OPENFABRICS ALLIANCE

A Brief Introduction to OpenFabrics Interfaces' libfabric

Dave Goodell, Paul Grun, Sean Hefty, Howard Pritchard,
Bob Russell, Jeff Squyres, Sayantan Sur

OpenFabrics libibverbs middleware

- *Widely adopted low-level RDMA API*
- *Ships with upstream Linux*
- *Intended as unified API for RDMA*

but…

- Designed around InfiniBand architecture
  - Targets specific hardware implementation
- Hardware, not network, abstraction
  - Too low level for most consumers
  - Interfaces not designed around HPC
- Hardware and fabric features are changing
  - Divergence is driving competing APIs
  - PSM, MXM, PAMI, uGNI …
- More applications require high-performance fabrics
  - Cloud systems, data analytics, virtualization, big data …

Solution

OpenFabrics Interfaces Working Group

OPENFABRICS ALLIANCE

Open Source

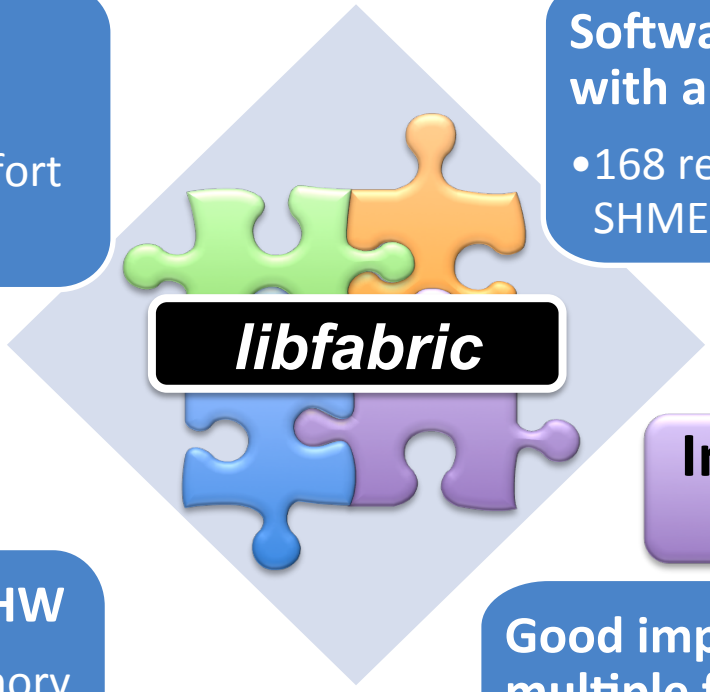
Leverage existing open source community

- 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

Design

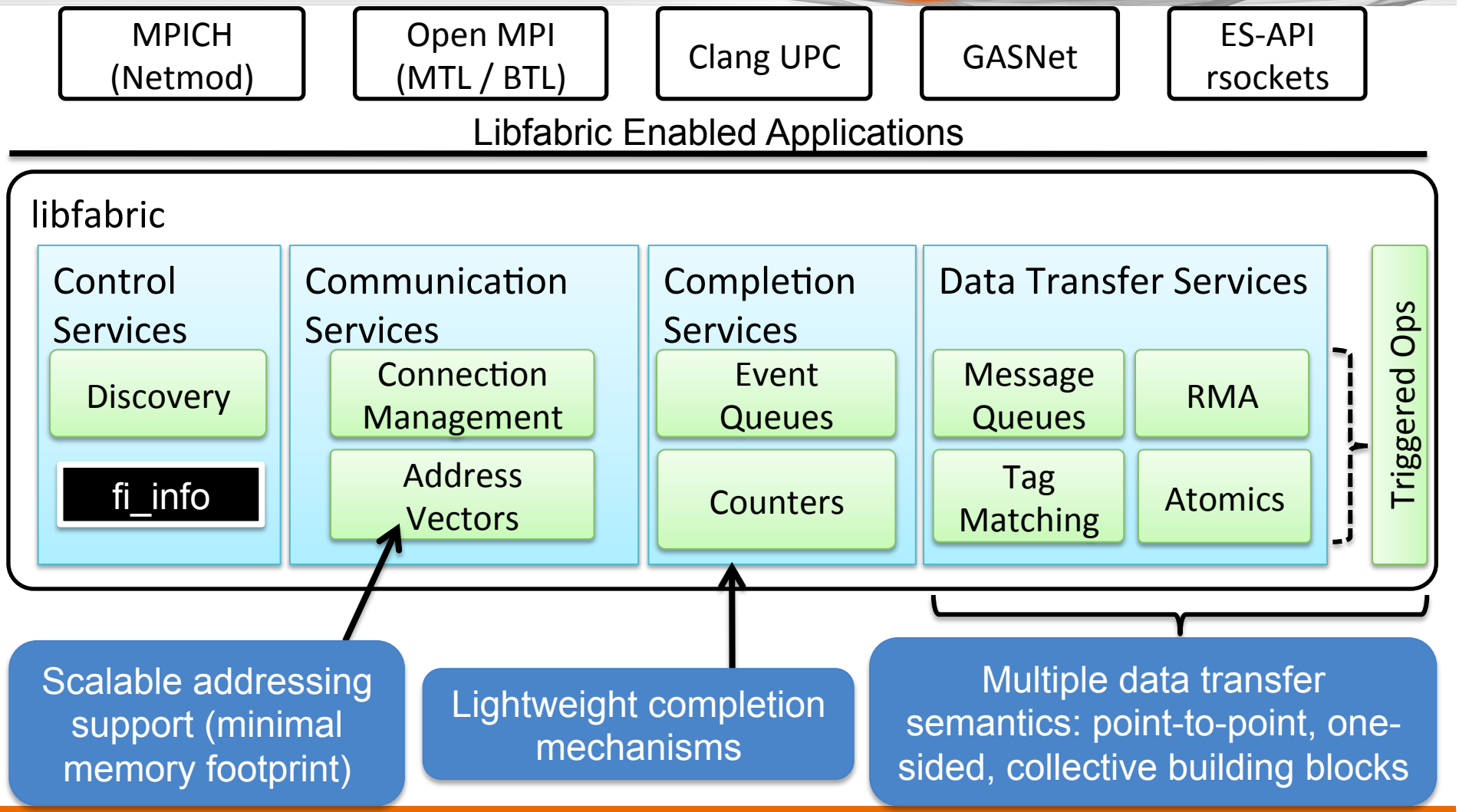
- Enable simple, basic usage
- Move functionality under libfabric
  - E.g. reliability over unreliable transports, SW atomic support

