MVAPICH2 on Intel® Omni-Path Architecture

Sayantan Sur

Legal Disclaimer & Optimization Notice

INFORMATION IN THIS DOCUMENT IS PROVIDED “AS IS”. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. INTEL ASSUMES NO LIABILITY WHATSOEVER AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO THIS INFORMATION INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.

Software and workloads used in performance tests may have been optimized for performance only on Intel microprocessors. Performance tests, such as SYSmark and MobileMark, are measured using specific computer systems, components, software, operations and functions. Any change to any of those factors may cause the results to vary. You should consult other information and performance tests to assist you in fully evaluating your contemplated purchases, including the performance of that product when combined with other products.

Optimization Notice

Intel’s compilers may or may not optimize to the same degree for non-Intel microprocessors for optimizations that are not unique to Intel microprocessors. These optimizations include SSE2, SSE3, and SSSE3 instruction sets and other optimizations. Intel does not guarantee the availability, functionality, or effectiveness of any optimization on microprocessors not manufactured by Intel. Microprocessor-dependent optimizations in this product are intended for use with Intel microprocessors. Certain optimizations not specific to Intel microarchitecture are reserved for Intel microprocessors. Please refer to the applicable product User and Reference Guides for more information regarding the specific instruction sets covered by this notice.

Notice revision #20110804

Intel® Omni-Path Architecture

The Intel® OPA 100 Series is an end-to-end Fabric solution

Scales to 10,000 nodes or more

Future Omni-Path Fabric to be deployed on Argonne Aurora that has greater than 50,000 nodes

Unique integration of CPU and Fabric

- Density ↑, Reliability ↑, Power ↓

Massively scaled up of the Host Fabric Interface (HFI) capabilities

  - 100 Gbps HFI with PCIe v3.0 x16 host interface
  - Host ASIC contains two separate full performance HFI instances
  - Scaling and optimization of the internal HFI micro-architecture

Fabric features: Adaptive Routing, Dispersive Routing, Traffic Flow optimizations, and many others

Software on Intel® Omni-Path 100

PSM:

PSM version 2: https://github.com/01org/opa-psm2

Fully backward compliant with PSM

New feature: tag size increased to 96-bits from 64-bits

Open Fabrics Interface (OFI):

Next-generation Fabric interface being defined and developed by the Open Fabrics Interfaces Working Group under the Open Fabrics Alliance (OFA)

libfabric version 1.1 was released recently

OFI libfabric PSM provider: https://github.com/ofiwg/libfabric

The PSM 2 provider will be available soon

Tag-bits usage in MVAPICH2

PSM:

Communicator	Source	Tag
16-bits	16-bits	32-bits

Under this scheme, the sender rank, thereby number of ranks in communicator, is limited to 64K

PSM2:

Communicator	Source	Tag
32-bits	32-bits	32-bits

There is more space for both communicator and source bits with adequate space up to Exascale limits

PSM2 API Changes relating to tag

```
typedef
struct psm_mq_tag {
        union {
                uint32_t tag[PSM_MQ_TAG_ELEMENTS];
                struct {
                        uint32_t tag0;
                        uint32_t tag1;
                        uint32_t tag2;
                };
        };
} psm_mq_tag_t;
```

Packed structure instead of flat uint64_t bitfield

```
psm_error_t
psm_mq_send2(psm_mq_t mq, psm_epaddr_t dest,
        uint32_t flags, psm_mq_tag_t *stag,
        const void *buf, uint32_t len);
```

Tag passed in as a pointer instead of by value

Fully working patch to MVAPICH2 available from Intel, working with the team to integrate into release

(thanks to Henry Estela who developed the patch)

