13th Annual Workshop Abstracts

March 27-31, 2017

Hyatt Regency Austin
Austin, TX

Keynote

Exascale Computing Project - Driving a HUGE Change in a Changing World

Al Geist, Oak Ridge National Laboratory

Future DOE supercomputers will need to solve emerging data science and machine learning problems in addition to the traditional modeling and simulations applications. In August 2016, the Exascale Computing Project (ECP) was approved to make a huge lift of all U.S. High Performance Computing (HPC) to a new trajectory. The ECP goals are to enable the delivery of capable exascale computers in 2022 and delivery of one early exascale system in 2021, to foster a rich exascale ecosystem, and to help ensure continued U.S. HPC leadership. This talk will describe how the ECP plans to achieve these goals and the positive impacts it may have on the OpenFabrics Alliance.

Accelerators, FPGAs, GPUs

Asynchronous Peer-to-peer Device Communication

Feras Daoud, Mellanox

Peer-to-peer communication between devices, such as RDMA HCAs and GPUs or other accelerators, offer a great opportunity to improve performance and reduce CPU utilization. Data coming in-from/out-of the network can be DMA'd directly to/from accelerator memory. This removes the need to stage data in host memory and to manage its transfer by the CPU.

However, even when peer-to-peer DMA is used, the CPU must still coordinate between communication and the accelerator computation. For example, the CPU must wait until incoming data transfers have completed before invoking the GPU kernel. Similarly, the CPU must detect the completion of a GPU kernel before posting a request to send out the data. This coordination adds significant latency when shipping data for remote processing. In addition, the CPU can become the bottleneck when there are multiple outstanding asynchronous jobs.

Asynchronous peer-to-peer communication solves this problem by moving the coordination task to the accelerator. While the CPU sets up connections, posts buffers, and sets up data transfers, the accelerator directly polls for completions to detect when data has arrived, and triggers pre-posted sends or RDMA transfers when computation has finished. This results in up to 40% improvement in performance.

Communications Middleware

Advanced PGAS Centric Usage of the OpenFabrics Interface (OFI)

Erik Paulson, Intel; Sayanthan Sur, Intel; Kayla Seager, Intel

The OpenFabrics Interface was derived using communication requirements from PGAS libraries such as OpenSHMEM and GASNet. Latest releases of libfabric have continued to improve the performance and scalability of the PGAS libraries on underlying fabrics such as the Aries and Omni-Path Interconnects. Further, OFI 1.5 interface was recently announced that contain several new API and ABI updates.

In this talk we describe the performance characteristics and semantic mapping for both GASNet and SHMEM libraries. We also describe how the new PGAS APIs are evolving and their continued semantic mapping to the OpenFabrics Interfaces.

Building Efficient HPC Clouds with MVAPICH2 and RDMA-Hadoop over SR-IOV enabled InfiniBand Clusters

Xiaoyi Lu, The Ohio State University; Dhabaleswar Panda, The Ohio State University

Single Root I/O Virtualization (SR-IOV) technology has been steadily gaining momentum for high-performance interconnects such as InfiniBand. SR-IOV can deliver near native performance but lacks locality-aware communication support. This talk presents an efficient approach to building HPC clouds based on MVAPICH2 and RDMA-Hadoop with SR-IOV. We discuss high-performance designs of the virtual machine and container aware MVAPICH2 library over SR-IOV enabled HPC Clouds. This talk will also present a high-performance virtual machine migration framework for MPI applications on SR-IOV enabled InfiniBand clouds.

The MVAPICH2 software for building HPC Clouds presented in this talk is publicly available from http://mvapich.cse.ohio-state.edu. We will also discuss how to leverage the high-performance networking features (e.g., RDMA, SR-IOV) on cloud environments to accelerate data processing through RDMA-Hadoop package, which is publicly available from http://hibd.cse.ohio-state.edu/. Comprehensive performance evaluations on NSF-supported Chameleon Cloud (https://www.chameleoncloud.org) show that our design can deliver the near bare-metal performance.

Designing MPI and PGAS Libraries for Exascale Systems: The MVAPICH2 Approach

Dhabaleswar Panda, The Ohio State University

The MVAPICH2 software libraries have been enabling many HPC clusters during the last 15 years to extract performance, scalability and fault-tolerance using OpenFabrics verbs. As the HPC field is moving to Exascale, many new challenges are emerging to design the next generation MPI, PGAS and Hybrid MPI+PGAS libraries with capabilities to scale to millions of processors while taking advantages of the latest trends in accelerator technologies. In this talk, we will present the approach being taken by the MVAPICH2 project including support for new verbs-level capabilities (DC, UMR, ODP, and offload), PGAS (OpenSHMEM, UPC, CAF and UPC++), Hybrid MPI+PGAS models, tight integration with NVIDIA GPUs (with GPUDirect RDMA and GPUDirect Async) and Intel KNL, and designs leading to reduced energy consumption. We will also highlight a co-design approach where the capabilities of InfiniBand Network Analysis and Monitoring (INAM) can be used together with the new MPI-T capabilities of the MPI standard to analyze and introspect performance of an MPI program on an InfiniBand cluster and

tune it further. We will also present features and plans of the MVAPICH2 project to provide support for emerging technologies: Omni-Path, KNL, OpenPower, SHArP, and Deep Learning.

