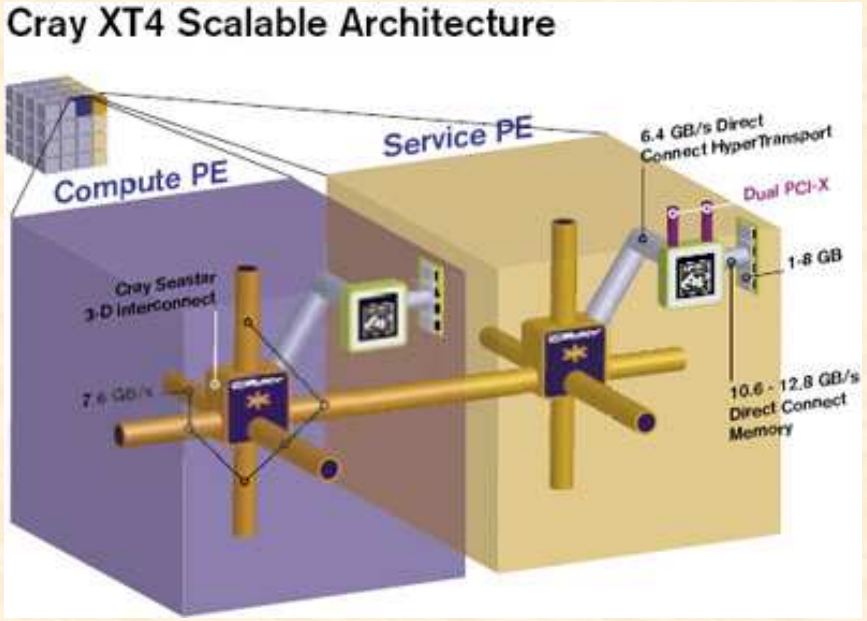
HPCC Results and Analysis from ORNL's XT3/XT4 System

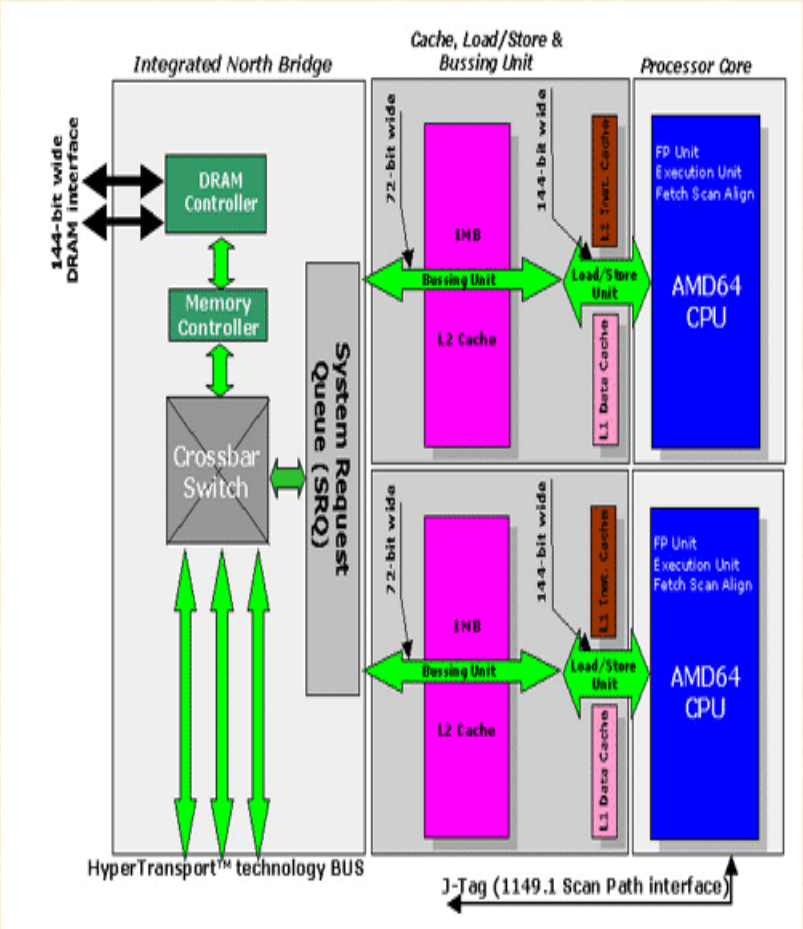
Jeffery A. Kuehn, ORNL

Jeff Larkin, Cray Inc

Nathan Wichmann, Cray Inc

What Changed?





Major Changes:

- **Processor: Single core -> Dual Core**
- **Processor: 2.4 -> 2.6 GHz**
- **Memory: DDR-400 -> DDR2-667**
- **SeaStar 1.2 -> 2.1 fixes injection bandwidth**

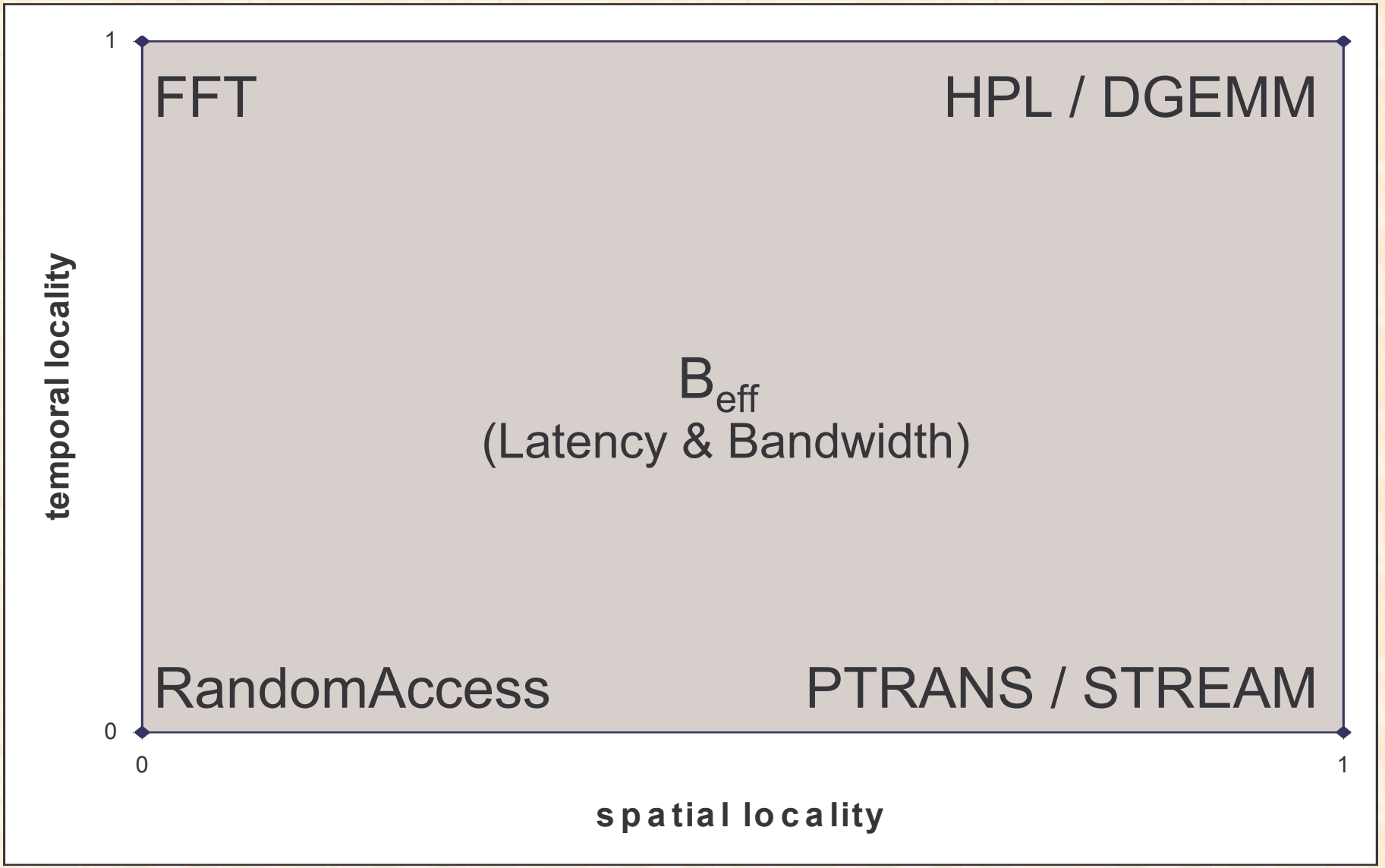
What didn't change:

- **Memory Capacity held at 2GB/core**

Overview of the HPCC Benchmark

- Global Performance
  - HPL
  - PTRANS
  - MPI FFT
  - MPI RandomAccess
- Local Performance (2 modes: SP and EP)
  - DGEMM
  - STREAM
  - FFT
  - RandomAccess
- Network Latency and Bandwidth
  - PingPong (min – avg – max)
  - Natural Ring
  - Random Ring

Overview of the HPCC Benchmark

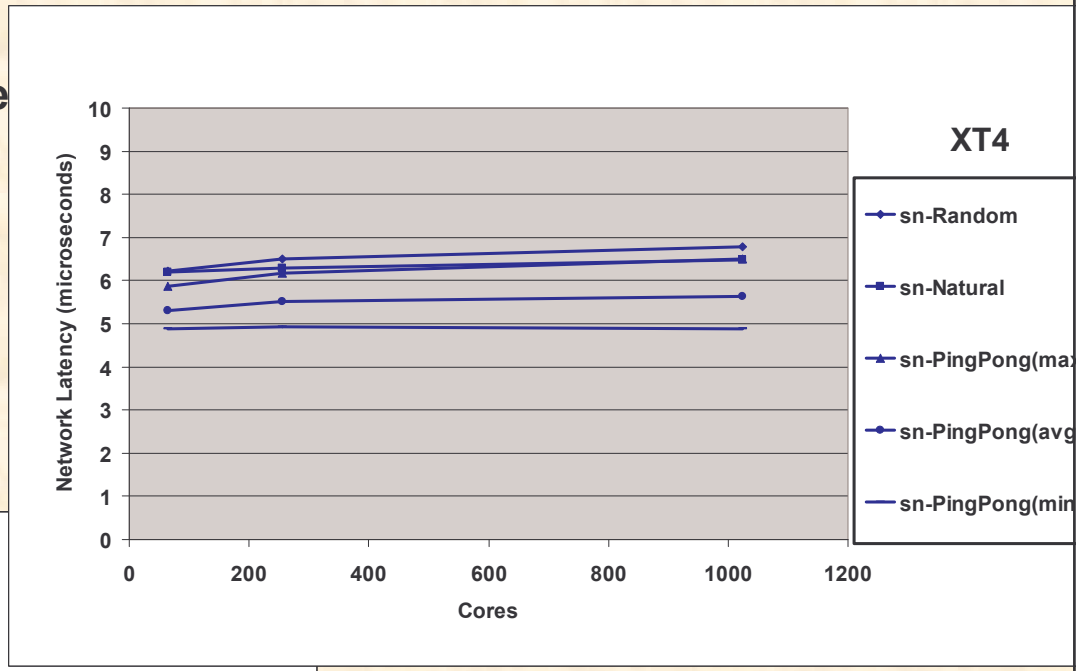


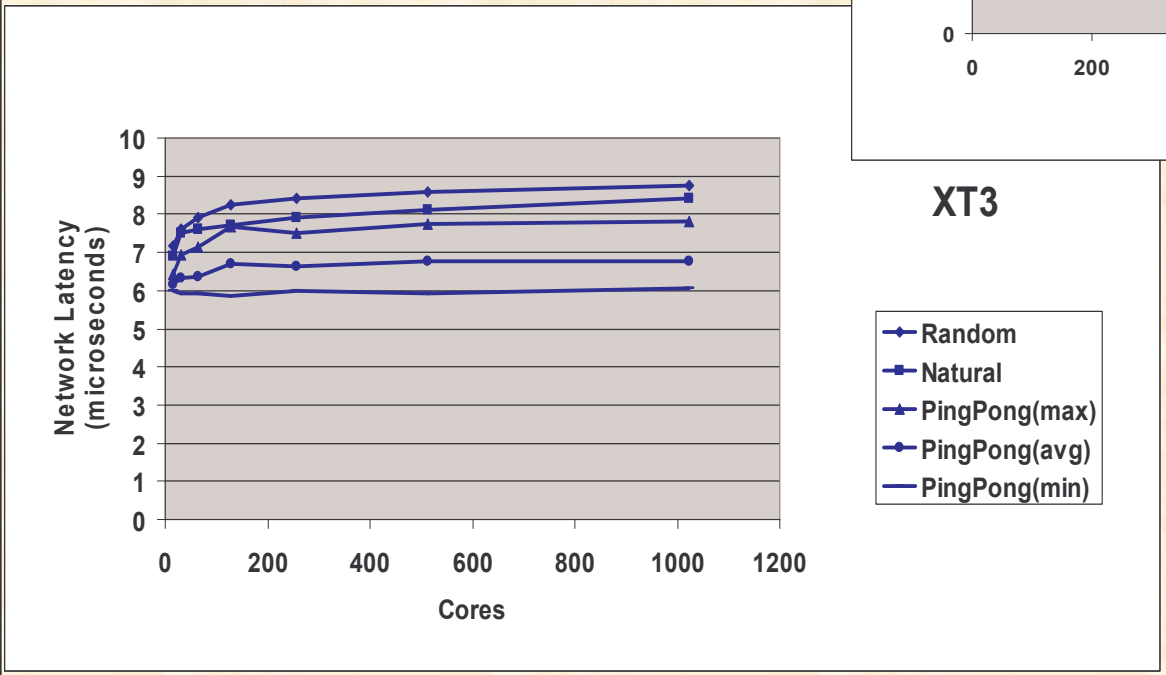
Benchmark OS/PE modules

- ***pgi/6.2.5***
- xt-boot/1.5.31
- ***xt-catamount/1.5.31***
- xt-crms/1.5.31
- xt-libc/1.5.31
- xt-libsci/1.5.31
- xt-lustre-ss/1.5.31
- xt-mpt/1.5.31
- xt-os/1.5.31
- xt-pbs/5.3.5
- xt-pe/1.5.31
- xt-service/1.5.31
- Base-opts/1.5.31
- DefApps
- MiscApps
- MySQL/4.0.27
- PrgEnv-pgi/1.5.31
- ***acml/3.6***
- iobuf/1.0.4
- moab/5.0.0
- modules/3.1.6
- mts/0.1
- totalview/7.3.0

XT3 vs XT4 Latency Summary

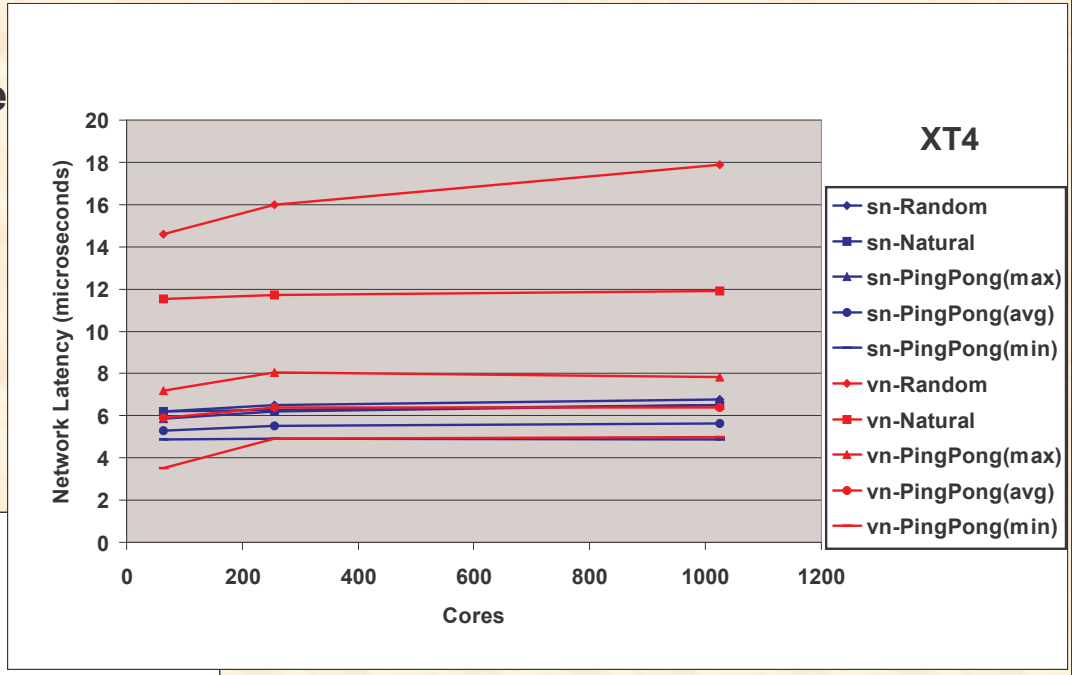
- Latency generally increases as core count increases – more hops
- VN latencies spread higher
  - NIC contention between cores

