Shared Memory Mirroring for Reducing Communication Overhead on Commodity Networks

Bruce Palmer, Jarek Nieplocha, and Edo Apra
Pacific Northwest National Laboratory
Richland, WA 99352
USA

Introduction

- Clusters built from commodity processors are frequently connected via low performing networks (e.g. Ethernet)
- Calculations with high communication costs are inefficient on these clusters
- Most nodes on current clusters contain two or more processors and have large amounts of memory (more than a GByte)

Methods for Latency Hiding

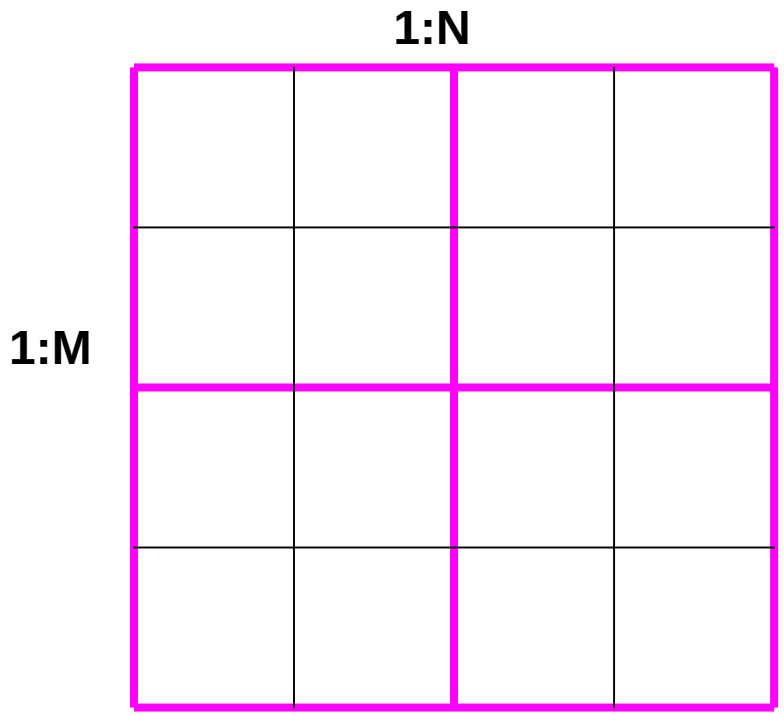
- Aggregating small messages
  - Not always feasible, depending on algorithm
- Data replication
  - Large memory requirement
- Nonblocking communication
  - Not always supported by either the communication layer (MPI) or the network

Goal

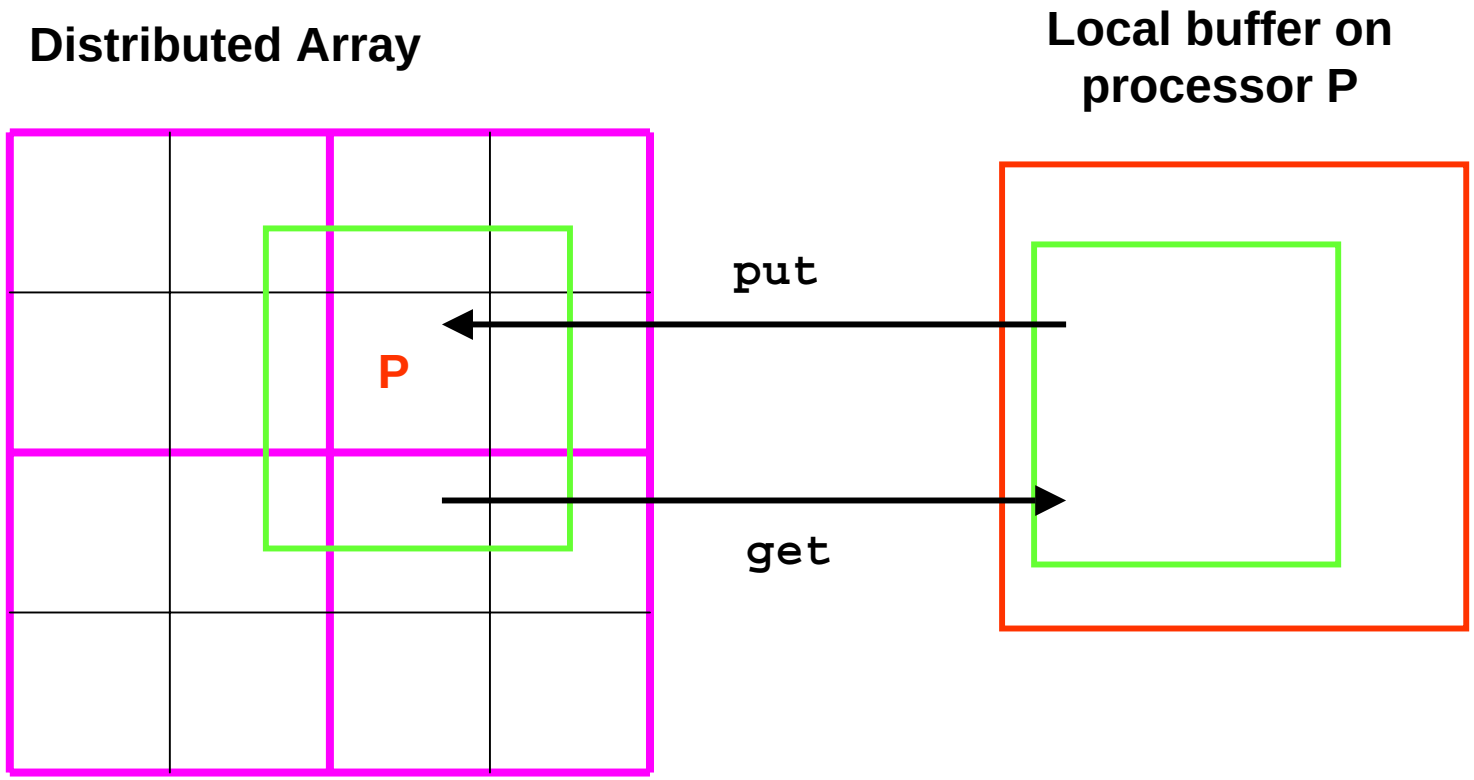
Use memory to hide latency and low bandwidth costs associated with communication between nodes

