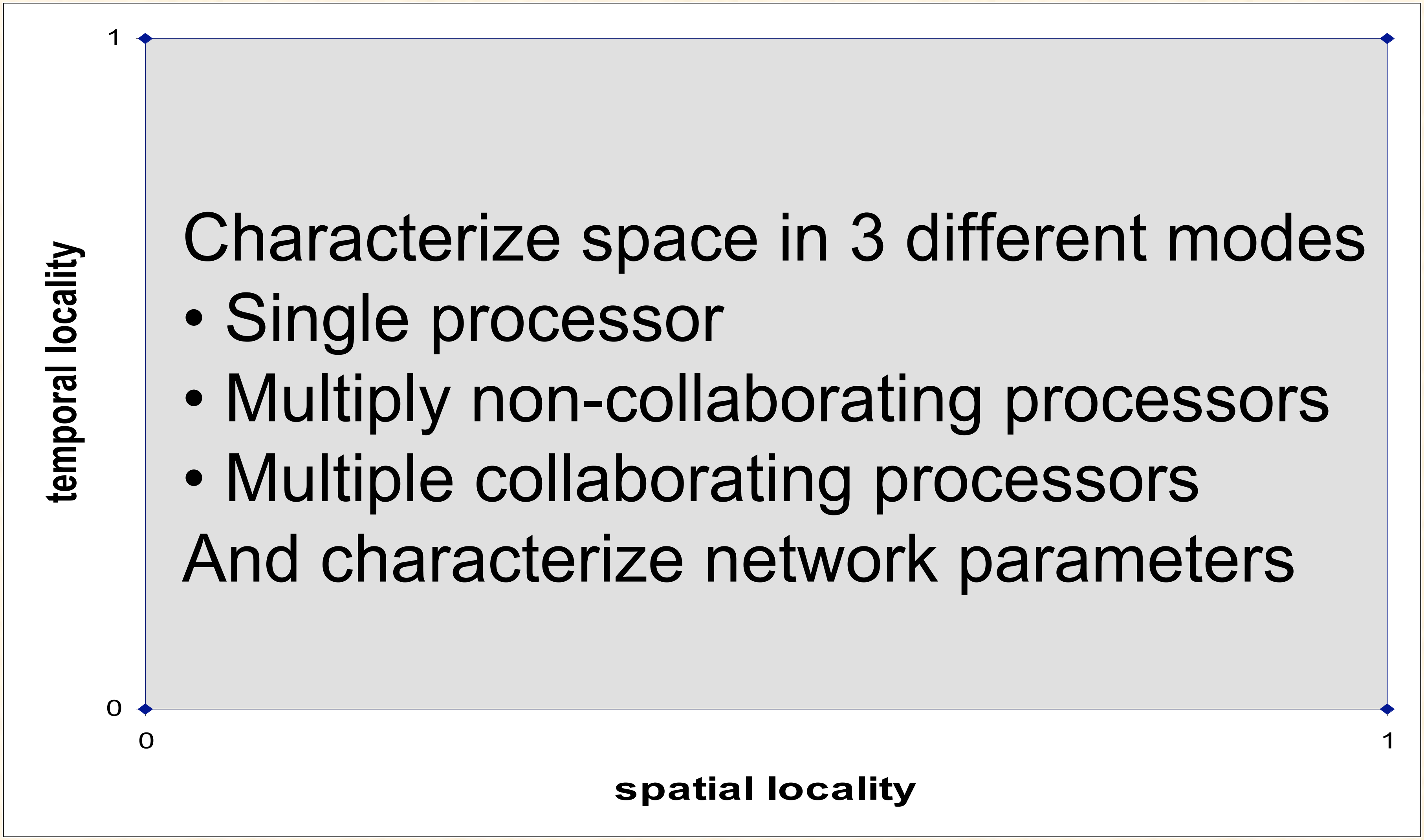
HPCC Update and Analysis

May 2006

Jeffery A. Kuehn, ORNL

Nathan L. Wichmann, Cray, Inc

HPCC Overview

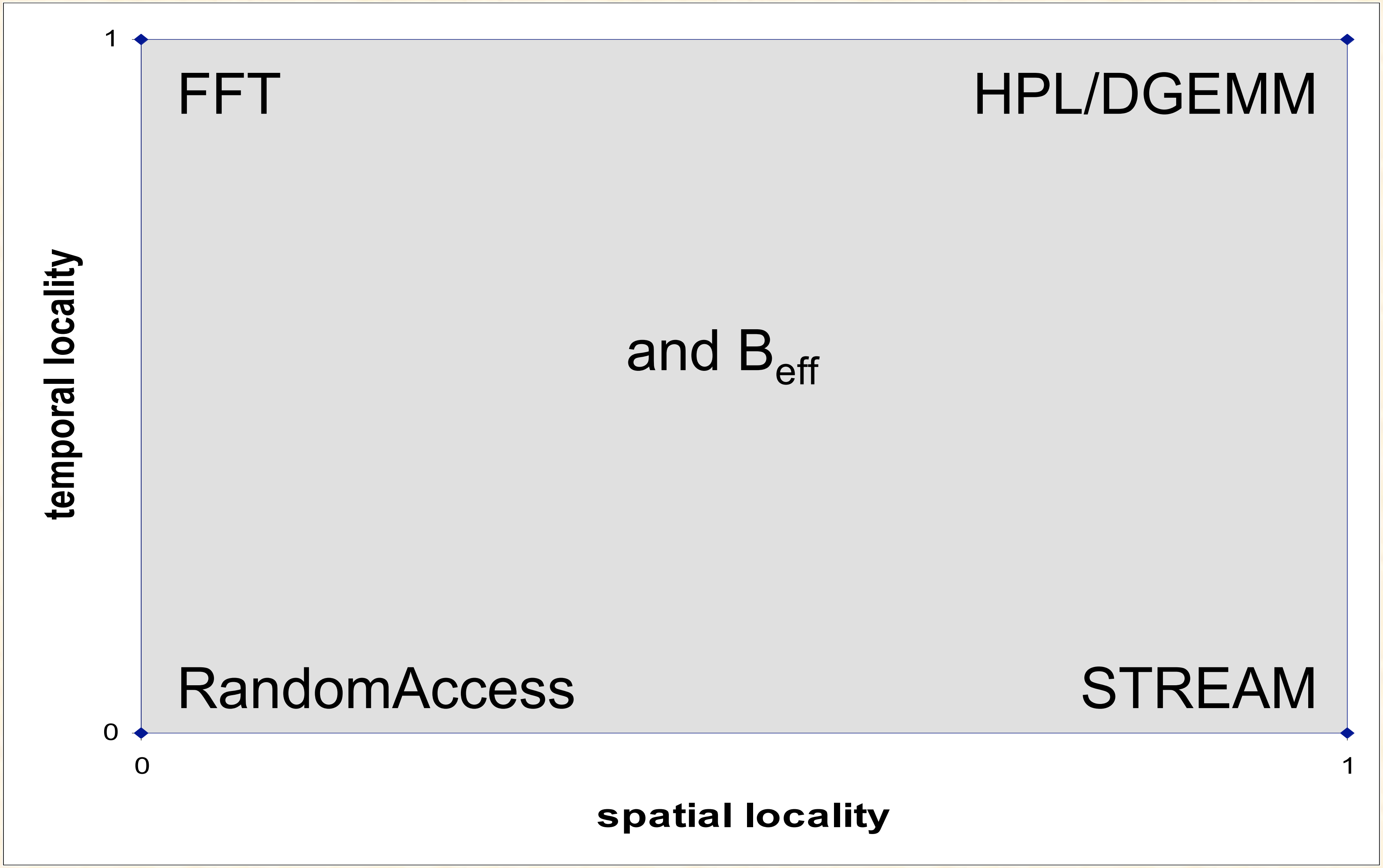


Characterize space in 3 different modes

- Single processor
- Multiply non-collaborating processors
- Multiple collaborating processors