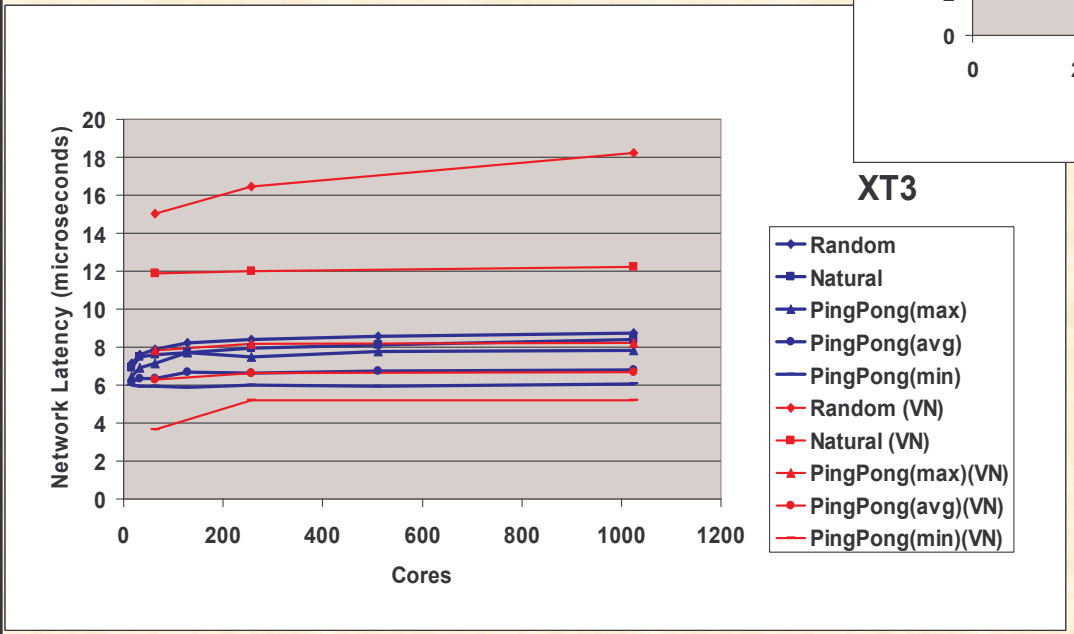




XT3 vs XT4 Latency Summary

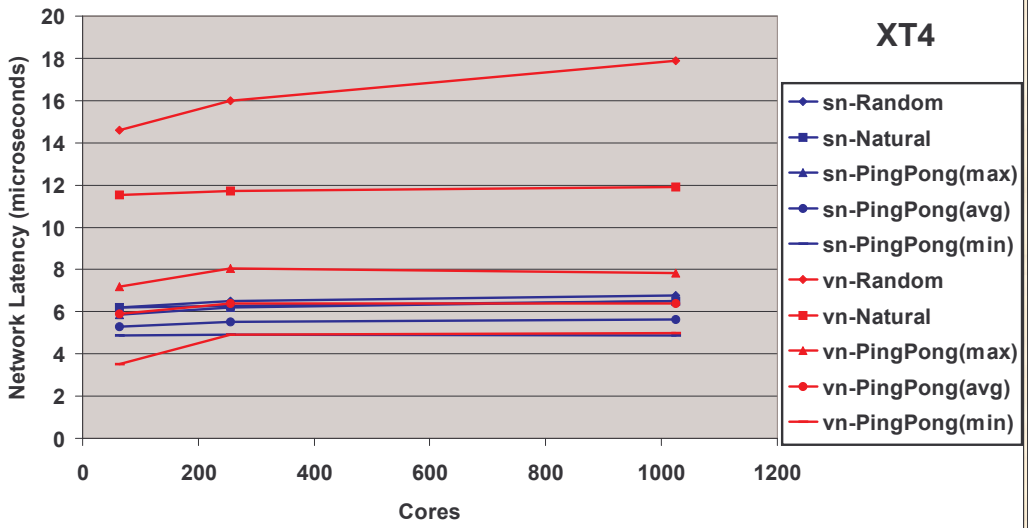
- Latency generally increases as core count increases – more hops
- VN latencies spread higher
  - NIC contention between cores

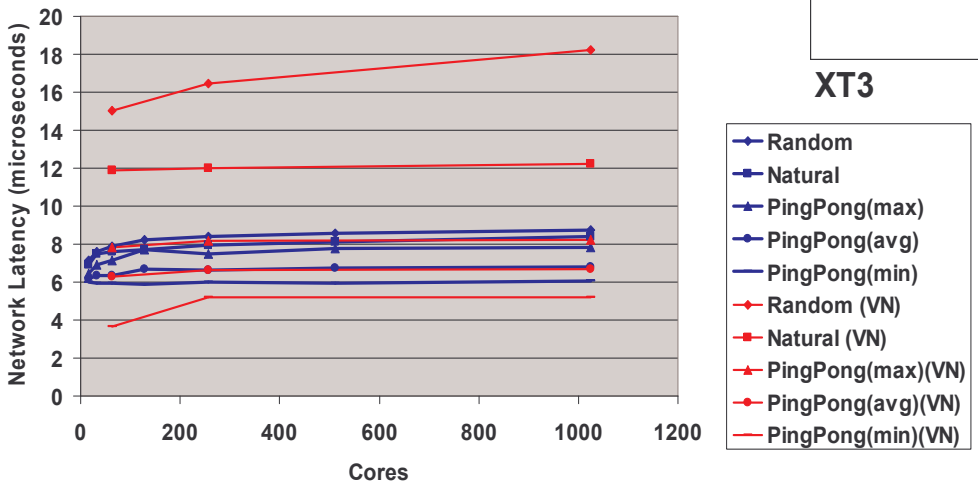




XT3 vs XT4 Latency Summary

- Latency generally increases as core count increases – more hops
- VN latencies spread higher
  - NIC contention between cores

