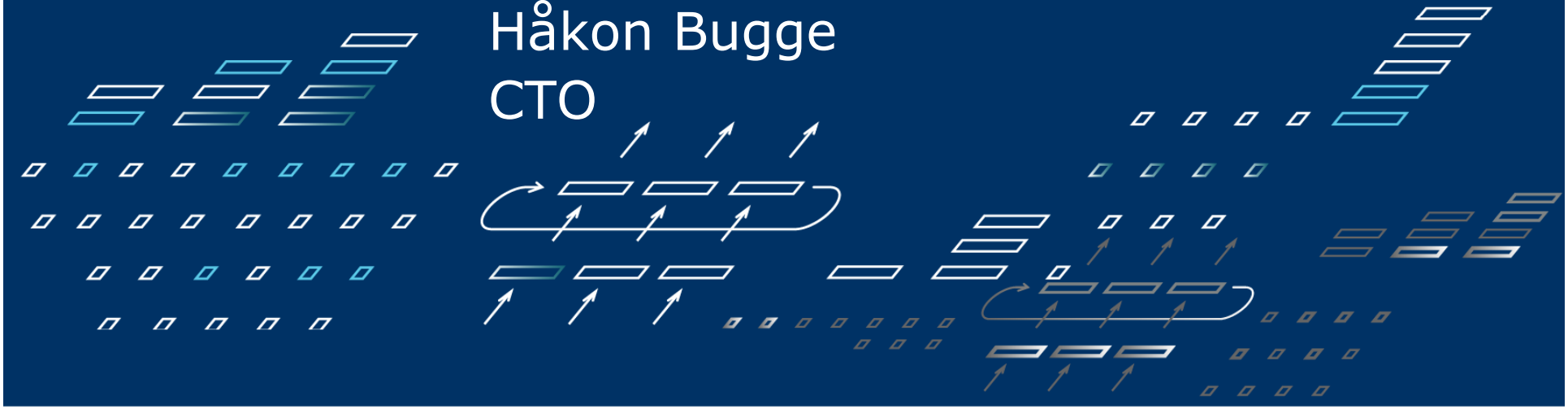
MPI in a Multicore environment

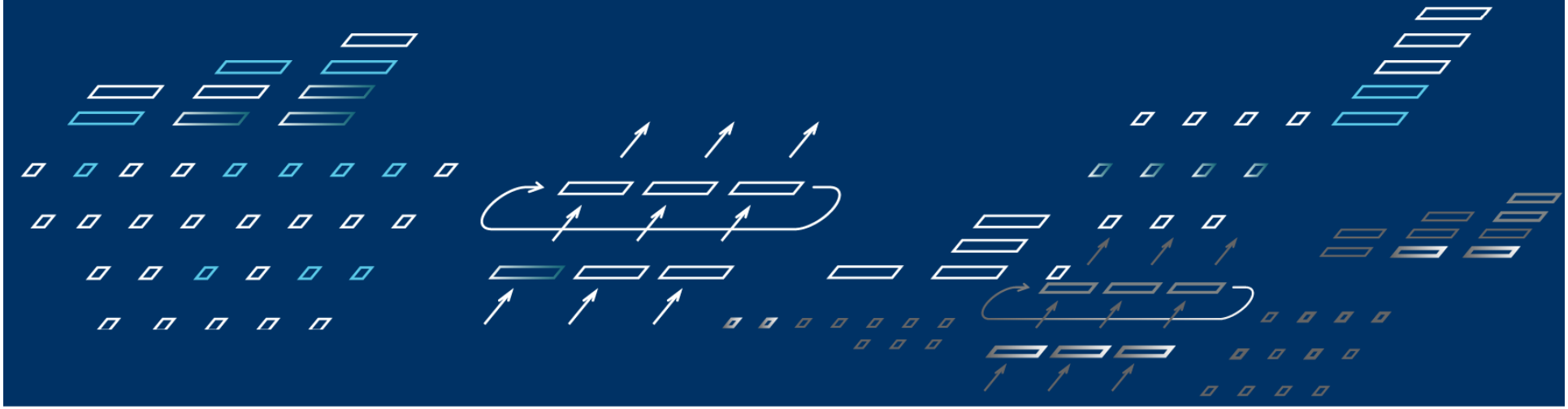
Håkon Bugge
CTO



Outline

- Introduction to Scali
- Trends and numbers
- Multicore affinity in Scali MPI Connect
- Q & A

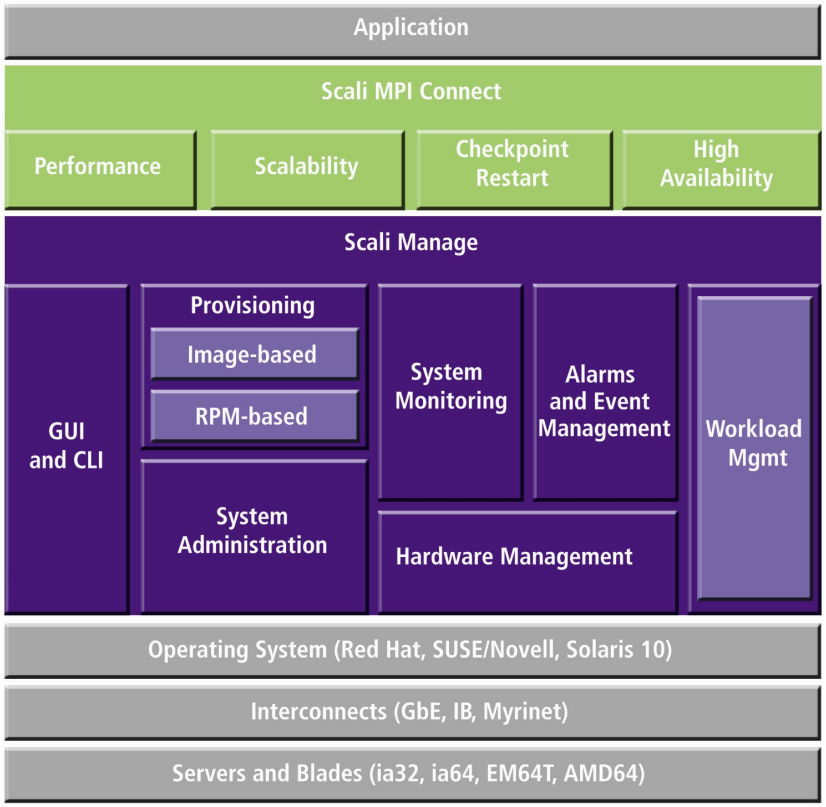
Introduction to Scali



Scali – Higher Performance Computing

- Founded in 1997 with a focus on Linux clustering and MPI
- R&D in Oslo, Norway, Support and QA in Nagpur, India
- Headquartered in Marlboro, MA
- Customers in over 25 countries

