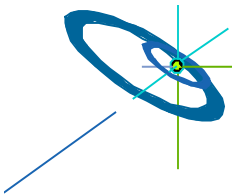
Balance of HPC Systems Based on HPCC Benchmark Results

Rolf Rabenseifner
rabenseifner@hlrs.de

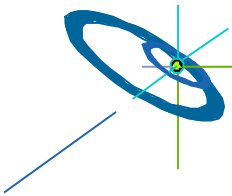
University of Stuttgart
High-Performance Computing-Center Stuttgart (HLRS)
www.hlrs.de

CUG 2005
Albuquerque, NM, May 16-19, 2005
(Revised with HPCC data status June 27, 2005)

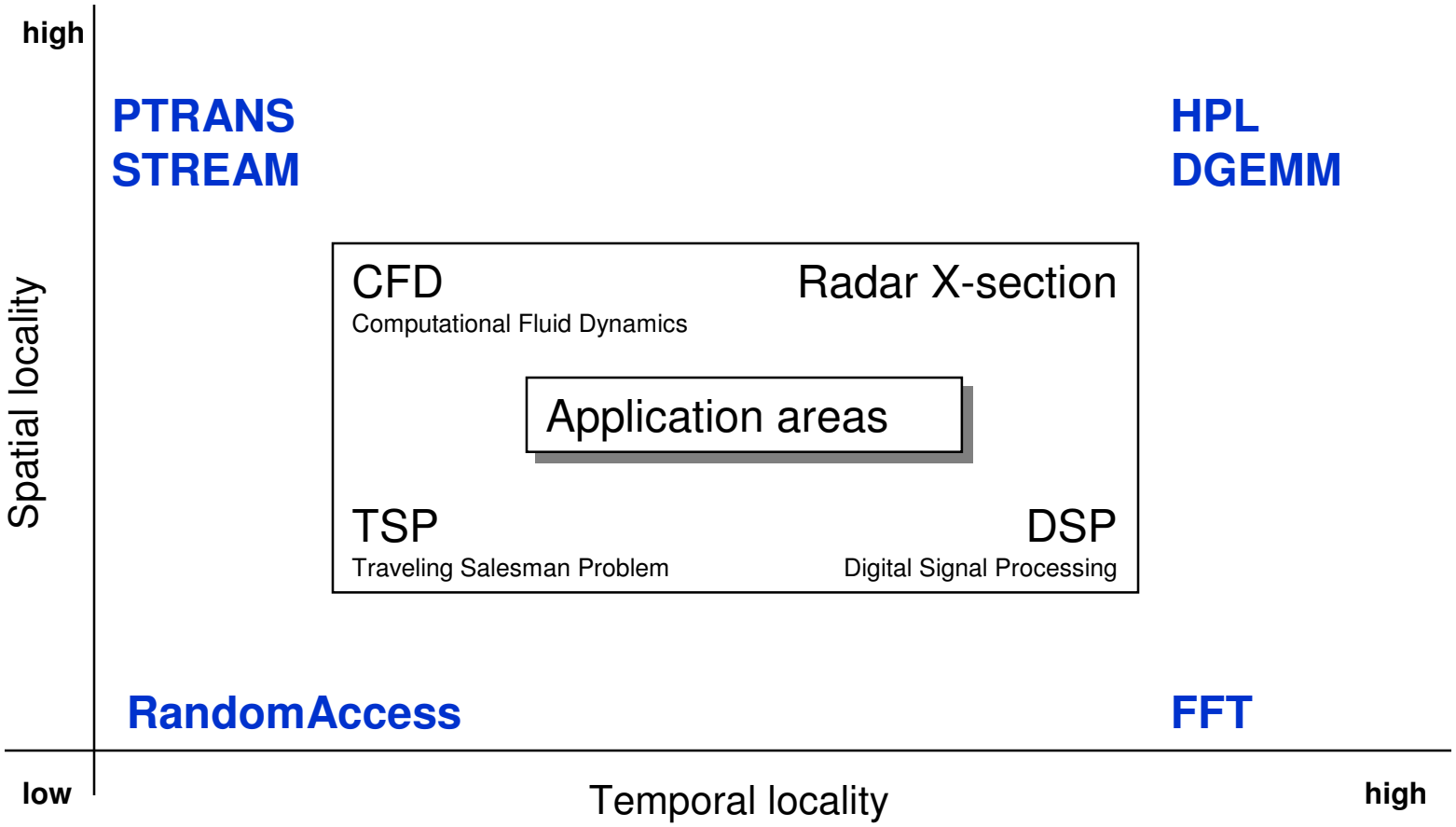


Balance Analysis with HPC Challenge Benchmark Data

- How HPCC data can be used to analyze the balance of HPC systems
  - **Details on ring based benchmarks**
- Resource based ratios
  - **Inter-node bandwidth and**
  - **memory bandwidth**
  - **versus computational speed**
- HPCC footprint
  - **Comparing the platforms**



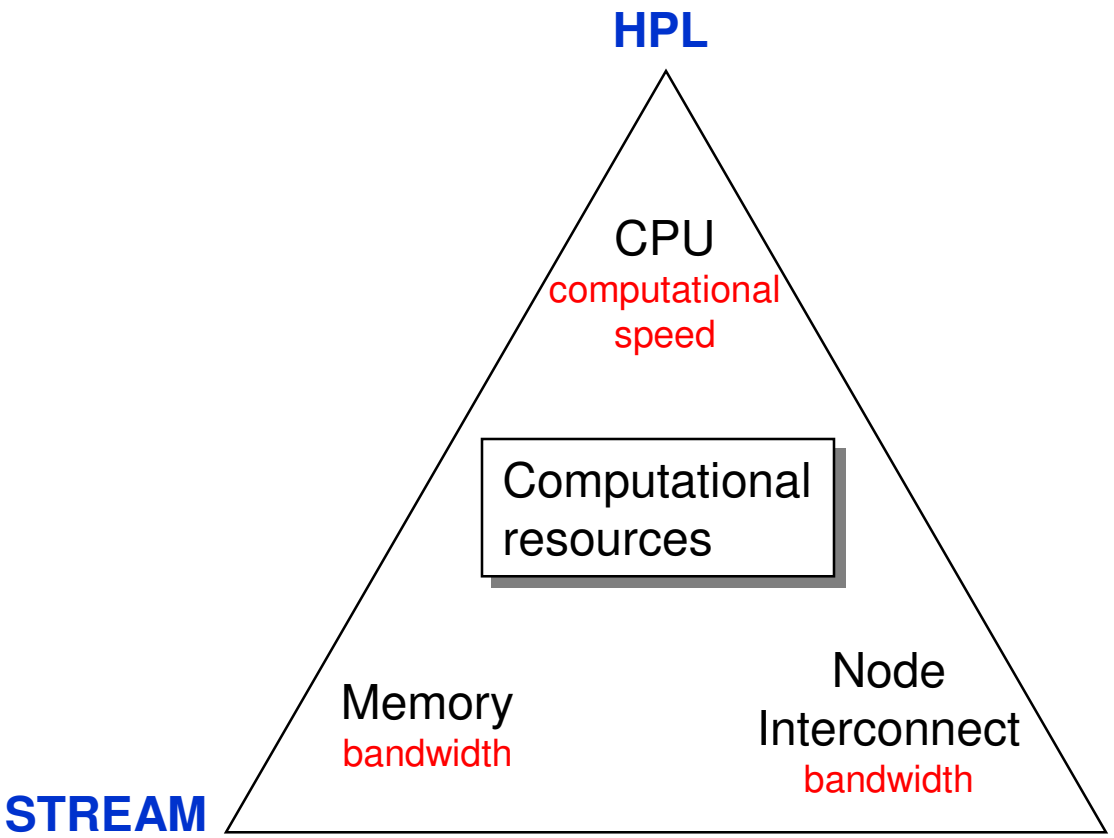
Application areas & HPC Challenge Benchmarks



