



Performance Migration to Open Solutions:

OFED* for Embedded Fabrics

Kenneth Cain
kcain@mc.com

* OpenFabrics Enterprise Distribution

Outline

- **Performance Migration to Open Solutions**
 - What's open, what is the cost?
 - Opportunities for a smooth migration
 - Migration example: ICS/DX to MPI over sRIO
- **OFED sRIO Performance Data / Analysis**
 - Message passing and RDMA patterns, benchmarking approach
 - Varied approaches within sRIO (MPI, OFED, ICS/DX, etc.)
 - Common approaches across fabrics (sRIO, IB, iWARP)
- **Conclusions**

Related to Recent HPEC Talks

- **HPEC 2008**

- *Using Layer 2 Ethernet for High-Throughput, Real-Time Applications*
- Robert Blau / Mercury Computer Systems, Inc.

- **HPEC 2009**

- *The “State” and “Future” of Middleware for HPEC*
- Anthony Skjellum / RunTime Computing Solutions, LLC and University of Alabama at Birmingham

- **Middleware has been a significant focus for the HPEC community for a long time**



Performance Migration to Open Solutions

High Level Perspective

Scalability Reference Model

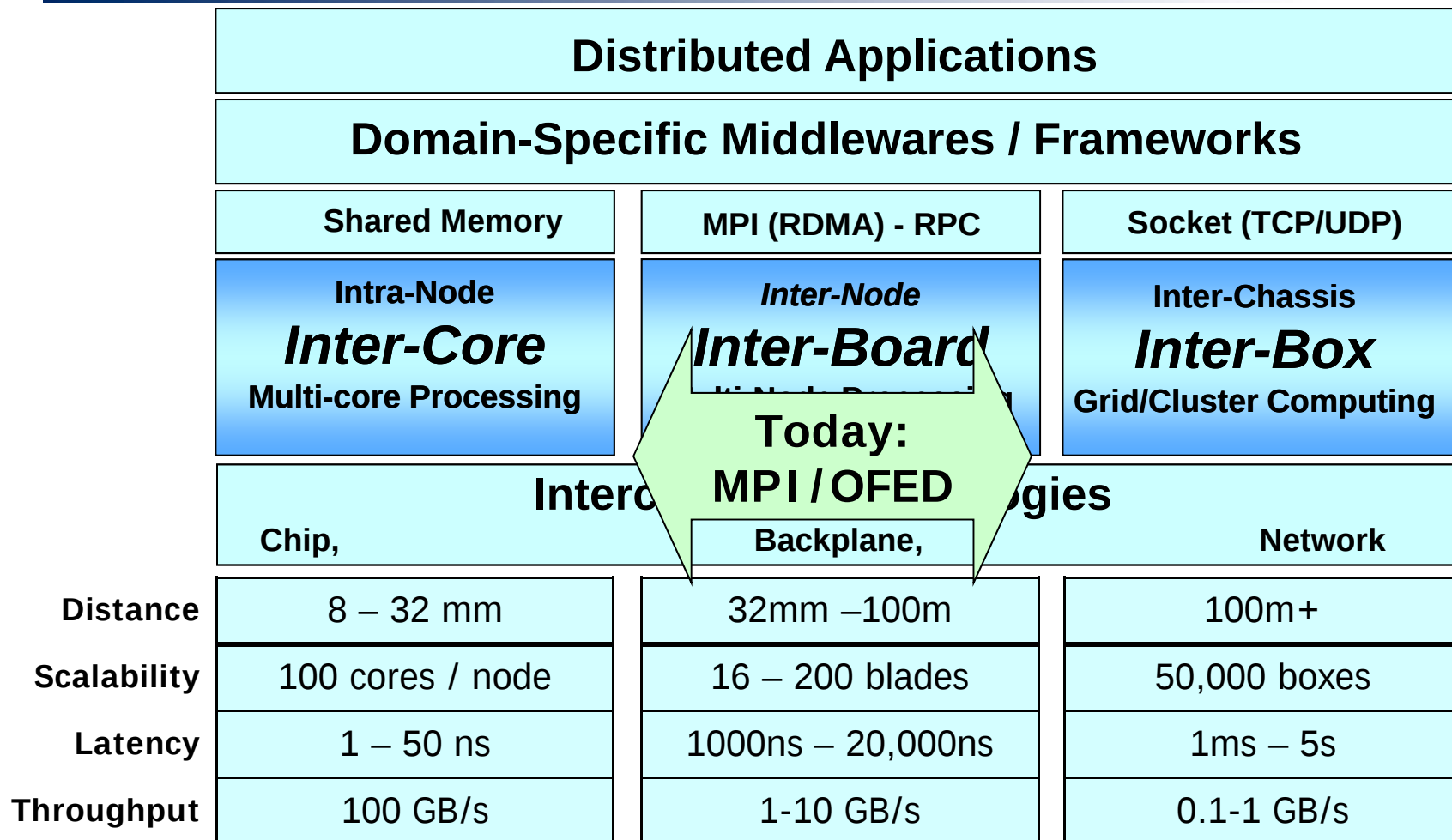
Distributed Applications			
Domain-Specific Middlewares / Frameworks			
Shared Memory		MPI (RDMA) - RPC	Socket (TCP/UDP)
Intra-Node Inter-Core Multi-core Processing		<i>Inter-Node</i> Inter-Board Multi-Node Processing	Inter-Chassis Inter-Box Grid/Cluster Computing
Interconnect Technologies			
Chip,		Backplane,	Network
Distance	8 – 32 mm	32mm –100m	100m+
Scalability	100 cores / node	16 – 200 blades	50,000 boxes
Latency	1 – 50 ns	1000ns – 20,000ns	1ms – 5s
Throughput	100 GB/s	1-10 GB/s	0.1-1 GB/s

Scalability Reference Model

Distributed Applications			
Domain-Specific Middlewares / Frameworks			
Shared Memory		MPI (RDMA) - RPC	Socket (TCP/UDP)
Intra-Node Inter-Core Multi-core Processing		Inter-Node Inter-Board Multi-Node Processing	Inter-Chassis Inter-Box Grid/Cluster Computing
Interconnect Technologies			
Chip,		Backplane,	Network
Distance	8 – 32 mm	32mm –100m	100m+
Scalability	100 cores / node	16 – 200 blades	50,000 boxes
Latency	1 – 50 ns	1000ns – 20,000ns	1ms – 5s
Throughput	100 GB/s	1-10 GB/s	0.1-1 GB/s

- Only a Model – And Models Have Imperfections
- Innovation Opportunity in Bridging Domains

Scalability Reference Model



- Only a Model – And Models Have Imperfections
- Innovation Opportunity in Bridging Domains

Fabric Software Migrations and Goals

Transitions		Goals, Potential Solutions
Compute		• Establish Open Baseline in PPC • Only Recompile if Migrating to x86
PPC,FPGA	x86, GPGPU, FPGA	

Fabric Software Migrations and Goals

Transitions		Goals, Potential Solutions
Compute PPC,FPGA x86, GPGPU, FPGA		• Establish Open Baseline in PPC • Only Recompile if Migrating to x86
Interconnects and Data Rates		• Common RDMA API for All Fabrics • SW Binary Compatible among Fabrics • RDMA+LAN APIs for Selected Fabric • Programmable Fabric Device
10Gbps sRIO	10/20/40/100Gbps sRIO, Ethernet, IB	

Fabric Software Migrations and Goals

Transitions		Goals, Potential Solutions
Compute PPC,FPGA x86, GPGPU, FPGA Interconnects and Data Rates 10Gbps sRIO 10/20/40/100Gbps sRIO, Ethernet, IB		• Establish Open Baseline in PPC • Only Recompile if Migrating to x86 • Common RDMA API for All Fabrics • SW Binary Compatible among Fabrics • RDMA+LAN APIs for Selected Fabric • Programmable Fabric Device
Middleware, Use Cases		• Leveraged SW Investment, Reuse • HW Acceleration • RDMA, Middleware, Protocols
MPI, Vendor DMA Algorithm Library	Integrated MPI/Dataflow Parallel Algorithm Library Pub/Sub, Components	

Fabric Software Migrations and Goals

Transitions		Goals, Potential Solutions
Compute		• Establish Open Baseline in PPC • Only Recompile if Migrating to x86
PPC,FPGA	x86, GPGPU, FPGA	
Interconnects and Data Rates		
10Gbps sRIO	10/20/40/100Gbps sRIO, Ethernet, IB	• Common RDMA API for All Fabrics • SW Binary Compatible among Fabrics • RDMA+LAN APIs for Selected Fabric • Programmable Fabric Device
Middleware, Use Cases		• Leveraged SW Investment, Reuse • HW Acceleration • RDMA, Middleware, Protocols
MPI, Vendor DMA Algorithm Library	Integrated MPI/Dataflow Parallel Algorithm Library Pub/Sub, Components	
Dev / Run / Deployment Flexibility		• Productive Porting to Embedded • Few / No Code Changes • Good Performance Out of the Box • Then, Incremental Tuning
Dev+Run/Embedded Different Air/Ground SW	Dev/Cluster, Run/Embed Common Air/Ground SW	

Fabric Software Migrations and Goals

Transitions		Goals, Potential Solutions
Compute		• Establish Open Baseline in PPC • Only Recompile if Migrating to x86
PPC, FPGA	x86, GPGPU, FPGA	
Interconnects and Data Rates		• Common RDMA API for All Fabrics • SW Binary Compatible among Fabrics • RDMA+LAN APIs for Selected Fabric • Programmable Fabric Device
10Gbps sRIO	10/20/40/100Gbps sRIO, Ethernet, IB	
Middleware, Use Cases		• Leveraged SW Investment, Reuse • HW Acceleration • RDMA, Middleware, Protocols
MPI, Vendor DMA	Integrated MPI/Dataflow	
Algorithm Library	Parallel Algorithm Library	
	Pub/Sub, Components	
Dev / Run / Deployment Flexibility		• Productive Porting to Embedded • Few / No Code Changes • Good Performance Out of the Box • Then, Incremental Tuning
Dev+Run/Embedded	Dev/Cluster, Run/Embed	
Different Air/Ground SW	Common Air/Ground SW	

- **Performance Always a Requirement**
 - Optimize relative to SWaP, Overcome “price of portability”
- **New Approach – Community Open Architecture Goals Not Met**

Fabric Software Migrations and Goals

Transitions		Goals, Potential Solutions
Compute		• Establish Open Baseline in PPC • Only Recompile if Migrating to x86
PPC,FPGA	x86, GPGPU, FPGA	
Interconnects and Data Rates		• Common RDMA API for All Fabrics • SW Binary Compatible among Fabrics • RDMA+LAN APIs for Selected Fabric • Programmable Fabric Device
10Gbps sRIO	10/20/40/100Gbps sRIO, Ethernet, IB	
Middleware, Use Cases		• Leveraged SW Investment, Reuse • HW Acceleration • RDMA, Middleware, Protocols
MPI, Vendor DMA Algorithm Library	Integrated MPI/Dataflow Parallel Algorithm Library Pub/Sub, Components	
Dev / Run / Deployment Flexibility		• Productive Porting to Embedded • Few / No Code Changes • Good Performance Out of the Box • Then, Incremental Tuning
Dev+Run/Embedded Different Air/Ground SW	Dev/Cluster, Run/Embed Common Air/Ground SW	

- **Performance Always a Requirement**
 - Optimize relative to SWaP, Overcome “price of portability”
- **New Approach – Community Open Architecture Goals Not Met**

Industry Scaling Solutions

VSIPL++ *OpenCV*
VSIPL

MPI
DRI

DDS *AMQP* *CORBA*
EIB

Inter-Core

Inter-Board

Inter-Box

Industry Scaling Solutions

VSIPL++ OpenCV
VSIPL

MPI
DRI

DDS AMQP CORBA
EIB

Middleware , Application Frameworks
Mercury, 3rd Party, Customer Supplied....