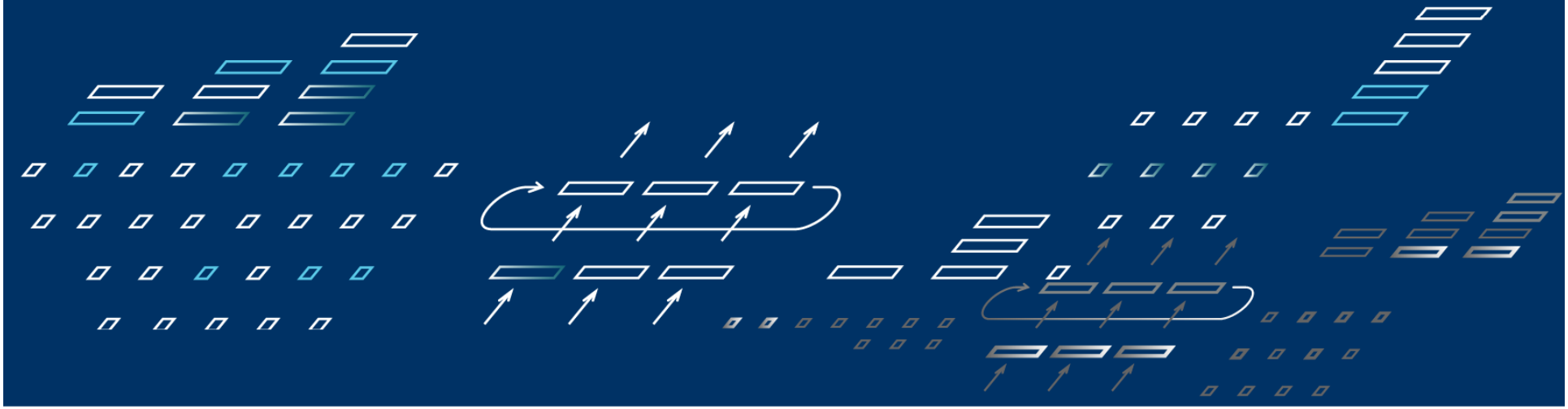
Scali Functionality



Scali Ecosystem

SCALI

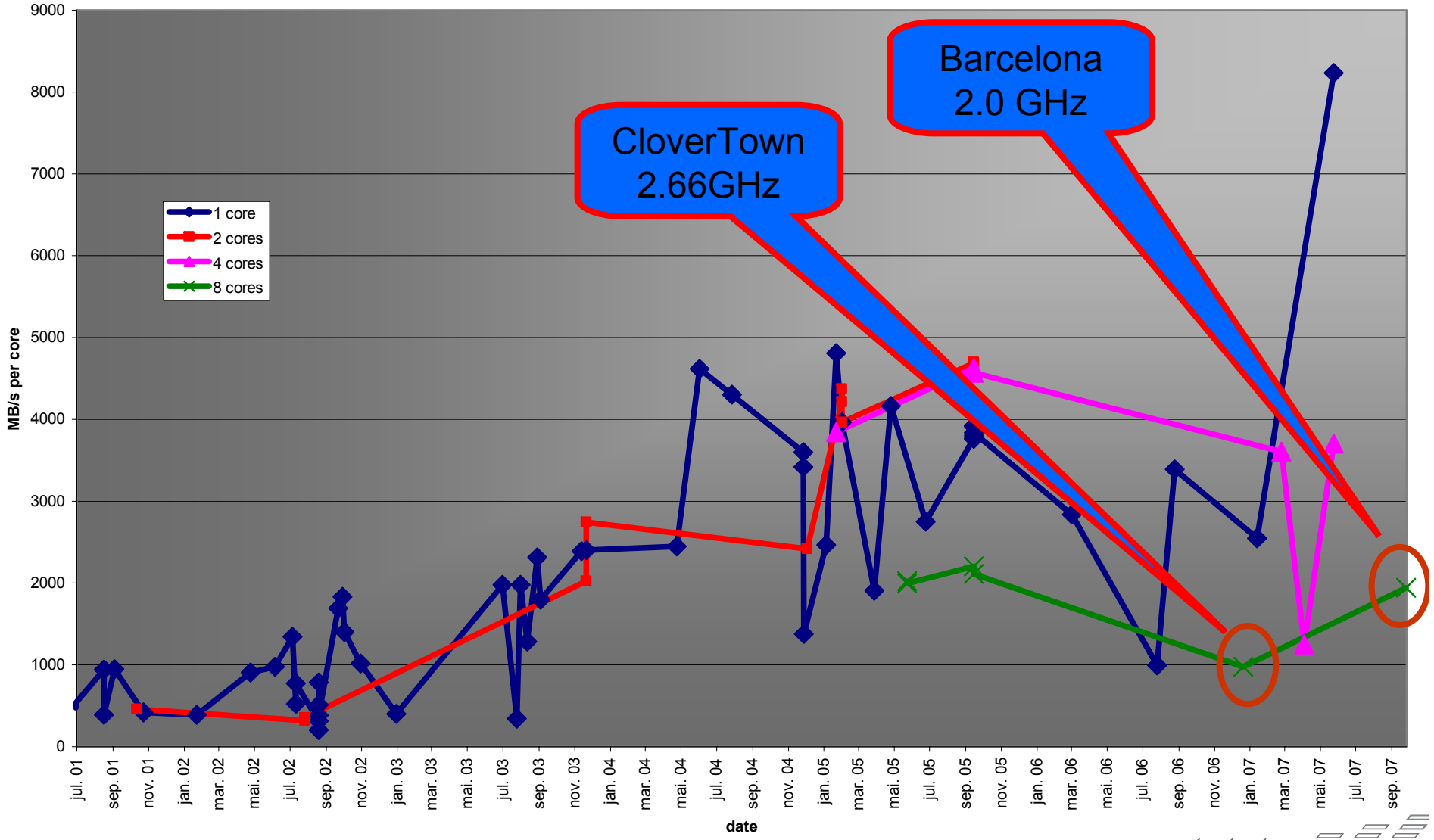
Trends and real numbers



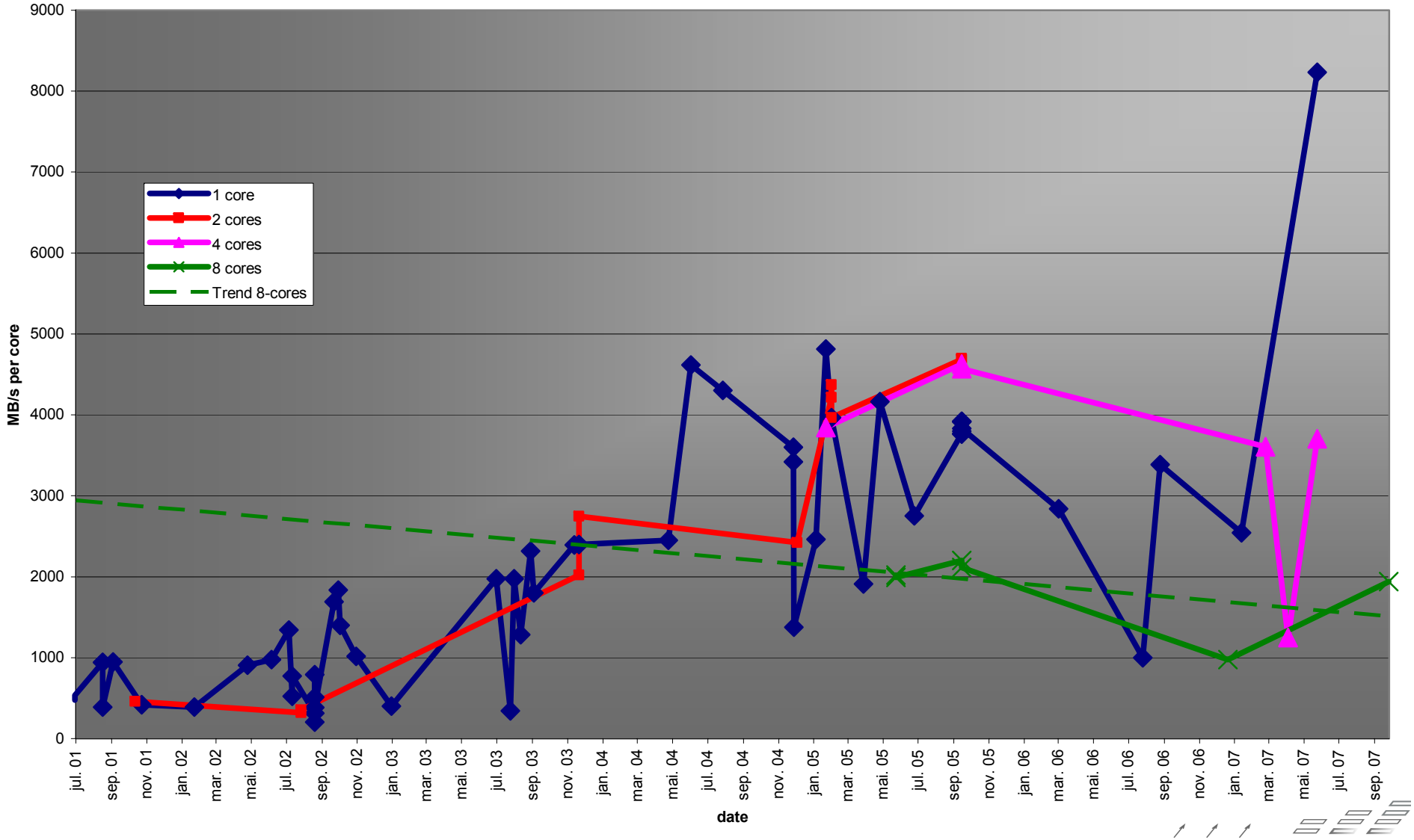
Moore's law and related

- Transistor budget increases with ~55% per year
  - Smaller geometries and faster switching implied *faster* processors in the past
  - Today, it implies *more* processors, i.e., cores
- Memory bandwidth has historically had a much slower growth, around 11% per year
- Processor vendors and systems vendors are great in telling you everything that grows proportional to Moore's law
- But, today, memory bandwidth per core grows inversely proporsal to Moore's law!

McCalpin Stream Benchmark



McCalpin Stream Benchmark

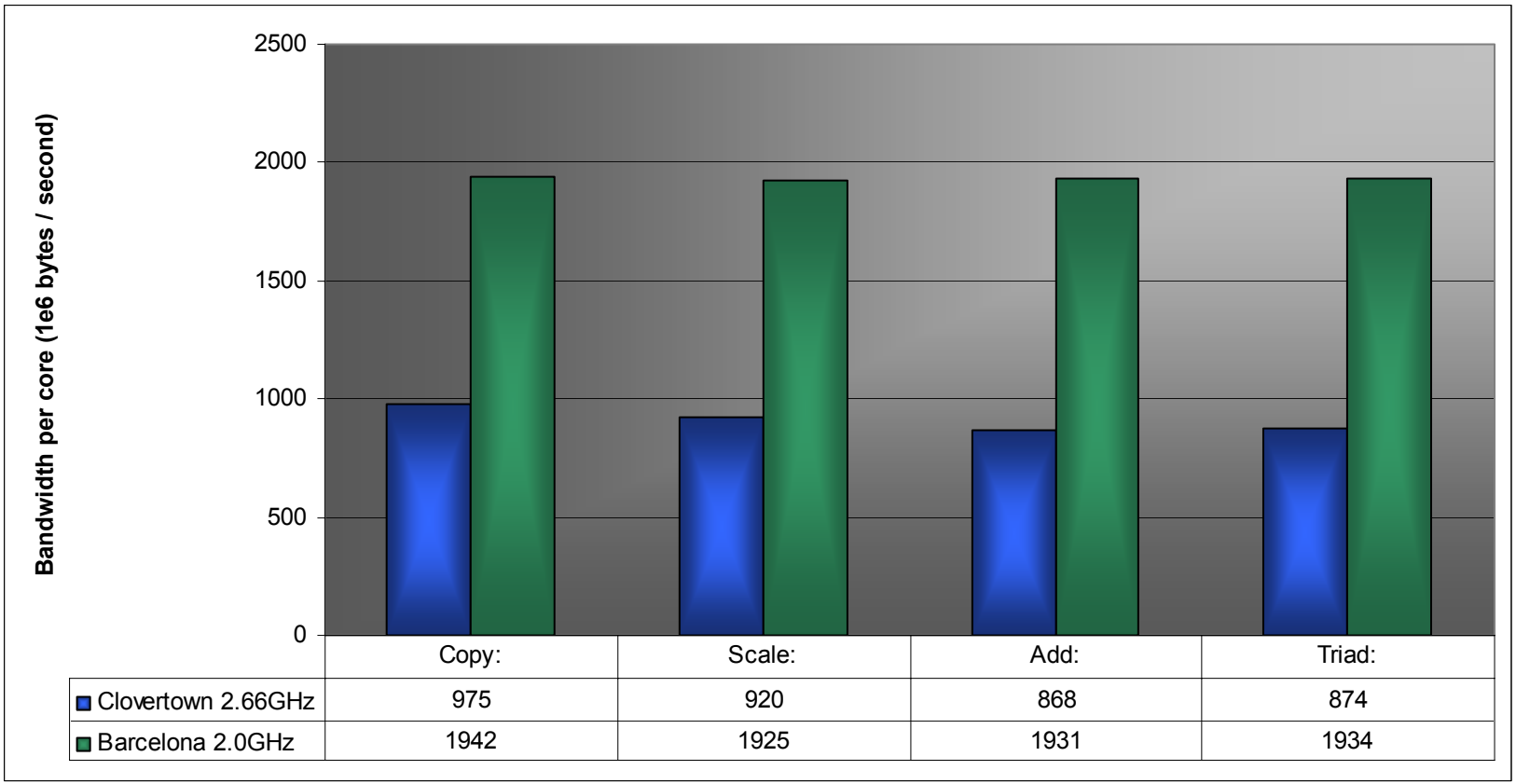


Multi-Core systems: The new challenge

- Until now: All focus has been on processor, memory, and interconnect characteristics
- The arrival of multi-core systems impose a paradigm shift:
  - Many more cores to share resources
  - Compute resources (cores) are available in excess
  - Memory bandwidth must be shared by many cores
  - Interconnect bandwidth and message rate must be shared by many cores
  - Message passing performance inside a single node really starts to matter
- Measuring said characteristics using a single core per node may be very misleading