Status of OFI in MPICH

Ken Raffenetti, Argonne National Laboratory

This session will give the audience an update on the OFI integration in MPICH. MPICH underwent a large redesign effort (CH4) in order to better support high-level network APIs such as OFI. We will show the benefits realized with this design, as well as ongoing work to utilize more aspects of the API and underlying functionality.

Data Intensive Computing & Analysis

Accelerating Apache Spark with RDMA

Yuval Degani, Mellanox; Liran Liss, Mellanox

Apache Spark is today's fastest growing Big Data analysis platform. Spark workloads typically maintain a persistent data set in memory, which is accessed multiple times over the network. Consequently, networking IO performance is a critical component in Spark systems. RDMA's performance characteristics, such as high bandwidth, low latency, and low CPU overhead, offer a good opportunity for accelerating Spark by improving its data transfer facilities.

In this talk, we present a Java-based, RDMA network layer for Apache Spark. The implementation optimized both the RPC and the Shuffle mechanisms for RDMA. Initial benchmarking shows up to 25% improvement for Spark Applications.

Crail: A High-Performance I/O Architecture for the Apache Data Processing Ecosystem

Bernard Metzler, IBM Research; Patrick Stuedi, IBM Research; Animesh Trivedi, IBM Research; Jonas Pfefferle, IBM Research; Ioannis Koltsidas, IBM Research; Radu Stoica IBM Research; Nikolas Ioannou, IBM Research

Recent years have seen major improvements in both networking and storage technology. To exploit the substantially increased performance and also the enriched I/O semantics, new interfaces such as RDMA and NVMe emerged. Nevertheless, even using these new interfaces, the efficient I/O integration into large scale distributed data processing systems remains challenging. In this session, we introduce the 'Crail' project as a new approach to tackle this challenge. Crail is a user-level I/O architecture for the Apache data processing ecosystem, designed from ground up targeting high-performance RDMA networking and storage hardware environments. With Crail, hardware performance advantages become visible at the compute layer and eventually translate into workload run-time improvements. We discuss the basic concepts of Crail and exemplify its impact on sorting and SQL workload performance. Crail is an active open source project. Completely built as an extensible set of loadable and pluggable components, it easily integrates with today's Big Data processing frameworks such as Spark or Flink.

HPC Meets Big Data: Accelerating Hadoop, Spark and Memcached with HPC Technologies

Dhabaleswar Panda, The Ohio State University

Modern HPC clusters are having many advanced features, such as multi-/many-core architectures, high-performance RDMA-enabled interconnects, SSD-based storage devices, burst-buffers and parallel file

systems. However, current generation Big Data processing middleware (such as Hadoop, Spark, and Memcached) have not fully exploited the benefits of the advanced features on modern HPC clusters. This talk will present RDMA-based designs using OpenFabrics Verbs and heterogeneous storage architectures to accelerate multiple components of Hadoop (HDFS, MapReduce, RPC, and HBase), Spark and Memcached. An overview of the associated RDMA-enabled software libraries (being designed and publicly distributed as a part of the HiBD project, http://hibd.cse.ohio-state.edu) for Apache Hadoop (integrated and plug-ins for Apache, HDP, and Cloudera distributions), Apache Spark and Memcached will be presented. The talk will also address the need for designing benchmarks using a multi-layered and systematic approach, which can be used to evaluate the performance of these Big Data processing middleware.

Performance of a Task-Parallel PGAS Programming Model Using OpenSHMEM and UCX

Max Grossman, Rice University; Howard Pritchard, Los Alamos National Laboratory; Vivek Sarkar, Rice University; Zoran Budimlic, Rice University; Nathan Graham, Los Alamos National Laboratory

AsyncSHMEM is a hybrid, task-parallel, PGAS programming model and runtime that combines OpenSHMEM and the Habanero task-parallel programming models. The primary goals of AsyncSHMEM are 1) to prepare OpenSHMEM for future system architectures by enabling the use of asynchronous computation to hide data transfer latencies, 2) to support tight coupling of OpenSHMEM with task parallel programming, and 3) to improve load balance of both computation and communication in OpenSHMEM programs. Since the goals of UCX include providing optimized asynchronous data transfer operations, as well as providing functionality that enables multi-threaded applications to efficiently use RDMA-capable networks, UCX makes the ideal basis for a high performance AsyncSHMEM implementation.

In this session, we describe the AsyncSHMEM runtime, and demonstrate the performance of this runtime when using OpenSHMEM over UCX for a number of benchmarks: distributed integer sort (ISx), unbalanced tree traversal (UTS), and breadth first search (Graph500). We compare the performance of the runtime using Mellanox EDR IB and Intel OPA interconnects, as well as the performance of AsyncSHMEM using a vendor supplied OpenSHMEM implementation.

