution to PGAS languages like Chapel that are being developed with parallelism in mind.

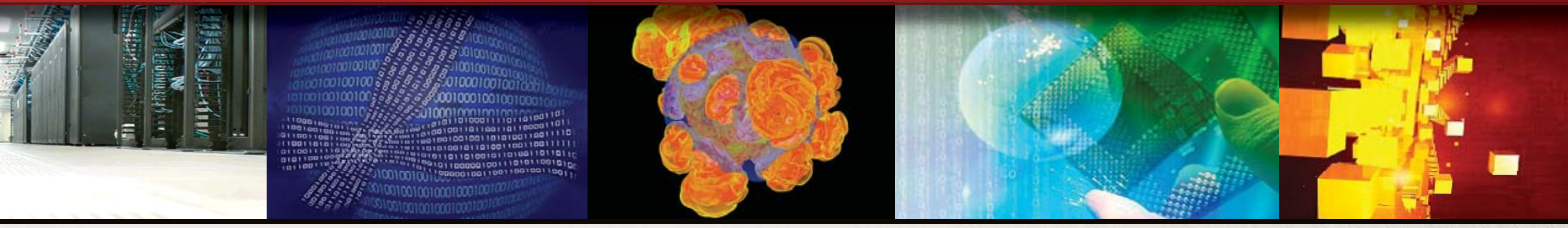
1:30 PM OpenSHMEM Extensions Session

Parallel I/O for OpenSHMEM

Presented by Edgar Gabriel, University of Houston

This talk discusses the necessity of I/O interfaces in any parallel programming model for the next generation of high end systems. Some suggestions for parallel I/O interfaces for OpenSHMEM will be presented based on the experience of the MPI I/O interfaces and some recent work on parallel I/O for OpenMP.

Note: All talks are 20 minutes with 5 minutes following for Questions.
At the conclusion of each presentation, there will be a 5 minute break.



Reducing Synchronization Overhead Through Bundled Communication

Presented by James Dinan, Intel Corporation

OpenSHMEM provides a one-sided communication interface that allows for asynchronous, one-sided communication operations on data stored in a partitioned global address space. While communication in this model is efficient, synchronizations must currently be achieved through collective barriers or one-sided updates of sentinel locations in the global address space. These synchronization mechanisms can over synchronize, or require additional communication operations, respectively, leading to high overheads. We propose a SHMEM extension that utilizes capabilities present in most high performance interconnects (e.g. communication events) to bundle synchronization information together with communication operations. Using this approach, we improve ping pong latency for small messages by a factor of two, and demonstrate significant improvement to synchronization-heavy communication patterns, including all-to-all and pipelined parallel stencil communication.

Implementing Split-Mode Barriers in OpenSHMEM

Presented by Michael Raymond, SGI Corporation

Barriers synchronize the state of many processing elements working in parallel. No worker may leave a barrier before all the others have arrived. High performance applications hide latency by keeping a large number of operations in progress asynchronously. Since barriers synchronize all these operations, maximum performance requires that barriers have as little overhead as possible. When some workers arrive at a barrier much later than others, the early arrivers must sit idle waiting for them. Split-mode barriers provide barrier semantics while also allowing the early arrivers to make progress on other tasks. In this paper we describe the process and several challenges in developing split-mode barriers in the OpenSHMEM programming environment.

OpenSHMEM Extensions and a Vision for its Future Direction

Presented by Pavel Shamis, Oscar Hernandez, Greg Koenig, Oak Ridge National Laboratory

The Extreme Scale Systems Center (ESSC) at Oak Ridge National Laboratory (ORNL), together with the University of Houston, led the effort to standardize the SHMEM API with input from the vendors and user community. In 2012, OpenSHMEM Specification 1.0 was finalized and released to the OpenSHMEM community for comments. As we move to future HPC systems, there are several shortcomings in the current specification that we need to address to ensure scalability, higher degrees of concurrency, locality, thread safety, fault-tolerance, I/O, etc. In this paper we discuss an immediate set of extensions that we propose to the current API and our vision for a future API, OpenSHMEM Next-Generation (NG), that targets future Exascale systems. We also explain our rational for the proposed extensions and highlight the lessons learned from other PGAS languages and communication libraries.

