RUPRECHT-KARLS-
UNIVERSITÄT
HEIDELBERG

Introduction to High Performance Computing

Lecture 08 – Benchmarks

Holger Fröning
Institut für Technische Informatik
Universität Heidelberg

Overview

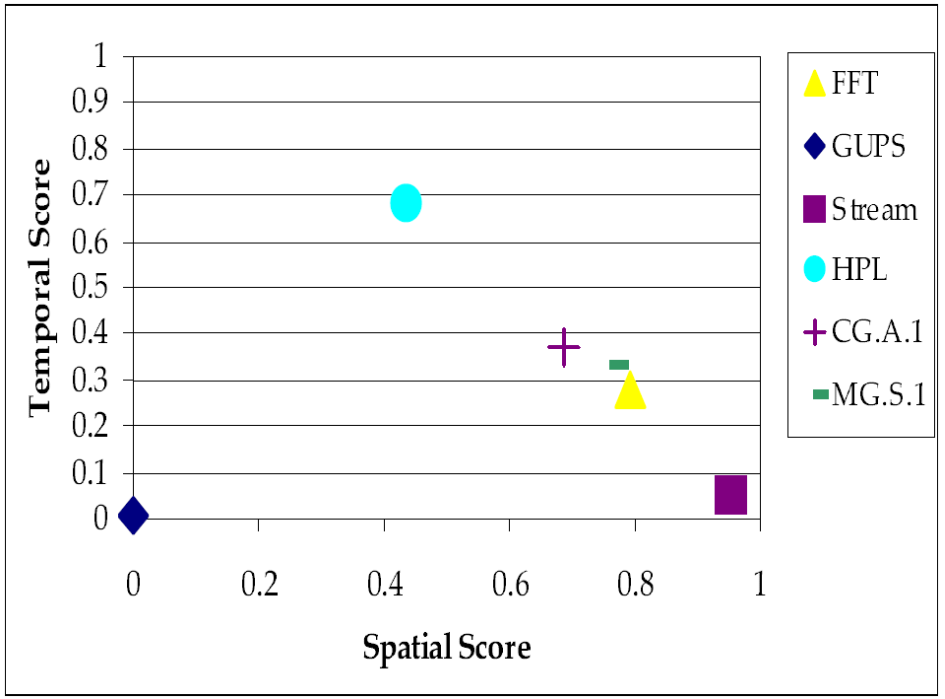
- Micro-benchmarks
  - Specialized to characterize only a certain aspect
- Latency
- Bandwidth
- Message rate
- Overlap

- Impact towards application performance unclear

- Application-level or complex benchmarks
  - Mimic the behaviour of applications or algorithms
- Examples for algorithms
  - LU decomposition, FFT, …
- Examples for applications
  - Weather and climate research, crash simulations, molecular dynamics, protein folding, graph computations, …

Characterization of Benchmarks/Applications

- Temporal and spatial locality for MPI applications
  - FFT: Fast Fourier Transform
  - GUPS: RandomAccess
  - Stream: Stream benchmark
  - HPL: High Performance Linpack
  - CG & MG: part of NAS Parallel Benchmark Suite (NPB)
- Metric how much pressure an application puts on the network?

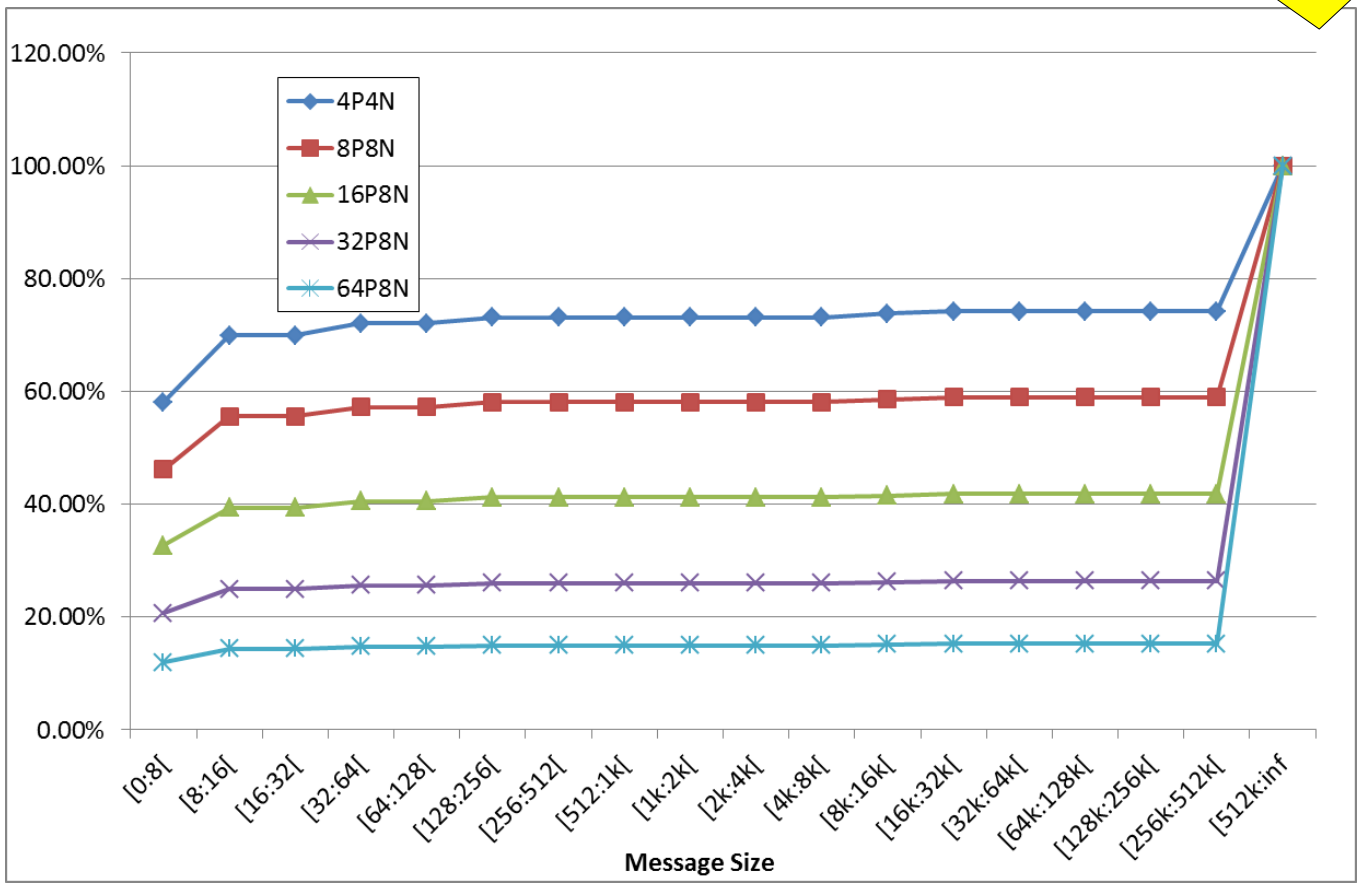


[Weinberg et. al, Quantifying Locality In The Memory Access Patterns of HPC Applications, Supercomputing 2005]

Characterization of Benchmarks/Applications

- Benchmark: MPIFFT
  - Medium temporal locality, high spatial locality

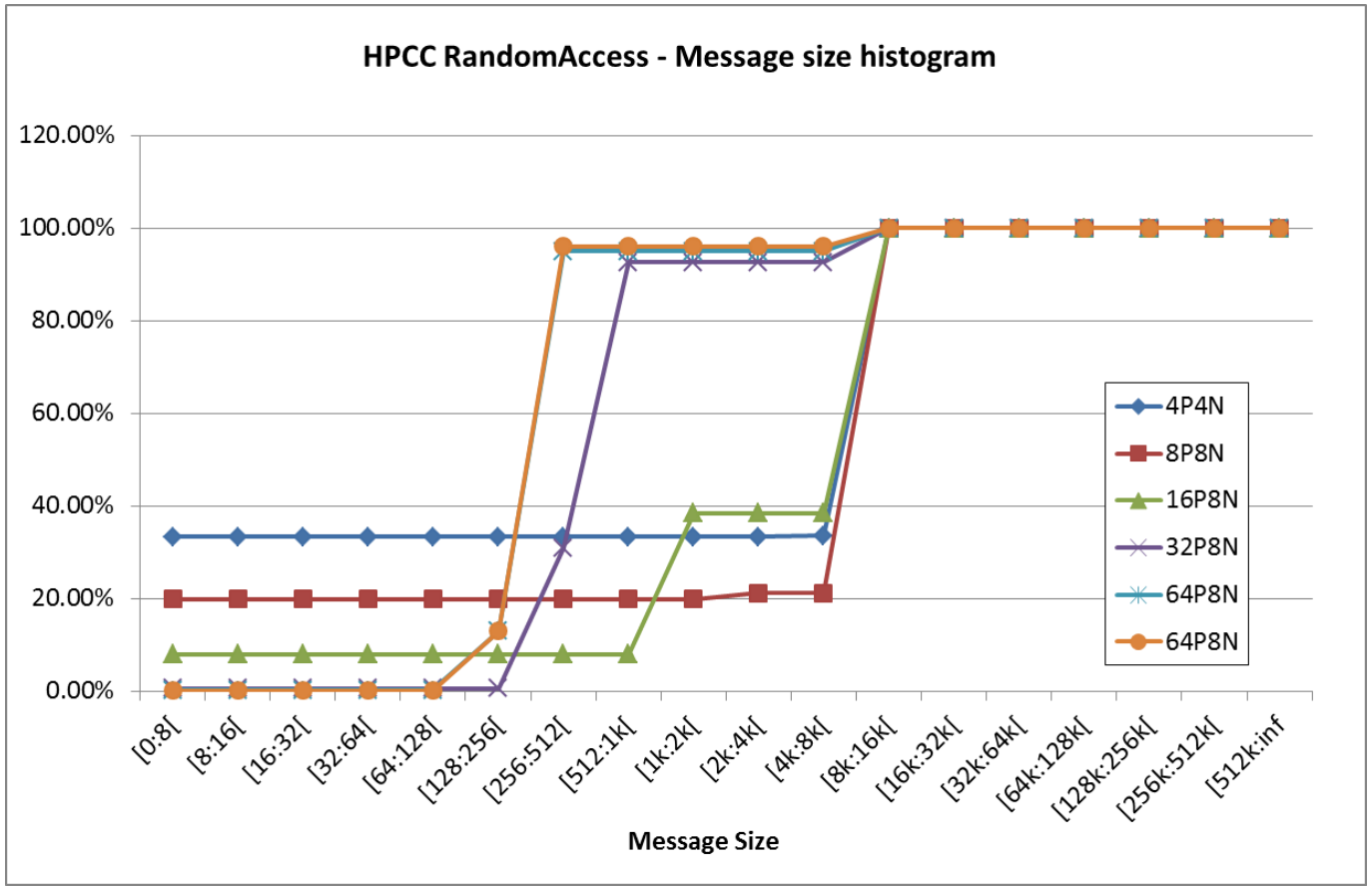
Cumulative distribution function (CDF)



Shift towards **larger messages** with increasing process count!

Characterization of Benchmarks/Applications

- Benchmark: RandomAccess (GUPS)
  - Both spatial and temporal locality very low



Shift towards **smaller messages** with increasing process count!