Deploying RDMA

Deploying OFS Technology in the Wild: A Case Study

Susan Coulter, Los Alamos National Laboratory

Any deployment of a complex system that depends on OFS presents a unique challenge. Policies and guidelines from the organization involved, experience levels among the team doing the deployment, and the combination of technologies being integrated all affect the end product. As these choices and combinations expand, the challenge deepens. This talk will tell the story of just such a deployment challenge at LANL, the various solutions tested, and the lessons learned.

Tera's Data Network: From Storage Cluster to Multi Purpose IB EDR Network

Jérôme David, Commissariat à l'Energie Atomique

This session will present the data network of the computing center Tera, from CEA in France. Mainly

used for storage purpose, it has also evolved to permit user access and cross clusters communication, becoming a multipurpose InfiniBand network. It interconnects all computing clusters of Tera computing center, visualization cluster, and the storage resources. Near future plans exist as to integrate a compilation cluster for users and, maybe an administration cluster with integrated services. Those evolutions and the imagination of both Computer and Storage architects lead us to a network topology bigger than the maximum chassis size (648 ports) and to find a way to interconnect several chassis. We are using InfiniBand to Ethernet interconnection, and a fine grained partition membership for each resource within the network. We'll see how QoS is implemented to guarantee the behavior of the concurrent interactive, services and data flows. Also we'll see how routing analysis with resources placement is important in this topology, leading us to use the routing chain feature of MOFED's opensm. Finally, give you a feedback of our first EDR network and open to perspectives.

Use Cases for Raw Ethernet Queue Pairs

Christoph Lameter, Jump Trading LLC

The RDMA subsystem supports raw ethernet queue pairs which can be used to receive frames directly in user space as well as for sending frames. This allows a convenient way to bypass the kernel network stack with all its baggage that causes high latency and limited throughput. This talk presents the use cases that I have seen for this feature.

Validating RoCEv2 for Production Deployment in the Cloud Datacenter

Sowmini Varadhan, Oracle; Santosh Shilimkar, Oracle

With the increasing prevalence of ethernet switches and NICs in Data Center Networks, we have been experimenting with the deployment of RDMA over Commodity Ethernet (RoCE) in our DCN. RDMA needs a lossless transport, and, in theory, this can be achieved on ethernet by using priority based PFC (IEEE 802.1qbb) and ECN (IETF RFC 3168).

We describe our experiences in trying to deploy these protocols in a RoCEv2 testbed running @ 100 Gbit/sec consisting of a multi-level CLOS network.

In addition to addressing the documented limitations around PFC/ECN (livelock, pause-frame-storm, memory requirements for supporting multiple priority flows), we also hope to share some of the performance metrics gathered, as well as some feedback on ways to improve the tooling for observability and diagnosability of the system in a vendor-agnostic, interoperable way.

Distributed Applications & Services

Lustre Network (LNet) Health

Amir Shehata, Intel

LNet Multi-Rail has implemented the ability for multiple interfaces to be used on the same LNet network or across multiple LNet networks. The LNet Health feature adds the ability to resend messages across different interfaces when interface or network failures are detected. This allows LNet to mitigate communication failures before passing the failures to upper layers for further error handling. To

accomplish this, LNet Health depends on health information reported by the underlying fabrics such as MLX and OPA.

LNet Health will monitor three different types of failures:
- local interface failures as reported by the underlying fabric
- remote interface failures as reported by the remote fabric
- network timeouts.

Each one of these classes of failures are dealt with separately at the LNet layer. The implementation of this health feature at the LNet layer allows LNet to retransmit messages across different types of interfaces. For example if a peer has both MLX and OPA interfaces and a transmit error is detected on one of them then LNet can retransmit the message on the other available interface.

Objects over RDMA

Jeff Inman, Los Alamos National Laboratory; Will Vining, Los Alamos National Laboratory; Garrett Ransom, Los Alamos National Laboratory

At LANL, the Campaign storage-tier is expected to deliver access to 10s and eventually 100s of PBytes, with bandwidth on the order of 30 GB/s, eventually approaching 100 GB/s. This tier is implemented as the “Mar” file-system (MarFS). MarFS puts scalable near-POSIX file-system semantics (i.e. files and directories) on top of a scalable, distributed, erasure-coded backing-store based on REST-ful object-storage semantics (i.e. put/get/delete). The physical storage is currently SMR disks on Infiniband EDR fabric. The backing-store interface can be swapped out in a modular way. To improve bandwidth and scalability over a commodity object-store, we wrote a module to replace the object-store with a custom, scalable file-based storage system, transferring erasure-coded data over NFS to remote file-based “stripes”. However, NFS presents its own performance issues, as well as awkward interactions with the SMR disks. To address this, we’re experimentally replacing NFS with a custom service, based on RDMA sockets, using simple semantics (i.e. open/write/read/close) to move erasure-coded data to and from remote file-based object stripes.

Future Directions in Networking

InfiniBand Trade Association TWG - Recent Topics in the IBTA, and a Look Ahead

Bill Magro, InfiniBand Trade Association

This talk discusses some recent activities in the IBTA including recent specification updates. It also provides a glimpse into the future for the IBTA.