3:30PM **Panel Discussion**

The Future of OpenSHMEM

Moderator: Steve Poole, Oak Ridge National Laboratory

Panelists: Monika ten Bruggencate, Cray, Inc
Gary Grider, Los Alamos National Laboratory
Oscar Herandez, Oak Ridge National Laboratory
Nick Park, Department of Defense
Michael Raymond, SGI
Pavel Shamis, Oak Ridge National Laboratory

5:00PM **The 2013 OpenSHMEM Workshop Closes**

INVITED SPEAKER BIOGRAPHIES

DOTAN BARAK

MELLANOX TECHNOLOGIES

Dotan Barak is a Senior Software Manager at Mellanox Technologies working on RDMA Technologies. He has worked at Mellanox for more than 10 years in various roles, both as a developer and a manager. He has been involved in several documentation projects for InfiniBand, including the man pages for libibverbs. Check out his blog on the RDMA technology: http://www.rdmamojo.com.

DONALD BECKER

NVIDIA

NVIDIA's Donald Becker has a long history in compilers, device drivers, networking, parallel computing and cluster computing. He co-founded the Beowulf Project at NASA's Goddard Space Flight Center in 1994 and received the Gordon Bell Prize in 1997. He was a major developer of the early Linux kernel networking subsystem, writing essentially all of the network device drivers through 2000.

ALAN BENNER

IBM, INC.

Alan is a Sr. Technical Staff Member at IBM Systems and Technology Group, doing system architecture, design, development, and manufacturing of optical and electronic networks for high-performance servers and supercomputers since 1992. Dr. Benner worked at AT&T Bell Laboratories before joining CU Boulder's Optoelectronic Computing Systems Center, receiving M.S./ Ph.D. degrees in 1990/1992. He has over two dozen technical publications, including books and book chapters on Fibre Channel, optical interconnect packaging, and specifications for the InfiniBand architecture, plus over 45 issued patents in the U.S. and other countries.

MONIKA TEN BRUGGENCATE

CRAY, INC.

Monika ten Bruggencate received her Ph.D. in computer engineering from the University of Wisconsin-Madison. Since that time, she has worked in the HPC industry with a focus on software for high performance networks. She is currently employed by Cray Inc., where she is responsible for design and development of components of the network software stack with a focus on one-sided Program Models and their performance.

TONY CURTIS

UNIVERSITY OF HOUSTON

Tony is a research engineer in Computer Science at the University of Houston. He has been involved in high performance computing research and UNIX/Linux systems support for more than 20 years in Europe, the U.S. and Canada. He's currently the lead on the OpenSHMEM project.

NICK FORRINGTON

ALLINEA SOFTWARE

Nick Forrington is a software developer for Allinea Software, working with users on site at Oak Ridge National Laboratory. Since earning his Masters degree in Computer Science from the University of Warwick, UK, he has worked extensively on Allinea's parallel debugging tool, DDT, which allows users to manage the complexity associated with debugging issues across multiple processing elements.

RICHARD GRAHAM

MELLANOX TECHNOLOGIES

Dr. Richard Graham is a Senior Solutions Architect at Mellanox Technologies, Inc. His primary focus is on the High Performance Computing market, working on OFED and communication middleware architecture issues, as they relate to extreme-scale computing. Prior to moving to Mellanox, Rich spent thirteen years at Los Alamos National Laboratory and Oak Ridge National Laboratory, in computer science technical and administrative roles, with a technical focus on communication libraries and application analysis tools.