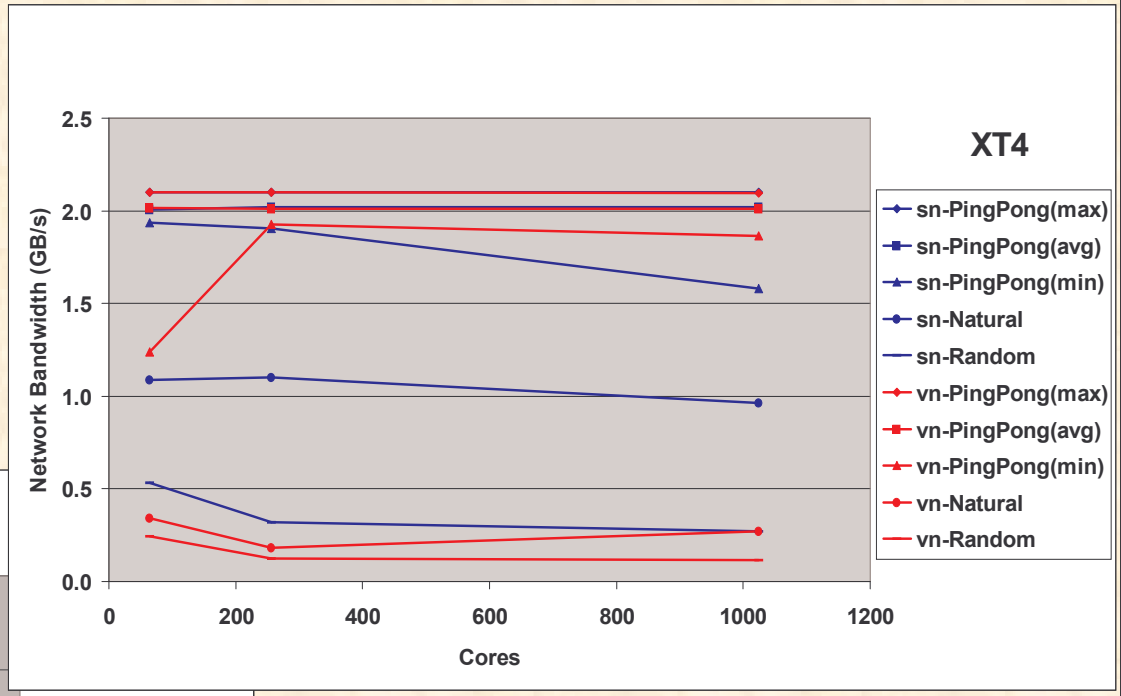


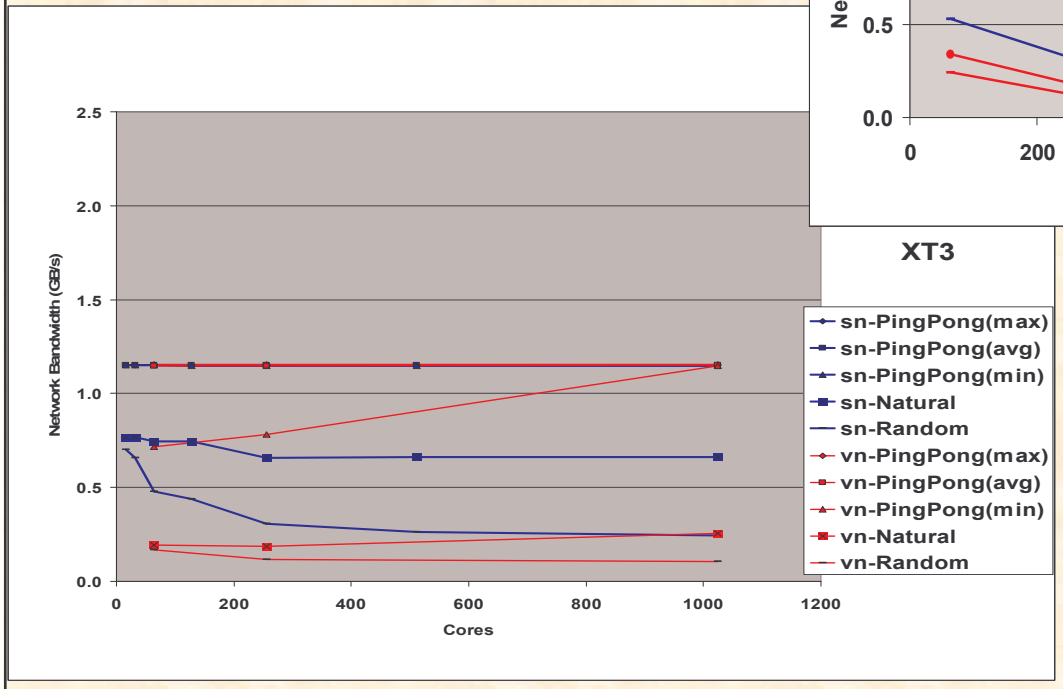


MPICH_PTL_MATCH_OFF environment variable (1.5.39+) has been shown to improve latency 10-44% by disabling registration of receive requests with portals

XT3 Bandwidth Summary

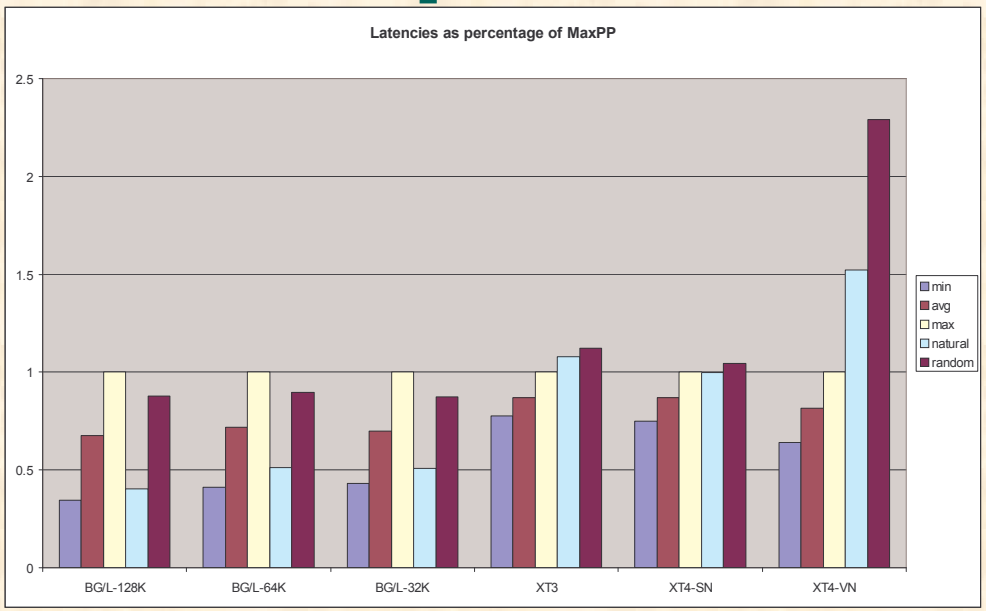
- XT4 improved injection BW
  - PingPong
- Per core Ring BW:
  - SN improved
  - VN better per socket
    - Link contention again