Management, Monitoring & Configuration

Fabric Performance Management and Monitoring

Todd Rimmer, Intel

Fabric management has moved beyond simply bringing up a fabric, but now also includes day to day monitoring and analysis of fabric operation. Analysis needs include not only errors, but also data movement, congestion, and utilization analysis. Such analysis includes both the whole fabric as well as key subsystems such as compute or storage. The Intel® Omni-Path FM and FM GUI which are open sourced include a powerful set of performance monitoring and analysis capabilities. This session will discuss the Performance Manager/Administrator subsystem of the Omni-Path FM as well as the various tools and FM GUI features which permit analysis of current and historic performance data for a cluster.

Host Based Infiniband Network Fabric Monitoring

Michael Aguilar, Sandia National Laboratories; James Brandt, Sandia National Laboratories; Douglas Pase, Sandia National Laboratories

Synchronized host based Infiniband network counter monitoring of local connections at 1Hz can provide a reasonable system snapshot understanding of traffic injection/ejection into/from the fabric. This type of monitoring is currently used to enable understanding about the data flow characteristics of applications and inference about congestion based on application performance degradation. It cannot, however, enable identification of where congestion occurs or how well adaptive routing algorithms and policies react to and alleviate it. Without this critical information the fabric remains opaque and congestion management will continue to be largely handled through increases in bandwidth. To reduce fabric opacity, we have extended our host based monitoring to include internal Infiniband fabric network ports. In this presentation we describe our methodology along with preliminary timing and overhead information. Limitations and their sources are discussed along with proposed solutions, optimizations, and planned future work.

Managing Individual Nodes in Large Fabrics

Ira Weiny, Intel

Individual node configuration when managing 1000s or 10s of thousands of nodes in a cluster can be a daunting challenge. 2 key daemons are now part of the rdma-core package which aid the management of individual nodes in a large fabric.

1) Ibacm – Once a simple PathRecord cache, ibacm has been extended to allow for plugins for enhanced access to SA data. Furthermore work has been done within the Linux kernel to allow the kernel to access the services provided by the ibacm deamon. This talk will present the advantages of using ibacm, what providers are currently available on various distribution channels, and future work to be done.
2) rdma-ndd – manages node description names once and for all. Knowing which node you are talking to is invaluable in a large fabric. Not only does rdma-ndd set node descriptions but more importantly it keeps them updated. A brief overview will be provided.

Omni-Path Fabric Topologies and Routing

Renae Weber, Intel

As clusters grow, the cost/performance trade-offs necessary invite exploration of unique topologies. Such topologies often require specialized routing engines and fabric management capabilities. This session will provide an overview of some of the unique routing capabilities and route tuning provided in the open source Intel® Omni-Path FM. This will include discussion of device group routing, permitting hosts at the root of trees, hypercube, torus, up/down fat tree, spine first and other specialized techniques which permit the deployment of a wide range of unique innovative topologies by end users.

Using Sandia's LDMS for IB Fabric Analyses, Especially of LNET Router Behavior

Serge Polevitzky, Sandia National Laboratories

"Infiniband is not well documented, nor is it well understood."
--- Anonymous

Sandia's Lightweight Distributed Metric Service (LDMS) attempts to reveal Infiniband's normally opaque statistics. We will explore LDMS' error, performance, and congestion metrics.

Some discussion on perfquery and the Subnet Manager's Performance Manager.

This session is intented for anyone interested in understanding Infiniband (IB) behaviors. This is intended to be an informal work session, and will present metrics like VL15s dropped, port_transmit_wait counts, port_transmit_discards, etc., gathered on Sandia's clusters by LDMS, with the data presented visually.

Network APIs, Libraries & Software

Advancing Open Fabrics Interfaces

Sean Hefty, Intel

With its initial release two years ago, libfabric advanced the state of fabric software interfaces. One of the promises of OFI was extensibility: adapting to increased demands of fabric services from applications. This session explores the first major enhancements to the libfabric API in response to user demands and learnings.

IPoIB Acceleration

Rony Efraim, Mellanox; Liran Liss, Mellanox; Tzahi Oved, Mellanox

The IPoIB protocol encapsulates IP packets over Infiniband datagrams. As a direct RDMA Upper Layer Protocol (ULP), IPoIB cannot support HW features that are specific to the IP protocol stack. Nevertheless, RDMA interfaces have been extended to support some of the prominent IP offload features, such as TCP/UDP checksum and TSO. This provided reasonable performance for IPoIB.

However, new network interface features are one of the most active areas of the Linux kernel. Examples include TSS and RSS, tunneling offloads, and XDP. In addition, the basic IP offload features are insufficient to cope with the increasing network bandwidth. Rather than continuously porting IP network

interface developments into the RDMA stack, we propose adding abstract network data-path interfaces to RDMA devices.

In order to present a consistent interface to users, the IPoIB ULP continues to represent the network device to the IP stack. The common code also manages the IPoIB control plane, such as resolving path queries and registering to multicast groups. Data path operations are forwarded to devices that implement the new API, or fallback to the standard implementation otherwise. Using the forgoing approach, we show how IPoIB closes the performance gap compared to state-of-the-art Ethernet network interfaces.