Distributed Computing

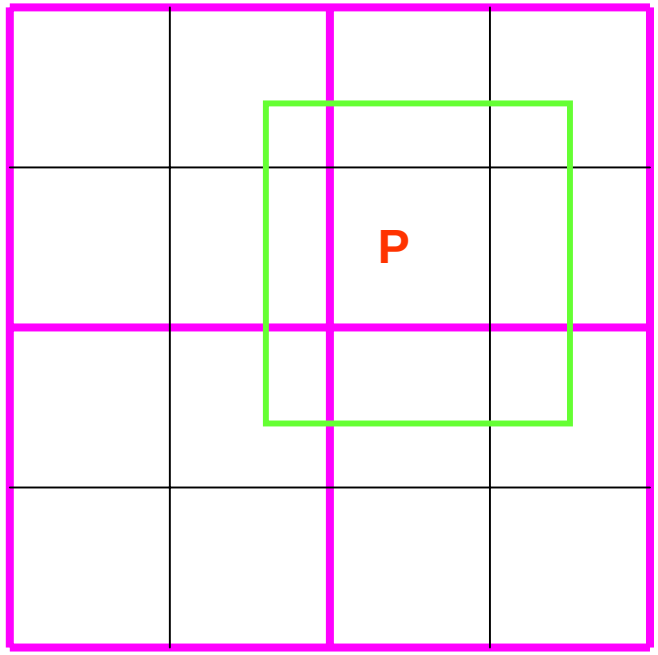
Data is distributed across ***processors***

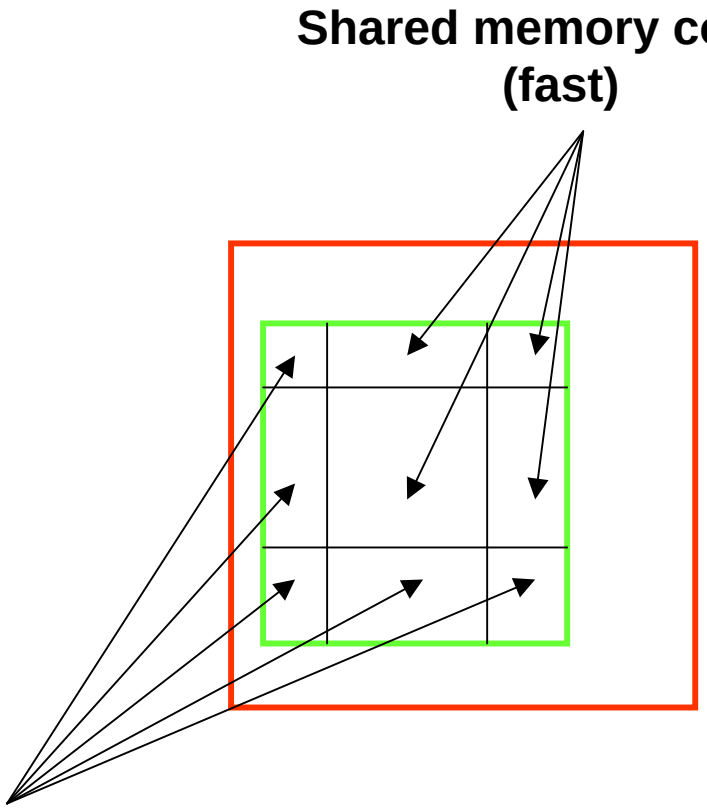


Accessing Distributed Data



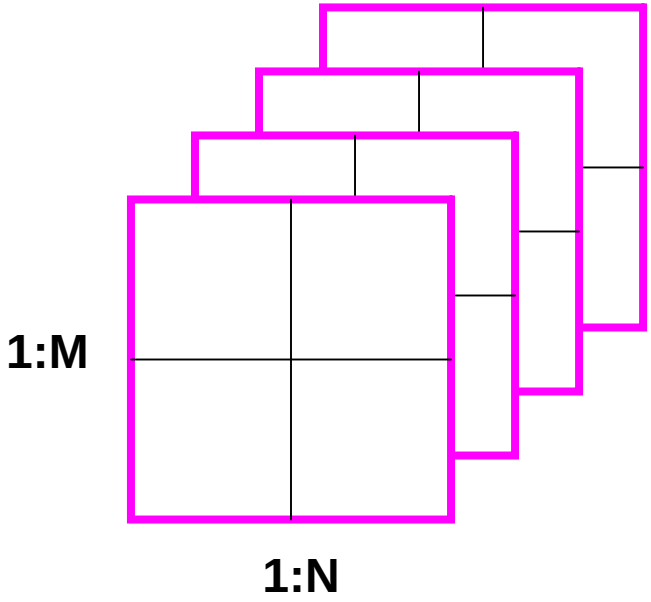
Decomposing Access into Point-to-Point Communication