H L R S

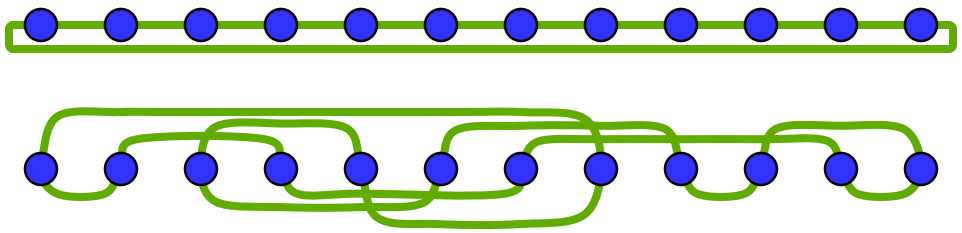
Courtesy of Jack J. Dongarra and Piotr Luszczek

Computational Resources & HPC Challenge Benchmarks



Random & natural ring bandwidth & latency

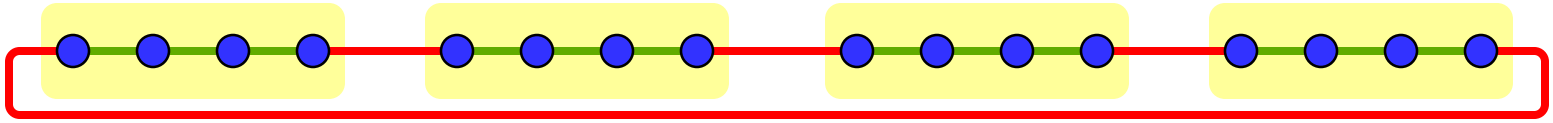
- Parallel communication pattern on all MPI processes (●)
  - Natural ring
  - Random ring



- Bandwidth per process
  - Accumulated message size / wall-clock time / number of processes
  - On each connection messages in both directions
  - With *2xMPI_Sendrecv* and *MPI non-blocking* → best result is used
  - Message size = 2,000,000 bytes
- Latency
  - Same patterns, message size = 8 bytes
  - Wall-clock time / (number of sendrecv per process)

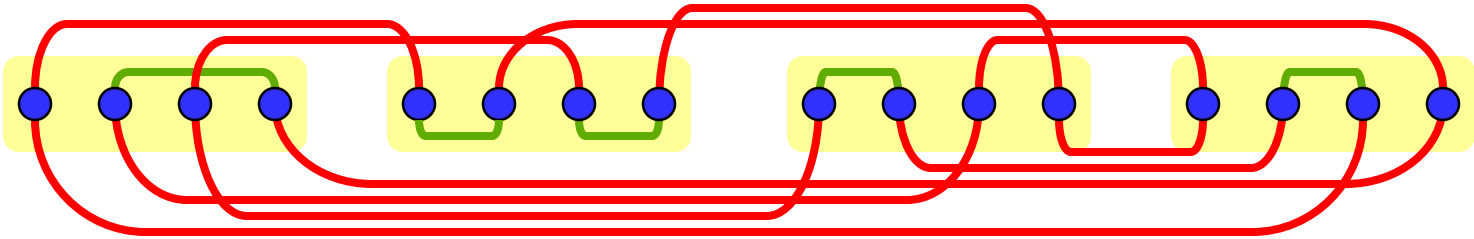
Inter-node bandwidth on clusters of SMP nodes

- Natural Ring:
  - Only one incoming and one outgoing message per node at the same time
  - Accumulated bandwidth
    := bandwidth per process x #processes
  - Does **not** reflect the accumulated inter-node bandwidth
    (nor the bi-section bandwidth)



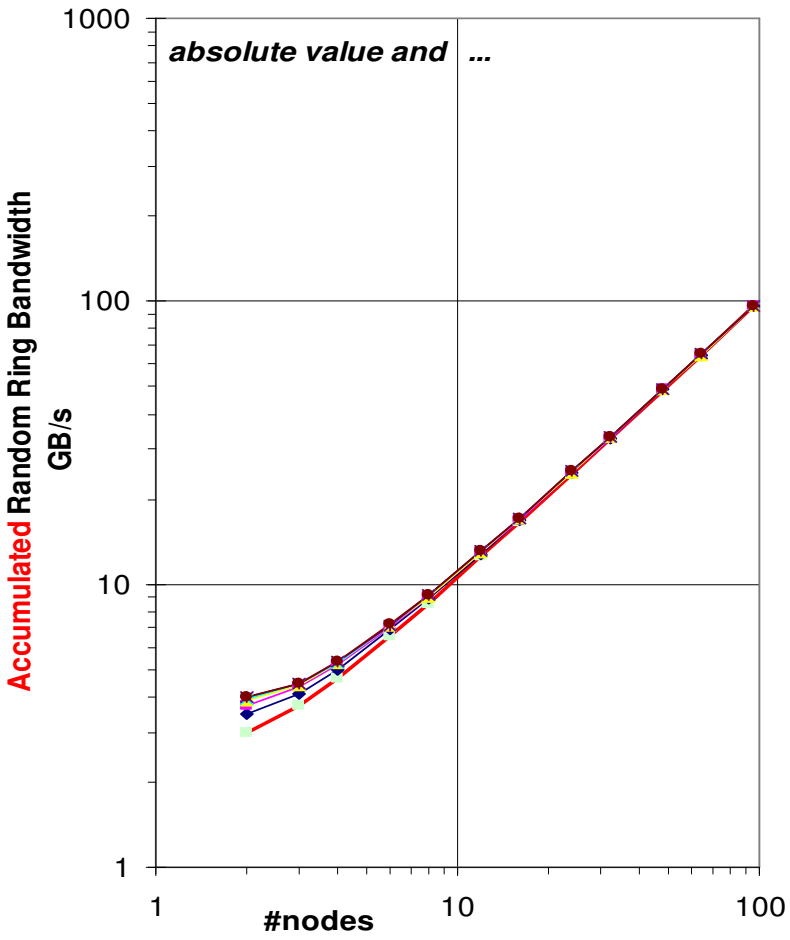
Inter-node bandwidth on clusters of SMP nodes