EDGAR GABRIEL

UNIVERSITY OF HOUSTON

Edgar Gabriel is an associate professor in the Department of Computer Science at the University of Houston. He received his PhD from the University of Stuttgart, Germany in 2002. His research interest are in High Performance Computing, Parallel I/O and message passing systems. Gabriel received the NSF Early CAREER Award in 2009, and was an early contributor to the popular Open MPI library.

OSCAR HERNANDEZ

OAK RIDGE NATIONAL LABORATORY

Oscar is a member of the Computer Science and Mathematics Division at Oak Ridge National Laboratory. His research focus has been on languages, compilers, static tools performance tools integration, and optimization techniques for parallel languages, especially for OpenSHMEM, OpenMP, and accelerator programming. He is responsible for the programming environment of Titan and future Oak Ridge leadership class systems. Oscar represents ORNL at the OpenACC, OpenMP ARB and OpenSHMEM organizations. He graduated from the University of Houston with a Ph.D. and Msc. degree in the area of compilers and high performance computing.

THOMAS HERAULT

UNIVERSITY OF TENNESSEE

Thomas Herault is a Research Scientist at the Innovative Computing Laboratory, University of Tennessee. He received his Ph.D. from the University of Paris-Sud in 2003, working on resilience in self-stabilizing systems. His research interests include fault tolerance, high-performance computing and distributed algorithms. He contributes to several widely distributed open source software, applying his research to the service of a more efficient, and reliable, usage of high performance computers. He coauthored more than sixty papers in international conferences, and journals

DUNCAN POOLE

NVIDIA CORPORATION

Duncan Poole is responsible for strategic partnerships for NVIDIA's Accelerated Computing Division. His responsibilities reach across the developer tool chain to drive successful partnerships where engineering interfaces are adopted by external parties building tools for accelerated computing. This includes open standards, compilers, profilers, debuggers, performance analysis tools, compute and communications libraries. He is responsible for NVIDIA's partnerships for MPI and their adoption of GPU Direct RDMA, especially in partnership with Mellanox. Duncan is also the president of OpenACC, and responsible for NVIDIA's membership in OpenMP.

SWAROOP PROPHALE

UNIVERSITY OF HOUSTON

Swaroop Pophale is a doctoral candidate in the Computer Science Department at the University of Houston. She has been involved in high performance computing research and the OpenSHMEM project for the last four years.

MICHAEL RAYMOND

SGI CORPORATION

Michael Raymond is a software developer specializing in communications middleware. He joined TimeSys in 2000, scaling their real-time Linux kernel product. In 2001, he went to work for SGI doing scalability and real-time development in the IRIX kernel. He moved to the SGI Message Passing Toolkit team in 2007 and now leads the group. Michael has a bachelor's in electrical & computer engineering from Carnegie Mellon University, a master's in computer science from the University of Minnesota, and a masters in intelligence studies from American Military University.

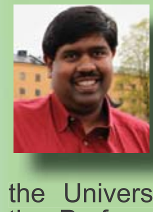
PAVEL SHAMIS

OAK RIDGE NATIONAL LABORATORY

Pavel Shamis is a scientist at Oak Ridge National Laboratory, focuses on a high-performance communication middleware and collective communication. Prior to joining Oak Ridge National Laboratory, Mr. Shamis spent ten years at Mellanox Technologies in different technical roles, including Senior Software Developer and HPC Team Leader.

SAMEER SHENDE

PARATOOLS, INC.



Sameer Shende received his Ph.D. from the University of Oregon in 2001 in Computer & Information Science and his Bachelor of Technology from the Indian Institute of Technology, Bombay in 1991. He has helped develop the TAU Performance System, and the Program Database Toolkit projects at the University of Oregon. He serves as the Director of the Performance Research Laboratory at the University of Oregon and the President of ParaTools, Inc

VINOD TIPPARAJU

ADVANCED MICRO DEVICES, INC.

