Session R02

Understanding xSeries Server Performance and Scaling

Gregg McKnight

IBM Distinguished Engineer

xSeries Development Performance

Research Triangle Park, North Carolina

IBM @server xSeries Technical Conference

Aug. 9 - 13, 2004

Chicago, IL

RETURN TO INDEX

Intro to Server Performance and Scaling

Understanding Server Performance and Scaling

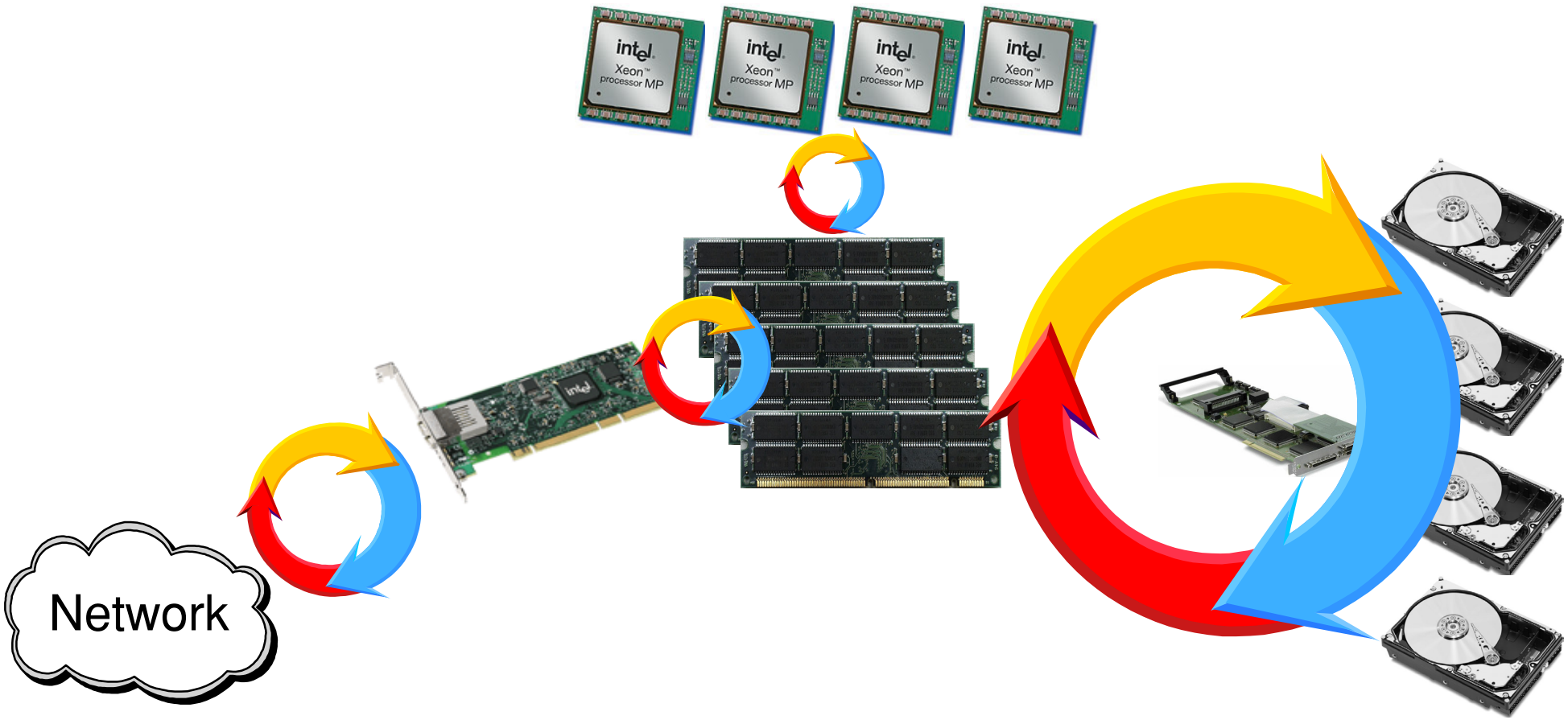
- A good way to visualize server operation and server performance is to think of a set of gears (or a transmission).
  - The LAN is one gear that drives the entire server
    - Workloads with a high memory hit rate will drive the:
      - Network / Memory / Processor gears
    - Workloads with a low memory hit-rate will drive:
      - The LAN / Memory / Processor gears and the Memory / Processor / Disk gears
  - To make things interesting the gear ratio is constantly changing
- A key fundamental component of understanding server performance is to recognize that any one component can slow the entire transmission (server)
- Only when each component (LAN, Processor, Memory, Disk) is capable of performing the required load does the entire system perform optimally

Server Gear Model - Optimized Configuration

Optimized Server Performance

- An optimal server will be running a workload on a configuration that has sufficient memory to buffer most read and write data so LAN requests almost never have to wait for disk I/O to be completed
  - Disk I/O will be done asynchronously and will not delay LAN transfer
- LAN transfers will happen at rated speed without delay
- Disk utilization will typically be low
  - Assuming adequate disk configuration
- Processor utilization will be moderate
  - Assuming ample processor performance to run network at required speed

Server Gear Model - Memory or Disk Bottleneck



Memory or Disk Bottleneck

- A memory and disk bottleneck will have similar symptoms
  - Insufficient memory will require synchronous disk I/O for most network requests
    - Because memory buffer will not be large enough to contain data for most requests
  - A slow disk will likely result in memory buffers filling with write data (or waiting for read data) which will delay all requests because free memory buffers are unavailable for write requests (or response is waiting for read data in disk queue)
- Disk/controller/memory utilization will typically be very high
- Most LAN transfers will happen only after disk I/O has completed
  - Very long response time, low network utilization
- Processor utilization will usually be low
  - Since disk I/O can take a relatively long time, and disk queues will become full, the processor will be idle or have low utilization as it waits long periods of time before processing next request

Server Gear Model - Network Bottleneck

Network Bottleneck

- To have a network bottleneck, the network must not be waiting on any disk I/O
  - So the server probably has sufficient memory
- Xeon DP and MP processors can drive two (or more) Gbit Ethernet adapters to saturation (for most but not all workloads)
  - So processor utilization may be high but probably not a bottleneck
- Understanding network traffic will help determine if network workload can scale up to multiple network adapters
  - We discuss this more in network section

Server Gear Model - CPU Bottleneck

Server Performance Basics Review Checklist

- Server performance improvements come from:
  - Parallelism or concurrence (Major)
  - Scalar or speed (Minor)

- **Throughput or capacity are key server performance metrics**
  - A benchmark that does not measure throughput or capacity with concurrent users is unlikely to represent actual server performance

- Understanding bottlenecks can occur in different components is key:
  - To understanding that the same server will perform differently for different applications
  - E.g. Each application causes different bottlenecks to occur

- As a result, it is not possible to say server x has 20% better performance than server y unless the workload AND configuration are well defined
  - E.g. Performance varies depending upon the application

Introduction to Server Scaling

What is Server Scaling?

- Most think: Scaling is the performance gain achieved by simply adding processors to a system
- But to achieve maximum scaling:
  - Network, Disk, Memory resources must be added or already configured with sufficient capacity to achieve maximum gains
  - Performance for most server workloads will not improve by adding CPUs alone
    - Unless one is "lucky" and the system already has sufficient memory, network, and disk resources to support increased CPU capacity
- **If CPU utilization is NOT 100% one has not achieved maximum scaling**
  - And even then one can stumble into software scaling bottlenecks
    - E.g. Serialization events which keep processors at 100% utilization but do no work
- Database applications are the most frequently used application when measuring scaling because:
  - Few applications scale as well as database

Why is Server Scaling so Important?

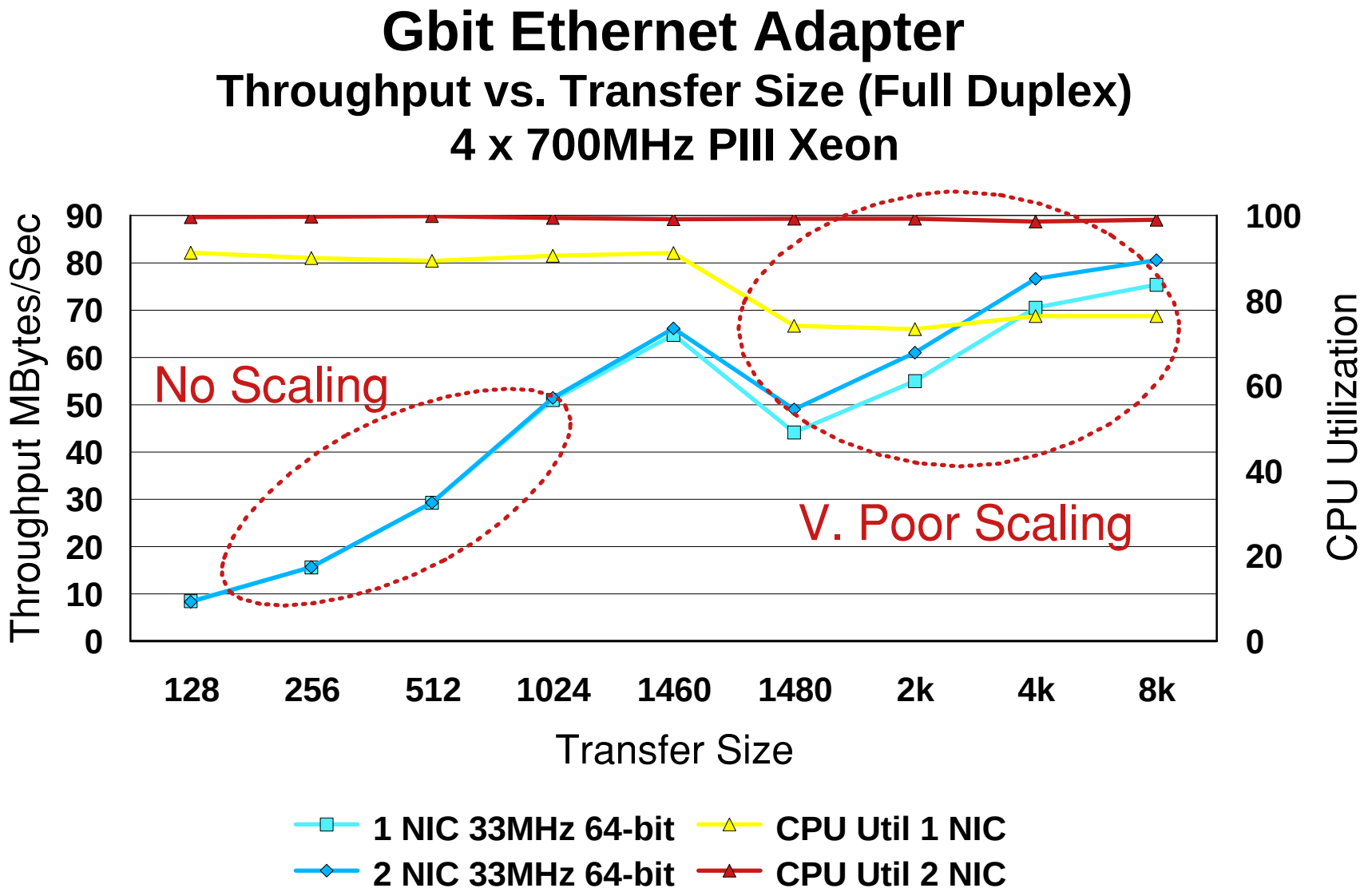
- Server replacement
  - Need to replace existing server that has outlasted it's usefulness with a more powerful server to achieve lower response time and increased throughput
  - To do this successfully one must understand how newer server improves scaling or performance over old model
- Server consolidation
  - Need to replace multiple existing servers with newer more powerful server
  - In this case we need to understand how replacement server will scale for existing workload to support the load of significantly greater numbers of users
- Server Expansion
  - Rapid growth businesses want to buy a server with headroom for future expansion
  - In this case the expected workload will grow over time and new server should have headroom to grow without experiencing poor performance

Each Server Subsystem Effects Server Scaling

- Server network subsystem
  - NIC hardware
  - NIC device driver stack
  - Network Infrastructure
- Disk subsystem
  - Disk controller
  - Controller device driver stack
  - Disk to controller interface (fiber, SCSI, SATA, SAS)
  - Disk driver technology and disk workload
- Processor and memory subsystem
  - Processor core performance
  - Processor to memory performance
  - Memory capacity
- Operating System and Applications
  - Concurrency and multithreading efficiency
  - Large memory support