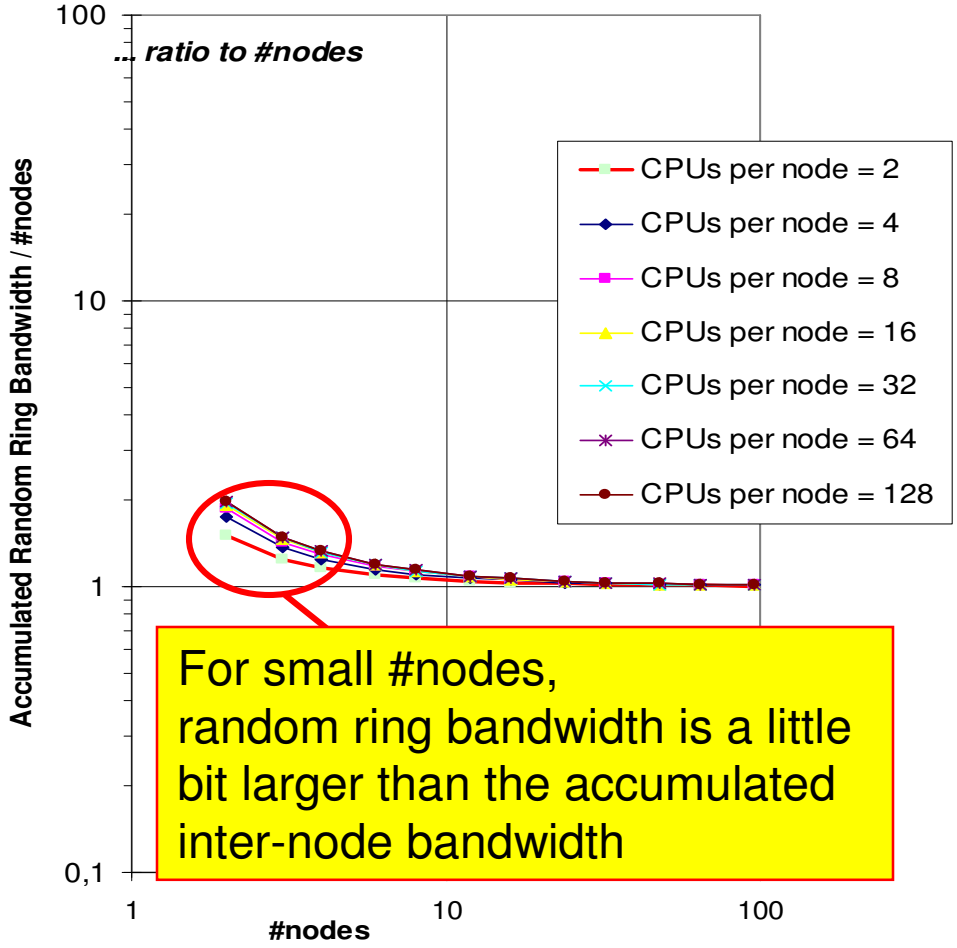
- Random Ring
  - Some connections are inside of the nodes
  - Most connections are inter-node
  - Depends on #nodes and #MPI processes per node



  - Accumulated bandwidth
    := bandwidth per process x #processes
  - ~= accumulated inter-node bandwidth x (1 – 1 / #nodes)$^{-1}$

On an ideally switched cluster …

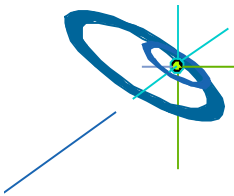




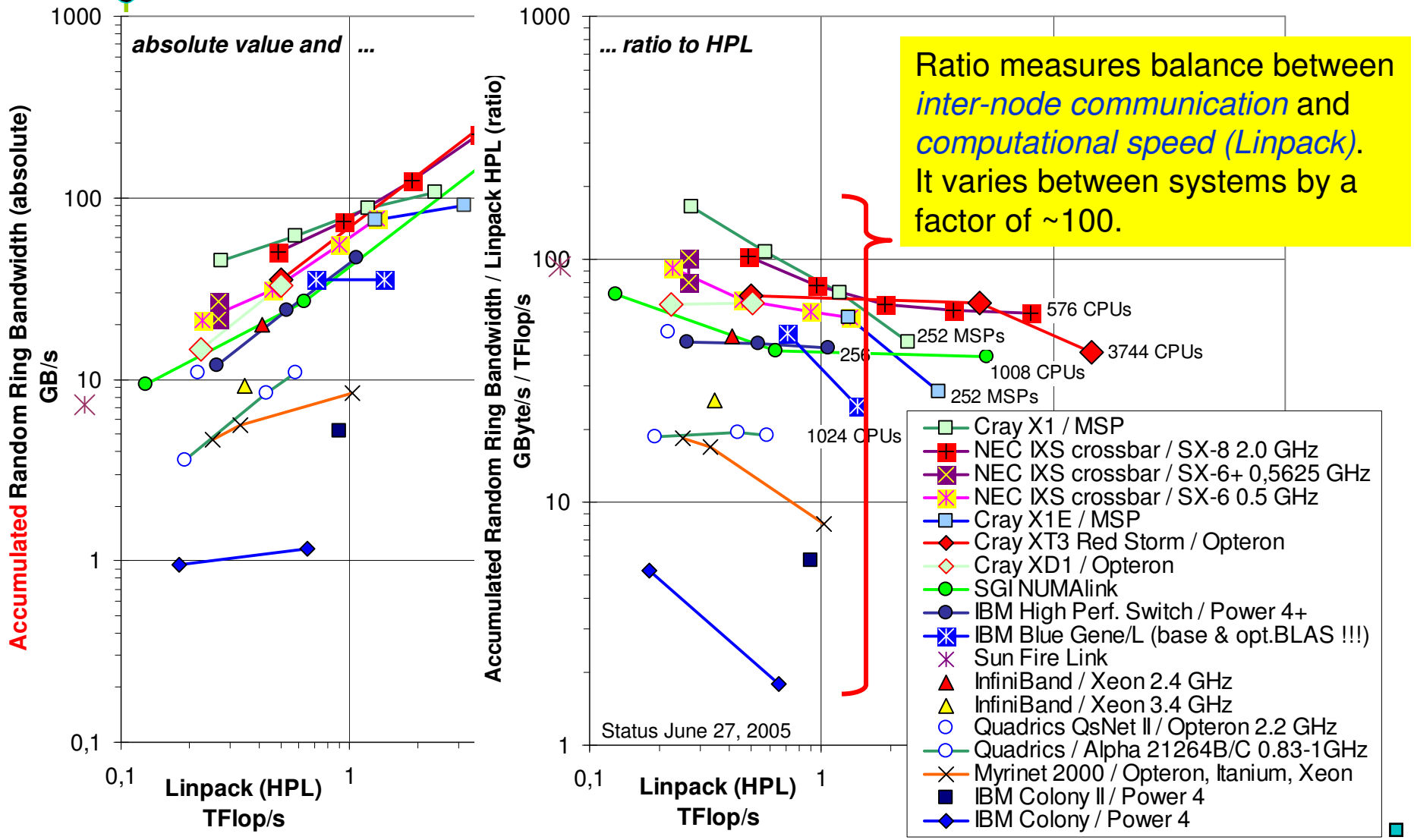
For small #nodes,
random ring bandwidth is a little bit larger than the accumulated inter-node bandwidth