On-demand-paging in Practice

Liran Liss, Mellanox

On Demand Paging (ODP) allows RDMA applications to register virtual memory address ranges without pinning the underlying physical memory. This allows memory regions to exceed physical memory size, but more importantly, to enable a multitude of canonical virtual memory optimizations, such as delayed allocation, COW, and NUMA migration. Recently, ODP has been enhanced to support complete address spaces. This effectively relieves the programmer from the burden of memory registration all together, which greatly simplifies the development of RDMA applications and middleware. In this session, we detail the latest developments in ODP and show, through extensive application benchmarking, the benefits of ODP for both HPC and storage applications. Finally, we outline the synergy between ODP and upcoming cache-coherent accelerators and peer devices.

Packet Processing Verbs for Ethernet and IPoIB

Alex Rosenbaum, Mellanox

As a prominent user-level networking API, the RDMA stack has been extended to support packet processing applications and user-level TCP/IP stacks, initially focusing on Ethernet. This allowed delivering low latency and high message-rate to these applications.

In this talk, we provide an extensive introduction to both current and upcoming packet processing Verbs, such as checksum offloads, TSO, flow steering, and RSS. Next, we describe how these capabilities may also be applied to IPoIB traffic.

In contrast to Ethernet support, which was based on Raw Ethernet QPs that receive unmodified packets from the wire, IPoIB packets are sent over a “”virtual wire””, managed by the kernel. Thus, processing selective IP flows from user-space requires coordination with the IPoIB interface.

RDMA-core Community Collaboration

Jason Gunthorpe, Obsidian Research Corp.

Discuss how to contribute to rdma-core, some interesting need work areas and a brief overview of what has been accomplished so far. This talk can be adjusted to compliment whatever Doug and/or Leon decide to present for rdma-core, including an explanation of introduction and purpose depending on schedule slot.

Ubiquitous RoCE

Alex Shpiner, Mellanox; Liran Liss, Mellanox; Eitan Zahavi, Mellanox

In recent years, the usage of RDMA in datacenter networks has increased significantly, with RoCE (RDMA over Converged Ethernet) emerging as the canonical approach to deploying RDMA in Ethernet-based datacenters.

Initially, RoCE required a lossless fabric for optimal performance. This is typically achieved by enabling Priority Flow Control (PFC) on Ethernet NICs and switches. The RoCEv2 specification introduced RoCE congestion control, which allows throttling transmission rate in response to congestion. Consequently, packet loss may be minimized and performance is maintained even if the underlying Ethernet network is lossy.

In this talk, we discuss the details of latest developments in the RoCE congestion control. Hardware congestion control reduces the latency of the congestion control loop; it reacts promptly in the face of congestion by throttling the transmission rate quickly and accurately; when congestion is relieved, bandwidth is immediately recovered. The short control loop also prevents network buffers from overfilling in many congestion scenarios.

In addition, fast hardware retransmission complements congestion control in heavy congestion scenarios, by significantly reducing the penalty of packet drops.

User Verbs Extensions for Scaled Performance with Shared Memory

Santosh Shilimkar, Oracle; Avneesh Pant, Oracle; Sumanta Chatterjee, Oracle

An Oracle database instance consists of a collection of processes attached to a shared memory segment. An instance can consist of a large number of processes (20K+). Current upstream PD verb implementation which require each process to register entire memory region is not scale-able. We propose an extension to the memory registration verb called Shared PD which requires only a single process to register the entire shared memory segment. All other processes attach to the shared PD handle and can re-use the memory mapping which greatly reduces MPT and MTT entries for an instance. All processes have their own RDMA resources (like QPs, SRQs, CQs etc) and context.

Leveraging MTT cache with minimal entries by use of large/contiguous page(s) MR is well known optimization. We would like to exploit it further by using such large MRs across processes with a shared PD. The key idea is to allocate contiguous pages as shared memory that can be mapped across processes which share a PD.

In our view, these extensions are not IB specific and can benefit wider RDMA technologies for shared memory model and hence potentially good candidates as standard verb APIs than just extensions.

New & Advanced Network Technologies

Extended Memory Windows

Alex Margolin, Mellanox; Liran Liss, Mellanox; Tzahi Oved, Mellanox

RDMA memory windows allow applications to specify an aperture with specific remote access rights into an existing memory region. In this talk, we show how this concept could be extended beyond access rights, for describing complex memory layouts.

Many HPC applications receive regular structured data, such as a column of a matrix. In this case, the application would typically receive a chunk of data and scatter it by the CPU, or use multiple RDMA writes to transfer each element in-place. Both options introduce significant overhead. By using a memory window that specifies strided access, this overhead could be completely eliminated: the initiator posts a single RDMA write and the target HCA scatters each element into place.

Similarly, standard memory windows cannot describe non-contiguous buffers, forcing applications to generate remote keys for each buffer. However, by allowing a memory window to span multiple address ranges, optionally registered by different memory regions, an application may scatter remote data scatter with a single remote key.

Using memory windows, such memory layouts may be created, accessed, and invalidated using efficient, non-privileged, user-level interfaces.