And characterize network parameters

HPCC & Spatial-Temporal Cache Locality

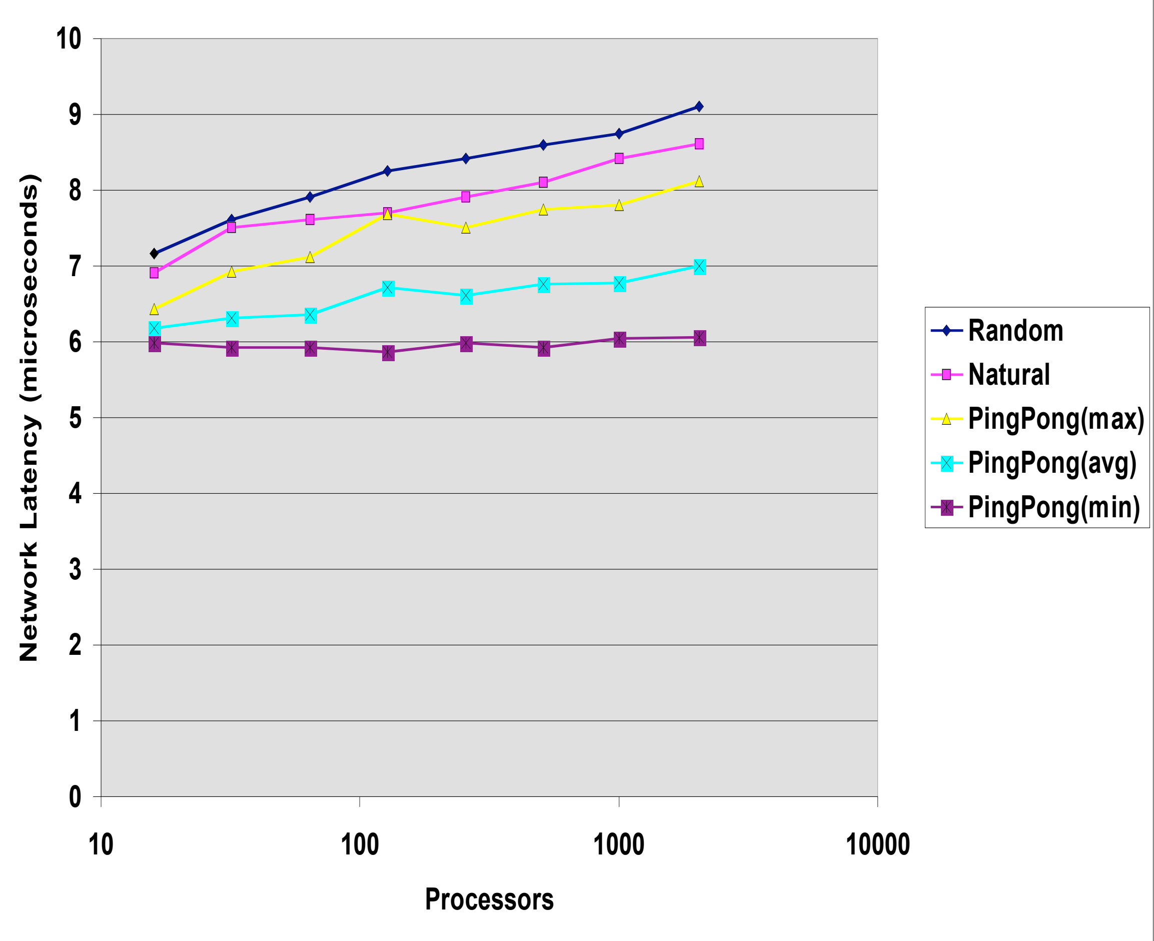


Understanding the Results

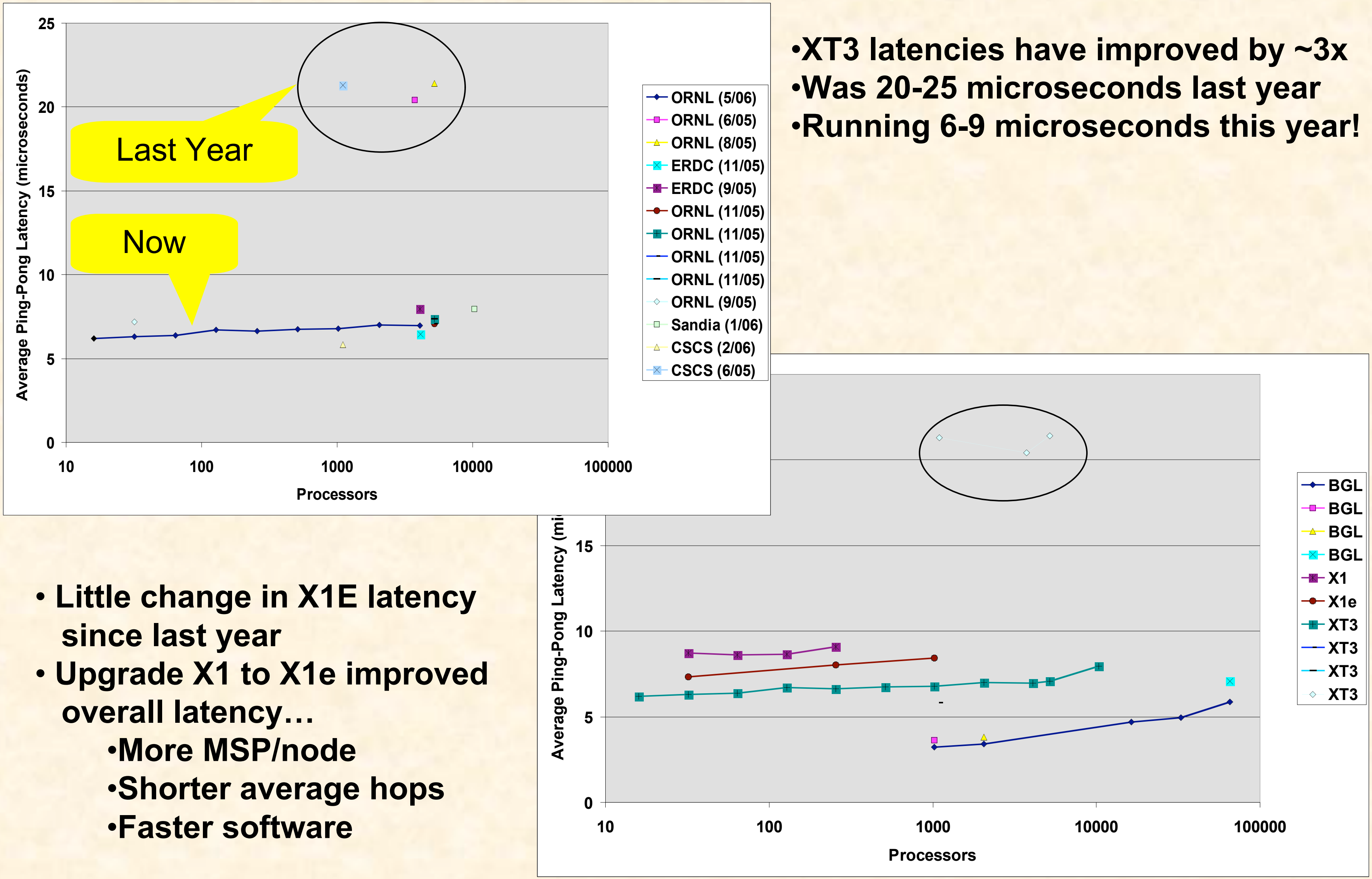
- Start with examining the basics
  - Latency
  - Bandwidth
  - SP mode and EP mode versions of the “4 corners”
    - DGEMM, FFT, RA, STREAM
- Look at global “4 corners” last
  - To understand contributions from basics

XT3 Current Latency Summary

- Two approaches to characterizing network performance
  - Time min and max xfer time -- report “latency” and assymptotic Bandwidth
  - Time multiple intermediate sizes – report chart of bandwidth vs msg size
- Measure Latency with short packets
- Network Latency for 3 Patterns
  - PingPong – indicative of good grid layout
  - RandomRing – indicative of pathological grid layout
  - Natural Order Ring – should look more like PingPong… poor job layout
  - Also see paper by D. Weisser (PSC) in CUG 2006 proceedings
- Latencies increase with PE count because of additional hops across torus/mesh
  - Job layout topology exacerbates the problem
- Think:
  - Random ring latency -- pathologically bad layout
  - PingPong -- good layout



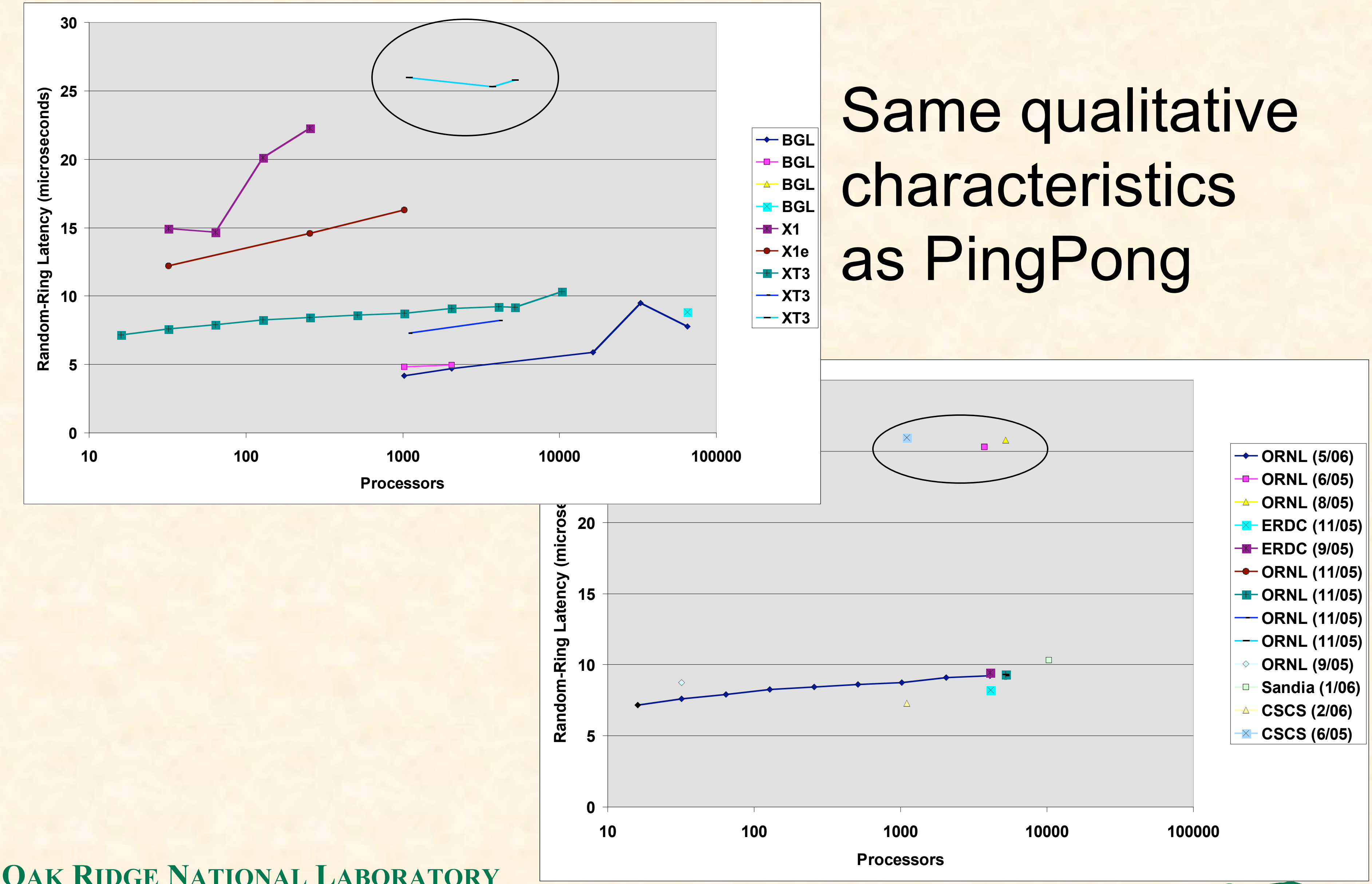
Latency Comparisons (PingPong)



- XT3 latencies have improved by ~3x
- Was 20-25 microseconds last year
- Running 6-9 microseconds this year!