Balance Analysis with HPC Challenge Benchmark Data

- Balance can be expressed as a set of ratios
  - e.g., accumulated memory bandwidth / accumulated Tflop/s rate
- Basis
  - **Linpack (HPL)** → **Computational Speed**
  - **Random Ring Bandwidth** → **Inter-node communication**
  - **Parallel STREAM Copy or Triad** → **Memory bandwidth**
- Be careful:
  - Some data are presented for the **total system**
  - Some per **MPI process** (HPL processes)
  - i.e., balance calculation always
    - **with accumulated data on total system, or**
    - **with divided data to one MPI process**



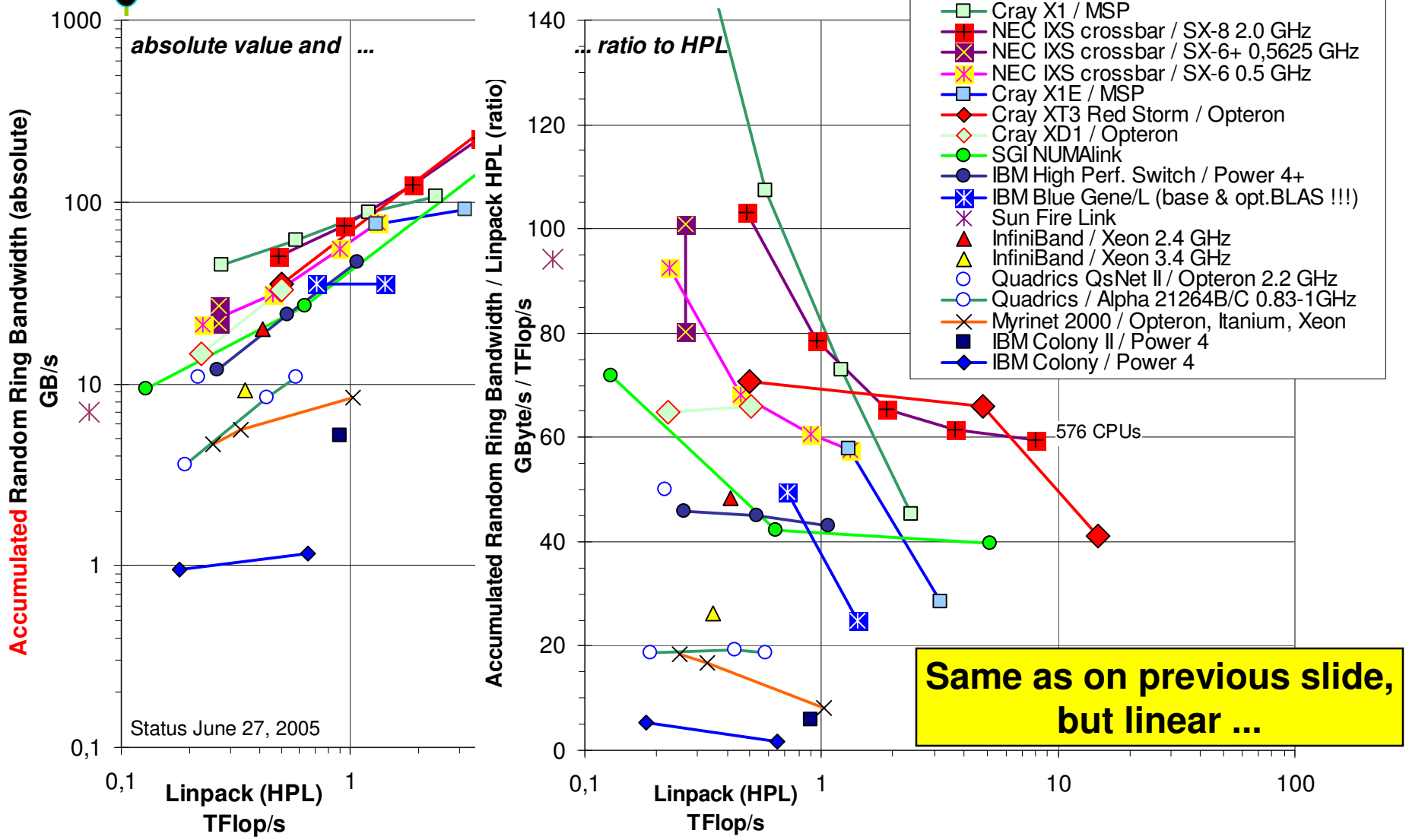
... and now Random Ring Bandwidth on real platforms



Ratio measures balance between *inter-node communication* and *computational speed (Linpack)*. It varies between systems by a factor of ~100.

Measurements with smaller #CPUs on XT3 and SX-8 courtesy to Nathan Wichmann, Cray, CUG 2005, and Sunil Tiyyagura, HLRS.

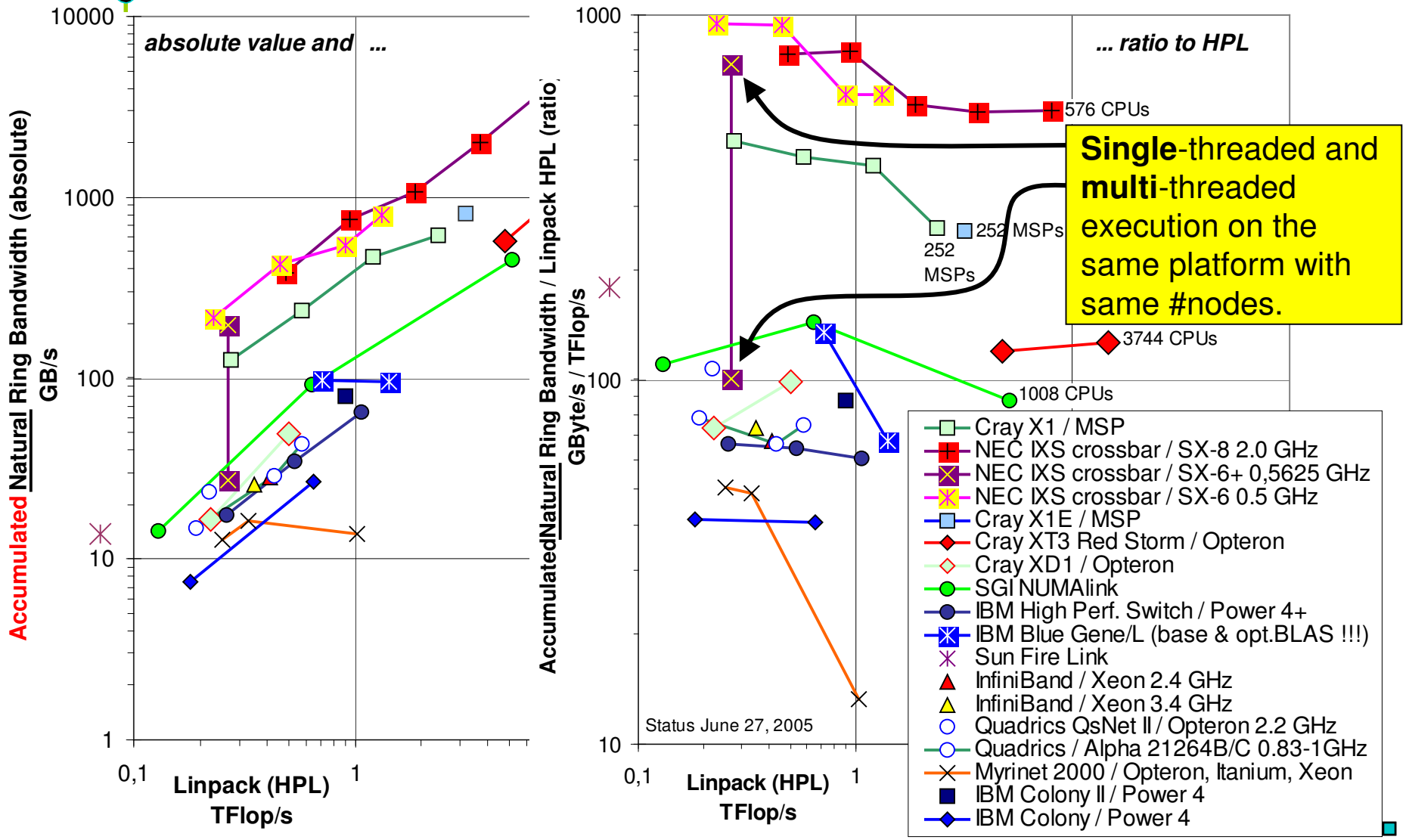
... and now Random Ring Bandwidth on real platforms



