High Performance Computing in the Multi-core Area

Arndt Bode

Technische Universität München

ISPDC 2007

Hagenberg, Austria

07 July 2007

Technology Trends for Petascale Computing

Architectures:	Multicore – Accelerators – Special Purpose – Reconfigurable – Memory and Cache – Network – Secondary Storage
Computing and Software:	Programming Models – Heterogeneous vs. Homogeneous – Compilers and Threading – Operating Systems – Tools – Adaptivity
Applications:	Scalability

Trends in Microprocessor Technology

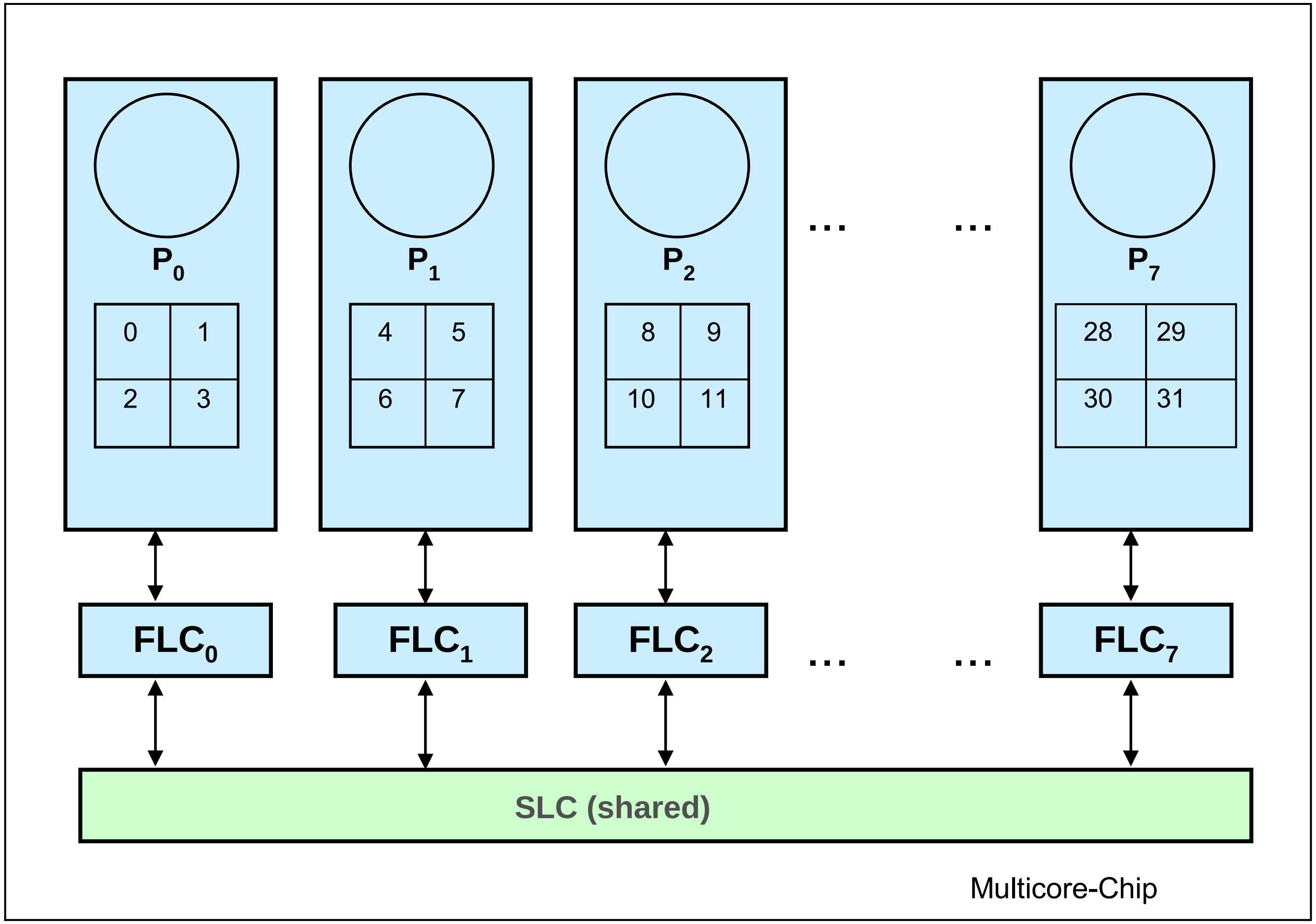
- Compute Performance of COTS Microprocessors depends of
  - Clock Frequency
  - Computer Organization
- Clock Frequency:
  - Exponential Increase impossible (Power, Cooling, 135 W/Chip)
  - Partly Solutions: Clock- and Power reduction, Sleep Transistors, ...
  - New Goal for Optimization: Energy-Efficiency: MIPS, MFLOPS per W
- Computer Organization: ILP, Processor-internal Organization fully exploited
  - Pipelining, Superscalar, VLIW, Wordlength, ..., contra-productive to Energy-Efficiency (Speculation)
  - Future is Processor/Thread level-Parallelism (Parallelism of Simpler Units)
  - Multi-Core and Application-specific Processors , Fault Tolerance

