MERCURY COMPUTER SYSTEMS™

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

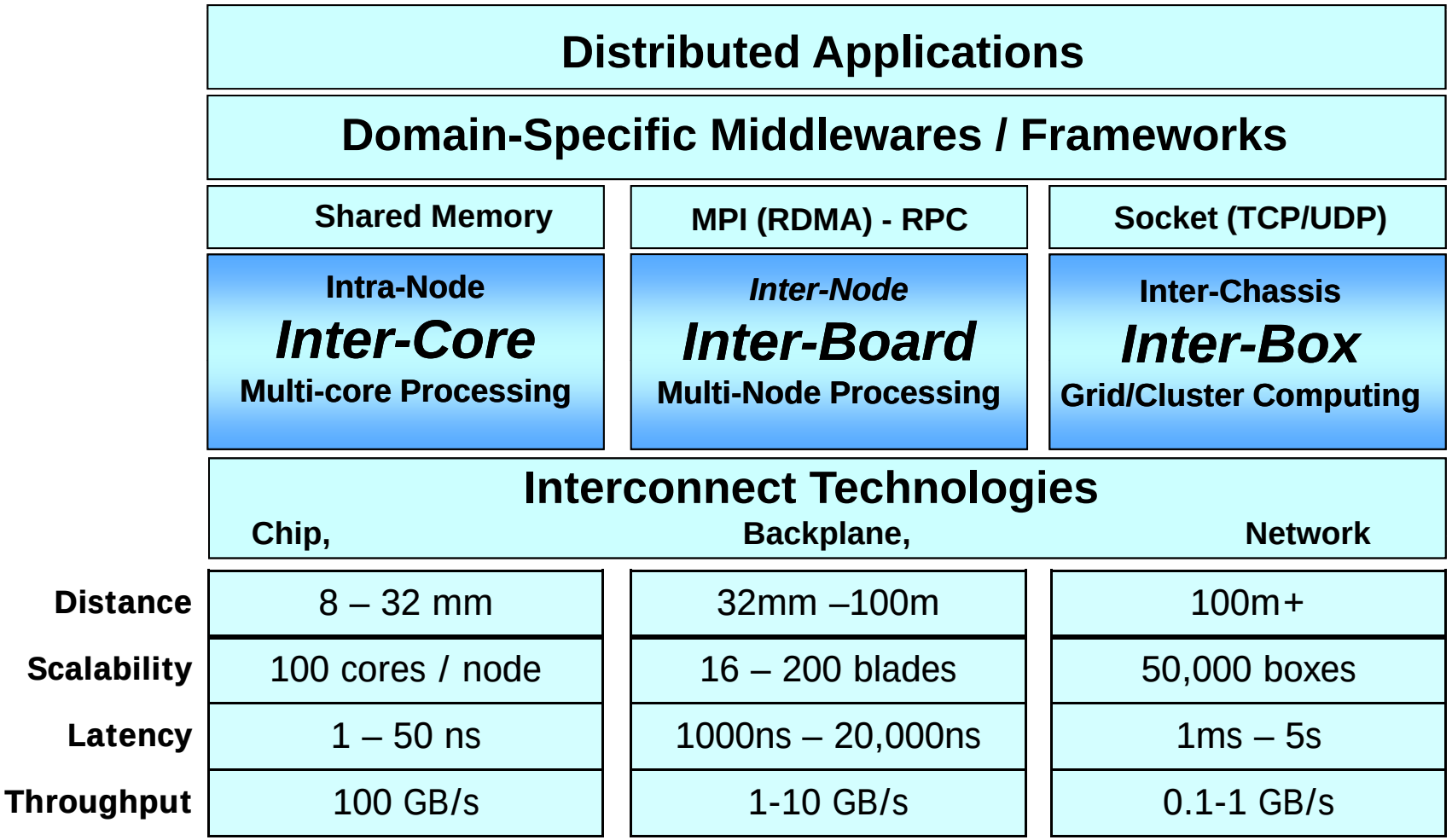
- **Middleware has been a significant focus for the HPEC community for a long time**

MERCURY COMPUTER SYSTEMS™

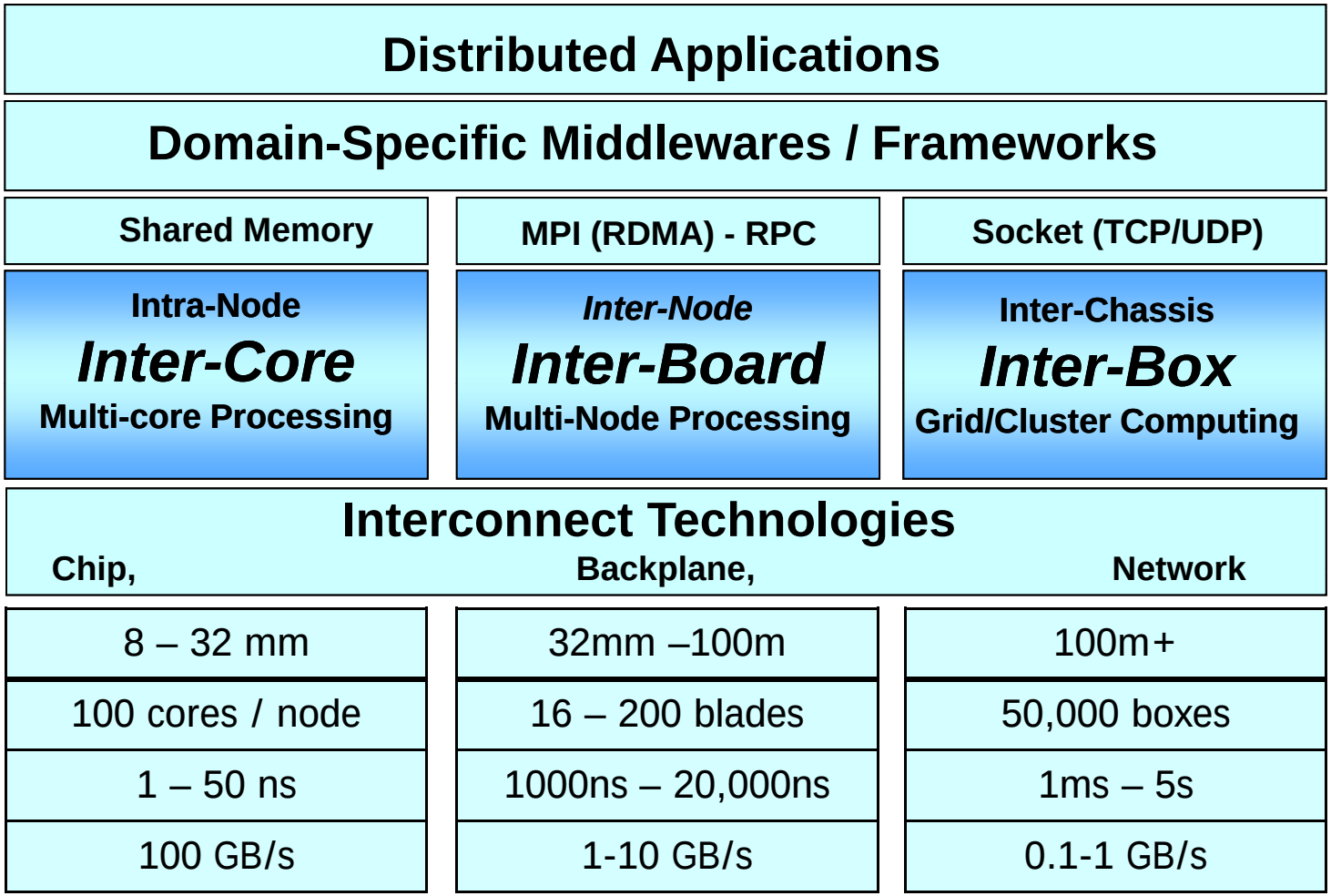
Performance Migration to Open Solutions

High Level Perspective

Scalability Reference Model

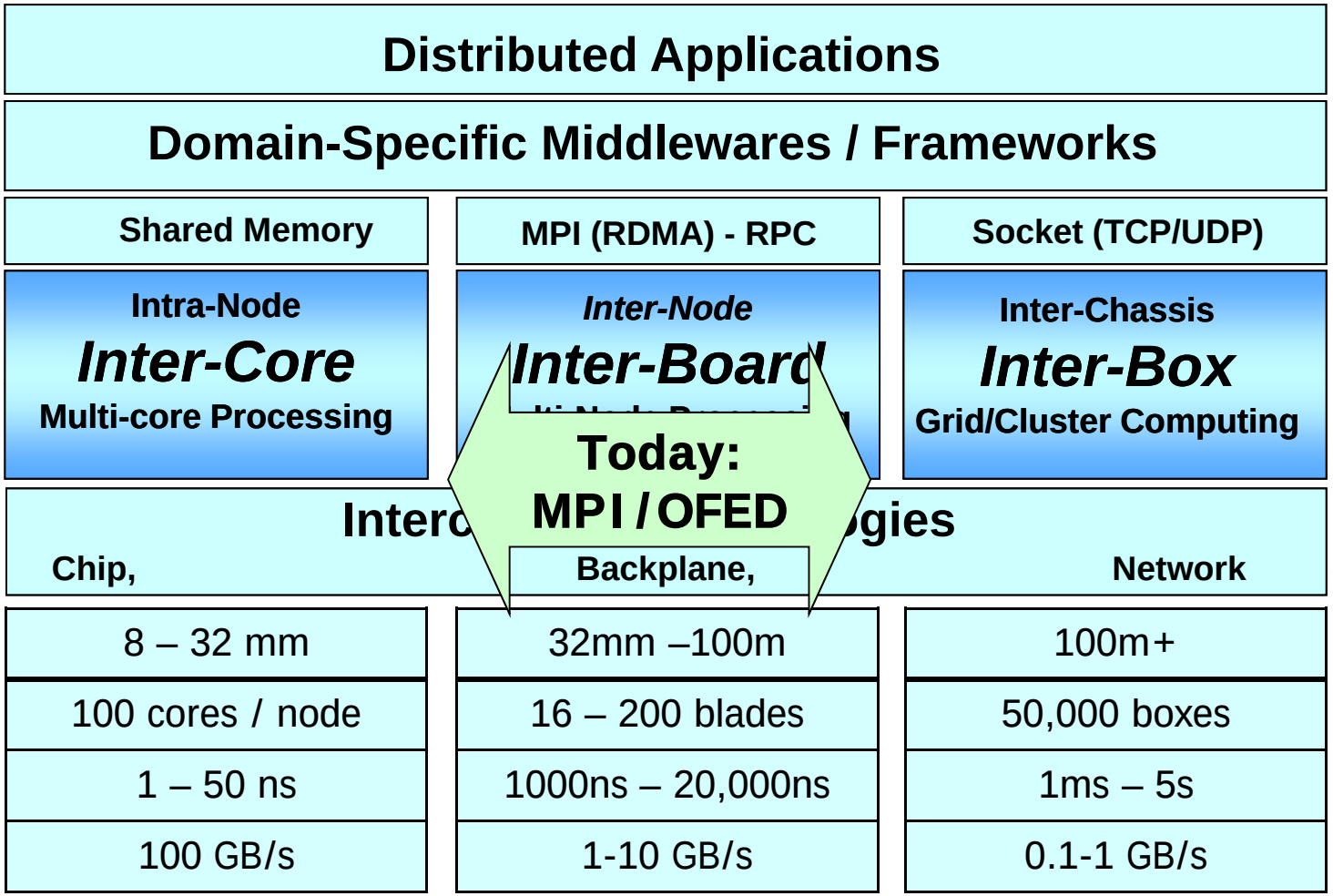


Scalability Reference Model



- **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		• 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	