Messaging Software Stack

Application
- Need to communicate & synchronize

MPI Interface
- MPI_SEND/RECV/BARRIER

MPI Layer
- Translation into available communication primitives
- Messaging protocol (eager, rendezvous)

API Interface
- Send/recv, Put/get
- Specialized calls like barrier, multicast

API Layer
- Flow Control

Network Layer
- Packets no larger than MTU

- Reliability
- Ordering
- etc

Characterization of Benchmarks/Applications

- Interconnection Network (IN) is the bottleneck
  - Bandwidth & latency mismatch
  - For almost any application/benchmark
- But how big is the pressure?
  - Data volume transferred?
  - Number of messages?
  - Time spent in MPI layers?
- Maybe all of them - everything depends on the application
  - **MPI time** – the execution time fraction spent in MPI layers
  - **Message Density Distribution** – message count between pairs
  - **Data Density Distribution** – data transferred between pairs
  - **Message Size Distribution** – message sizes used

Benchmarks

Overview

- There is no „one fits all“ benchmark
- We will look at:
  - High Performance LINPACK – TOP500
  - NAMD – molecular dynamics
  - NAS Parallel Benchmarks
  - HPC Challenge
  - Weather Research and Forecasting Model
  - GRAPH500
- Many more out there:
  - GROMACS – molecular dynamics
  - SPECMPI2007
  - ANSYS FLUENT CFD
  - See 13 dwarfes in „The Landscape of Parallel Computing Research: A View From Berkeley”
  - HPCG
  - …

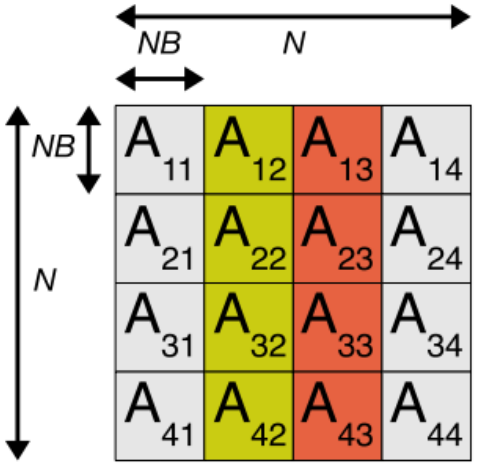
High Performance Linpack (HPL)

- Used to rank supercomputers in the TOP500 list
- Solve a dense NxN system of linear equations (Ax = b)
  - DAXPY most important (double precision alpha * x + y; alpha scalar, x,y vectors)
  - $f = 2/3 \cdot N^3 + 2 \cdot N^2 = O(N^3)$
- Computationally very intensive
- Rather relies on computing performance and cache/ memory capacity than on messaging
  - See performance of GE based supercomputers in the TOP500

Thom Dunning, director of the National Center for Supercomputing Application, 2011:
"The Linpack benchmark is one of those interesting phenomena -- almost anyone who knows about it will deride its utility. They understand its limitations but it has mindshare because it's the one number we've all bought into over the years."
(http://www.technologyreview.com/blog/mimssbits/25981/)

High Performance Linpack (HPL)

- Data is distributed on a PxQ grid of processes, each owning a block of the size NBxNB
- Plenty of more parameters
  - Ordered by importance on the right
  - Hand optimization!
  - Keep CPUs busy!
- HPL should achieve >= 95% efficiency when run on a single node

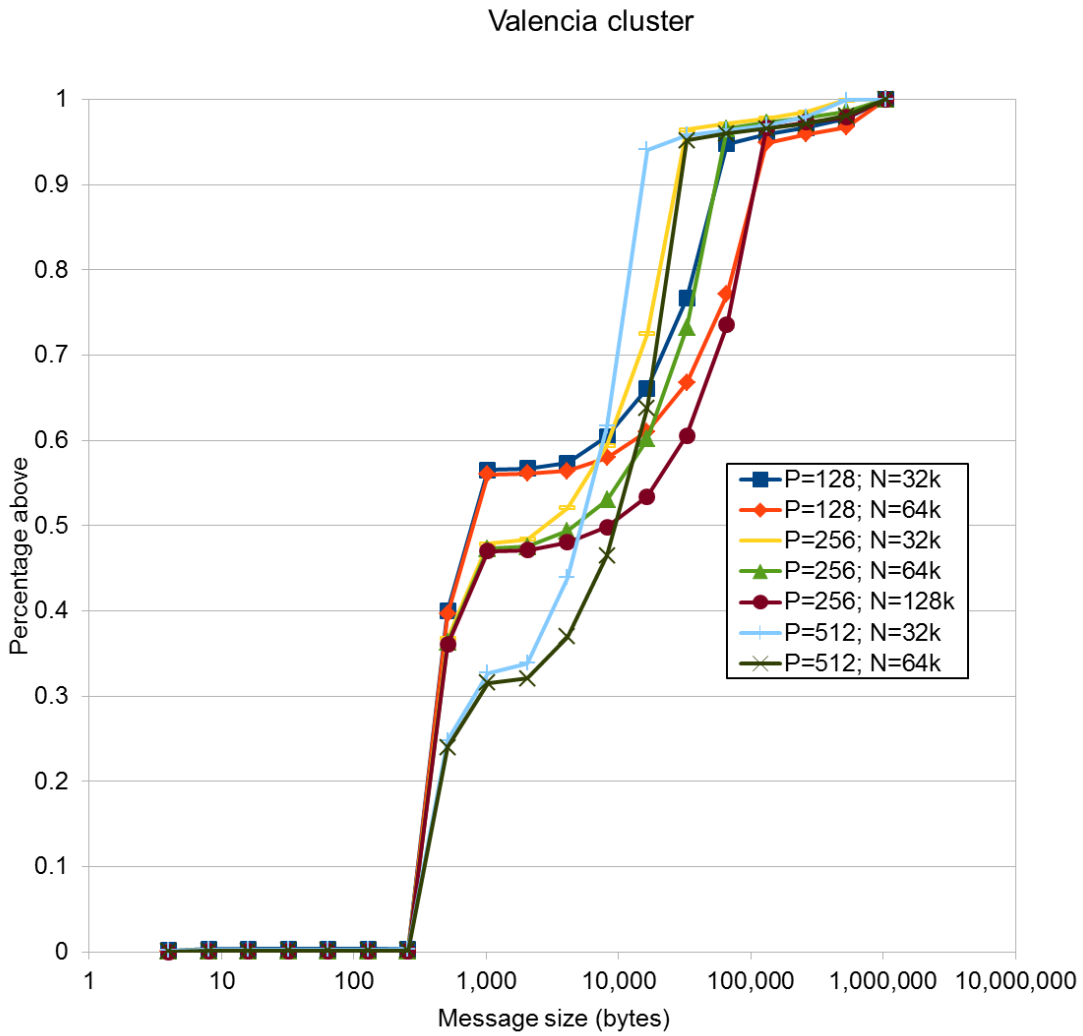


http://hpc-ga-bench.gforge.uni.lu/

Optimization parameter	Description
NB	Block size
N	Problem size
P	Process matrix rows
Q	Process matrix columns
BCAST	Broadcast algorithm
DEPTH	Lookahead depth
FSWAP	Swapping algorithm
NBMIN	Recursive stopping criterion
NDIV	Number of panels in recursion
PFACT	Panel factorization algorithm
TSWAP	Swapping threshold for mix algorithm
…	…

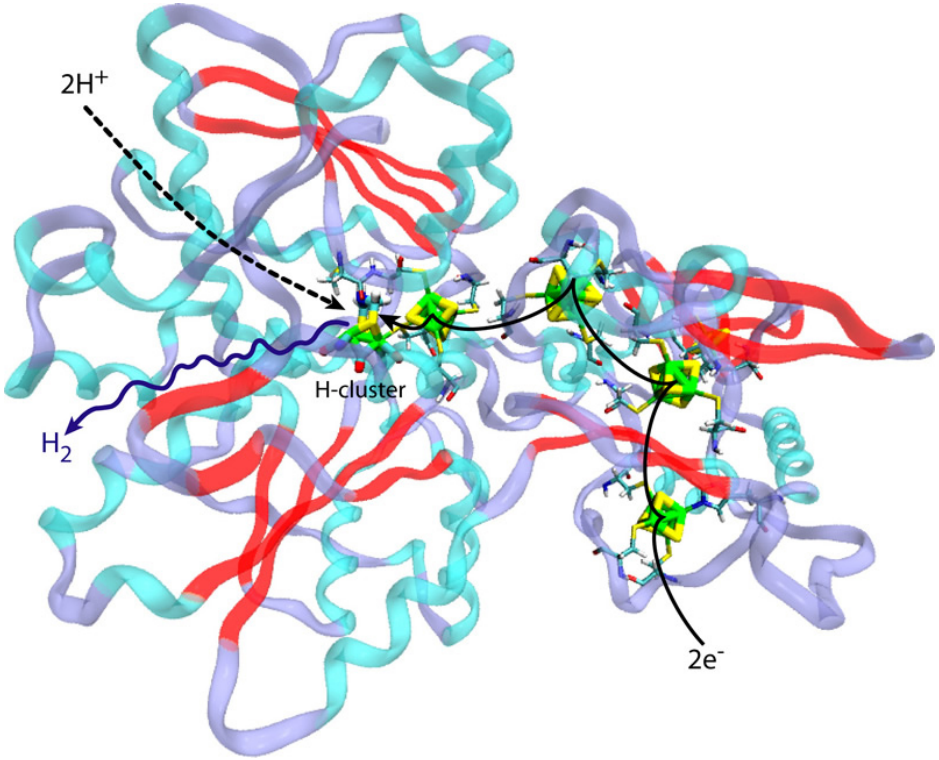
High Performance Linpack (HPL)

- Characteristics
  - Mostly MPI_Recv, MPI_Send, MPI_Irecv
  - MPI time is about 25%
- Performance numbers
  - See TOP500



NAMD

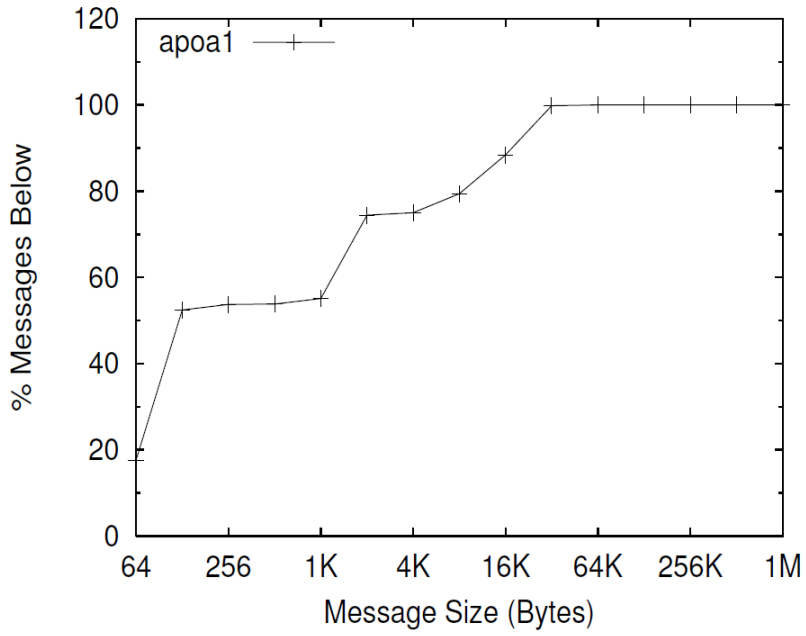
- Fully featured, production molecular dynamics program for high performance simulation of large bio-molecular systems
  - Electrostatic and Van der Waals forces
  - Millions of objects
  - Open-source
- N-Body problem, f = $O(N^2)$
- Time scale of 1fs (10^{-15}s)
- Example workload: Satellite Tobacco Mosaic Virus (STMV)
  - 100M atoms, 160 genes
  - Petascale-class: 100ns/day of simulation time
- Complex structures like parasites: ~ 6k genes
  - Estimation: 1400x runtime increase