Energy - Efficiency

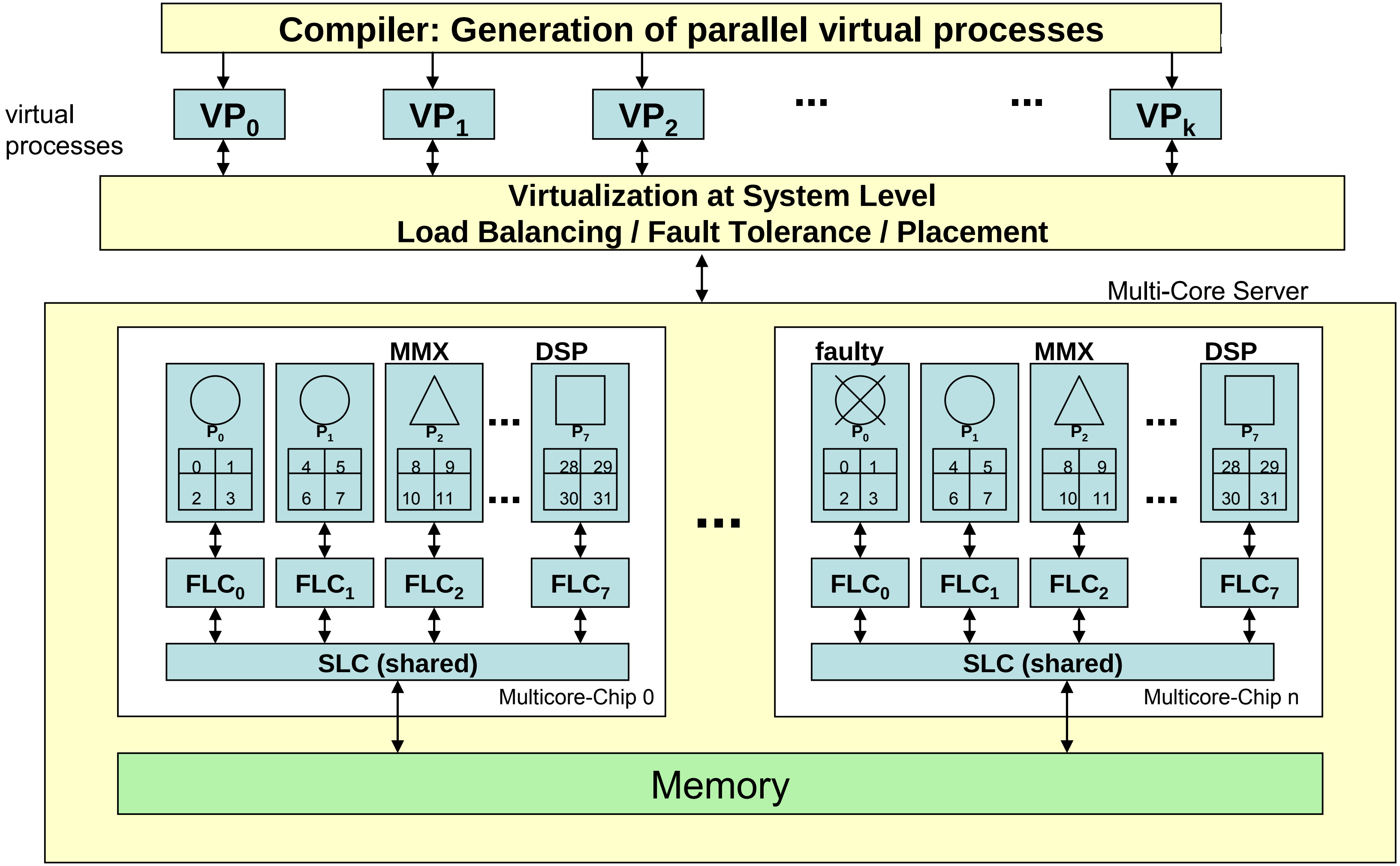
$P \sim A\,C\,V^2 f$

P:	Power
A:	Activity Factor (Active Transistors on Chip)
C:	Total Capacity
V:	Voltage
F:	Clock Frequency