Memory bandwidth

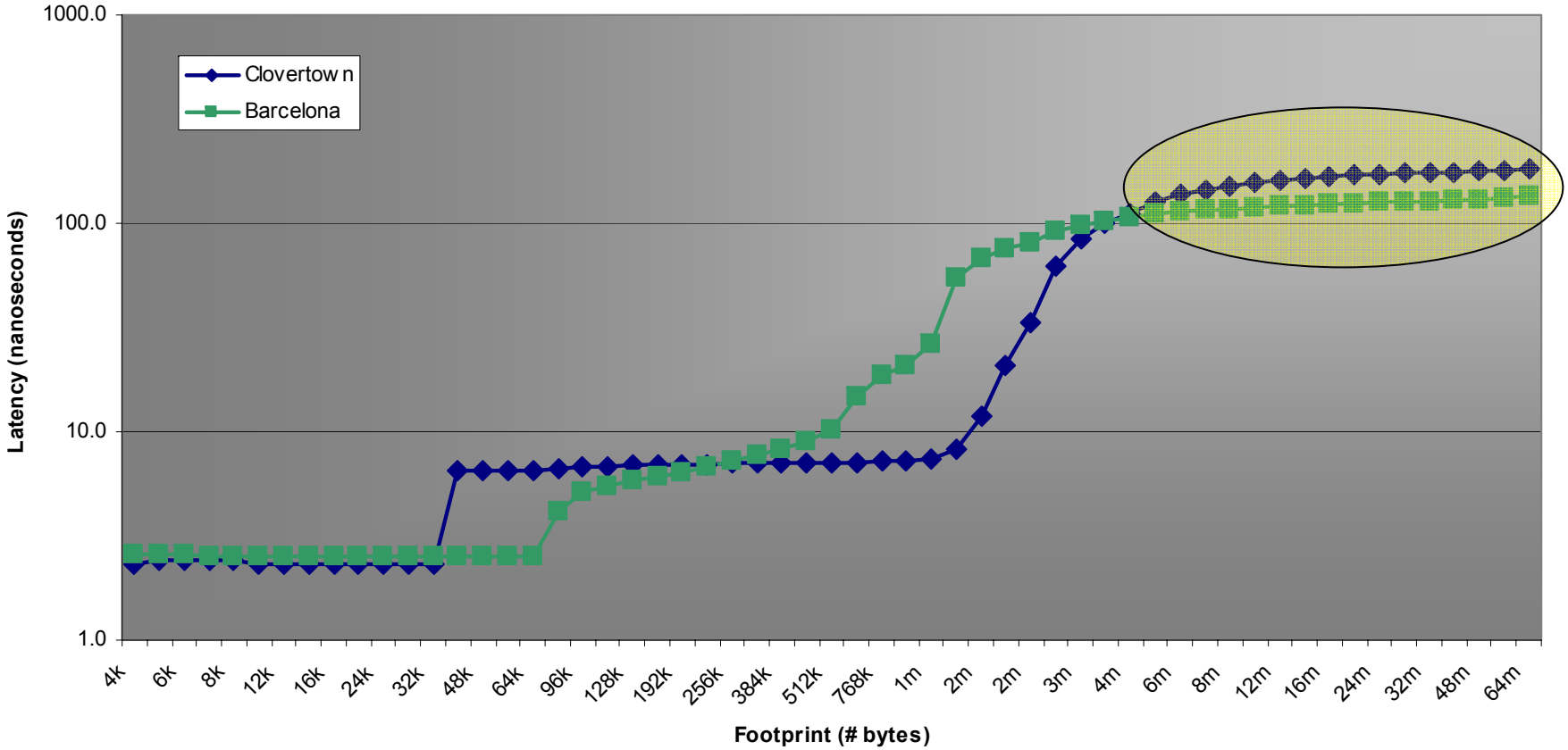
- What is the available bandwidth per core when <u>all</u> cores request maximum bandwidth?



	Copy:	Scale:	Add:	Triad:
Clovertown 2.66GHz	975	920	868	874
Barcelona 2.0GHz	1942	1925	1931	1934

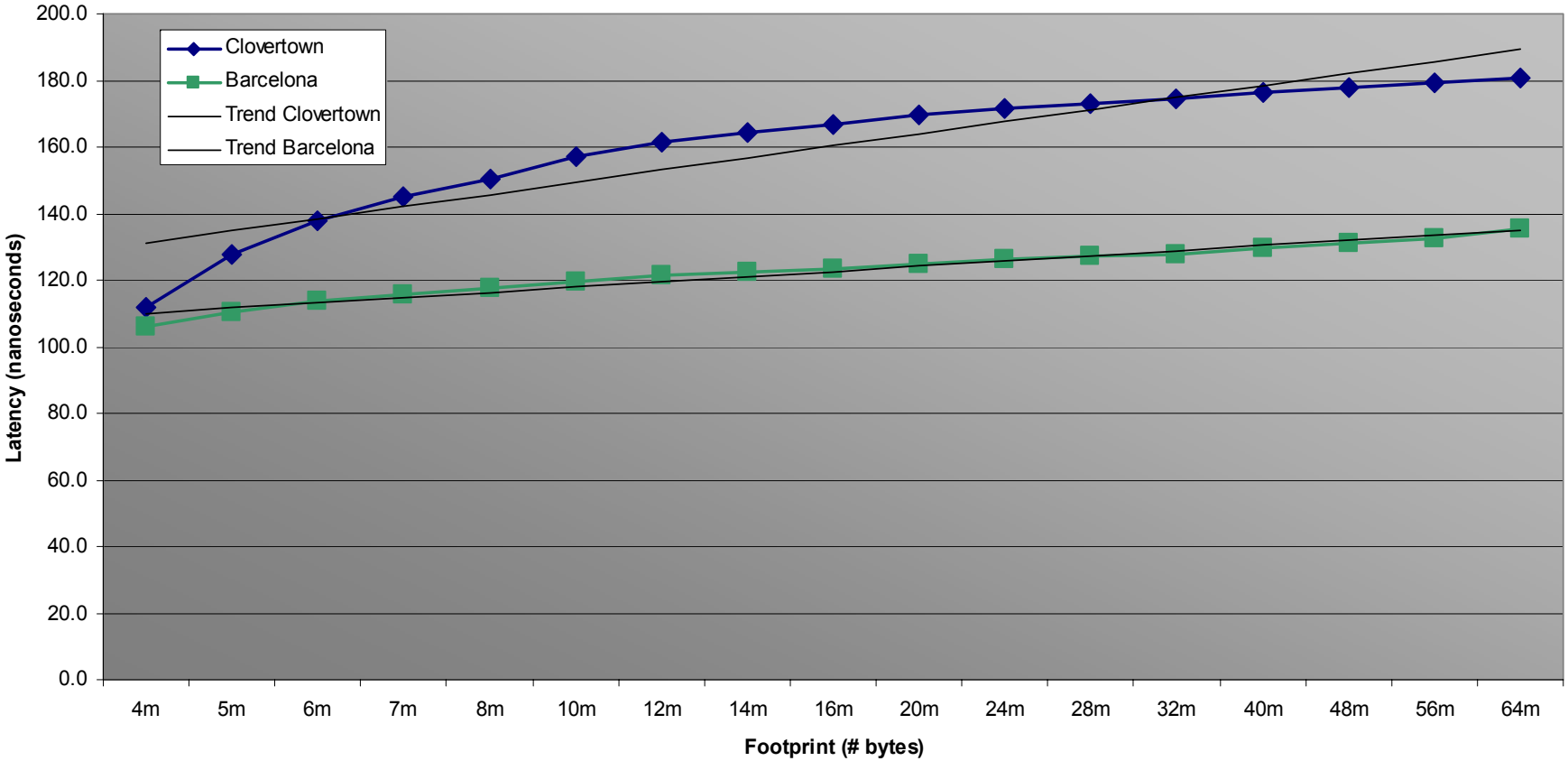
Memory latency

- What is the memory latency when all cores are active?



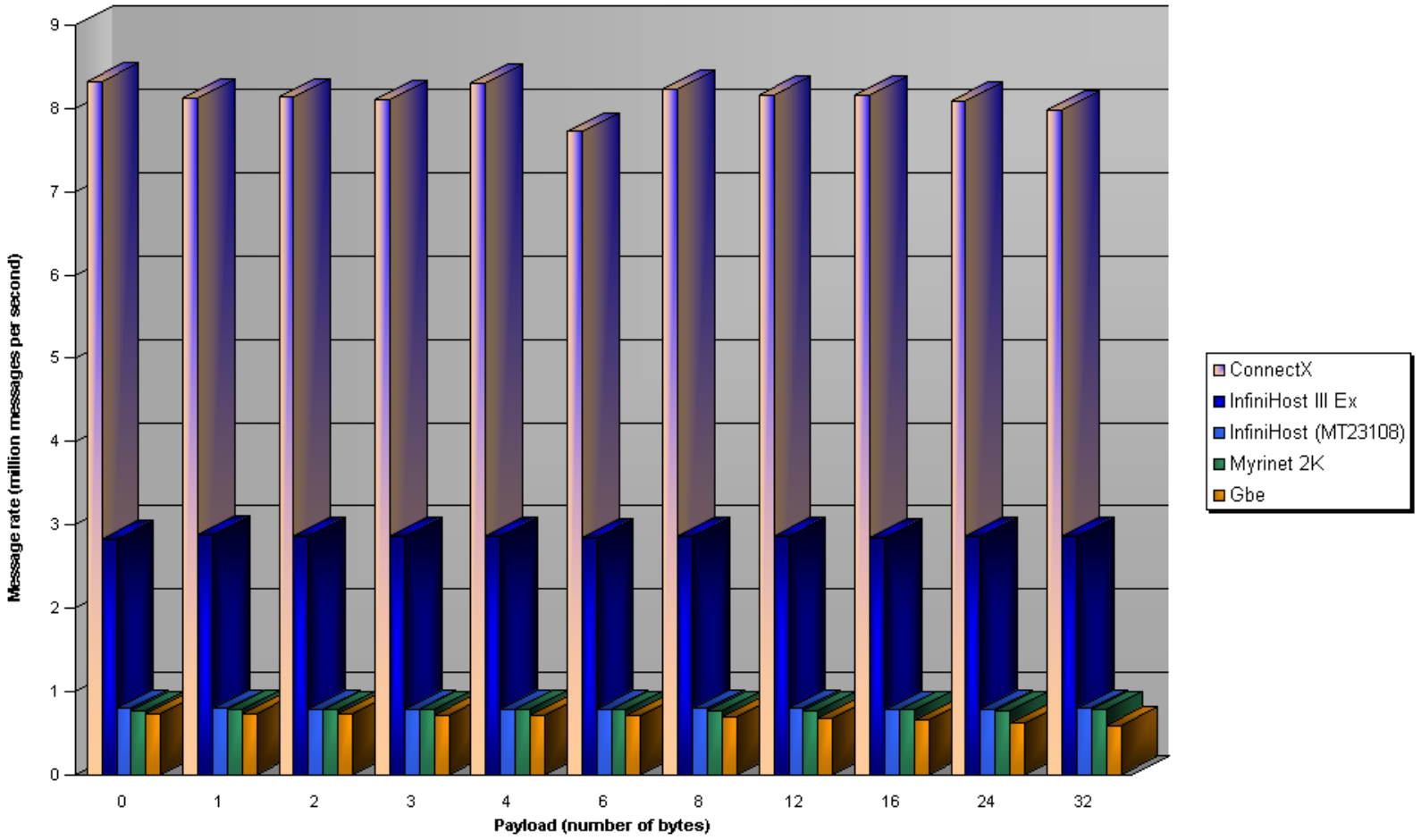
Memory latency cont'd

- Beyond the cache sizes:



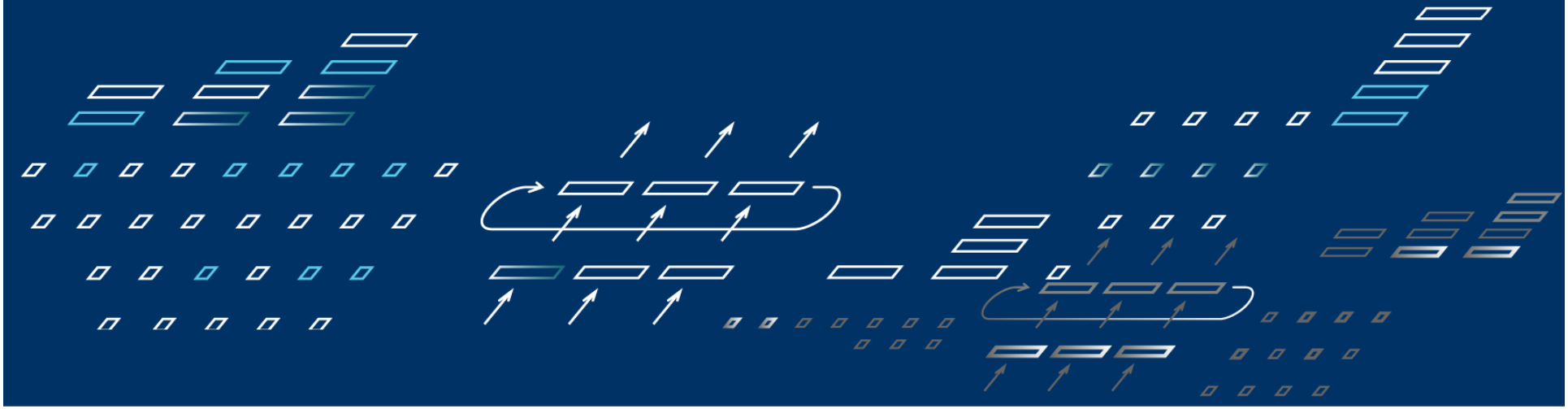
Interconnect performance

- Its more to it than just latency and bandwidth between a single process per node:
  - Use all cores per node
  - Use more than two nodes
  - Here is an example of a message-rate benchmark, where all processes send and receive a message from all other processes. This uses 6 nodes and 4 processes per node



SCALI

Multicore affinity in Scali MPI Connect



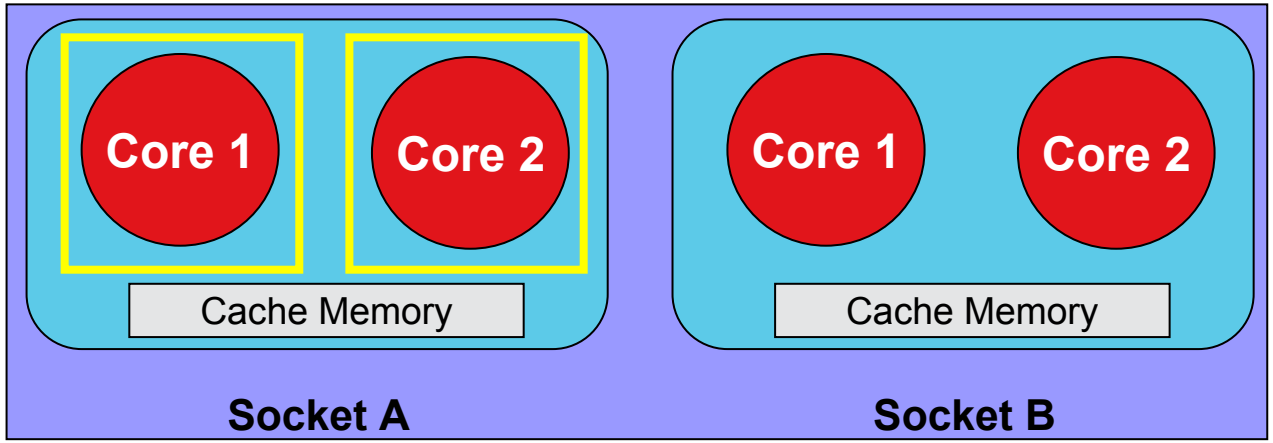
Multicore Affinity Terminology

- MxNxO – M jobs, N sockets per node, O cores per socket
- Scali MPI Connect Core Affinity specification:
  - *Manual*
  - *Automatic*
    - *Bandwidth* - i.e., use as many sockets as possible
    - *Latency* – i.e., use as few sockets as possible
  - Resolution
    - *Thread* - bind the process to the smallest possible resolution, i.e., a particular hyperthread or SMT-thread
    - *Core*: - bind to all threads constituting a core
    - Socket: - bind to all cores constituting a socket
  - and, it is dynamic!

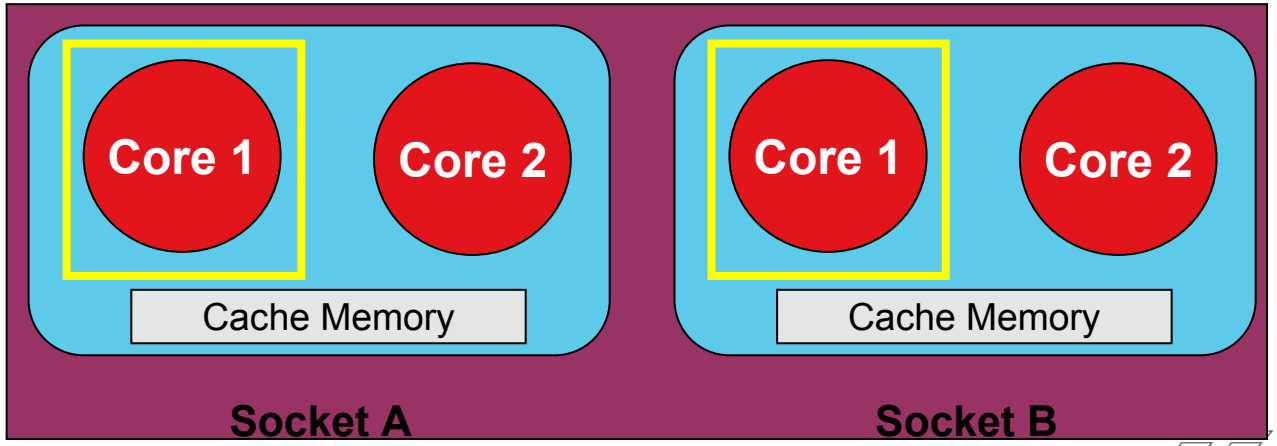
Multicore Affinity Optimizations

• Scali MPI Connect intelligently distributes processes across cores within a system based on policies delivering up to **25% performance improvement**