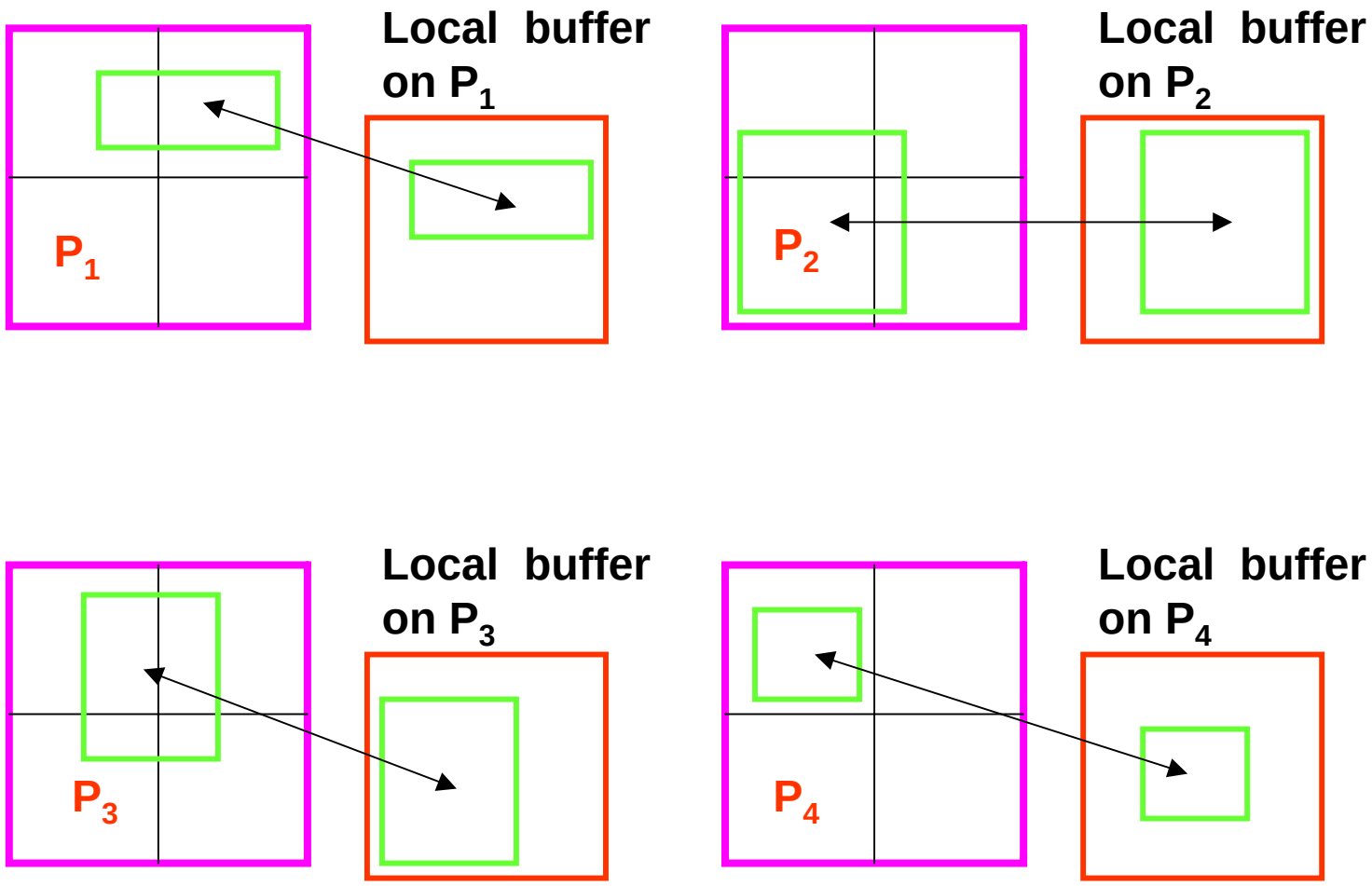


Mirrored Arrays

Data is replicated ***across*** nodes but distributed ***within*** nodes



Access on Mirrored Arrays



Comments

- All data access operations between local buffers and mirrored arrays now consist of shared memory copies, which are fast
- Data on each copy of mirrored array can be accessed and manipulated independently
- Some kind of merge operation is need to synchronize data

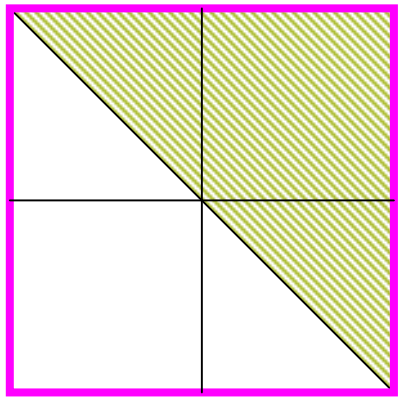
Merging Mirrored Arrays

- Merge is defined as the addition of copies of array across nodes
- Data for each mirrored array must be laid out in the same way on each node
- Some gaps may exist between data associated with each processor
- Use a single global sum across nodes to implement merge

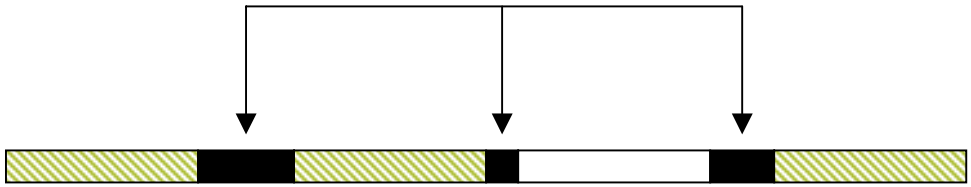
Mirrored Arrays in GA Toolkit

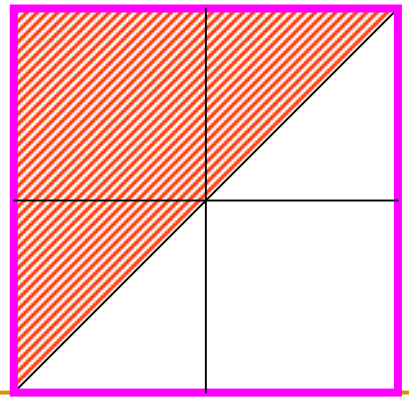
- Global Arrays are distributed arrays with shared memory-style access
  - Global address space
  - One-sided put, get, accumulate
  - Vector-matrix operations
- Mirrored arrays support most of the same operations as regular distributed Global Arrays
  - Put, get, accumulate operate between local buffers and mirrored array on the same node
- Additional operations are copy between distributed and mirrored arrays and a merge operation

Implementing Merge

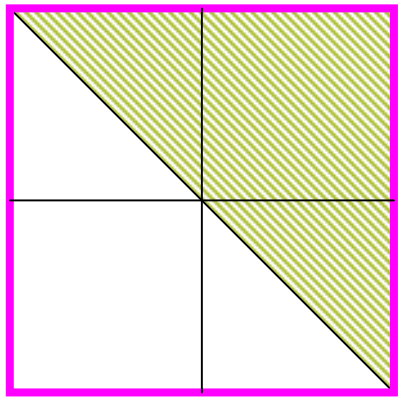


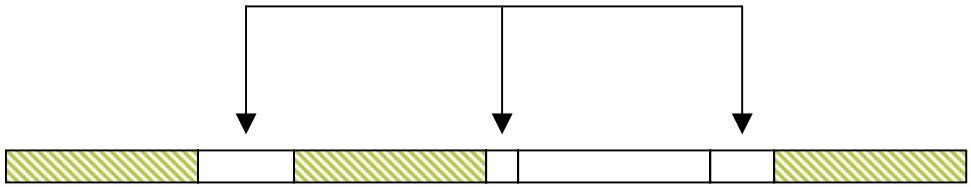
Gaps in shared memory with uninitialized data