Infiniband Virtualization

Liran Liss, InfiniBand Trade Association

Infiniband Virtualization allows a single Channel Adapter to present multiple transport endpoints that share the same physical port. To software, these endpoints are exposed as independent Virtual HCAs (VHCAs), and thus may be assigned to different software entities, such as VMs. VHCAs are visible to Subnet Management, and are managed just like physical HCAs. This session provides an overview of the Infiniband Virtualization Annex, which was released on November 2016. We will cover the Virtualization model, management, addressing modes, and discuss deployment considerations.

Omni-Path Status, Upstreaming and Ongoing Work

Todd Rimmer, Intel

Intel® Omni-Path was first released in early 2016. Omni-Path host and management software is all open sourced. This session will provide an overview of Omni-Path including some of the technical capabilities and performance results as well as some recent industry results. The session will also highlight some of the areas of change and challenges encountered when adding Omni-Path into Open Fabrics and how they have been addressed as well as ongoing work in order to support Omni-Path within the existing Open Fabrics architecture.

RDMA on ARM

Pavel Shamis, ARM Research

Applications, programming languages, and libraries that leverage sophisticated network hardware capabilities have a natural advantage when used in today's and tomorrow's high-performance and data center computer environments. Modern RDMA based network interconnects provides incredibly rich functionality (RDMA, Atomics, OS-bypass, etc.) that enable low-latency and high-bandwidth communication services. The functionality is supported by a variety of interconnect technologies such as InfiniBand, RoCE, iWARP, Intel OPA, Cray's Aries/Gemini, and others. OFA organization and Linux-RDMA community have been playing a predominant role in the enablement efficient and vendor agnostic software stack for those interconnects. Over the last decade, the community has developed variety user/kernel level protocols and libraries that enable a variety of applications over RDMA including MPI, SHMEM, NFS over RDMA, IPoIB, and many others.

With the emerging availability server platforms based on ARM CPU architecture, it is important to understand ARM integrates with RDMA hardware and software eco-system. In this talk, we will overview ARM architecture and system software stack. We will discuss how ARM CPU interacts with network devices and accelerators. In addition, we will share our experience in enabling RDMA software stack (OFED/MOFED Verbs) and one-sided communication libraries (Open UCX, OpenSHMEM/SHMEM) on ARM and share preliminary evaluation results.

urdma: RDMA Verbs over DPDK (Data Plane Development Kit)

Patrick MacArthur, University of New Hampshire

Software RDMA implementations allow development and use of RDMA verbs without relying on expensive RDMA-capable hardware, at the expense of performance. Existing software RDMA implementations, including softiwarp and softroce (rxe), are implemented in the kernel. Applications may benefit from a software RDMA solution that performs data transfers in userspace, to avoid the overhead of context switches into the kernel. DPDK (Data Plane Development Kit) is an open-source framework for Linux and BSD which enables direct access to standard Ethernet NICs from userspace as well as packet handling utilities and data structures. This talk introduces urdma, an open-source OpenFabrics verbs driver built on top of DPDK. The urdma driver uses a variant of the standard iWARP protocol over UDP. While urdma uses a small kernel module as required for verbs setup and connection management, all fast path data transfer operations are implemented entirely in userspace. This talk will cover the design of the driver, issues encountered, and a performance evaluation of the driver compared to software and hardware iWARP solutions using standard benchmarks.

Persistant (non-volatile) Memory

Experiences with NVMe over Fabrics

Oren Duer, Mellanox; Liran Liss, Mellanox; Max Gurtovoy, Mellanox

NVMe is an interface specification to access non-volatile storage media over PCIe buses. The interface enables software to interact with devices using multiple, asynchronous submission and completion queues, which reside in memory. Consequently, software may leverage the inherent parallelism and low latency of modern NMV devices with minimal overhead. Recently, the NMVe specification has been

extended to support remote access over fabrics, such as RDMA and Fibre Channel. Using RDMA, NVMe over Fabrics (NVMe-oF) provides the high BW and low-latency characteristics of NVMe to remote devices. Moreover, these performance traits are delivered with negligible CPU overhead as the bulk of the data transfer is conducted by RDMA.

In this session, we present an overview of NVMe-oF and its implementation in Linux. We point out the main design choices and evaluate NVMe-oF performance for both Infiniband and RoCE fabrics.

NVM-aware RDMA-based Communication and I/O Schemes for High-Performance Big Data Analytics

Xiaoyi Lu, The Ohio State University; Dhabaleswar Panda, The Ohio State University

The convergence of Big Data and HPC has been pushing the innovation of accelerating Big Data analytics and management on modern HPC clusters. Recent studies have shown that the performance of Apache Hadoop, Spark, and Memcached can be significantly improved by leveraging the high-performance networking technologies, such as Remote Direct Memory Access (RDMA). Most of these studies are based on `DRAM+RDMA' schemes. On the other hand, Non-Volatile Memory (NVM) technologies can offer byte-Addressability with DRAM-like performance along with persistence on HPC clusters. NVMs provide the opportunity to build novel high-performance communication and high-throughput I/O subsystems for data-intensive applications. In this talk, we propose a new high-performance runtime, called NRCIO, which is designed with NVM-aware RDMA-based Communication and I/O schemes for Big Data analytics. Our preliminary studies show that through NRCIO, we can significantly improve the communication, I/O, and application performance for Big Data analytics and management middleware, such as HDFS, HBase, Spark, Memcached, etc.