Network Subsystem Scaling

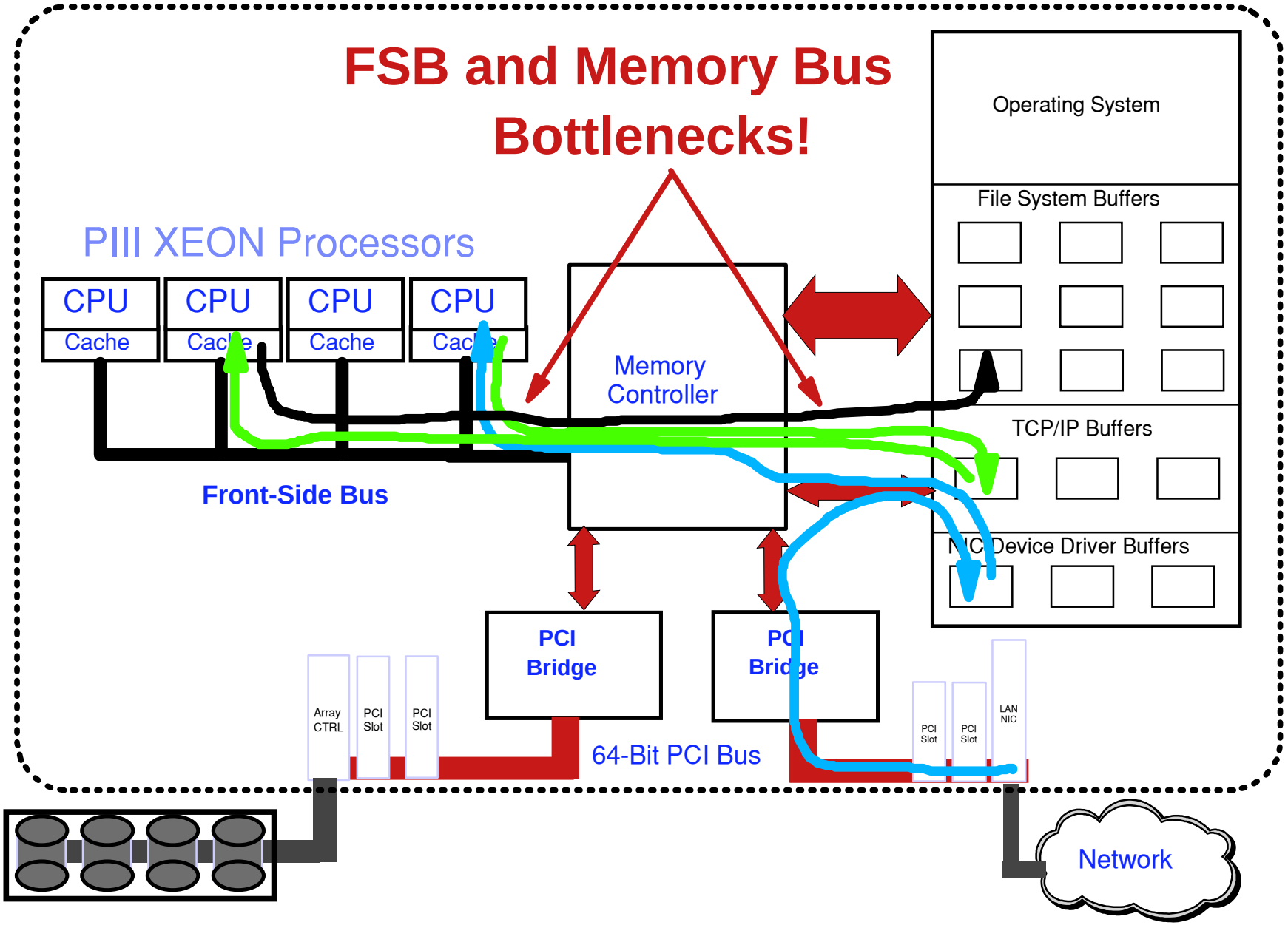
Legacy PIII Xeon System
Dual Gbit Ethernet Throughput - on Different PCI BUS



LAN Adapter Observations

- LAN adapter throughput is a function of payload (packet size = header + payload)
  - Small payload
    - Low throughput
    - High interrupt frequency
    - High CPU utilization
  - Large payload
    - High throughput
    - Low interrupt frequency
    - Low CPU utilization
- Maximum physical packet size is ~1500 bytes
  - This explains the decrease in throughput at 1480
    - 1480 is payload size NOT frame size
    - Header is about 48 bytes
  - Payload >1460 bytes requires multiple frames

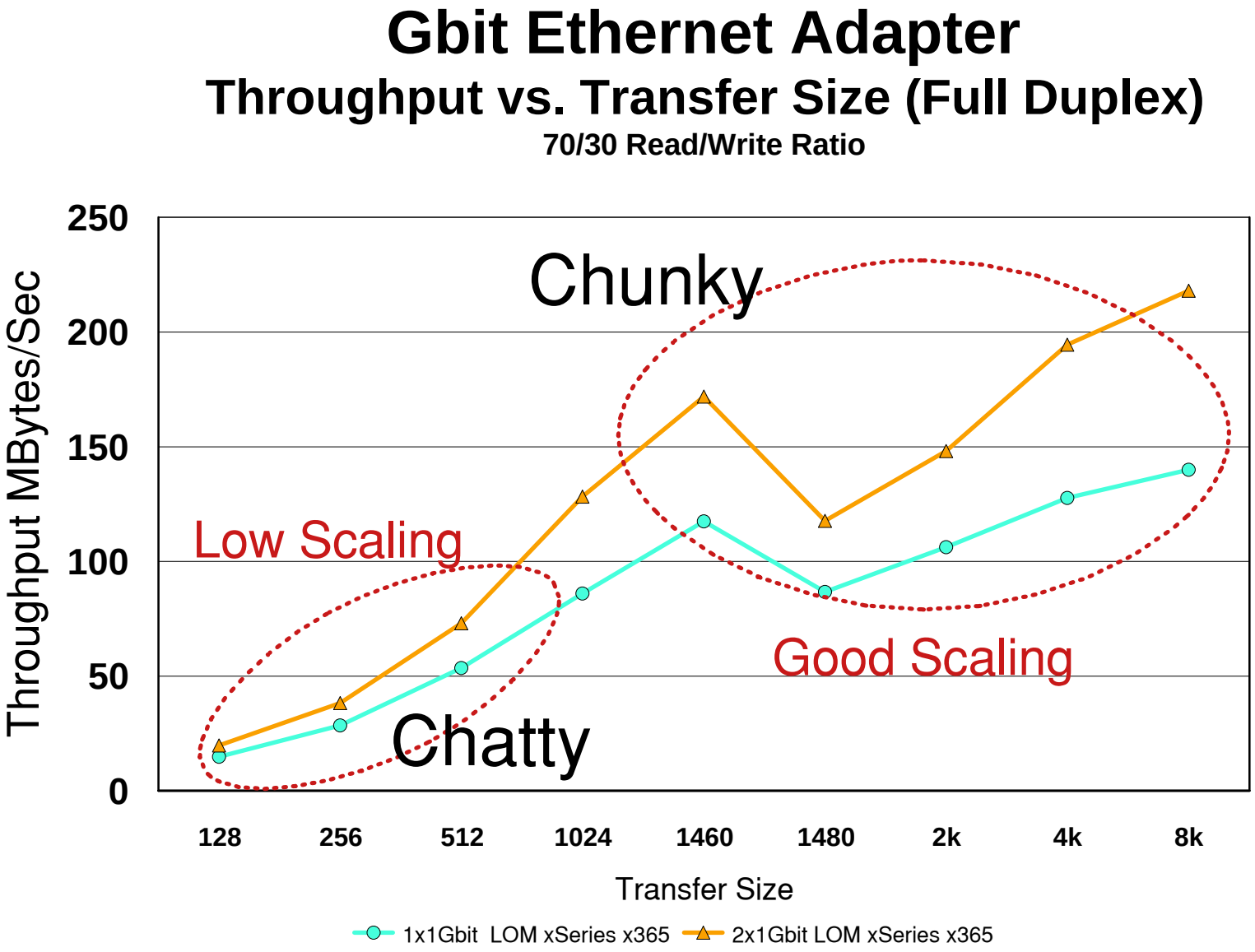
FSB and Memory Bus Bottlenecks!



LAN Adapter Observations

- Dual Gbit Ethernet adapters offer little throughput improvement vs. one Gbit adapter?
- Why?
  - Other bottlenecks limit aggregate throughput
  - It might be easy to say faster CPU will fix this
    - But faster CPUs do not significantly help
- Memory must provide enormous bandwidth for dual Gbit Ethernet adapters
  - Multiple trips over memory and FSB bus
- Front-side bus must snoop all PCI I/O

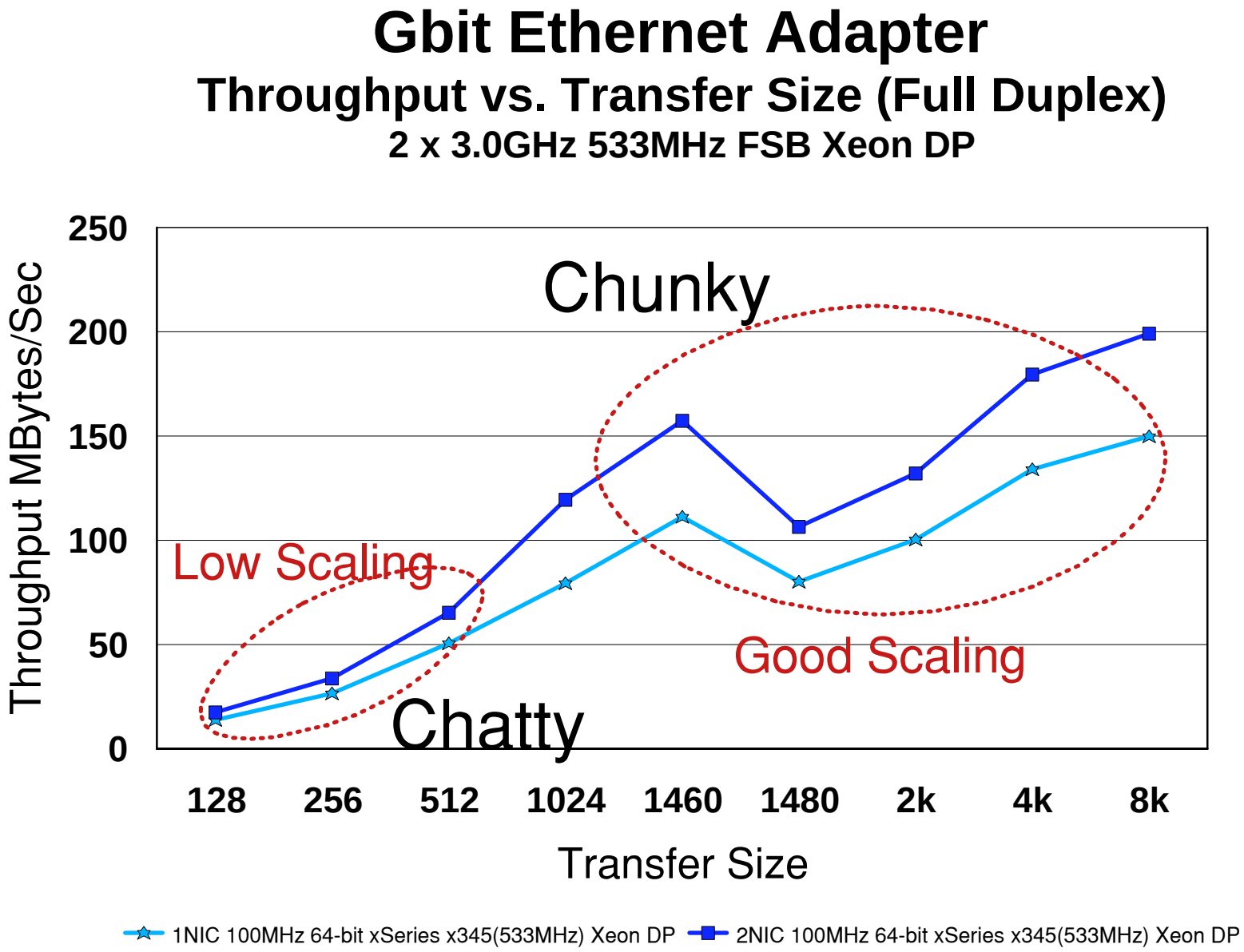
x365 2.8GHz Gbit Ethernet Throughput



Gbit Ethernet Throughput Observations

- x365 has optimized design to speed LAN throughput
  - Single Port LOM - 86% compared to PIII Xeon
  - Dual Port LOM - 190% (2.9 times greater)
  - Lower latency chipset combined with mature high-speed LOM Gbit Ethernet yield much improved TCP/IP performance
  - Efficient LOM can scale to multiple Gbit Ethernet
    - First generation Gbit Ethernet could not scale
    - LOM Efficiency due to
      - Improved ASIC
      - Driver/OS Interface Optimization
      - Interrupt batching