Measurements with smaller #CPUs on XT3 and SX-8 courtesy to Nathan Wichmann, Cray, CUG 2005, and Sunil Tiyyagura, HLRS.

skipped

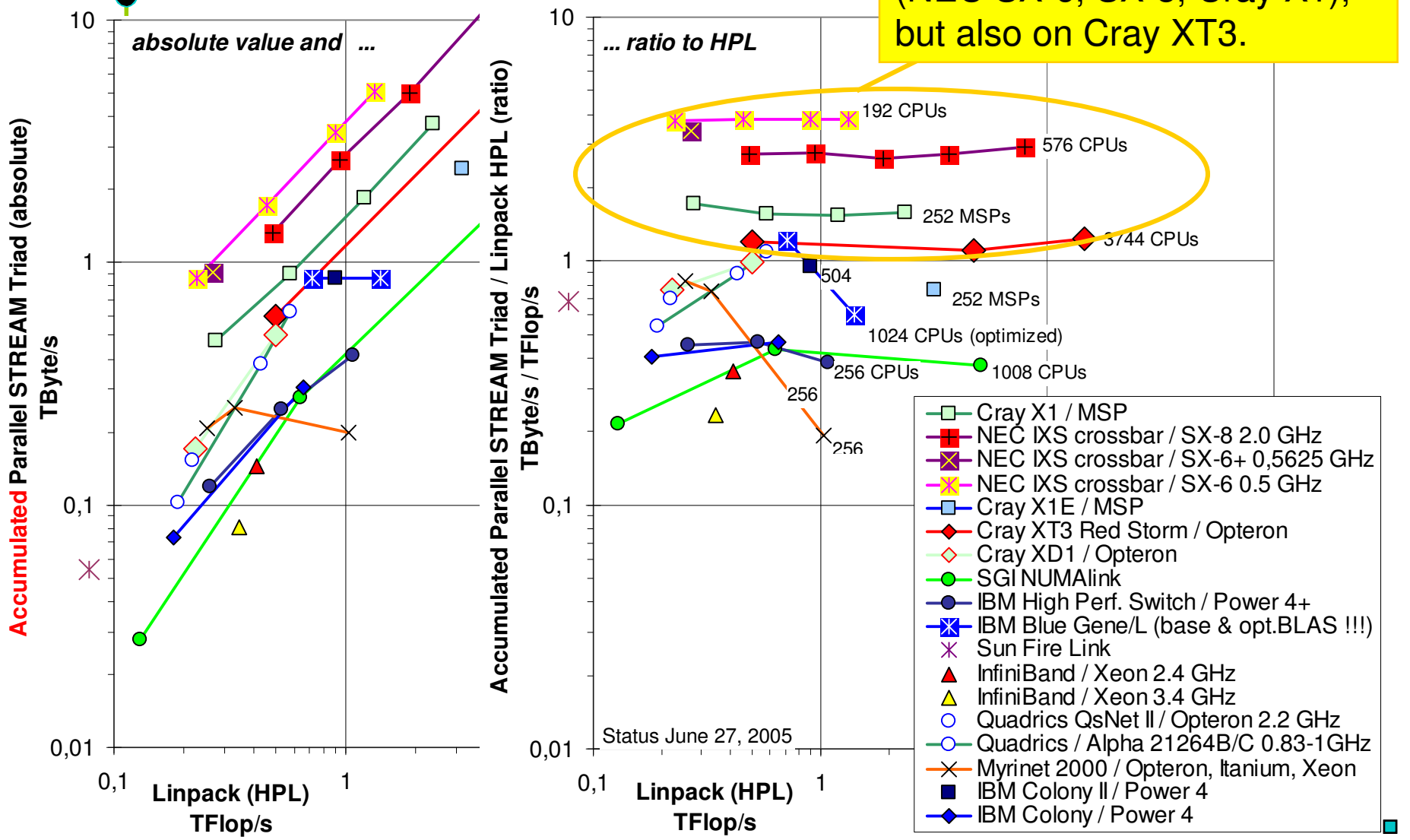
Natural Ring Bandwidth



Measurements with smaller #CPUs on XT3 and SX-8 courtesy to Nathan Wichmann, Cray, CUG 2005, and Sunil Tiyyagura, HLRS.

Balance between memory and CPU

High memory bandwidth ratio on vector-type systems (NEC SX-6, SX-8, Cray X1), but also on Cray XT3.