Update on RDMA Extensions for PMEM - Current Trends and Standardization Activity

Chet Douglas, Intel

-Problem Statement
-Using RDMA with PMEM - How do you commit data to persistence?
-Current ULP SW techniques when using RDMA with PMEM
-Current NVML memory replication using RDMA
-Possibly long term RDMA with PMEM - Performance, reducing messaging round trips
-Current Standardization efforts: IETF LWG, SNIA NVM TWG, OFA Libfabric, Libibverbs
-Current RDMA extension directions from standards discussions

RDMA in Commercial Environments

Ceph RDMA Support

Adir Lev, Mellanox; Liran Liss, Mellanox; Oren Duer, Mellanox

Ceph is a unified, distributed storage system designed for high performance, reliability and scalability. Ceph uses messenger services to exchange messages over the network. Recently, an RDMA asynchronous messenger has been added to the existing Simple and Asynch messengers, which use TCP. The RDMA messenger transfers large buffers efficiently while minimizing data copies. As a result, Ceph delivers higher bandwidth with less CPU overhead.

Ceph RDMA is fully integrated into the upstream code base, and is available in the latest Ceph repositories. It is supported for all major Linux distributions, and managed by the standard Ceph deployment tools. Additionally, migrating existing deployments from TCP to RDMA is straight-forward.

Challenges Faced While Building Enterprise Block Storage Application on Top of the OFED Stack

Subhojit Roy, IBM; Tej Parkash, IBM

Block storage has traditionally been ruled by Fiber Channel. However in recent years High speed Ethernet fabrics has made quick inroads as the primary interconnect for storage. RDMA is seen as the vehicle that enables storage to take advantage of high speed Ethernet fabrics with the promise of low latencies, low CPU utilization and high bandwidth. Two key technologies that enable RDMA over Ethernet are RoCE and iWARP.

This talk is about the challenges faced while designing an enterprise block storage application on top of the OFED stack that works for both RoCE and iWARP. While OFED promises to be a technology independent layer on top of iWARP or RoCE or Infiniband, in practice it isn't entirely there yet. That causes design issues for storage applications that have very specific requirements. This talk will highlight some of those challenges in terms of timeouts, resource allocation, error recovery etc. that are key asks to make it easier for storage applications to be designed using the OFED stack in a technology vehicle that enables storage to take advantage of high speed Ethernet fabrics with the promise of low latencies, low CPU utilization and high bandwidth. Two key technologies that enable RDMA over Ethernet are RoCE and iWARP.

This talk is about the challenges faced while designing an enterprise block storage application on top of the OFED stack that works for both RoCE and iWARP. While OFED promises to be a technology independent layer on top of iWARP or RoCE or Infiniband, in practice it isn't entirely there yet. That causes design issues for storage applications that have very specific requirements. This talk will highlight some of those challenges in terms of timeouts, resource allocation, error recovery etc. that are key asks to make it easier for storage applications to be designed using the OFED stack in a technology independent fashion.

Developer Experiences of the First Paravirtual RDMA Provider and other RDMA Updates

Adit Ranadive, VMware; Aditya Sarwade, VMware; Jorgen Hansen, VMware; Bryan Tan, VMware; George Zhang, VMware; Shelley Gong, VMware; Na Zhang, VMware; Josh Simons, VMware

VMware's Paravirtual RDMA (PVRDMA) device is a new NIC in vSphere 6.5 that allows VMs in a cluster to communicate using Remote Direct Memory Access (RDMA), while maintaining latencies and bandwidth close to that of physical hardware. Recently, the PVRDMA driver was accepted as part of the Linux 4.10 kernel and our user-library was added as part of the new rdma-core package.

In this session, we will provide a brief overview of our PVRDMA design and capabilities. Next, we will discuss our development approach and challenges for joint device and driver development. Further, we will highlight our experience for upstreaming the driver and library with the new changes to the core RDMA stack.

We will provide an update on the performance of the PVRDMA device along with upcoming updates to the device capabilities. Finally, we will provide new results on the performance achieved by several HPC

applications using VM DirectPath I/O.

This session seeks to engage the audience in discussions on: 1) new RDMA provider development and acceptance, and 2) hardware support for RDMA virtualization.

RDMA in the Kernel

Omni-Path HFI Virtual Network Interface Controller (VNIC)

Niranjana Vishwanathapura, Intel

Supporting Ethernet over Omni-Path fabric allows us to make full use of standard Ethernet support provided by the Operating System (including VLAN etc.) over the fabric without having any new layering in the stack. Intel Omni-Path Host Fabric Interface (HFI) Virtual Network Interface Controller (VNIC) feature supports Ethernet functionality over Omni-Path fabric by encapsulating the Ethernet packets between HFI nodes. The patterns of exchanges of Omni-Path encapsulated Ethernet packets involves one or more virtual Ethernet switches overlaid on the Omni-Path fabric topology. In this presentation, we will provide an overview of the virtual Ethernet switch architecture and will focus on the HFI VNIC driver functionality being upstreamed into the Linux kernel including the new architectural changes to the rdma core to allow for stream lined SKB processing on rdma hardware and additional use cases where VNIC can be used.