Latency driven -
Both processes in one socket, one in each core

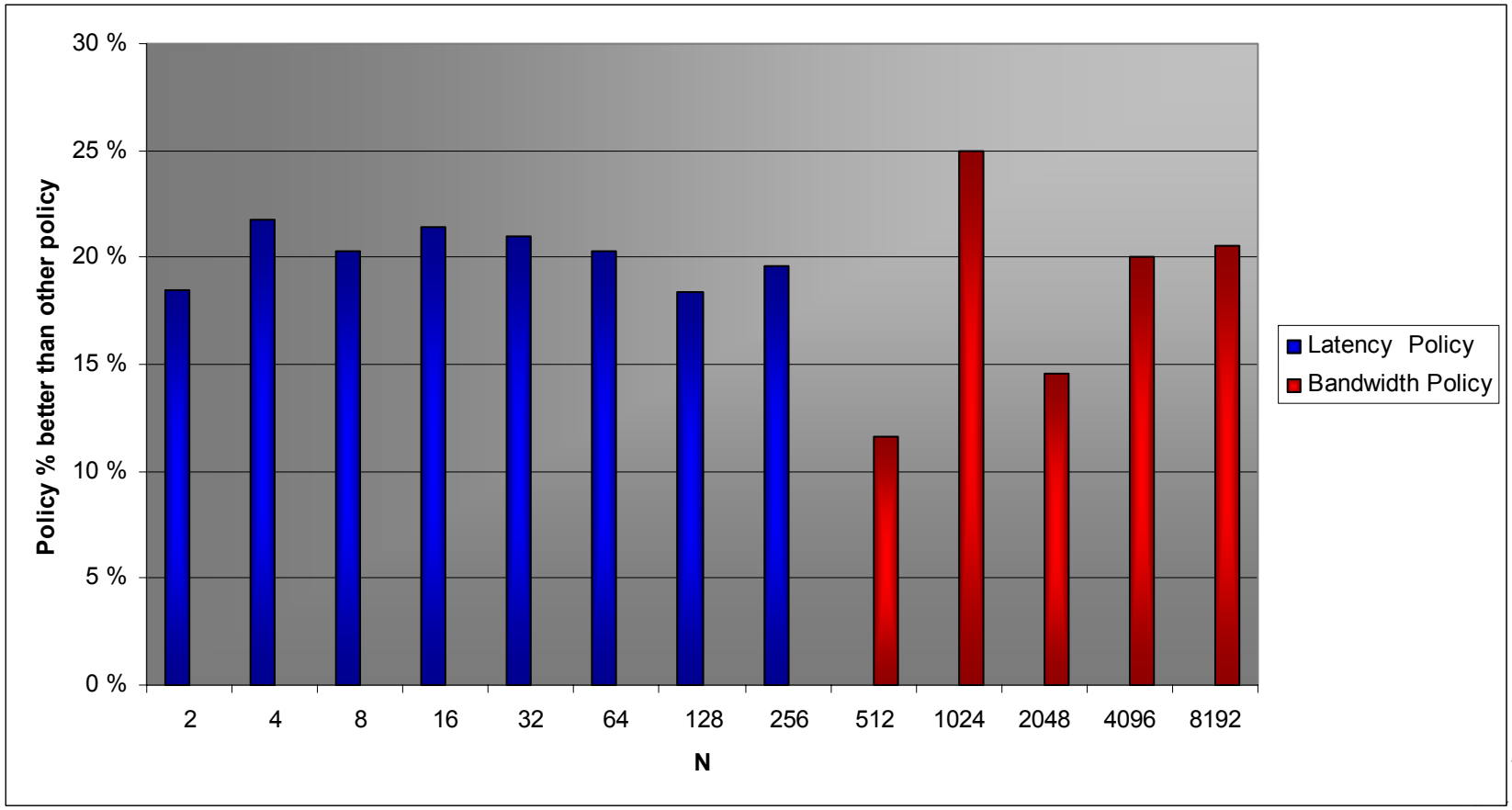


Bandwidth driven -
One process in each CPU / socket



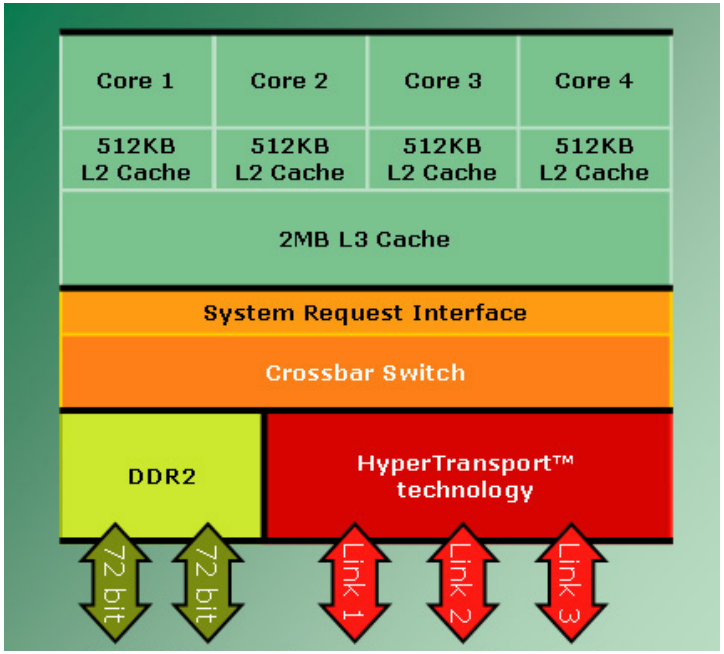
Multicore Affinity Impact

- Impact of policy matters
- Resuls from *Halo* benchmark executed on 2.66GHz CloverTown, 4 processes on a node

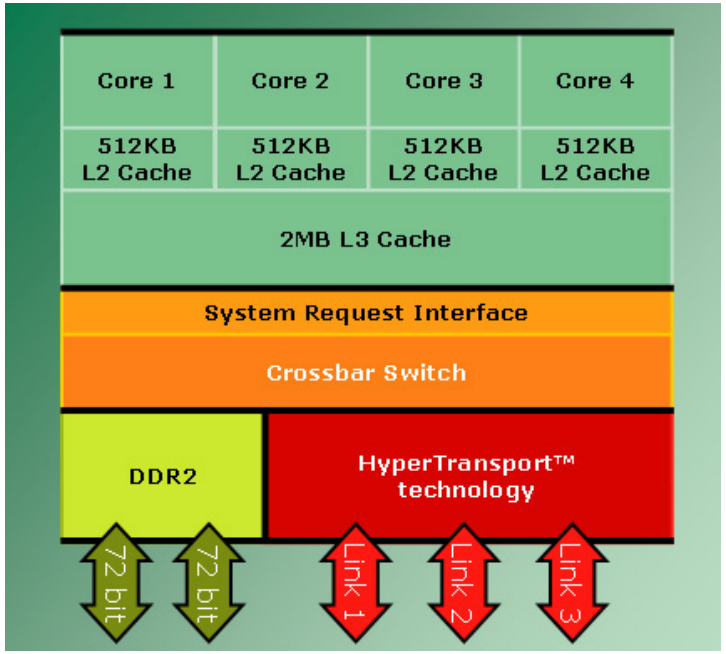


How to map jobs to a cluster

- Assume you have 16 nodes, 8 cores per node, in total 128 cores
- What possibilities exist?



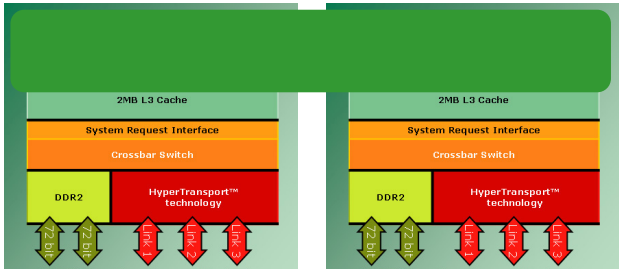
Socket #0



Socket #1

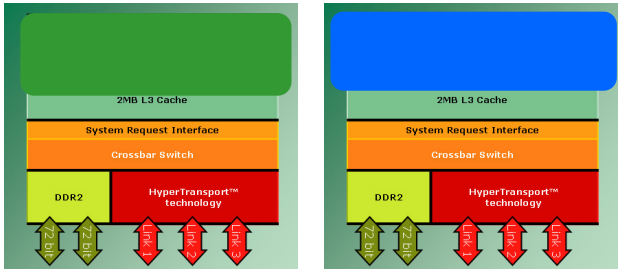
Mapping jobs to a node

1 x N x 8



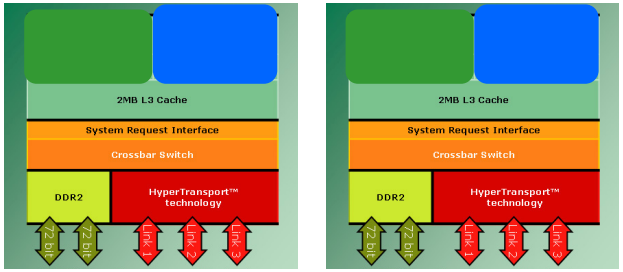
Socket #0 Socket #1

2 x N x 4 (latency binding)



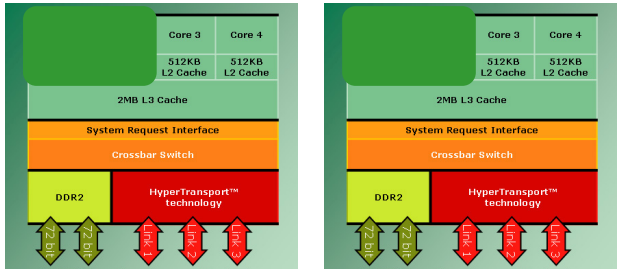
Socket #0 Socket #1

2 x N x 4 (bandwidth binding)



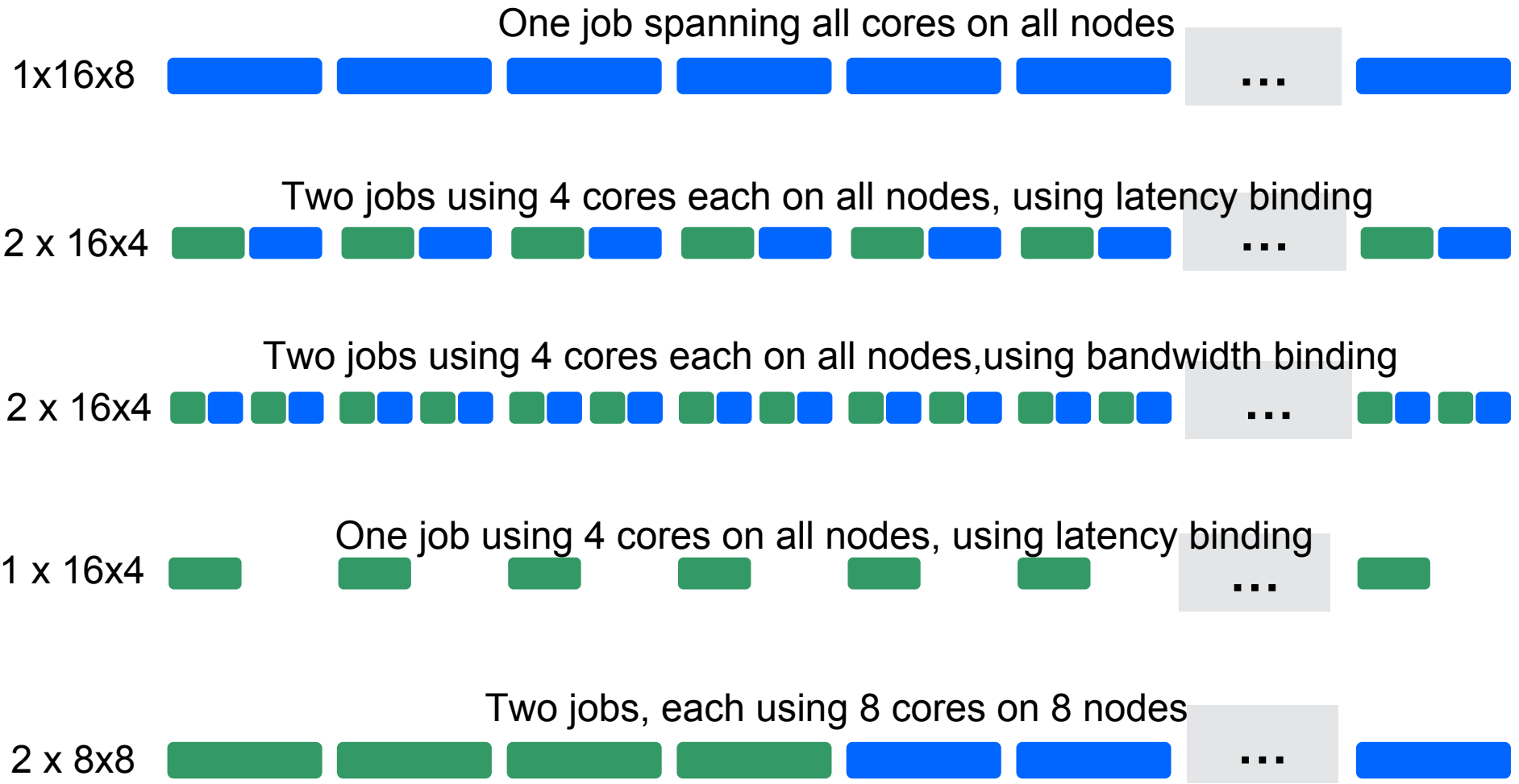
Socket #0 Socket #1

1 x N x 4 (bandwidth binding)