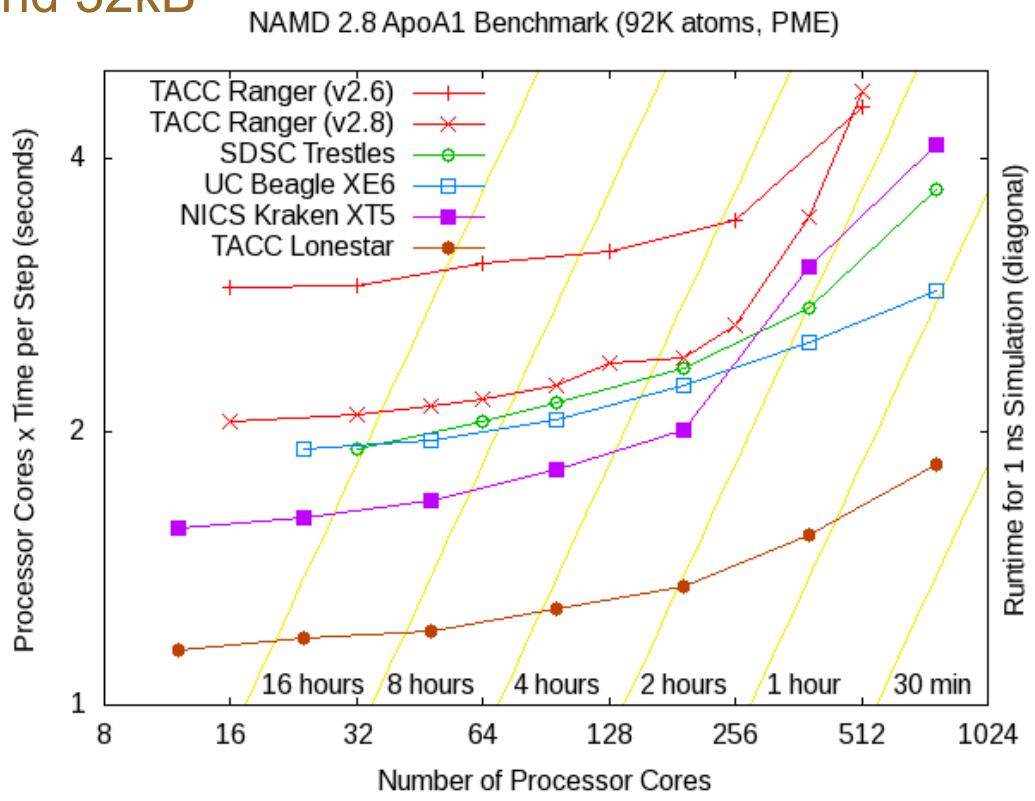


Source: http://www.ncsa.illinois.edu

NAMD

- ApoA1: protein with 92K atoms of lipid, protein, and water
  - Mostly MPI_Isend, MPI_Send, MPI_Recv, MPI_Barrier
  - 50% below 128B
  - Other 50% between 128 and 32kB
- MPI time is about 24%

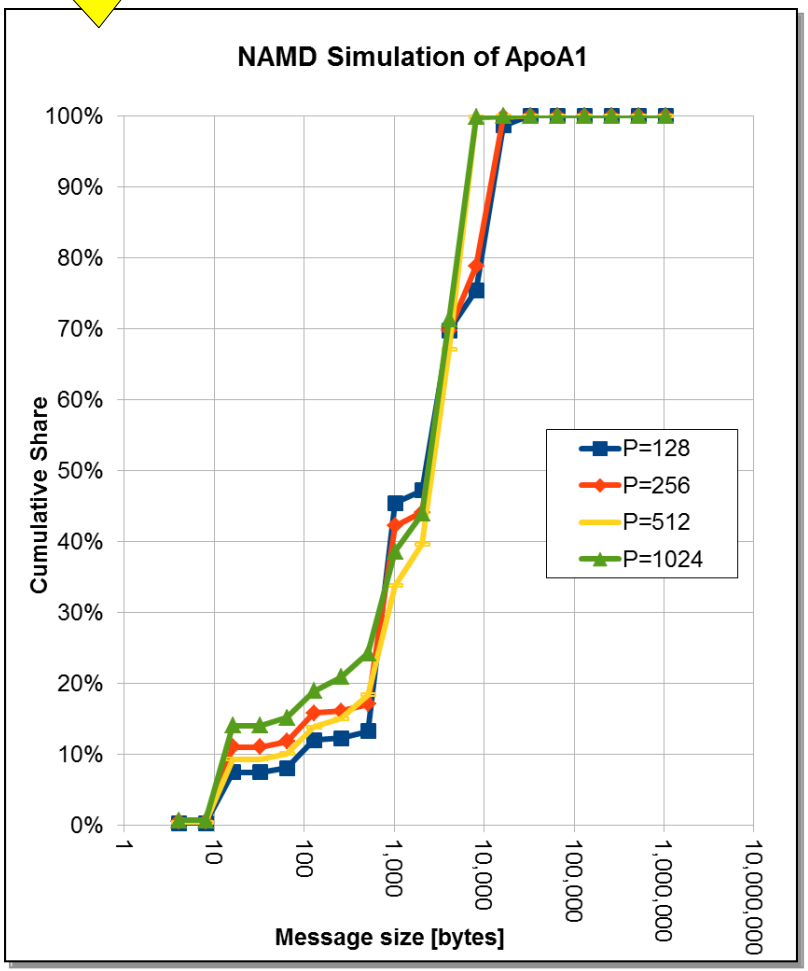




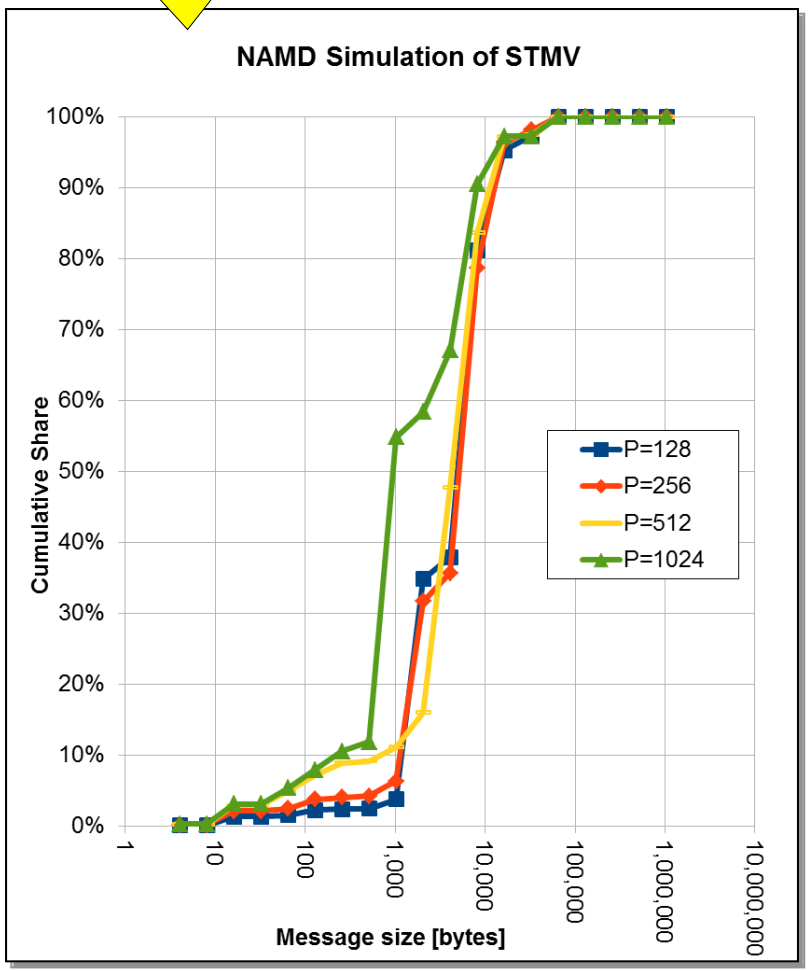
http://www.ks.uiuc.edu/Research/namd/performance.html

NAMD

92k atoms



1M atoms



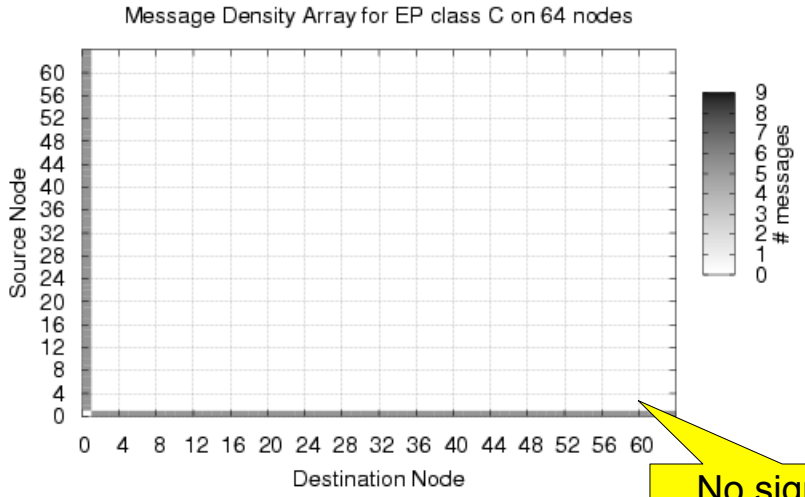
NAS Parallel Benchmarks (NPB)

- EP: „Embarrassingly Parallel“ (0% MPI time)
  - Generates pairs of random values typically used in Monte Carlo simulations
  - Upper achievable limit of floating point performance, no significant communication
- MG: „Multi-grid kernel“ (9% MPI time)
  - Approximate solution to the discrete Poisson problem (used in CFDs)
  - Highly structured long distance communication, both short and long data communication
- CG: „Conjugate Gradient“ (34% MPI time)
  - Approximation to the smallest eigenvalue of large, sparse, symmetric matrix
  - Unstructured grid communication, irregular long distance communication, unstructured matrix-vector multiplication
- FT: „3D-FFT PDE“ (37% MPI time)
  - Solve 3D partial differential equation (PDE) using forward and inverse FFT
  - Heavily relies on long distance communication
- IS: „Integer Sort“ (47% MPI time)
  - Sort N integer keys in parallel, often used in particle methods
- BT: „Block Tridiagonal” (10% MPI time)
  - Solve nonlinear PDE using block triagonal matrix algorithms

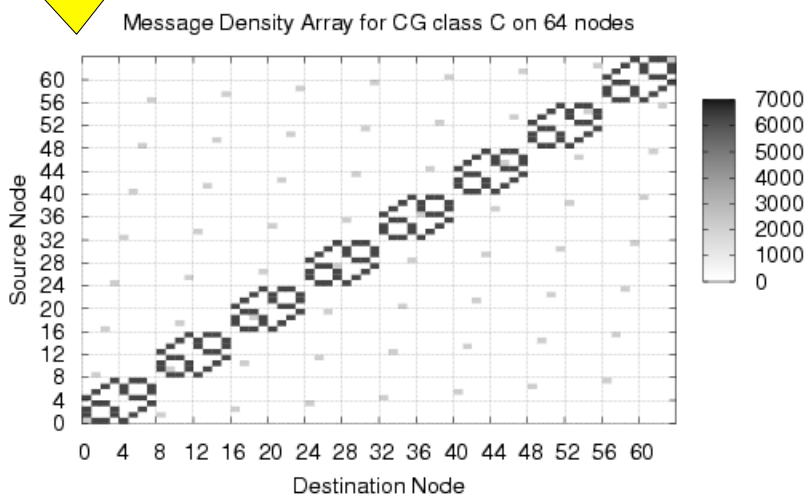
Numbers from [S.Sur, M. J. Koop, D. K. Panda, High-Performance and Scalable MPI over InfniBand With Reduced Memory Usage: An In-Depth Performance Analysis, 2006 ACM/IEEE conference on Supercomputing]