- Little change in X1E latency since last year
- Upgrade X1 to X1e improved overall latency…
  - More MSP/node
  - Shorter average hops
  - Faster software

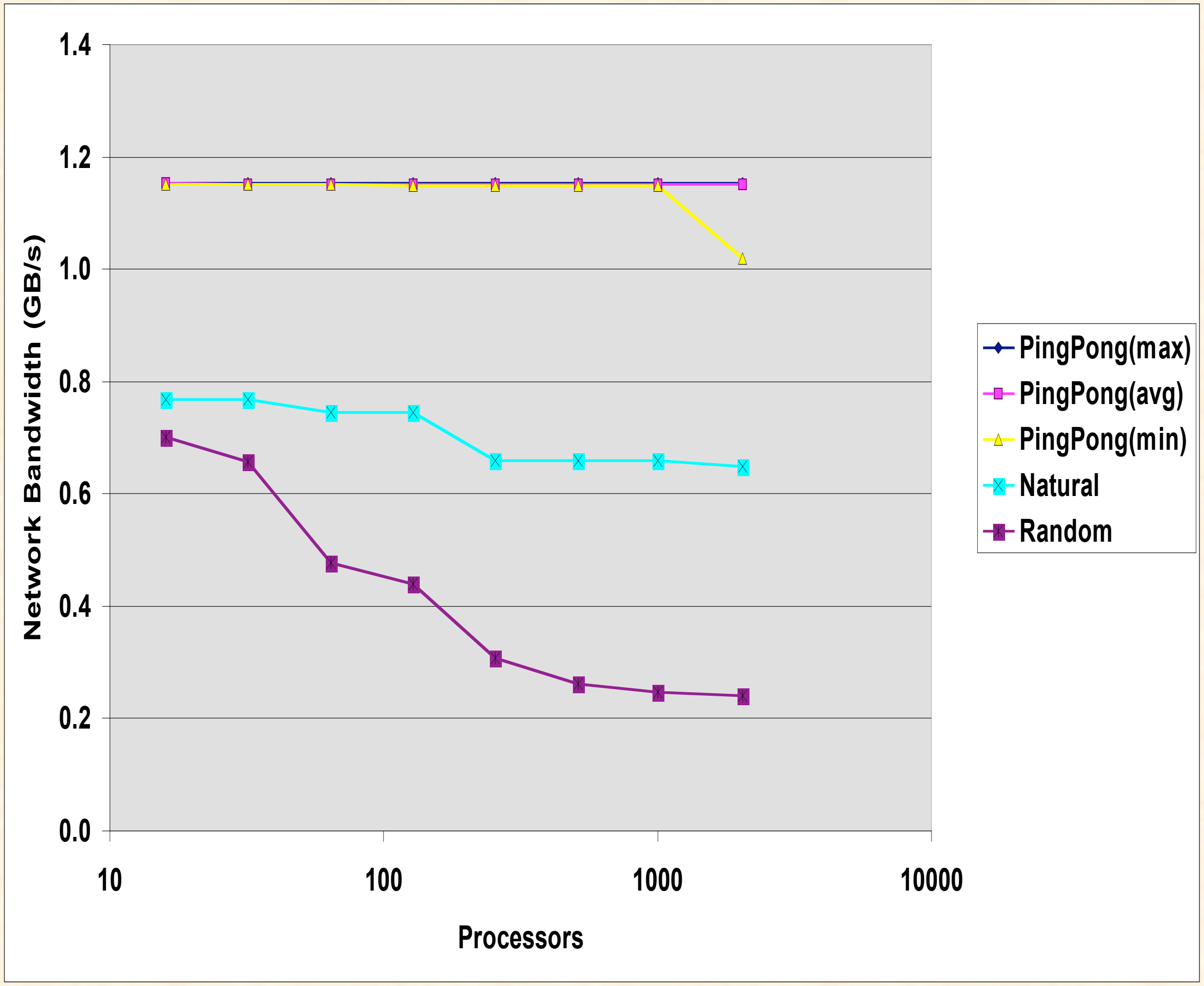
Latency Comparison (RandomRing)



Same qualitative characteristics as PingPong

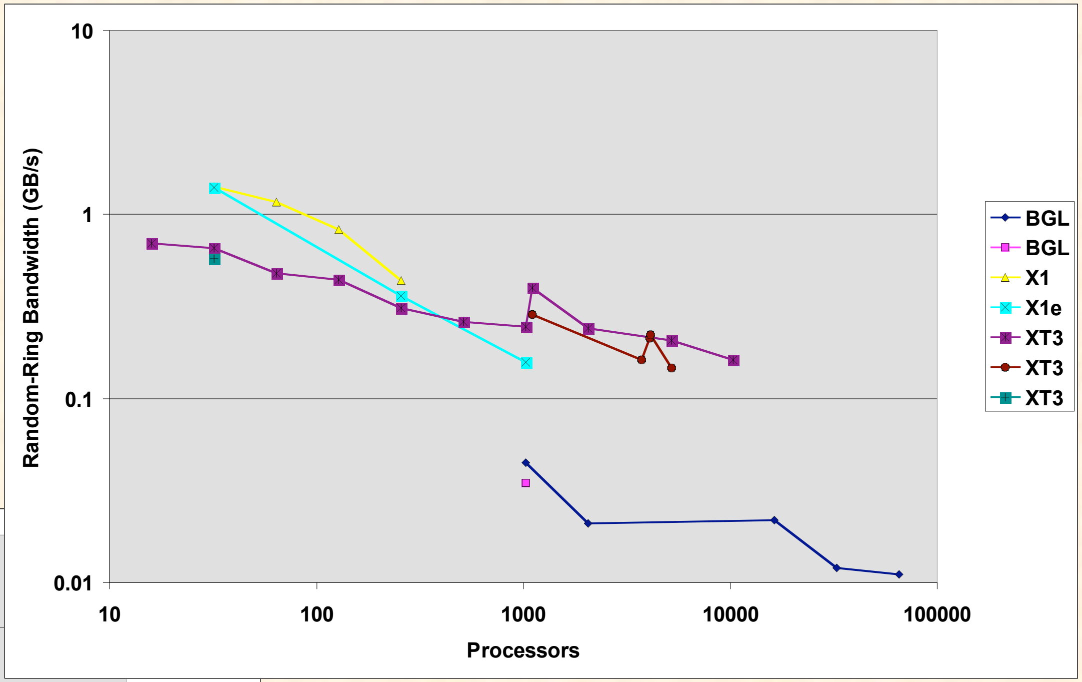
XT3 Bandwidth Summary

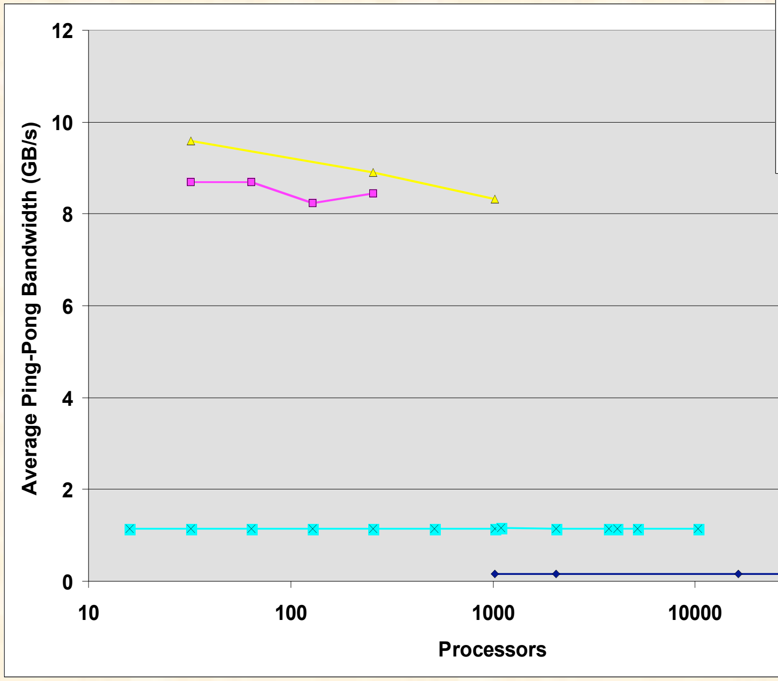
- Recall: two approaches to characterizing network performance
  - Time min and max xfer time -- report “latency” and assymptotic Bandwidth
  - Time multiple intermediate sizes – report chart of bandwidth vs msg size
- Job layout problem again
  - Natural ring should be closer to PingPong than RandomRing
  - See also paper by Deborah Weisser (PSC), et.al. in this CUG’s proceedings



Network Bandwidth Comparison

- X1 vs X1e performance
  - Faster processor may account for difference between X1 and X1e