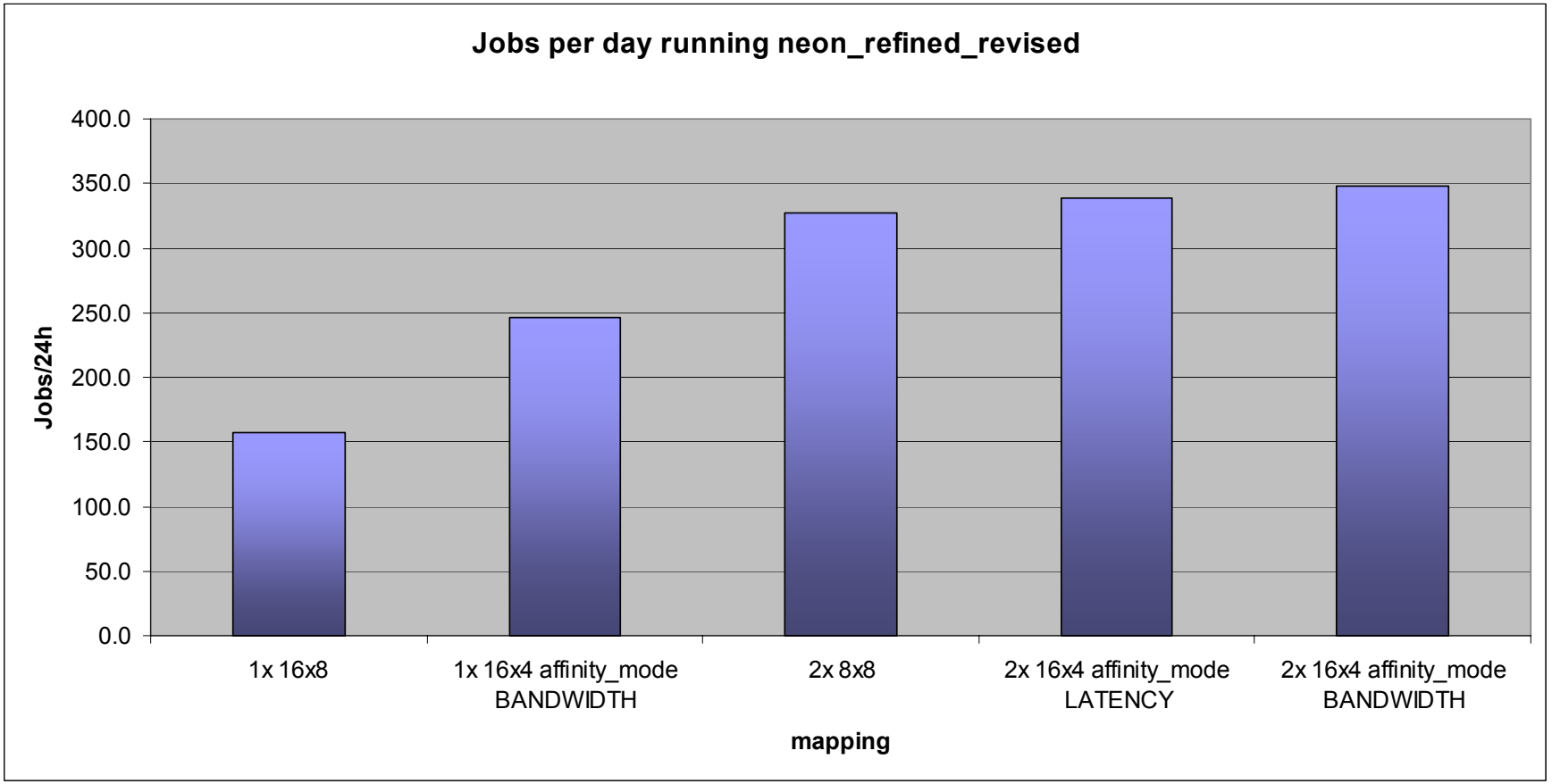


Socket #0 Socket #1

Mapping jobs to a 16-node cluster



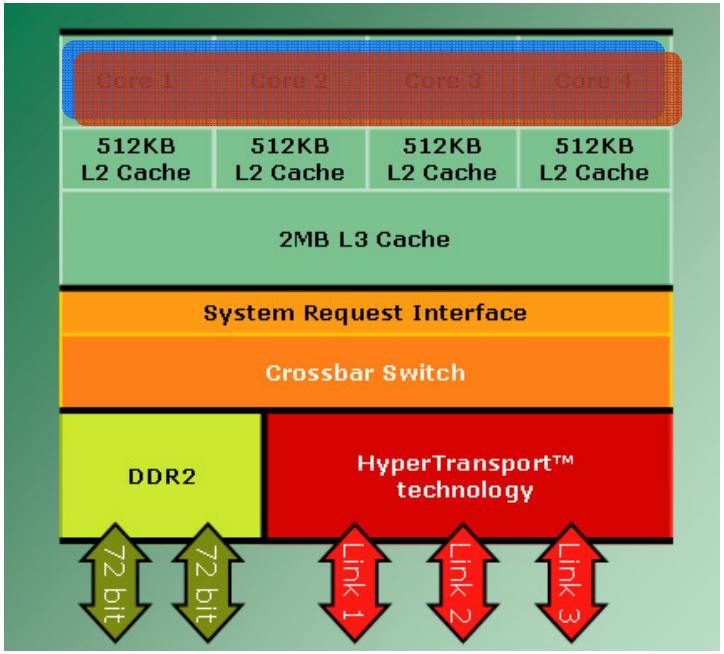
Impact on Jobs / Day



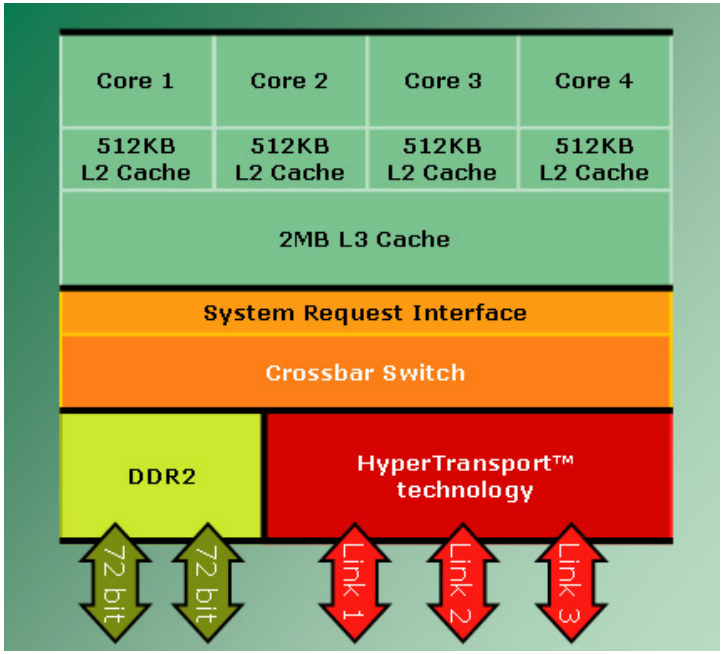
- Running multiple jobs at the same time on systems connected by IB is 2x faster than running one job on all cores

Common problems with affinity binding

- Affinity implementation is not dynamic but uses a fixed scheme:
  - Two jobs, four processes per job bound using a latency policy



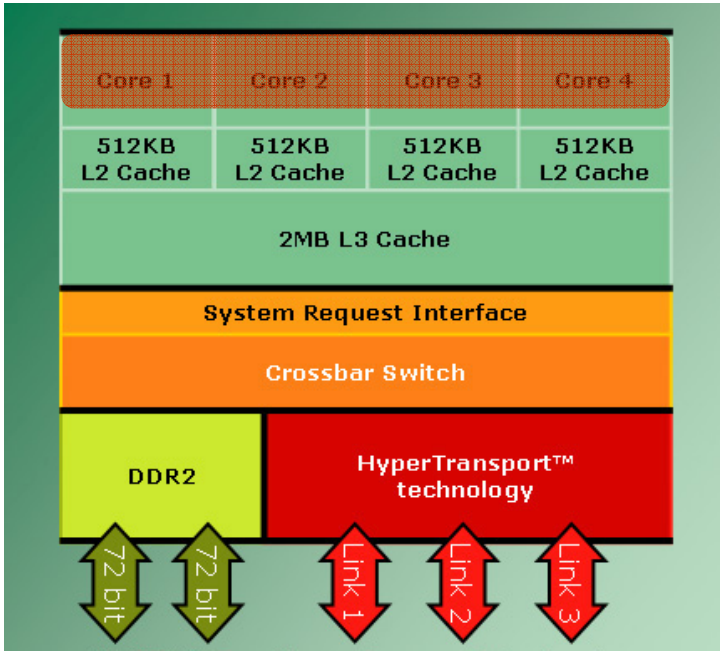
Socket #0



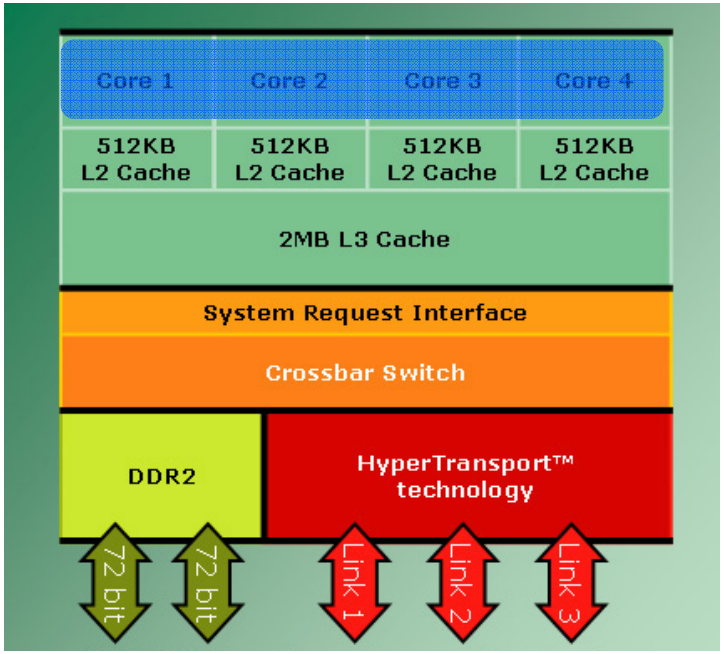
Socket #1

Common problems with affinity binding

- Scali MPI Connect uses dynamic information:
  - Two jobs, four processes per job bound using a latency policy