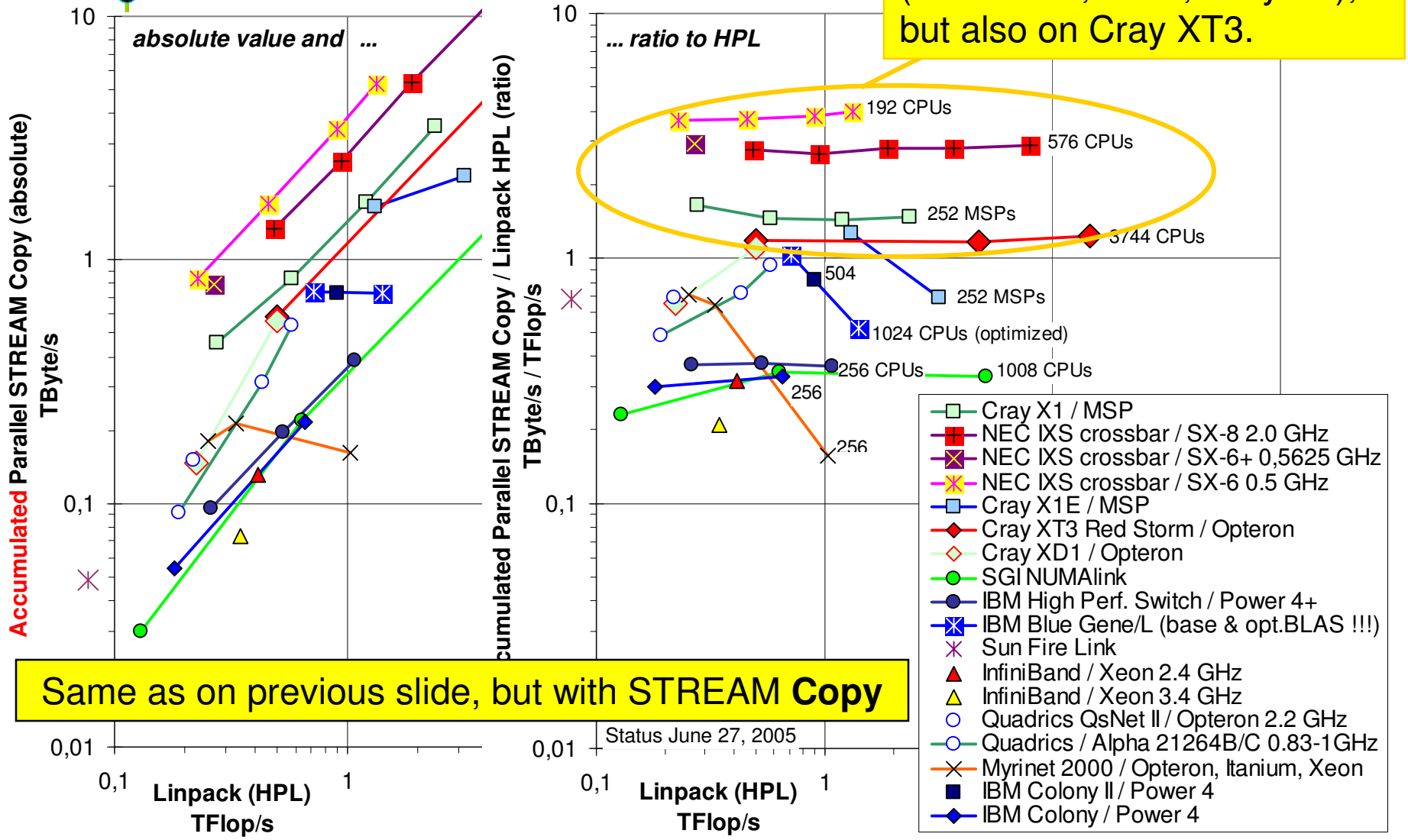


Measurements with smaller #CPUs on XT3 and SX-8 courtesy to Nathan Wichmann, Cray, CUG 2005, and Sunil Tiyyagura, HLRS.

skipped

Balance between memory and CPU

High memory bandwidth ratio on vector-type systems (NEC SX-6, SX-8, Cray X1), but also on Cray XT3.



Same as on previous slide, but with STREAM **Copy**

Measurements with smaller #CPUs on XT3 and SX-8 courtesy to Nathan Wichmann, Cray, CUG 2005, and Sunil Tiyyagura, HLRS.

HPCC footprints

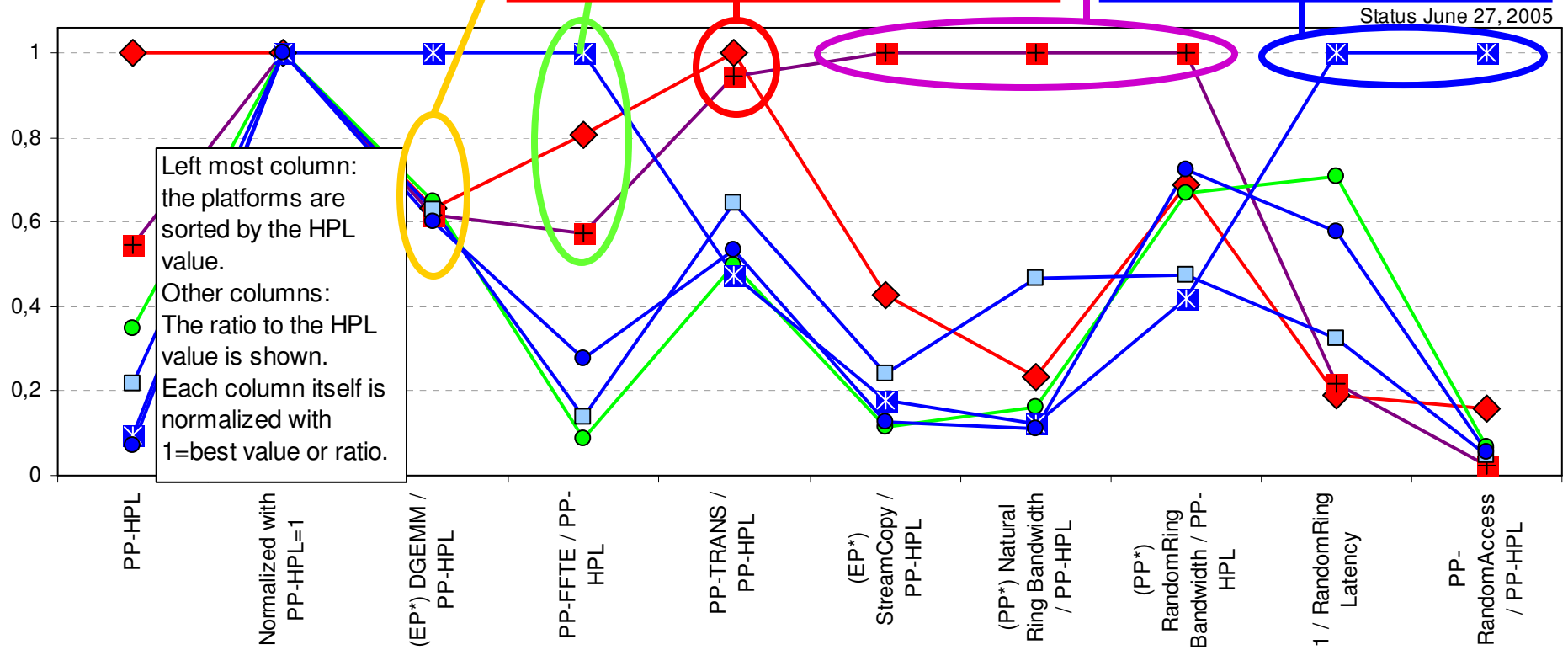
DGEMM: Similar results on all platforms, except BlueGene