Gbit Ethernet Throughput for x345 3.0GHz 533MHz FSB



Network Scaling Checklist

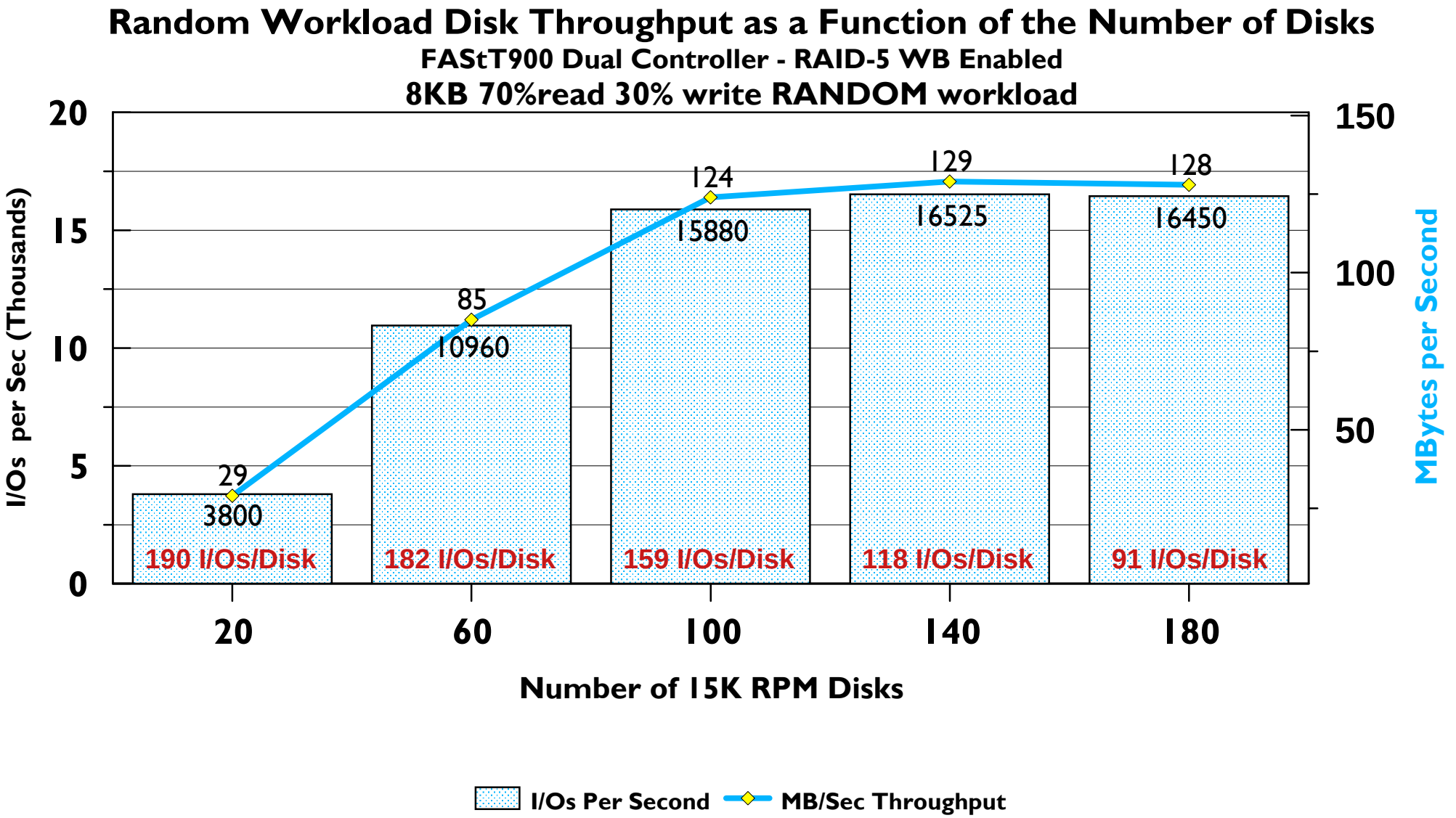
- PIII Servers scaled to at most 1 Gbit Ethernet
- Xeon MP and DP systems scale Gigabit Ethernet to at most 2 adapters
  - In general this much more bandwidth than most applications need for systems scaling
  - But consider network scaling when network attached storage is used because PIII systems did not scale while Xeon systems will scale.
- Chatty applications (packet sizes less than 512Bytes)
  - Do not scale as well due to serialization and overhead in TCP/IP stack
- Chunky applications scale network throughput well up to Gbit adapters
  - Gbit adapter = 150MB/Sec
  - 100MHz PCI = 800MB/Sec
  - So don't need to worry much about PCI slot configuration for Gbit Ethernet

Disk Subsystem Scalability

Disk Subsystem Scaling

- Disk subsystem perhaps most challenging subsystem to properly configure
- Many just look at disk interface speed and disk capacity
  - Too many don't consider configuration and workload
    - Random or sequential?
    - Large I/O or small I/O?
- 15,000 RPM disk
  - Average latency 2.0mSec
  - Average Seek 3.8mSec
  - Command and data transfer < 1mSec
    - Total random access time = 2.0 + 3.8 + 1.0 = 6.8mSec
    - 1 / 0.0068sec = **147 I/Os per second per disk**
    - **At 8KB per I/O that is about 1.15 MB/Second**
- 10,000 RPM disk
  - Average latency 3.0 mSec
  - Average Seek 4.9mSec
  - Command and data transfer < 1mSec
    - Total random access time = 3.0 + 4.9 + 1.0 = 8.9mSec
    - 1 / 0.0089 = **112 I/0s per second per disk**
    - **At 8KB per I/O that is about 900 KB/Sec!**

Drive Scaling - RAID-5

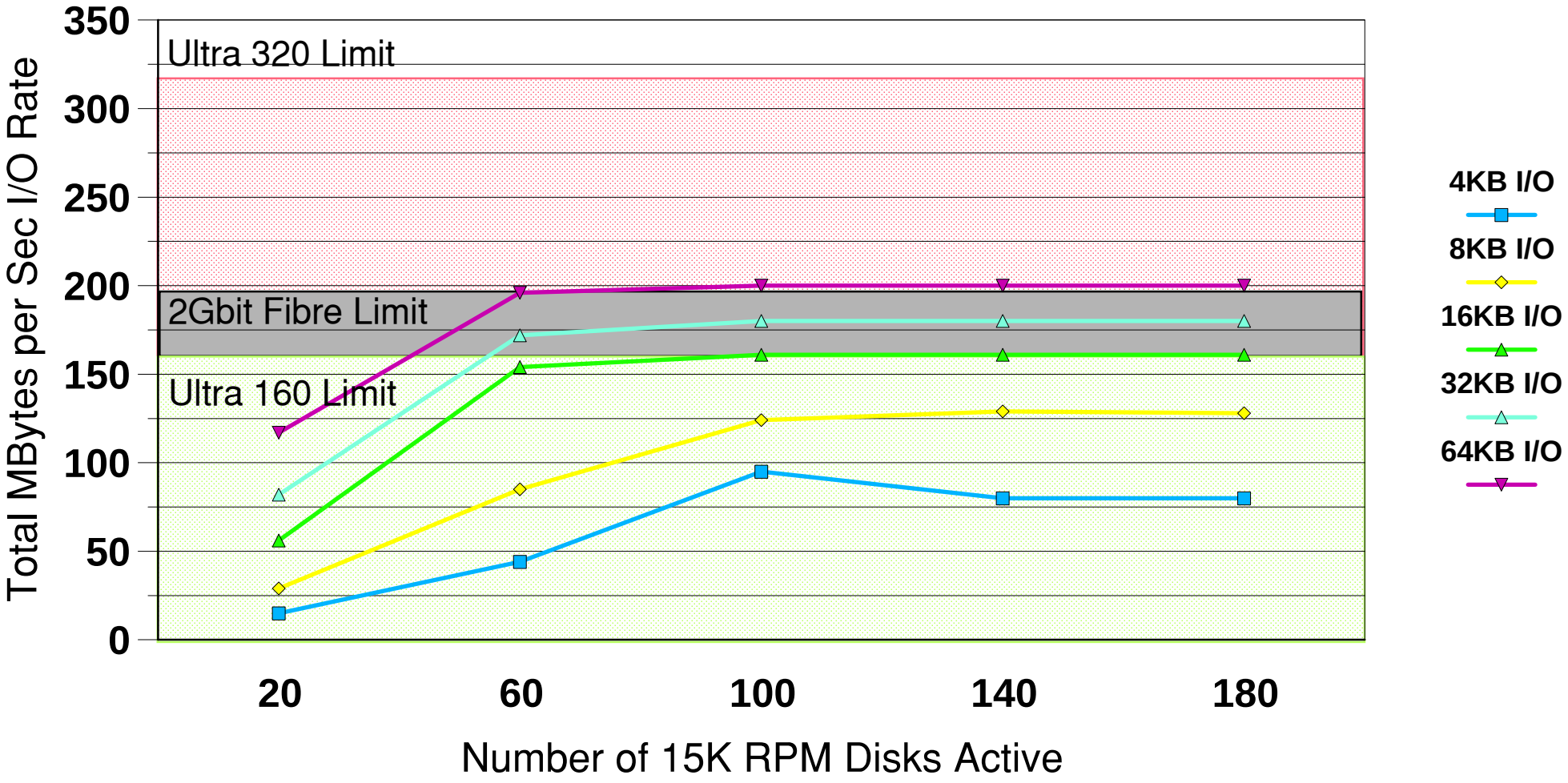


Disk Subsystem Scaling - RAID-5

Random Workload Disk Throughput as a Function of the Number of Disks

FAStT900 Dual Controller - RAID-5 WB Enabled

8KB 70% read 30% write RANDOM workload

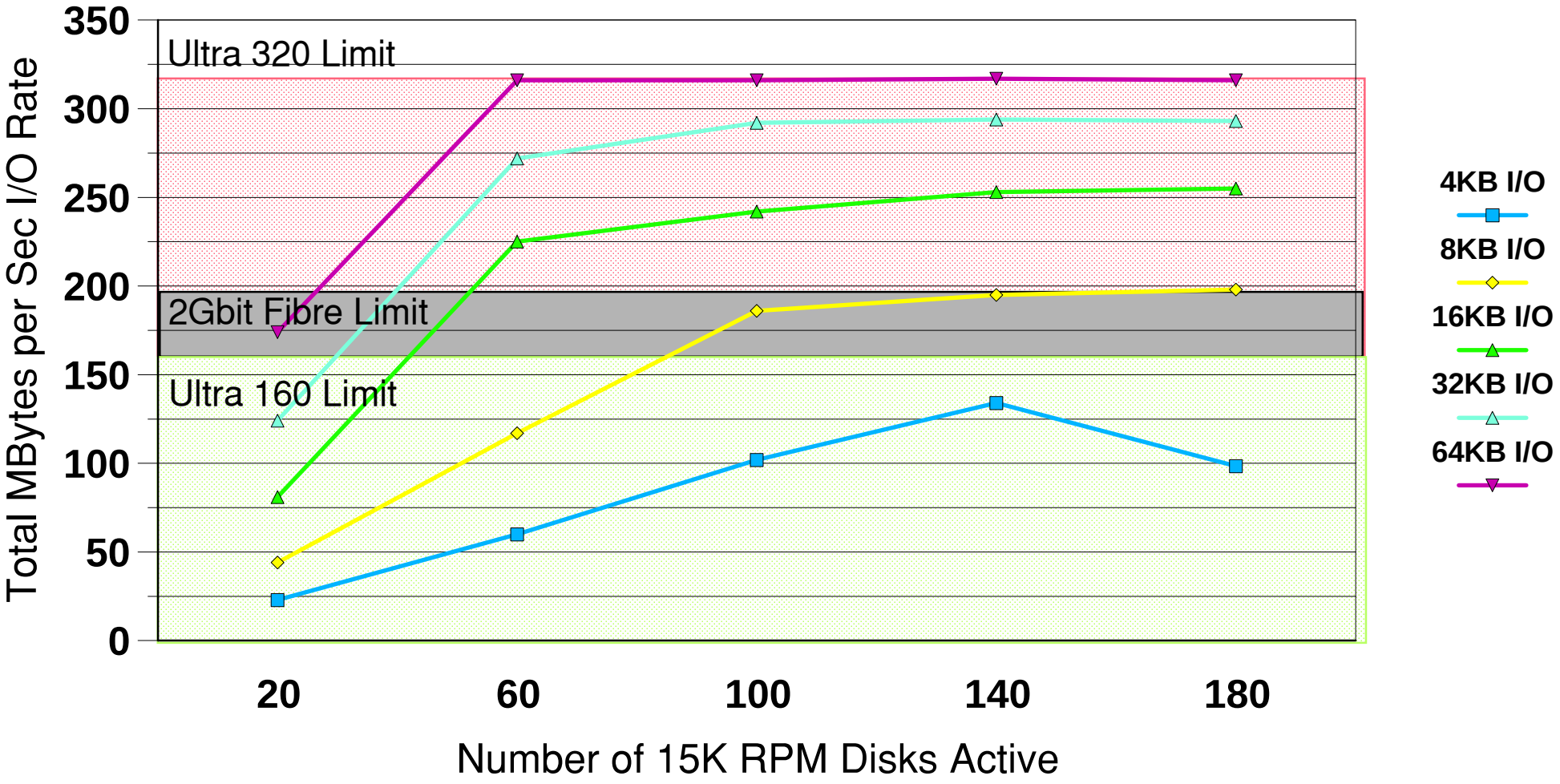


Disk Subsystem Scaling - RAID-10

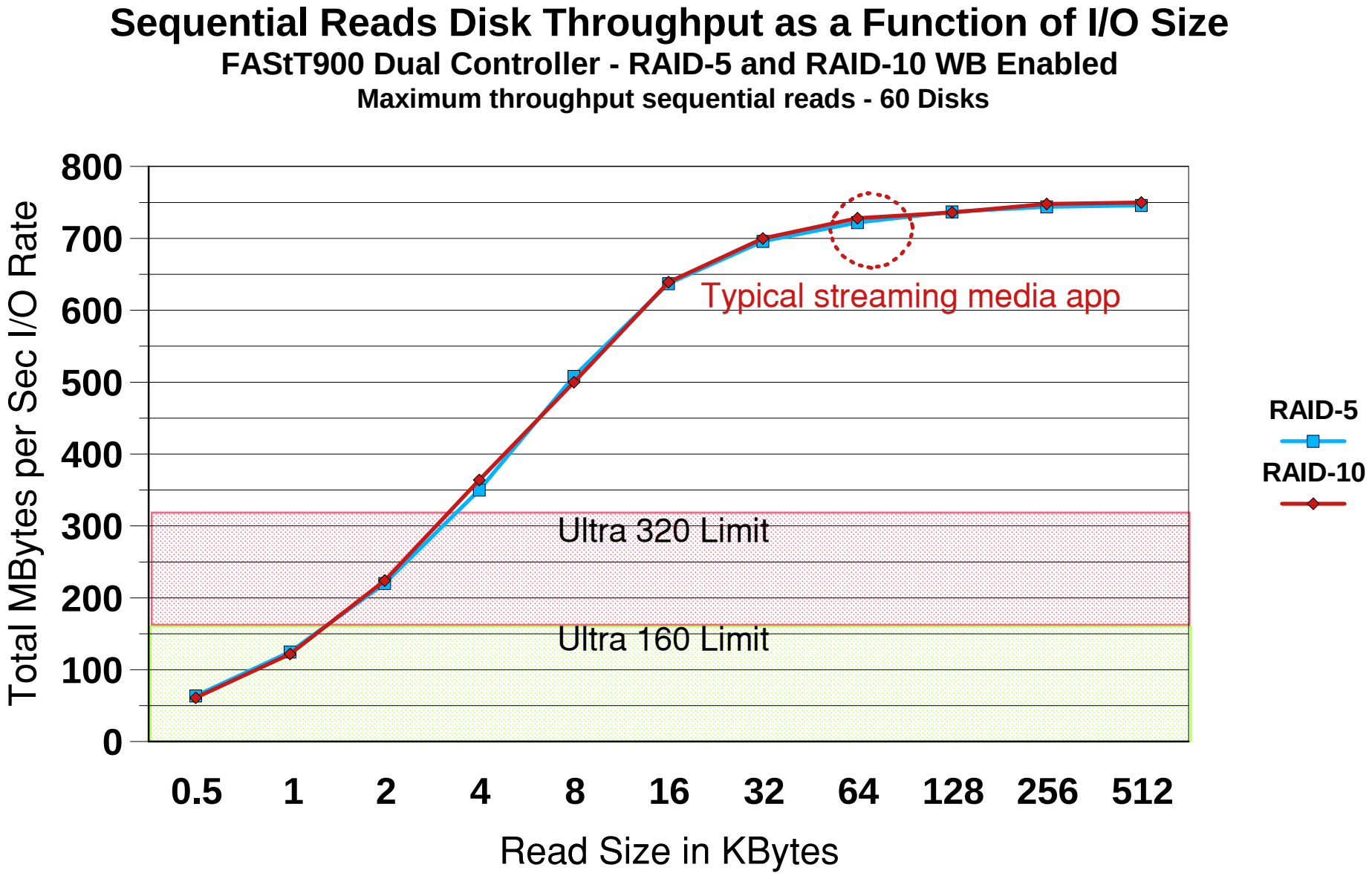
Random Workload Disk Throughput as a Function of the Number of Disks