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	

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	

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**

Fabric Software Migrations and Goals

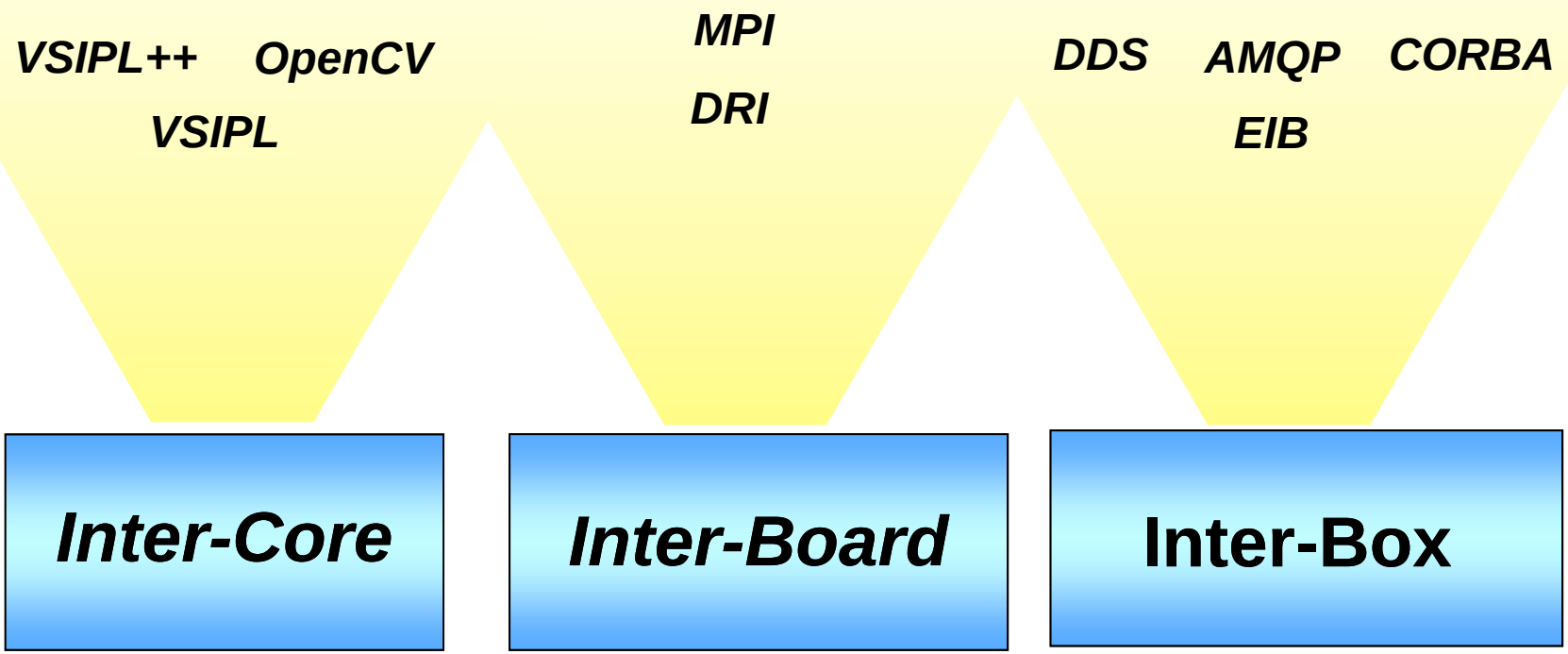
Transitions		Goals, Potential Solutions
Compute		• Establish Open Baseline in PPC • Only Recompile if Migrating to x86
PPC,FPGA	x86, GPGPU, FPGA	
Interconnects and Data Rates		• Common RDMA API for All Fabrics • SW Binary Compatible among Fabrics • RDMA+LAN APIs for Selected Fabric • Programmable Fabric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



Industry Scaling Solutions

VSIPL++ *OpenCV*

VSIPL

MPI

DRI

DDS *AMQP* *CORBA*

EIB

Middlewares , 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*

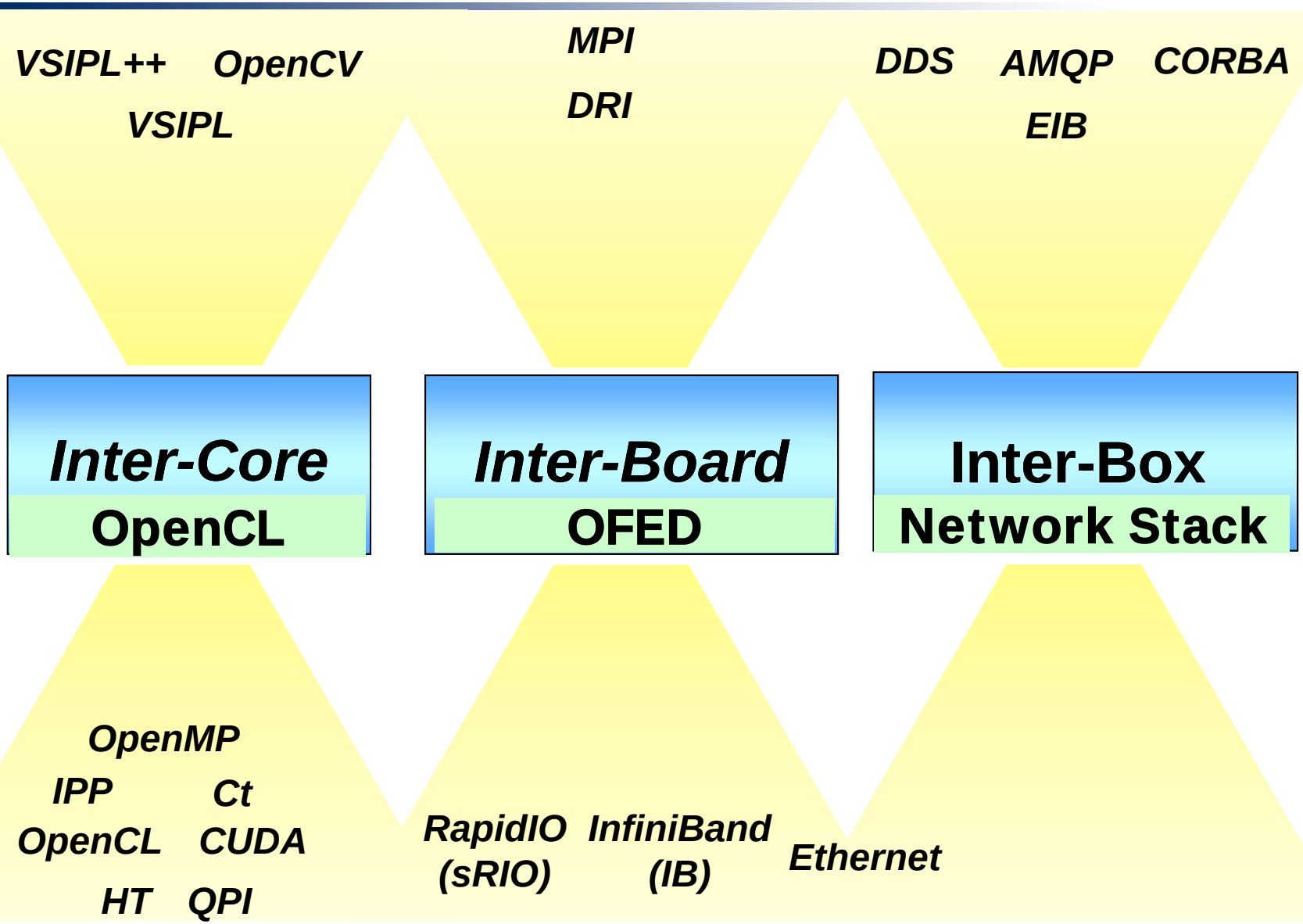
Middlewares , Application Frameworks
Mercury, 3rd Party, Customer Supplied....

Inter-Core	*Inter-Board*	Inter-Box

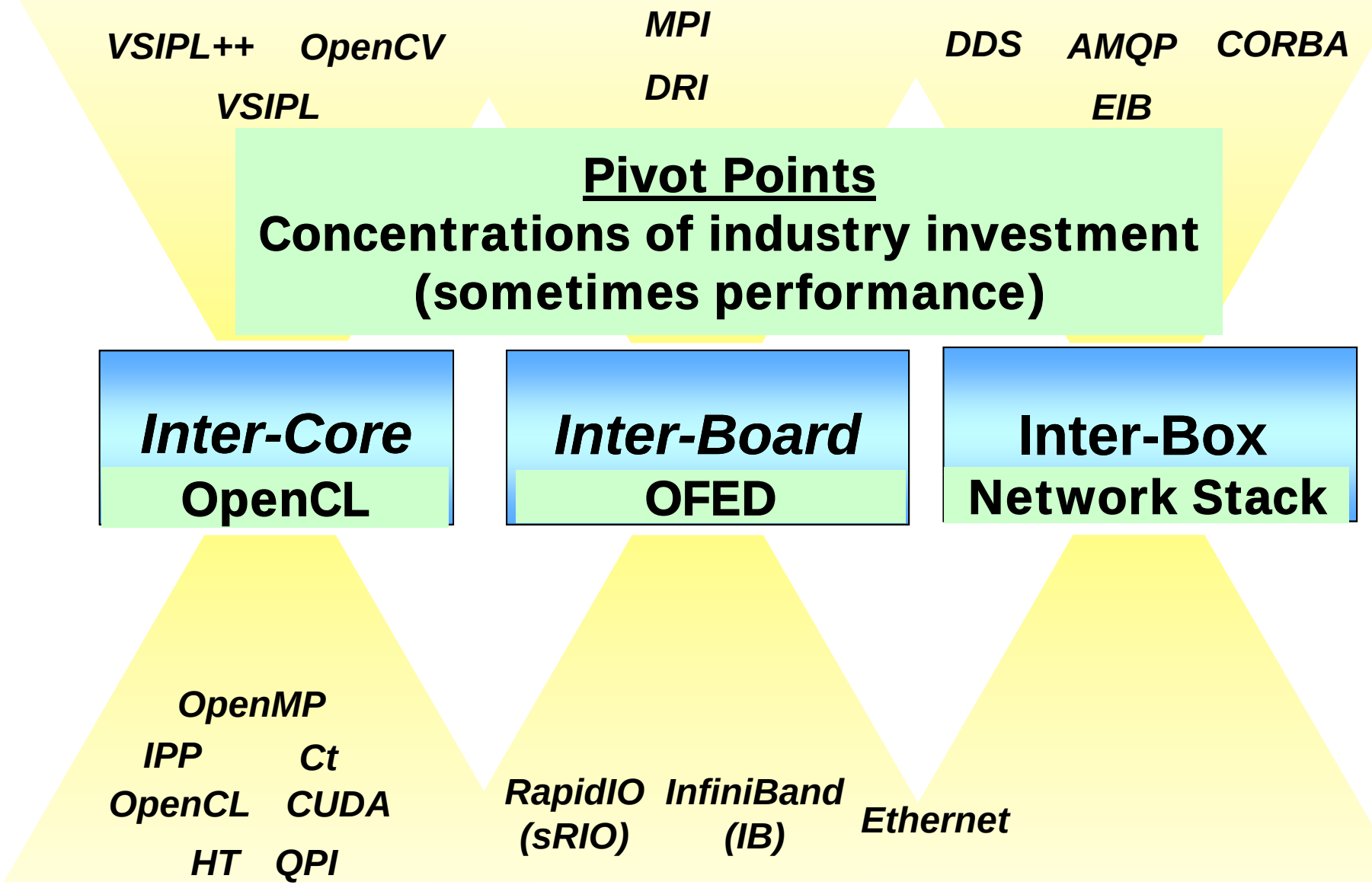
Middlewares Engage Platform Services for Performance and Scalability

OpenMP *IPP* *Ct* *OpenCL* *CUDA* *HT* *QPI* *RapidIO (sRIO)* *InfiniBand (IB)* *Ethernet*

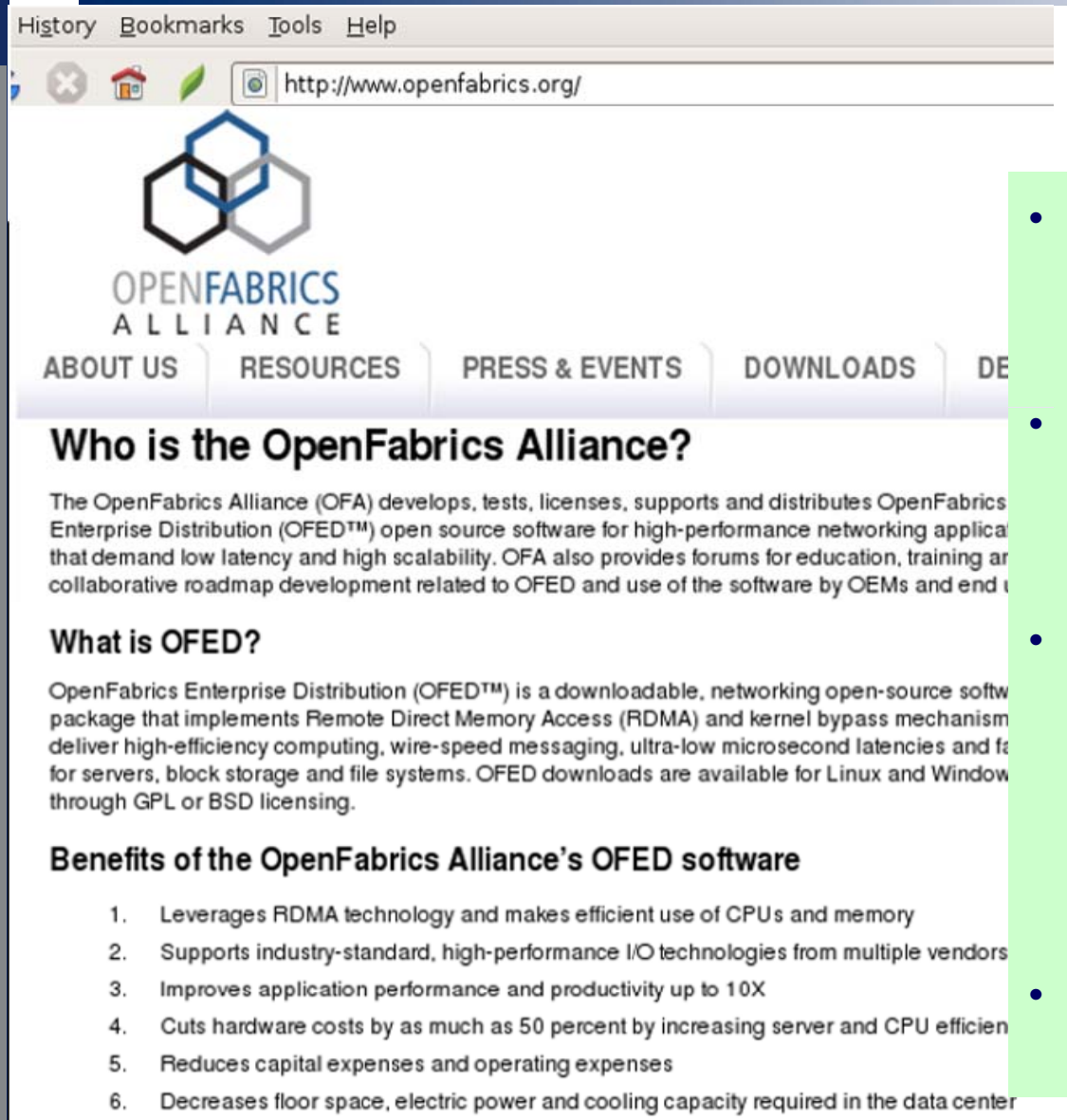
Industry Scaling Solutions



Industry Scaling Solutions



What is OFED?