Vinod Tipparaju is a Principal Member of Technical Staff at Advanced Micro Devices, he is the architect of the HSA runtime and spec editor of the HSA runtime specification for HSA Foundation. Vinod works on various low-level issues and designs abstractions to expose GPU and MMU hardware features to users. Prior to AMD, Vinod worked as a HPC Scientist at Oak Ridge National Laboratory and at Pacific Northwest National Laboratory. He has published over 50 papers in these areas and has contributed to several books

KEITH UNDERWOOD

INTEL CORPORATION

Keith received his BS and PhD in Computer Engineering from Clemson University, where he worked on using FPGAs to process the network data stream in Beowulf Clusters. He worked at Sandia National Labs, where his interest turned to "real HPC networks" as part of the Red Storm project. Next, Keith started research efforts into network interface architectures. He was a part of a team that co-designed the next generation of the Portals Network API with architectural building blocks that could be implemented in hardware, with a focus on improving support for PGAS. His research into HPC oriented network architectures continued after he joined Intel Corporation.

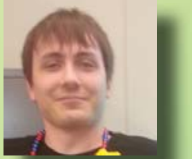
JEAN-CHARLES VASNIER

CAPS ENTERPRISE

Jean-Charles Vasnier is Sales engineer at CAPS enterprise. Built on over seven years of advanced research and development, CAPS provides high quality and cost effective programming tools that leverage the computing power of evolving manycore hybrid platforms. Jean-Charles has been involved with general computing on graphics processor units from it early time. He was working on-site at ORNL to help the Titan users to leverage better performance with their application using OpenACC directives. He is now the sales representative for CAPS enterprise in America.

AARON WELCH

UNIVERSITY OF HOUSTON



Aaron is a graduate student at the University of Houston. He has two years of experience in the high performance computing field and has been actively working on integrating UCCS with OpenSHMEM reference library. He is also involved in new ideas and extensions for OpenSHMEM.

FRANK WINKLER

TECHNISCHE UNIVERSITÄT DRESDEN

Frank Winkler's work primary focus is on performance analysis of HPC applications. He is a member of the development team of the interactive trace analysis tool Vampir. He gained additional experiences with other performance analysis tools like Cube and Scalasca as well as with the corresponding measurement systems VampirTrace and Score-P.

HISTORIC INNS OF ANNAPOLIS
Est. 1776

Lodging Information

Step back in time to colonial 18th century America and experience the most historic lodgings in Annapolis, Maryland. Located in the heart of downtown Annapolis, the Historic Inns of Annapolis is comprised of three unique hotels that date back to the 18th century. We are just steps away from the water and Naval Academy, and right in the midst of an abundance of activities, restaurants, pubs and boutiques.

The Historic Inns of Annapolis is comprised of three buildings; the Governor Calvert House, the Robert Johnson House and the Maryland Inn. All three Inns overlook the State Capitol and are within a 4 minute walk of each other. Historic Inns of Annapolis hotels are located in the Annapolis Historic District and are steps from the State House, US Naval Academy, Chesapeake Bay and St. Johns College. Historic Inns of Annapolis hotels are entirely smoke-free.

The Maryland Inn - The Maryland Inn is the crown jewel of the Historic Inns of Annapolis portfolio. Presidents, statesmen, and political dignitaries have enjoyed the comforts this historic Annapolis lodging since the late 1700s. This charming 44-room boutique hotel is beautifully restored with Victorian-era reproductions and elegant décor. The Maryland Inn features views of the waterfront, Main Street, and the State House.

Governor Calvert House - Housed in one of the oldest historic buildings in Annapolis, the 51-room Governor Calvert House offers a refreshing blend of history and modern luxury. This charming boutique lodging in the Annapolis Historic District was the residence of two former Maryland governors. Enjoy beautiful Annapolis accommodations where history and hospitality make you feel relaxed and comfortable.

Robert Johnson House - Overlooking the State House and Governor's Mansion, the 29-room Robert Johnson House was built in 1773. Elegantly-appointed accommodations in the heart of downtown Annapolis are furnished with beautiful 19th century antiques and period reproductions.