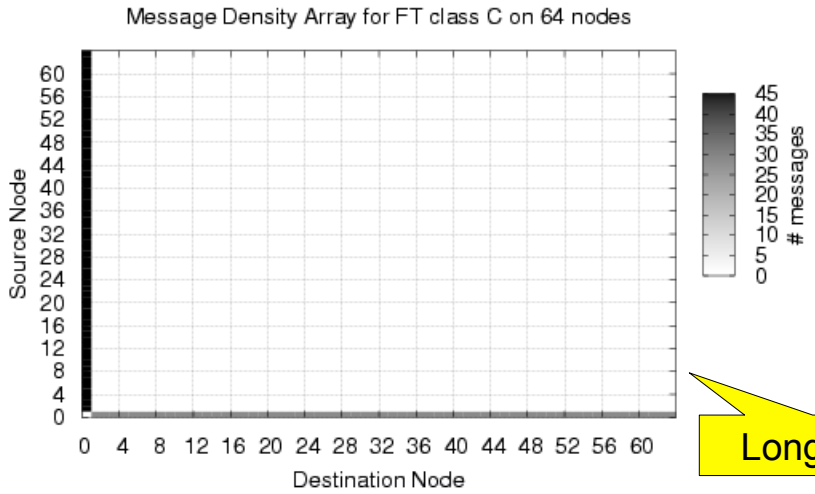
NPB - Message Density Distribution

Unstructured

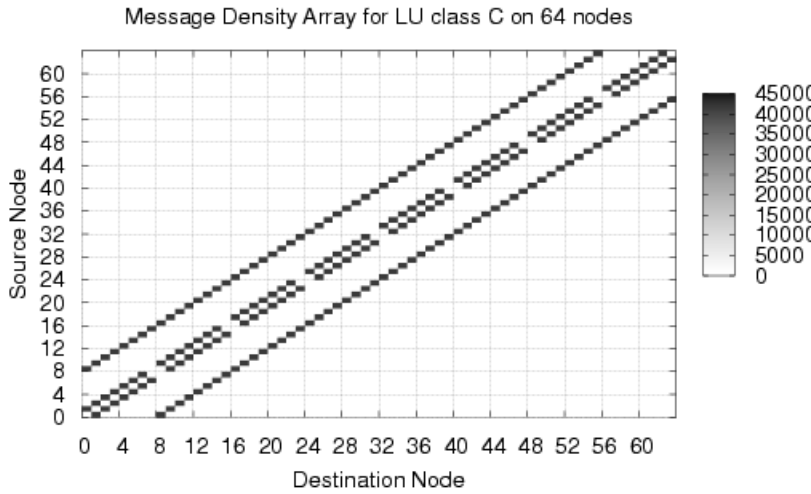


No significant communication





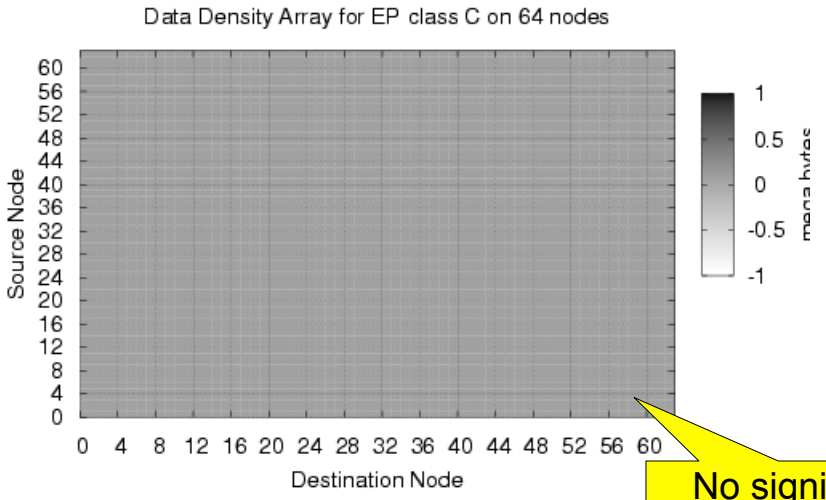
Long range

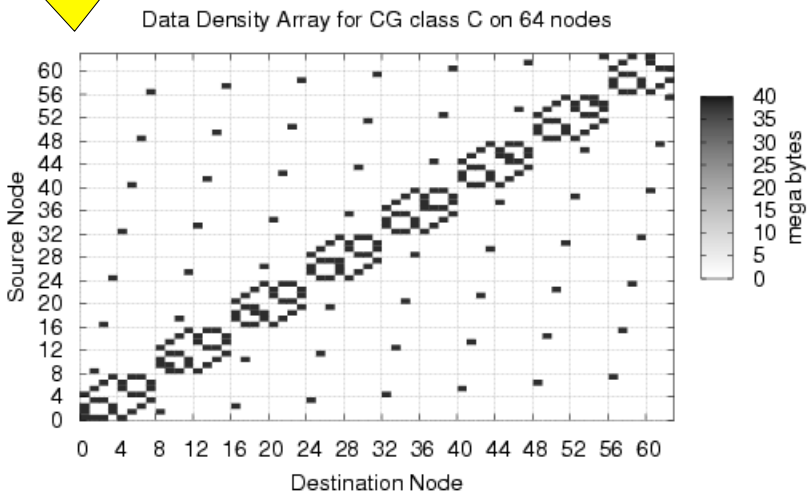


http://www.cs.sandia.gov/~rolf/NAS/index.html

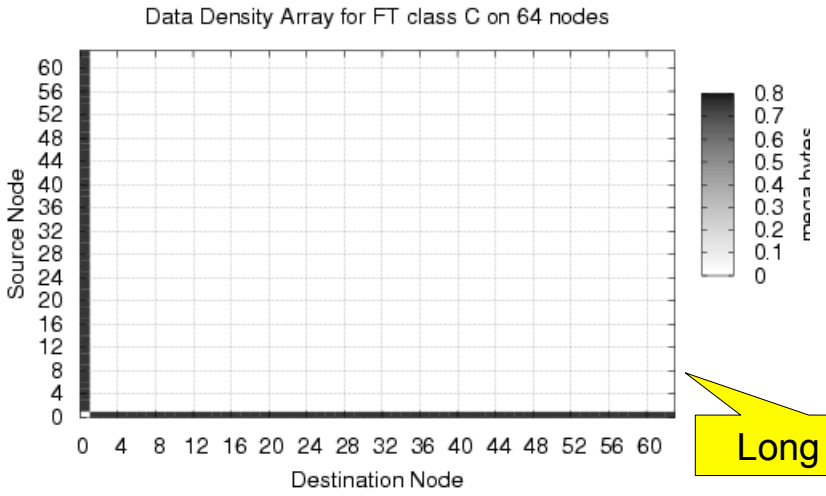
NPB – Data Density Distribution

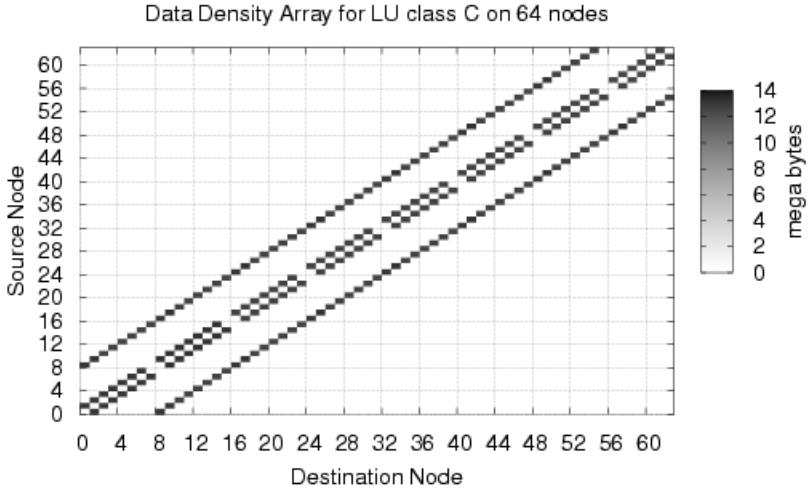
Unstructured





No significant communication

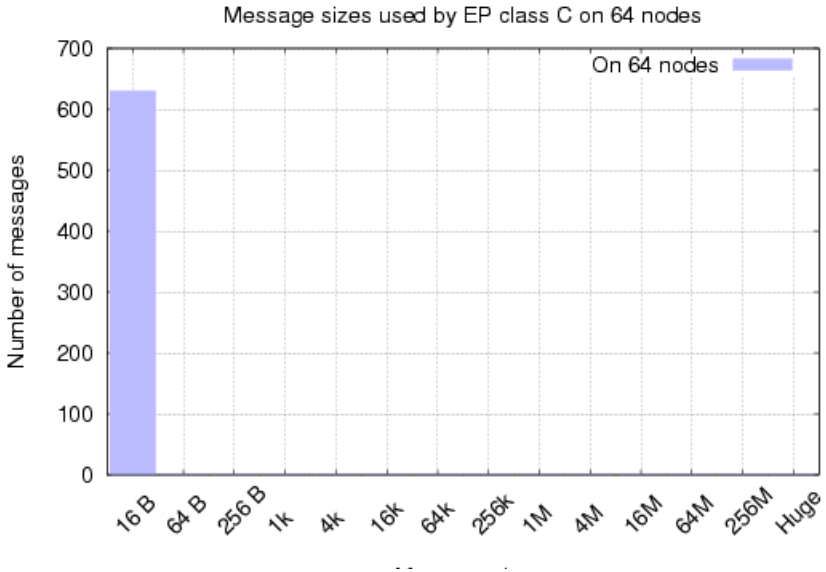


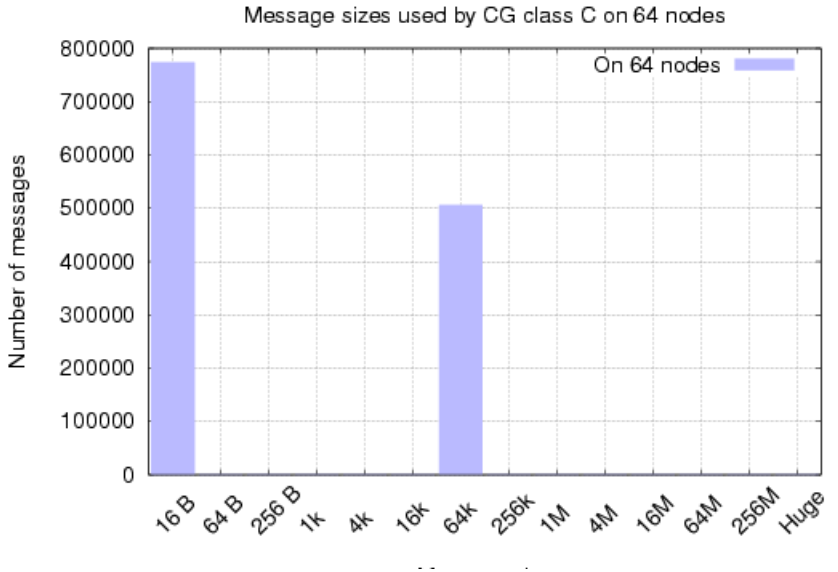


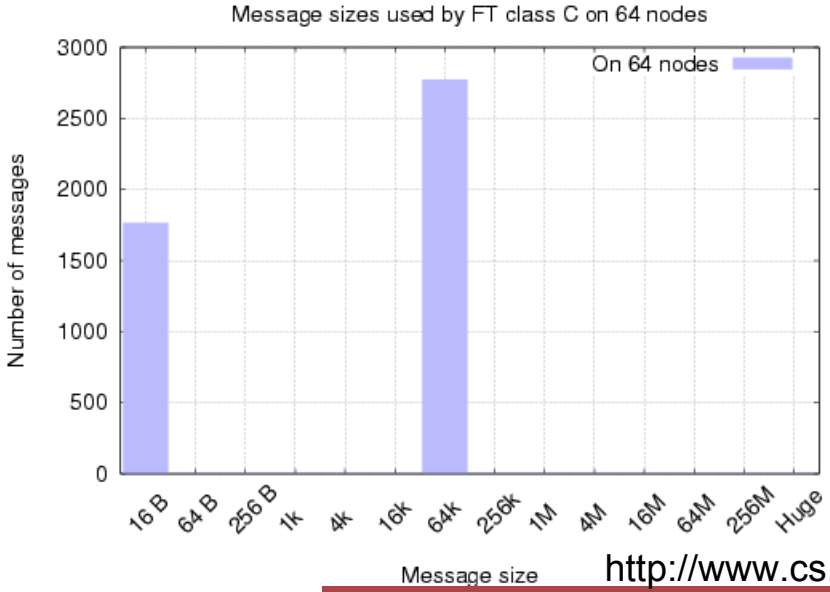
Long range

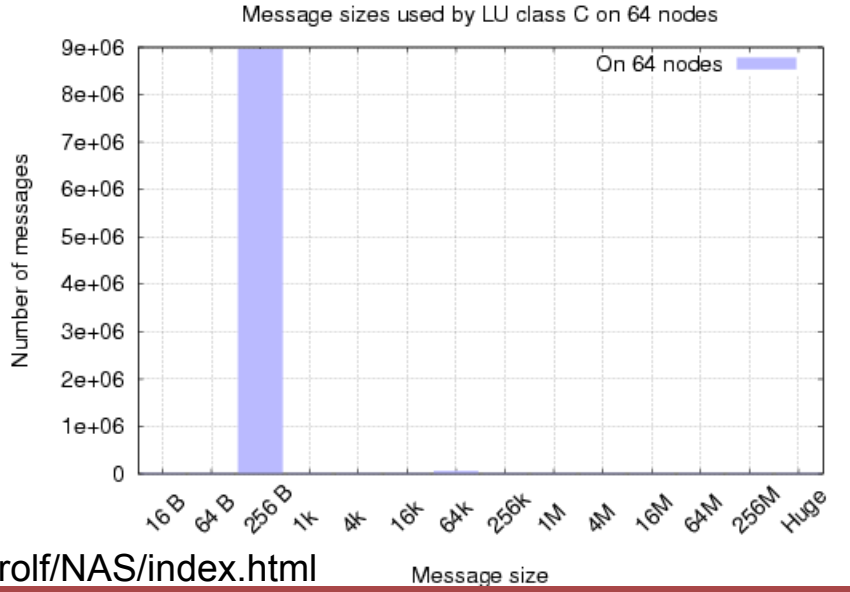
http://www.cs.sandia.gov/~rolf/NAS/index.html

NPB - Message Size Distribution





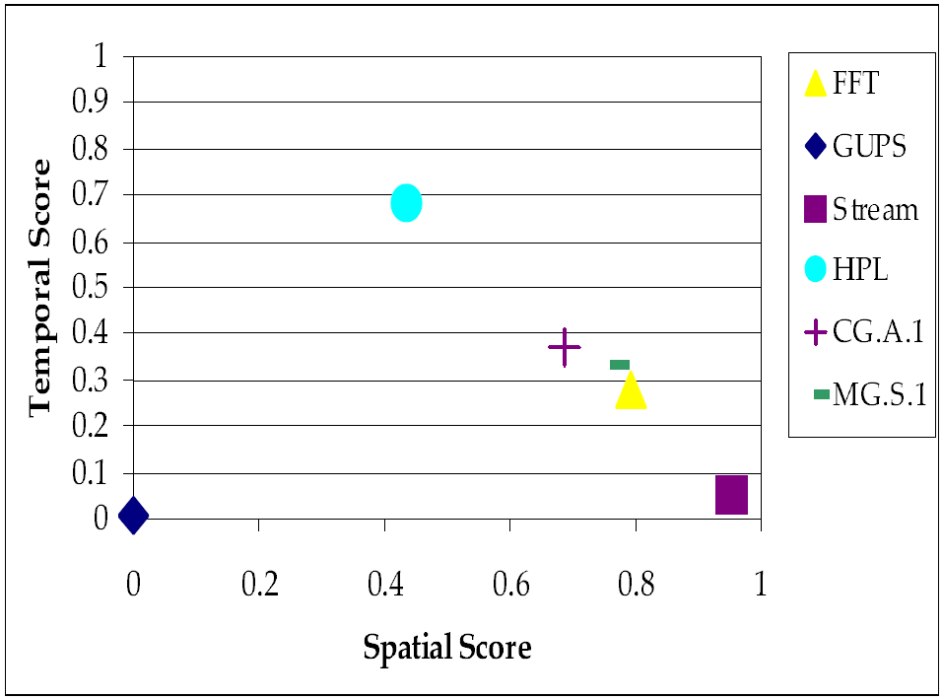




http://www.cs.sandia.gov/~rolf/NAS/index.html