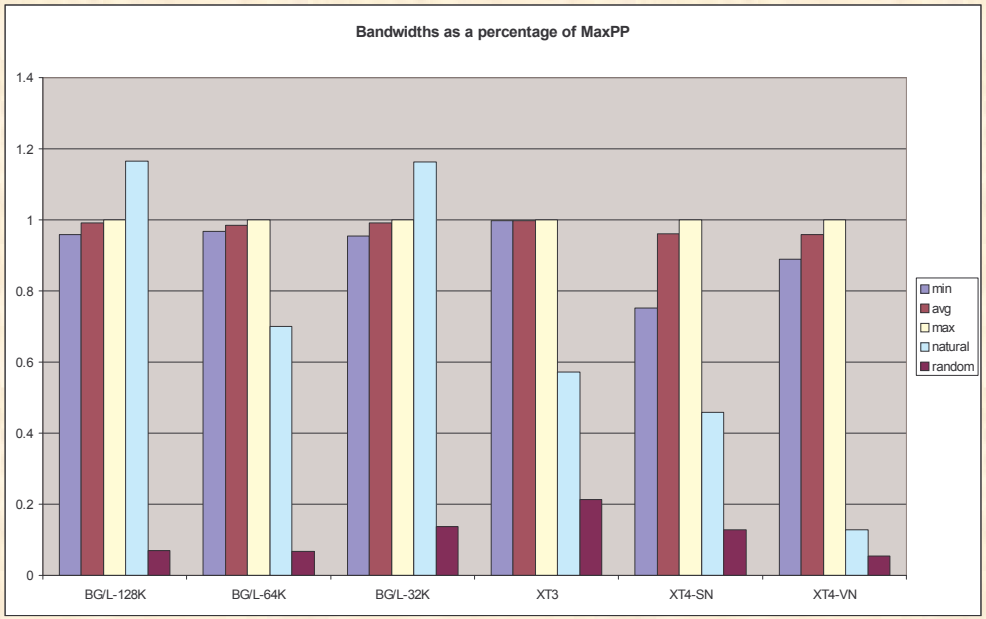




Latency & Bandwidth Comparison

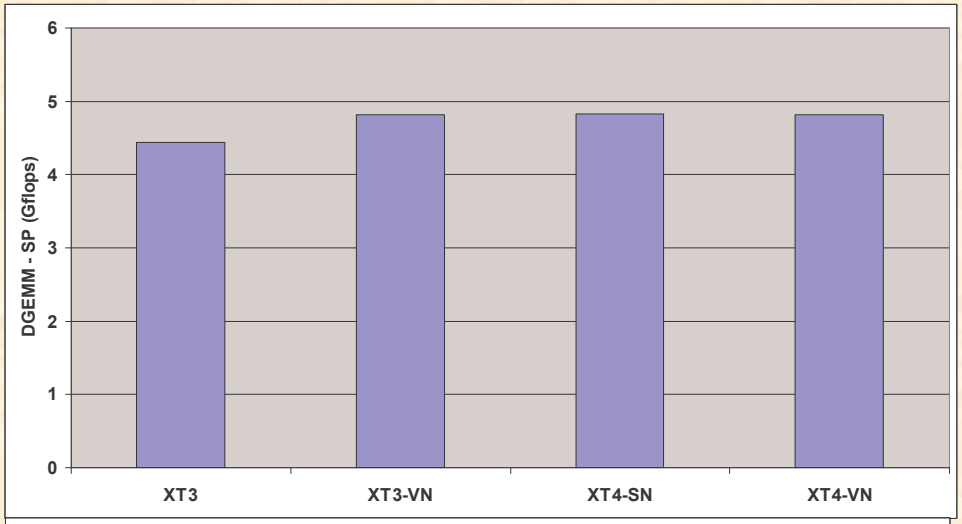
- Charts are normalized to MaxPP
  - Latencies are on par but BGL bandwidth is much lower
- Think:
  - NaturalRing ~ nearest neighbor
- Latency:
  - Hope:
    - NatRing~MinPP<AvgPP
  - Get:
    - NatRing>MaxPP
    - (NN is far away)
- Bandwidth:
  - Hope:
    - NatRing~MaxPP>AvgPP
  - Get:
    - NatRing<MinPP
    - (NN link is heavily shared)
- Typical of many Top500 systems
- But, compare to BG/L results…
- Job Layout previously identified
- Likely exacerbated by improvement in injection BW & NIC contention

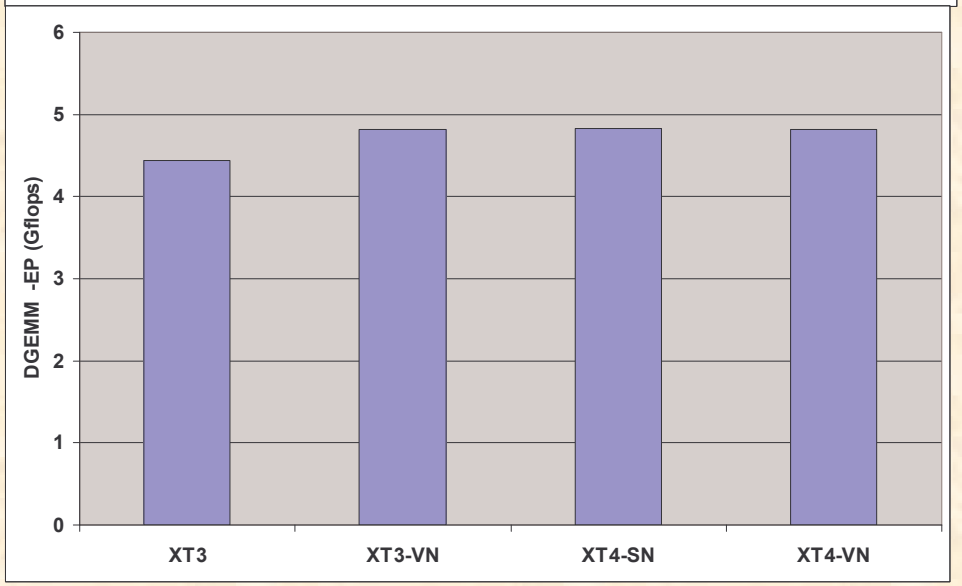




SP/EP DGEMM Summary

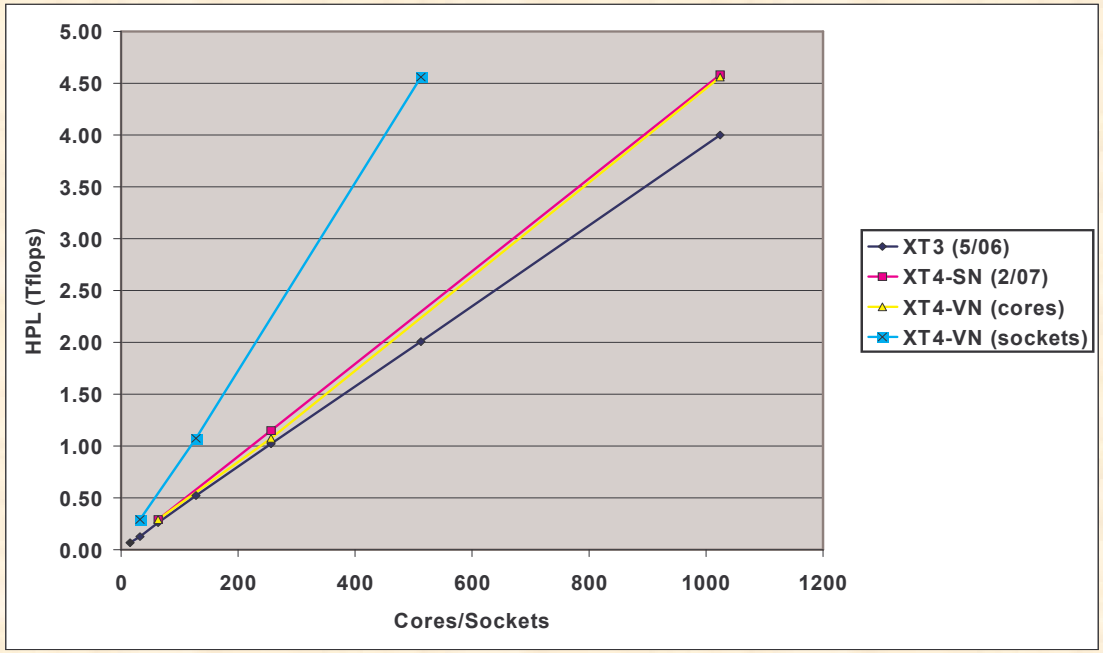
- Combined spatial and temporal locality
- Effect of slightly faster processor in XT4 visible (2.6GHz vs 2.4GHz)
- Best case result
- Performance relies heavily on BLAS library