Linux NFS/RDMA 2017 Roadmap

Chuck Lever, Oracle

The presentation will:
-Introduce the current feature set of the upstream Linux NFS/RDMA implementation, including support for Remote Invalidation, NFS/Kerberos, and NFSv4.1 on RDMA
-Describe plans for new features in the upstream Linux NFS/RDMA implementations during 2017
-Explore the potential of the combination of using the pNFS block layout with NFS/RDMA and NVMe/F (Christoph Hellwig's recent work), and how Tom Talpey's ""push mode"" can be implemented in NFS/RDMA (unless of course Christoph will be there to present this himself)
-Report progress on IETF standards related to NFS/RDMA, including features currently being considered for RPC-over-RDMA Version Two

RDMA Subsystem Issues and the Core Linux Kernel

Christoph Lameter, Jump Trading LLC

There are a number of issues with the core Linux kernel that influence the operations of the RDMA subsystem. One problem that surfaced again recently is the limitations of performance because of the lack of contiguous memory. Another point of problems is related to the pinning / locking of memory for DMA operations. Then we have the inconsistencies between the regular network stack and the RDMA devices because they are separate from regular network devices. Other minor annoyances also exist.

This talk gives an overview of the integration problems, their history and either the work in progress to resolve these issues or the thinking of the kernel community on how to approach a solution.

The Linux SoftRoCE Driver

Yonatan Cohen, Mellanox; Liran Liss, Mellanox

SoftRoCE is a software implementation of the RDMA transport protocol over Ethernet. Thus, any host to conduct RDMA traffic without necessitating a RoCE-capable NIC, allowing RDMA development anywhere.

This session presents the Linux SoftRoCE driver, RXE, which was recently accepted to the 4.9 kernel. In addition, the RXE user-level driver is now part of rdma-core, the consolidated RDMA user-space codebase. RXE is fully interoperable with HW RoCE devices, and may be used for both testing and production. We provide an overview of the RXE driver, detail its configuration, and discuss the current status and remaining challenges in RXE development.

The RDMA Kernel ABI Framework

Matan Barak, Mellanox; Liran Liss, Mellanox

The RDMA subsystem is composed of both kernel and user-space components. User-space libraries, such as libibverbs and user-space driver libraries, communicate with the kernel through an ABI. The existing ABI is limited to fixed set of object types and methods. Although the list of attributes for a given method is extensible, it can only be extended by appending new attributes at the end. We propose an extendable ABI, which allows adding arbitrary object types, methods and attributes. Extensions could be either unique to a specific hardware, be shared among certain hardware families, or accepted as a generic interface. This opens the door to enriching the API and letting applications take advantage of novel HW features and offloads, without burdening the implementation for devices that do not support them. The ABI infrastructure automates the common tasks of parameter packing and validation, user-handle to kernel object mapping, and method dispatching. Consequently, the implementation of both existing and future APIs is more robust and maintainable. Finally, extensions are described in a formal parse tree. This provides a common specification for new features across multiple devices, and forms the basis for an accurate and detailed query subsystem for device features.

Standards Update

CCIX, Gen-Z, and OpenCAPI: A Comparison of Three Emerging System Interconnect Technologies

Brad Benton, AMD

Over the past year, three new industry consortia have been formed to define new system interconnect technologies for use in future servers and datacenters: CCIX, Gen-Z, and OpenCAPI. All are open standards and are focused on designs to enable more efficient transfer of data and use of accelerators, such as GPUs, FPGAs, as well as storage technologies such as Storage Class Memory (SCM)/ Persistent Memory (PM). While each approach has unique aspects, there are also similarities. This talk will discuss the goals and attributes of each project, where they overlap and where they are distinct, and implications for hardware and associated software.

Gen-Z - An Overview and Use Cases

Greg Casey, Dell EMC; Kurtis Bowman, Dell

This session will focus on the new Gen-Z memory-semantic fabric. The speaker will show the audience why Gen-Z is needed, how Gen-Z operates, what is expected in first products that employ Gen-Z, and encourage participation in finalizing the Gen-Z specifications. Gen-Z will be connecting components inside of servers as well as connecting servers with pools of memory, storage, and acceleration devices through a switch environment.

OpenFabrics Alliance Business

Accelerating OFI Libfabric Adoption Through Compliance Testing

Paul Bowden, Intel; Robert D. Russell, UNH-IOL; Paul Grun, Cray

The OFA not only develops and maintains network software stacks, it also supports those stacks through a robust interoperability program. The Interoperability Working Group (IWG) has been exploring avenues to increase the program's value by expanding the existing Interoperability Program in significant ways, but also by augmenting the program to accommodate newer OFA efforts such as the OpenFabrics Interfaces program. Because of the transport-independent approach taken by the OFI program, it is clear that the existing interoperability program should be augmented by a corresponding compliance program. This talk reviews the current state of the Interoperability Program, discusses proposals currently in flight to expand that program to include significant participation by Linux distros. The talk also presents the current thinking vis-à-vis an emerging proposal to develop a compliance program to support the libfabric user libraries.