FAStT900 Dual Controller - RAID-10 WB Enabled

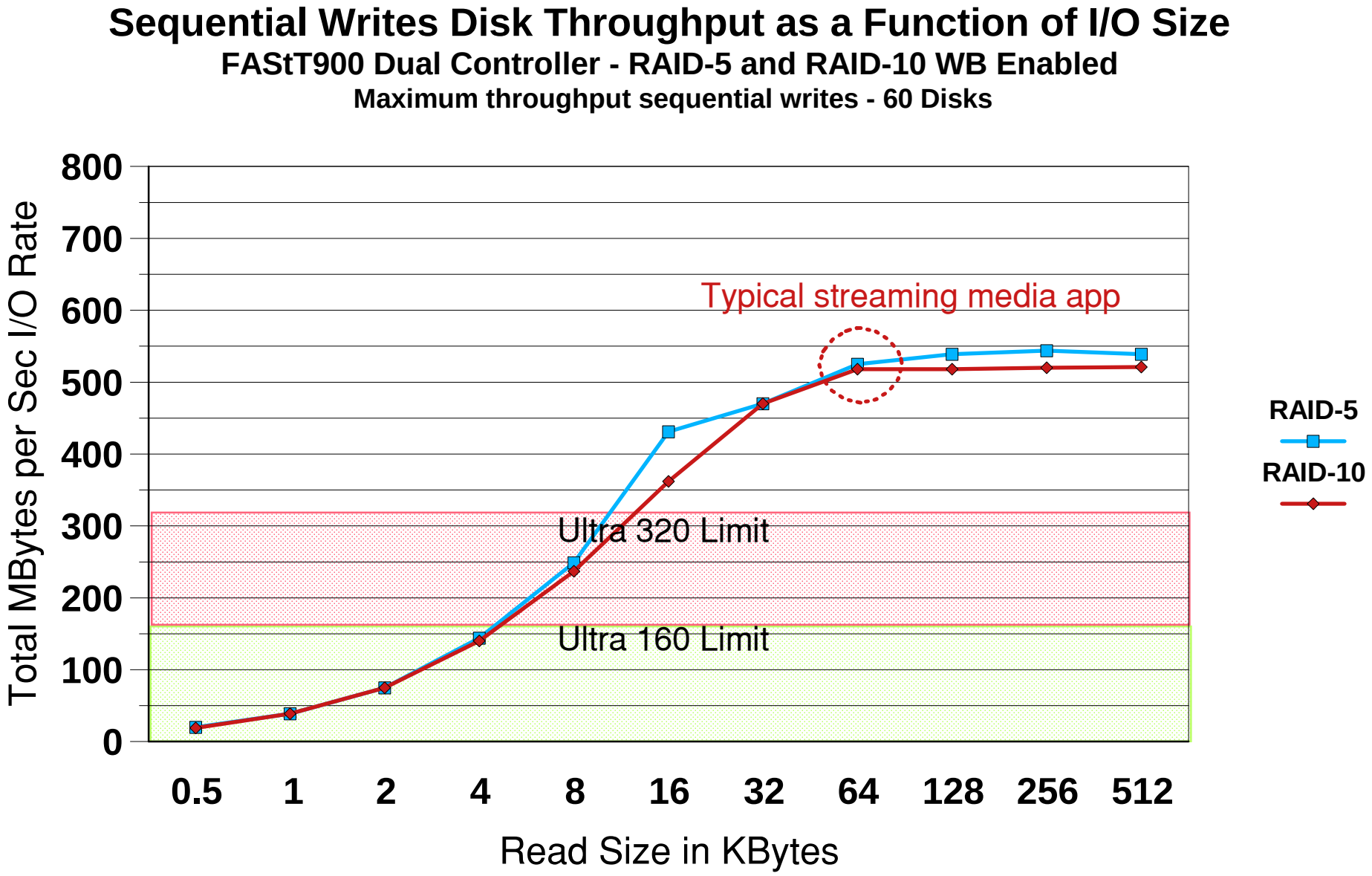
8KB 70% read 30% write RANDOM workload



Disk Subsystem Scaling - RAID-5 Same as RAID-10



Disk Subsystem Scaling - RAID-5 Same as RAID-10

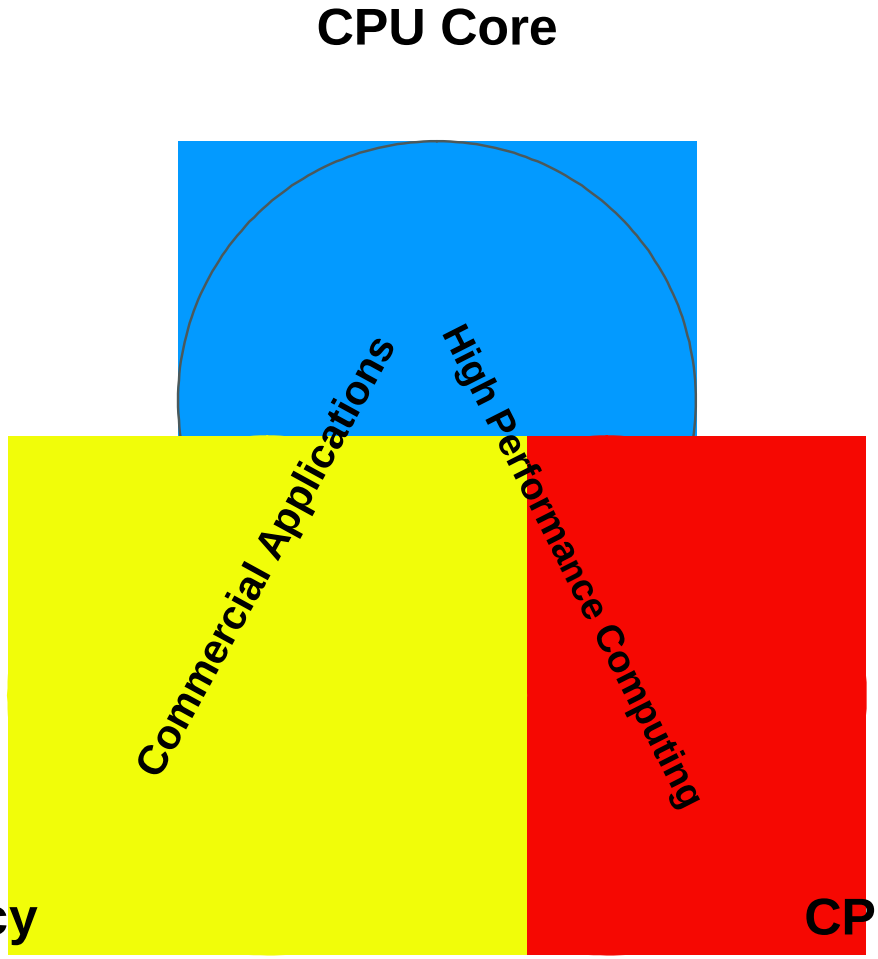


Disk Scaling Checklist

- Disk configuration is probably the most frequently misunderstood server problem
- Random read/write workloads usually require lots of disks to scale
  - Less concern for SCSI / Fibre Channel bus bandwidth limitations
  - Larger databases = more disks
  - Greater number of processors = more disks
  - RAID-10 50 - 60% greater throughput than RAID-5
    - Depends upon % of write commands
    - 50 - 60% is 70% reads 30% writes - typical commercial workload
- Sequential workloads will stress bus and fibre channel
  - Examine # SCSI Bus and # Fibre channels / dual-controllers
  - Greater connection bandwidth recommended where maximum throughput is desired
  - RAID-10, RAID-0, RAID-5 all have similar streaming read and write throughput
    - Assuming same number of disks being read
    - WB Disk controller caches

Processor Performance Scalability

Processor Scaling Bottlenecks



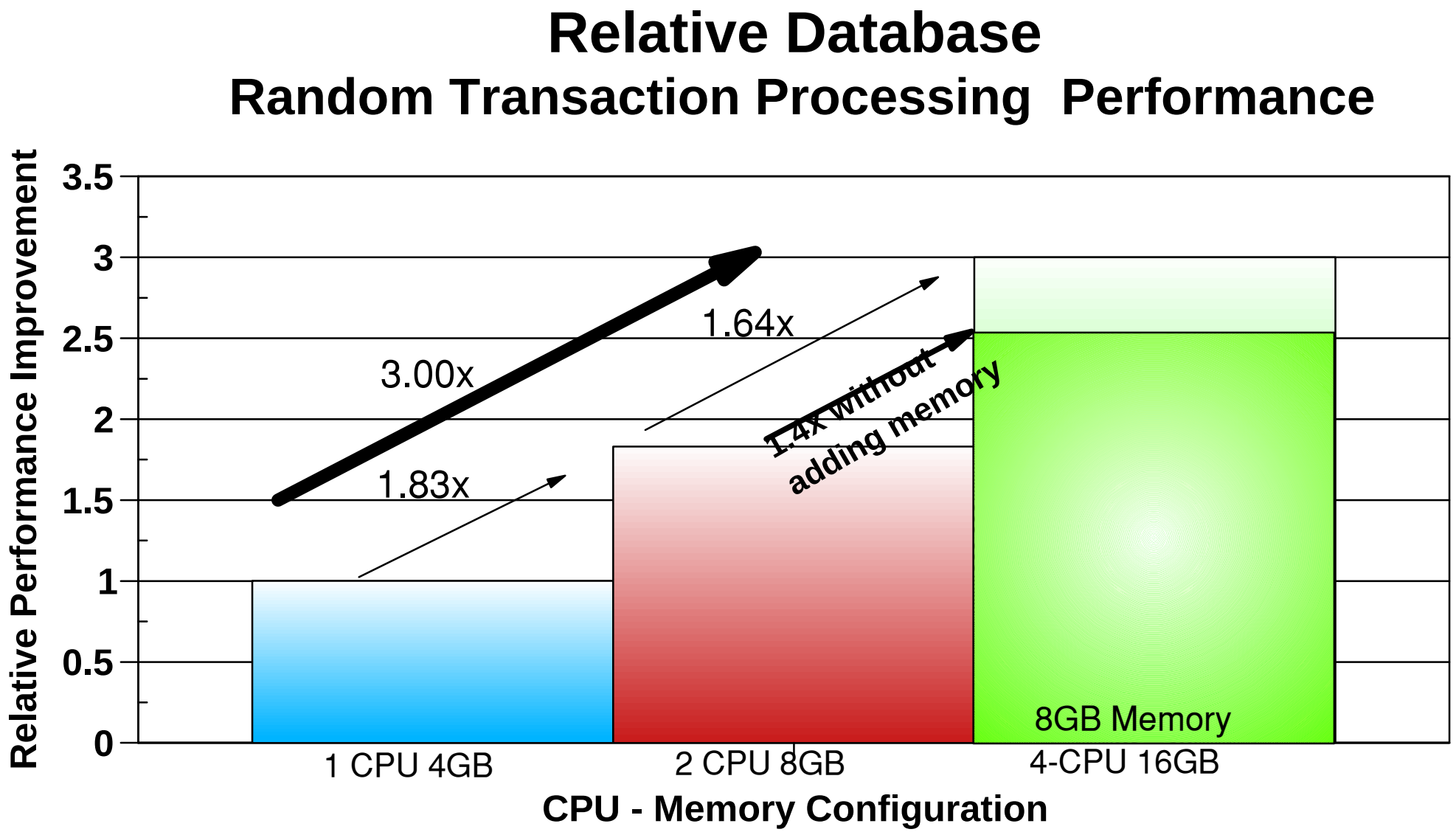
Processor Scaling Differences

- Performance for memory latency sensitive applications is determined by loaded memory latency efficiency and ability of memory subsystem to parallelize memory access
- Performance for bandwidth sensitive applications is determined by processor to memory interconnect bandwidth
- Performance for processor core sensitive applications is determined by processor microarchitecture, core frequency, and core multithreading