A block of rooms for March 3 through 7, 2014, will be held until February 2, 2014, at the government rate of $101 plus tax. After February 2, rooms may not be available at the government rate. Rooms must be guaranteed via a major credit card. You may reserve by calling 1-800-847-8882 and referencing group name SHMEM or going to http://www.csm.ornl.gov/workshops/openshmem2013/lodging.html and clicking the link there.

Logistics

http://www.csm.ornl.gov/workshops/openshmem2013

The Workshop

The workshop will be held at the Historic Inn's Governor Calvert House Ballroom and Atrium located at 58 State Cir, Annapolis, MD 21401.

Transportation Information

Transportation options coming from Baltimore/Washington International Airport (BWI) can be located at:
http://www.bwiairport.com/en/travel/ground-transportation

Transportation options coming from Reagan National Airport (DCA) can be located at:
http://www.metwashairports.com/reagan/1179.htm

Driving Directions to Historic Inns of Annapolis

Getting to the hotels in historic downtown Annapolis is easy from three major airports, where you can easily find a taxi or shuttle to the Governor Calvert House, where you can check in.

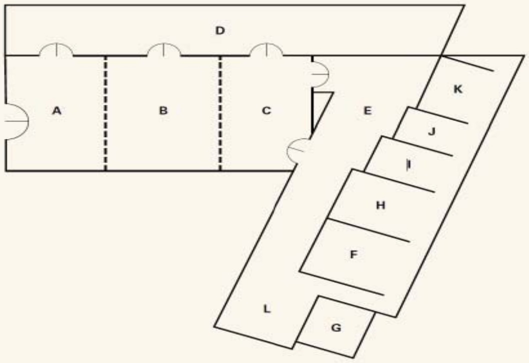
From Baltimore/ Washington International (BWI) -
Follow Interstate 97 South to Route 50 East (approx. 20 miles). Take Rt. 50 east, to exit 24 Rowe Blvd. Follow Rowe Blvd. towards downtown historic Annapolis. At the end of Rowe Blvd. you will turn right onto College Ave. Follow College Ave. to Church Circle. Go around the circle and make a right onto School St. Follow School St. to State Circle and make a right. Follow State Circle halfway around to the Governor Calvert House (58 State Circle).

From Dulles International Airport -
Follow Virginia 267 East to Interstate 495 South (approx. 5 miles). Follow Interstate 495 South (Capital Beltway) towards Route 50 East (approx. 30 miles). Take Rt. 50 east, to exit 24 Rowe Blvd. Follow Rowe Blvd. towards downtown historic Annapolis. At the end of Rowe Blvd. you will turn right onto College Ave. Follow College Ave to Church Circle. Go around the circle and make a right onto School St. Follow School St. to State circle and make a right. Follow State Circle halfway around to the Governor Calvert House (58 State Circle).

From Reagan International/Arlington -
Follow Interstate 395 North to Interstate 295 South (approx. 2 miles). Follow Interstate 295 (approx. 1 mile) to Pennsylvania Ave. Follow Pennsylvania Ave. for a mile to DC-295 North. Follow DC-295 North to Route 50 East (approx. 5 miles). Take Rt. 50 East, follow to exit 24 Rowe Blvd. Follow Rowe Blvd. towards downtown historic Annapolis. At the end of Rowe Blvd. you will turn right onto College Ave. to Church Circle. Go around the circle and make a right onto School St. Follow School St. to State Circle and make a right. Follow State Circle halfway around to the Governor Calvert House (58 State Circle).

GOVERNOR CALVERT HOUSE

The Governor Calvert House provides meeting rooms for groups of 10 to 400 with more than 7,800 square feet of flexible function and banquet space.



The Extreme Scale Systems Center (ESSC) is a program within the Computer Science & Mathematics Division at the Oak Ridge National Laboratory and is supported by the Department of Defense.