HPC Challenge (HPCC)

- http://icl.cs.utk.edu/hpcc/
- Suite consisting of 7 tests
  1. **HPL**: well-known
  2. DGEMM: matrix multiply
  3. STREAM: synthetic benchmark that measures sustainable memory bandwidth (local!)
  4. **PTRANS**: parallel matrix transpose, good indicator for total network capacity
  5. **RandomAccess**: integer random updates
  6. **FFT**: one-dimensional discrete fourier transform
  7. Communication bandwidth and latency

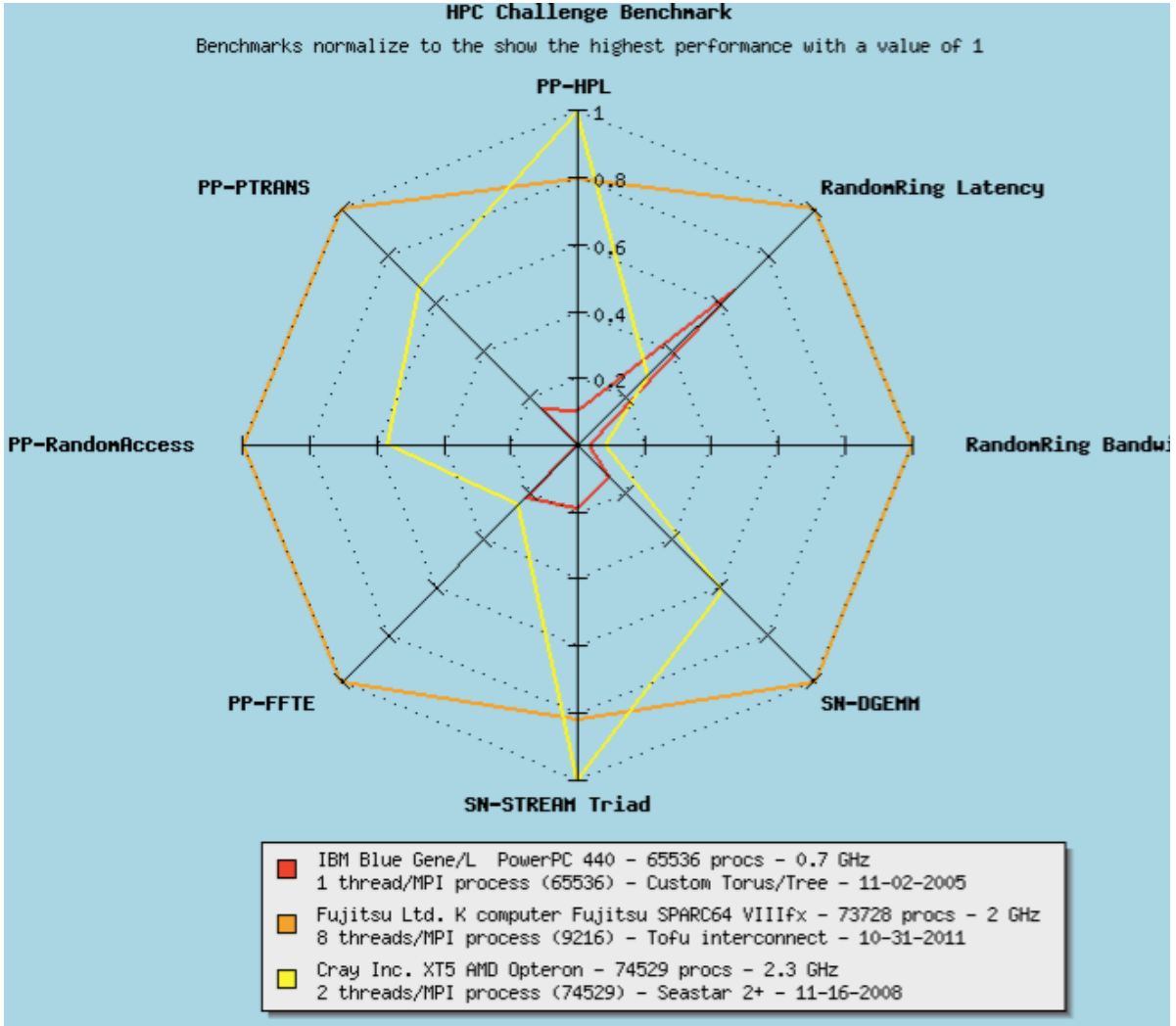


[Weinberg et. al, Quantifying Locality In The Memory Access Patterns of HPC Applications, SC 2005]

Benchmark	Spatial locality	Temporal locality	MPI time fraction (64P8N)
HPCC RandomAccess [19]	Low	Low	89.00%
HPCC MPIFFT [19]	Low	High	77.18%
WRF [12]	High	High	45.00%

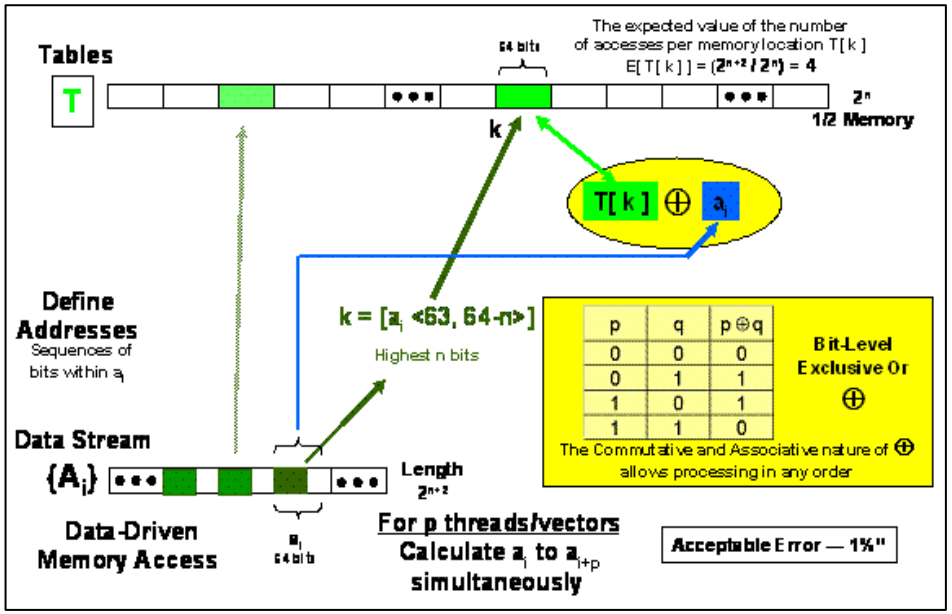
HPC Challenge (HPCC)