Open Fabrics Interfaces

Next-Generation OFA Interfaces, and future direction for OpenFabrics

Open Source

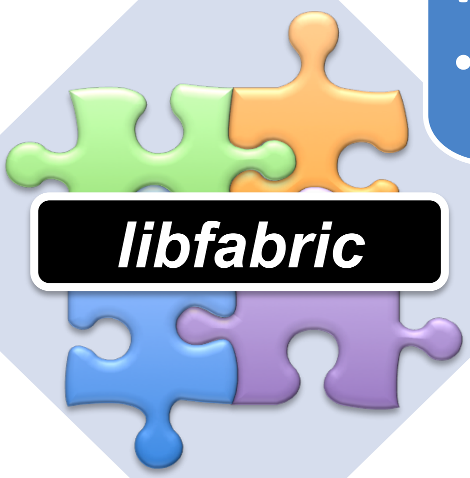
Leverage existing open source community

- Inclusive development effort
- App and HW developers

Application-Centric

Software interfaces aligned with application requirements

- 168 requirements from MPI, PGAS, SHMEM, DBMS, sockets, NVM, …



Scalable

Optimized SW path to HW

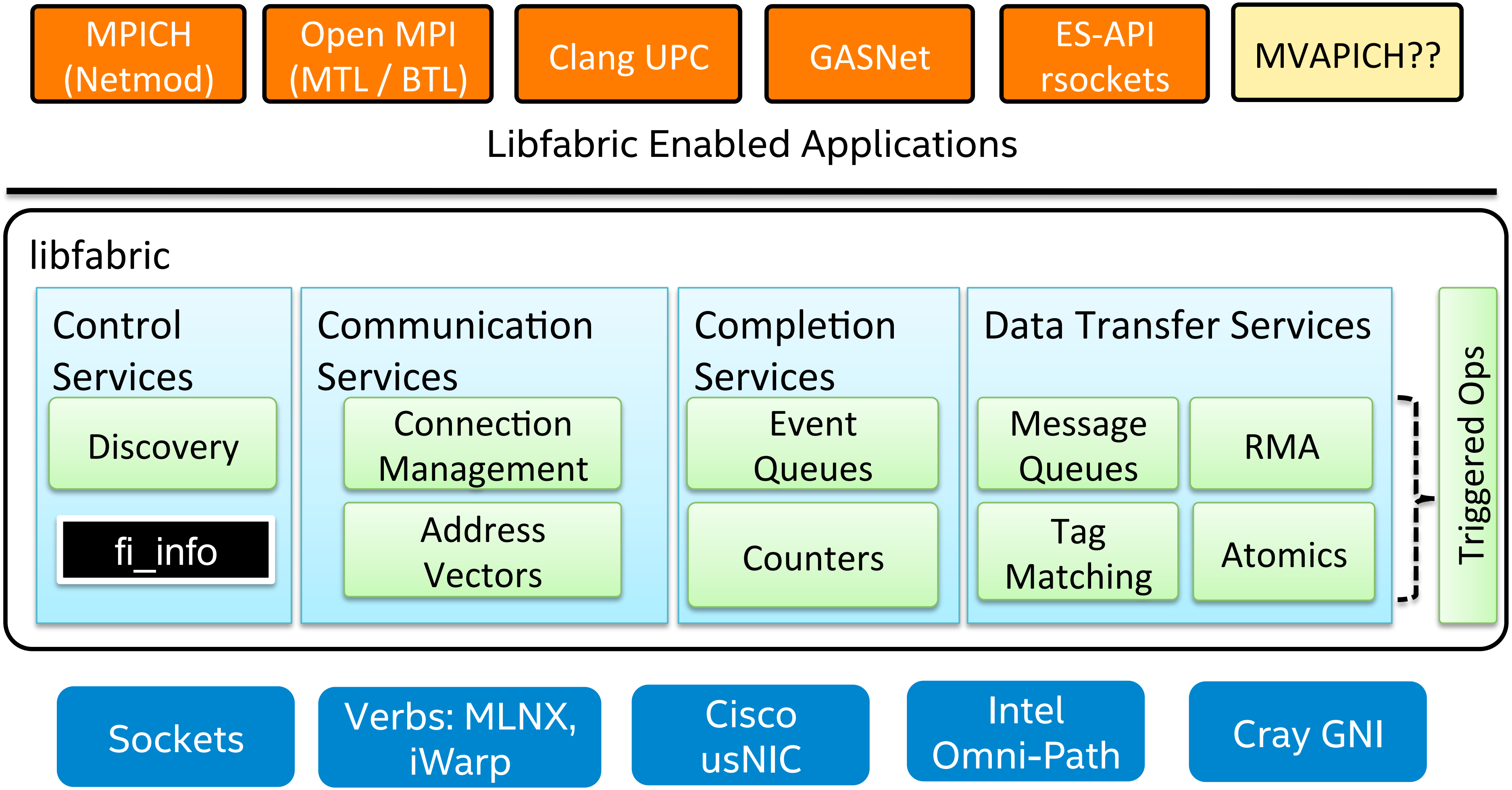
- Minimize cache and memory footprint
- Reduce instruction count
- Minimize memory accesses

Implementation Agnostic

Good impedance match with multiple fabric hardware

- InfiniBand, iWarp, RoCE, raw Ethernet, UDP offload, Omni-Path, GNI, others

Open Fabrics Interface Architecture



libfabric API Analysis: Critical path send

Issues apply to many APIs: Verbs, AIO, DAPL, Portals, NetworkDirect, …

Table from Libfabric paper at HOTI 2015

libibverbs with InfiniBand				libfabric with InfiniBand			
Structure	**Field**	**Write Size**	**Branch?**	**Type**	**Parameter**	**Write Size**	**Branch?**
sge		16		void *	buf	8	
send_wr		60		size_t	len	8	
	next		Yes	void *	desc	8	
	num_sge		Yes	fi_addr_t	dest_addr	8	
	opcode		Yes	void *	context	8	
	flags		Yes				
Totals		**76+8 = 84**	**4+1 = 5**			**40**	**0**

Generic entry points result in additional memory reads/writes

Interface parameters can force branches in the provider code

Move operation flags into initialization code path for optimal SW paths

libfabric API Analysis: Memory Footprint

Per peer addressing data

Table from Libfabric paper at HOTI 2015

libibverbs with InfiniBand			libfabric with InfiniBand		
Type	**Data**	**Size**	**Type**	**Data**	**Size**
struct *	ibv_ah	8	uint64	fi_addr_t	8
uint32	QPN	4			
uint32	QKey	4 [0]			
ibv_ah		24			
Total		**36**			**8**

Map Address Vector :

- encodes peer address
- direct mapping to HW command data

IB Data:	**DLID**	**SL**	**QPN**
Size:	2	1	3

Index Address Vector :

- minimal footprint
- requires lookup/calculation for peer address

Shared Address Table:
easily shareable for all processes on the node!!

Growing OFI Ecosystem and adoption

Officially sanctioned by OFA

Developed by a broad set of stakeholders

Adoption, contributions from industry and lab partners

Positive feedback from users and implementers alike

Collaborative publications and tutorial material being developed

31 members on GitHub, five different fabric providers

Intel views libfabric as the best way to foster and support fabric innovation, including Omni-Path

Summary

MVAPICH2 over Omni-Path works out of the box (no changes)

Can be further enhanced for scalability by using 96-bit tag

Patch to MVAPICH2 available as open-source

OFI libfabric has made progress

- Two releases
- Middleware support available: MPICH, Open MPI, UPC, GASNet, Sockets

Offers fundamental performance and scaling benefits compared to libibverbs

MVAPICH2 on OFI would benefit users by enabling multiple fabrics and encouraging fabric innovation by vendors!