Inter-Core

Inter-Board

Inter-Box

Industry Scaling Solutions

VSIPL++ *OpenCV*
VSIPL

MPI
DRI

DDS *AMQP* *CORBA*
EIB

Middleware , Application Frameworks
Mercury, 3rd Party, Customer Supplied....

Inter-Core

Inter-Board

Inter-Box

Middleware Engage Platform Services
for Performance and Scalability

OpenMP
IPP *Ct*
OpenCL *CUDA*
HT *QPI*

RapidIO *InfiniBand*
(sRIO) *(IB)* *Ethernet*

Industry Scaling Solutions

VSIPL++ *OpenCV*
VSIPL

MPI
DRI

DDS *AMQP* *CORBA*
EIB

Inter-Core
OpenCL

Inter-Board
OFED

Inter-Box
Network Stack

OpenMP
IPP *Ct*
OpenCL *CUDA*
HT *QPI*

RapidIO *InfiniBand*
(sRIO) *(IB)* *Ethernet*

Industry Scaling Solutions

VSIPL++ *OpenCV* *MPI* *DDS* *AMQP* *CORBA*
VSIPL *DRI* *EIB*

Pivot Points
**Concentrations of industry investment
(sometimes performance)**

Inter-Core

OpenCL

Inter-Board

OFED

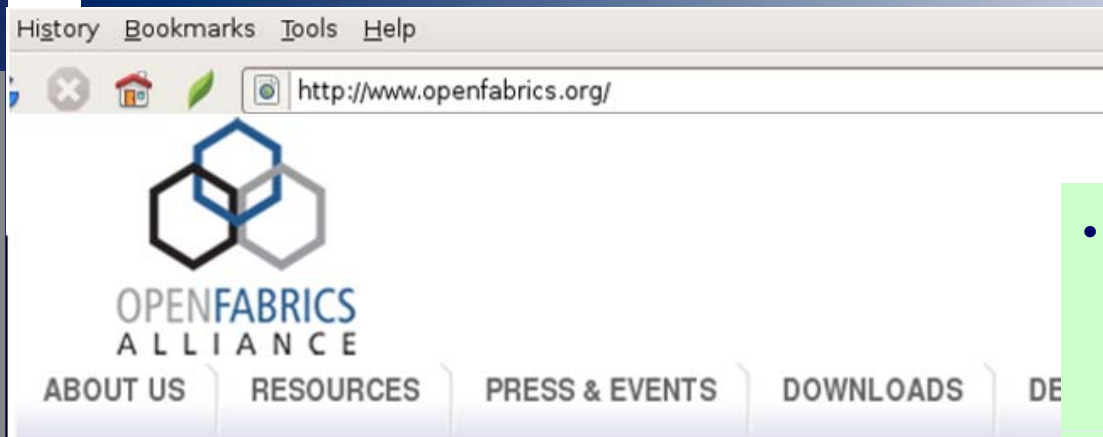
Inter-Box

Network Stack

OpenMP
IPP *Ct*
OpenCL *CUDA*
HT *QPI*

RapidIO *InfiniBand*
(sRIO) *(IB)* *Ethernet*

What is OFED?



Who is the OpenFabrics Alliance?

The OpenFabrics Alliance (OFA) develops, tests, licenses, supports and distributes OpenFabrics Enterprise Distribution (OFED™) open source software for high-performance networking applications that demand low latency and high scalability. OFA also provides forums for education, training and collaborative roadmap development related to OFED and use of the software by OEMs and end users.

What is OFED?

OpenFabrics Enterprise Distribution (OFED™) is a downloadable, networking open-source software package that implements Remote Direct Memory Access (RDMA) and kernel bypass mechanisms to deliver high-efficiency computing, wire-speed messaging, ultra-low microsecond latencies and fault tolerance for servers, block storage and file systems. OFED downloads are available for Linux and Windows through GPL or BSD licensing.

Benefits of the OpenFabrics Alliance's OFED software

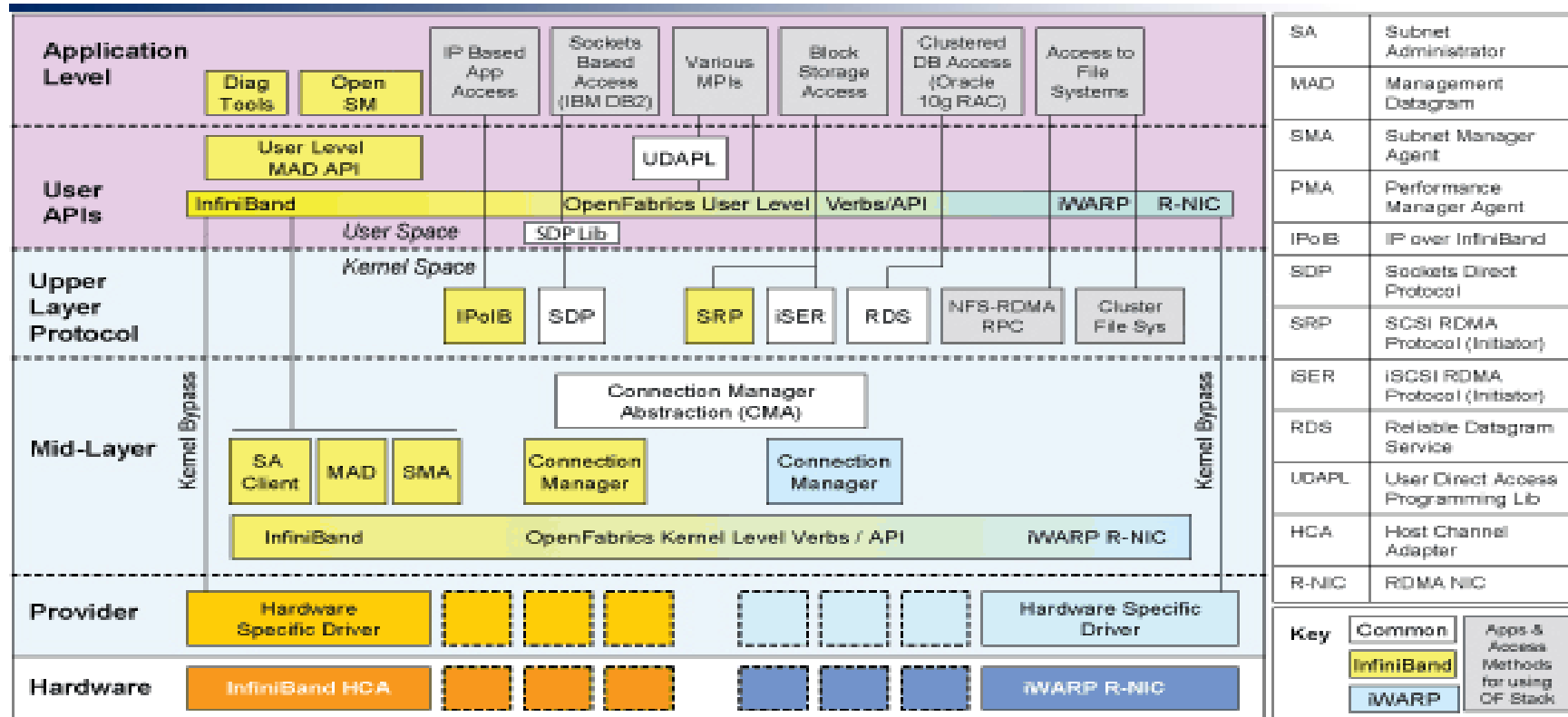
1. Leverages RDMA technology and makes efficient use of CPUs and memory
2. Supports industry-standard, high-performance I/O technologies from multiple vendors
3. Improves application performance and productivity up to 10X
4. Cuts hardware costs by as much as 50 percent by increasing server and CPU efficiency
5. Reduces capital expenses and operating expenses
6. Decreases floor space, electric power and cooling capacity required in the data center

- **Open Source Software by The OpenFabrics Alliance**
 - Mercury is a member
- **Multiple Fabric/Vendor**
 - Ethernet/IB, sRIO (this work)
 - “Offload” HW Available
- **Multiple Use**
 - MPI (multiple libraries)
 - RDMA: uDAPL, verbs
 - Sockets
 - Storage: block, file
- **High Performance**
 - 0 Copy – RDMA, Channel I/O