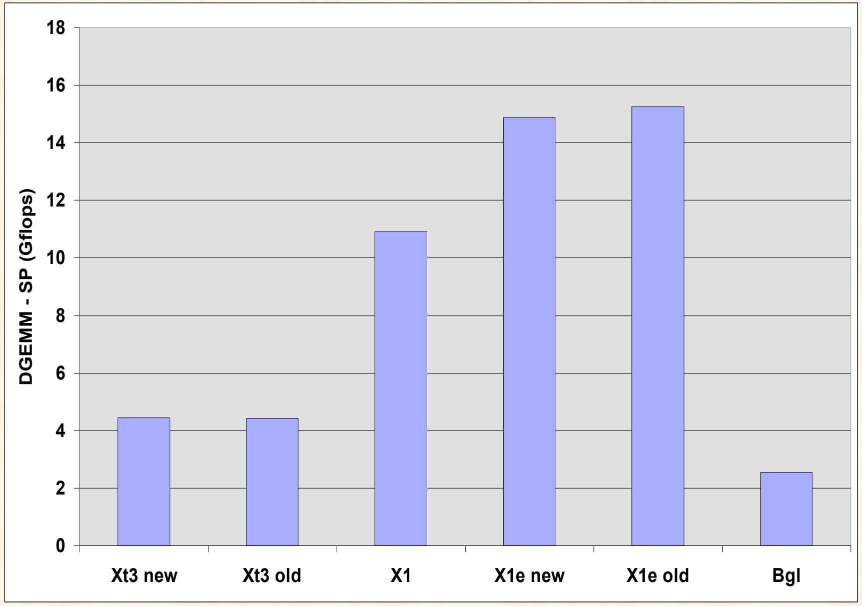


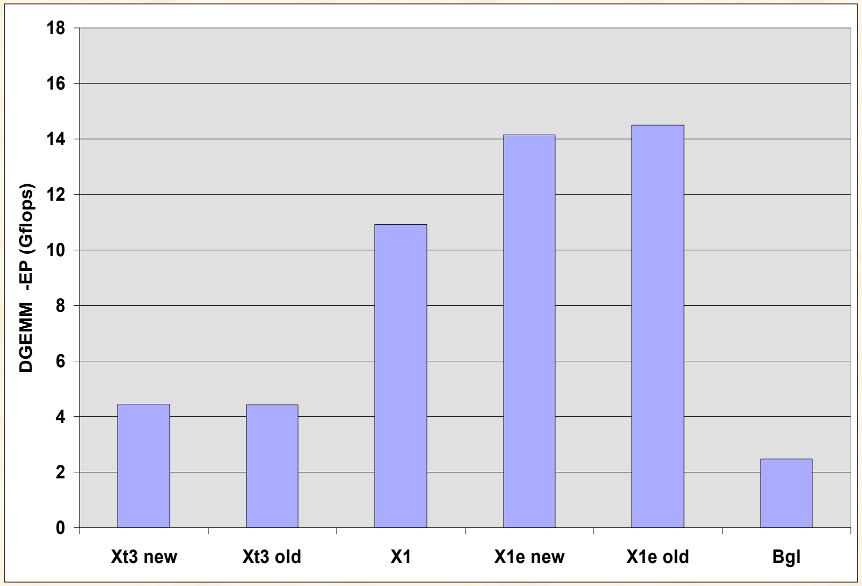


- Note 2.6GHz XT3 points
  - Seastar-Opteron interrupts to handle messages?
  - But… spikes not seen on XT3 latency plots
  - More likely job layout dependent on XT3

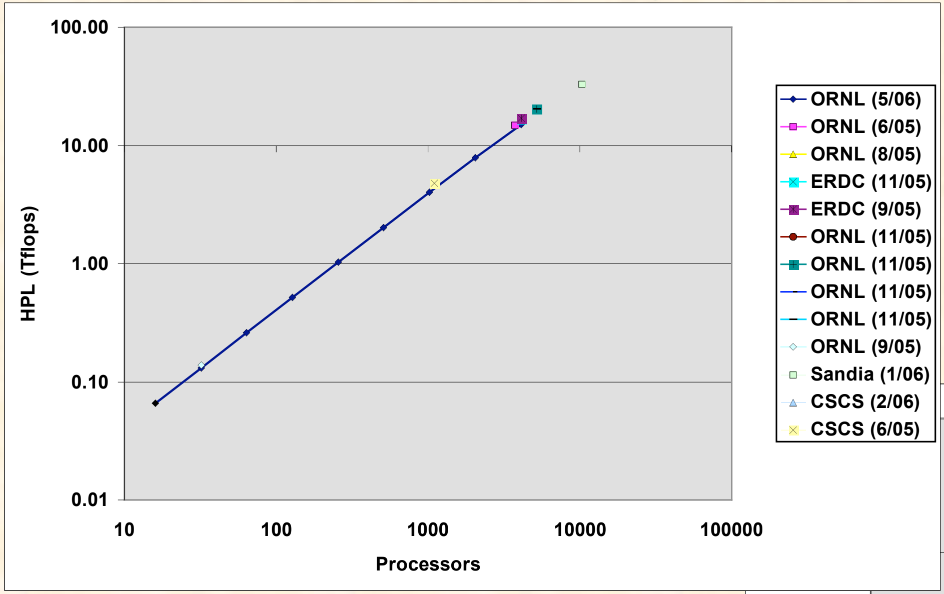
SP/EP DGEMM Summary

- Linear Algebra (matrix-matrix multiply)
- High spatial and temporal cache locality
- Depends on BLAS library
- Sensitive to problem size and blocking factor
  - N and NB from HPL input
  - Good ~ 10x Bad
- XT3 shows little change
  - Blame ACML…for 92.5% of peak ☺
  - ACML 2.6 vs 3.0
- X1E also shows little change
  - Dedicated vs non-dedicated resource
  - Craylibs 5.4 vs 5.5

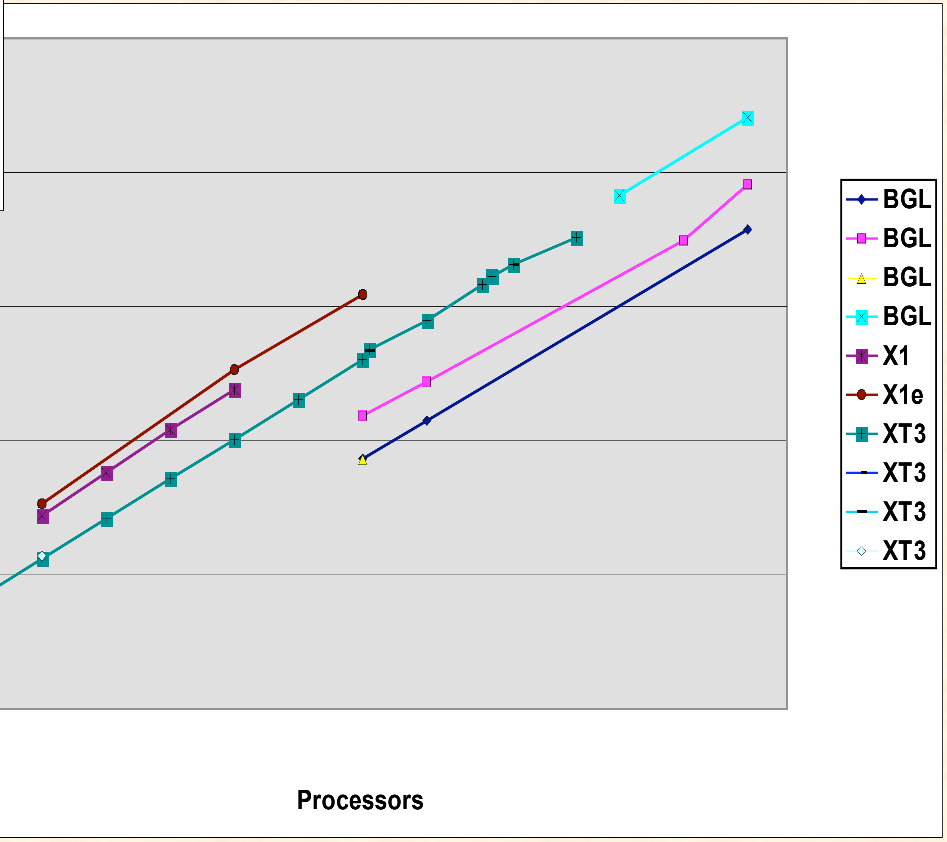




HPL Summary



- •XT3: scales well to 10K pe's
- •XT3: dip at 10K – 2 GHz system@Sandia
- •BGL: "opt" line similar to XT3
- •BGL: "base" much slower
- •Accounts for most of the runtime on majority of systems



Caveats:

- Beware: slopes on log-log charts!
- BGL "optimized" runs
- XT3: 2.0 vs. 2.4 vs 2.6
- Problem size variations
- Compiler and Library Variations
- Lines indicate trends only!!
- Very sensitive to N, NB, P, Q
- Good ~10x Bad

“Normalized” HPL Summary

Pros:

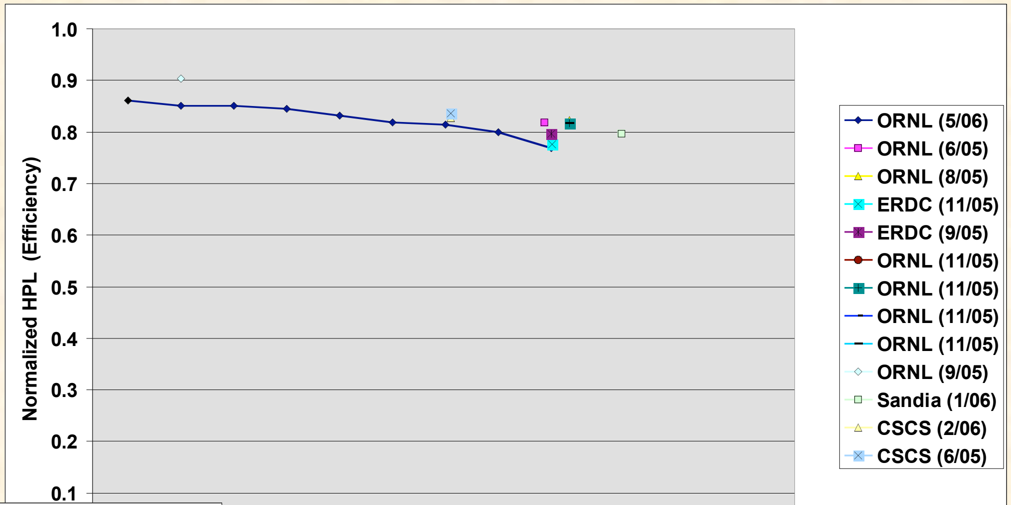
- Removes “wallet size” from equation

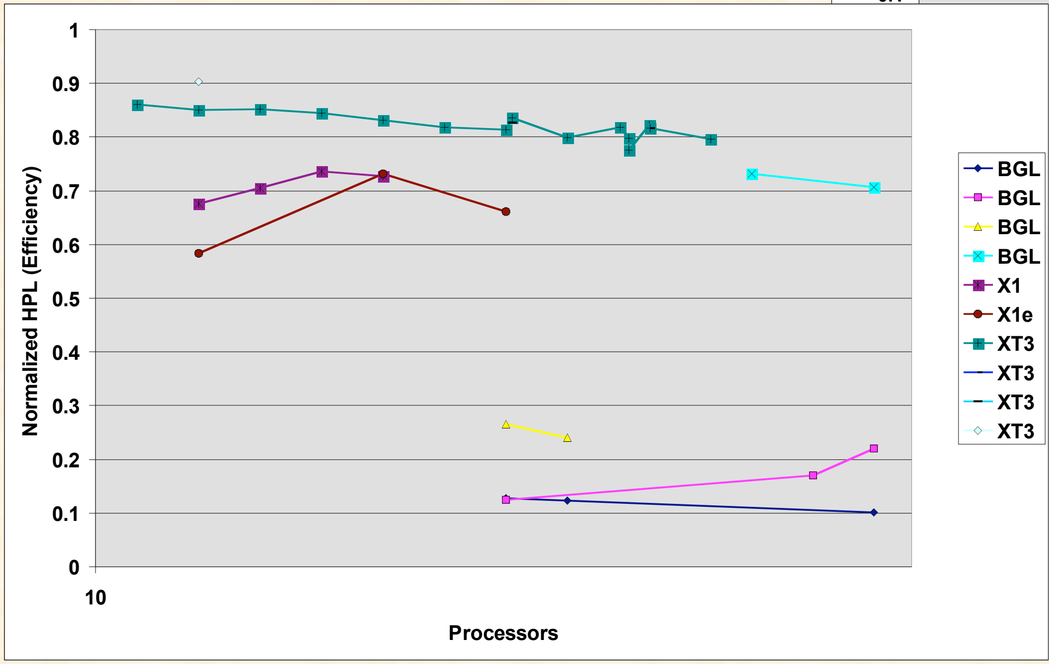
Cons:

- Removes “wallet size” from equation ☺

Thoughts:

- Another scalability/balance perspective
- For HPL, just % of peak
- Similarity to EP/SP metrics
- Normalize vs Memory Bandwidth, FP Peak, number of nodes, etc.

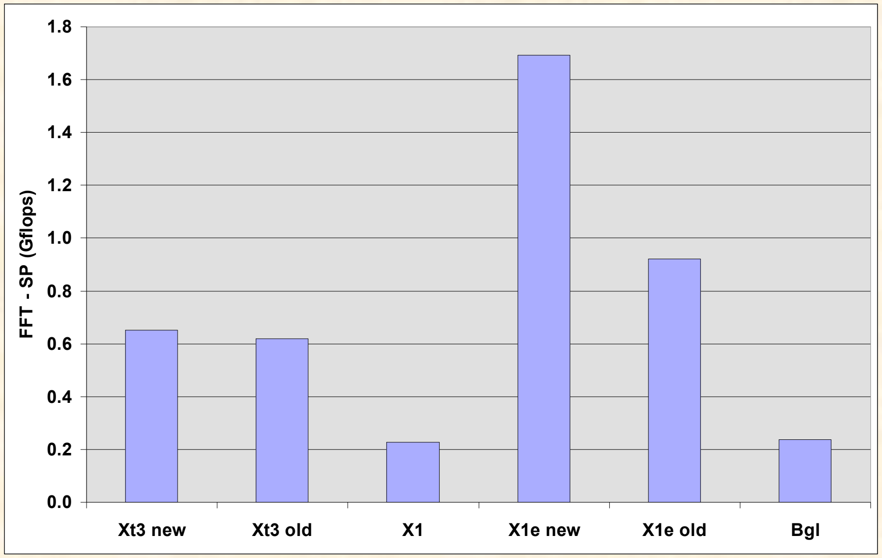


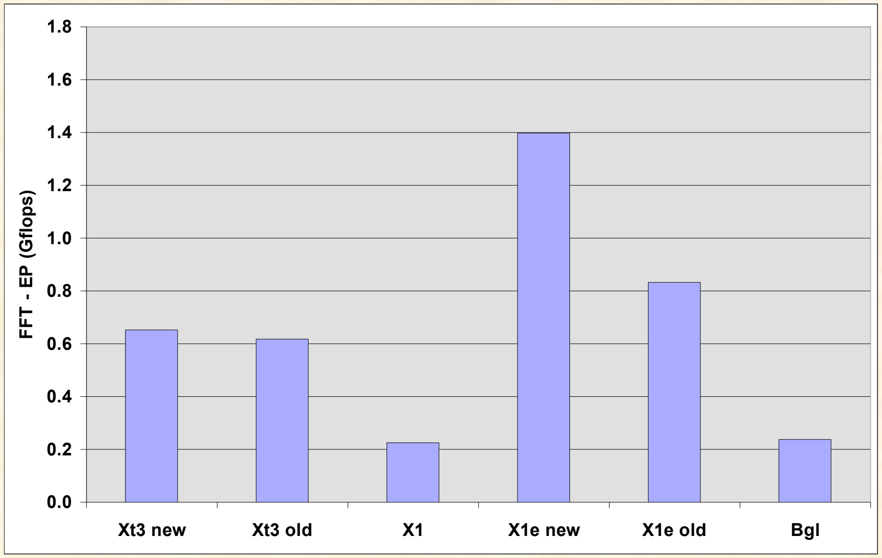


- XT3 performance stable at scale
- XT3 sustains high percentage of peak
- BGL performance requires both CPUs and both FPUs/CPU
- BGL competitive per node w/ “opt”
- Vectors don’t always win %peak

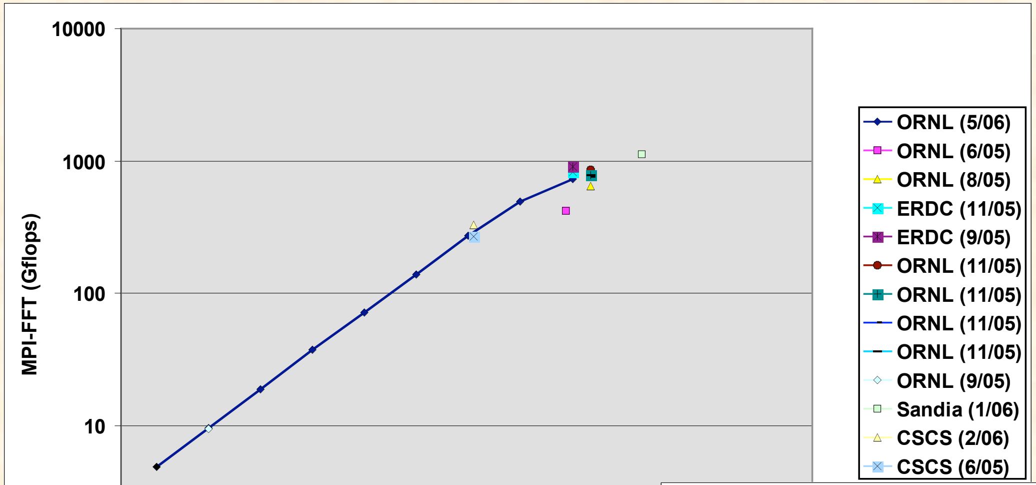
SP/EP FFT Summary

- Fast Fourier Transform
  - High temporal locality
    - Re-use recently accessed data
  - Low spatial locality (adjacent)
    - Accesses typically non-adjacent
- Used by spectral solvers
- Fortran code processed by “f2c”
- Demands much from compiler
- XT3 FFT has improved a bit
  - PGI 6.0 vs 6.1
- X1E FFT performance has almost doubled
  - PrgEnv 5.4 vs 5.5





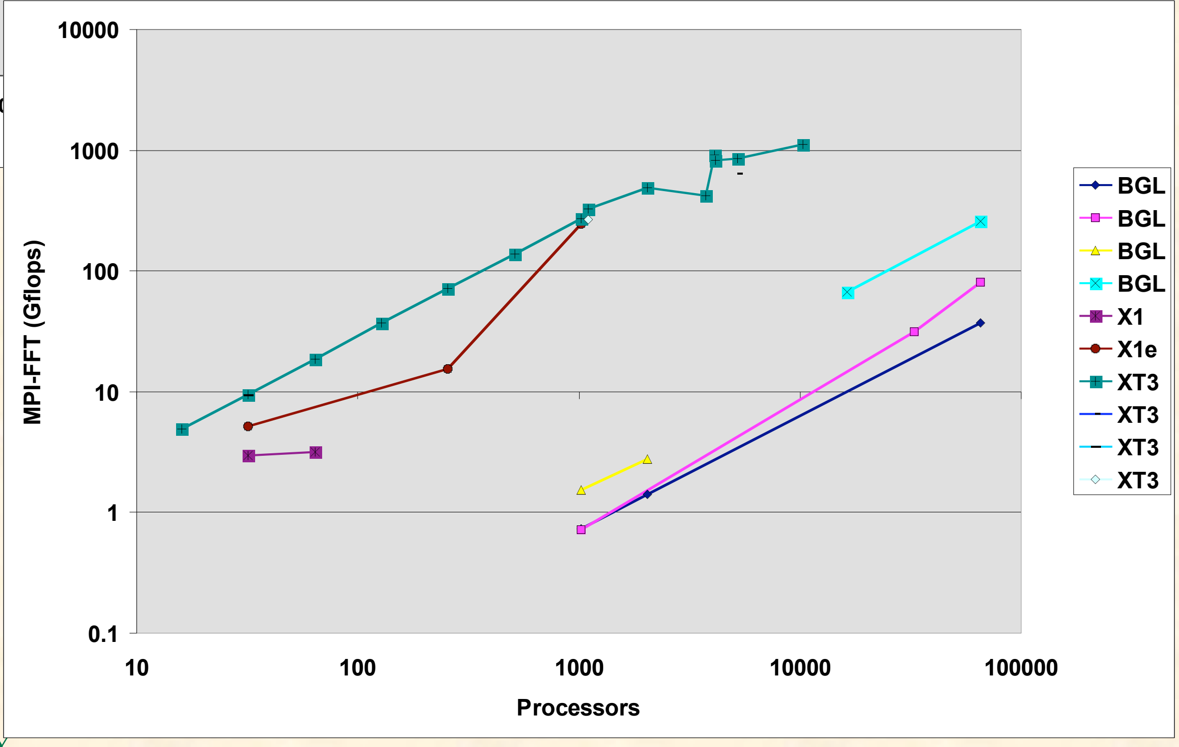
MPI FFT Summary



- Note “flat step”
  - 3744 processors
  - 5200 processors
  - implementation uses power-of-2 processor counts