FFTE: IBM **BlueGene**, Cray **XT3** and NEC **SX-8** best

Extremely **fast memory** and **interconnect** on **NEC SX-8**

Best **PTRANS** on **XT3** and **SX-8**

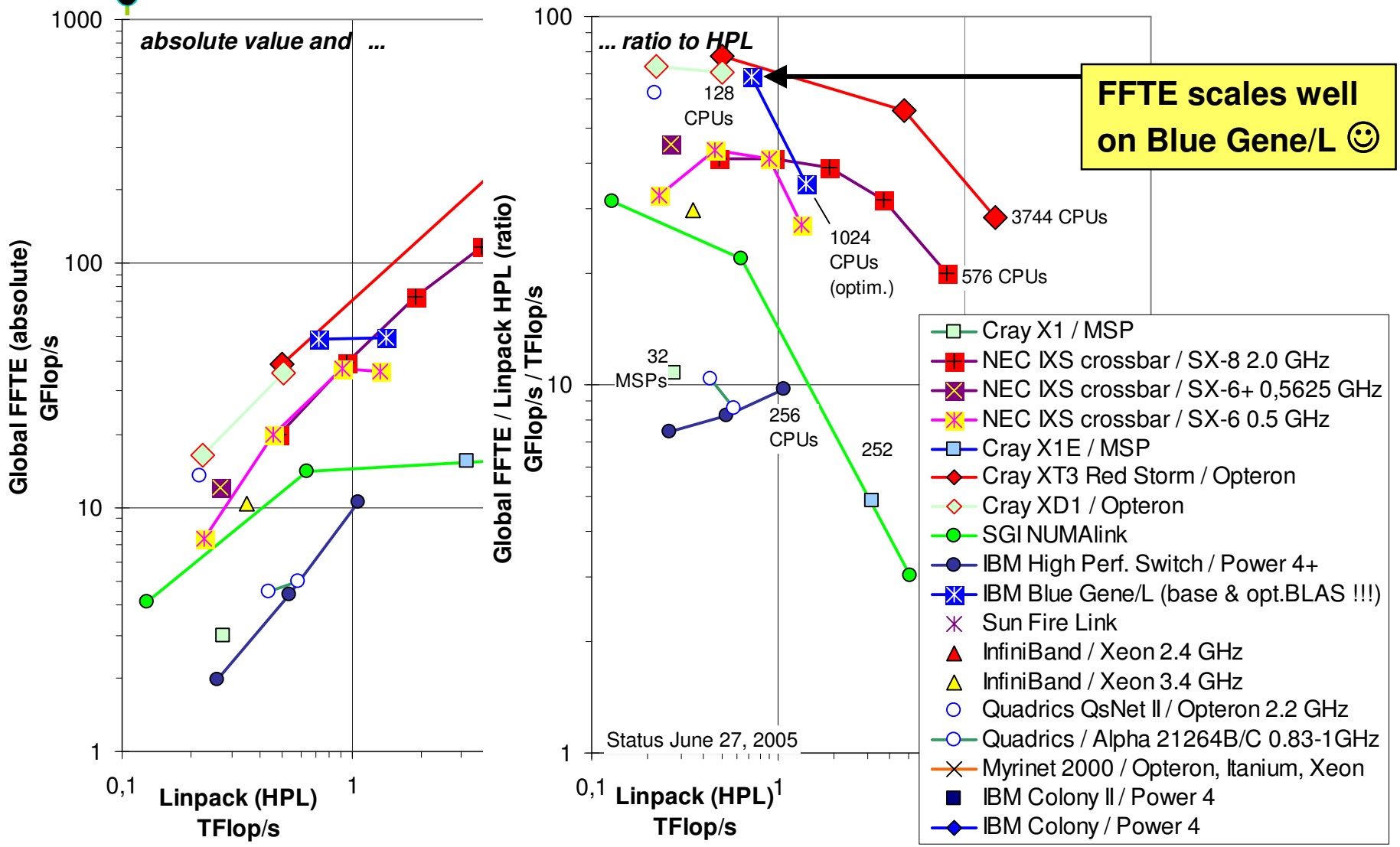
BlueGene:Super **latency**



PP = total system performance; PP* = PP calculated from per-process-value; EP* = embarrassingly parallel value per process is multiplied by the number of processes

- 14,7 TFlop/s HPL - Cray - XT3 - AMD Opteron - 2,4 GHz - 3744x1 (MPI processes x threads) - Cray XT3 MPP Interconnect - 2005-06-21 *)
- 8,01 TFlop/s HPL - NEC - SX-8 - SX-8 - 2 GHz - 576x1 (MPI processes x threads) - IXS crossbar - 2005-06-21
- 5,14 TFlop/s HPL - SGI - Altix 3700 Bx2 - Intel Itanium 2 - 1,6 GHz - 1008x1 (MPI processes x threads) - NUMAlink - 2005-03-29
- 3,19 TFlop/s HPL - Cray - X1E - CrayX1E MSP - 1,13 GHz - 252x1 (MPI processes x threads) - Cray modified 2D torus - 2005-06-16
- 1,42 TFlop/s HPL - IBM - Blue Gene/L - IBM PowerPC 440 - 0,7 GHz - 1024x1 (MPI processes x threads) - Custom - 2005-04-11 (opt. BLAS!!!)
- 1,07 TFlop/s HPL - IBM - eServer pSeries 655 - IBM Power 4+ - 1,7 GHz - 64x4 (MPI processes x threads) - High Perf.Switch HPS - 2004-08-26