OFA – Significant Membership Roster

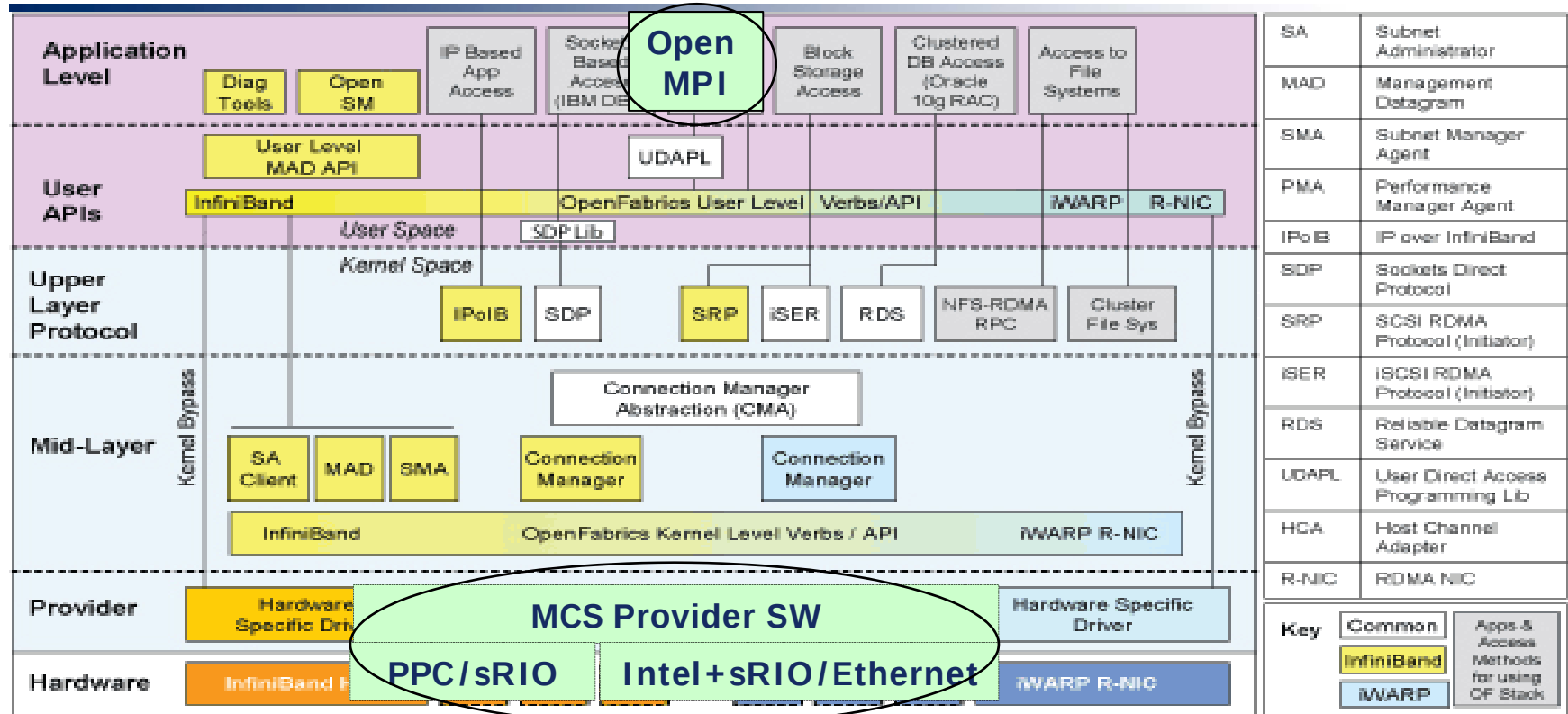
- See <http://www.openfabrics.org> for the list
- HPC, System Integrators including “Tier 1”
- Microprocessor Providers
- Networking Providers (NIC/HCA, Switch)
- Server/Enterprise Software + Systems Providers
- NAS / SAN Providers
- Switched Fabric Industry Trade Associations
- Financial Services / Technology Providers
- National Laboratories
- Universities
- Software Consulting Organizations

OFED Software Structure



- Diagram From OpenFabrics Alliance, <http://www.openfabrics.org>

OFED Software Structure, MCS Work



- Diagram From OpenFabrics Alliance, <http://www.openfabrics.org>
- An OFED “Device Provider” Software Module
 - (now) Linux PPC 8641D/sRIO
 - (future) Linux Intel PCIe to sRIO / Ethernet Device
- Open MPI with OFED Transport (<http://www.open-mpi.org>)

Embedded Middleware Migration

Embedded SW Migration

Open Solution SW Migration

Embedded Middleware Migration

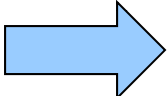
Embedded SW Migration

<u>ICS/DX</u>	<u>ICS/DX+MPI</u>	<u>ICS/DX+MPI tuned</u>
•Mercury middleware •No MPI content •Preserve SW investment	• OFED in kernel driver •Incremental MPI use •App. buffer integration <code>MPI_Send(smb_ptr,)</code> •MPI as MCP extension <code>MPI_Barrier(world)</code>	• OFED OS bypass • Coordinate DMA HW Sharing

Open Solution SW Migration

Embedded Middleware Migration

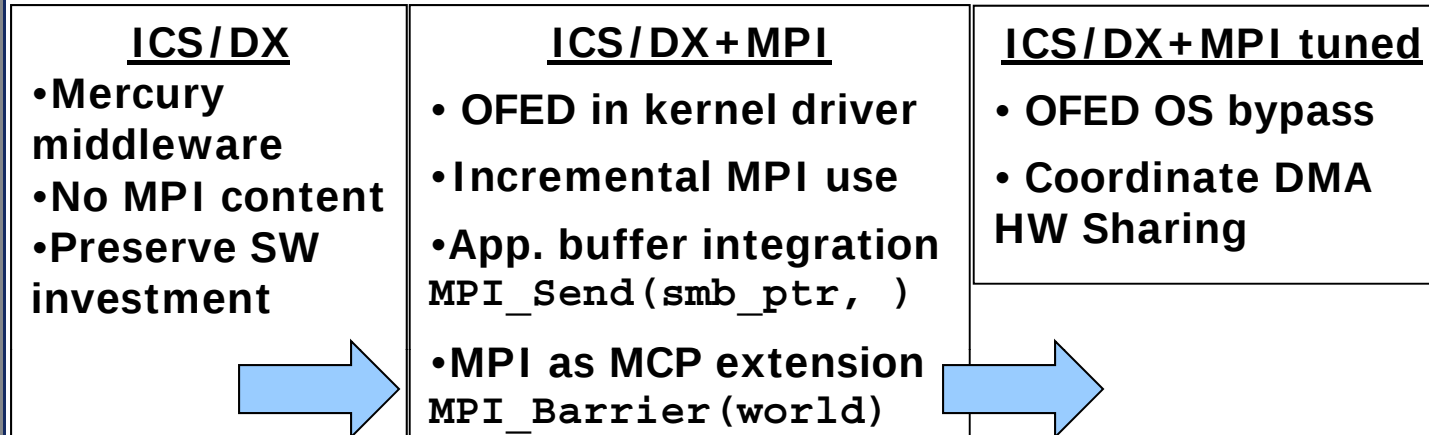
Embedded SW Migration

<u>ICS/DX</u>	<u>ICS/DX+MPI</u>	<u>ICS/DX+MPI tuned</u>
•Mercury middleware •No MPI content •Preserve SW investment 	• OFED in kernel driver •Incremental MPI use •App. buffer integration <code>MPI_Send(smb_ptr,)</code> •MPI as MCP extension <code>MPI_Barrier(world)</code>	• OFED OS bypass • Coordinate DMA HW Sharing

Open Solution SW Migration

Embedded Middleware Migration

Embedded SW Migration



Open Solution SW Migration

Embedded Middleware Migration

Embedded SW Migration

Open Solution SW Migration

MPI (malloc)

- Baseline for port / embedded
- 0-copy RDMA out of the box
- Fabric / OFED
- Intra-node / SAL, OFED

MPI (MPI Alloc mem)

- malloc -> MPI_Alloc_mem
- App still uses open APIs
- MCS “PMEM”, contig. mem.
- Fabric / OFED+PMEM

Embedded Middleware Migration

Embedded SW Migration

Open Solution SW Migration

MPI+DATAFLOW
CONCEPT

Single library,
integrated
resources

DMA HW assist
for MPI, RDMA

Embedded Middleware Migration

Embedded SW Migration

<u>ICS/DX</u>	<u>ICS/DX+MPI</u>	<u>ICS/DX+MPI tuned</u>
•Mercury middleware •No MPI content •Preserve SW investment	• OFED in kernel driver •Incremental MPI use •App. buffer integration <code>MPI_Send(smb_ptr,)</code> •MPI as MCP extension <code>MPI_Barrier(world)</code>	• OFED OS bypass • Coordinate DMA HW Sharing

MPI+DATAFLOW CONCEPT

Single library, integrated resources

DMA HW assist for MPI, RDMA

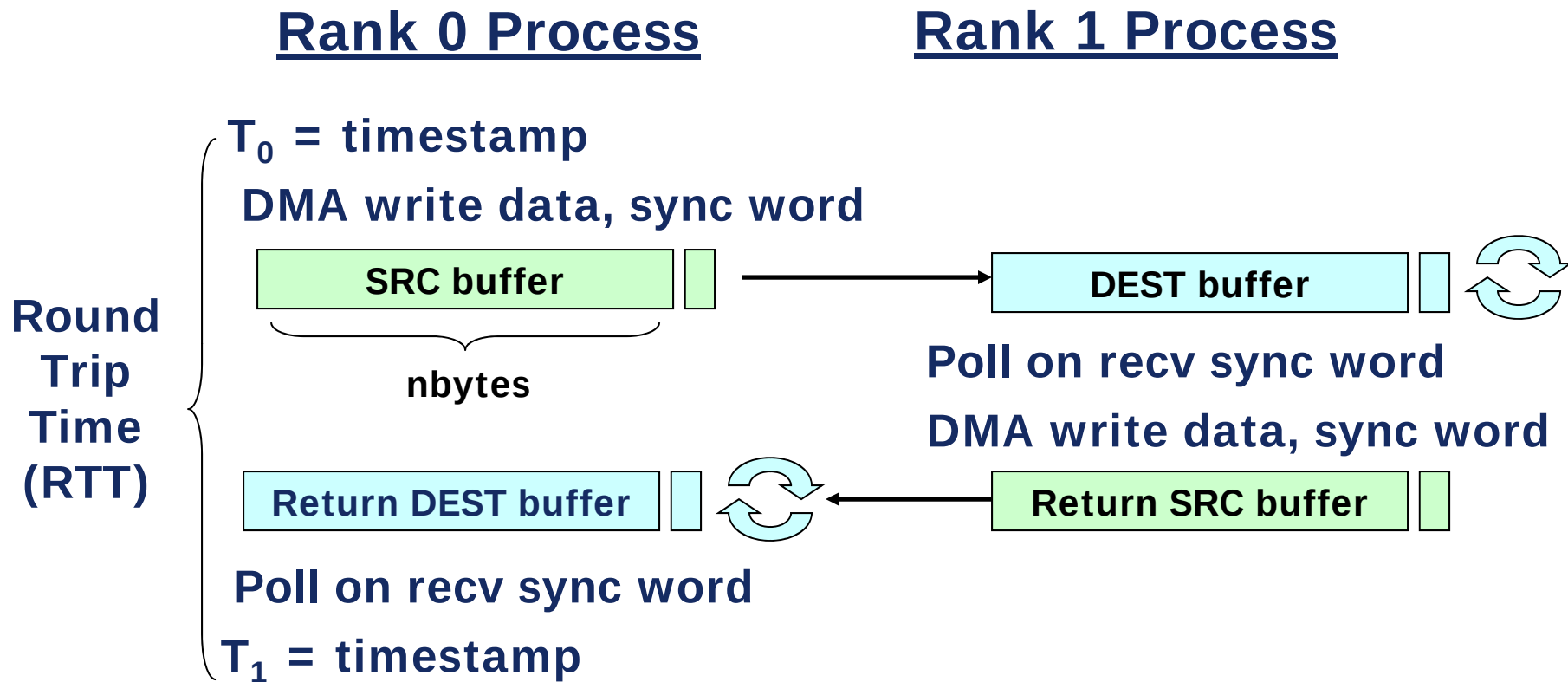
Open Solution SW Migration

<u>MPI (malloc)</u>	<u>MPI (MPI Alloc mem)</u>
•Baseline for port / embedded •0-copy RDMA out of the box •Fabric / OFED •Intra-node / SAL, OFED	•malloc -> MPI_Alloc_mem •App still uses <u>open</u> APIs •MCS “PMEM”, contig. mem. • Fabric / OFED+PMEM