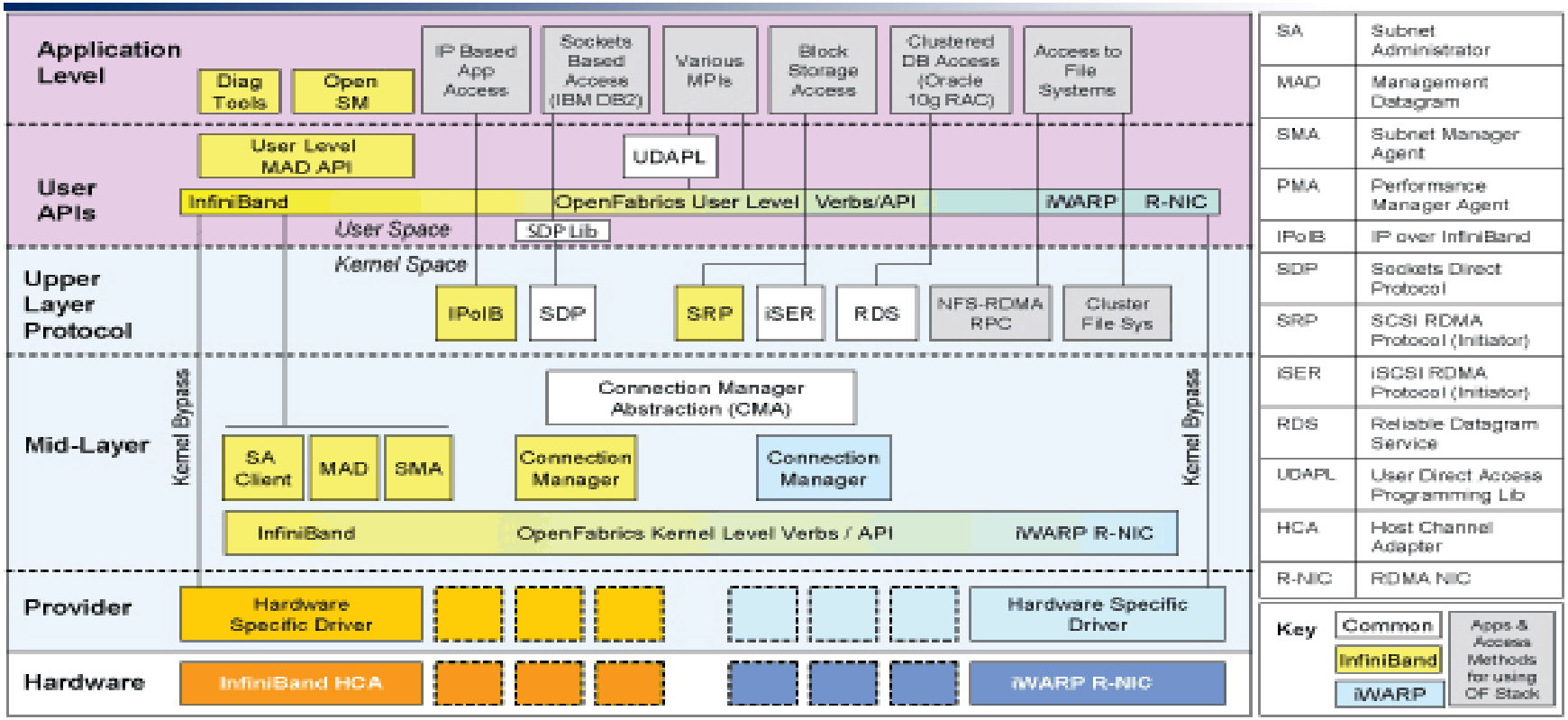
- **Open Source Software by The OpenFabrics Alliance**
  - Mercury is a member
- **Multiple Fabric/Vendor**
  - Ethernet/IB, sRIO (this work)
  - "Offload" HW Available
- **Multiple Use**
  - MPI (multiple libraries)
  - RDMA: uDAPL, verbs
  - Sockets
  - Storage: block, file
- **High Performance**
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

OFA – Significant Membership Roster

- See http://www.openfabrics.org for the list

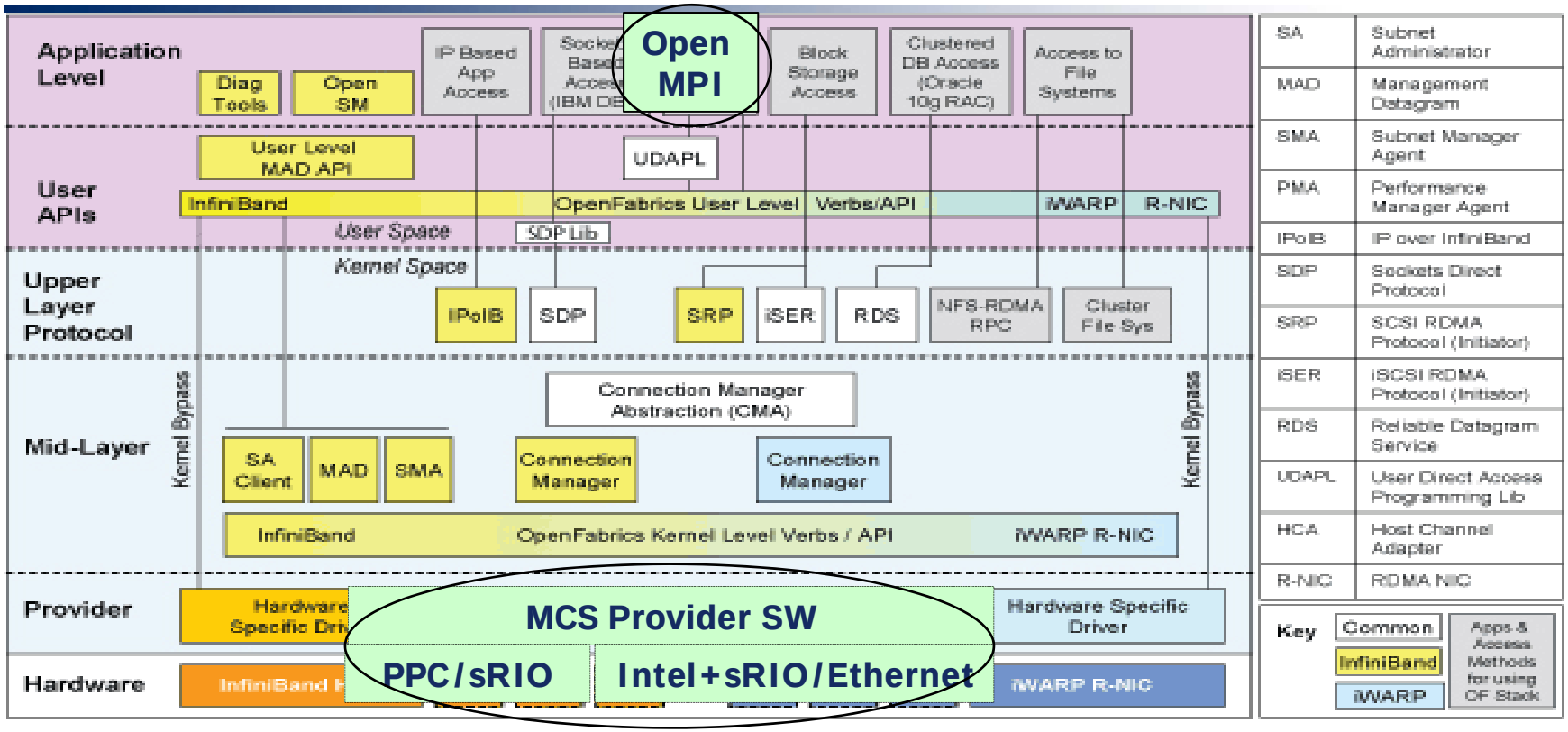
- HPC, System Integrators including “Tier 1”
- Microprocessor Providers
- Networking Providers (NIC/HCA, Switch)
- Server/Enterprise Software + Systems Providers
- NAS / SAN Providers
- Switched Fabric Industry Trade Associations
- Financial Services / Technology Providers
- National Laboratories
- Universities
- 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

ICS/DX	ICS/DX+MPI	ICS/DX+MPI tuned

•Mercury middleware
•No MPI content
•Preserve SW investment
	• OFED in kernel driver
	•Incremental MPI use
	•App. buffer integration
	`MPI_Send(smb_ptr, )`
	•MPI as MCP extension
	`MPI_Barrier(world)`
		• OFED OS bypass
		• Coordinate DMA HW Sharing

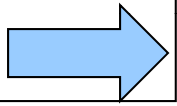
Open Solution SW Migration

Embedded Middleware Migration

Embedded SW Migration

ICS/DX	ICS/DX+MPI	ICS/DX+MPI tuned

•Mercury middleware
•No MPI content
•Preserve SW investment
	• OFED in kernel driver
	•Incremental MPI use
	•App. buffer integration
	`MPI_Send(smb_ptr, )`
	•MPI as MCP extension
	`MPI_Barrier(world)`
		• OFED OS bypass
		• Coordinate DMA HW Sharing



Open Solution SW Migration

Embedded Middleware Migration

Embedded SW Migration

ICS/DX

- Mercury middleware
- No MPI content
- Preserve SW investment

ICS/DX+MPI

- OFED in kernel driver
- Incremental MPI use
- App. buffer integration

```
MPI_Send(smb_ptr, )
```

- MPI as MCP extension

```
MPI_Barrier(world)
```

ICS/DX+MPI tuned

- OFED OS bypass
- Coordinate DMA HW Sharing

Open Solution SW Migration

Embedded Middleware Migration

Embedded SW Migration

Open Solution SW Migration

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

ICS/DX

- Mercury middleware
- No MPI content
- Preserve SW investment

ICS/DX+MPI

- OFED in kernel driver
- Incremental MPI use
- App. buffer integration

```
MPI_Send(smb_ptr, )
```

- MPI as MCP extension

```
MPI_Barrier(world)
```

ICS/DX+MPI tuned

- OFED OS bypass
- Coordinate DMA HW Sharing

MPI+DATAFLOW CONCEPT

Single library, integrated resources

DMA HW assist for MPI, RDMA

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)

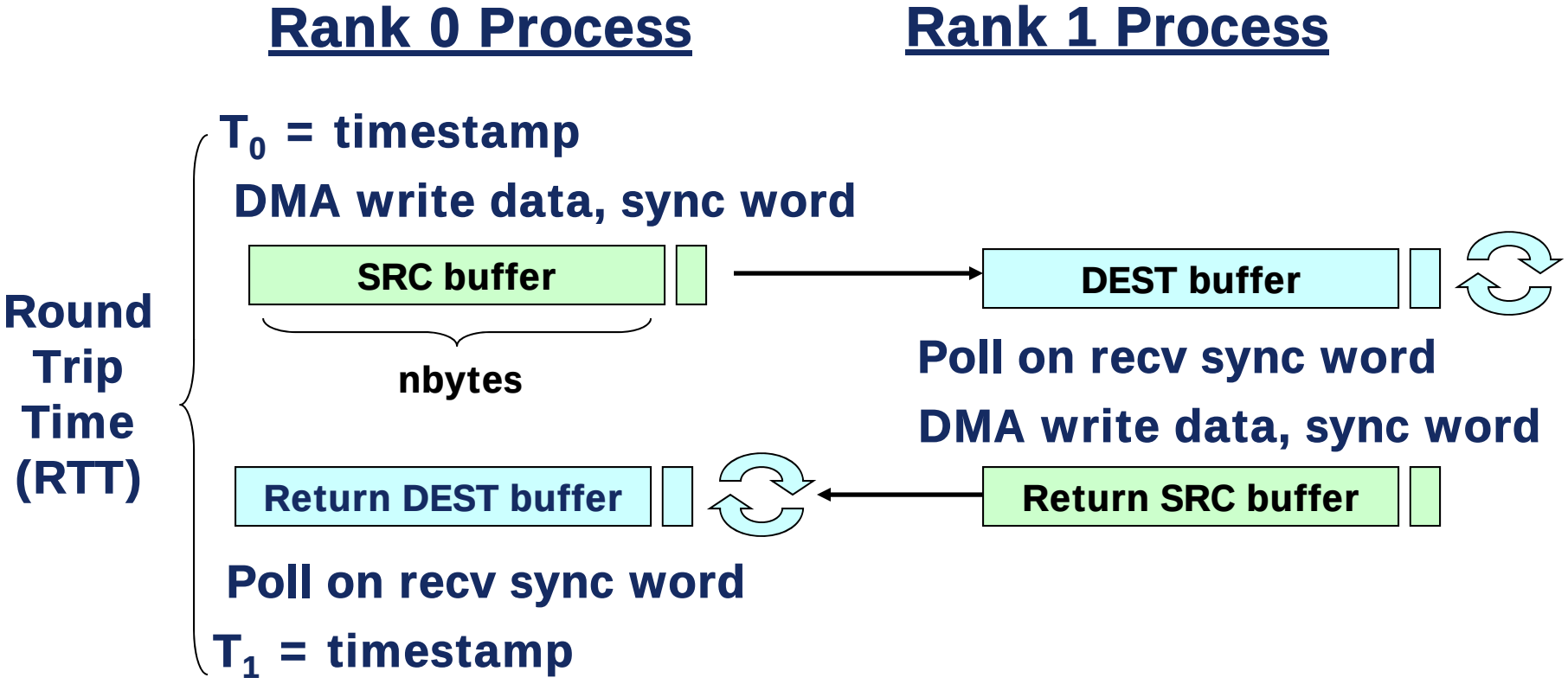
- malloc -> MPI_Alloc_mem
- App still uses open APIs
- MCS “PMEM”, contig. mem.
- Fabric / OFED+PMEM