Memory Latency and Processor Core Scaling

1 - 4 Way Data Base Scaling

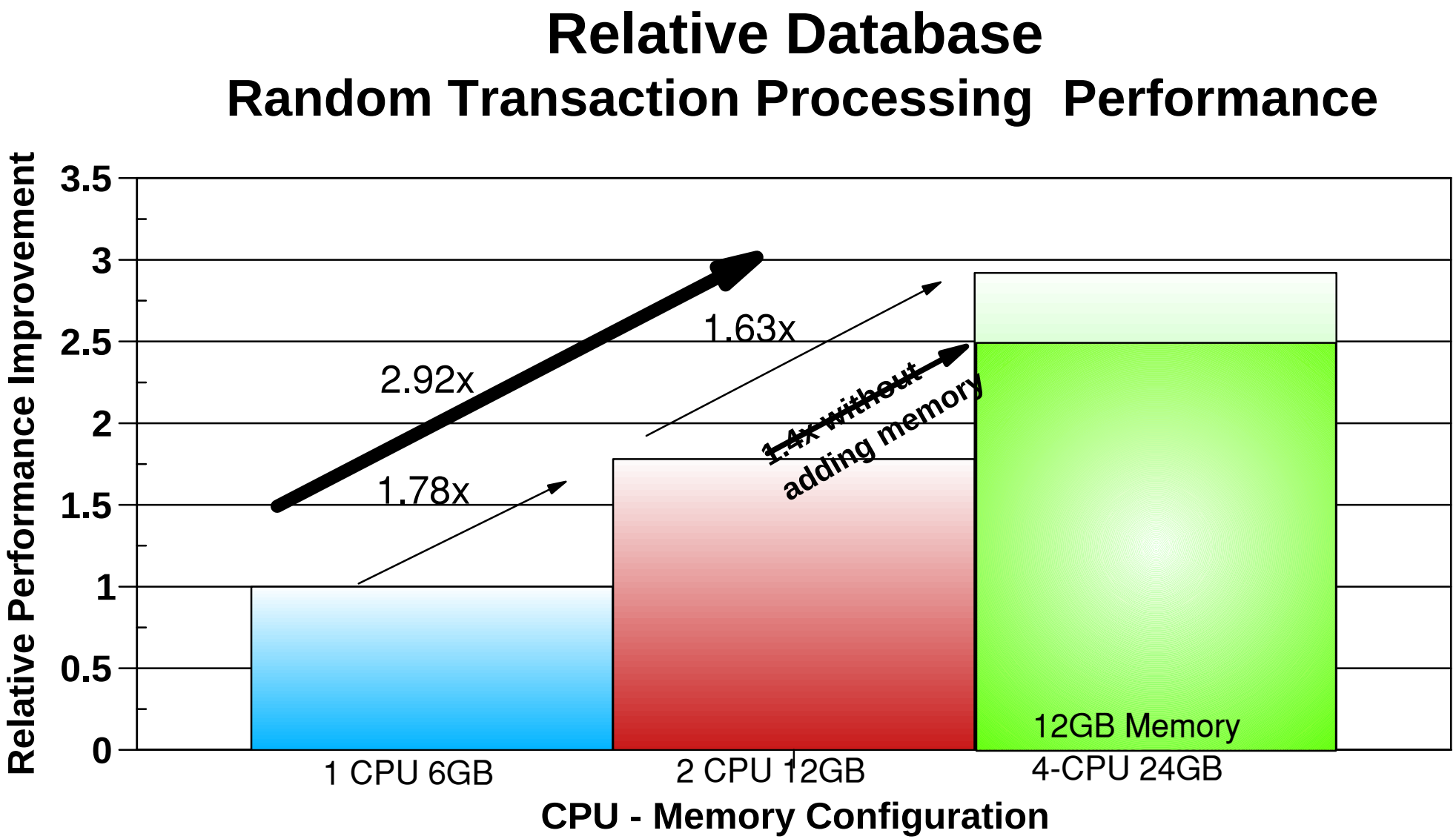
1.6GHz 1MB L3 Xeon MP Processors - **Hyperthreading Enabled**



Scaling data courtesy of Intel Corp.

1 - 4 Way Data Base Scaling

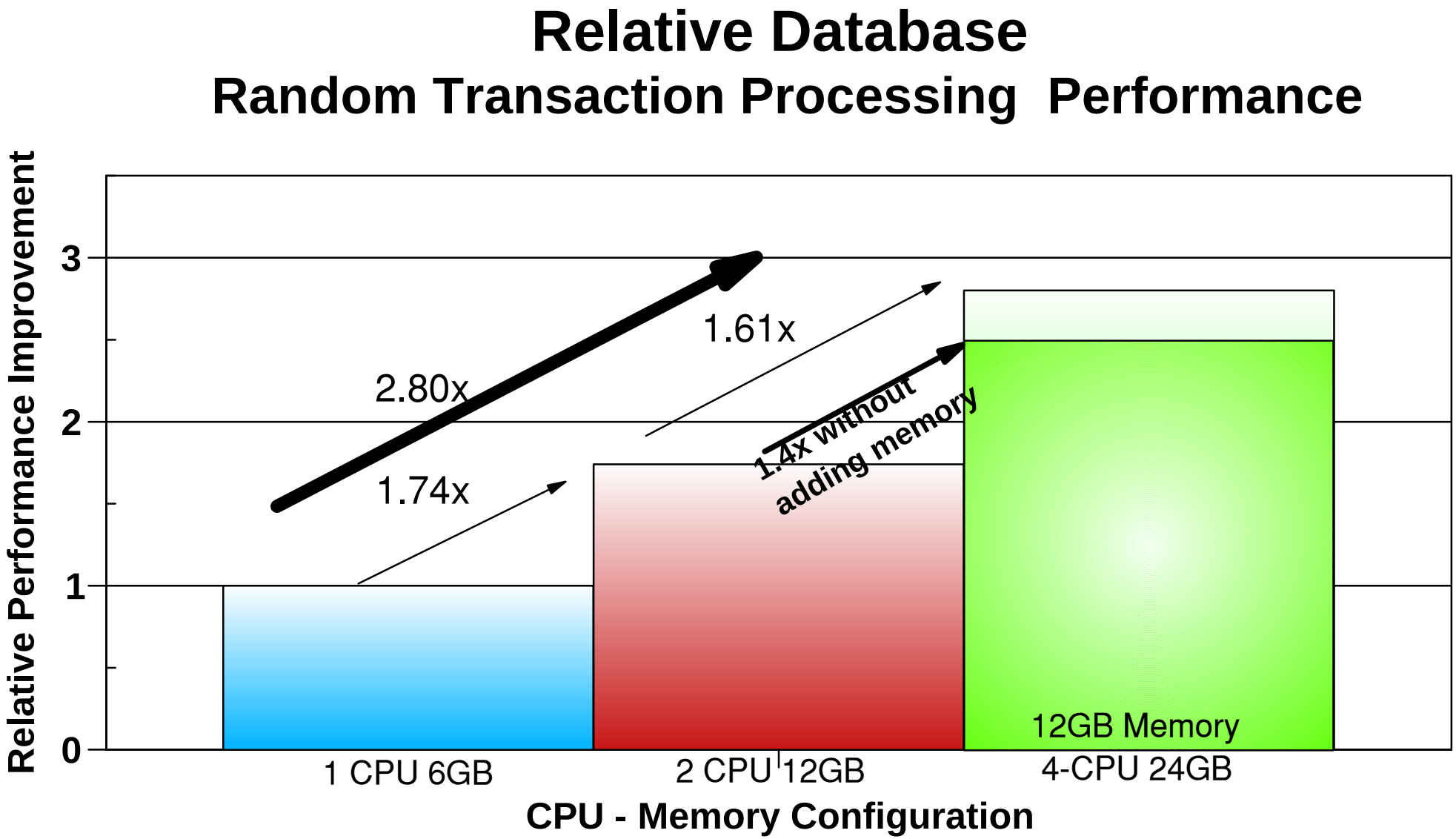
2.0GHz 2MB L3 Xeon MP Processors - **Hyperthreading Enabled**



Scaling data courtesy of Intel Corp.

1 - 4 Way Data Base Scaling

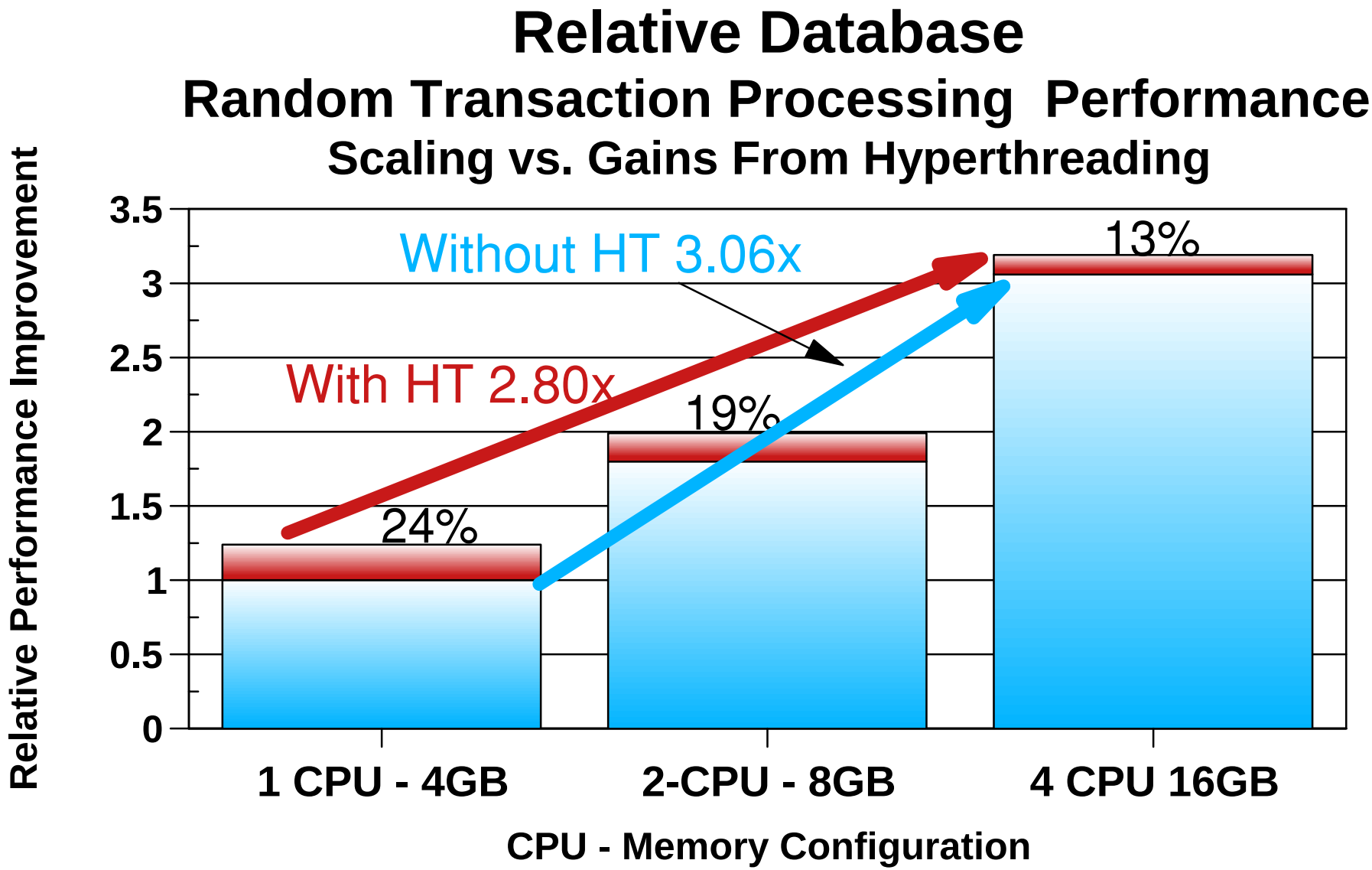
3.0GHz 4MB L3 Xeon MP Processors - **Hyperthreading Enabled**



Scaling data courtesy of Intel Corp.

1 - 4 Way Data Base Scaling

3.0GHz 4MB L3 Xeon MP Processors - **Hyperthreading On vs Off**



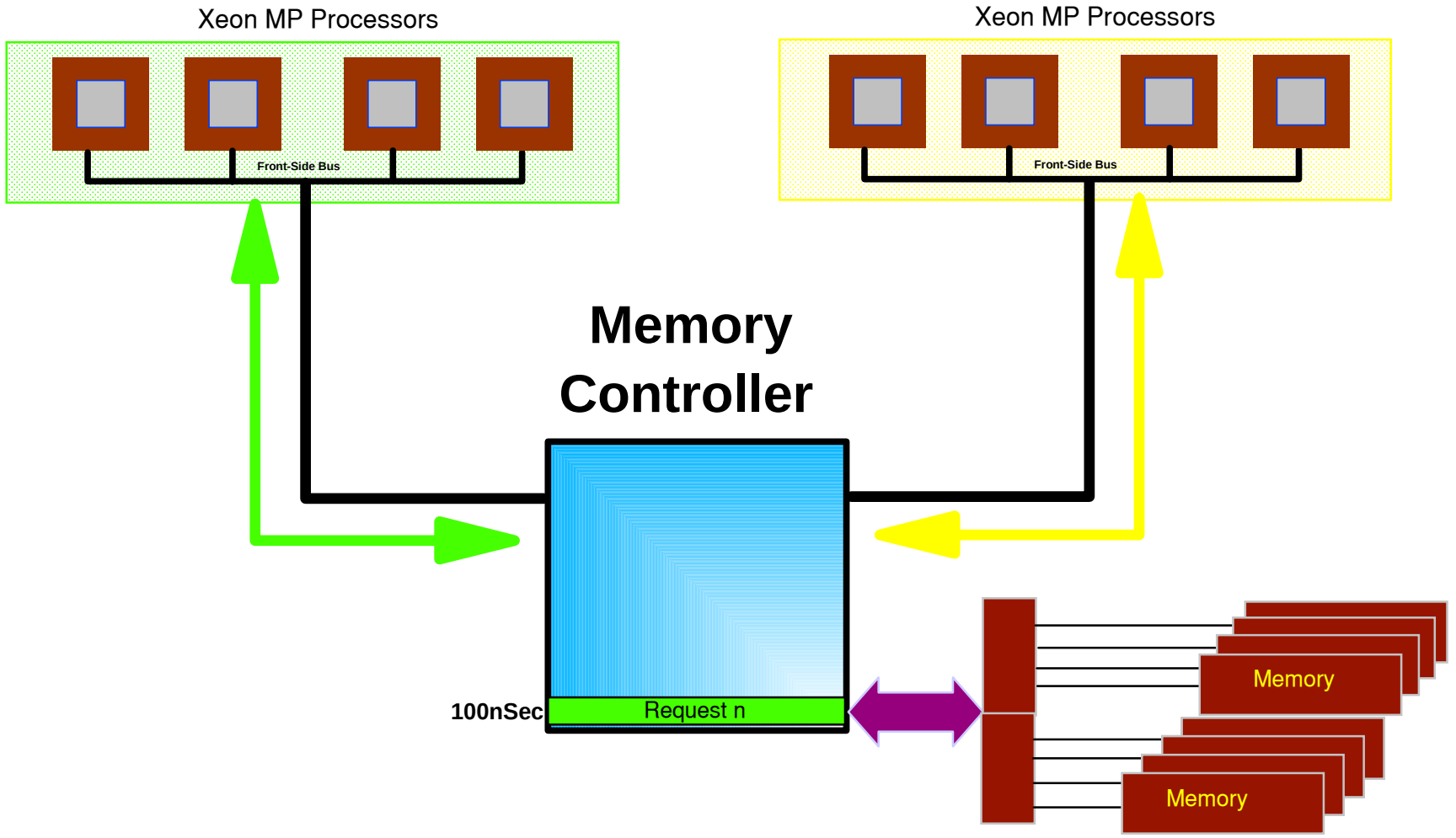
Scaling data courtesy of Intel Corp.