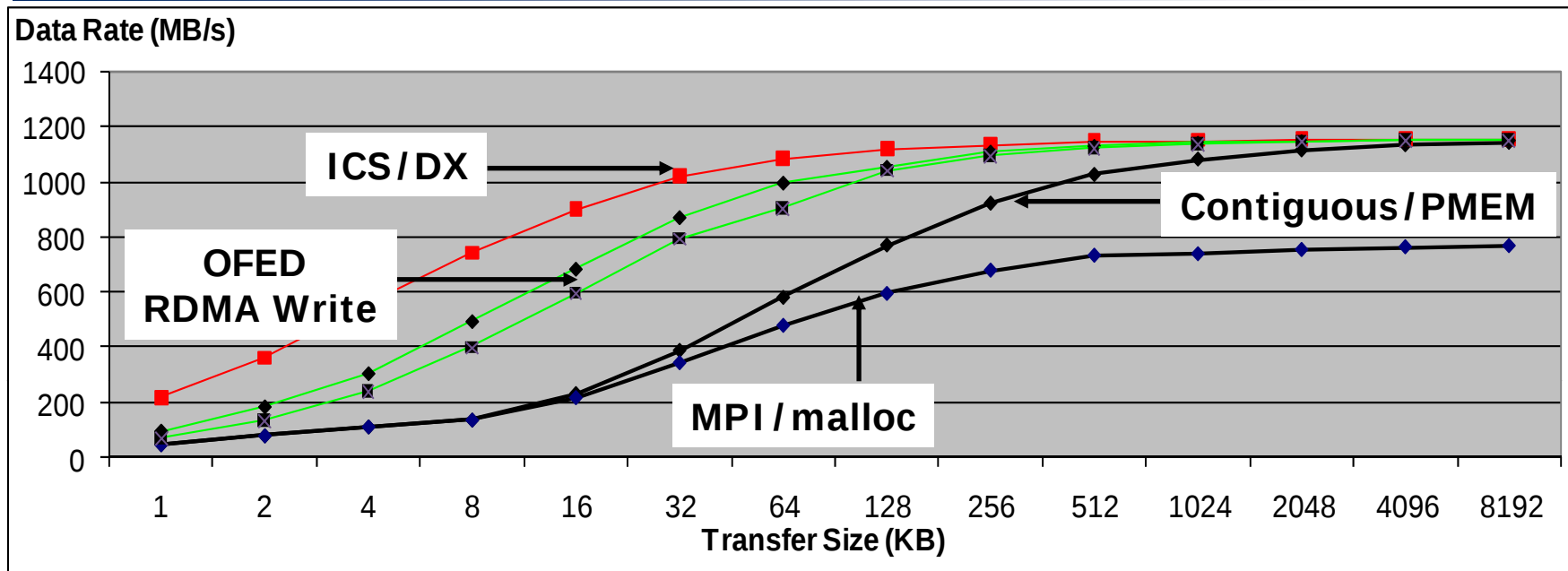
Motivating Example: ICS/DX vs. MPI Data Rate Performance

Benchmark $\frac{1}{2}$ Round Trip Time

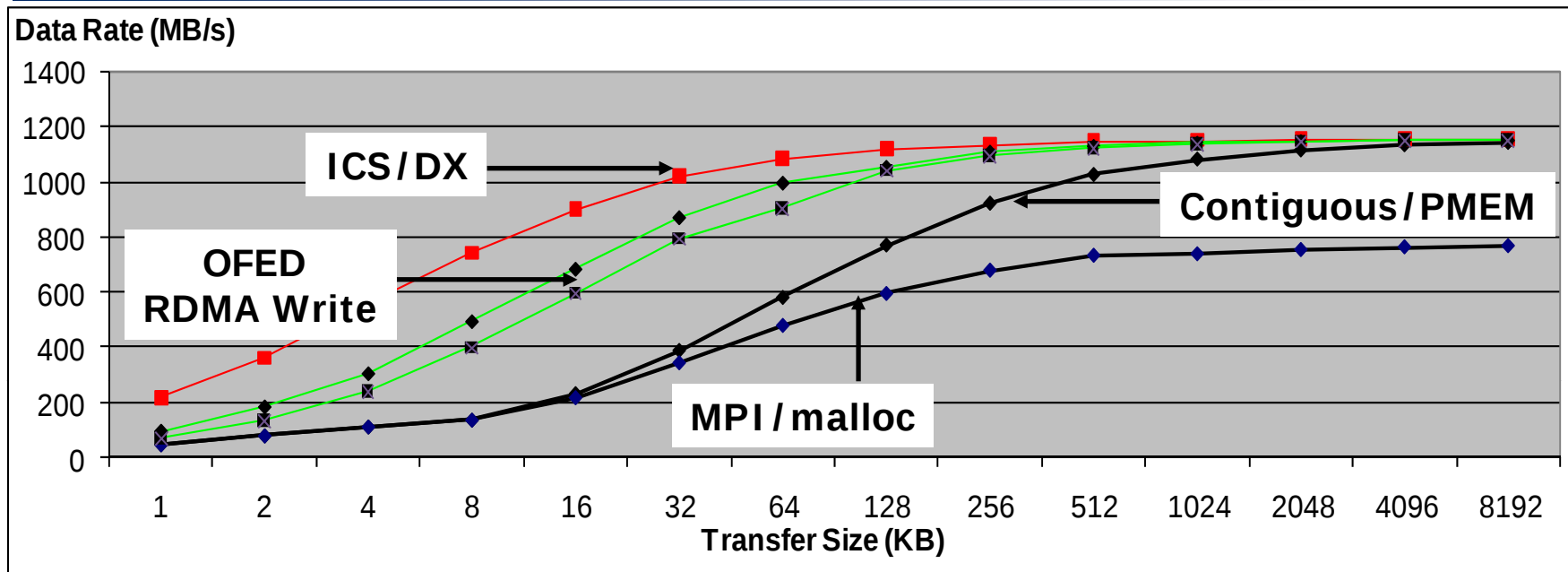


- Inter-node: ranks placed on distinct nodes
- Latency = $\frac{1}{2}$ RTT (record min, max, avg, median)
- Data rate calculated: transfer size, avg latency

ICS/DX, MPI / OFED: 8641D sRIO



ICS/DX, MPI / OFED: 8641D sRIO



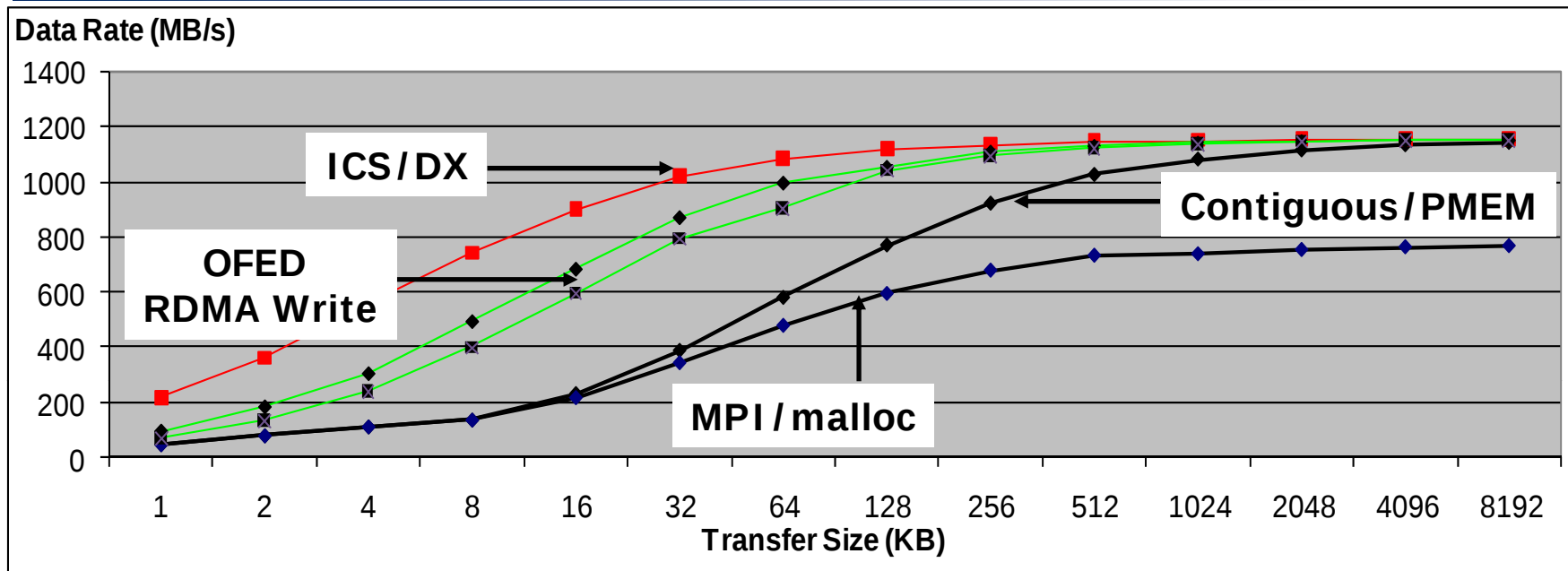
$\frac{1}{2}$ RTT Latency – Small Transfer (4 bytes)