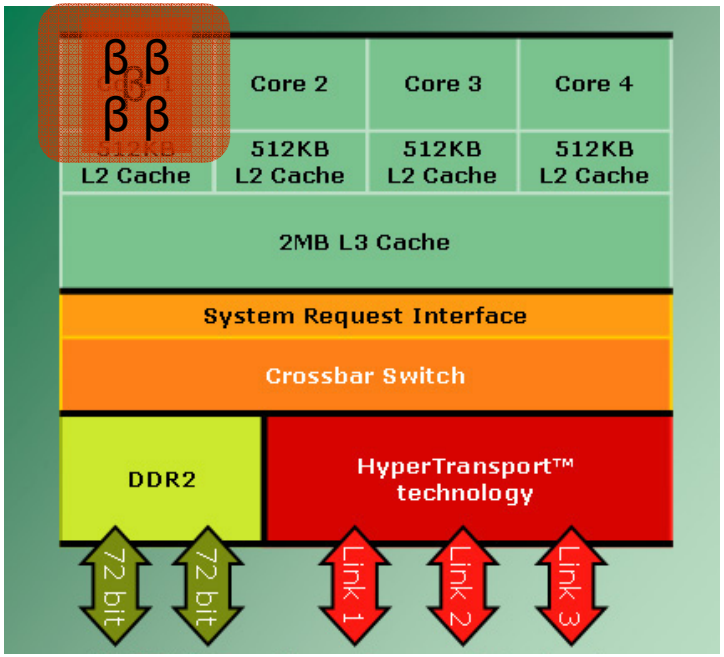
Socket #0



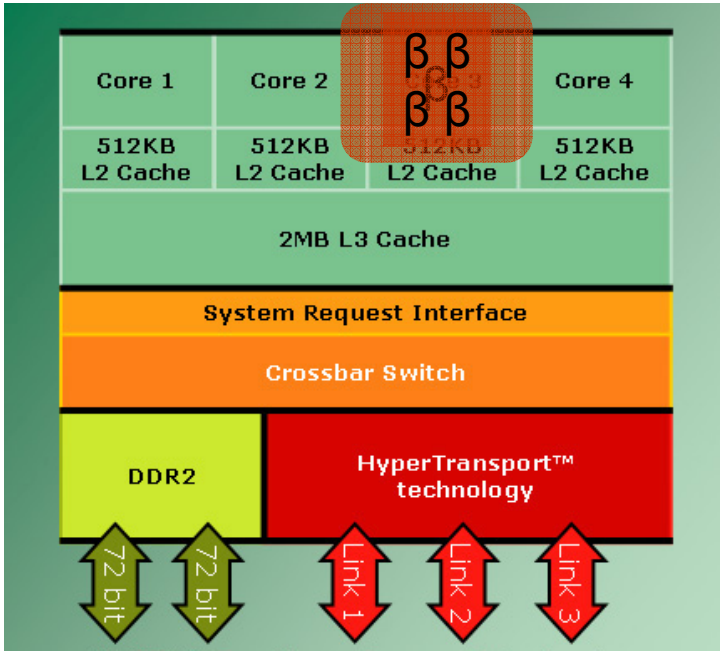
Socket #1

Common problems with affinity binding

- Hybrid models (OpenMP + MPI)
- Two MPI processes, bandwidth policy, **core** resolution, OMP_NUM_THREADS=4



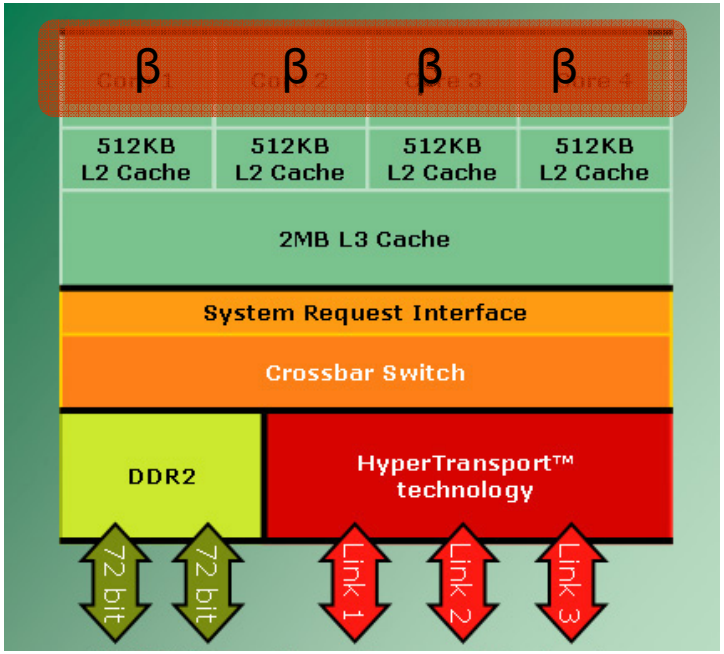
Socket #0



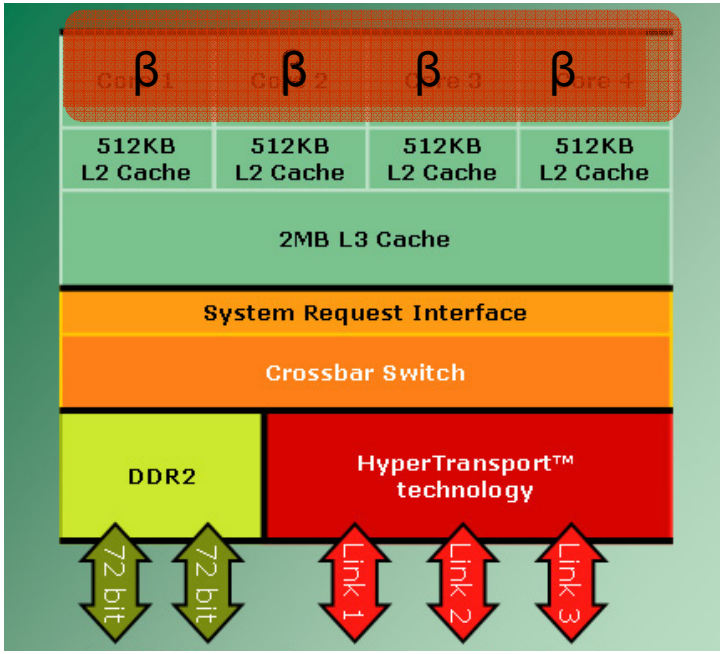
Socket #1

Common problems with affinity binding

- Hybrid models (OpenMP + MPI)
- Two MPI processes, bandwidth policy, **socket** resolution, OMP_NUM_THREADS=4



Socket #0

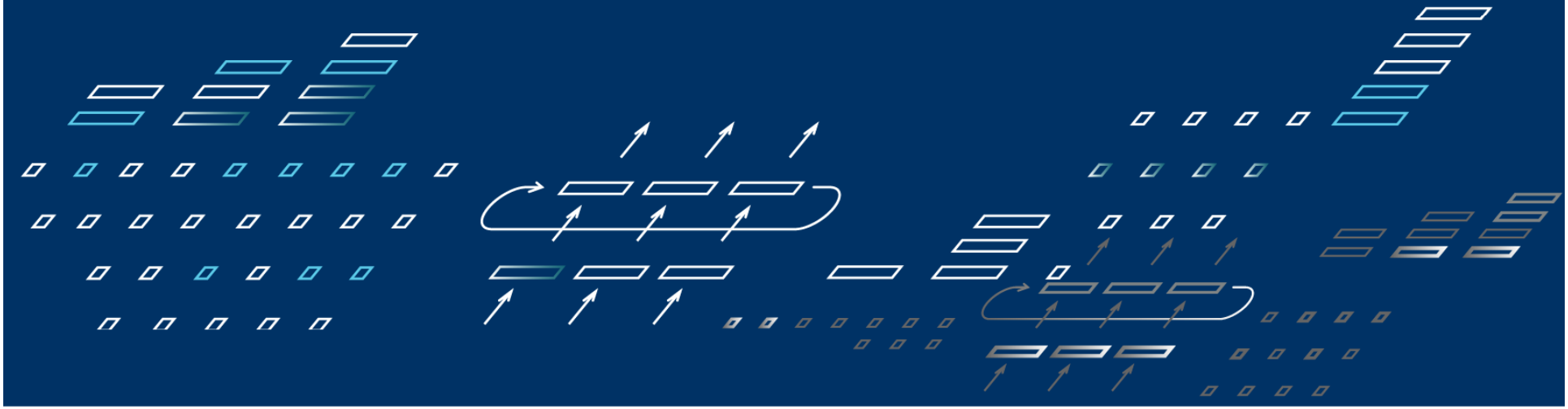


Socket #1

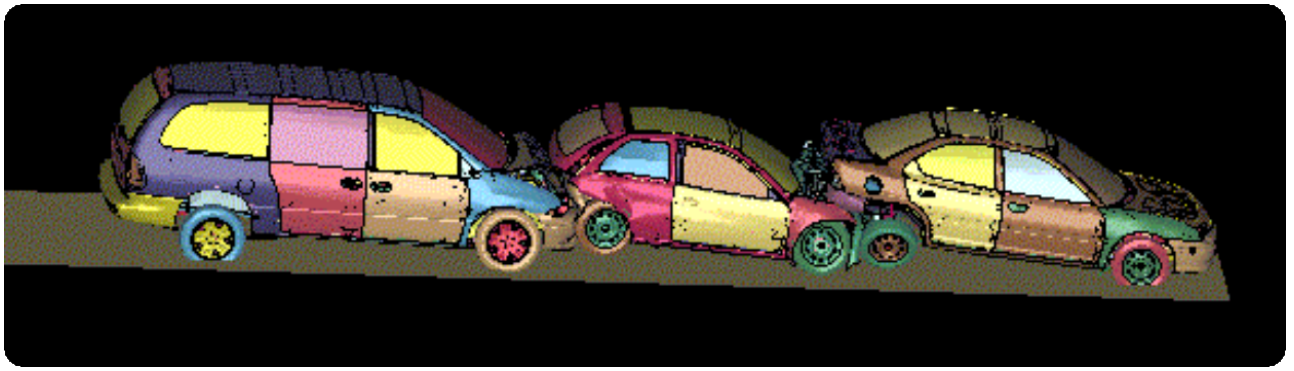
Affinity: Which layer?