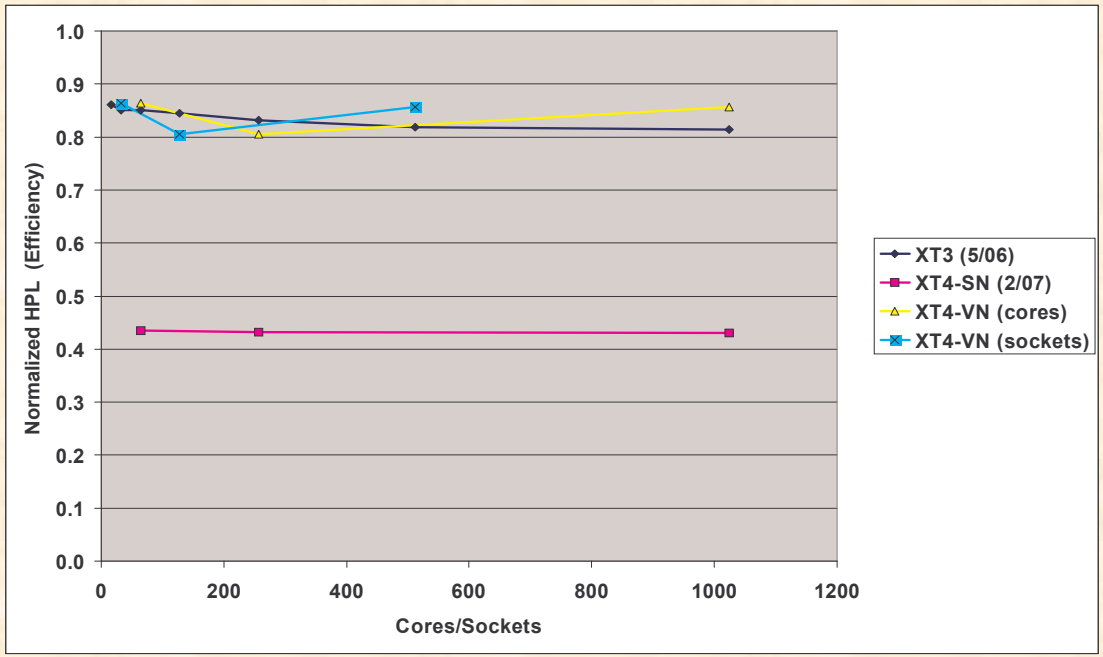




HPL Summary

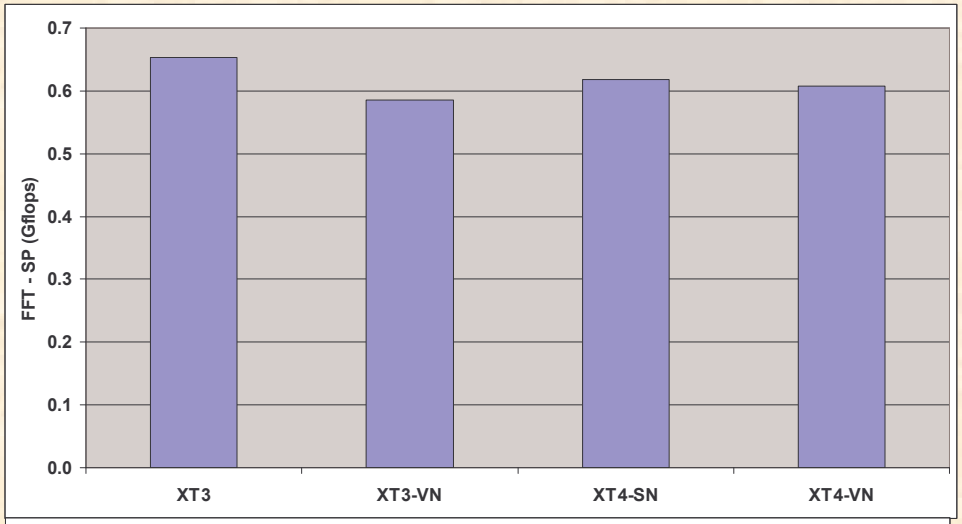
- Spatial-Temporal locality
  - Multi-core friendly
  - "Perfect" result
- Best case
  - Your code will probably never run this fast ☺
- Normalizing results
  - Pros
    - Just % of peak for HPL
    - Another scalability perspective
    - Removes "wallet-size"
  - Cons
    - Ignores "wallet-size"
  - Can normalize against several parameters. Mostly equivalent to a constant factor
  - Choose FP peak (familiar)

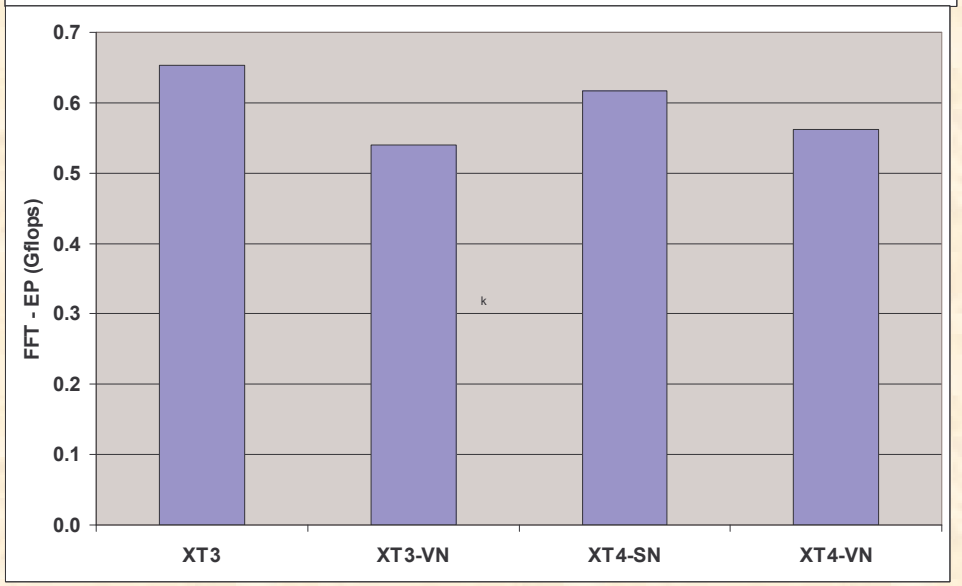




SP/EP FFT Summary

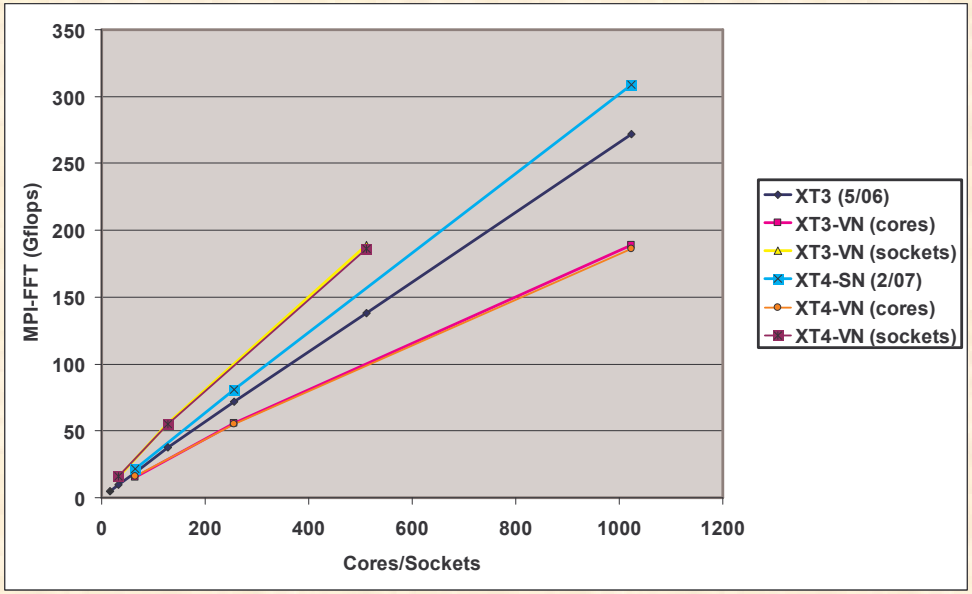
- Temporal locality emphasized
- XT3 data should be discounted because of unidentified difference in software stack and options
- SN ~10% faster than VN
- Still a strong overall win for multi-core