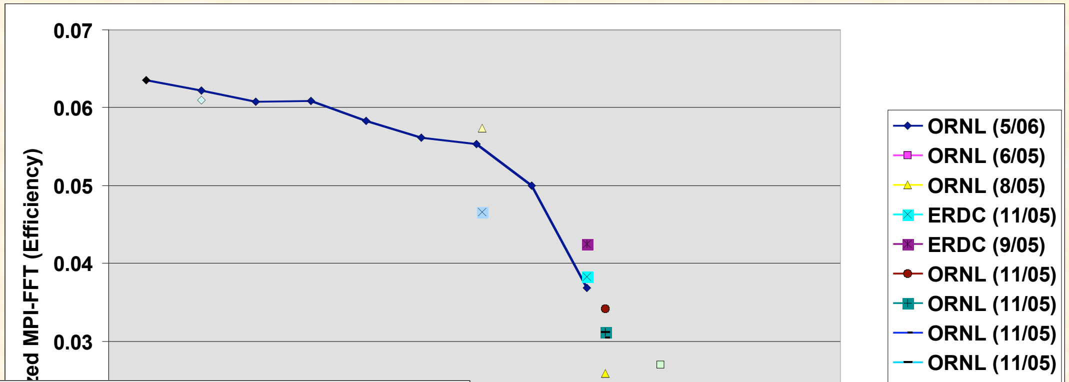
- XT3 compiler improvements were small for SP/EP FFT
- XT3 network latency increased dramatically
- XT3 bandwidth unchanged

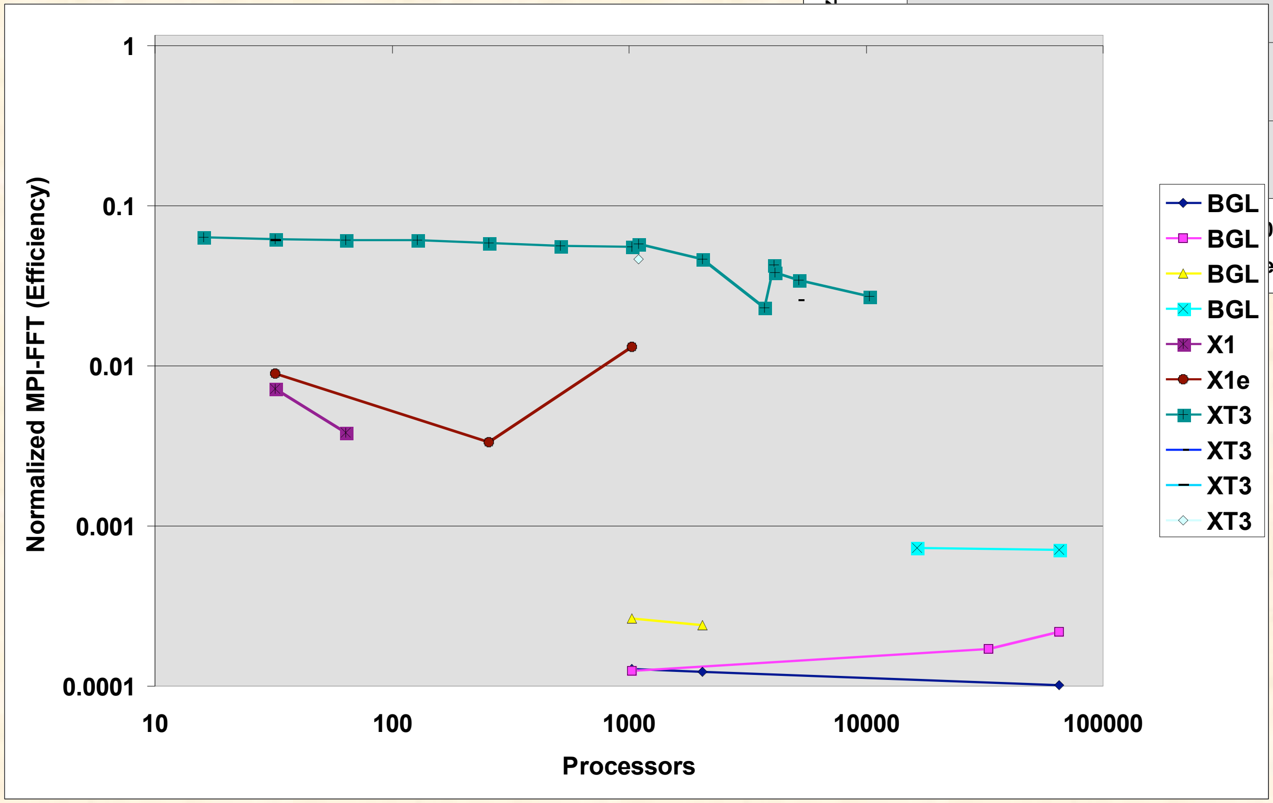
- Did XT3 MPI-FFT change?



Normalized MPI FFT Summary

- Again, note steps at 2^n PE's
- Older XT3 points vs trendline
  - •Was there an improvement?
  - •Depends on how trendline drawn
  - •Historical data not sufficiently consistent

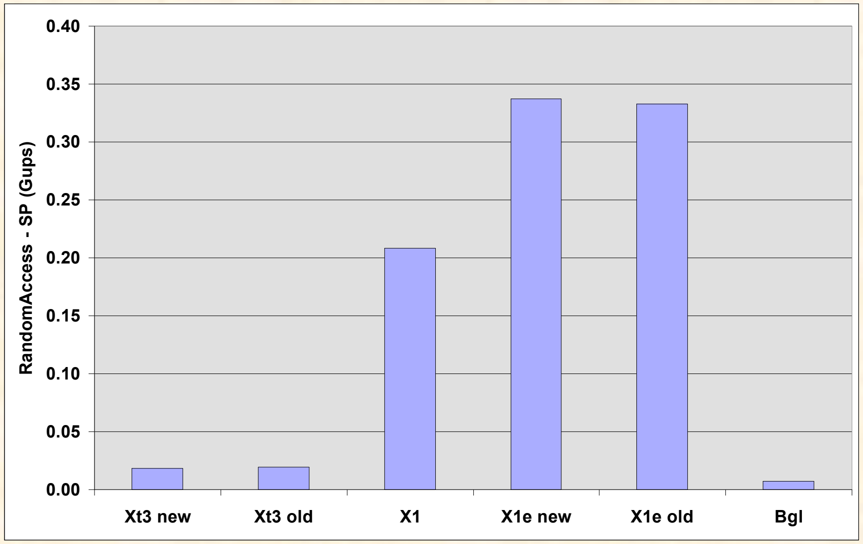


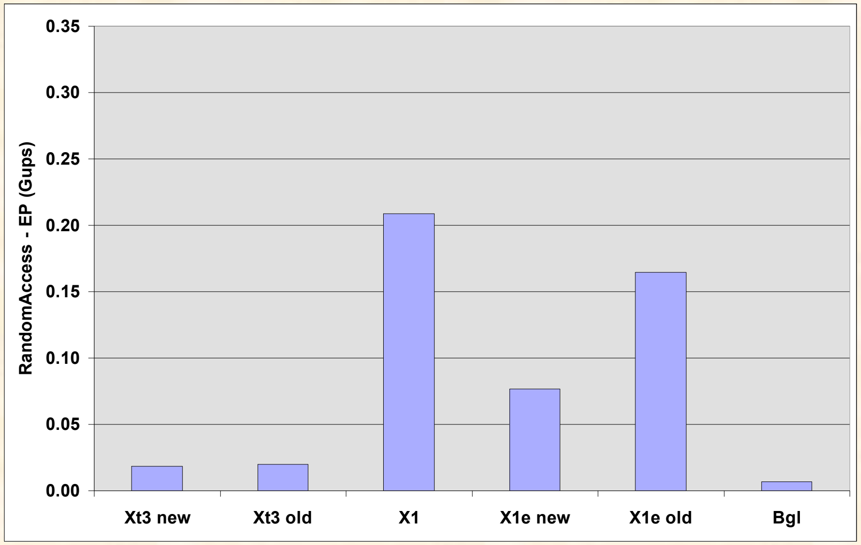


Note: X1 & X1e historical points not power of 2 processor count.