✕ OFED (data+flag) 13 usec

◆ OFED (data/poll last) 8.6 usec

■ ICS/DX (data+flag): 4 usec

ICS/DX, MPI / OFED: 8641D sRIO



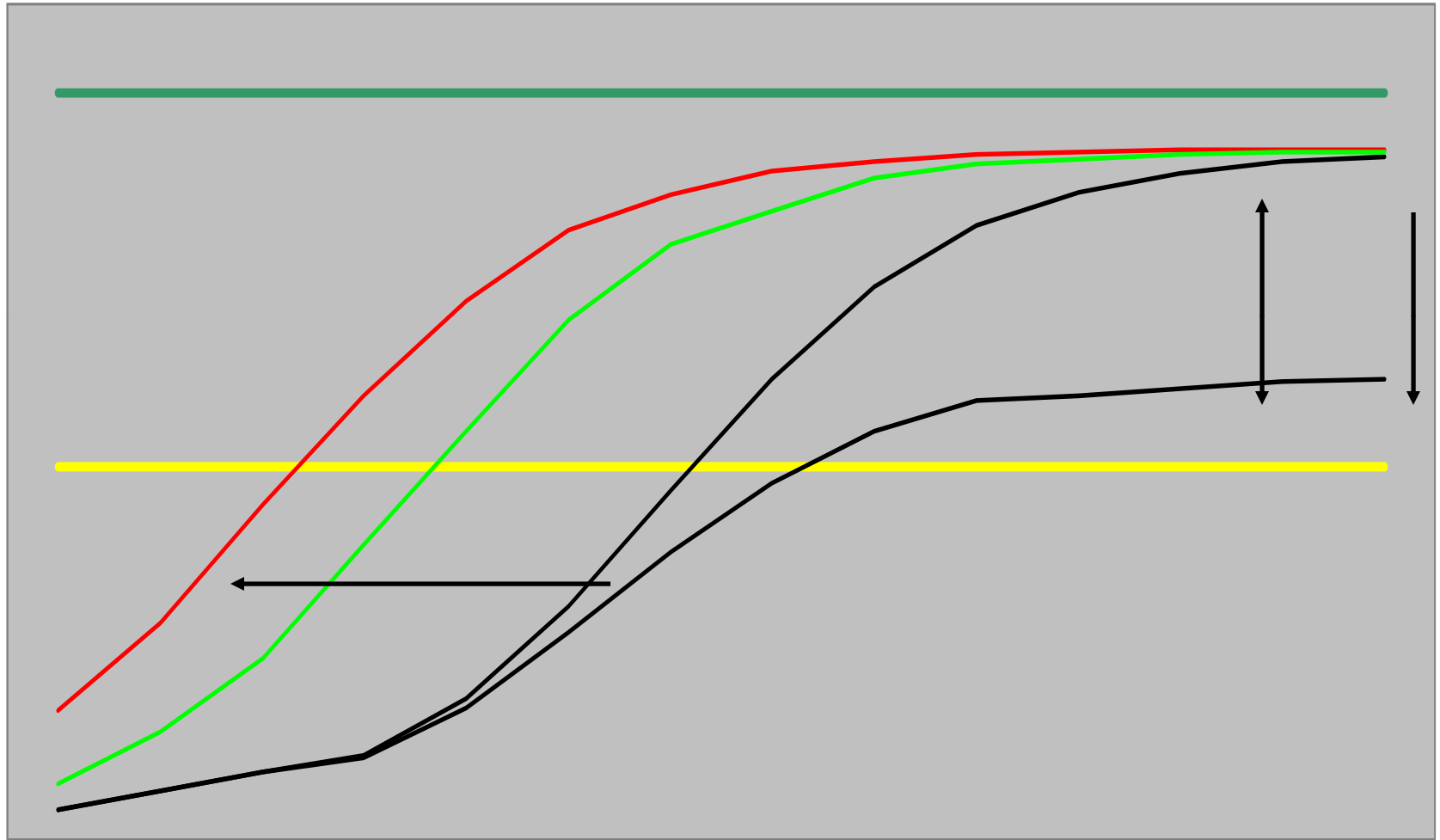
$\frac{1}{2}$ RTT Latency – Small Transfer (4 bytes)

✕ OFED (data+flag) 13 usec ♦ OFED (data/poll last) 8.6 usec ■ ICS/DX (data+flag): 4 usec

- MPI / malloc: good, 0-copy “out of the box”
- Comparable Data Rates (OFED, ICS/DX)
- Overhead Differences
 - Kernel (OFED) vs. user level DMA queuing
 - Dynamic DMA chain formatting (OFED) vs. pre-plan + modify

Preliminary Conclusions

Data Rate



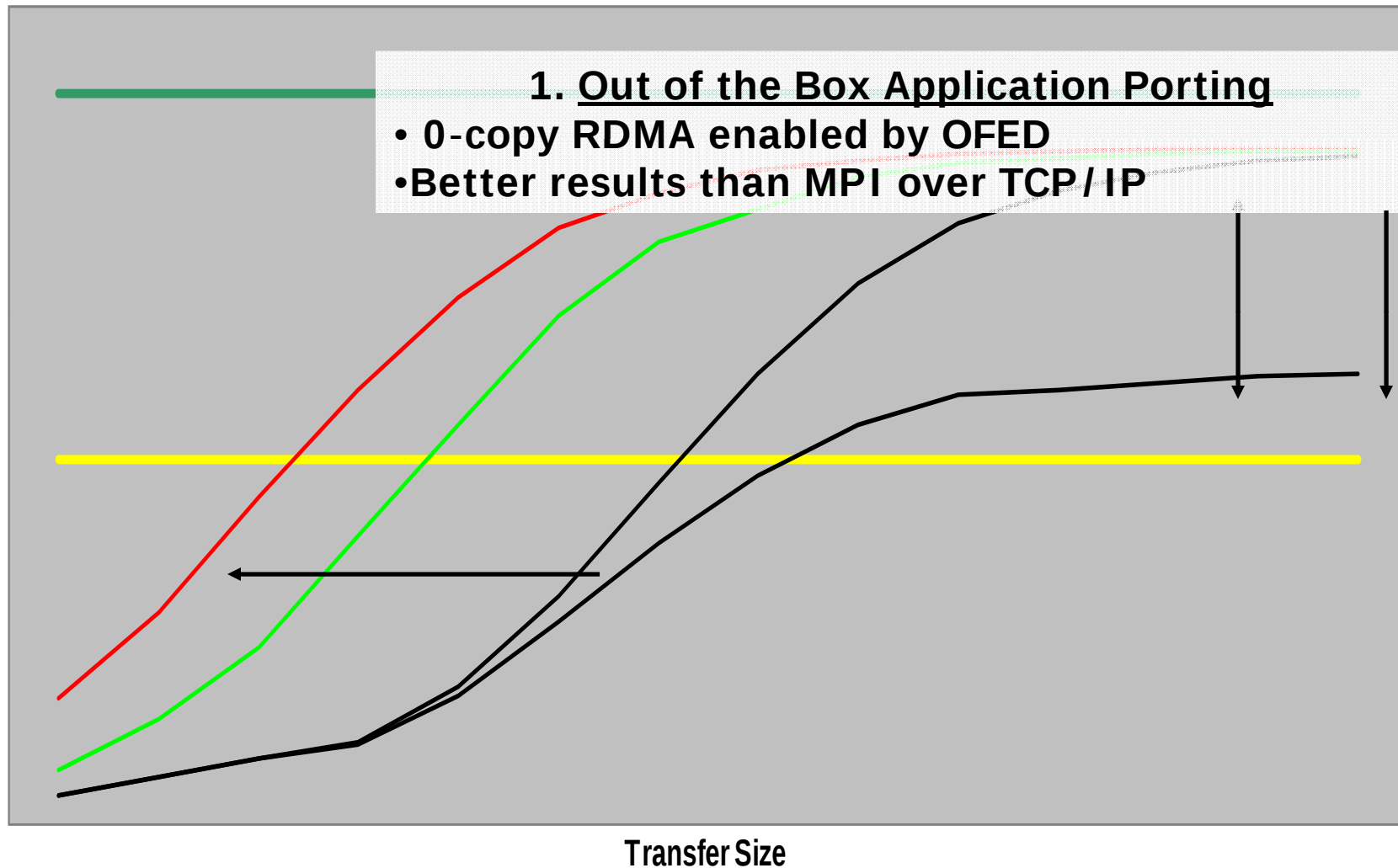
Transfer Size

Preliminary Conclusions

Data Rate

1. Out of the Box Application Porting

- 0-copy RDMA enabled by OFED
- Better results than MPI over TCP/IP



Preliminary Conclusions

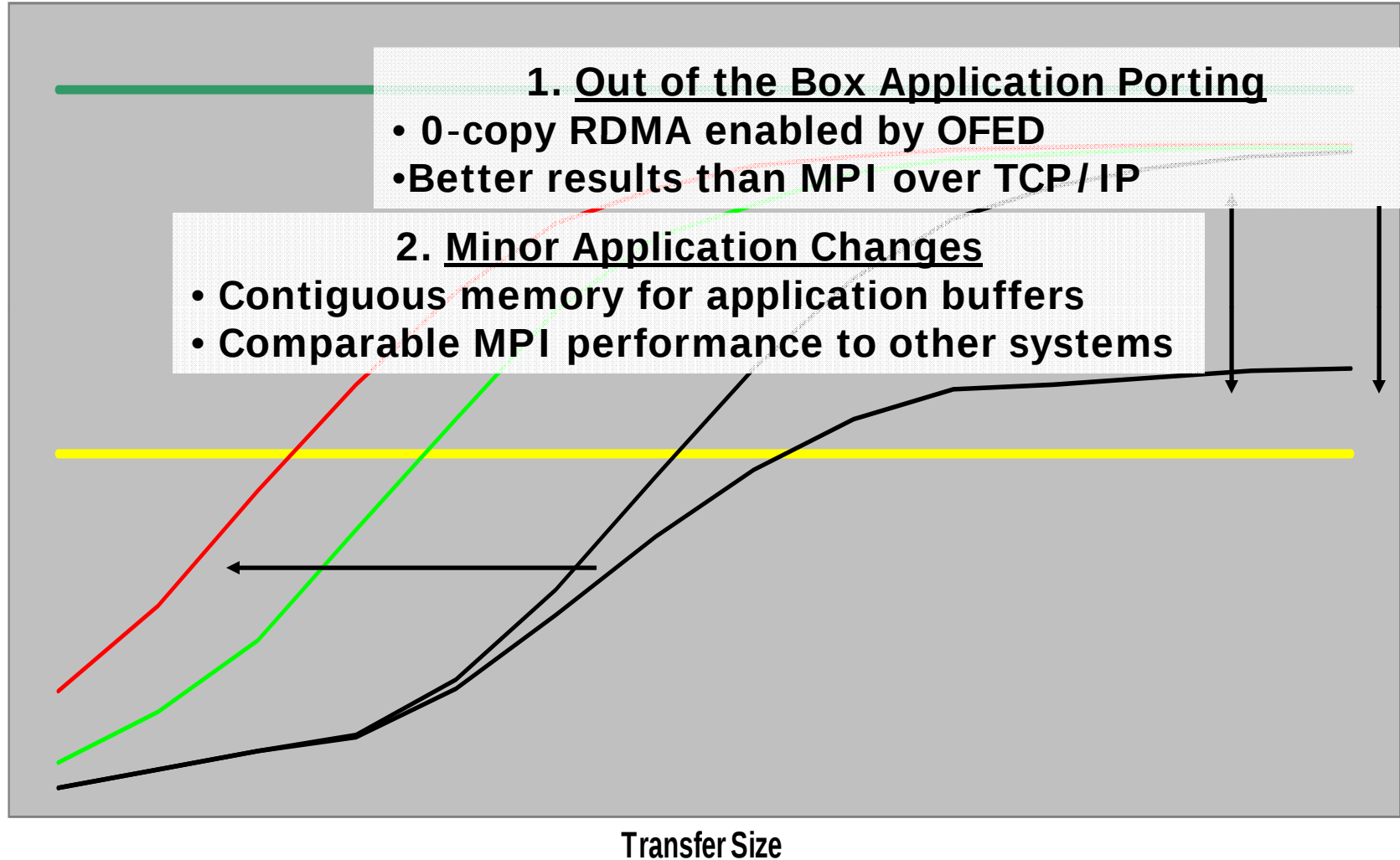
Data Rate

1. Out of the Box Application Porting

- 0-copy RDMA enabled by OFED
- Better results than MPI over TCP/IP

2. Minor Application Changes

- Contiguous memory for application buffers
- Comparable MPI performance to other systems



Preliminary Conclusions

Data Rate

1. Out of the Box Application Porting

- 0-copy RDMA enabled by OFED
- Better results than MPI over TCP/IP

2. Minor Application Changes

- Contiguous memory for application buffers
- Comparable MPI performance to other systems

3. Improvements Underway

- OFED: user-space optimized DMA queuing
- Moving performance curves “to the left”