4-Way Scaling Summary

- Notice that without adding memory
  - Systems scale poorly
    - Only 1.44x without adding memory
  - 2 - 4 way scaling = 1.64 X
    - When adding memory
- Scaling is reduced as the number of processors increases
  - 4 - 8 way will never be as good as 2 - 4 way
- Faster processors have poorer scaling than slower processors
  - Bottlenecks occur sooner with faster processors
- Also notice that Hyperthreading reduces scaling
  - HT improves overall performance but reduces scaling
  - Almost any technology that improves CPU performance will reduce scaling
    - Because faster CPUs increase the base comparison
    - And non-CPU bottlenecks occur sooner

Multiple Memory Controller Technology

Typical SMP Memory Controller

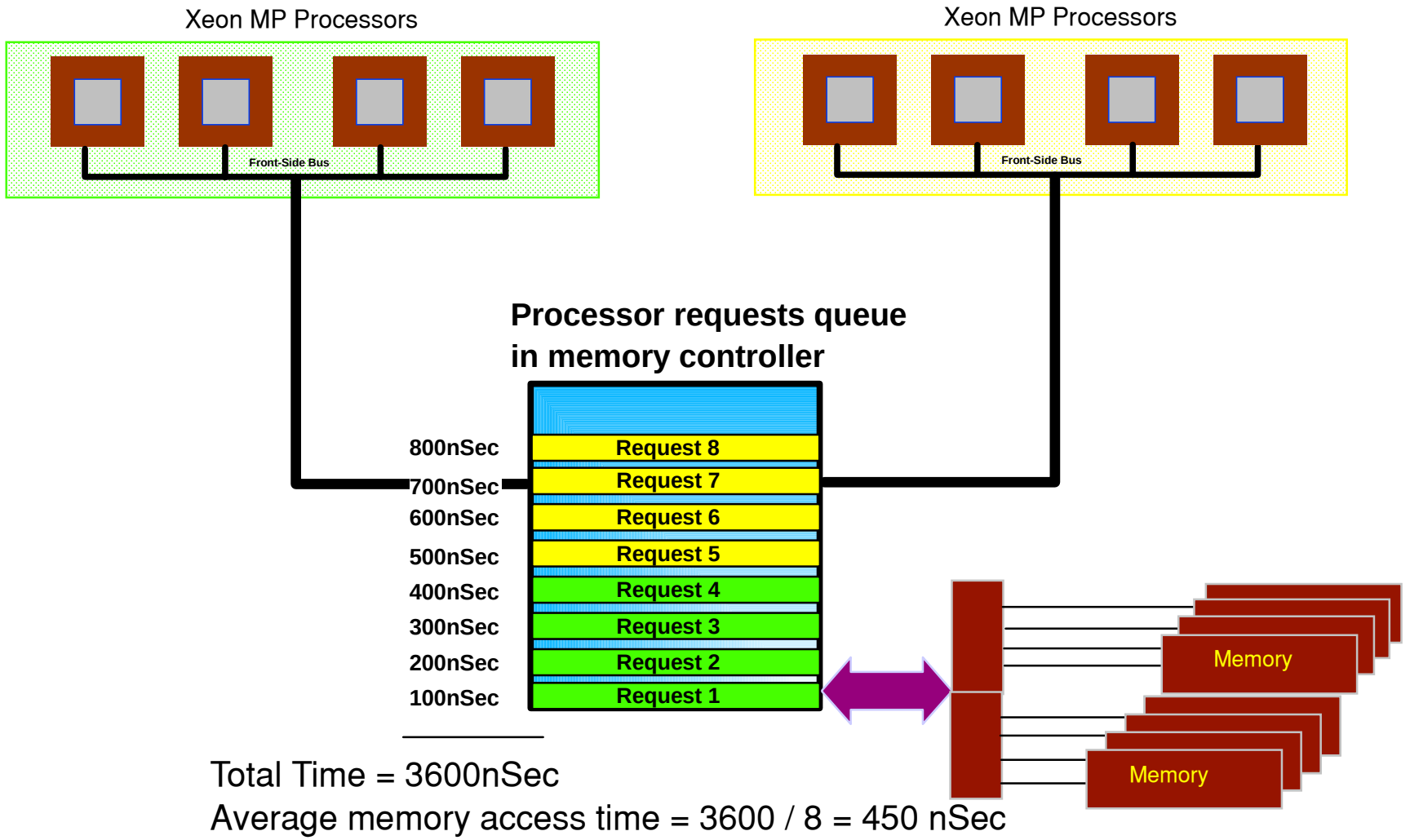


Assume arbitrary memory latency is 100nSec for one processor
In SMP it is the same for all processors

Assumptions

- For next set of charts I assume 100nSec access time to memory
  - This is an arbitrary number chosen to simplify the analysis and comparisons
  - Various designs have different values but using 100 nSec keeps the math simple

SMP Memory Controller Bottleneck



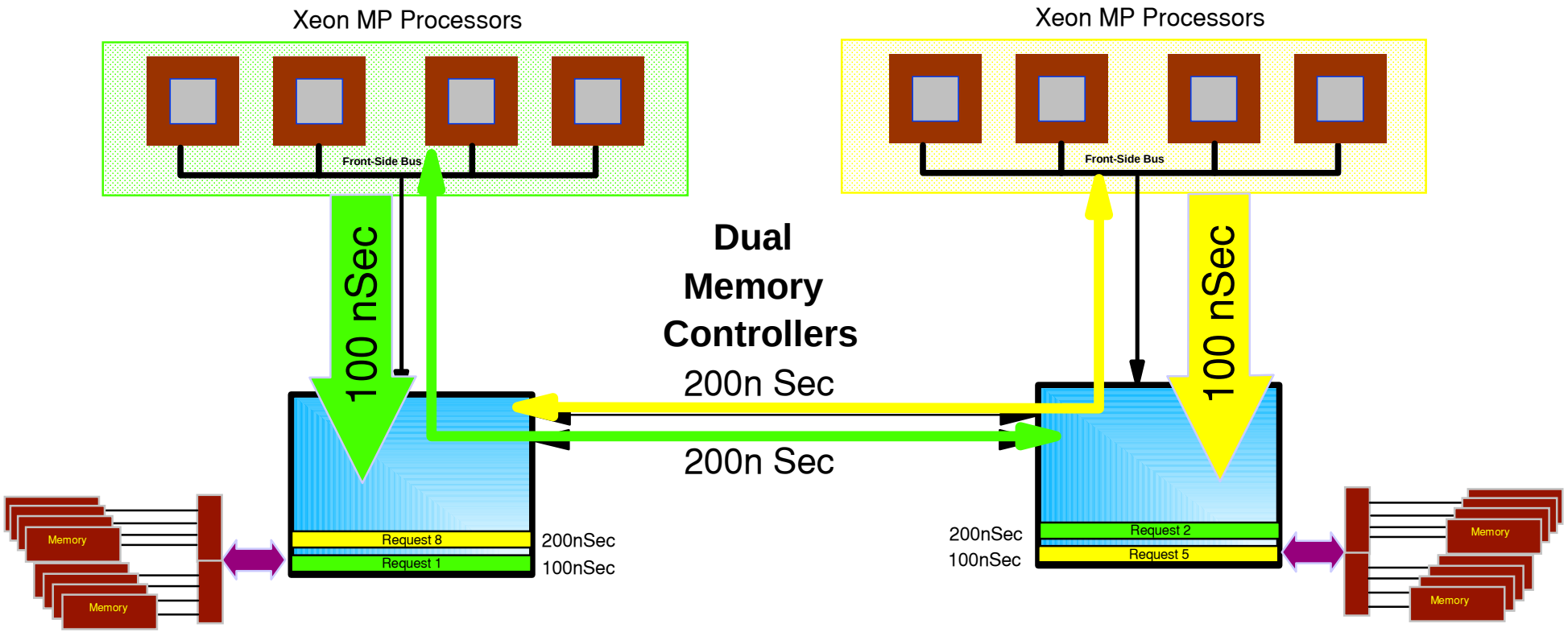
Worse Case Memory Access Time = 800 nSec

Average Memory Access Time = 450 nSec

Memory Controller Challenges

- As the number of processors increase...
- As processors get faster...
- As processors increase in threading (SMT)
  - Hyperthreading
- Bottlenecks occur because of the ability of the memory controller to respond to the more rapid processor requests for instructions and data
  - Queuing time at the single memory controller limits performance
- We are at the point in technology where new designs must be introduced to fix the inherent single memory controller bottleneck

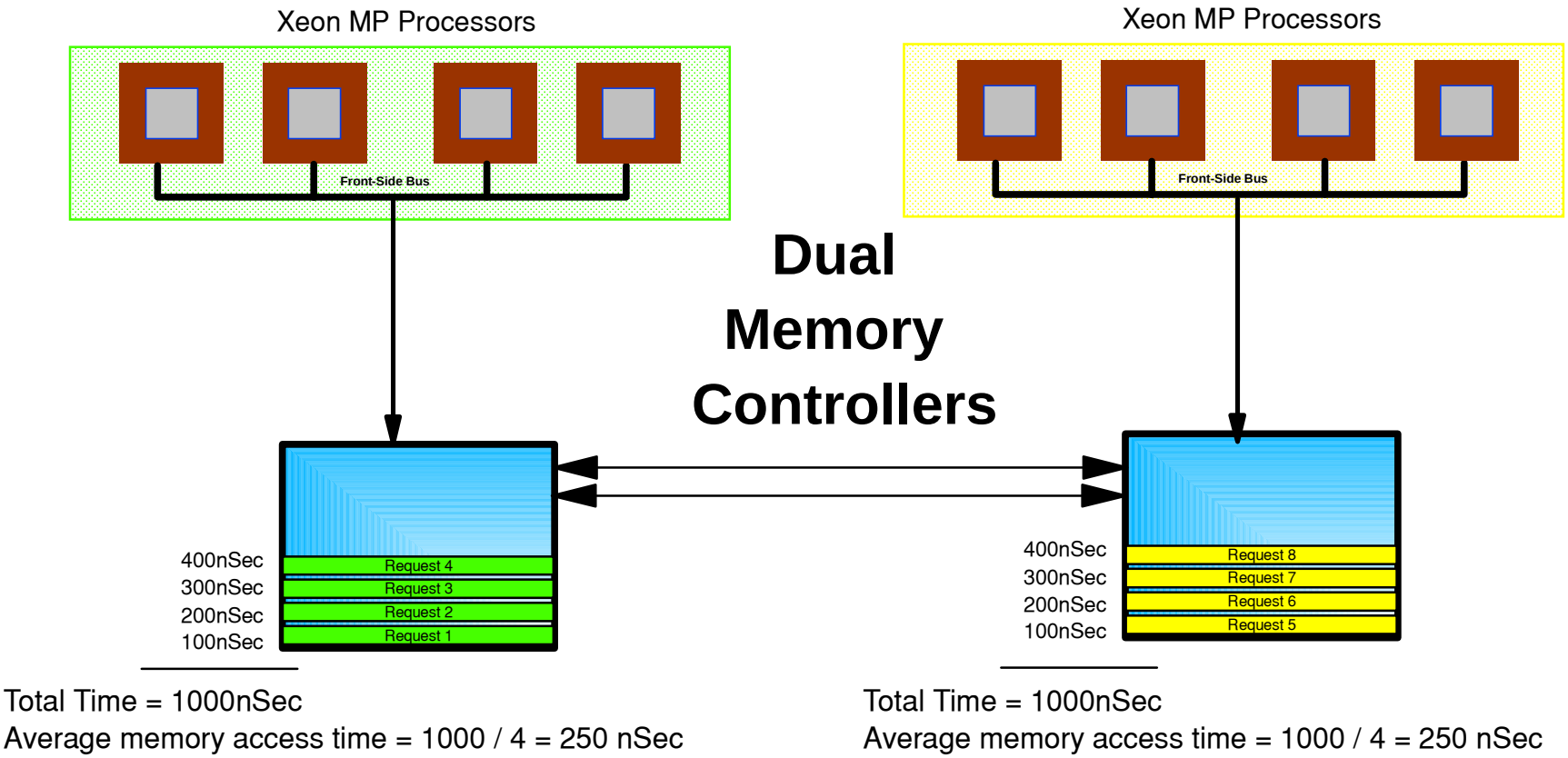
x440-445 Multiple Memory Controllers - NUMA