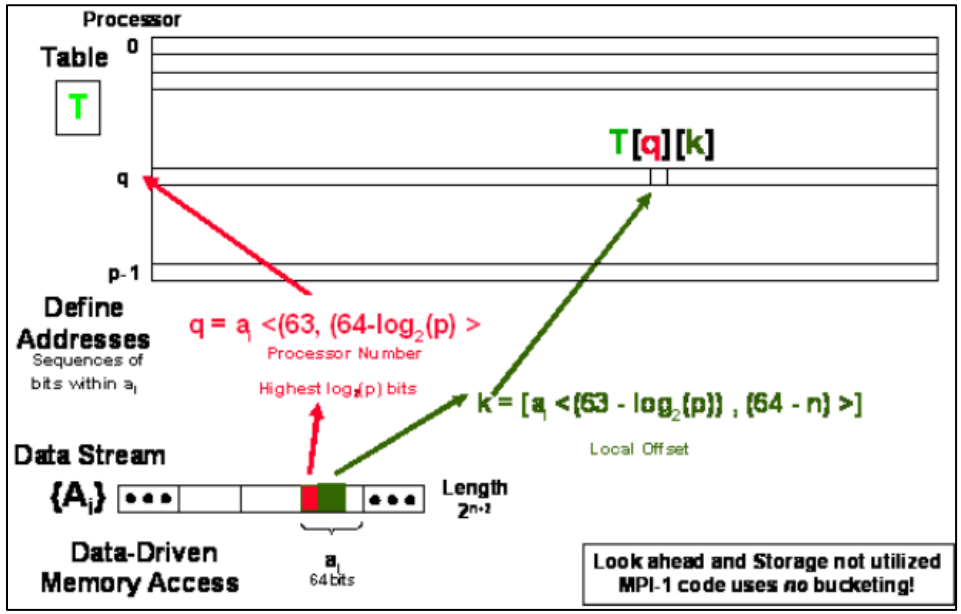
- Result database
  - Blue Gene P=64k
  - K computer P=74k
  - Cray XT5 P=74k



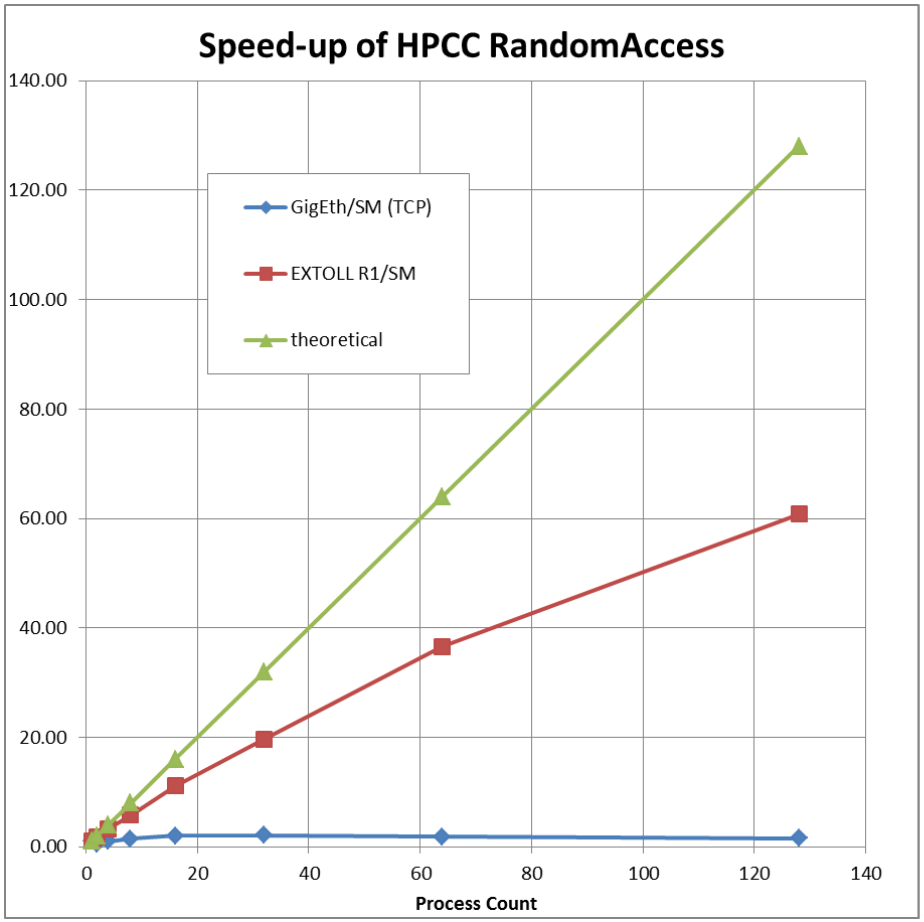
HPC Challenge (HPCC) - RandomAccess

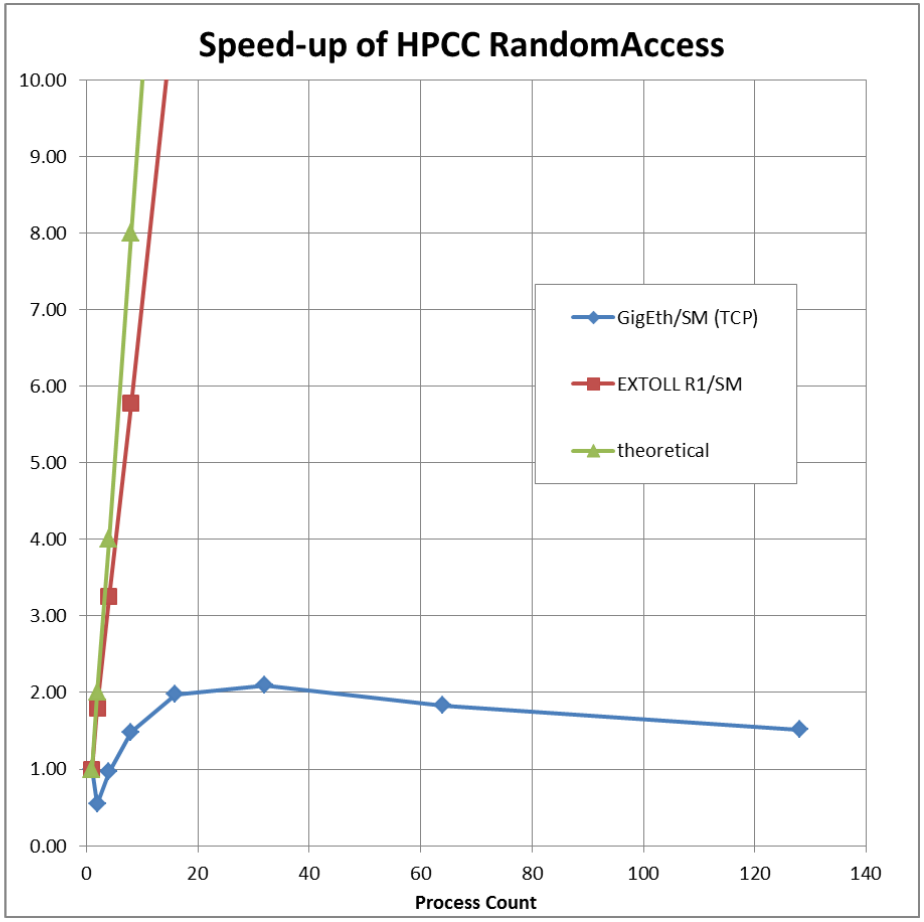
- Random location memory performance
  - Current trends favor stride performance at the expense of random access performance
- Test
  - Distribute table ***T*** over ***p*** processes
  - Random stream of integers ***{A$_i$}*** describes with addresses to update with which value
  - Update is commutative and associative!





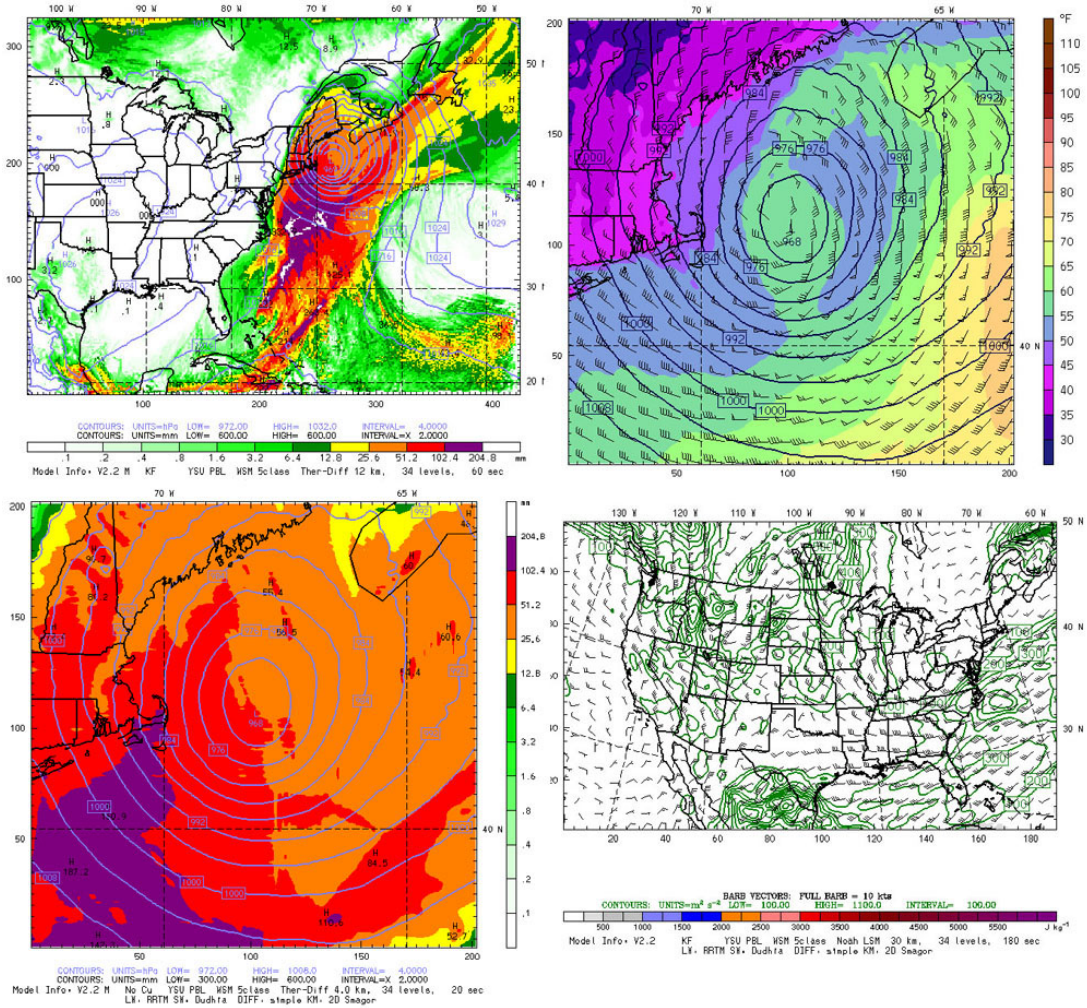
HPC Challenge (HPCC) - RandomAccess





Benchmark: Weather Research Forecast (WRF)