Transfer Size

Preliminary Conclusions

Data Rate

1. Out of the Box Application Porting

- 0-copy RDMA enabled by OFED
- Better results than MPI over TCP/IP

2. Minor Application Changes

- Contiguous memory for application buffers
- Comparable MPI performance to other systems

3. Improvements Underway

- OFED: user-space optimized DMA queuing
- Moving performance curves “to the left”

4. Communication Model Differences

- RDMA vs. message passing
- Limited “gap closing” using traditional means
 - Different solution approaches to be considered
- OFED “middleware friendly” RDMA

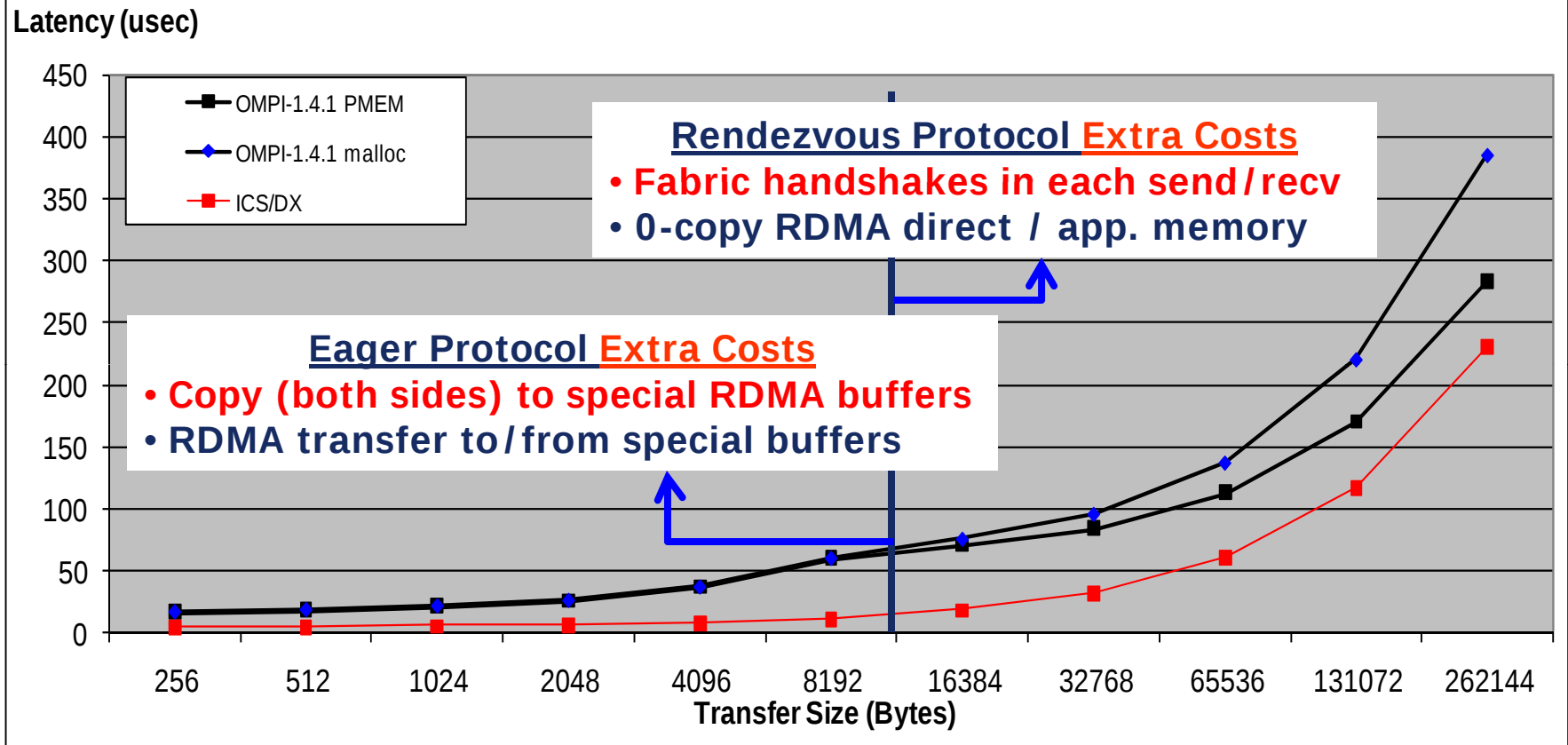
Transfer Size



Communication Model Comparison

Message Passing and RDMA

Message Passing vs. RDMA



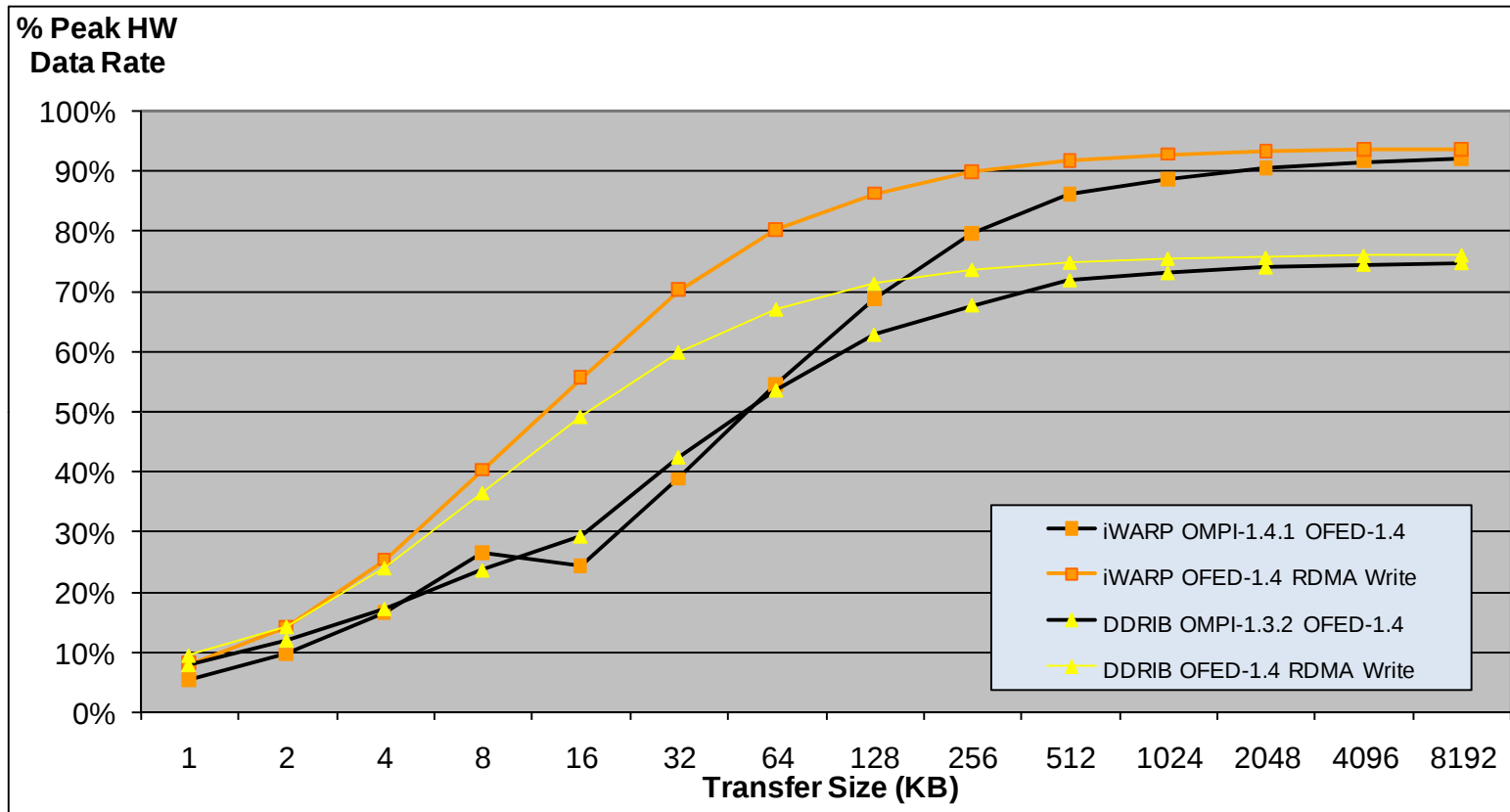
- Flexibility (MPI) vs. Performance (RDMA)
- Issue: memory registration / setup for RDMA
- MPI Tradeoffs: Copy vs. Fabric Handshake Costs



MPI and RDMA Performance

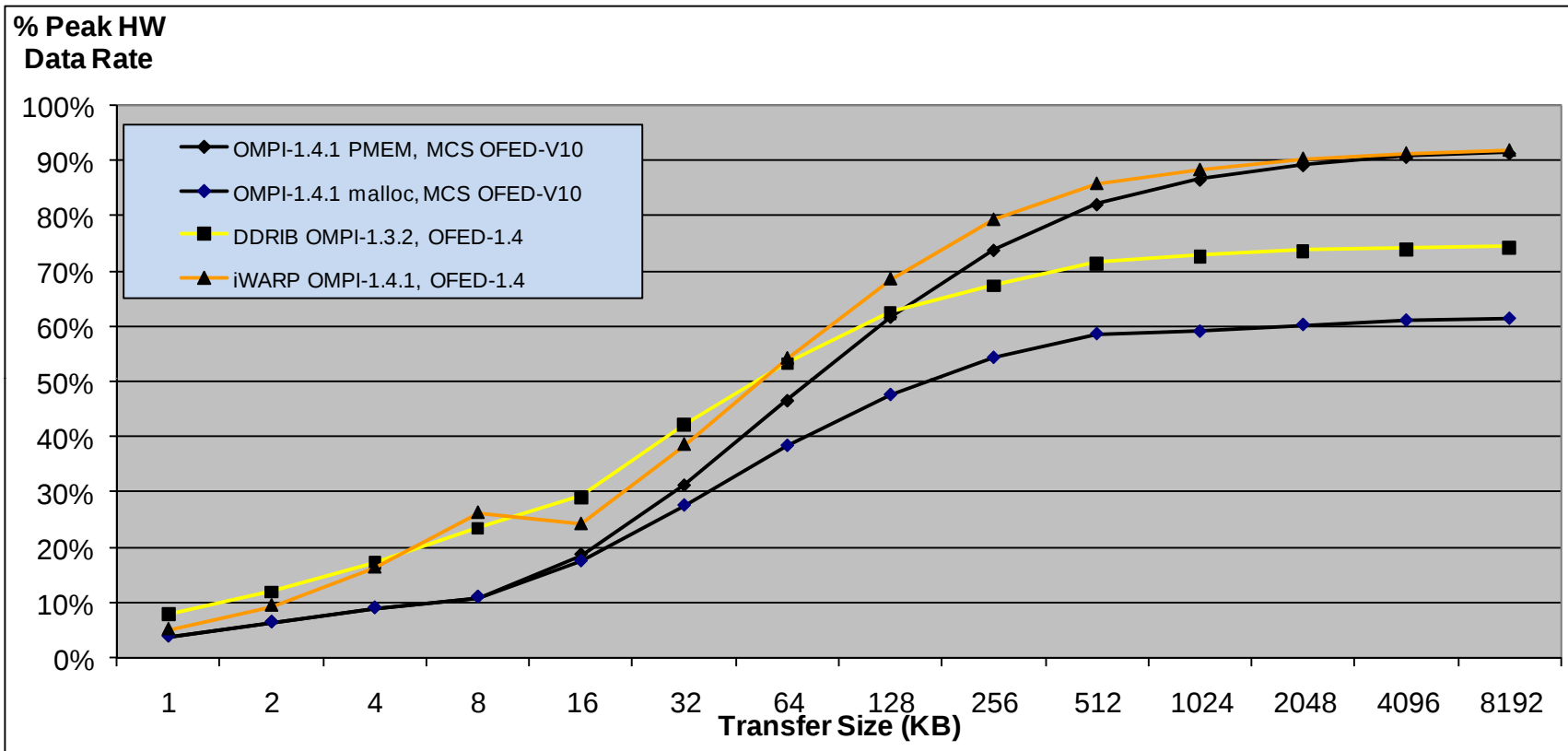
Server / HPC Platform Comparisons

MPI vs. OFED Data Rate: IB, iWARP



- **Without sRIO (Same Software)**
 - Double Data Rate InfiniBand (DDRIB), iWARP (10GbE)
- **Similar Performance Delta Observed**
- **% of peak rate – link speeds: 10, 16 Gbps**