Multi-Core and Multithreading



Multi-Core Architectures: A Parallel World for Everybody



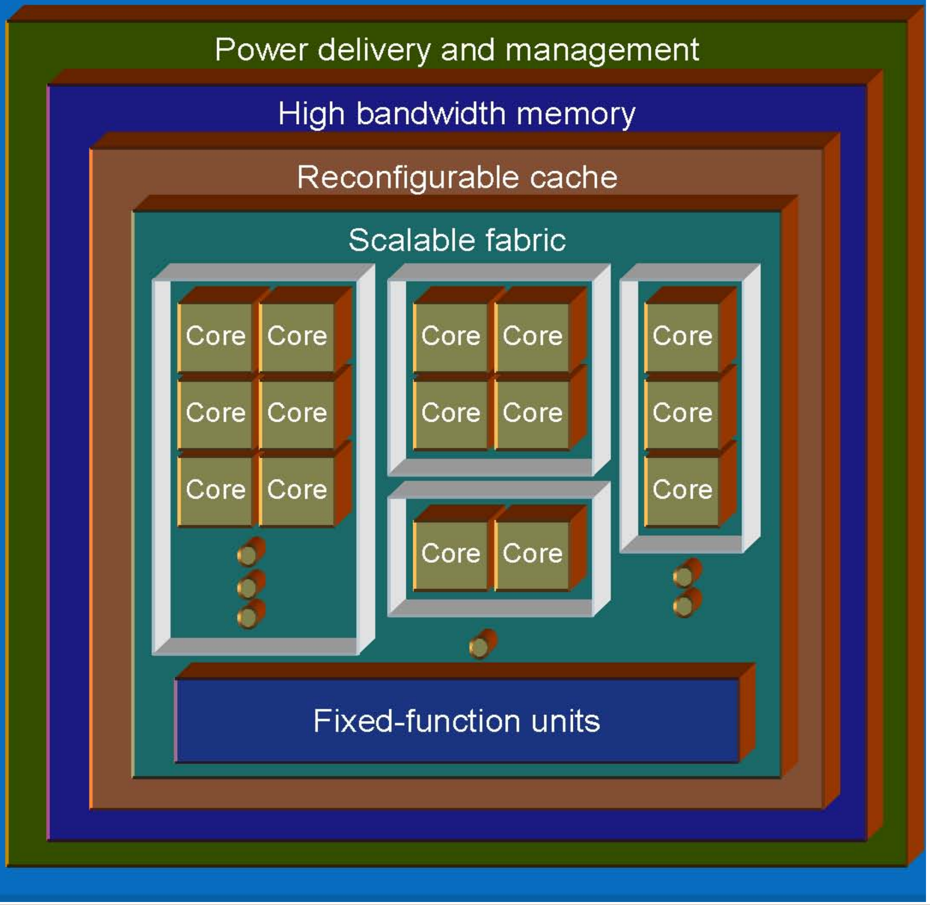
Platform Architecture

Parallel extension of IA

- Homogeneous array of cores
- Fixed-function units
- Coarse- and fine-grained data- and thread-level parallelism
- Global coherency hardware

Partitioned array

- Application domains
- Isolated communication traffic
- Fault tolerance



Processor-Design Options for Petaflop Systems

- Mega-Multicore Systems: General Purpose Applications (Itanium, Xeon, Power-7, ...)
- Multicore and Attached Accelerators (Cell, ClearSpeed, NVIDIA, ...)
- Special Purpose (BlueGene, QCD, POLARIS, ...)
- Reconfigurable and Adaptive (FPGA: continuous research)

General Purpose	vs.	Special Purpose
- Programmability		- High Performance
- Scalability		- Low Power
- Applicability		

History: Special Purpose „volatile“, Programming Interface not Compatible

Microprogrammable Devices	: Vertical Migration and Bitslice
Early Microprocessors	: Coprocessors (FP, IO, DSP, ...)
Early HPC	: Vector Processor Attachments, Array-Processors, Associative Processors

Petaflop Processor Options: Bode’s Oracle

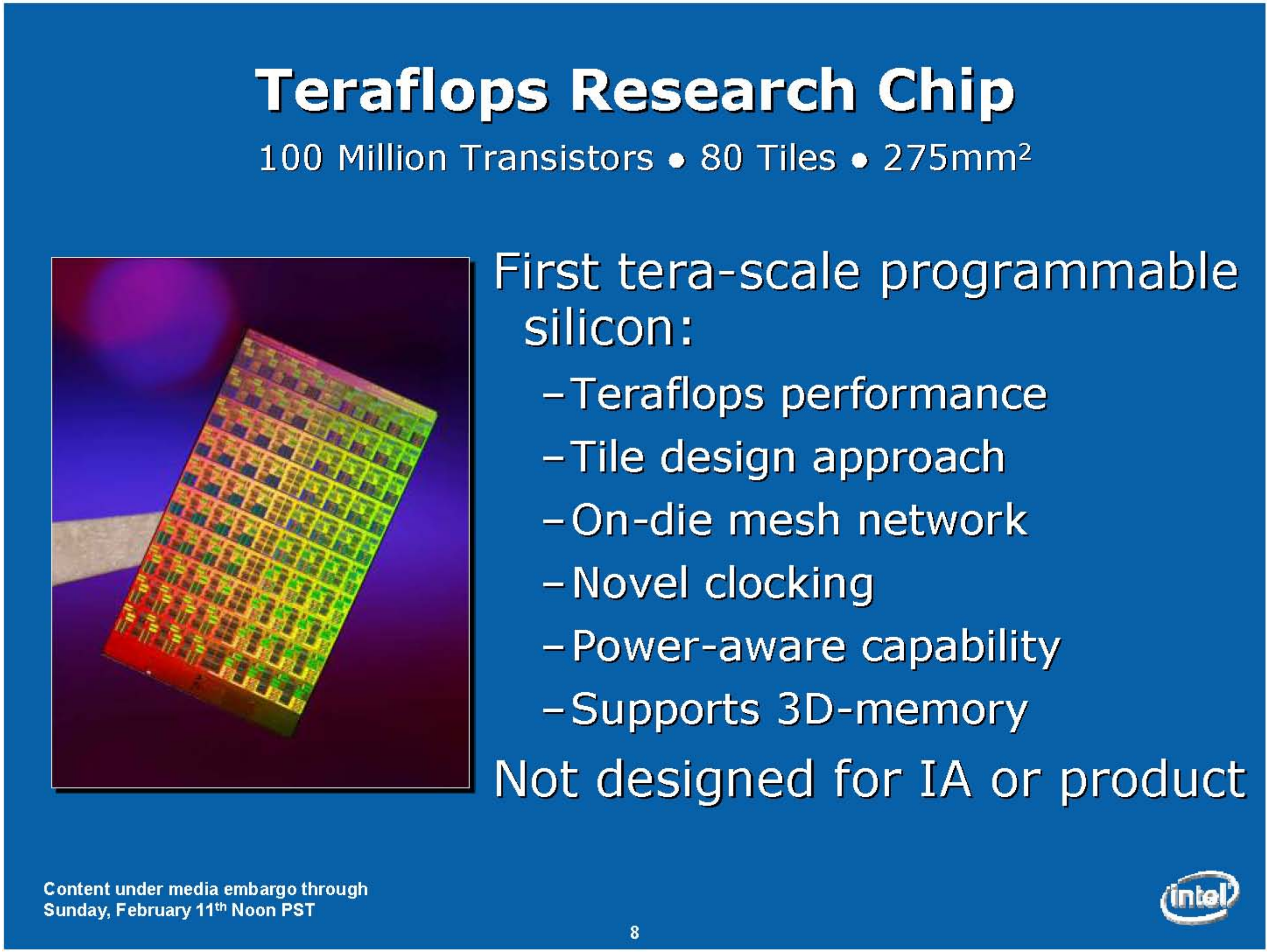
Two Types of Systems will persist:

- General Purpose Mega-Multicore (Programmability, Compatibility, Cost)
- Special Purpose Systems for Large Specific Application Classes (Energy Efficiency)

The Systems will be Highly Parallel

Thomas Sterling: Multicore and Heterogeneous is Disruptive Technology

Tsugio Makimoto: Makimoto’s Wave predicts “Customized”
for 2007 – 2017 (Pendulum: 10^{14} km)



Teraflops Research Chip

100 Million Transistors • 80 Tiles • 275mm²

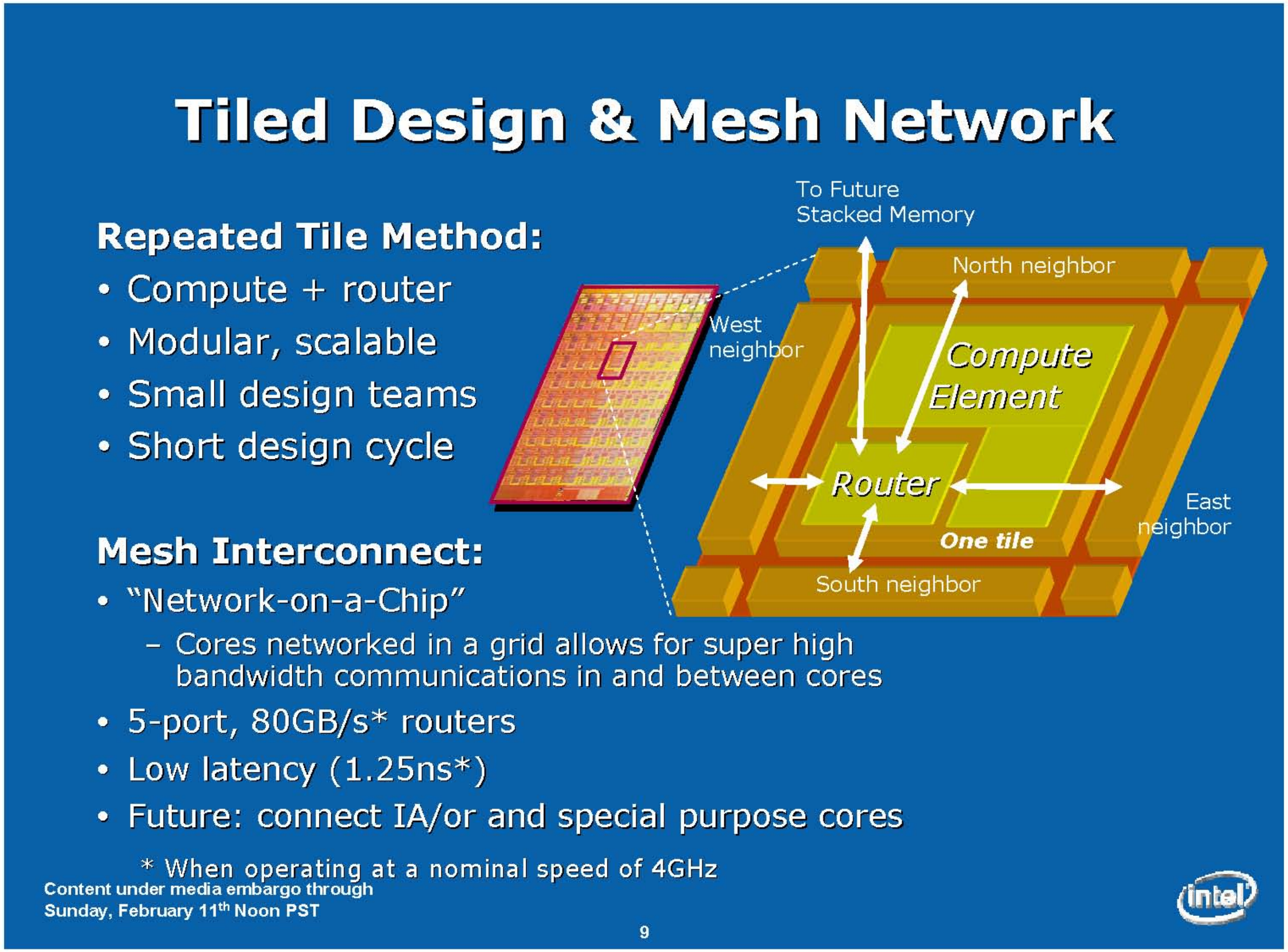
First tera-scale programmable silicon:

- Teraflops performance
- Tile design approach
- On-die mesh network
- Novel clocking
- Power-aware capability
- Supports 3D-memory

Not designed for IA or product

Content under media embargo through
Sunday, February 11th Noon PST

8



Tiled Design & Mesh Network

Repeated Tile Method:

- Compute + router
- Modular, scalable
- Small design teams
- Short design cycle

Mesh Interconnect:

- "Network-on-a-Chip"
  - Cores networked in a grid allows for super high bandwidth communications in and between cores
- 5-port, 80GB/s* routers
- Low latency (1.25ns*)
- Future: connect IA/or and special purpose cores

* When operating at a nominal speed of 4GHz

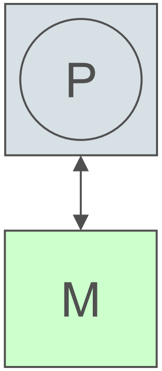
Content under media embargo through
Sunday, February 11th Noon PST

9

Consequences for Computer Architecture

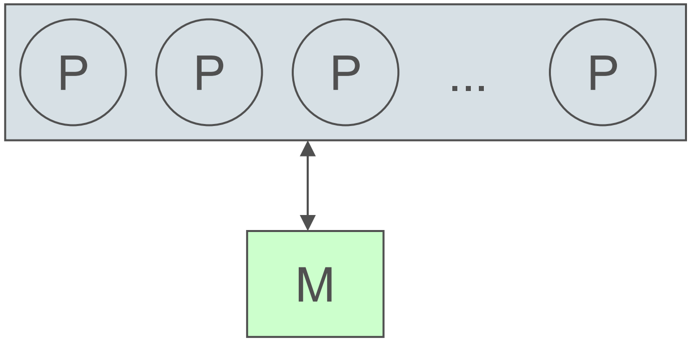
Memory-Bottleneck

Today:



Cache-Hierarchy

Multi-Core:



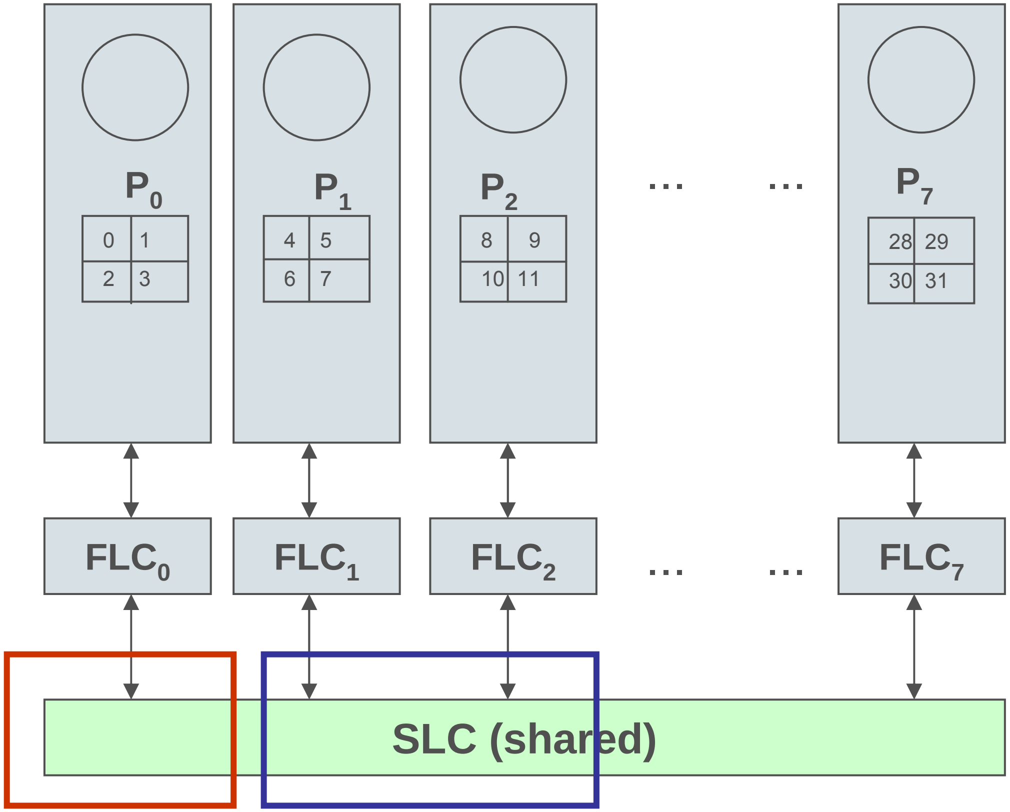
Bottleneck intensified

Cache Organizations

Distributed

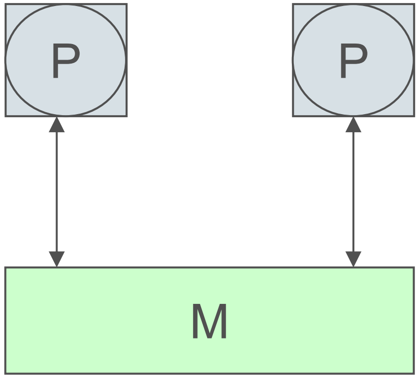
Shared

Groupwise shared



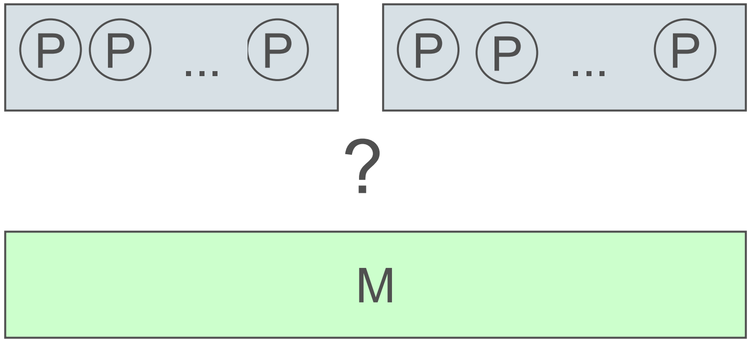
Communication Bottleneck

classical SMP:



1 Interface / Proc. IC

SMP based on Multi-Core



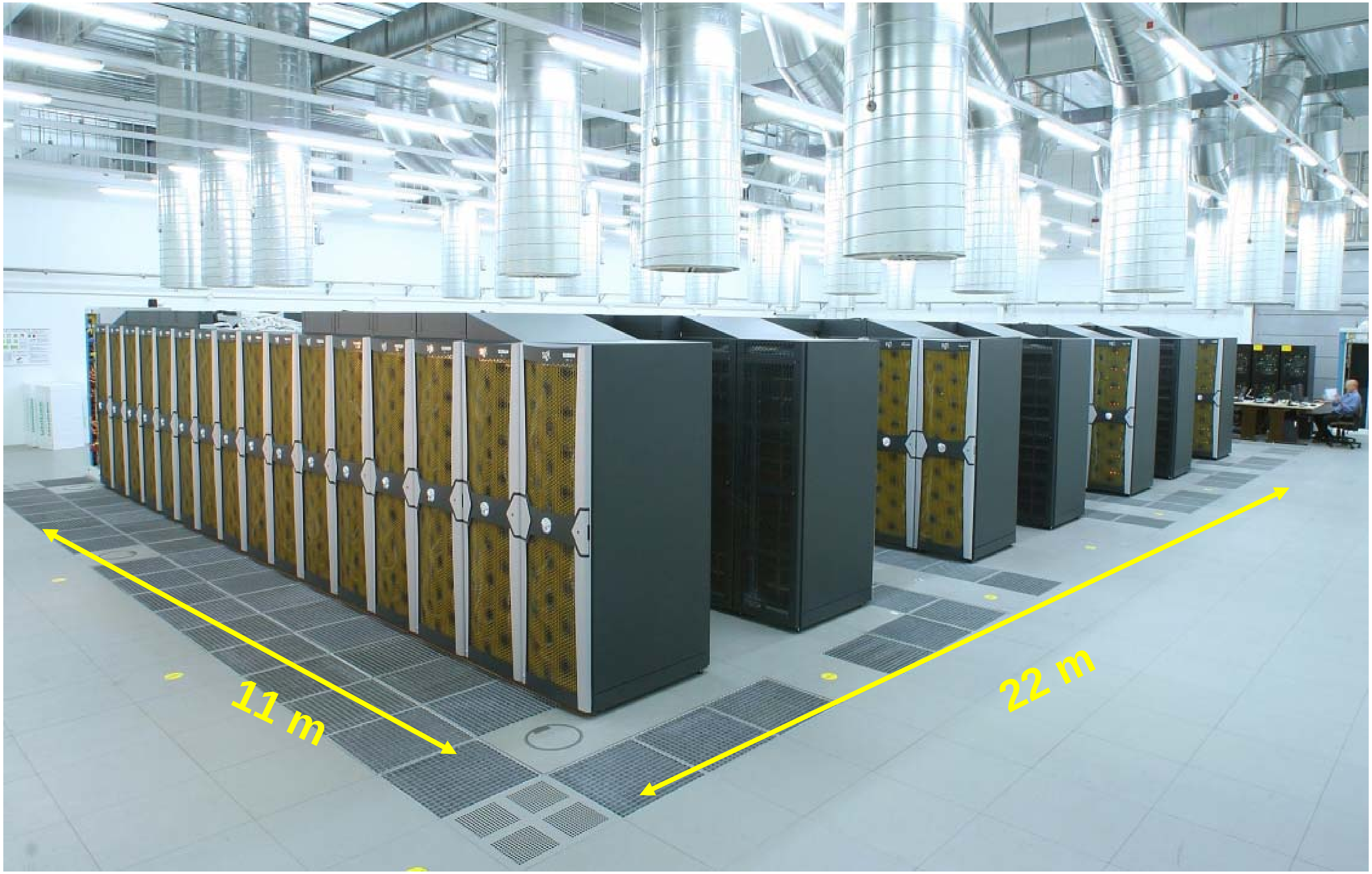
n Interfaces / Multi-Core IC

(e. g.: n cores

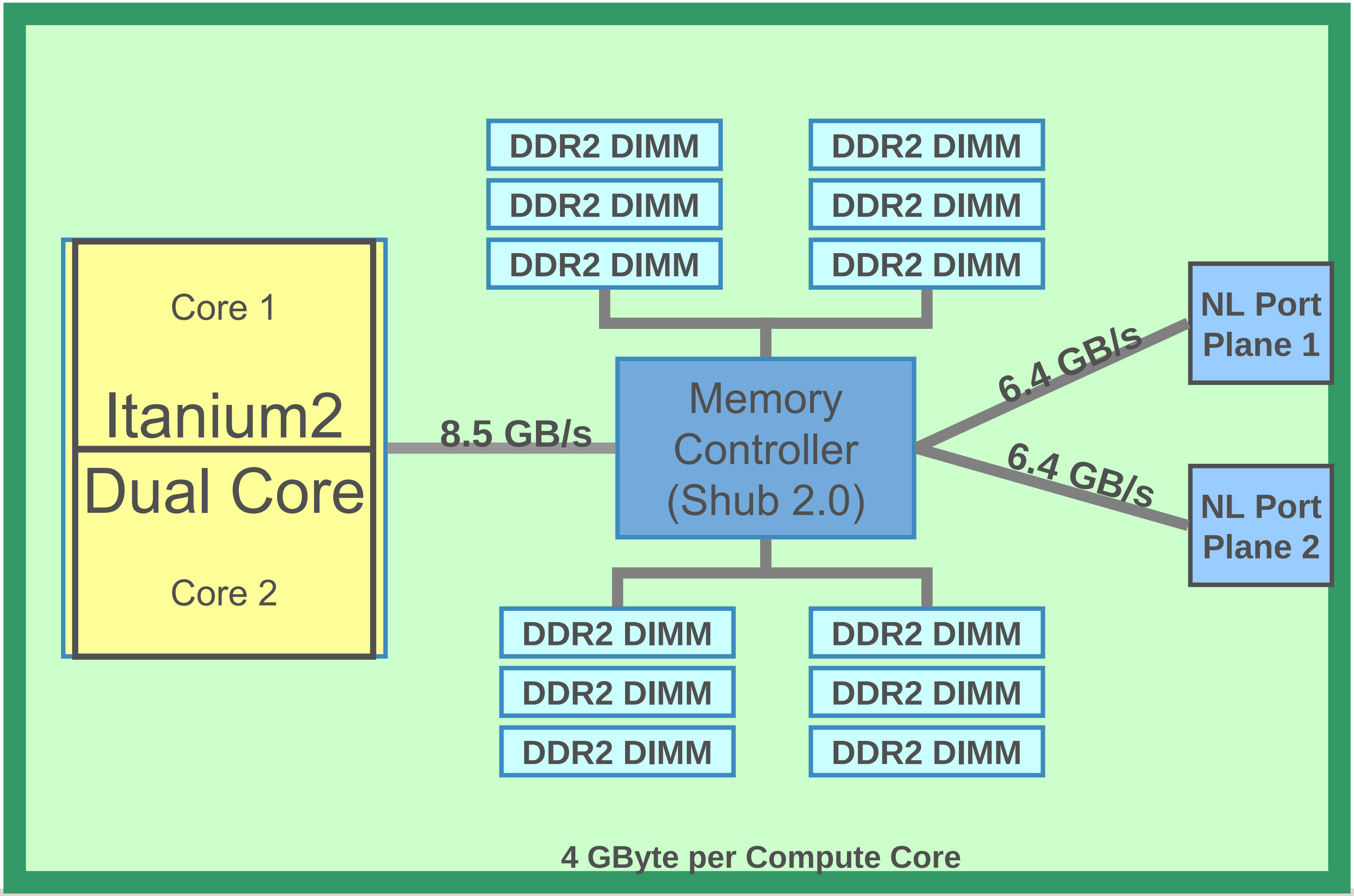
5n Interfaces

mesh-structure)