MERCURY COMPUTER SYSTEMS™

Motivating Example: ICS/DX vs. MPI

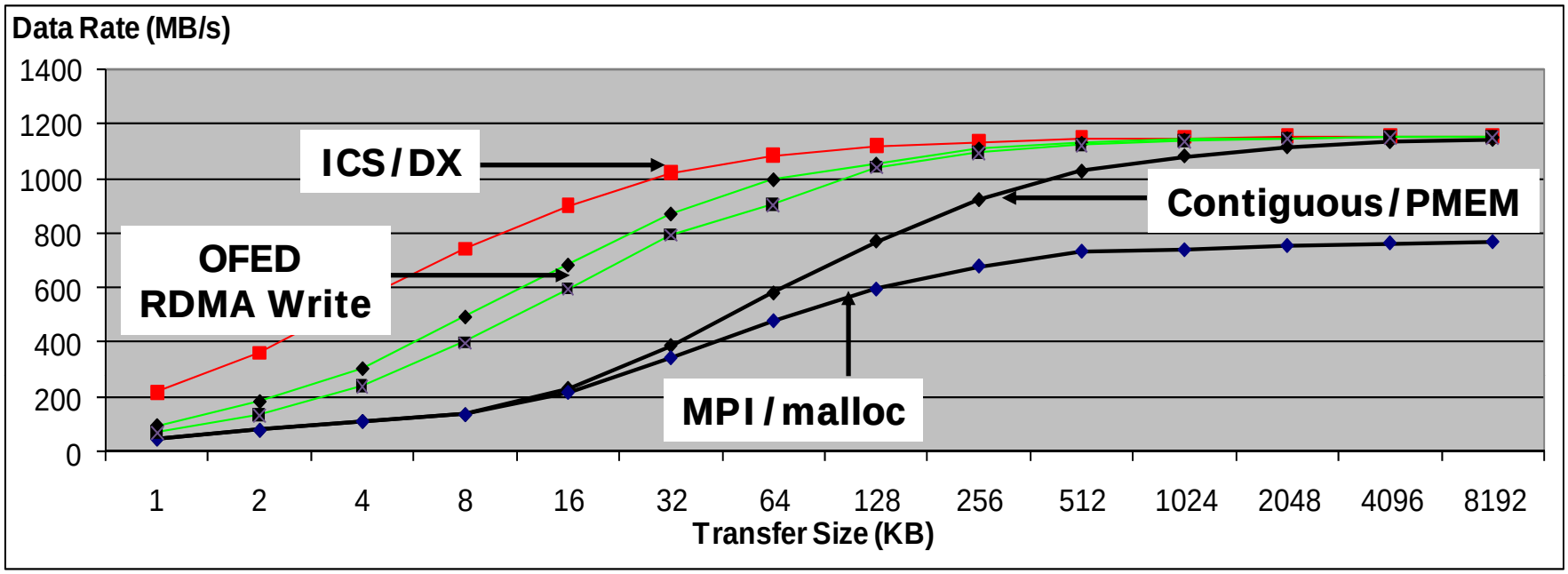
Data Rate Performance

Benchmark ½ Round Trip Time

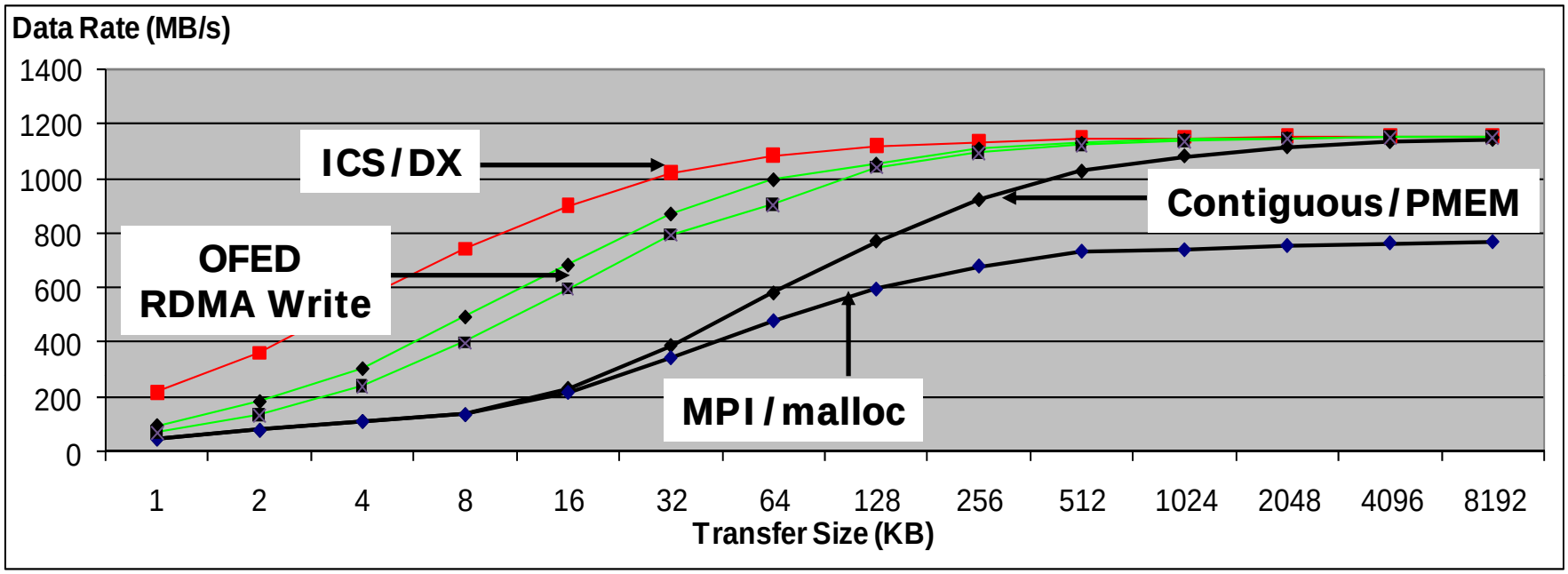


- Inter-node: ranks placed on distinct nodes
- Latency = ½ RTT (record min, max, avg, median)
- Data rate calculated: transfer size, avg latency

ICS/DX, MPI/OFED: 8641D sRIO



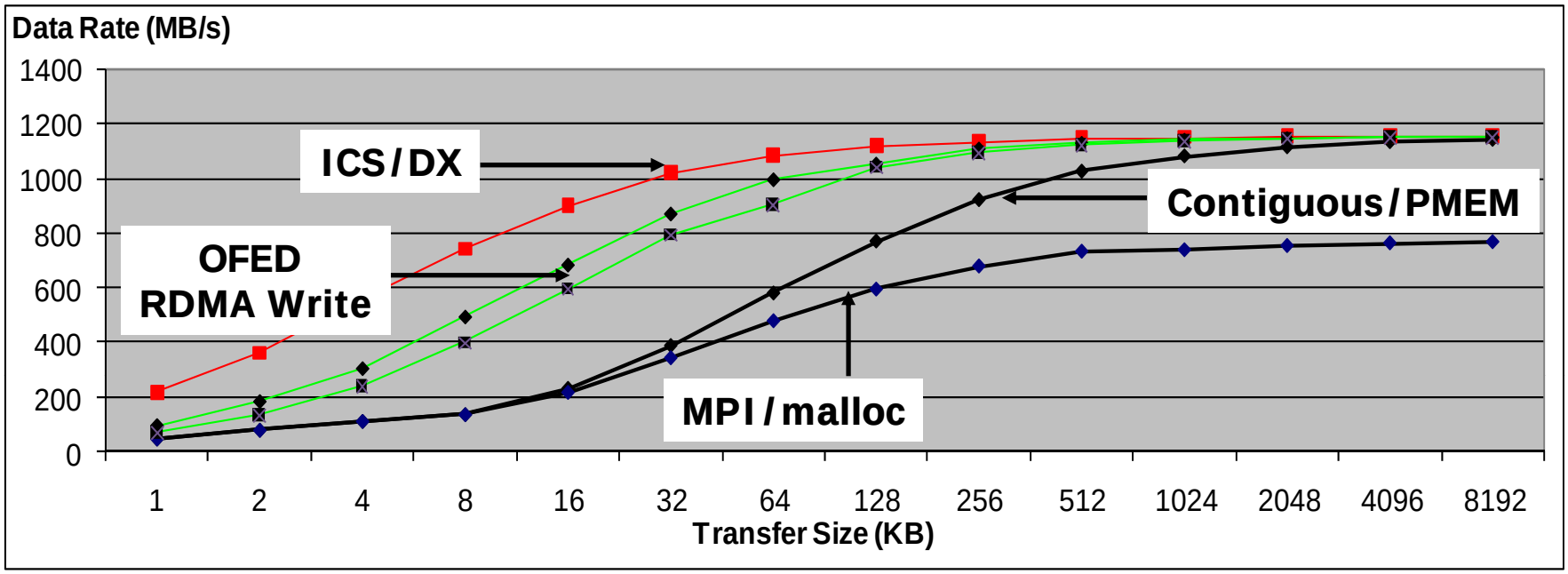
ICS/DX, MPI/OFED: 8641D sRIO



½ 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

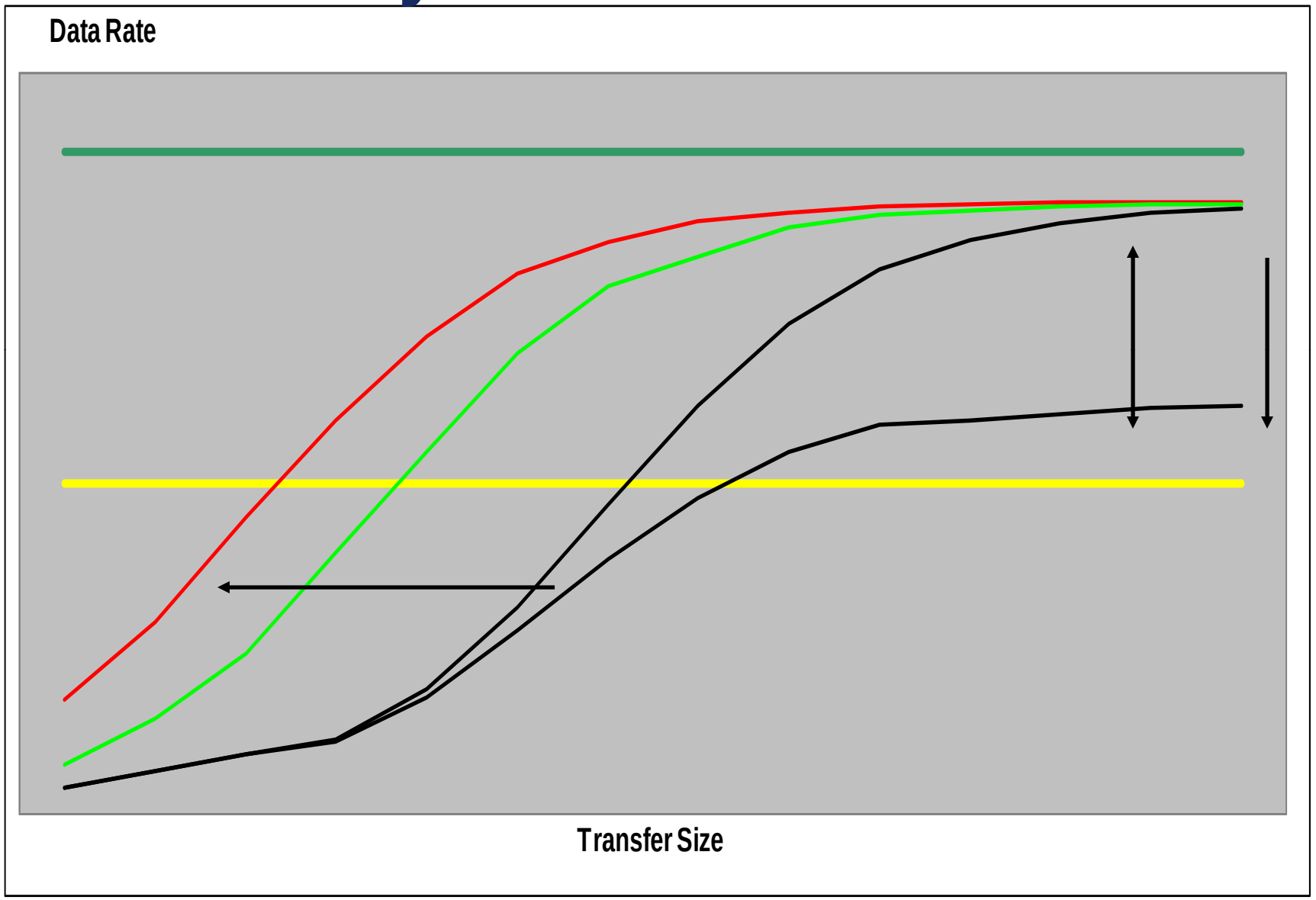


½ RTT Latency – Small Transfer (4 bytes)