MPI Data Rate: sRIO, IB, iWARP

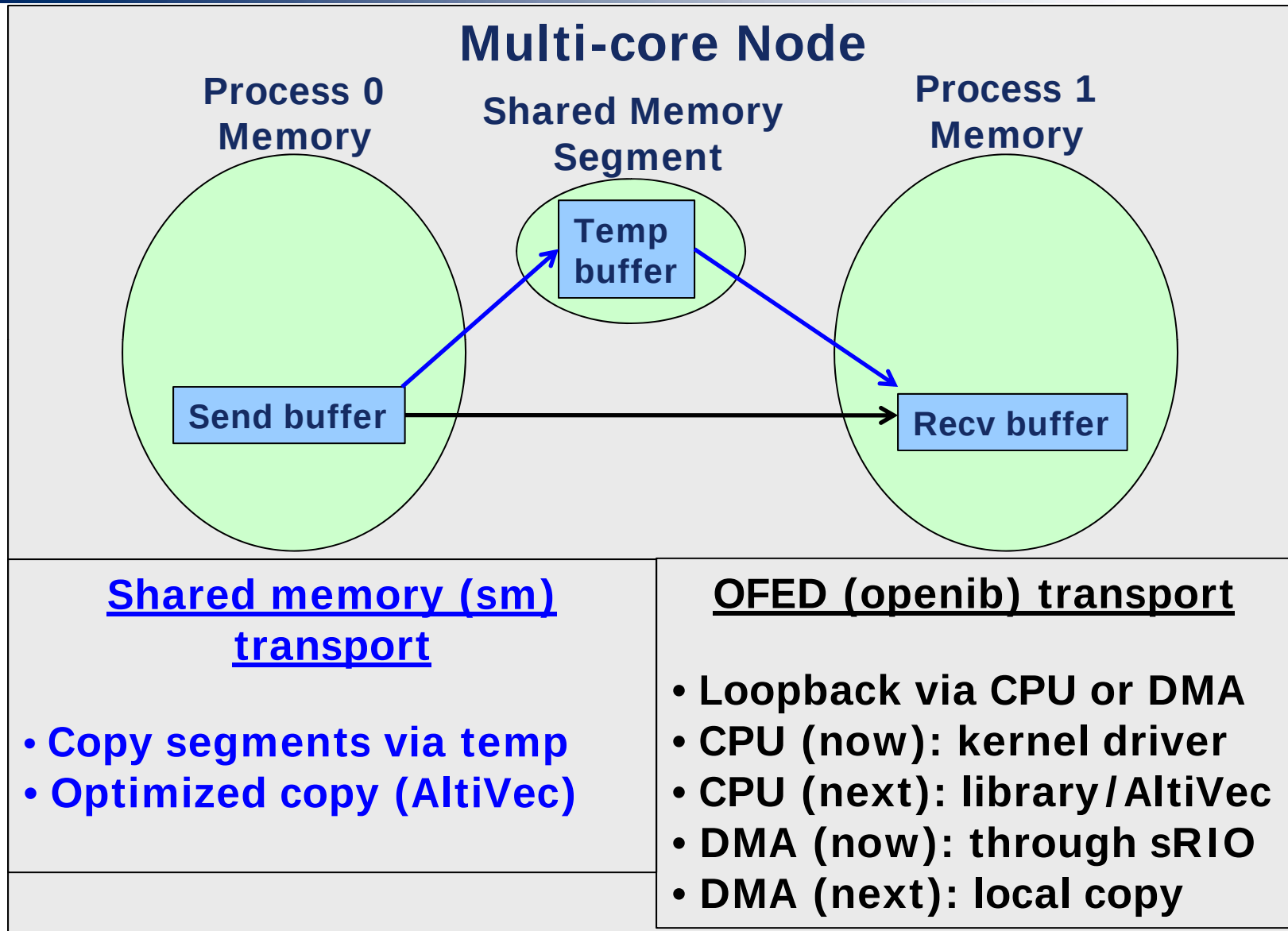


- MPI Data Rate Comparable Across Fabrics
- Small Transfer Copy Cost Differences
 - X86_64 (DDRIB, iWARP) faster than PPC32 (sRIO)