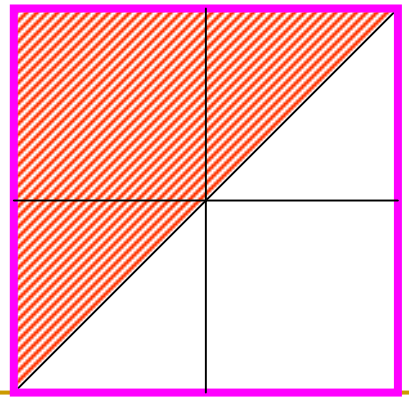


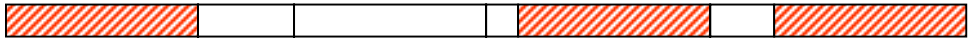


Zero Data in Gaps

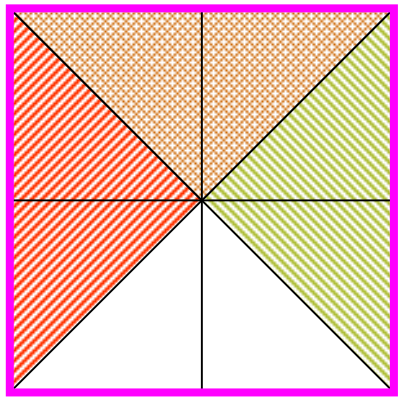






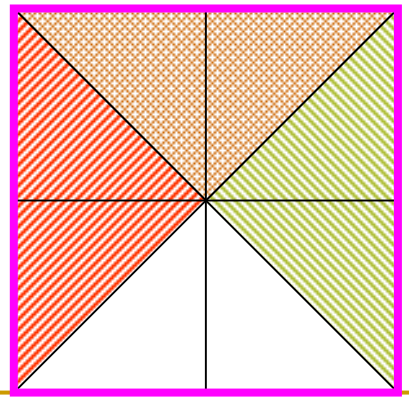


Perform Global Sum Across Nodes



Time to execute global sum is proportional to D*lnN

N is the number of nodes
D is the amount of data in array



Performance on Itanium-2 Cluster

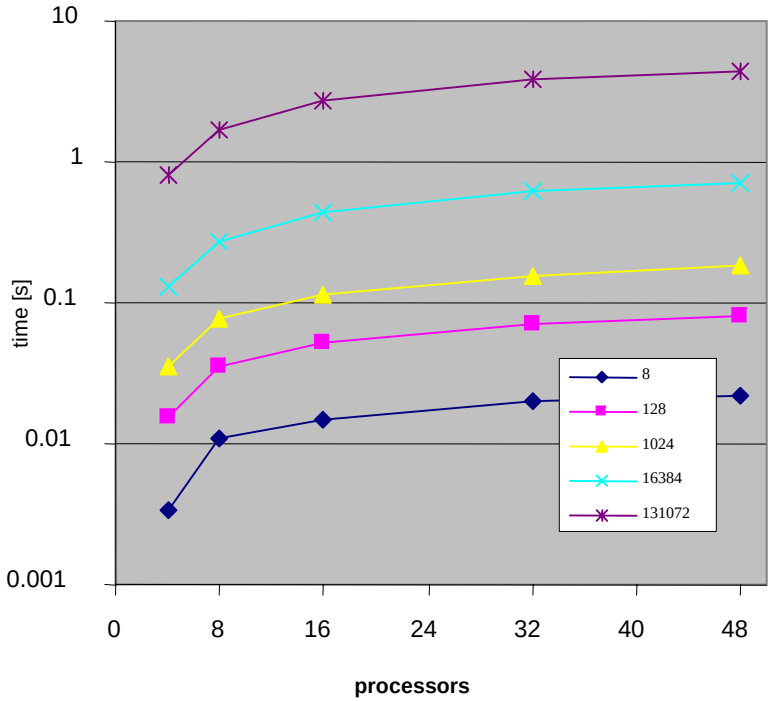
- HP RX2600 SMP nodes
- 2 1-GigaHertz Itanium-2 processors
- 3 networks
  - 100Mbit/s Ethernet
  - Myrinet 2000
  - Quadrics Elan-3

Network Characteristics

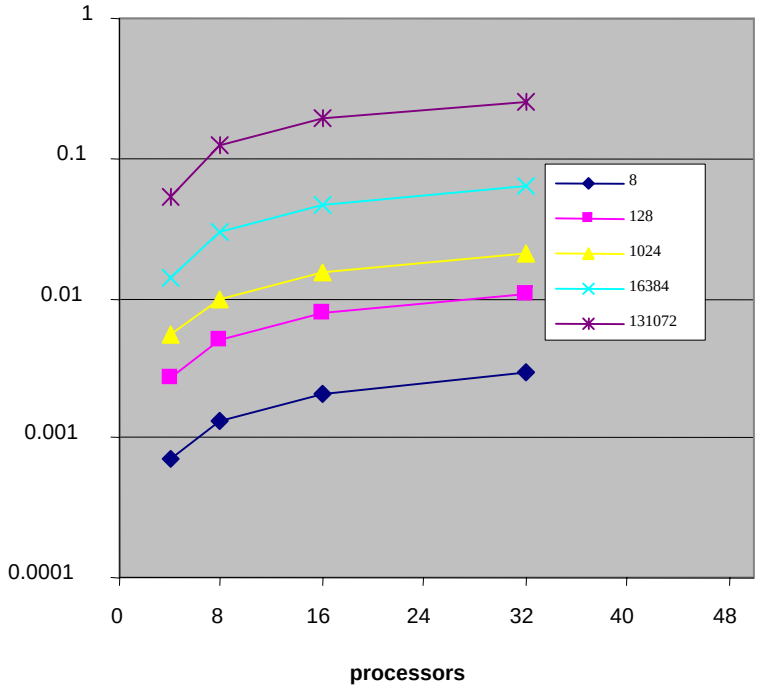
		Ethernet	Myrinet	Quadrics
Latency [µs]	distributed	144	30.8	12
	mirrored	3.58		
Bandwidth [MB/s]	distributed	11.7	219	225
	mirrored	1560		