Local access is 100 nSec

But remote access is usually 2x or 200 nSec

x440-445 Multiple Memory Controllers - Optimal Solution

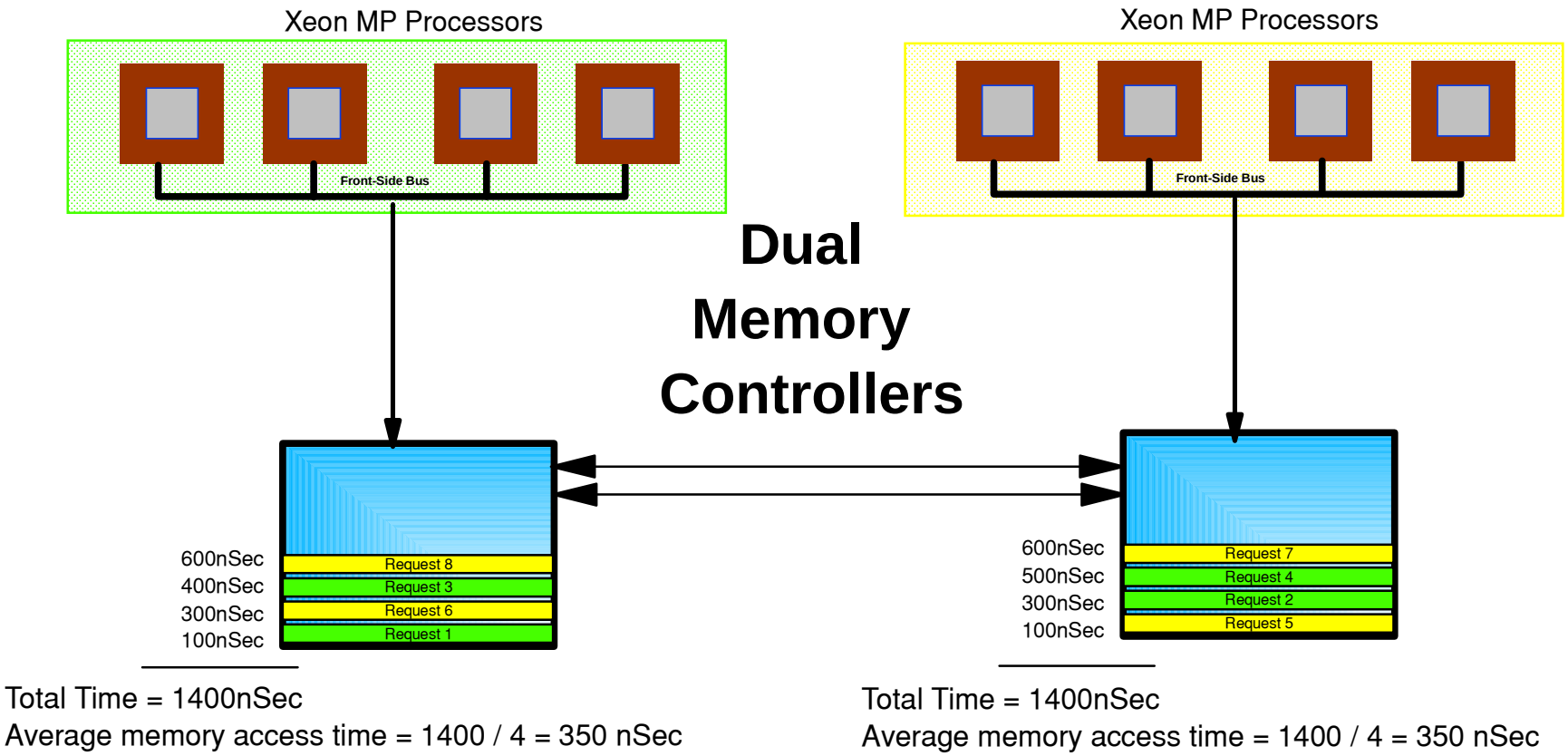


Worse Case Memory Access Time = 400 nSec

Average Memory Access Time = 250 nSec

44% Reduction in memory latency compared to SMP

x440-445 Multiple Memory Controllers - Random Distribution



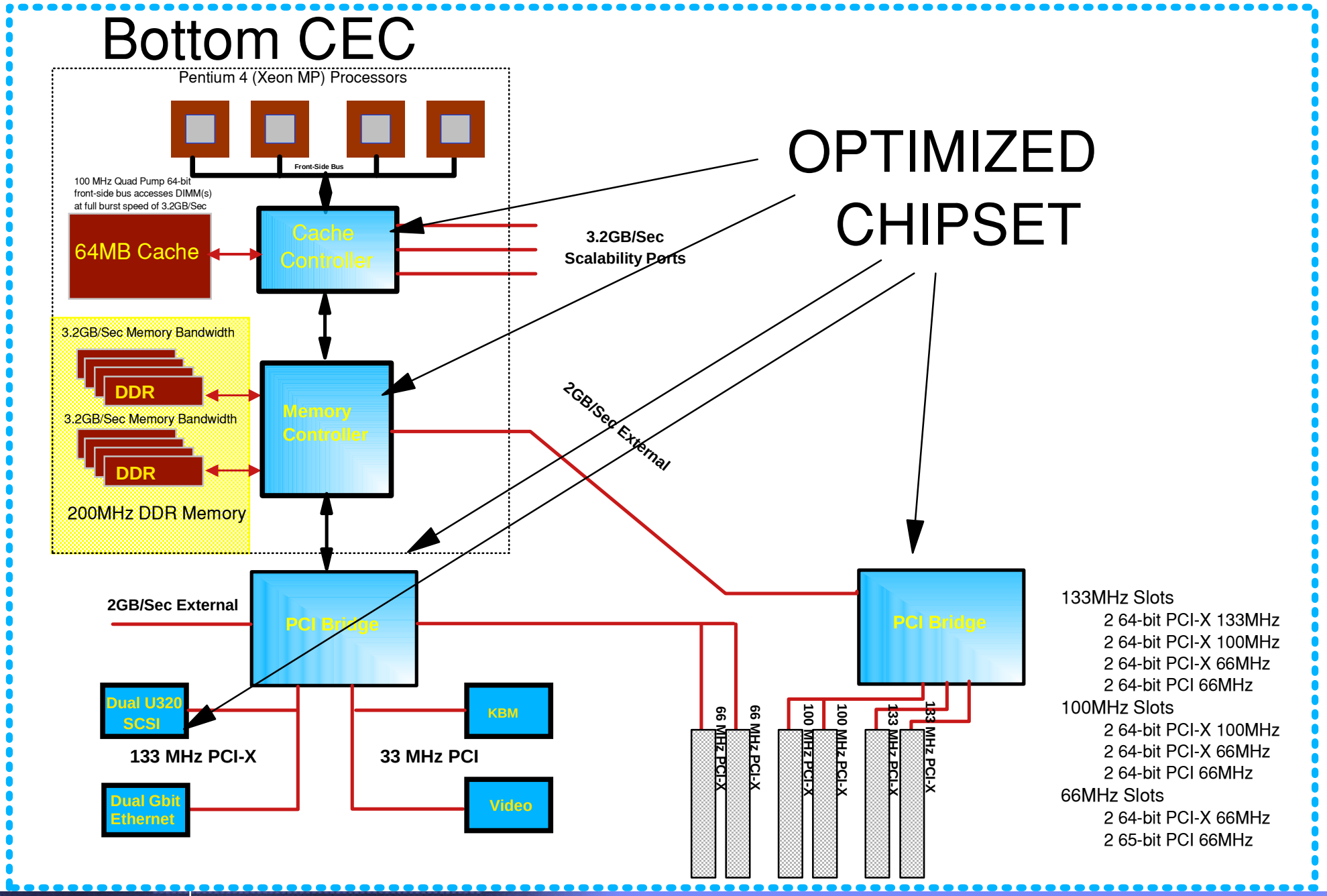
Worse Case Memory Access Time = 600 nSec

Average Memory Access Time = 350 nSec

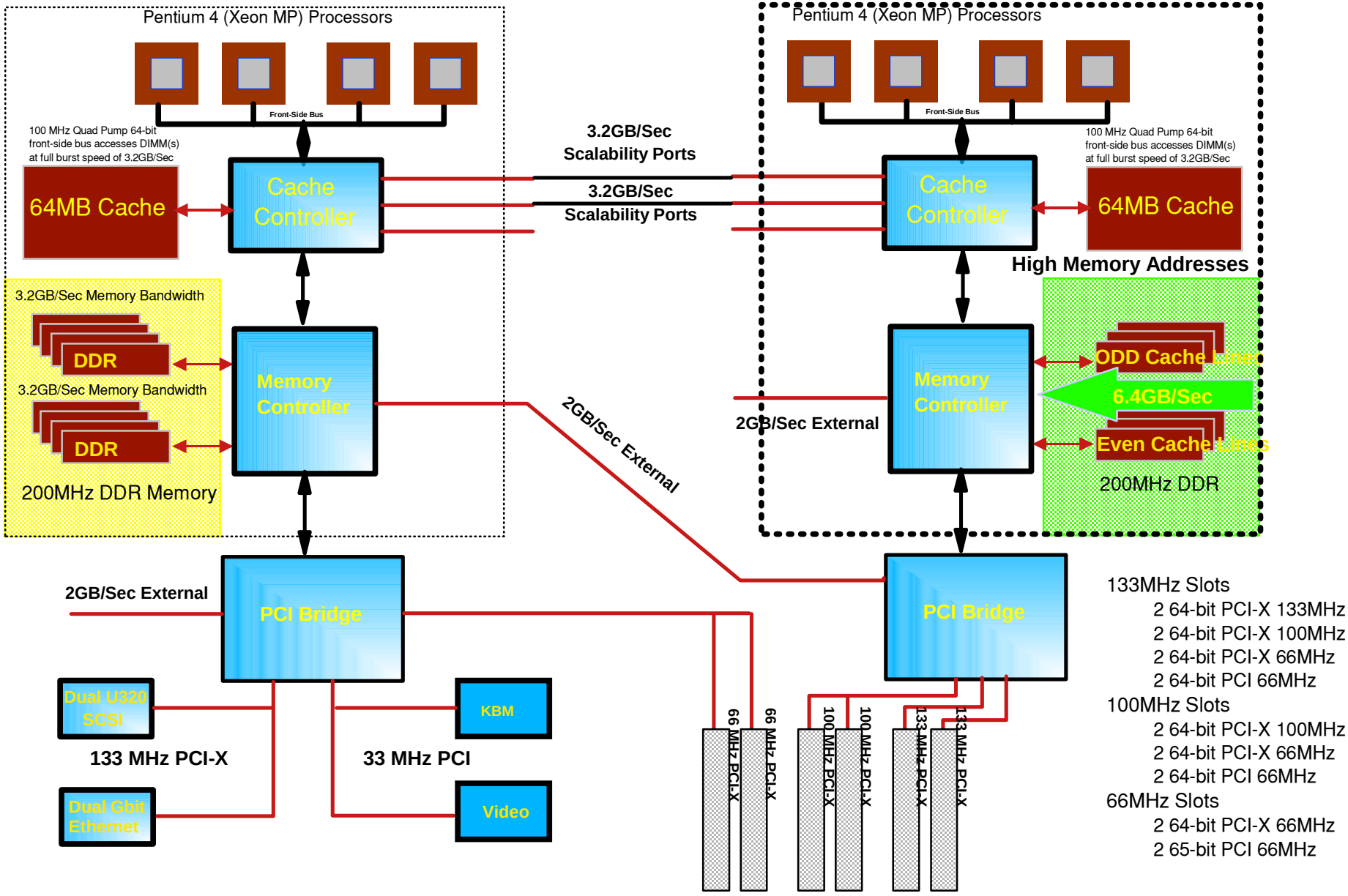
Still 22% reduction over SMP

Memory Controller Observations

- SMP memory controller provides a single path to memory for all processors
  - Same latency for any processor, fastest performance when running a single task or light memory load
- Disadvantage of SMP is single memory controller becomes the bottleneck when multiple, high-speed processors need data and instructions AT THE SAME TIME
  - Single memory controller can become flooded with just four or five busy processors
- Advantage of NUMA is multi-memory controller design shares the load generated by multiple high-speed processors
  - Improves scaling by adding memory throughput as processors are added
  - NUMA performance improves compared to SMP as load increases by reducing queuing delays at the memory controller
  - But can complicate software?