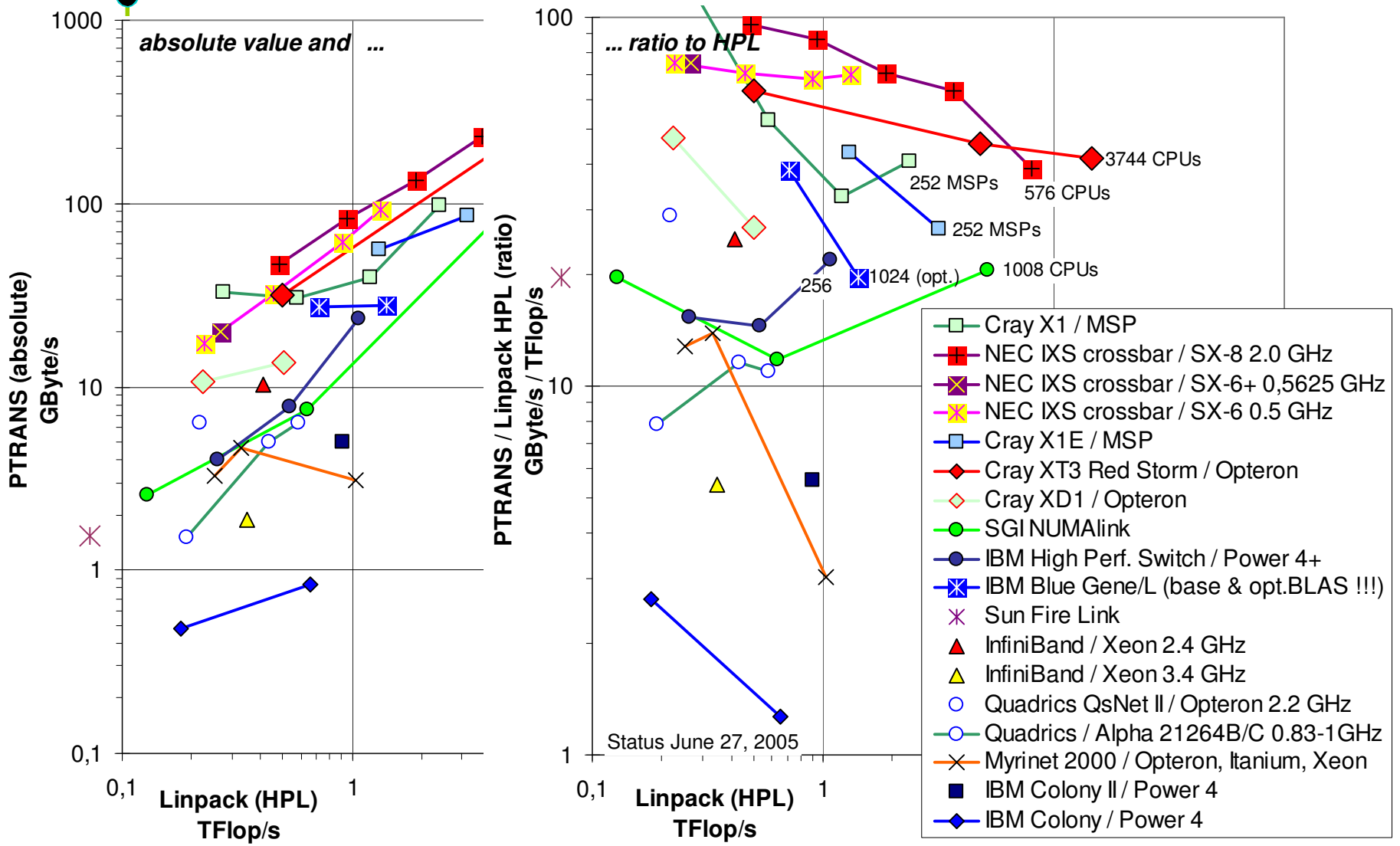
FFTE – Fast Fourier Transform (measured only on a only a few systems)



Measurements with smaller #CPUs on XT3 and SX-8 courtesy to Nathan Wichmann, Cray, CUG 2005, and Sunil Tiyyagura, HLRS.

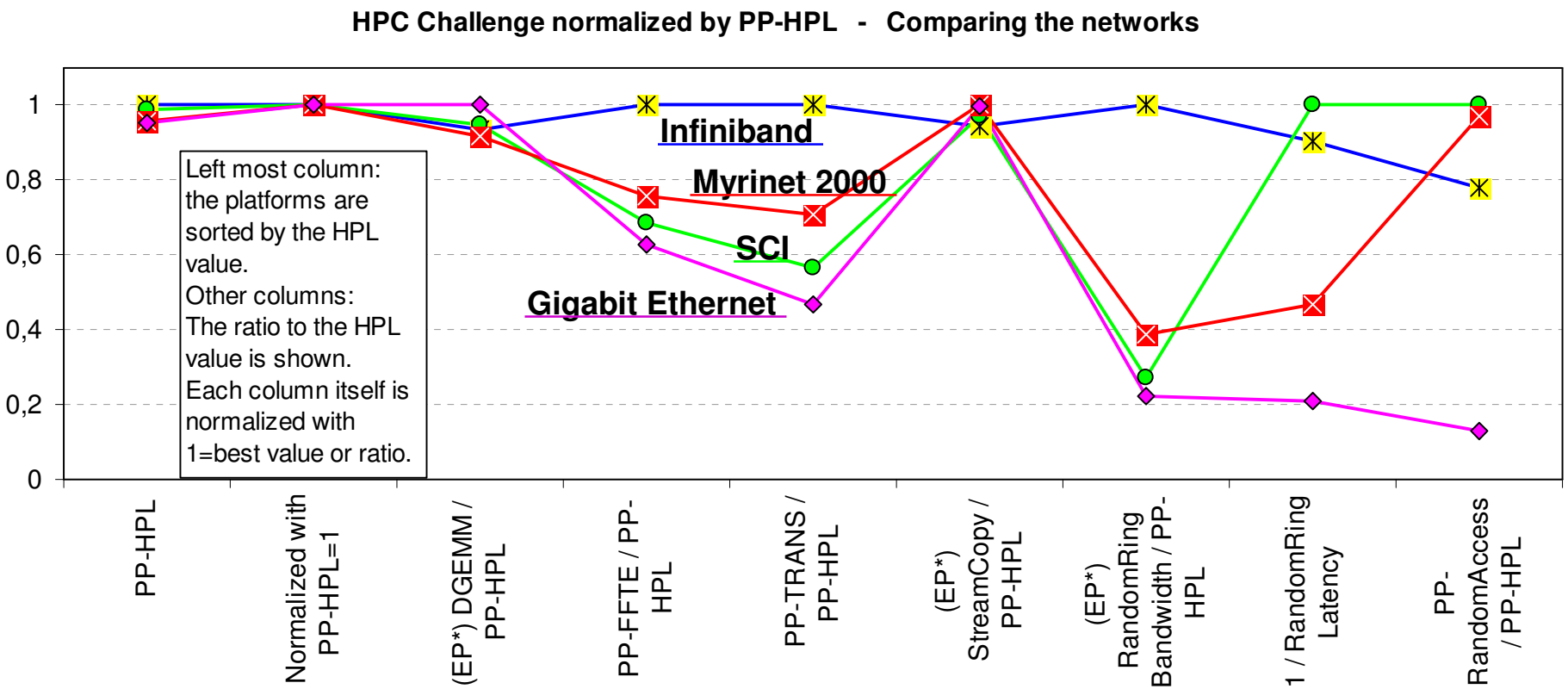
skipped

PTRANS – Matrix Transpose (measured only on a only a few systems)



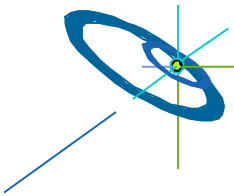
Measurements with smaller #CPUs on XT3 and SX-8 courtesy to Nathan Wichmann, Cray, CUG 2005, and Sunil Tiyyagura, HLRS.

Four different networks on the same system



Acknowledgments

- Thanks to
  - all persons and institutions that have uploaded HPCC results.
  - Jack Dongarra and Piotr Luszczek
    for inviting me into the HPCC development team.
  - Matthias Müller, Sunil Tiyyagura and Holger Berger
    for benchmarking on the SX-8 and SX-6 and discussions on HPCC.
  - Nathan Wichmann from Cray for Cray XT3 and X1E data.
  - David Koester
    for his helpful remarks on the HPCC Kiviat diagrams.



Conclusions

- HPCC is an interesting basis for
  - benchmarking computational resources
  - analyzing the balance of a system
  - scaling with the number of processors
  - with respect to application needs

- HPCC helps to show the strength of
  - vector systems
  - cluster networks