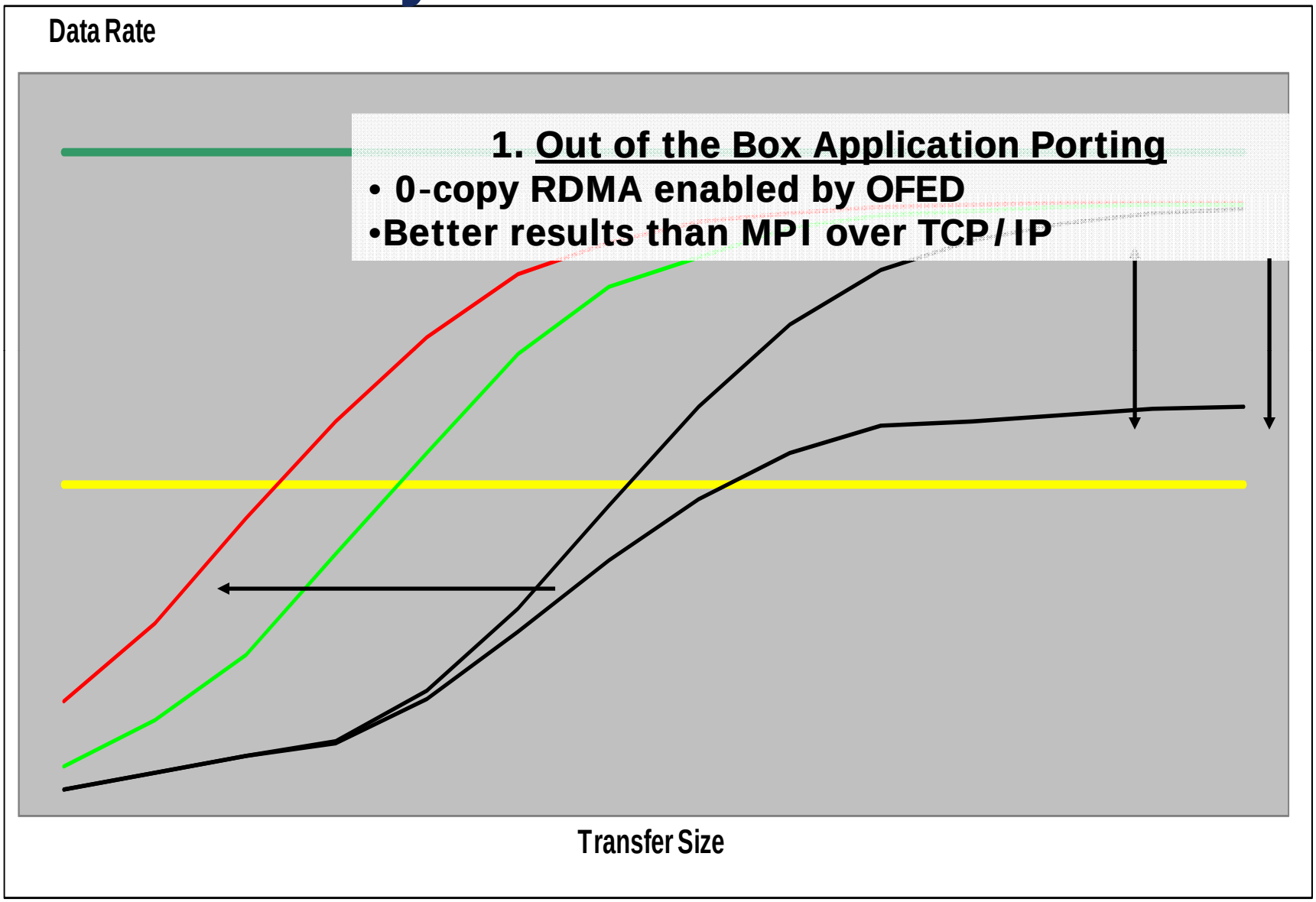
OFED (data+flag) 13 usec	OFED (data/poll last) 8.6 usec	ICS/DX (data+flag): 4 usec