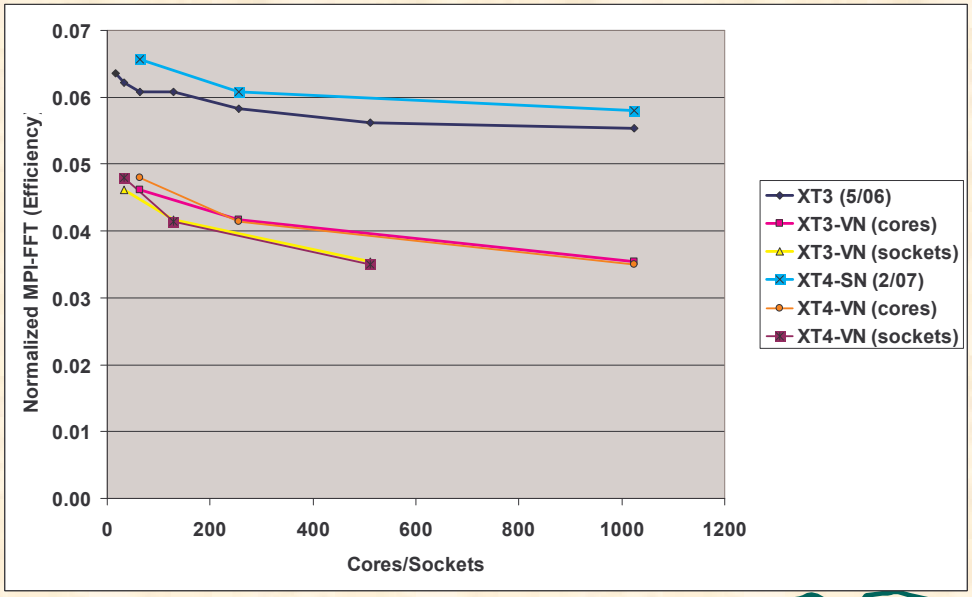




MPI FFT Summary

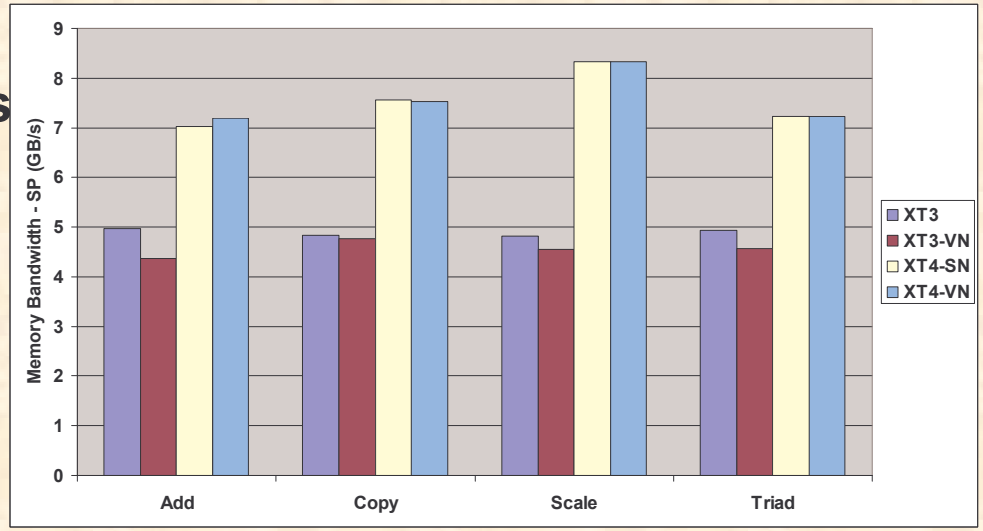
- Temporal locality emphasized
  - Not as good as HPL
  - But still performs well
- Global communication
  - Network impact
    - Higher latency
    - Lower per-core bandwidth
- Multi-core advantageous
- “SMP” Rank Reordering gives a nearly 20% improvement.

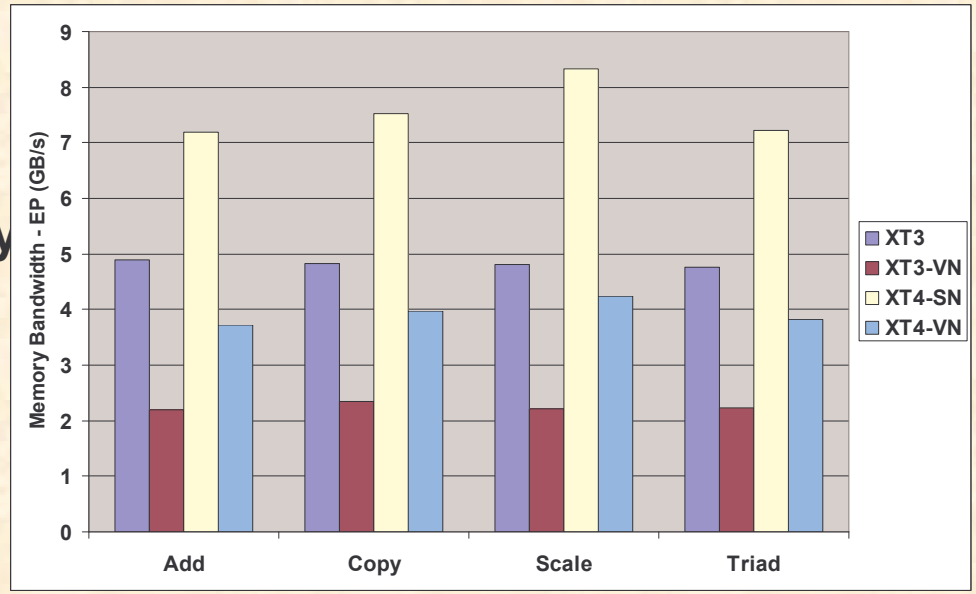




SP/EP STREAM Summary

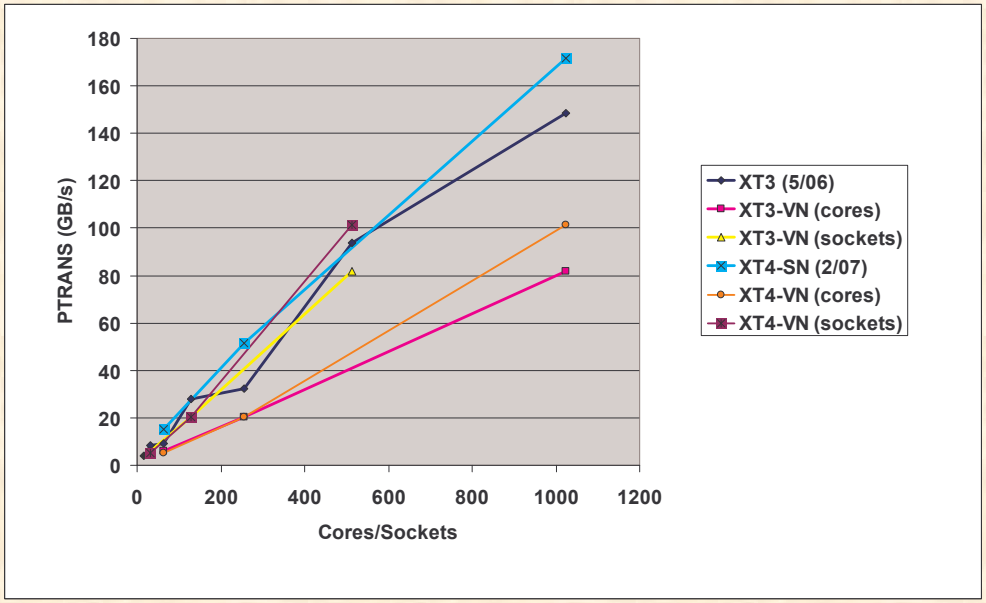
- Emphasizes spatial locality
- Faster DDR2 memory provides a distinct advantage over XT3
- Shared memory controller – dual (dueling?) cores see half bandwidth
- EP-Scale (best performer) shows one core can saturate memory interface
- Tip: best performance was achieved with less aggressive prefetch (9 on SN vs 8 on VN)
  - Prefetch bottleneck at memory controller?
- StarSTREAM improved up to 30% by adjusting several MPI environment variables...Huh???