Merge Performance

Ethernet

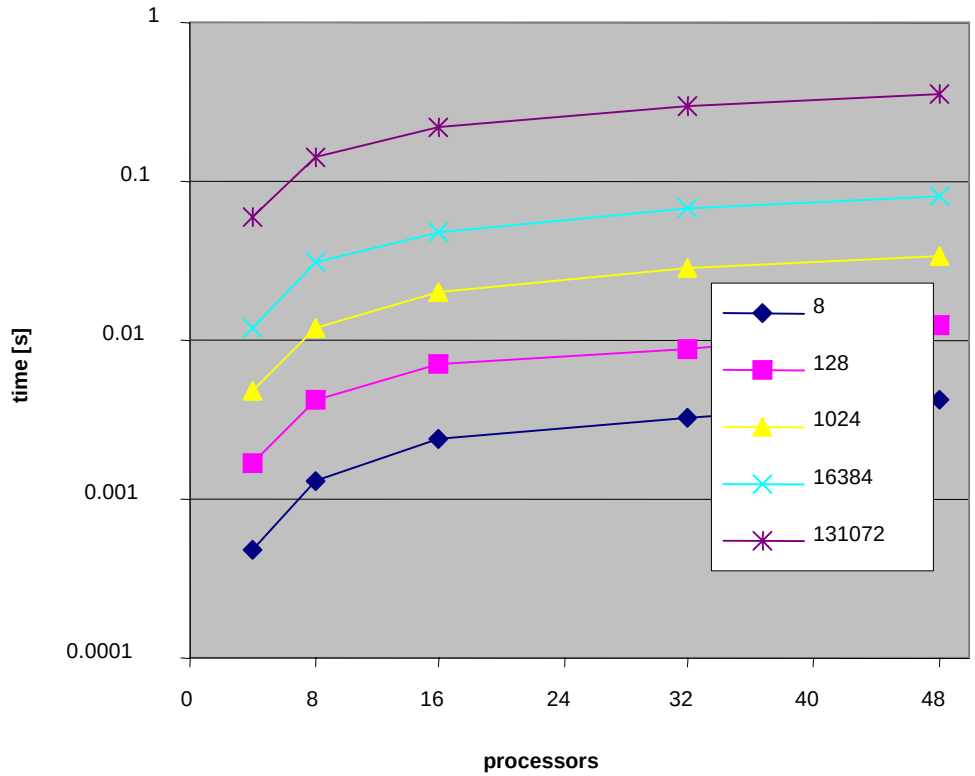


Myrinet

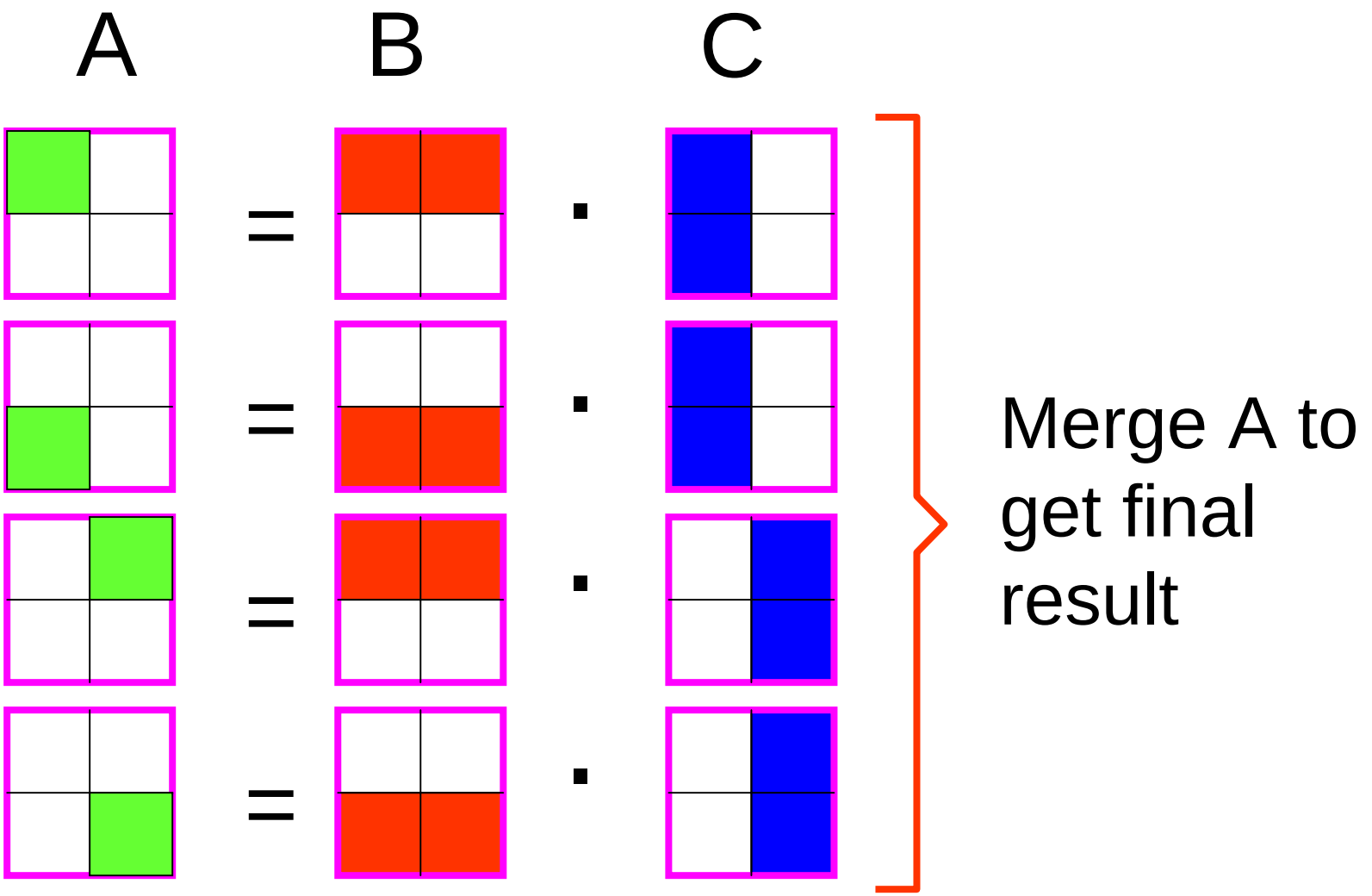


Merge Performance (cont.)