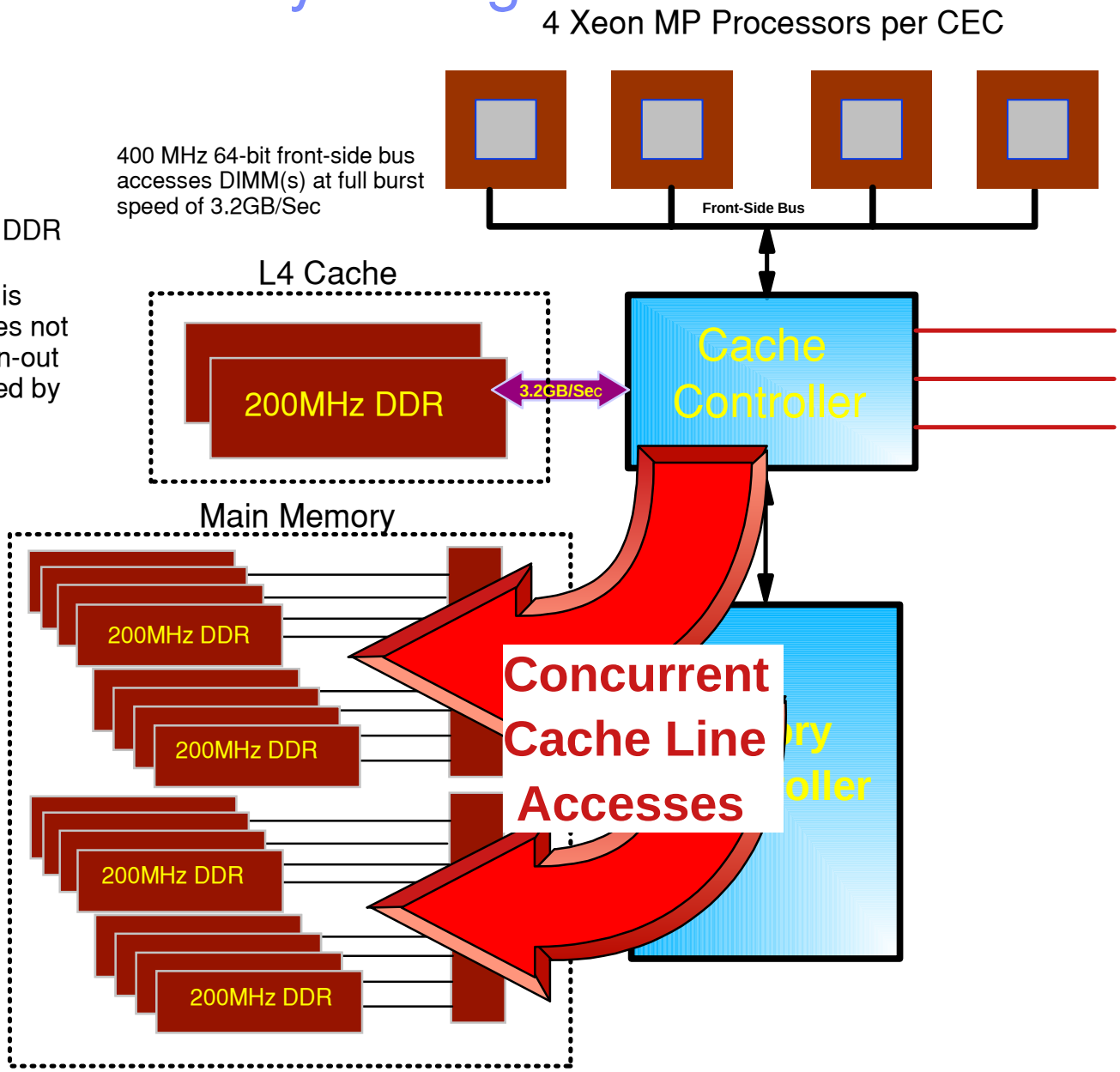
Bottom CEC



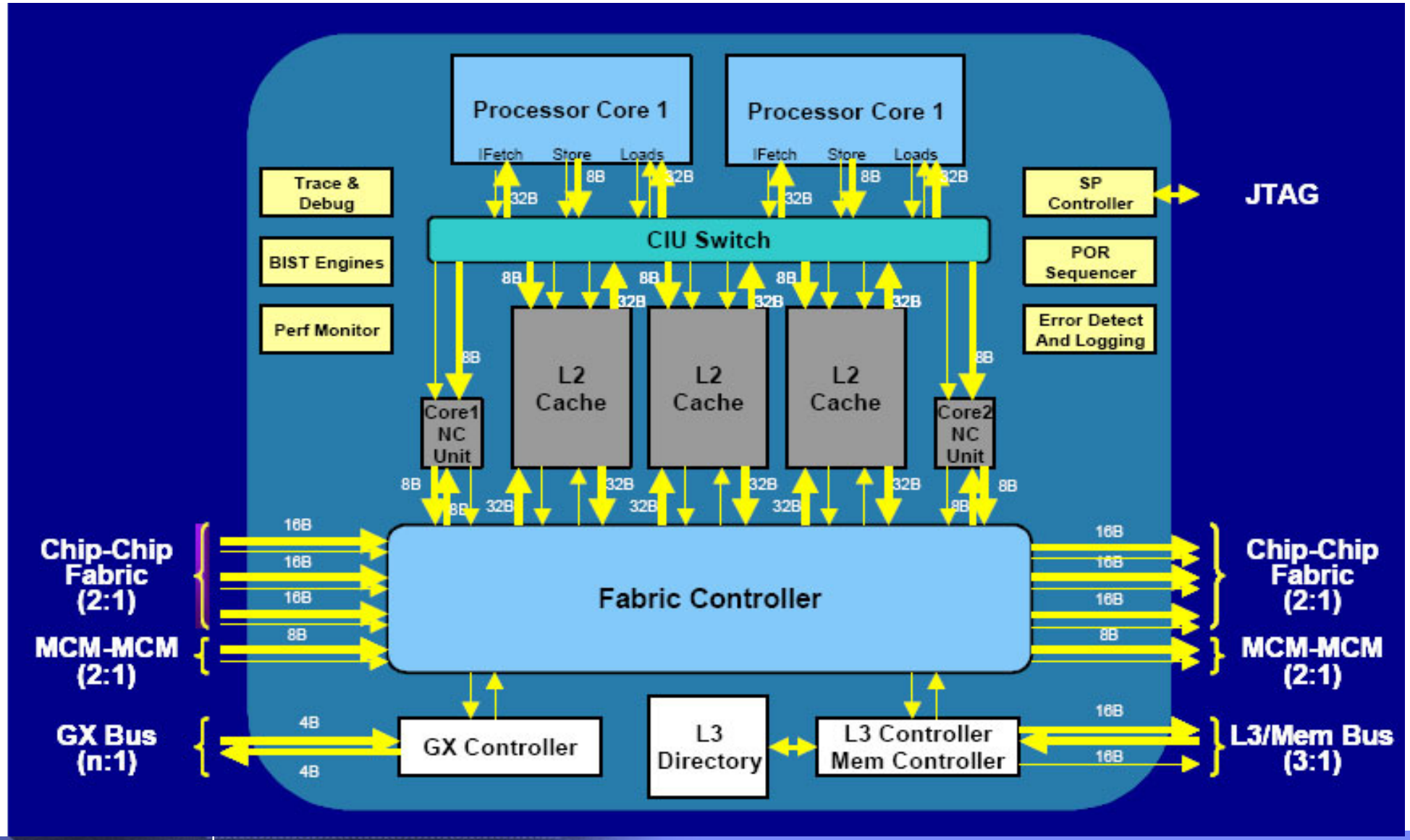
x445 System Internals - Memory Configuration

- Cache memory is two-way interleaved 200MHz DDR
- This is faster than standard memory because it is directly connected to memory controller and does not have additional latency associated with large fan-out necessary to support multiple DIMM slots needed by main memory

- Memory is added in pairs into same bank
- For best performance balance memory across all banks
- 6.4GB/Sec only with multiple of 4 DIMMS (Two ports populated)
- DIMMs within a bank must be the same size
- Different banks may use different size DIMMS
- Banks may be populated in any order
- Drawing implies banks are comprised of adjacent pairs of slots
  - They are actually every other slot



Power4 Processor designed to address scaling bottlenecks and was the model for x440 and x445 CEC



x445 Architecture Key Performance Points

- NUMA is a leading solution to single memory controller bottlenecks and is the future for scalable server solutions
  - Applications running on a single processor caused processor bottlenecks which lead to SMP designs
  - NUMA helps solve the memory controller bottleneck with multiple memory controllers
    - Multiple memory controllers to distribute the memory access load
- While typical SMP have SAME latency for any one access
  - Average latency when all processors access SAME memory controller is very long
    - Longer queuing delays since all processor requests go to the same memory controller and must wait in line for the all memory requests
      - Like a busy airport with only one ticket agent
- Software optimizations in .NET and Linux 2.5 Kernel Affinitized:
  - CPU Threads
  - Memory Allocs
  - Interrupts
  - Soon we will have I/O affinity with Multipath Data Routing

x445 Architecture Key Performance Points

- SMP Interconnect is replaced by coherent scalability port
  - Scalability port runs up to 3.2GB/Sec
    - Same speed as XEON MP front-side bus
  - Can be connected in double barrel mode
    - Two scalability ports connected in pairs for 8-way
      - +5% or more performance improvement measured in lab for tuned environment
      - Compared to single scalability port connection
    - Higher performance gains with standard applications
      - Non-optimized applications would make greater use of scalability ports
- Memory is interleaved dual-channel @ 6.4GB/Sec peak bandwidth
  - Four-way interleaved memory added in quads then mirrored
    - Mirrors are interleaved across ports on odd/even cache line boundary
    - Allows dual ports to service multiple independent memory requests in parallel
    - Reduces memory latency and increases bandwidth

NUMA Software Optimizations

Microsoft Win2003 NUMA Optimizations - Key Points

- Node topology
  - System Resource Allocation Table (SRAT)
    - Automatic node affinity for process and memory
- Memory management
  - Page faults satisfied from home node's memory if available
  - Process - granularity for management
  - System memory structures also NUMA-aware – Paged/Non-paged Pool, Heap
- Scheduling
  - Processes assigned a home node in addition to home processor at initiation
  - Assignment round robin between nodes
  - Thread is the unit for scheduling
  - Scheduler attempts to schedule threads –
    - First on home processor
    - Second on home node
    - Only schedules off-node if home node very busy and other nodes idle
- Node affinity is an inherited property
  - Process-oriented architectures, ported from Unix, show significant performance increase on application servers with little or no intervention (JDE, Siebel, SAP)

Future Systems Will Evolve to NUMA

- IBM is helping lead the industry with Intel NUMA technology
- To address memory controller bottlenecks future systems will evolve to NUMA
  - Scalability of future generation processors will depend upon concurrent access to memory
- Microsoft and Linux Community adapting operating systems for NUMA
  - This will greatly reduce the need for applications programmers to worry about NUMA effects
    - But remember - For commercial applications
      - Memory concurrency is more important than speed!

64-bit Extensions for IA-32 Processors And Memory Scaling

IA-32 Processor Memory Limitations

- Problem for some application is a single process limited to 4GB
  - But a real limit of 2 - 3GB for application without special coding techniques
  - Windows and Linux
  - 2GB mapped to kernel - 2GB mapped to application
    - Windows - Enterprise Ed. Apps can map 1GB to kernel and 3GB to process
- Applications such as SQL Server use AWE to expand buffer space and use up to 64GB
  - But this has limitations
  - Applications must specifically use AWE API
  - Pointer space utilization in kernel memory space
  - Can't execute code in expanded space
- IA-32 needs extensions to 64bit virtual space to grow into future

Intel EM64T

- EM64T is the evolution of IA-32 to full 64-bit virtual memory
  - It is not "DX2" or multiplexed addressing registers
  - **It is full 64-bit instruction pointer, address, and data registers**
- Binary compatible with Opteron AMD-64
  - Objective is one OS and application build for both processors
    - Some minor differences e.g. 3DNOW, SSE instructions, and driver issues
  - Code can do IFDEF and run on both processors
- For applications that are memory intensive EM64T will enable up to 1TB of physical memory
  - Typical apps to see significant improvement
    - Databases
    - Engineering / Scientific
- Performance gains will depend upon how much memory pressure an application generates
  - Typical 4-way server gains are in the 15 - 25% range
  - 16 and 32-way servers generate huge memory pressure and will experience much larger gains in performance
    - 30 - 50% range

EM64T Operating Modes

Legacy Mode

- 32-bit OS
- 32-bit apps
- 32-bit drivers

Compatibility Mode

- 64-bit OS
- 32-bit apps
- BIOS Support
- 64-bit drivers
- 4 GB address space
- GPRs are 32-bit

64-Bit Mode

- 64-bit OS
- 64-bit apps
- BIOS Support
- 64-bit drivers
- 64-bit flat virtual addr space
- GPRs are 64-bit

EM64T Operating Modes

Mode	Legacy 32-bit OS/App	Compatibility 64-bit OS 32 or 64-bit App	Full 64-bit
Address	32/16	32/16	64
Data or Operand Default Size	32/16	32/16	32-bit Default 64-bit Explicit
General Purpose Register Width	32	32	64
Operating System Support	32/16	64	64
Application Changes? E.g. Recompile	NO	NO	Yes

EM64T Checklist

- Intel will have EM64T technology in all future DP and MP processors
  - xSeries first EM64T products start to appear 2H 2004
  - Windows 64-bit operating system support now Dec. 2004 (32-bit support now)
    - http://www.microsoft.com/windowsserver2003/64bit/extended/default.mspx
  - Redhat expect support 2H 2004
    - No official announcement
  - SuSE expect testing and final support 2H 2004
    - http://www.suse.com/us/business/products/server/sles/index.html
- EM64T enables Intel IA-32 processors for true 64-bit computing
  - IA-32e now called EM64T
  - True 64-bit extensions enable higher-performance AND software compatibility for IA-32 processors
  - Binary compatible with existing 32-bit applications

Enterprise X-Architecture™

Industry-Standard Servers for e business on demand

Footnote

Information in this presentation concerning non-IBM products was obtained from the suppliers of these products, published announcement material or other publicly available sources. IBM has not tested these products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Prices do not include tax or shipping and are subject to change without notice. Price may include applicable discounts. Reseller prices may vary. Unless otherwise specified, pricing information is current as of original publication of this document.

MB, GB, and TB = 1,000,000, 1,000,000,000 and 1,000,000,000,000 bytes, respectively, when referring to storage capacity. Accessible capacity is less.; up to 3GB is used in service partition. Actual storage capacity will vary based upon many factors and may be less than stated. Some numbers given for storage capacities give capacity in native mode followed by capacity using data compression technology.

Maximum internal hard disk and memory capacities may require the replacement of any standard hard drives and/or memory and the population of all hard disk bays and memory slots with the largest currently supported drives available.

Notice

Information in this presentation concerning non-IBM products was obtained from the suppliers of these products, published announcement material or other publicly available sources. IBM has not tested these products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Prices do not include tax or shipping and are subject to change without notice. Price may include applicable discounts. Reseller prices may vary. Unless otherwise specified, pricing information is current as of original publication of this document.

MB, GB, and TB = 1,000,000, 1,000,000,000 and 1,000,000,000,000 bytes, respectively, when referring to storage capacity. Accessible capacity is less.; up to 3GB is used in service partition. Actual storage capacity will vary based upon many factors and may be less than stated. Some numbers given for storage capacities give capacity in native mode followed by capacity using data compression technology.

Maximum internal hard disk and memory capacities may require the replacement of any standard hard drives and/or memory and the population of all hard disk bays and memory slots with the largest currently supported drives available.