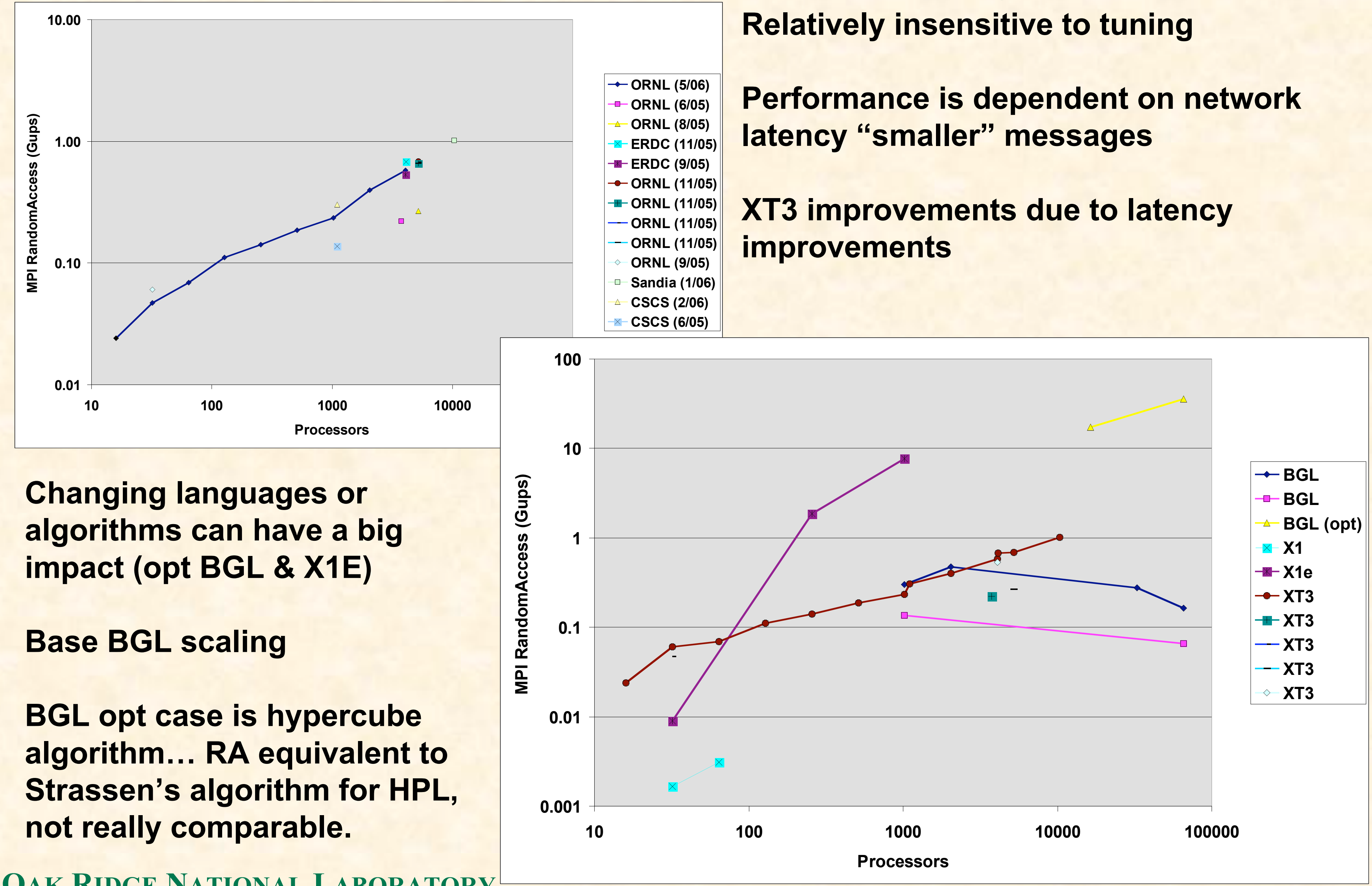
SP/EP RandomAccess Summary

- Graph theory Kernel
  - Updates table elements
- Memory access really does show zero spatial/temporal locality
  - If the cache hit rate isn't zero, the problem size isn't big enough ☺
- Power of 2 memory usage
- XT3 PGI 6.0 vs 6.1
  - ~5% loss
  - But worse: GCC 15% faster? Maybe… (size differences)
  - Room for improvement here
- X1E PE 5.4 vs 5.5
  - Small improvement on SP
  - Large loss on EP
    - Still investigating cause
    - Contention?
    - Dedicated vs non-dedicated resource?
    - No conclusion yet.





MPI RandomAccess Summary



Relatively insensitive to tuning

Performance is dependent on network latency “smaller” messages

XT3 improvements due to latency improvements

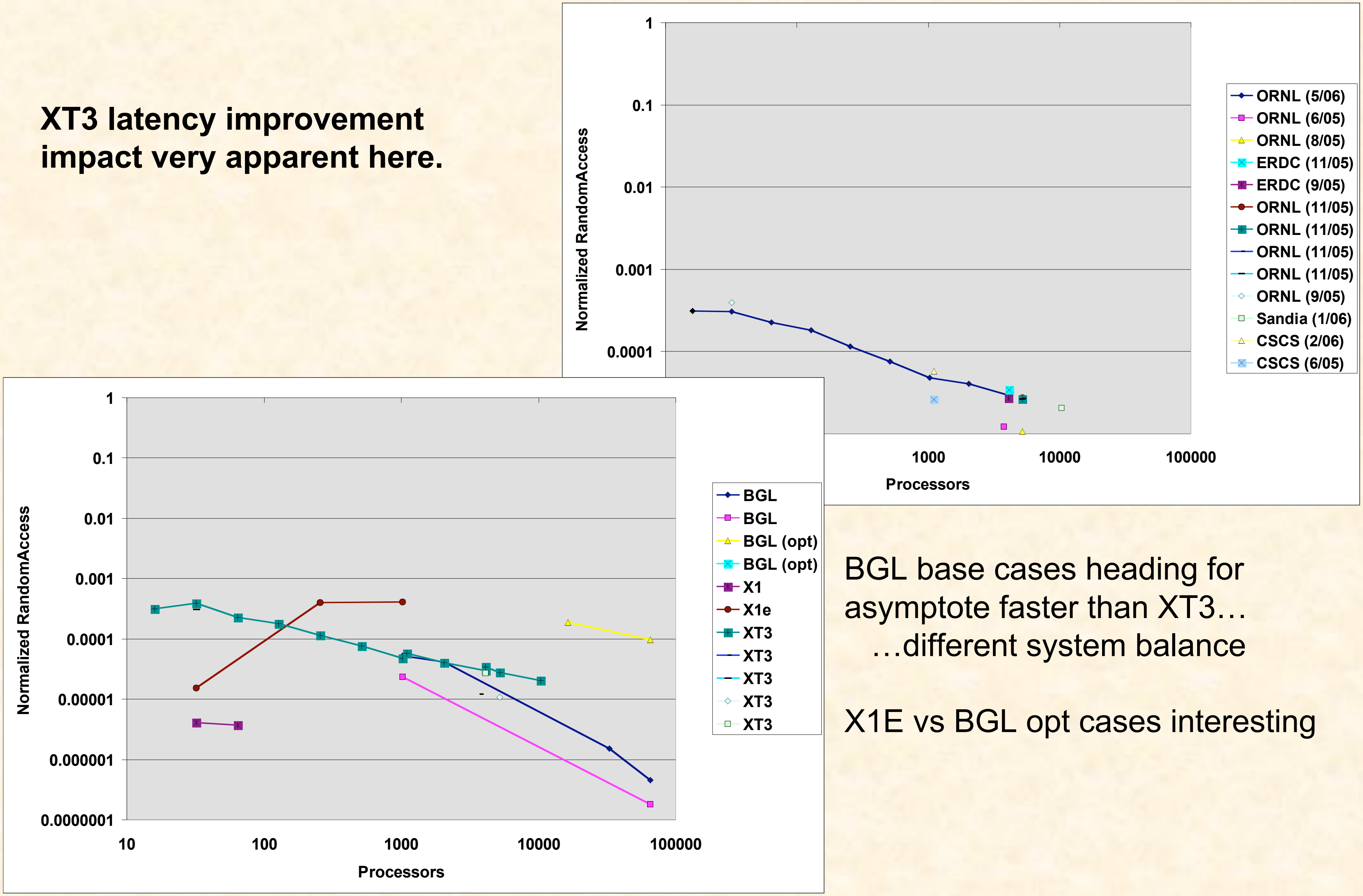
Changing languages or algorithms can have a big impact (opt BGL & X1E)

Base BGL scaling

BGL opt case is hypercube algorithm… RA equivalent to Strassen’s algorithm for HPL, not really comparable.

Normalized MPI RandomAccess

XT3 latency improvement impact very apparent here.

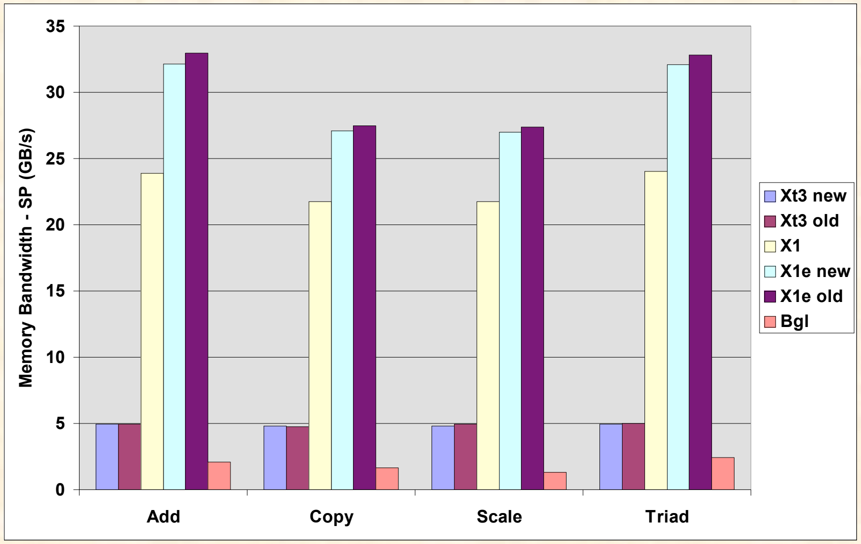


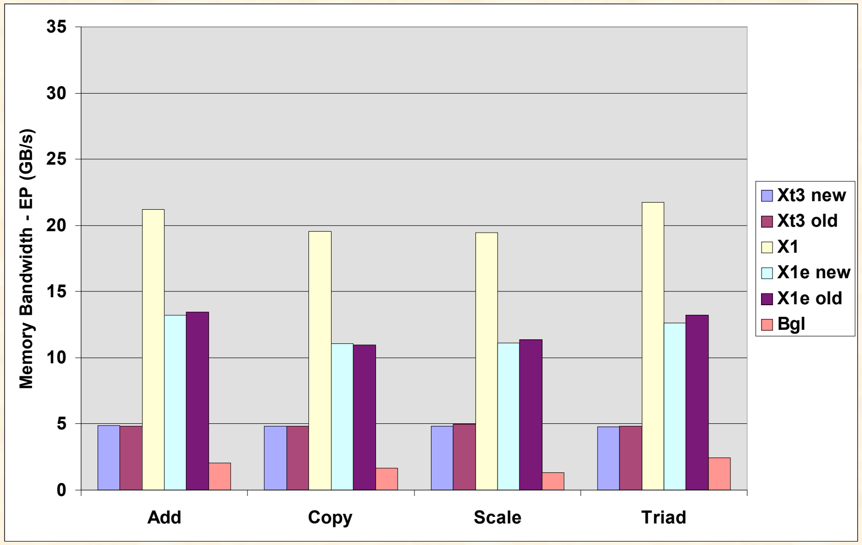
BGL base cases heading for asymptote faster than XT3…
…different system balance