- **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

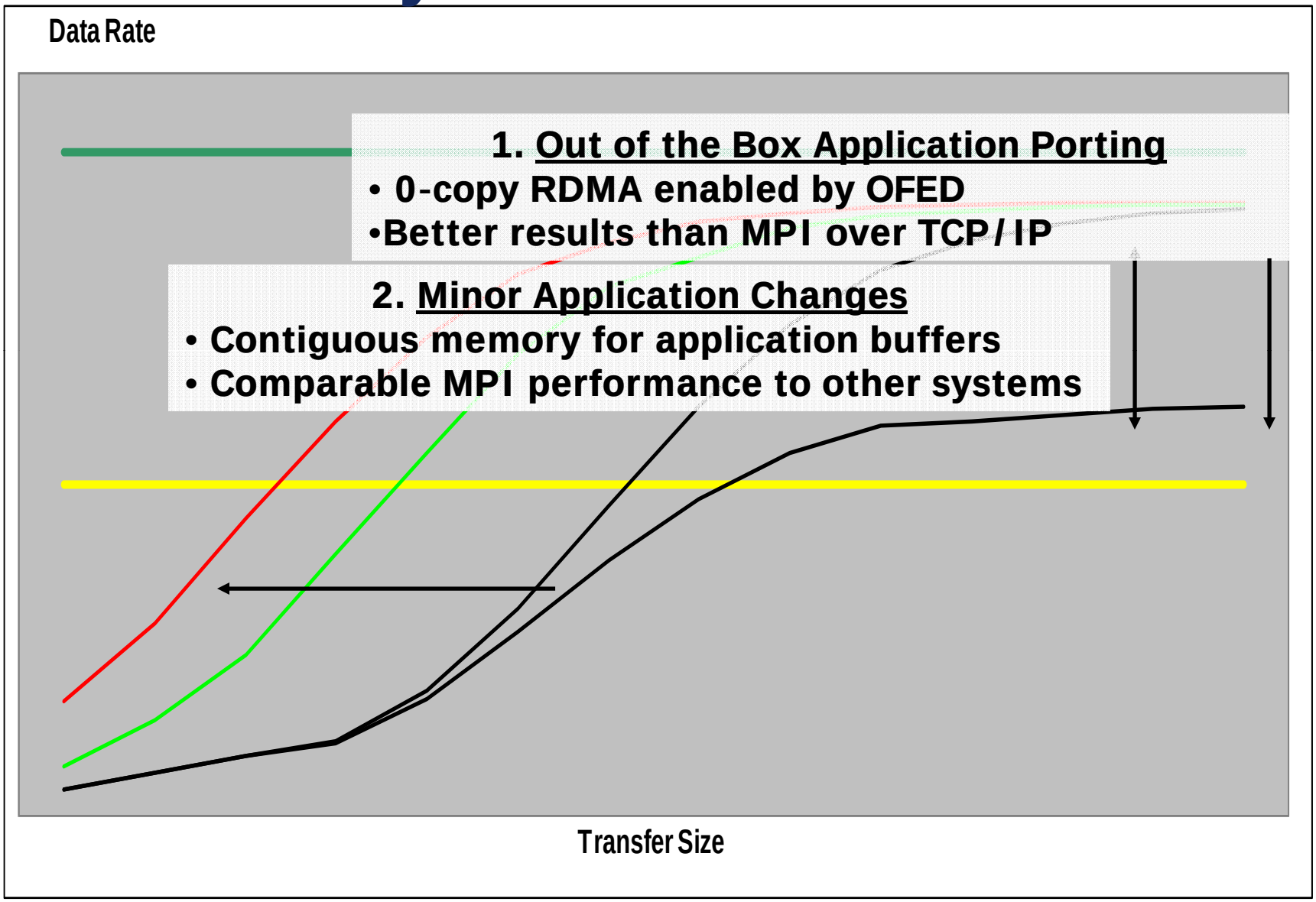
Preliminary Conclusions



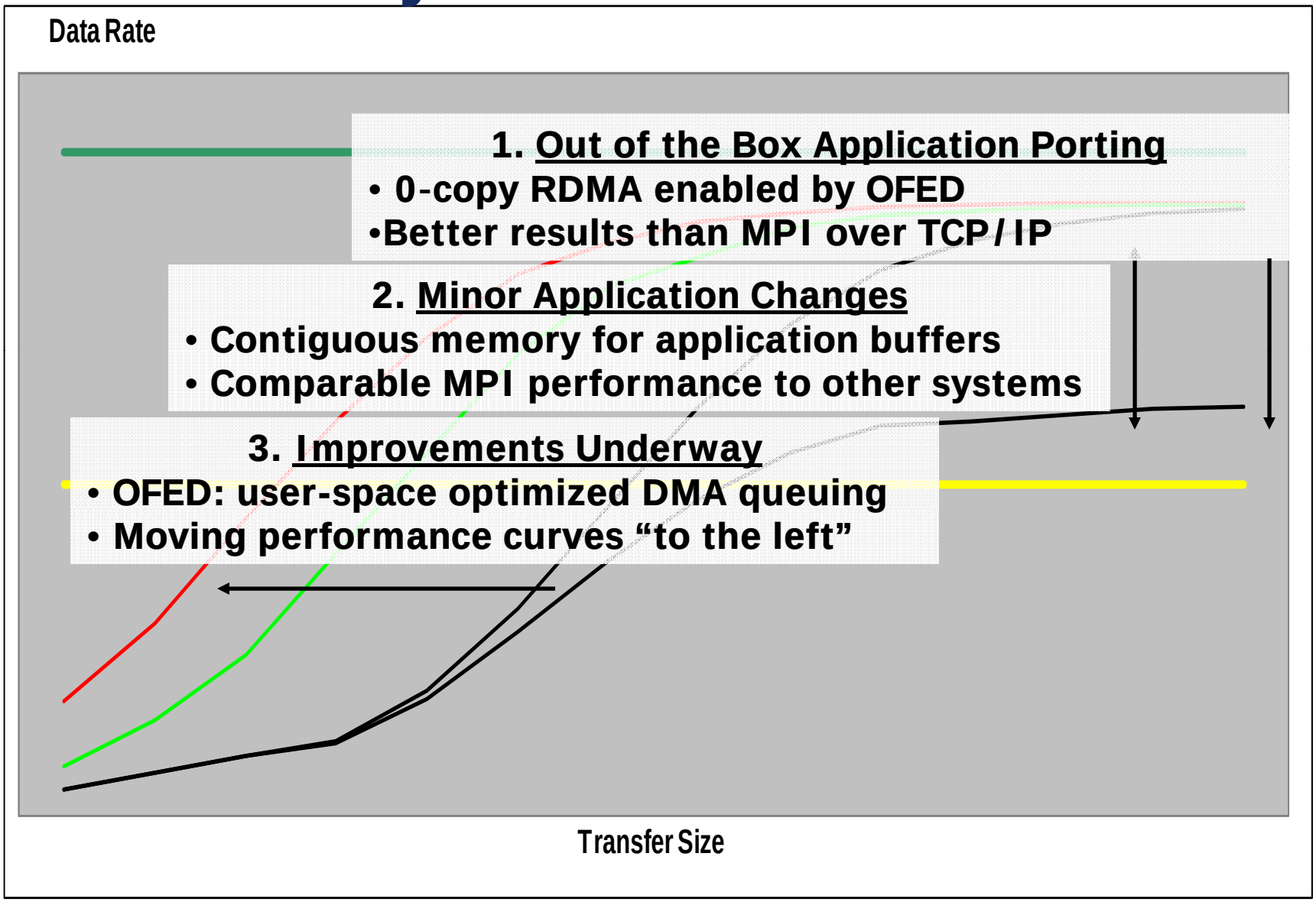
Preliminary Conclusions



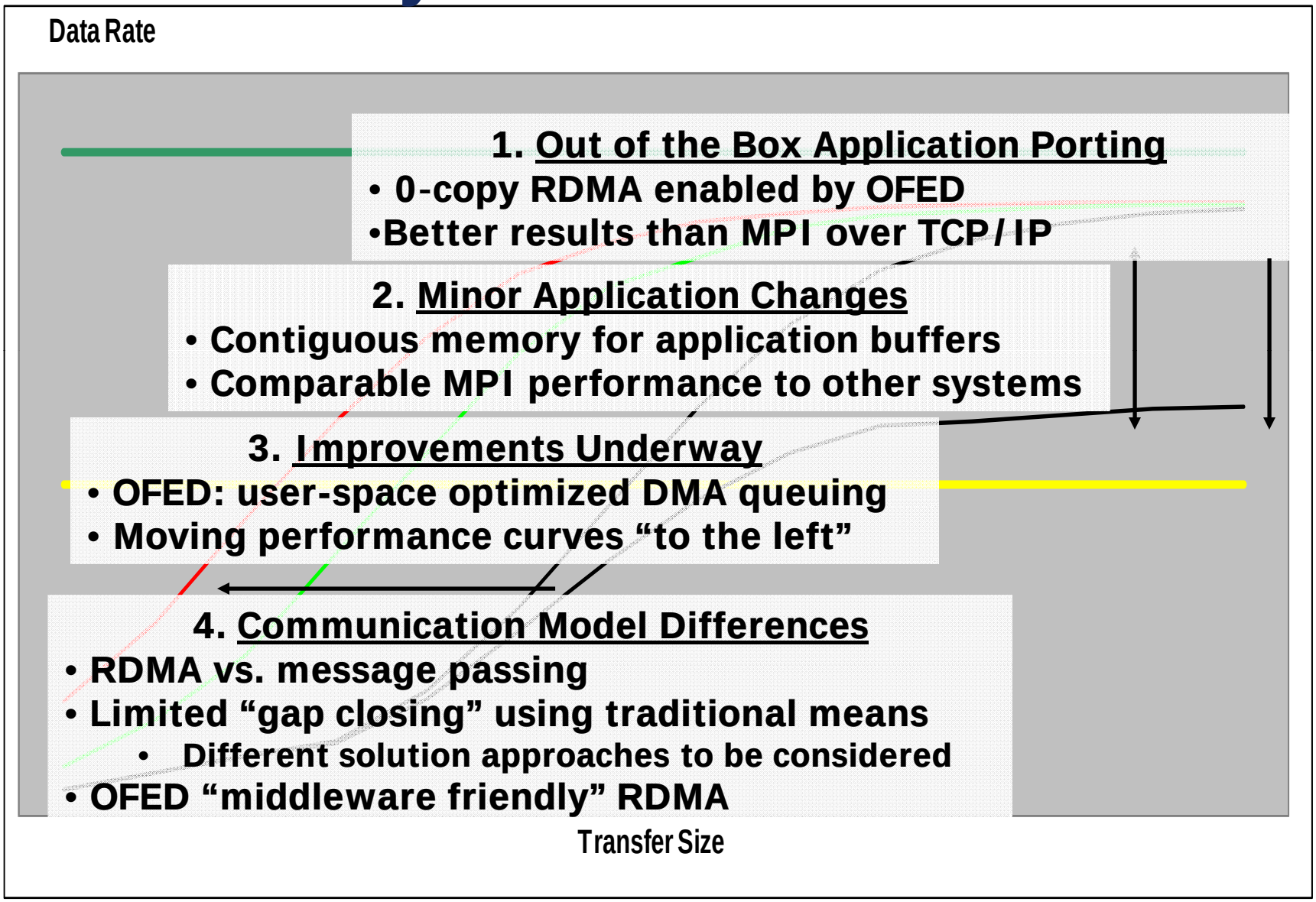
Preliminary Conclusions



Preliminary Conclusions



Preliminary Conclusions

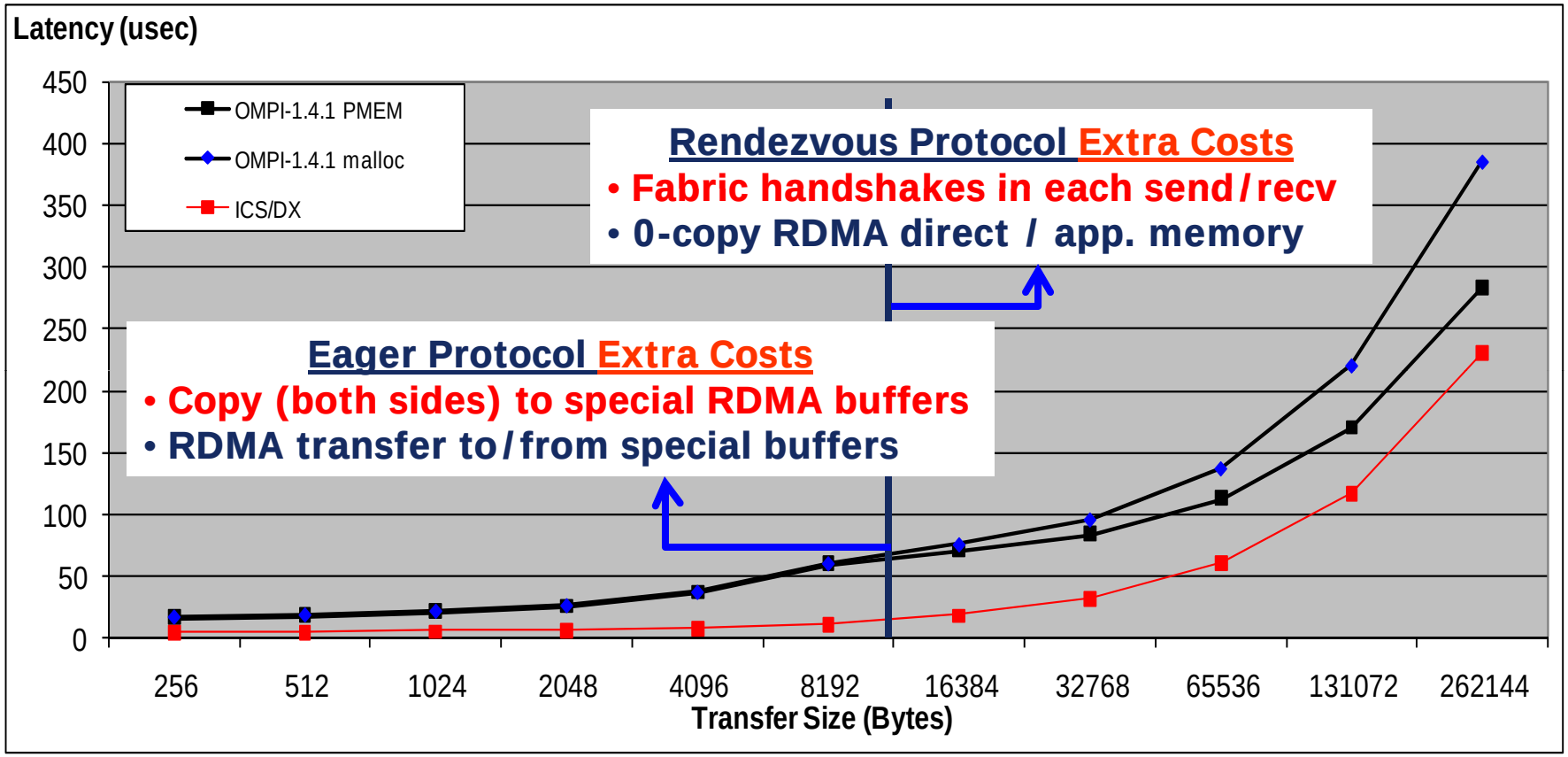


MERCURY COMPUTER SYSTEMS™

Communication Model Comparison

Message Passing and RDMA

Message Passing vs. RDMA



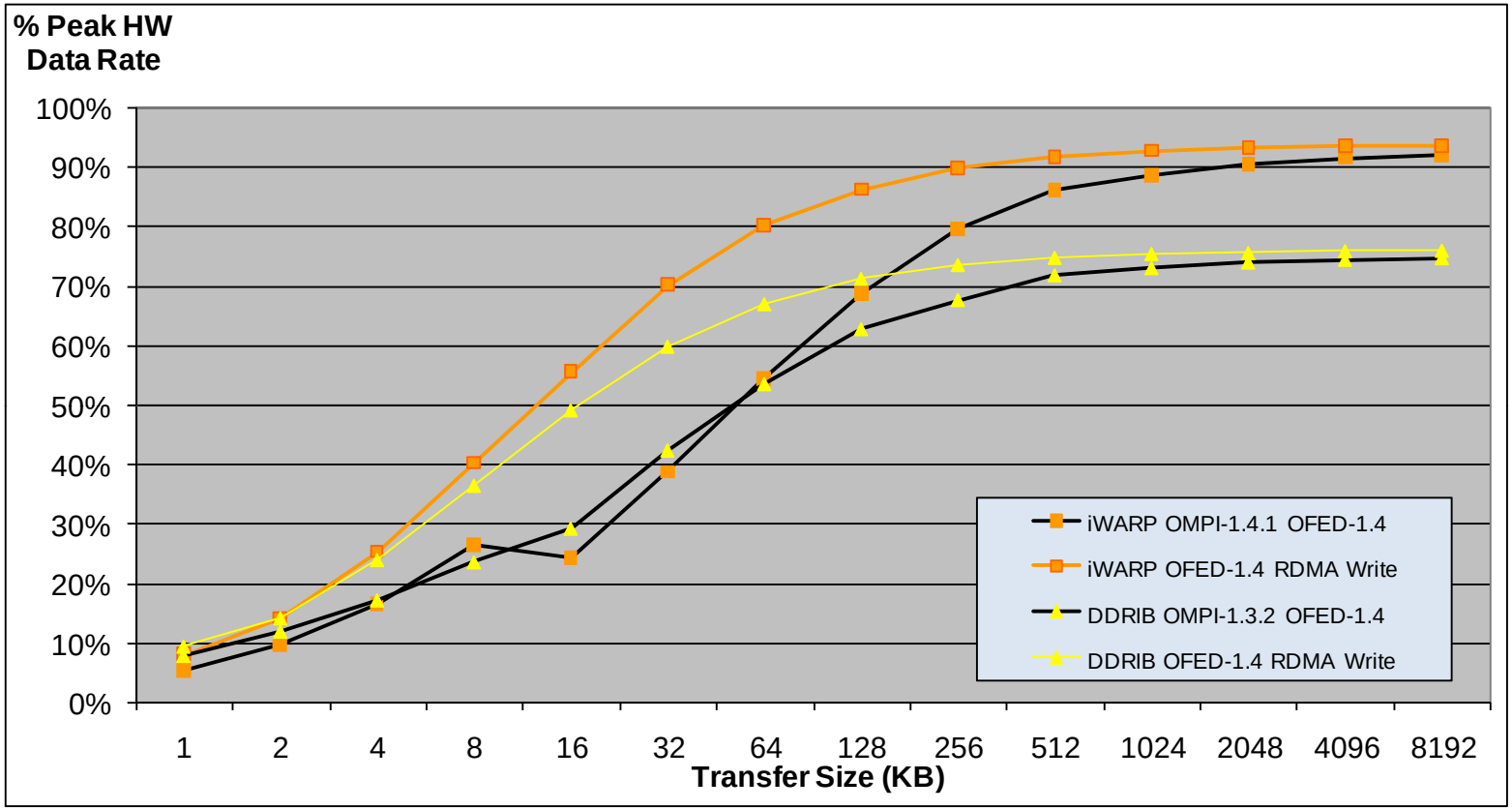
- Flexibility (MPI) vs. Performance (RDMA)
- Issue: memory registration / setup for RDMA
- MPI Tradeoffs: Copy vs. Fabric Handshake Costs