Quadrics

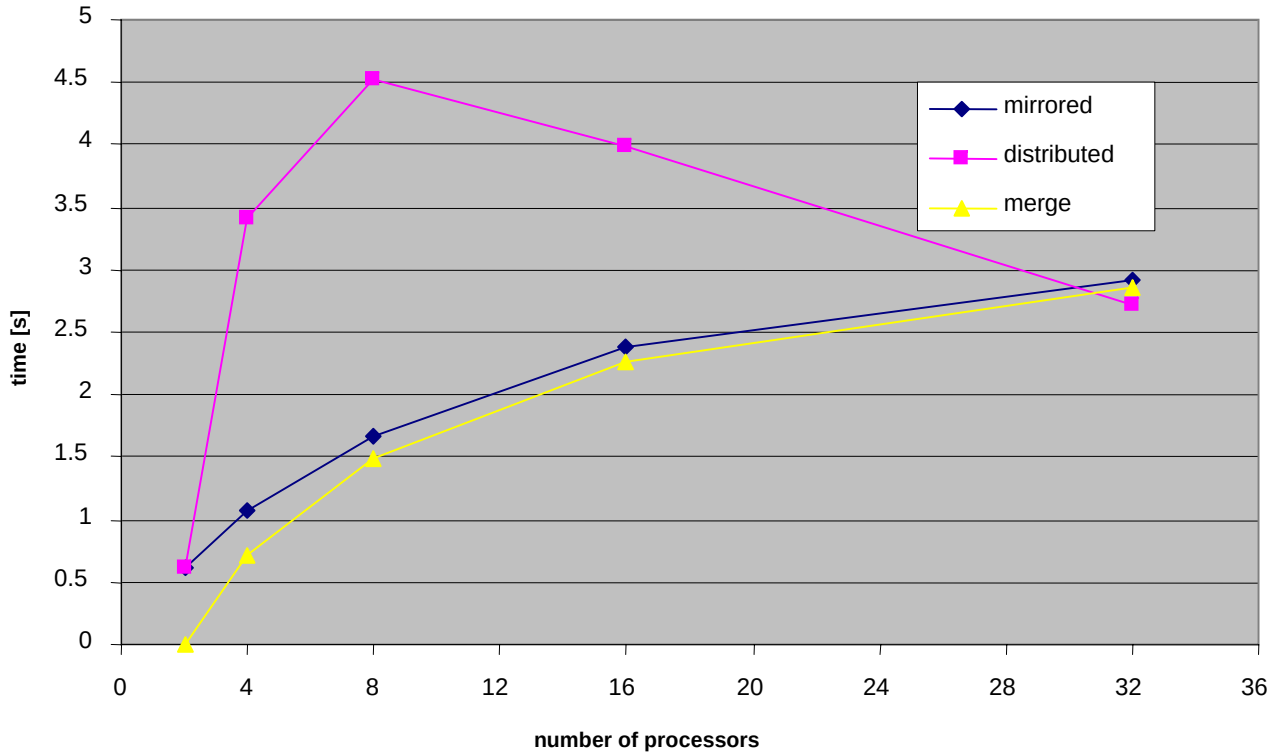


Mirrored Matrix Multiply

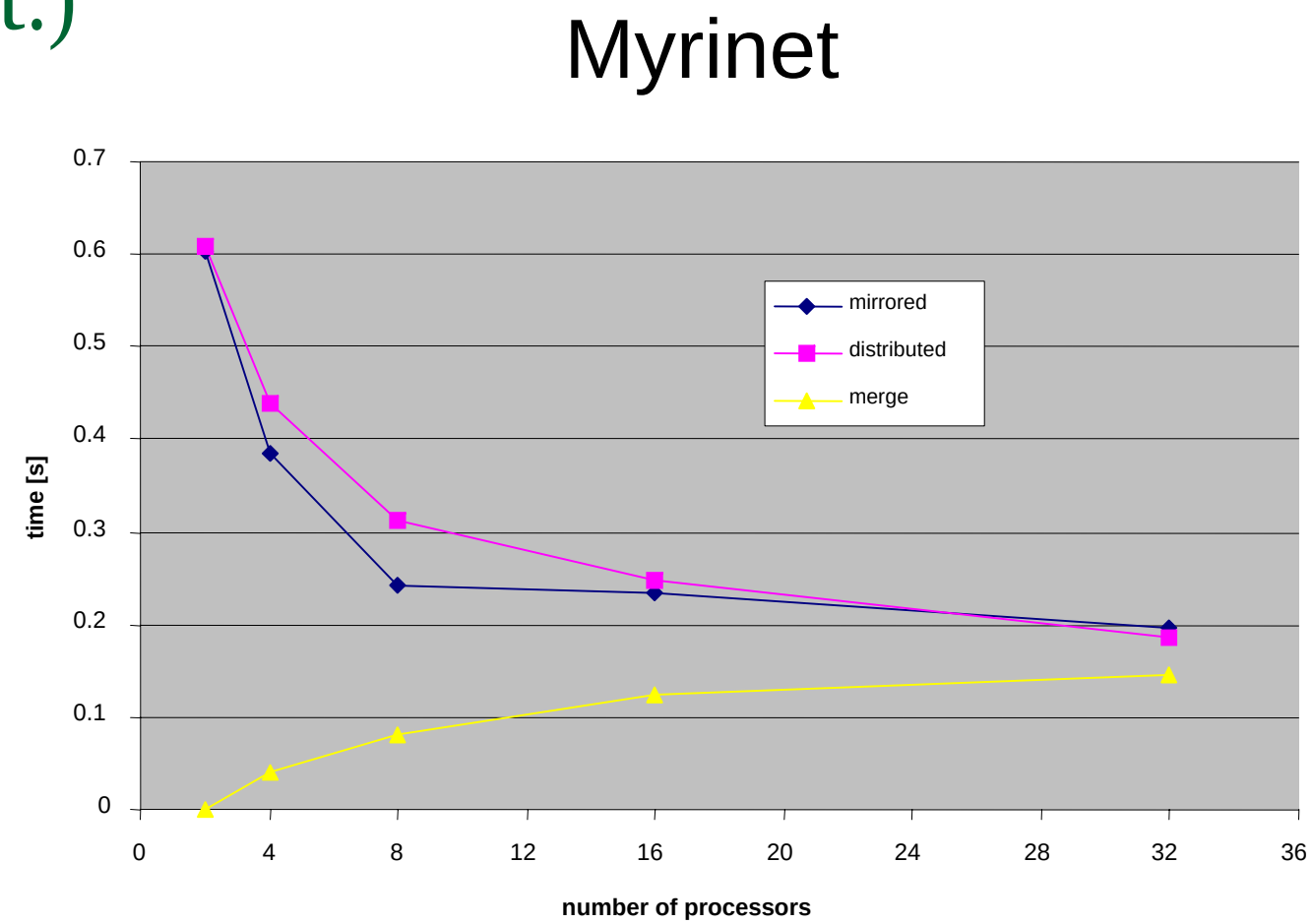


Mirrored Matrix Multiply Performance

Ethernet

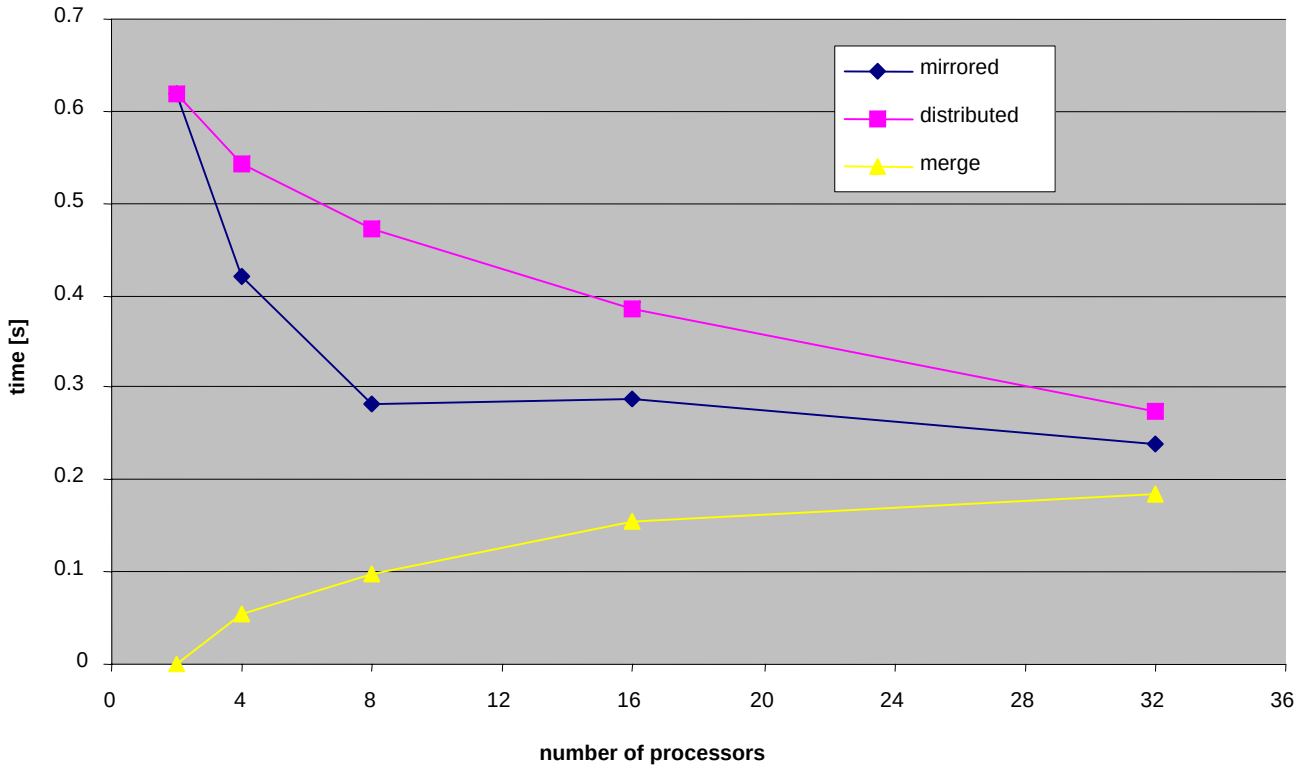


Mirrored Matrix Multiply Performance (cont.)



