

# 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.



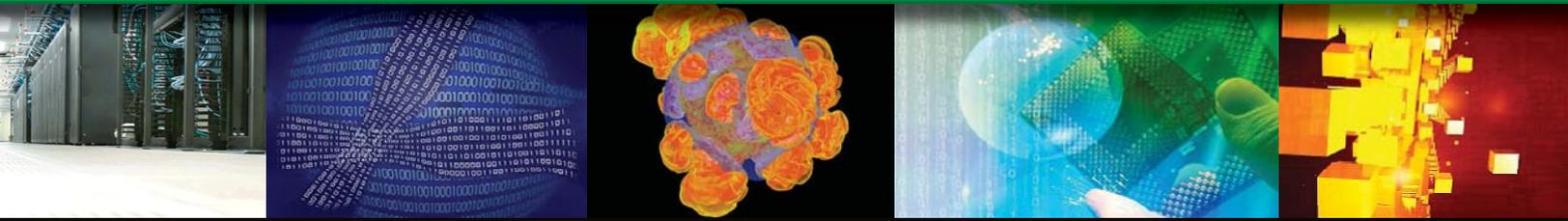
## ACCELERATOR PROGRAMMING WITH OPENACC AND OPENSHMEM TUTORIAL

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.





## 1:00 PM AFTERNOON TUTORIALS

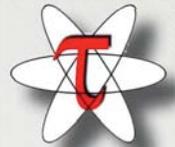
### OPENSHMEM TOOLS TUTORIAL

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

Tutorial led by *Nick Forrington, Alinea, 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 Alinea 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.



# AGENDA

## TUTORIALS

**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.

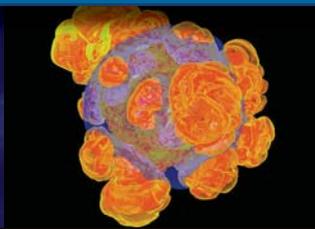
**11:50 AM** KEYNOTE: HYBRID PROGRAMMING CHALLENGES FOR EXTREME SCALE SOFTWARE  
*Presented by Vivek Sarkar, Rice University*

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](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](http://science.energy.gov/~media/ascr/ascac/pdf/reports/2013/ASCAC_Data_Intensive_Computing_report_final.pdf).



## 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?) (WORKING LUNCH)**

*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 PEMP1 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.

# AGENDA

## INDUSTRY PARTNERS



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

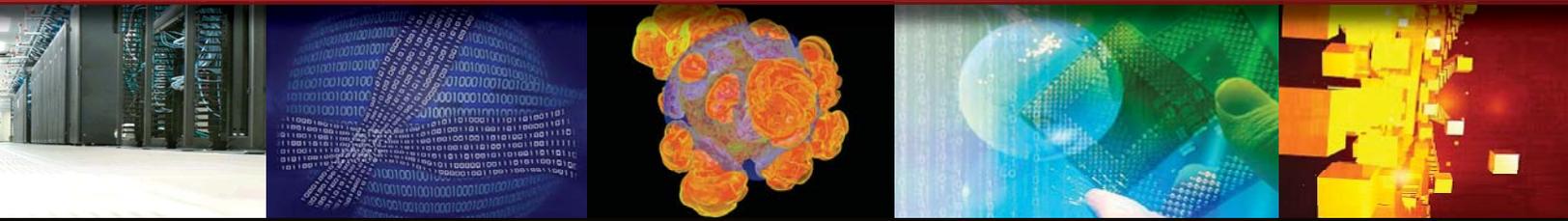
*Presented by Jithin Jose, Ohio State University*

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 Tiler 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.

**AGENDA**  
**CONTRIBUTED PAPER SESSIONS**



### 12:00PM **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*

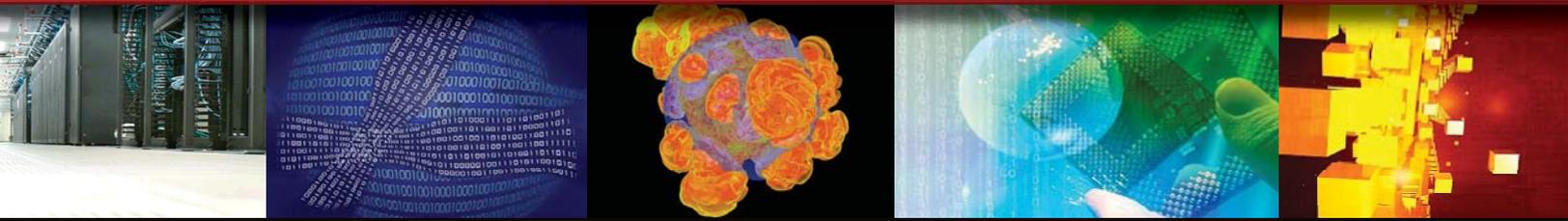
The multicore generation of scientific high performance computing has provided a platform for the realization of Exascale computing, and has also underscored the need for new paradigms in coding parallel applications. The current standard for writing parallel applications requires programmers to use 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.



## 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 OPENSHEM

*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.

## OPENSHEM 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 rationale for the proposed extensions and highlight the lessons learned from other PGAS languages and communication libraries.

**3:30PM**

### PANEL DISCUSSION

#### THE FUTURE OF OPENSHEM

*Moderator: Steve Poole, Oak Ridge National Laboratory*

Panelists: Monika ten Bruggencate, Cray, Inc

Gary Grider, Los Alamos National Laboratory

Oscar Hernandez, Oak Ridge National Laboratory

Nick Park, Department of Defense

Michael Raymond, SGI

Pavel Shamis, Oak Ridge National Laboratory

**5:00PM** THE 2013 OPENSHEM WORKSHOP CLOSES

# AGENDA

## CONTRIBUTED PAPER SESSIONS