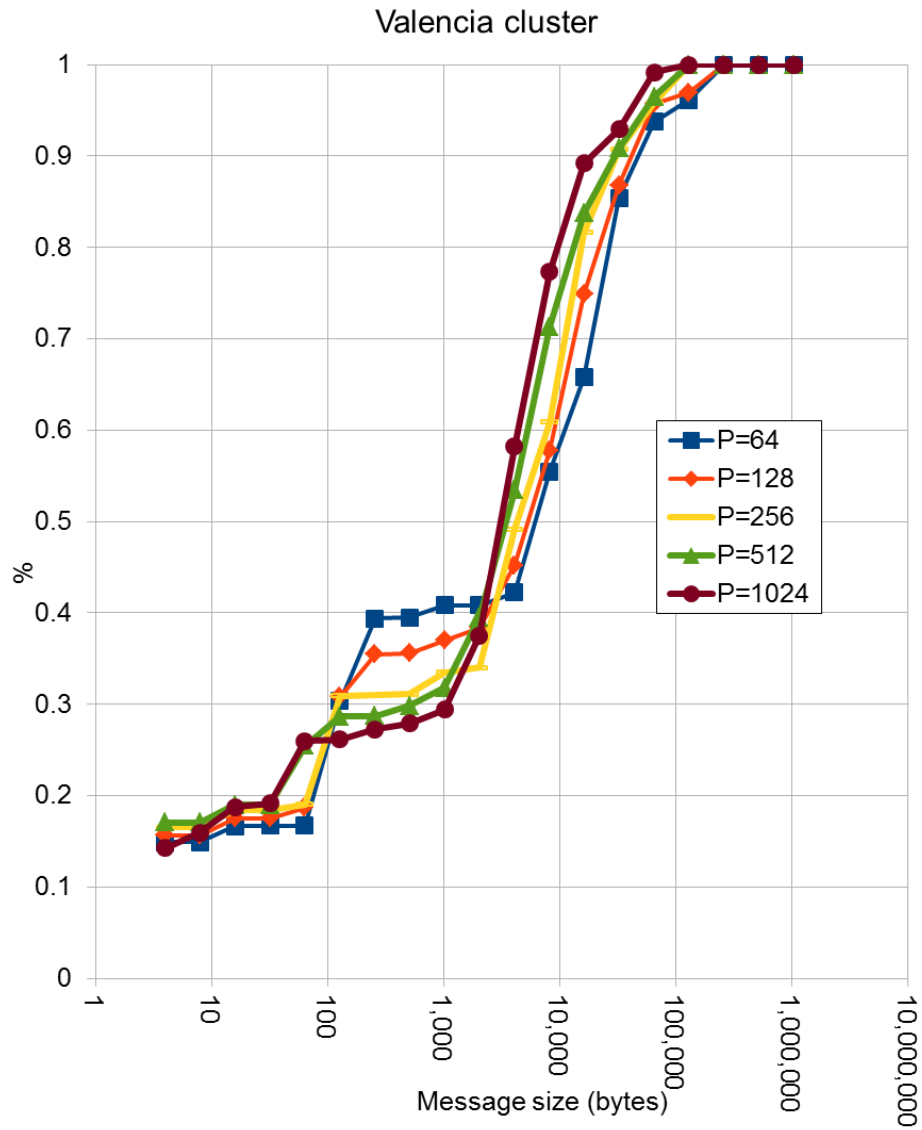
- Numerical weather prediction system
  - Operational forecast
  - Atmospheric research
- Weather and climate modeling
- Application and benchmark
  - Standard input data available
  - http://wrf-model.org



Courtesy: WRF

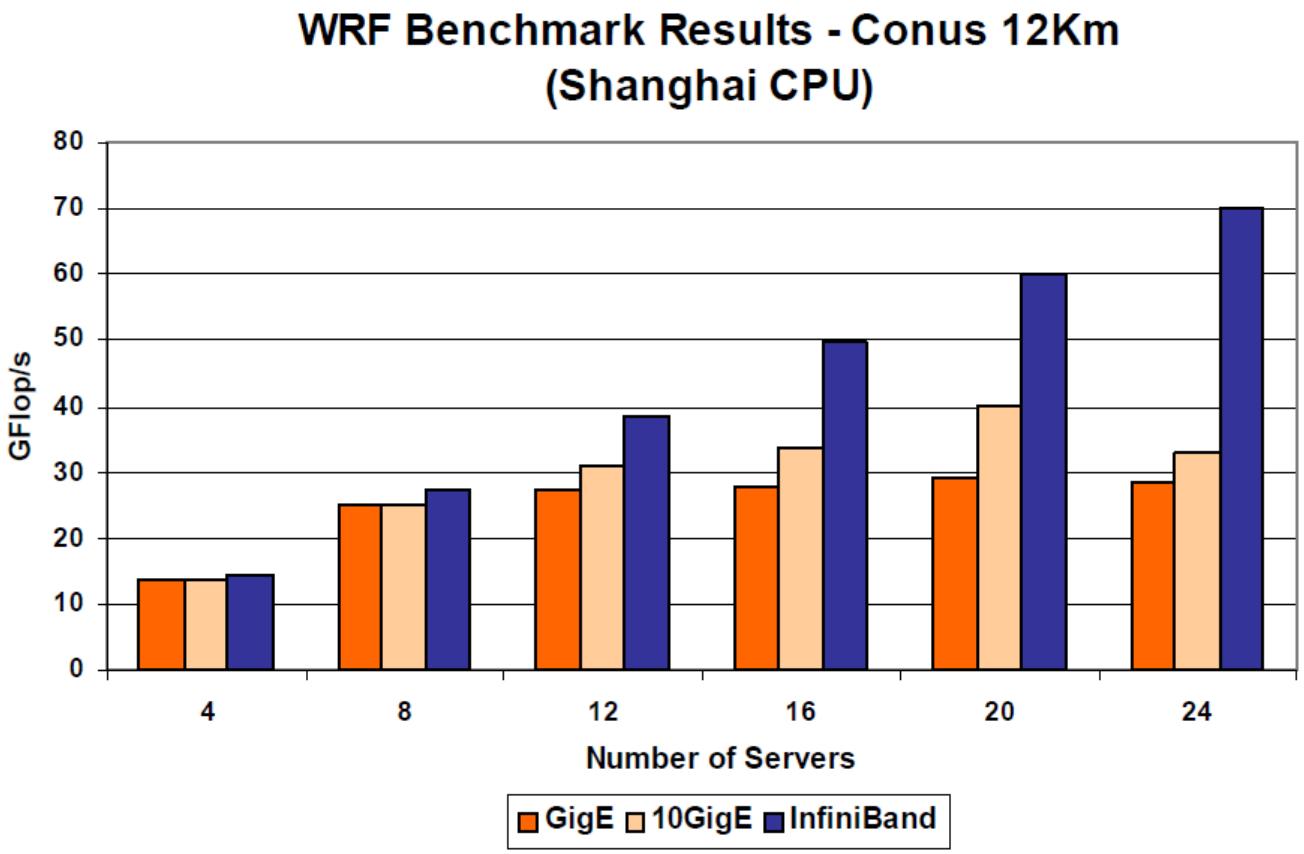
Benchmark: Weather Research Forecast (WRF)

- CONUS input data
- Peak at 256B and 64kB
  - For P=64 about 30% of all messages are smaller than 512B
- Communication dominated by MPI_Bcast (mostly 4B)



Benchmark: Weather Research Forecast (WRF)

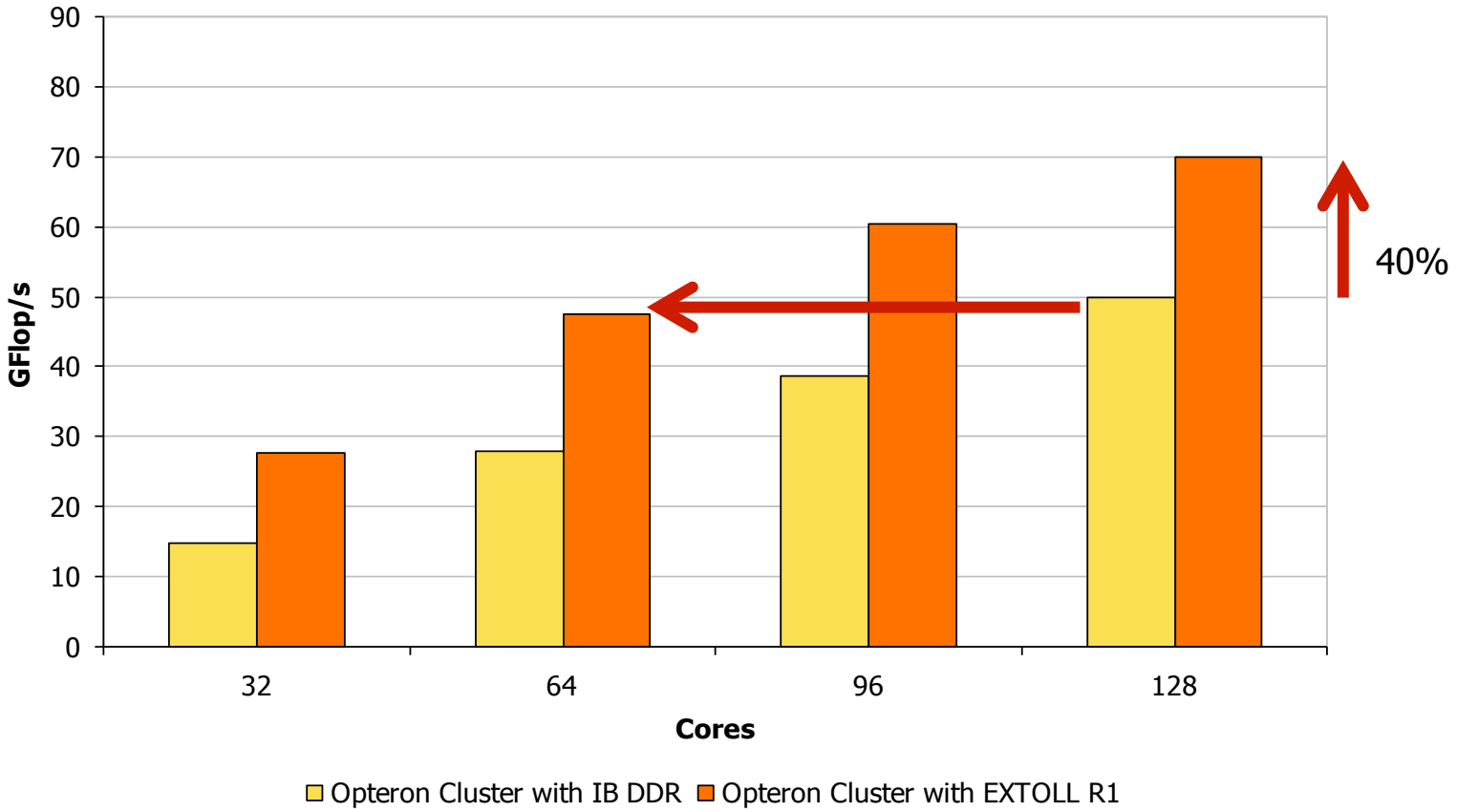
- Infiniband DDR
  - 16/20Gbps
- Gigabit Ethernet
  - 1Gbps
- 10 GE
  - 10Gbps
- AMD Shanghai
  - 2 CPUs/node
  - 8 cores/node



Courtesy: HPC Advisory Council

Benchmark: Weather Research Forecast (WRF)