Mirrored Matrix Multiply Performance (cont.)

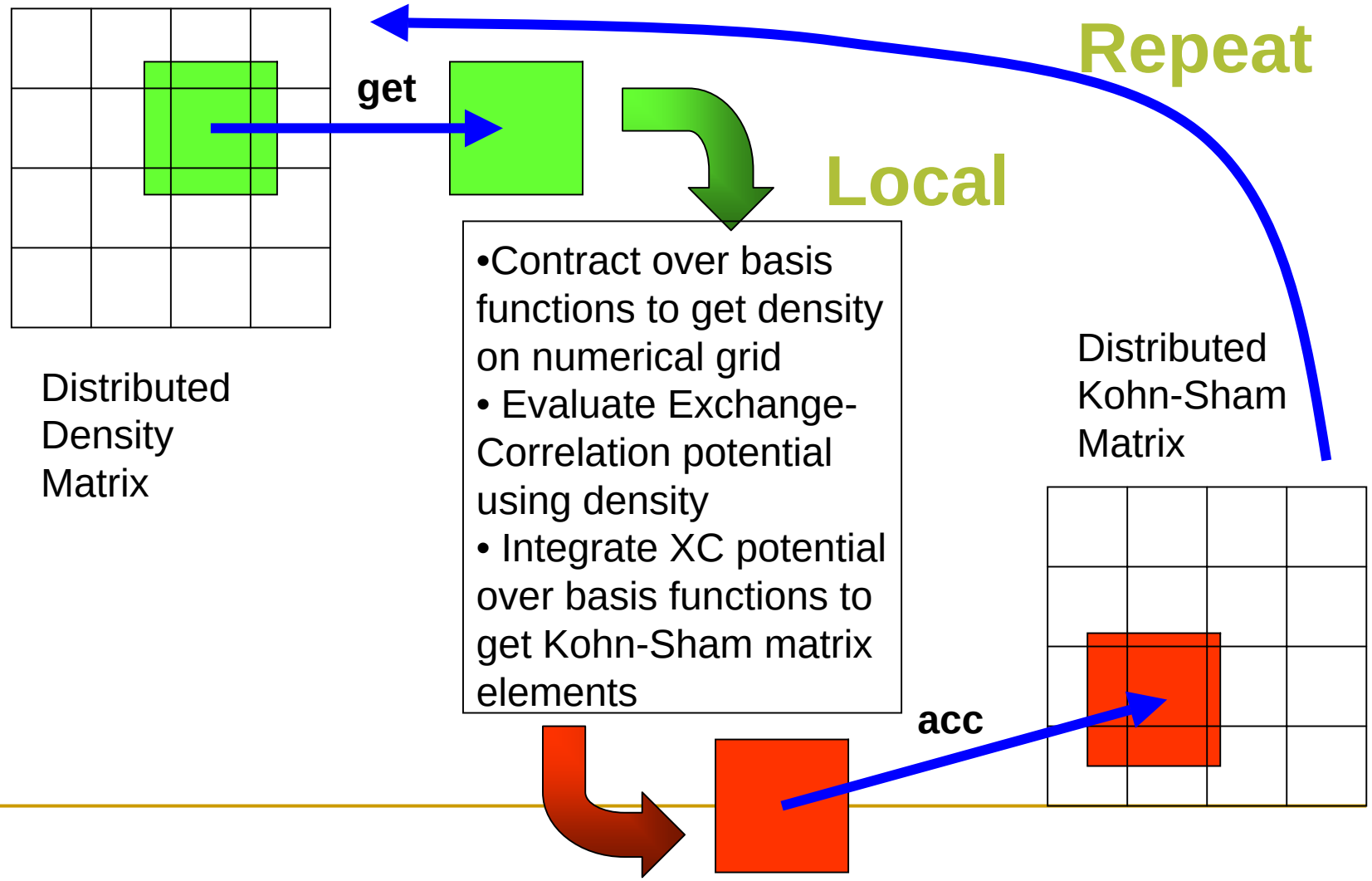
Quadrics



Application: DFT Electronic Structure Calculation

- Need to add copy operation between mirrored and distributed arrays
  - Copying from mirrored array to distributed array can be done completely with shared memory copies
  - Copying from distributed array to mirrored array can be done by zeroing mirrored array, doing shared memory copies from distributed array to mirrored array, and then merging mirrored array