MERCURY COMPUTER SYSTEMS™

MPI and RDMA Performance

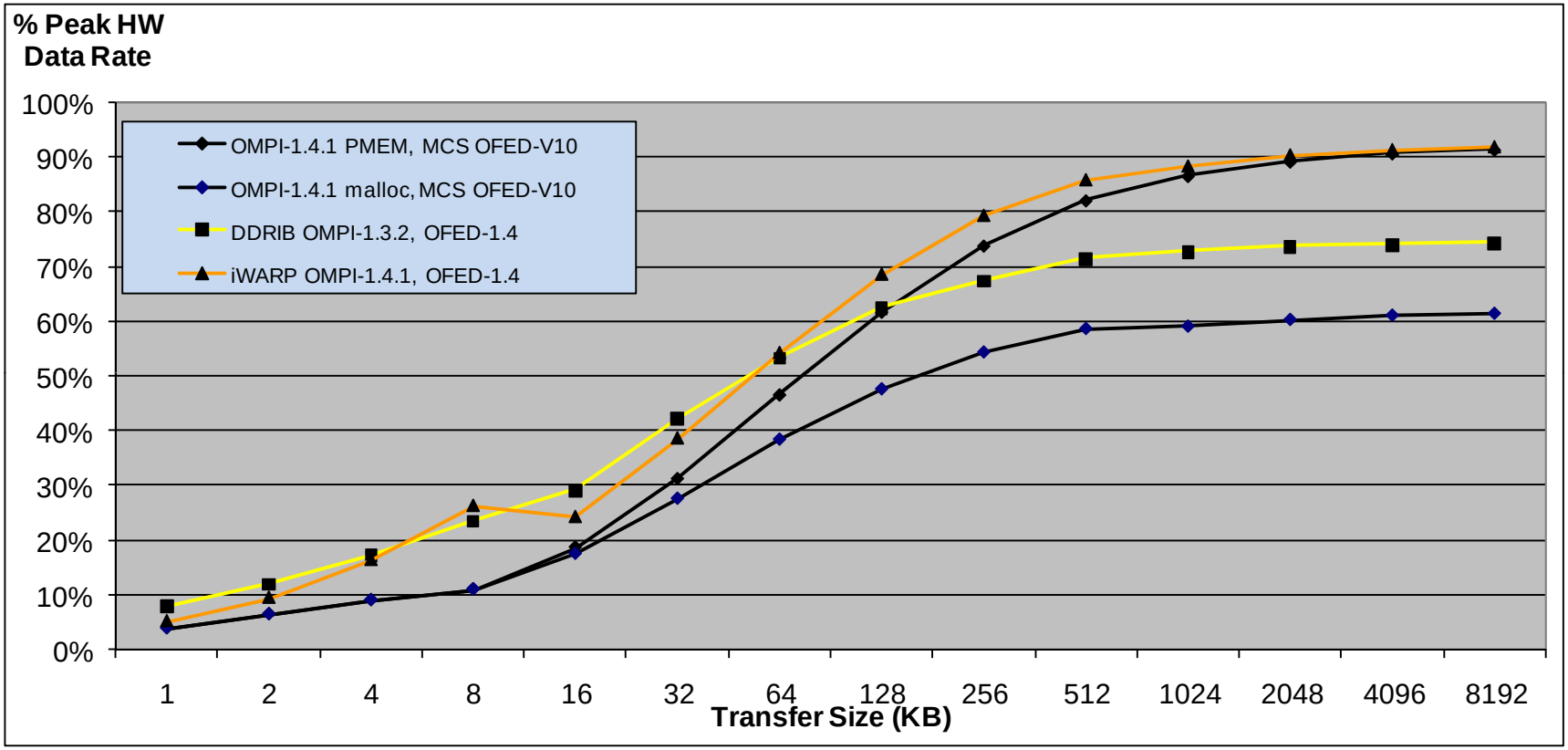
Server/HPC Platform Comparisons

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

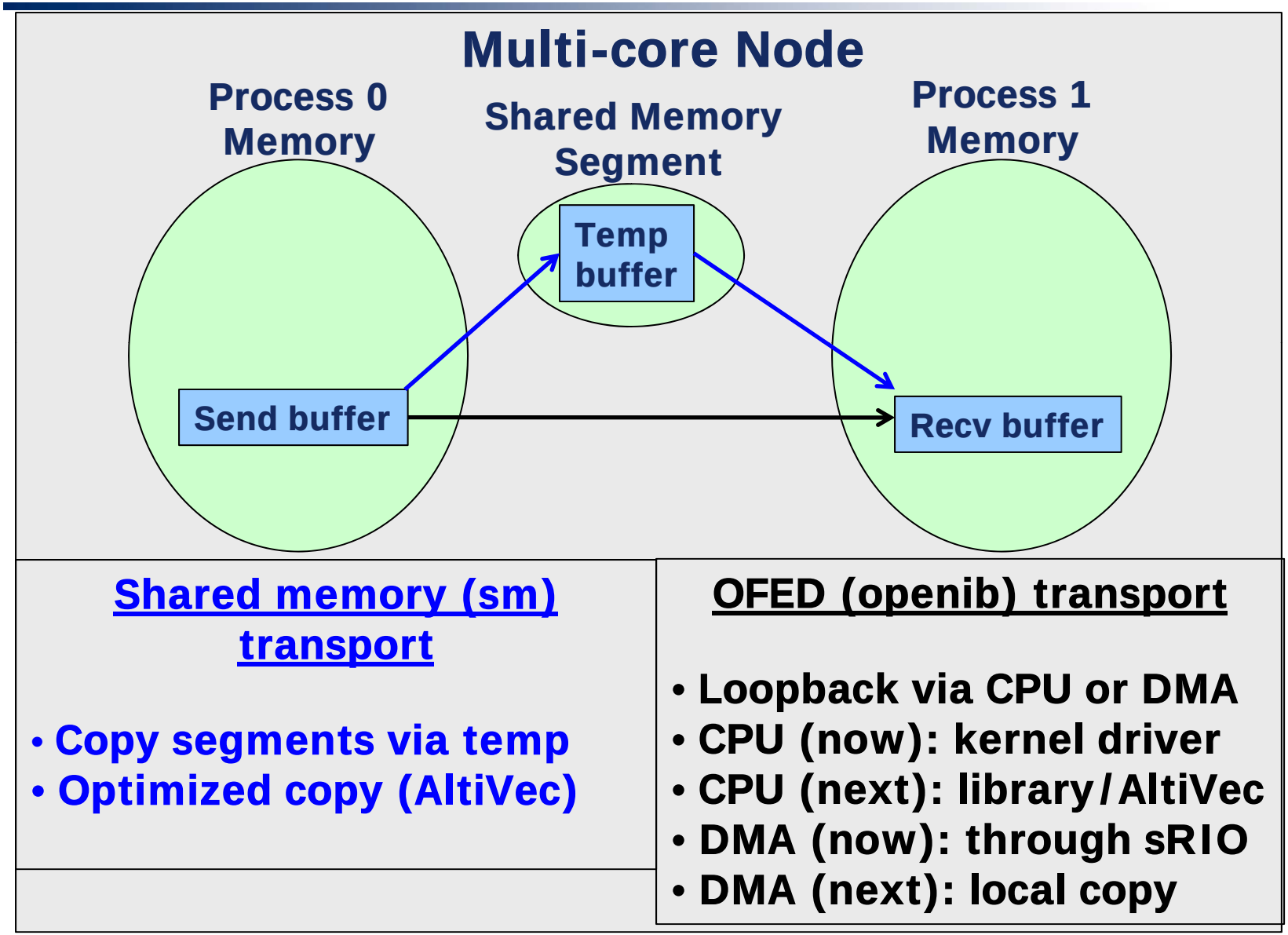


- **MPI Data Rate Comparable Across Fabrics**
- **Small Transfer Copy Cost Differences**
  - **X86_64 (DDRIB, iWARP) faster than PPC32 (sRIO)**

MERCURY COMPUTER SYSTEMS™

Mercury MPI/OFED Intra-Node Performance

Open MPI Intra-Node Communication



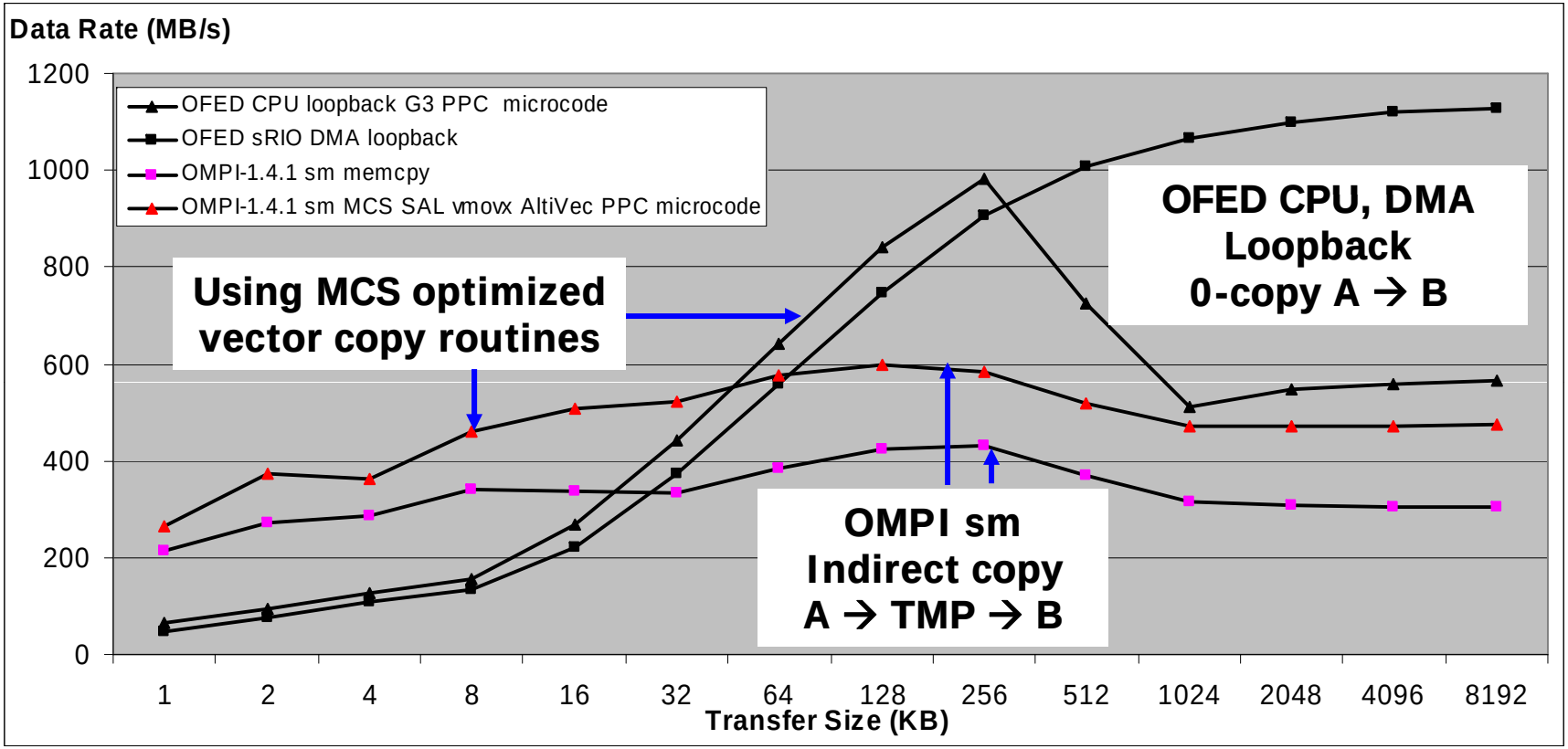
Shared memory (sm) transport

- Copy segments via temp
- Optimized copy (AltiVec)

OFED (openib) transport

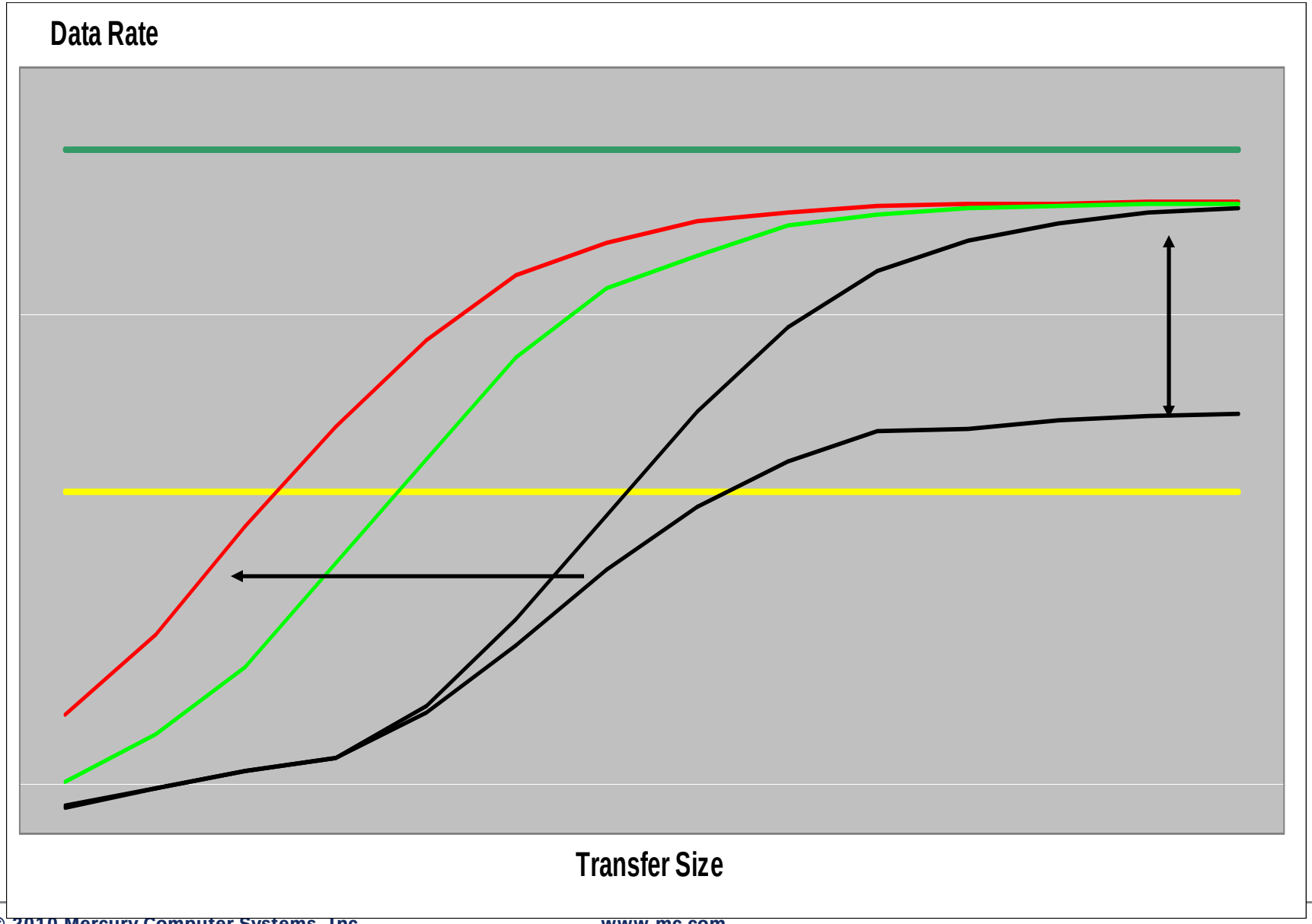
- Loopback via CPU or DMA
- CPU (now): kernel driver
- CPU (next): library/AltiVec
- DMA (now): through sRIO
- DMA (next): local copy