- Infiniband DDR vs. EXTOLL R1

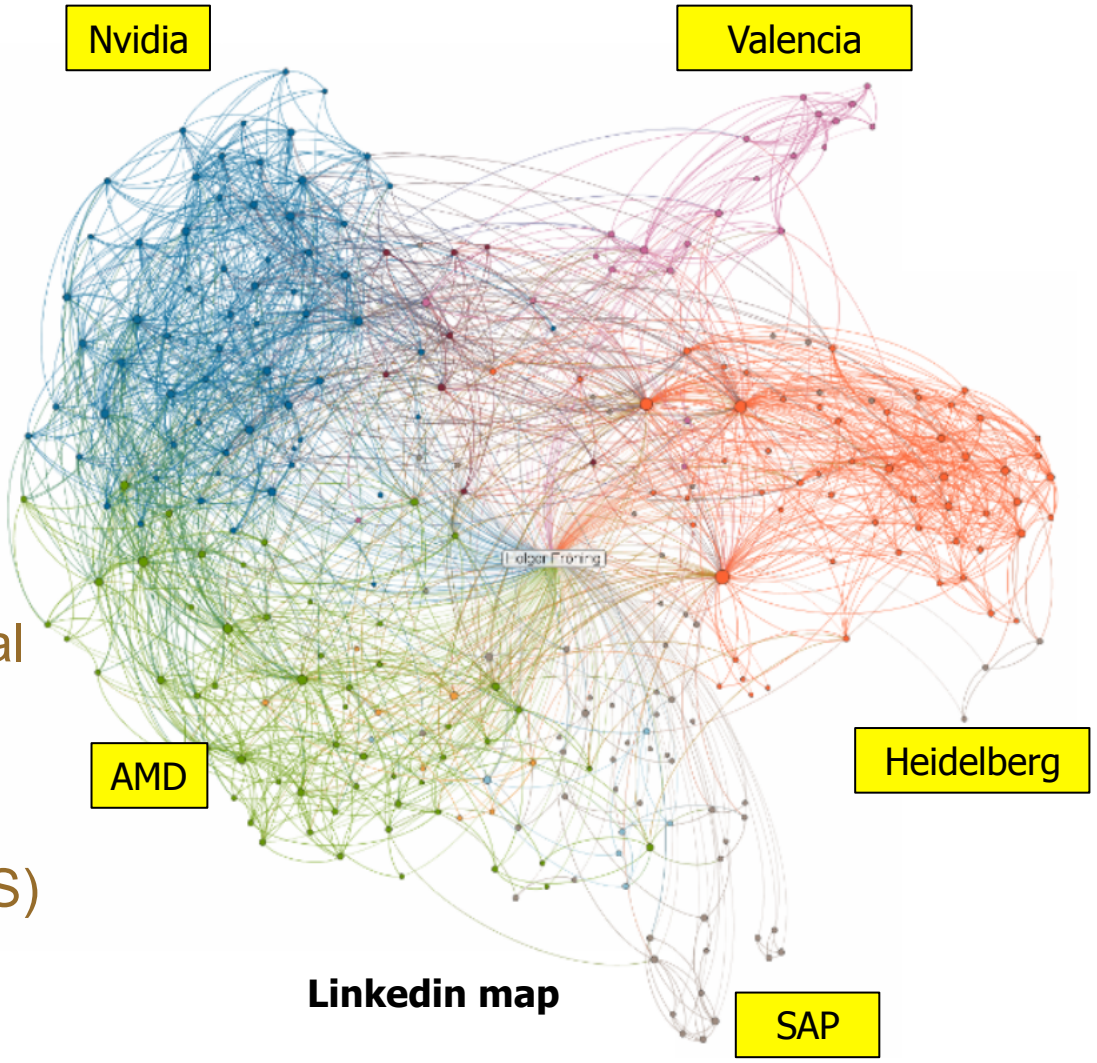


IB Cluster: Opteron Shanghai 2358 2.6GHz

EXTOLL Cluster: Opteron Shanghai 2380 2.5GHz

Graph Computations

- Graphs: natural representation of unstructured data
  - Biology, transportation, internet & power grids, communication data, social media
  - Relational data in general
- Search most common operation
  - Breadth-first search (BFS)



Linkedin map

GRAPH500

- Breadth-first search (BFS)
  - Large data sets
  - Irregular access patterns
  - Little reuse and spatial locality
  - Low computational intensity
  - Pointer-chasing requires heavy overlap
    - Prefetching almost useless
- GRAPH500 benchmark
  - Graph data generation (un-timed)
  - Data structure creation (timed)
  - BFS iterations (timed)
  - Result validation (un-timed)

November 2013

No.	Rank	Machine	Installation Site	Number of nodes	Number of cores	Problem scale	GTEPS
1	1	DOE/NNSA/LLNL Sequoia (IBM - BlueGene/Q, Power BQC 16C 1.60 GHz)	Lawrence Livermore National Laboratory	65536	1048576	40	15363
2	2	DOE/SC/Argonne National Laboratory Mira (IBM - BlueGene/Q, Power BQC 16C 1.60 GHz)	Argonne National Laboratory	49152	786432	40	14328
3	3	JUQUEEN (IBM - BlueGene/Q, Power BQC 16C 1.60 GHz)	Forschungszentrum Juelich (FZJ)	16384	262144	38	5848
4	4	K computer (Fujitsu - Custom supercomputer)	RIKEN Advanced Institute for Computational Science (AICS)	65536	524288	40	5524.12
5	5	Fermi (IBM - BlueGene/Q, Power BQC 16C 1.60 GHz)	CINECA	8192	131072	37	2567
6	6	Tianhe-2 (MilkyWay-2) (National University of Defense Technology - MPP)	Changsha, China	8192	196608	36	2061.48
7	7	Turing (IBM - BlueGene/Q, Power BQC 16C 1.60GHz)	CNRS/IDRIS-GENCI	4096	65536	36	1427
8	7	Blue Joule (IBM - BlueGene/Q, Power BQC 16C 1.60 GHz)	Science and Technology Facilities Council - Daresbury Laboratory	4096	65536	36	1427

http://www.graph500.org

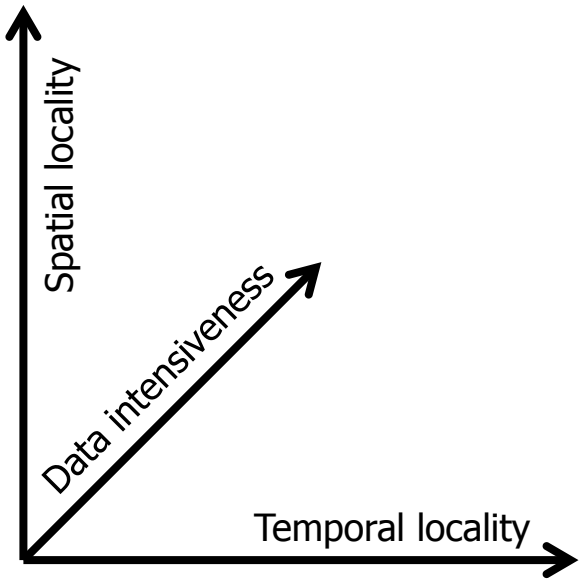
Summary

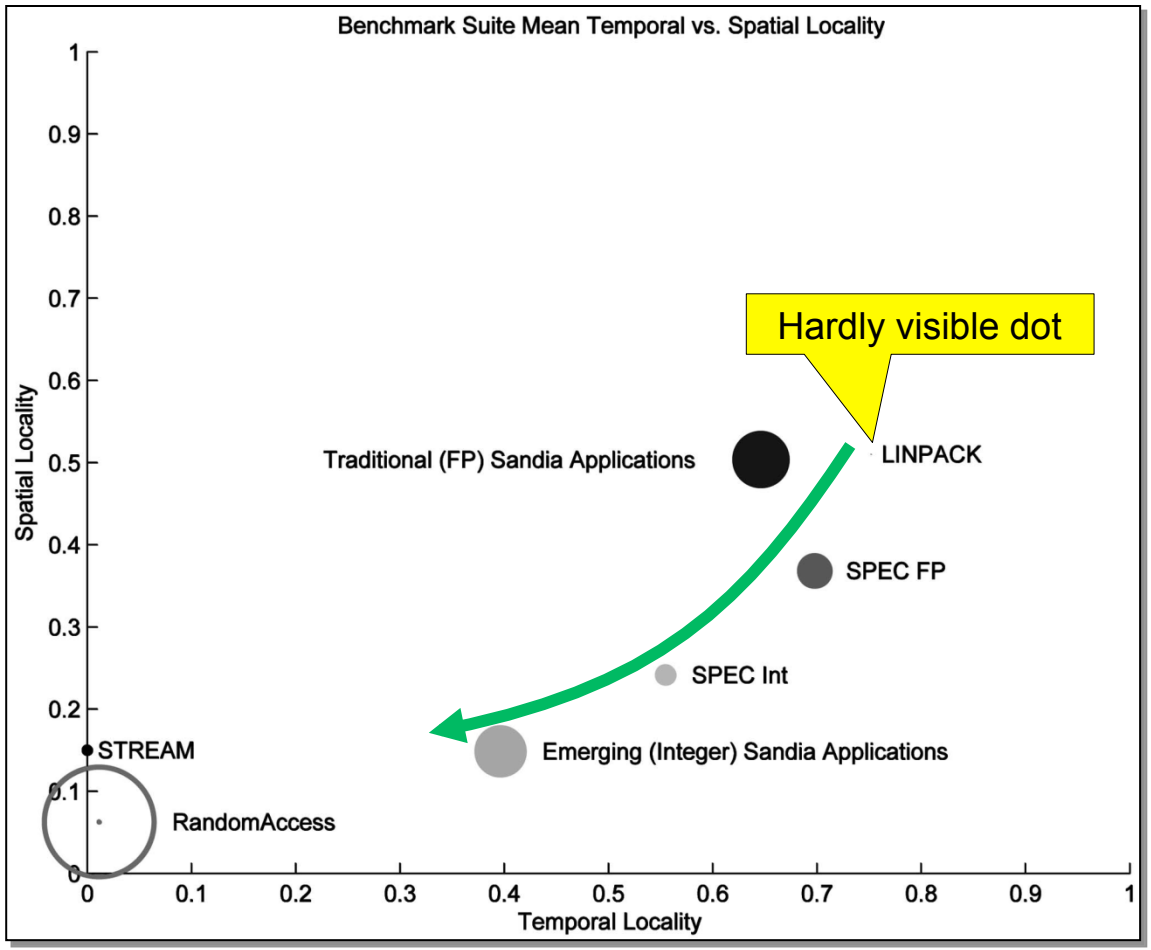
- Benchmarking is complex…
- The best choice always depends on the application
  - MPI time is a nice measure for a first impression, but highly depends on IN/ implementation
- Micro-benchmarks help to classify systems, but they cannot substitute real assessment
- Application characterization: payload size, communication pattern, …
  - Differences between messaging layer and functional layer
- Big Data?

Test	MPI Time
NPB-EP	0%
NPB-MG	9%
NPB-BT	10%
NAMD	24%
HPL	25%
NPB-CG	34%
NPB-FT	37%
NPB-IS	47%
WRF	45%
MPIFFT	77%
RandomAccess	89%

Future – Big Data

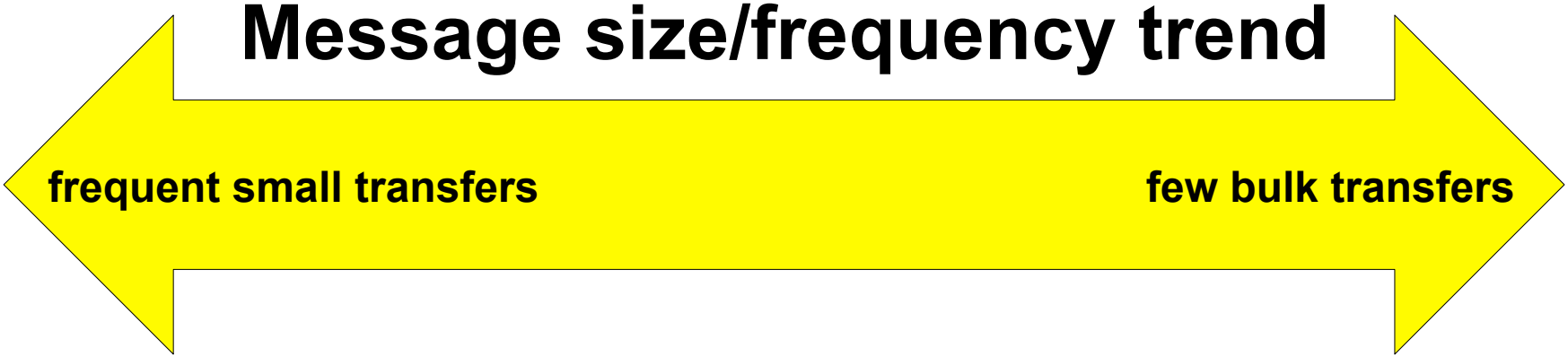
- Data intensiveness = number of unique bytes accessed
  - Size of the circles





Richard C. Murphy and Peter M. Kogge. 2007. On the Memory Access Patterns of Supercomputer Applications: Benchmark Selection and Its Implications. *IEEE Trans. Comput.* 56, 7 (July 2007), 937-945.

Future – Fine-grain Messaging



- Parallelism degree dramatically increasing
  - Data distribution
  - Synchronization
- Trend towards graph algorithms
  - Complex, unstructured patterns

- Optimization of MPI programms
  - Need to rely on bulk transfers to expose overhead
  - Not native to the problem
- Big Data era
  - Massive data set sizes