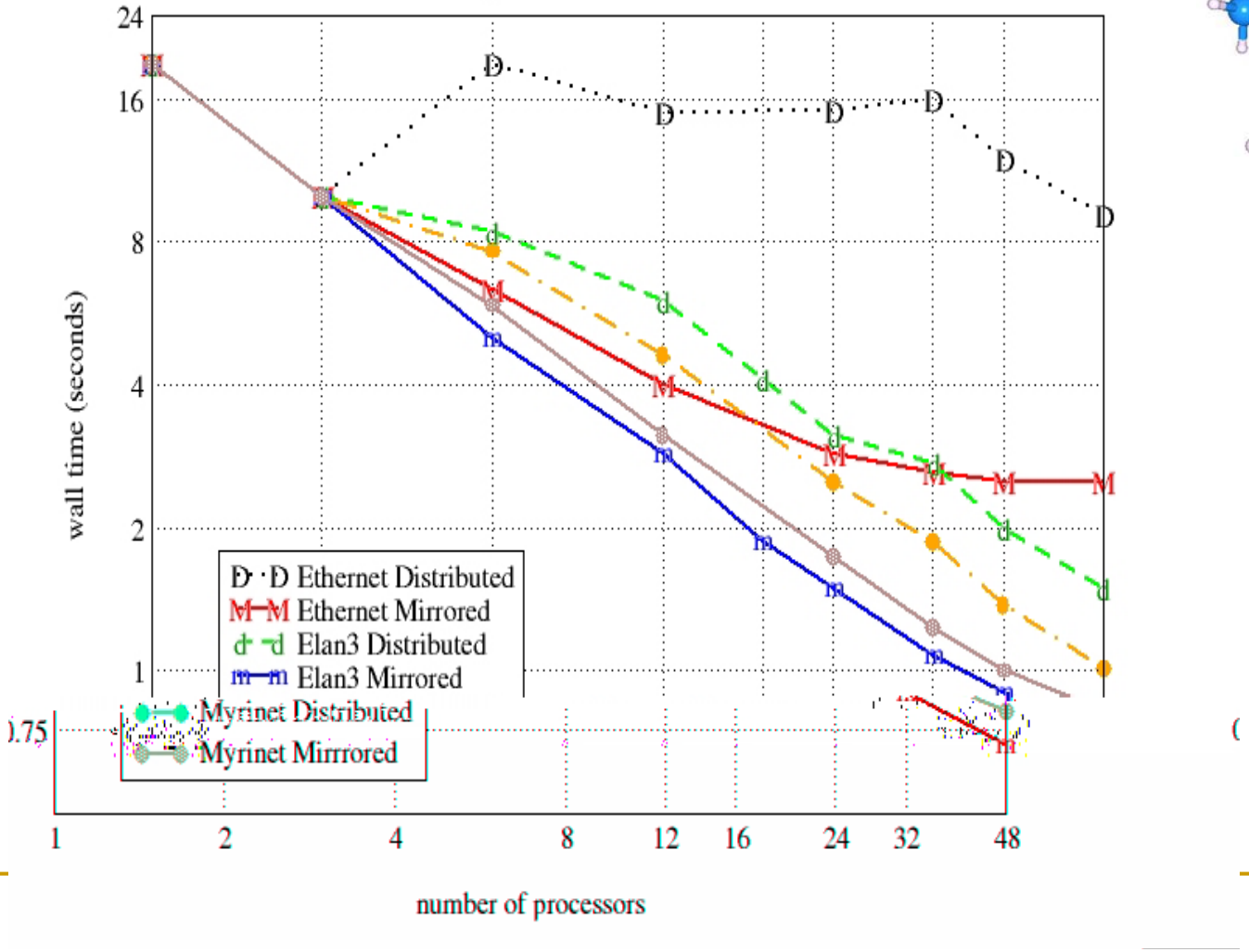
Application: DFT (cont.)



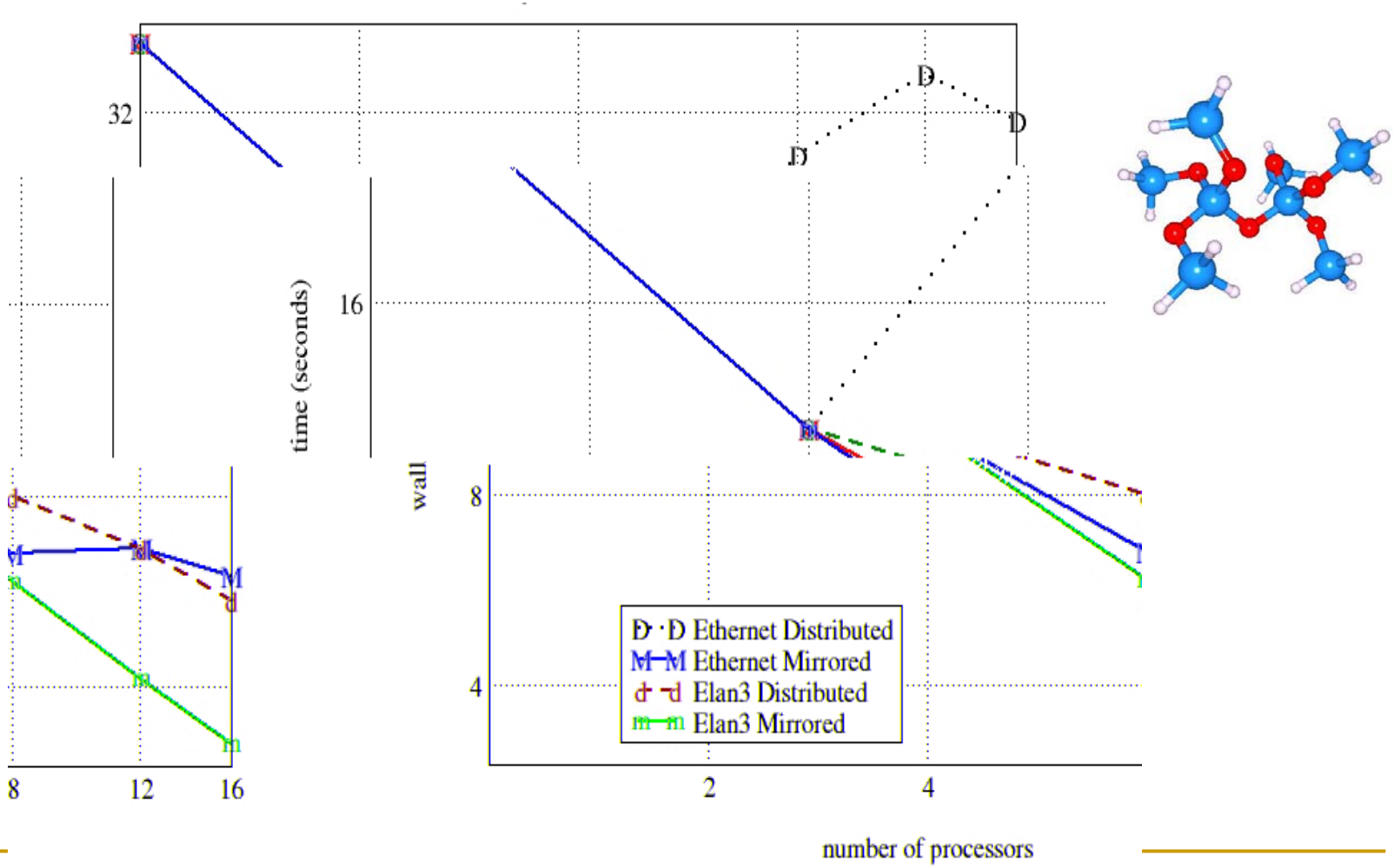
Application: DFT (cont.)

- Mirroring Density Matrix and Kohn-Sham matrix removes all non-local communication associated with get and accumulate operations

Application: DFT Results (Itanium-2)



Application: DFT (EV68 Alpha)



Conclusions

- Mirrored arrays can substantially reduce the communication cost for algorithms executed on clusters or other SMP configurations
- Scalability of algorithms can be significantly extended
- Method is limited by available memory and problem size (if problem does not fit on one node, mirroring cannot be used)

Acknowledgments

- DOE
  - Environmental Molecular Sciences Laboratory
  - Center for Programming Models for Scalable Parallel Computing (MICS/ASCR)
- National Computational Science Alliance under grant CH6MR1P