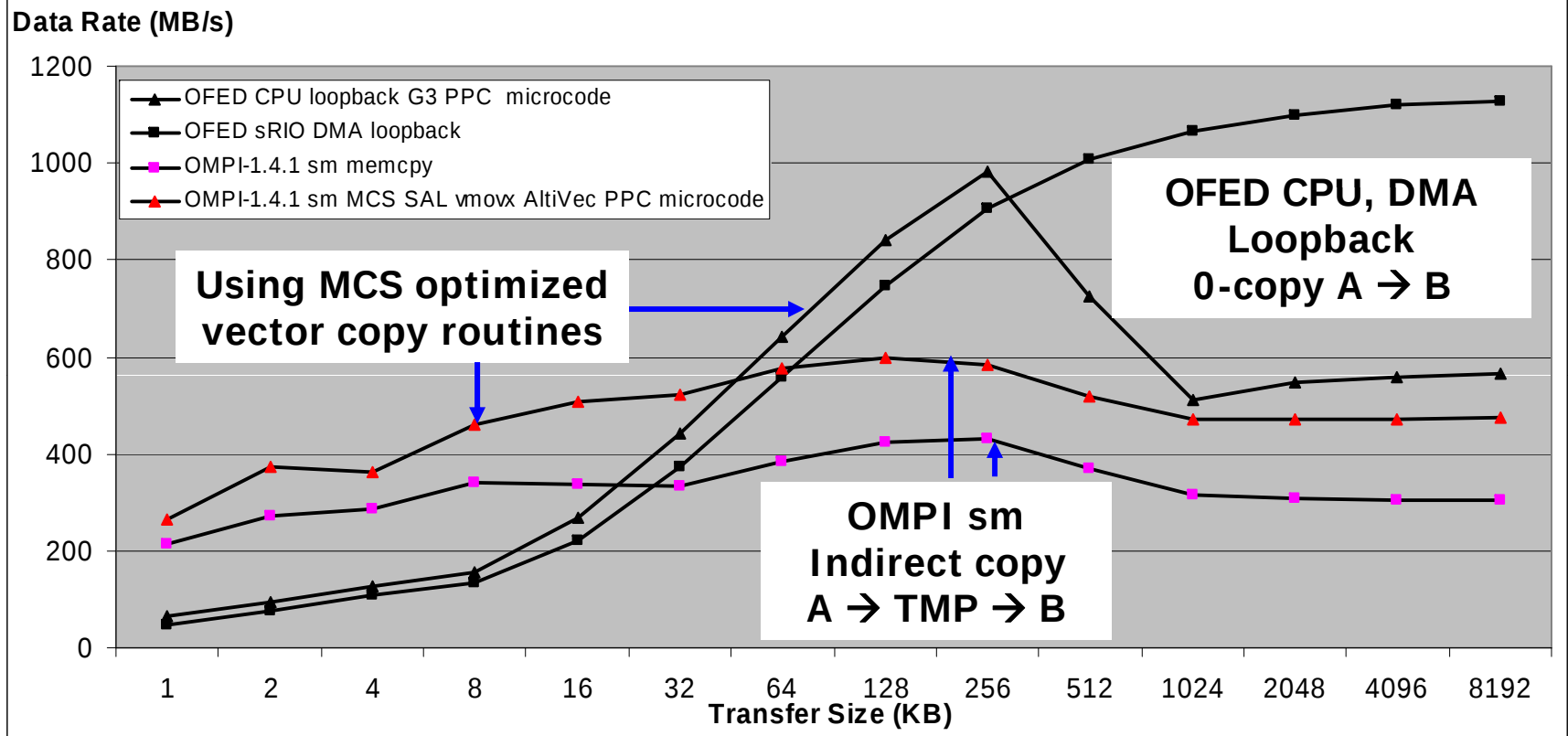


Mercury MPI / OFED Intra-Node Performance

Open MPI Intra-Node Communication



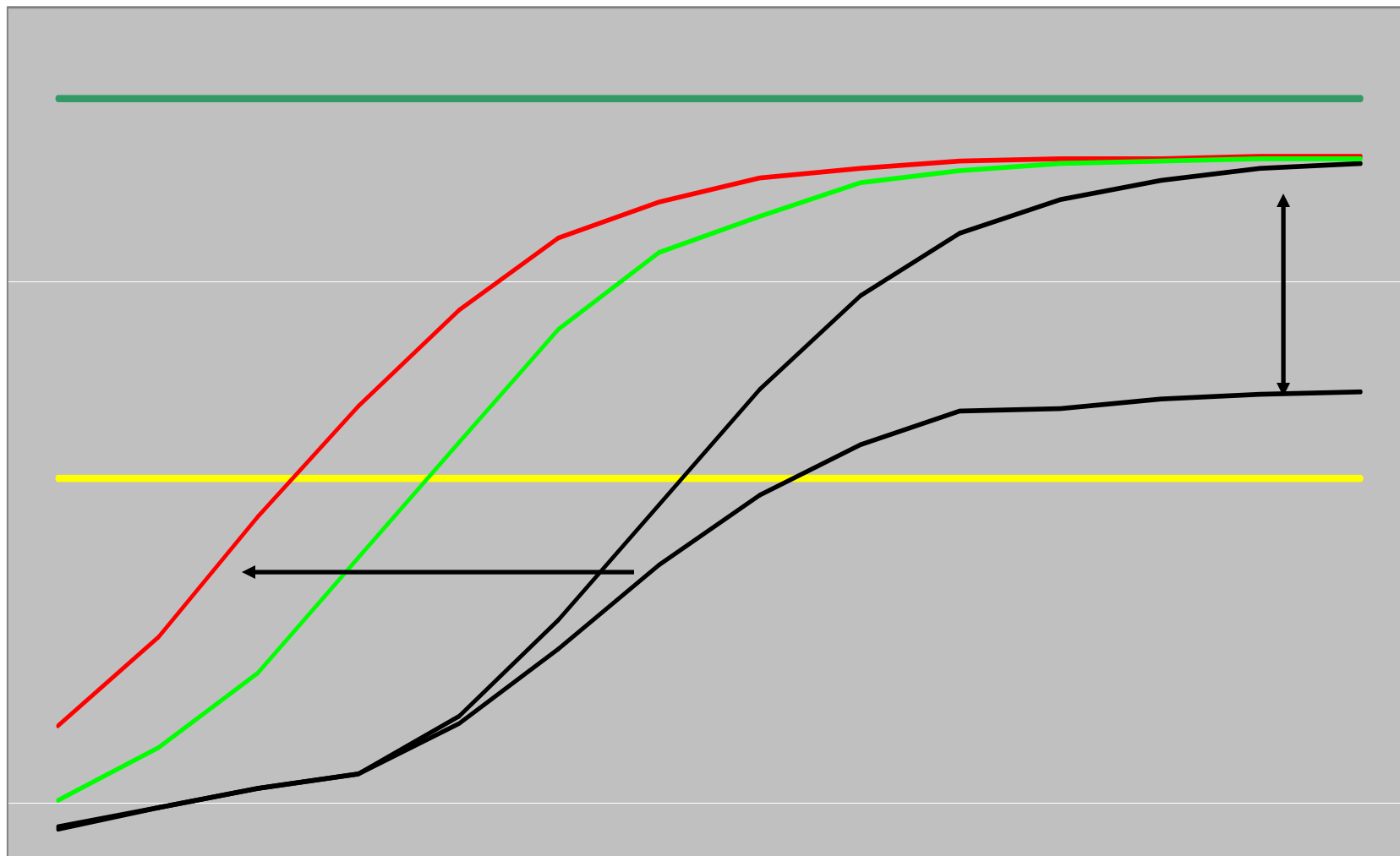
MPI Intra-Node Data Rate



- **OMPI sm latency better (very small messages)**
 - 2-3 usec latency (OMPI sm) vs. 11 usec (OFED CPU loopback)
- **OFED loopback better for large messages**
 - Direct copy – 50% less memory bandwidth used

Summary / Conclusions

Data Rate



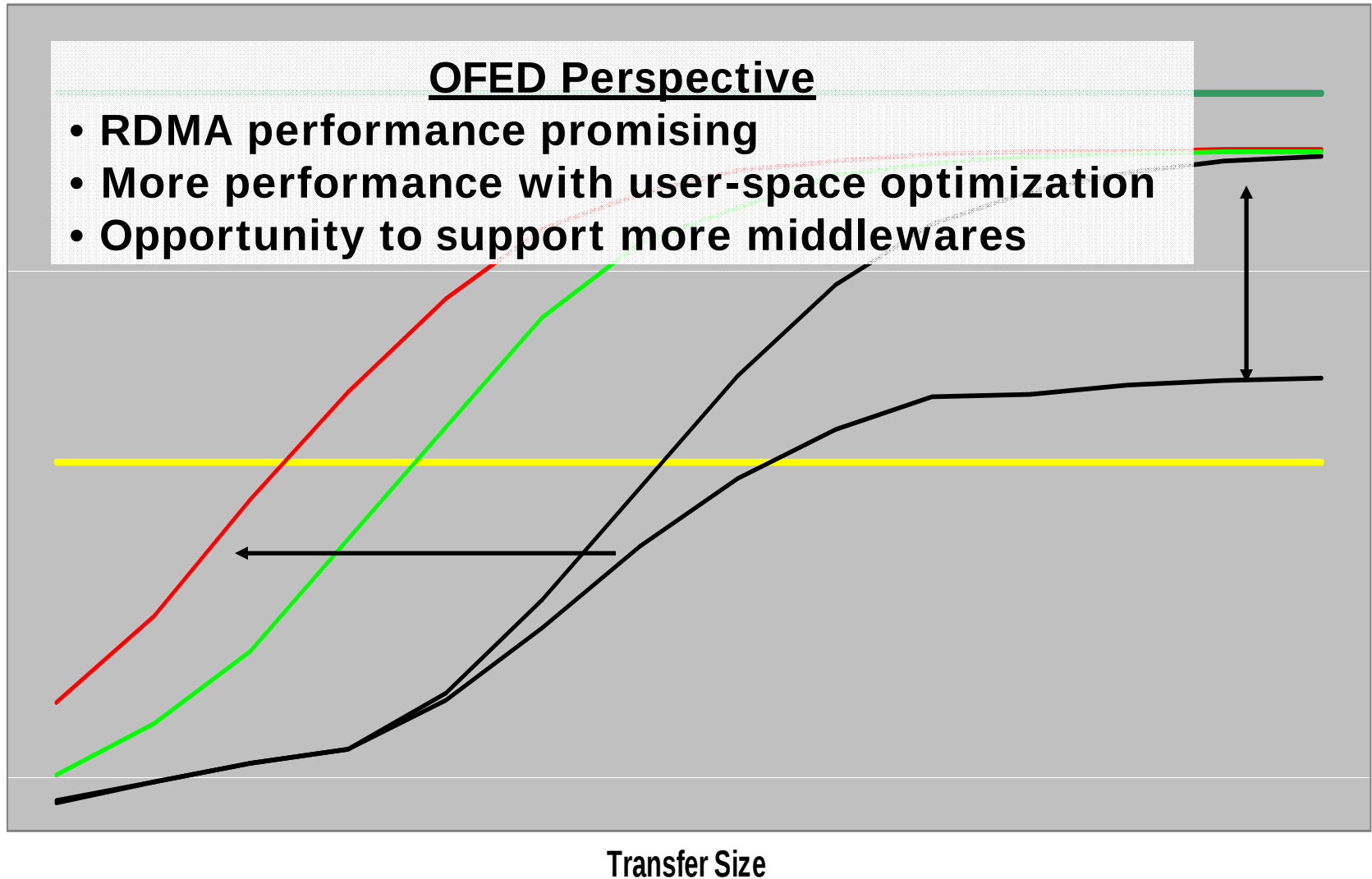
Transfer Size

Summary / Conclusions

Data Rate

OFED Perspective

- RDMA performance promising
- More performance with user-space optimization
- Opportunity to support more middlewares



Transfer Size

Summary / Conclusions

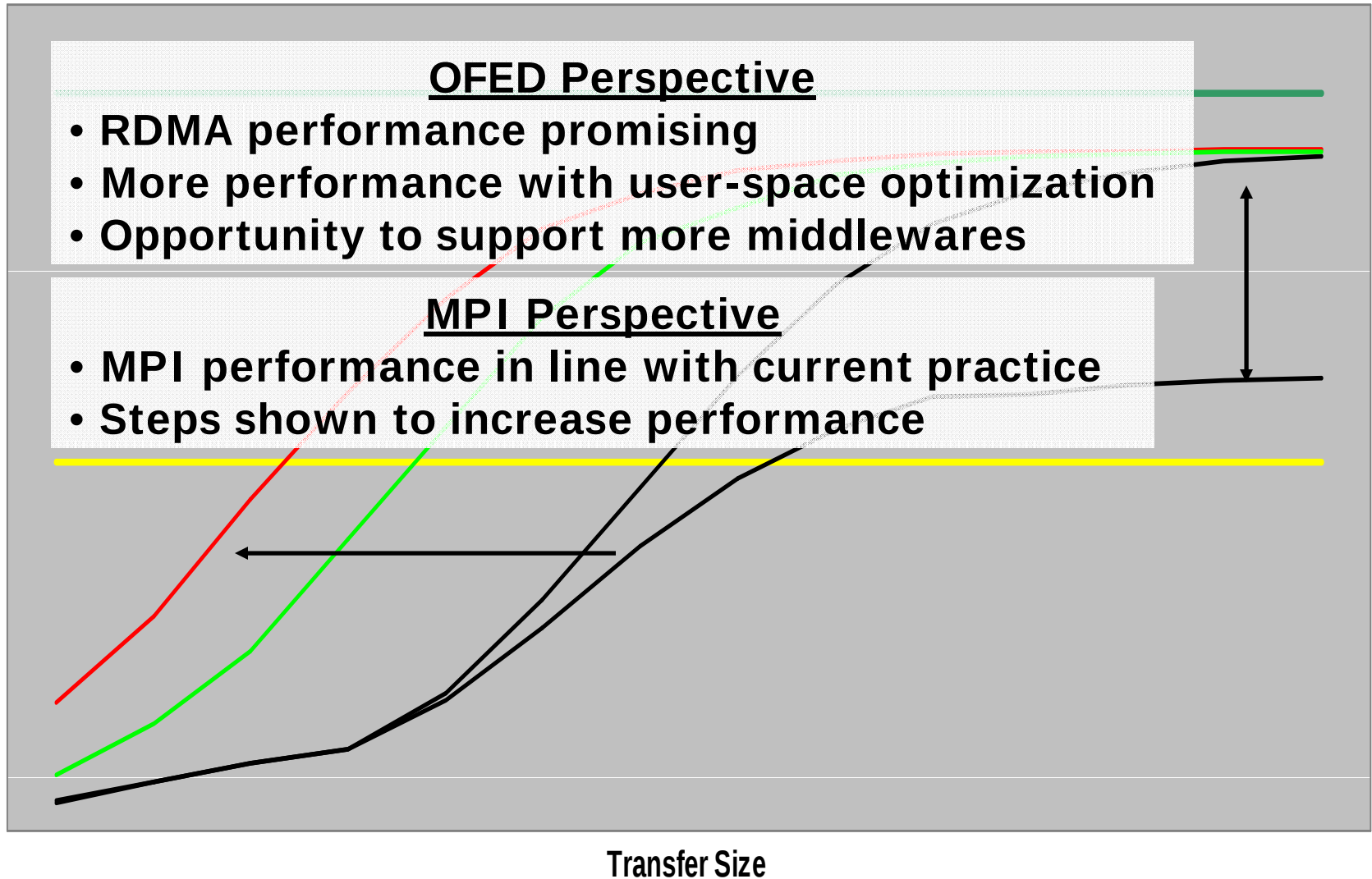
Data Rate

OFED Perspective

- RDMA performance promising
- More performance with user-space optimization
- Opportunity to support more middlewares

MPI Perspective

- MPI performance in line with current practice
- Steps shown to increase performance



Transfer Size

Summary / Conclusions

Data Rate

OFED Perspective

- RDMA performance promising
- More performance with user-space optimization
- Opportunity to support more middlewares

MPI Perspective

- MPI performance in line with current practice
- Steps shown to increase performance

Enhancing MPI Perspective

- Converged middleware / extensions end goal
- Vs. pure MPI end goal
- HW assist likely can help in either case

Transfer Size



Thank You Questions?

Kenneth Cain
kcain@mc.com