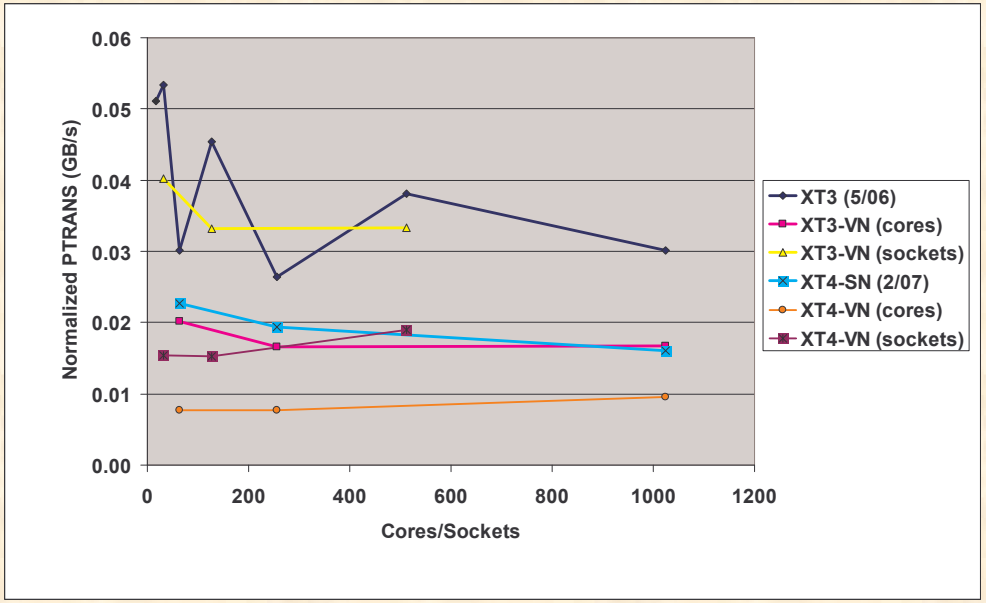




PTRANS Summary

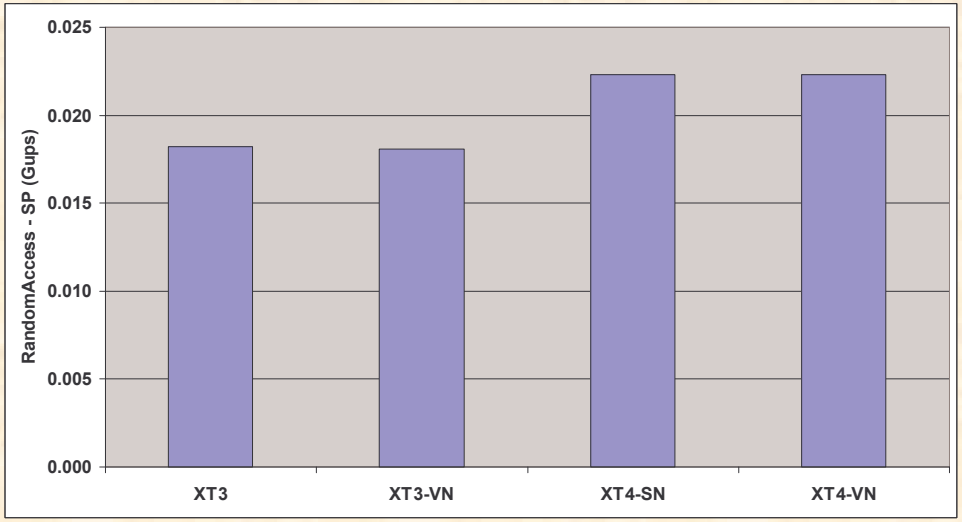
- Spatial locality
- Second core is a wash
- Impacted by network contention
- Layout Impact
  - CUG 2006 PSC paper showed 10% impact on XT3
  - XT4 has higher injection BW so increased:
    - Link contention
    - NIC contention
- Improved by nearly 50% with MPICH_PTL_MATCH_OFF... We have several theories.

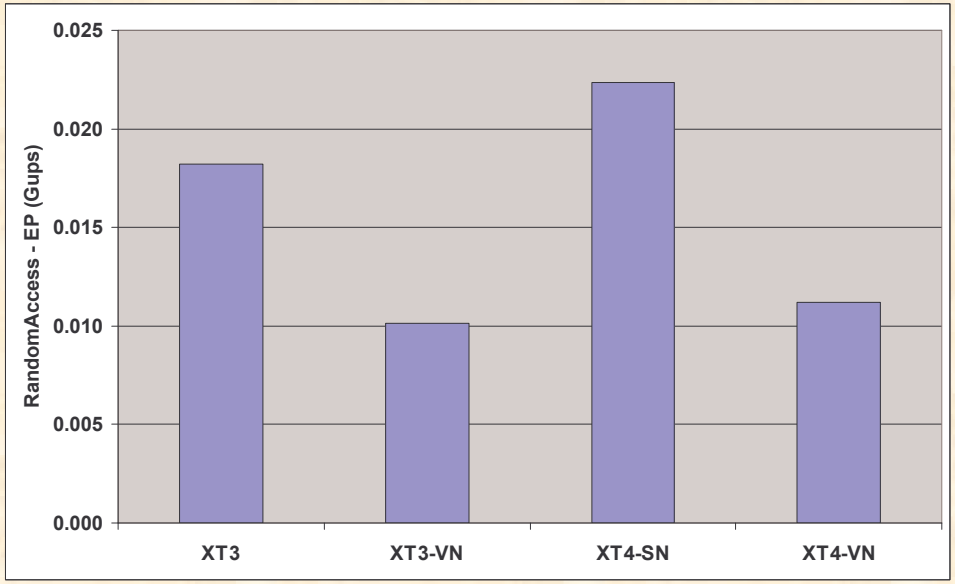




SP/EP RandomAccess Summary

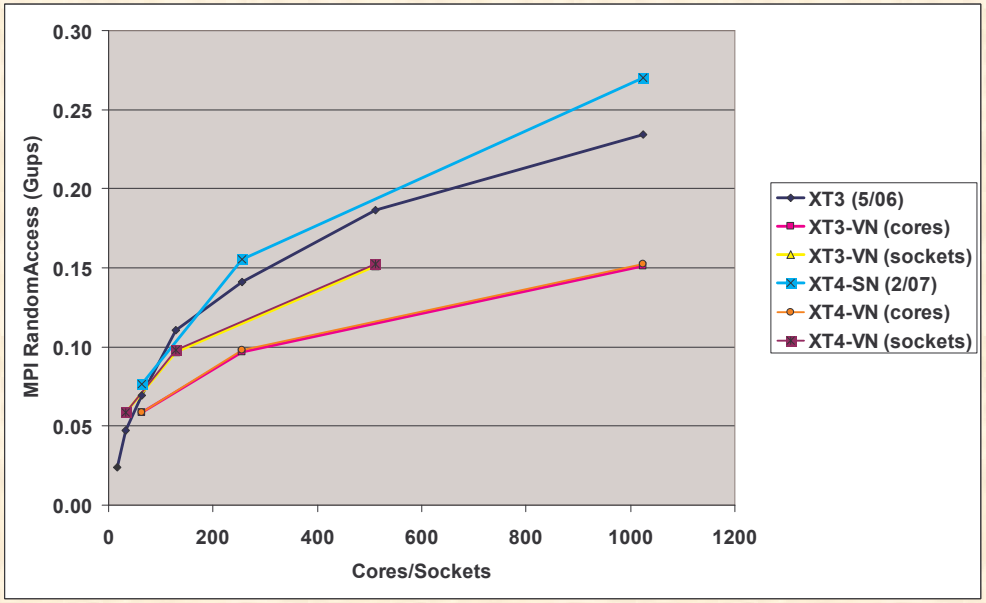
- Low locality
  - If the cache hit rate isn't zero, the test case isn't big enough ☺
- Note impact of DDR2 memory
- Comparing SP vs EP
  - Memory system is the bottleneck
  - Engaging second core provides no benefit
  - Good multi-core code should not look like this
  - But most code does ☺

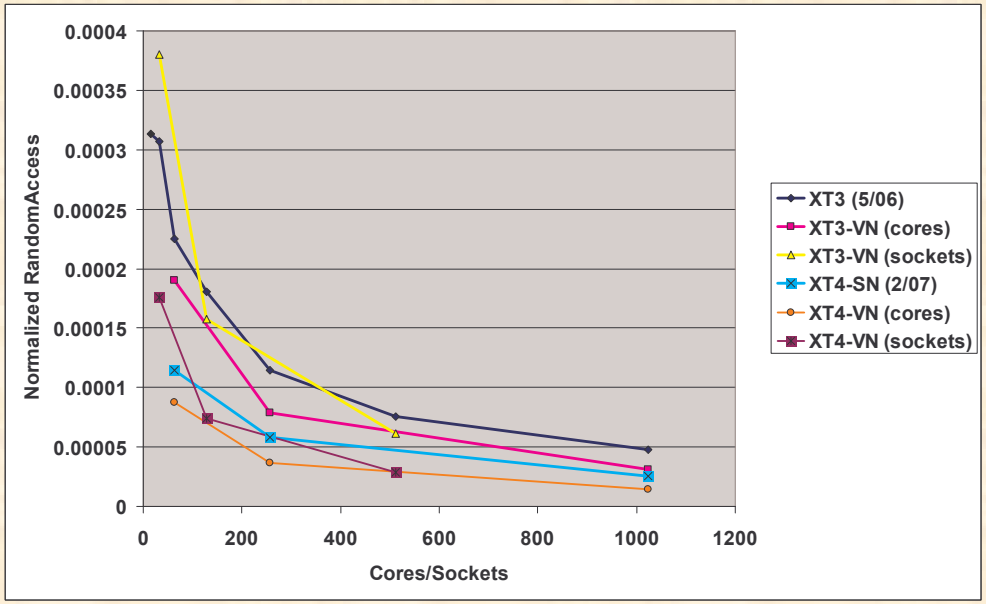




MPI RandomAccess Summary

- Low locality
- Note scaling
- Adding second core *reduces* overall performance
- Cores "share" NIC and memory controller
- Network Bandwidth plays little role
- Network Latency imposes 25% penalty





A Few Multi-core Tips

- *Slightly* reducing the prefetch distance improved dual core performance for the spatial-locality kernel
- Setting MPICH_RANK_REORDER_METHOD=1 (SMP mode) gave mixed results...just try it.
- Additionally setting MPICH_PTL_MATCH_OFF=1 helped several benchmarks
  - Improved Latency and Bandwidth*
  - Helped large messages in Ptrans
- Importance of locality
  - Kernels exhibiting high temporal locality will do well on multi-core processors
    - Adding spatial locality will further improve performance
  - Kernels exhibiting low temporal locality will do poorly on multi-core processors
    - Adding spatial locality will have little impact

Multi-core Summary