MPI Intra-Node Data Rate

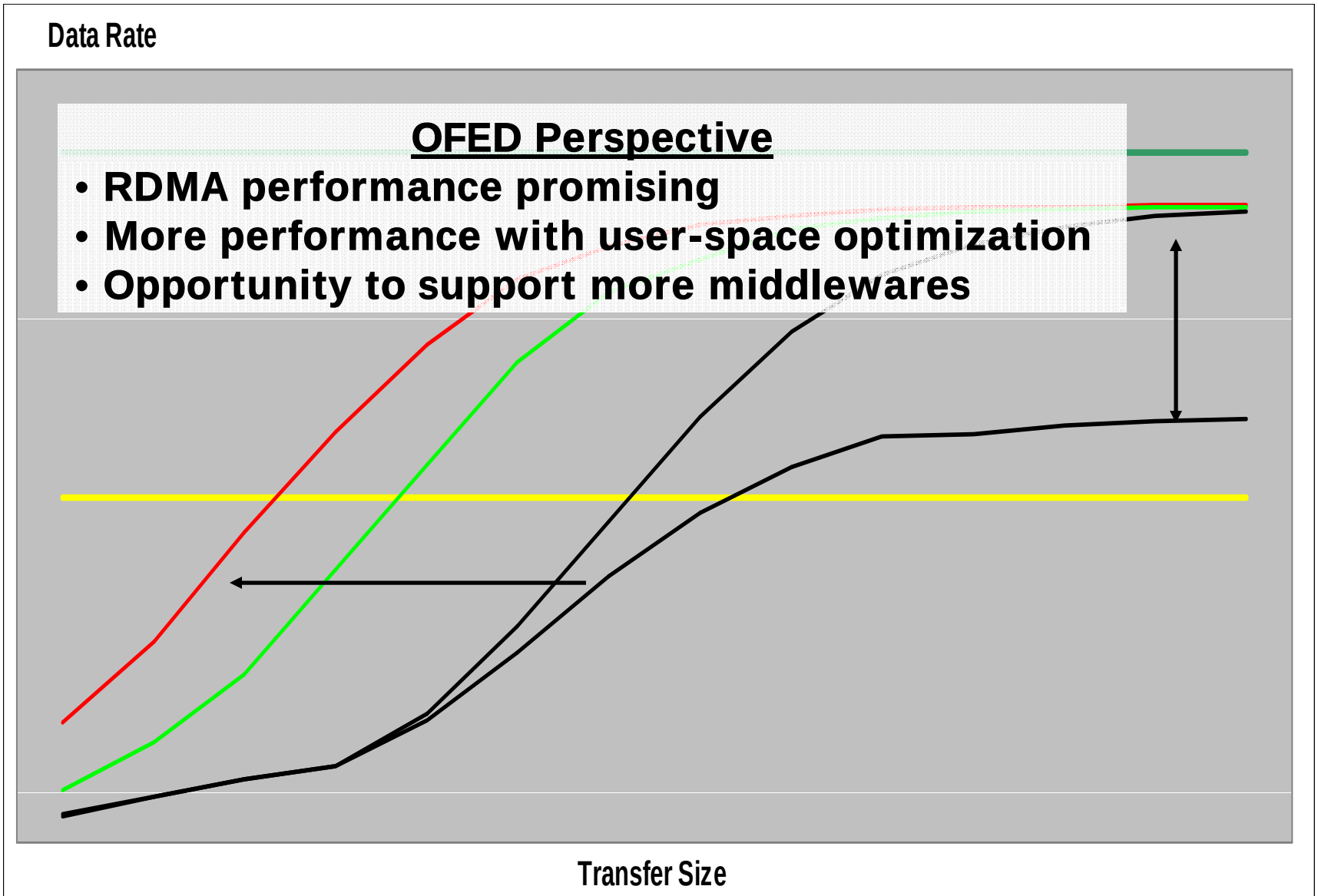


- **OMPI sm latency better (very small messages)**
  - 2-3 usec latency (OMPI sm) vs. 11 usec (OFED CPU loopback)
- **OFED loopback better for large messages**
  - Direct copy – 50% less memory bandwidth used

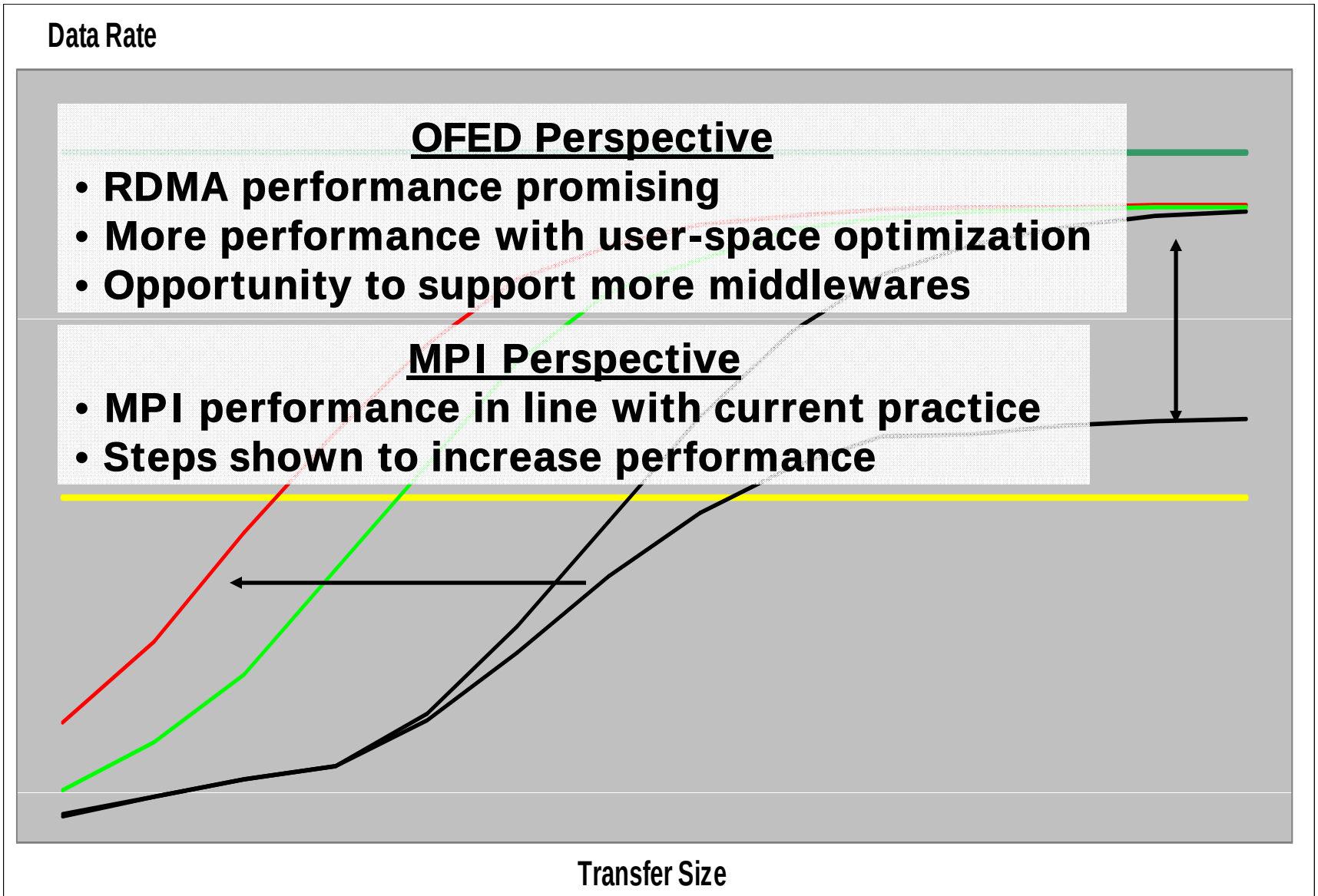
Summary / Conclusions



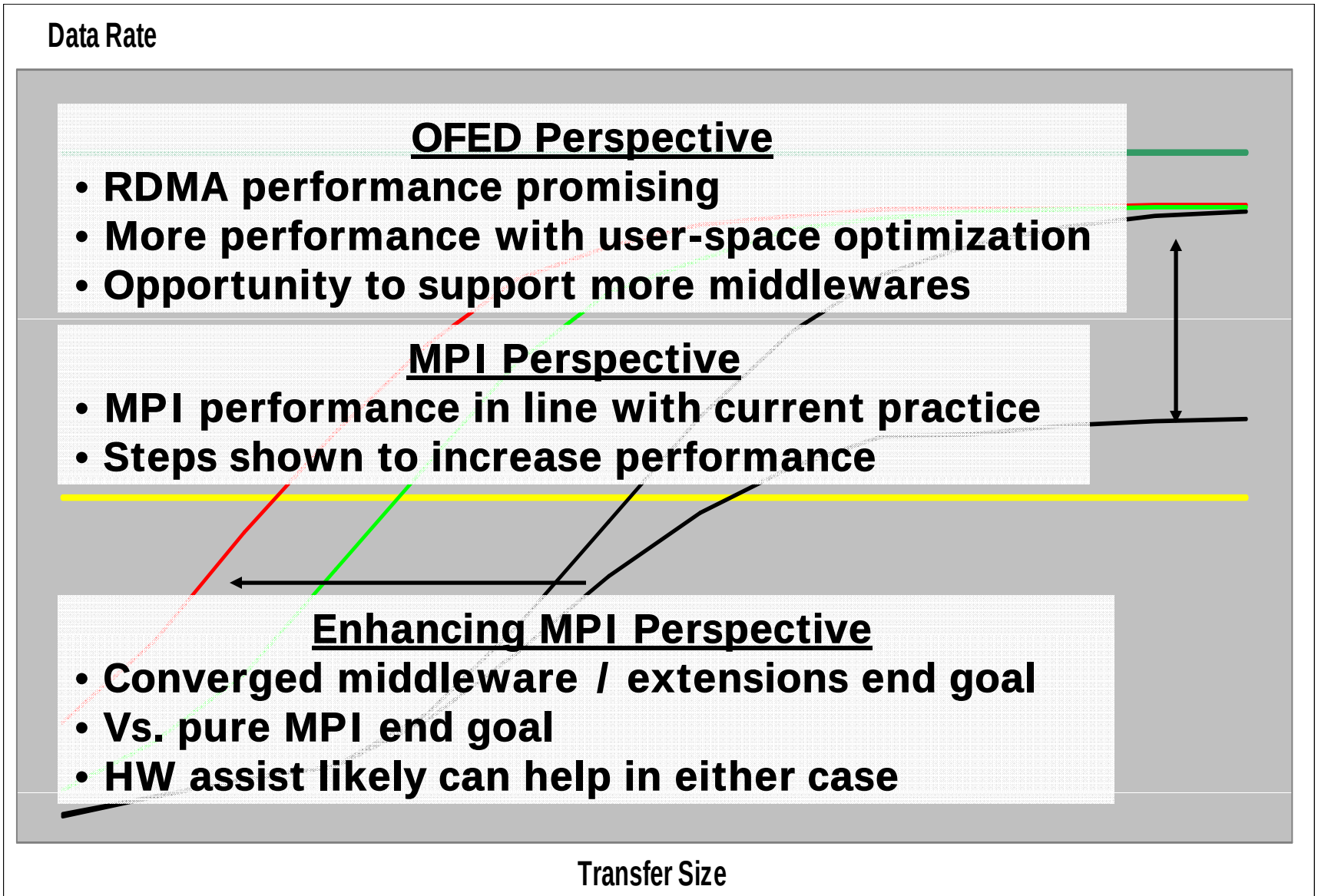
Summary / Conclusions



Summary / Conclusions



Summary / Conclusions



MERCURY COMPUTER SYSTEMS™

Thank You

Questions?

Kenneth Cain
kcain@mc.com