- Advanced application constructs
- Expose abstract HW capabilities
  - Data and message ordering, progress model, resource mgmt constraints, …

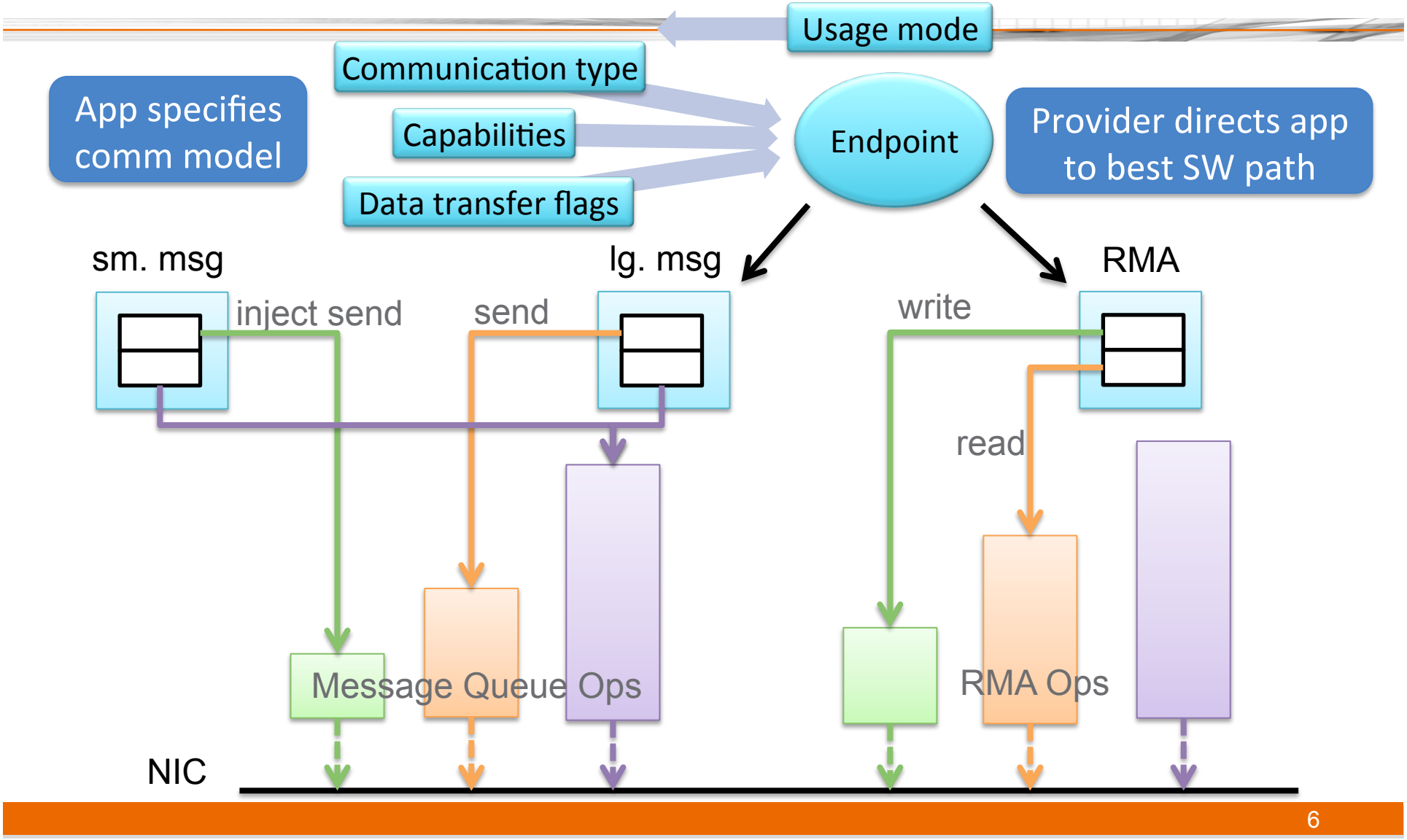
Range of usage models

Architecture

Note: current implementation focused on enabling applications



Application Configured Interfaces



API Performance Analysis

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

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

Memory Footprint

Per peer addressing data

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

Thank You

OPENFABRICS ALLIANCE

OpenFabrics Interfaces Working Group

Develop an extensible, open source framework and interfaces aligned with ULP and application needs for high-performance fabric services

ofiwg@lists.openfabrics.org

github.com/ofiwg

Architecture