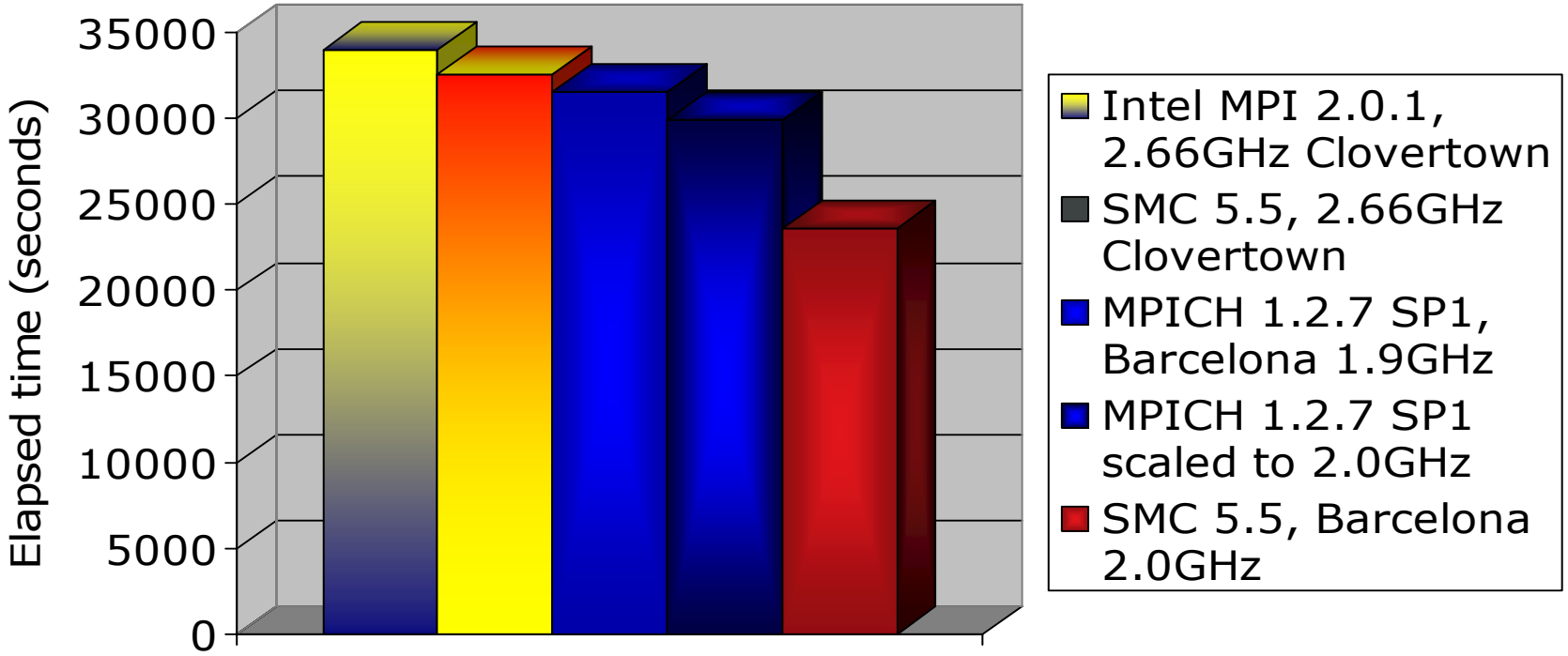
- Scheduling system
  - ”Master of ceremony” related to resource allocation
  - No detailed knowledge of cores, sockets, caches etc.
  - Multiple processes and jobs aware
- Compiler run-time library
  - Not aware of overall resources (e.g., multiple processes)
  - Aware of threads (hybrid model)
- Application
  - Yes, its true. Shouldn’t be bothered with this detal level
- MPI
  - Detailed knowledge of cores, sockets, caches etc.
  - Aware of multiple processes on a node
  - Must be aware of multiple jobs on a node (i.e., dynamic decision)

Thank You

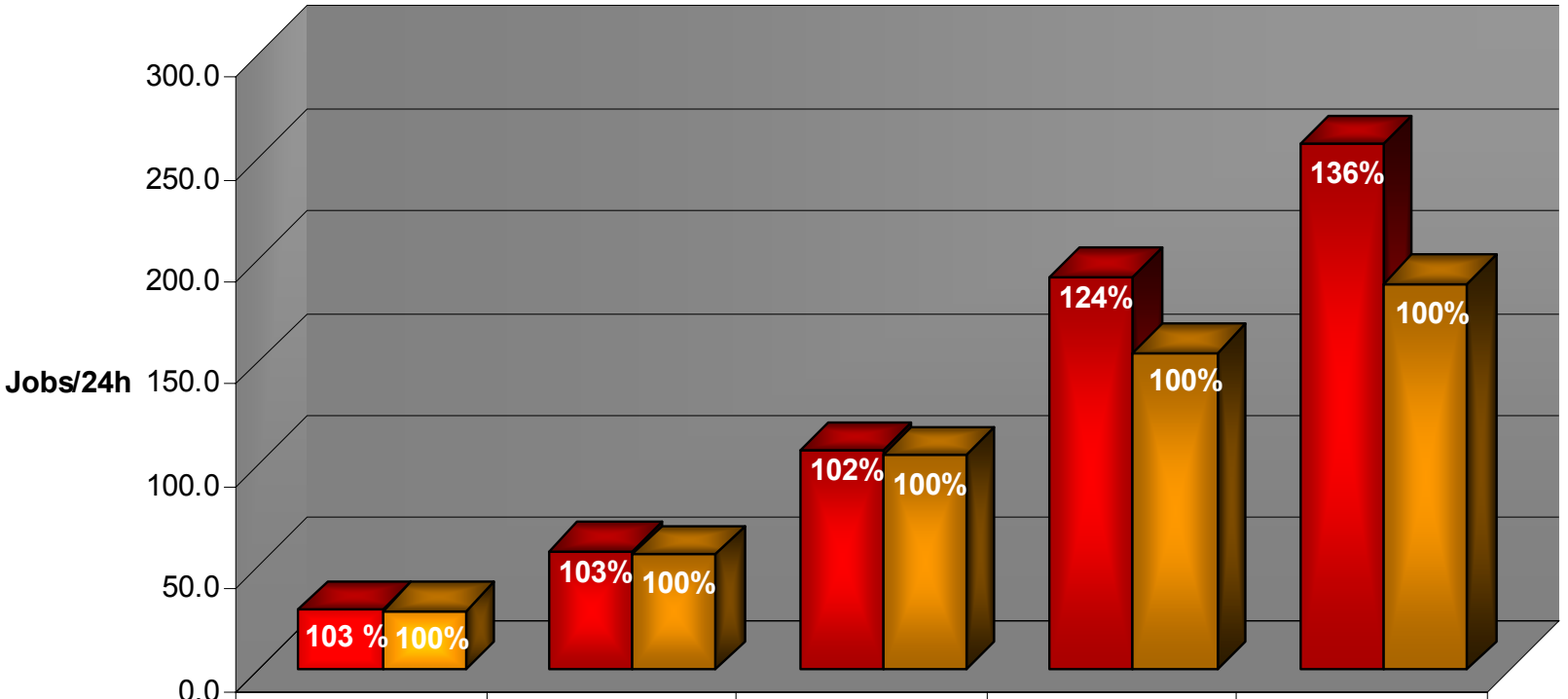


Single Node, 8-Core MPI Performance





Putting it all together: Application Performance and Scalabilty

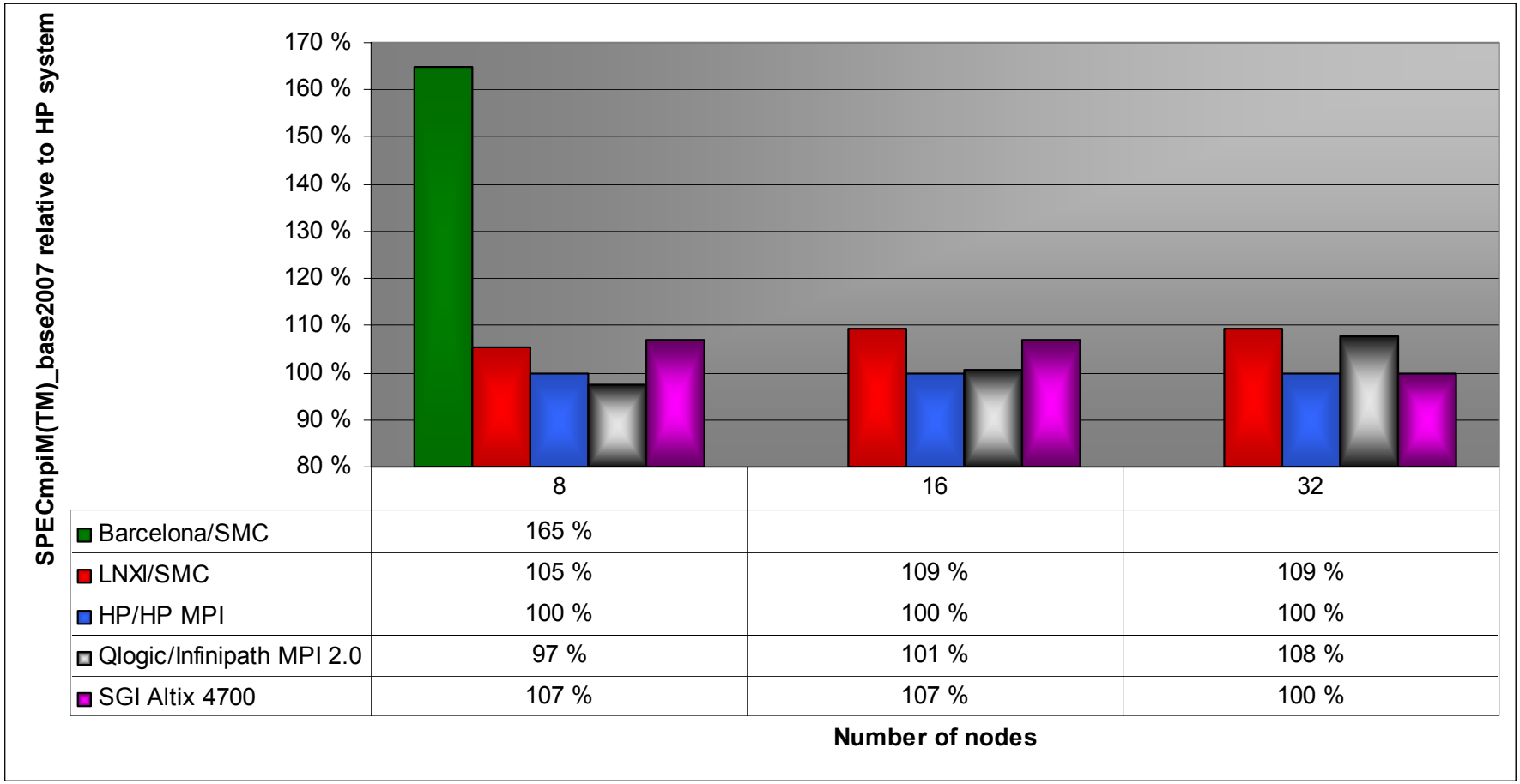


	1	2	4	8	16
Scali MPI Connect 5.5	29.7	57.9	107.3	192.4	257.1
HP MPI	28.8	56.3	105.0	155.1	189.1

Number of nodes

Comparison of dual-core, dual socket based clusters using LS-DYNA 971.7600 running neon_refined_revised. All data from www.topcrunch.org. Scali MPI Connect is used on SGI Altix 1200, HP MPI is used on CP3000 (BL460c on 1-8 nodes, 16 nodes run uses DL140). Both systems equipped with dual-core dual socket 3.0GHz Woodcrest. Interconnect is Mellanox Infiniband SDR for the SGI system, Mellanox DDR for the HP system.

SPEC MPI 2007 projections



	8	16	32
Barcelona/SMC	165 %		
LNXI/SMC	105 %	109 %	109 %
HP/HP MPI	100 %	100 %	100 %
Qlogic/Infinipath MPI 2.0	97 %	101 %	108 %
SGI Altix 4700	107 %	107 %	100 %

Disclaimer. To be filled in.