X1E vs BGL opt cases interesting

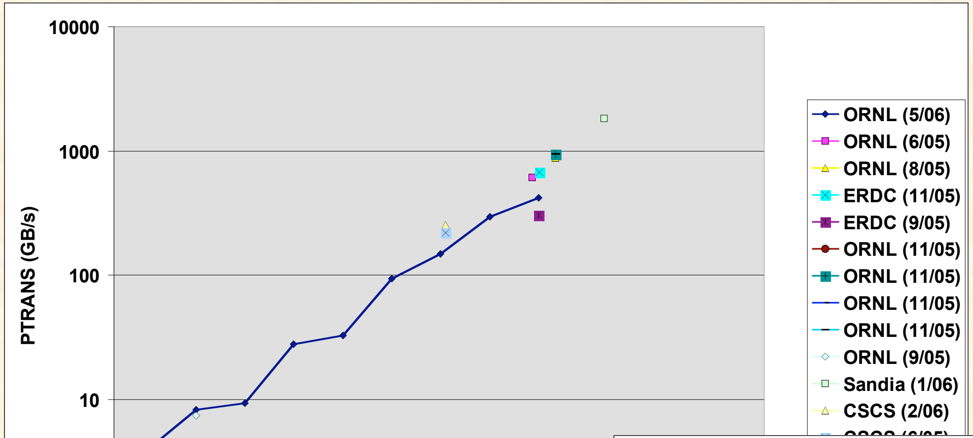
SP/EP STREAM Summary

- Sequential memory access at 4 different CI's:
  - Copy: A(i)=B(i) (0)
  - Add: A(i)=B(i)+C(i) (1/3)
  - Scale: A(i)=s*B(i) (1/2)
  - Triad: A(i)=s*B(i)+C(i) (2/3)
- Compiler can impact performance
- But array layout is critical
  - Bad layout can cost 2x
  - No direct way to control
  - Adjust HPL size (N)
- XT3: Little to no change
- X1E: small loss
  - Dedicated vs non-dedicated resource?

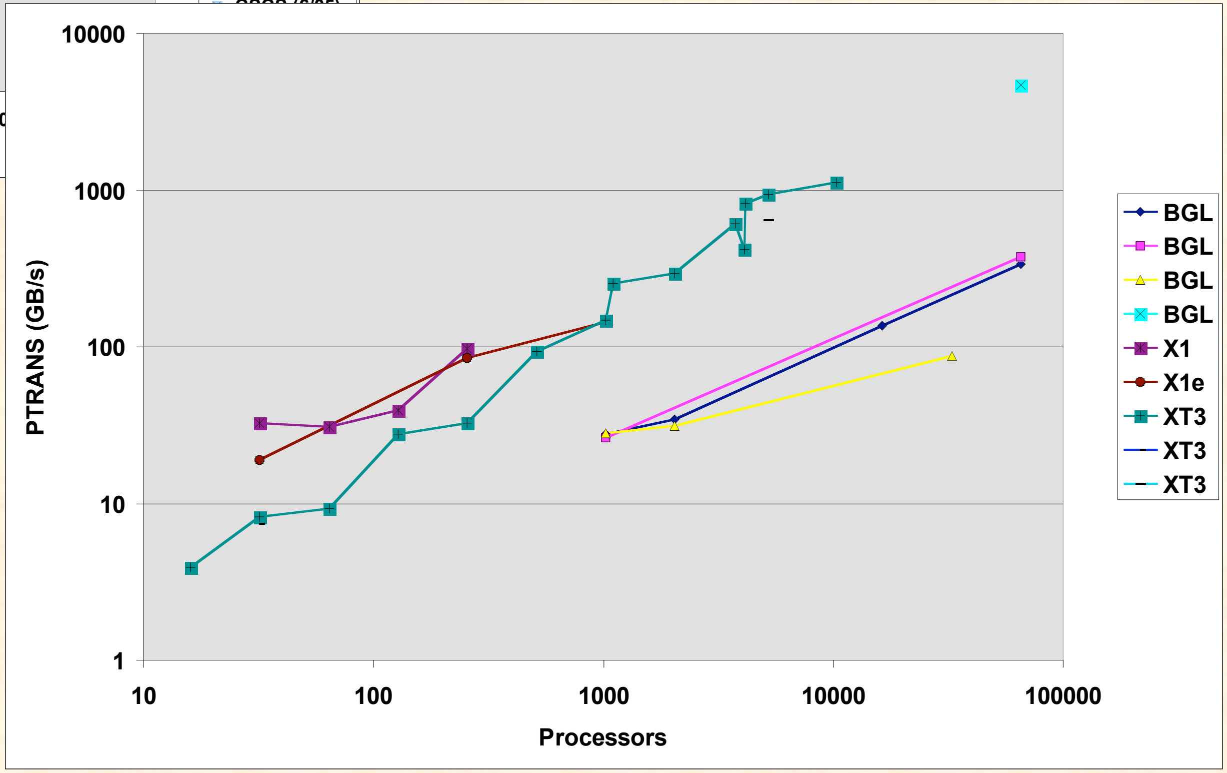




PTRANS Summary



- Memory and network bandwidth
- Extremely sensitive to input parameters
- Job layout also impacts (See Weisser paper for details)
- “Benchmark the benchmarkers”

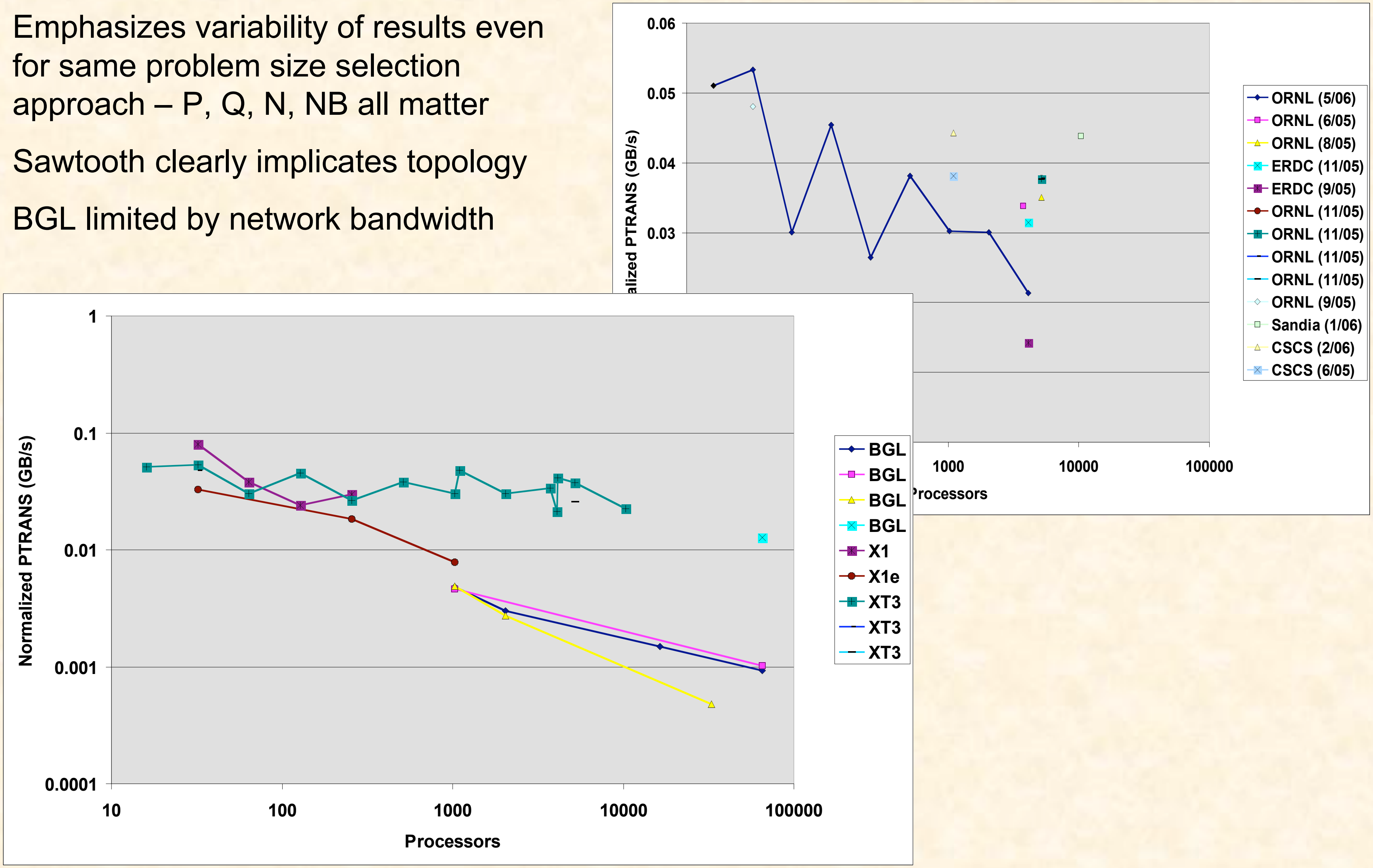


Normalized PTRANS Summary

Emphasizes variability of results even for same problem size selection approach – P, Q, N, NB all matter

Sawtooth clearly implicates topology

BGL limited by network bandwidth



Summary Improvements

- Big Improvements:
  - XT3 network latency and thus MPI-RandomAccess
  - X1E FFT performance (SP/EP)
- Small improvements
  - XT3 FFT performance (SP/EP)
- Small degradations / Further attention needed
  - XT3 Random Access (SP/EP)
  - Room for improvement here – GCC is ~15% faster
  - PTRANS, B_{eff} – job layout issues (see Weisser paper for details)
- Inconclusive
  - XT3 MPI FFT – more careful repeats of old experiments
  - X1E RandomAccess (SP/EP/MPI-FFT) – investigating

PetaFLOP in 2008

Want to help?

http://www.ornl.gov