HLRB II: SGI ALtix 4700 / 9728 Montecito cores

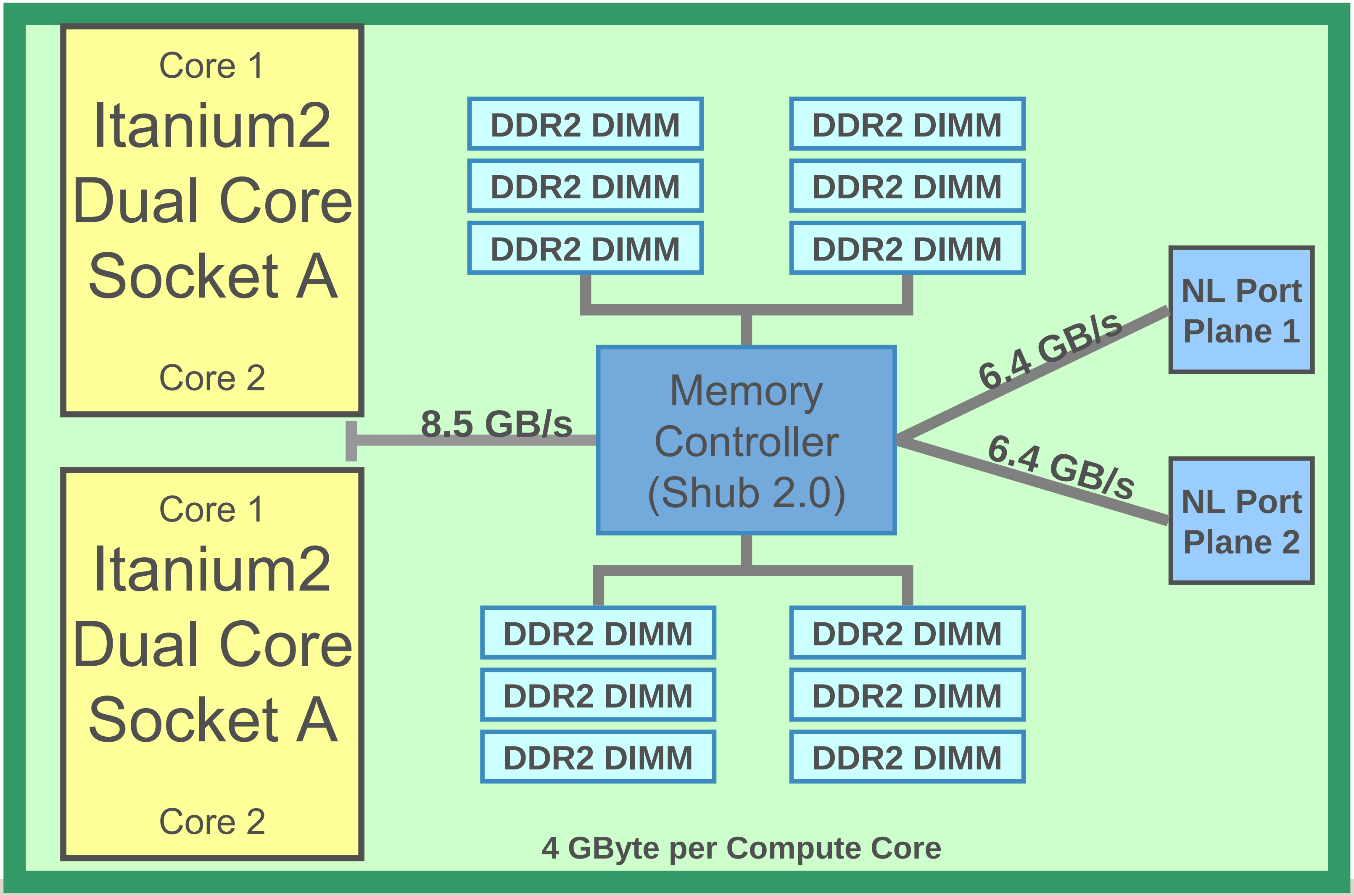


Blade



4 GByte per Compute Core

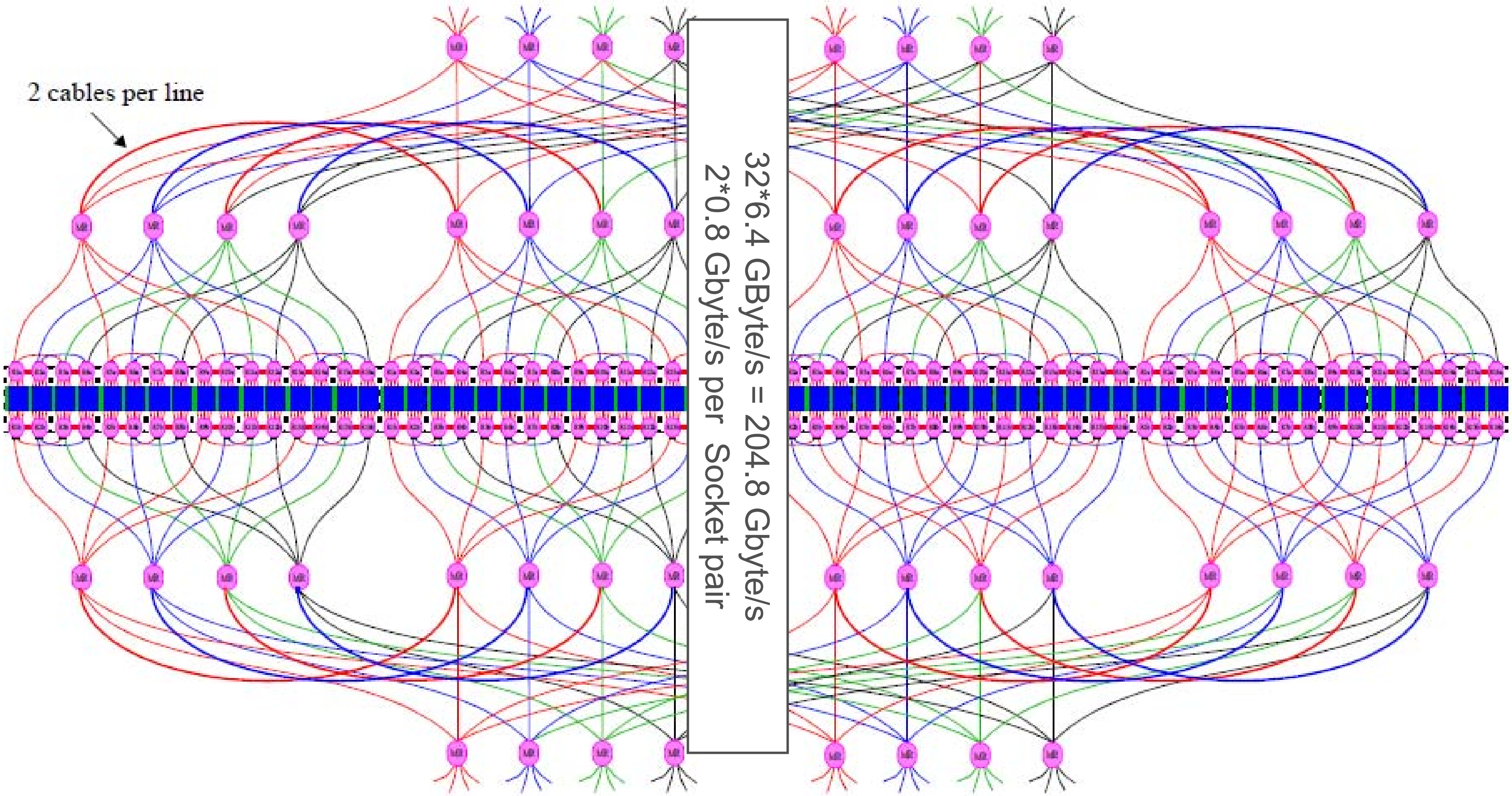
Dual Socket Blades - Density Compute Blades



4 GByte per Compute Core

One 256 Socket Partition: ccNUMA Shared Memory

32*6.4 GByte/s = 204.8 Gbyte/s

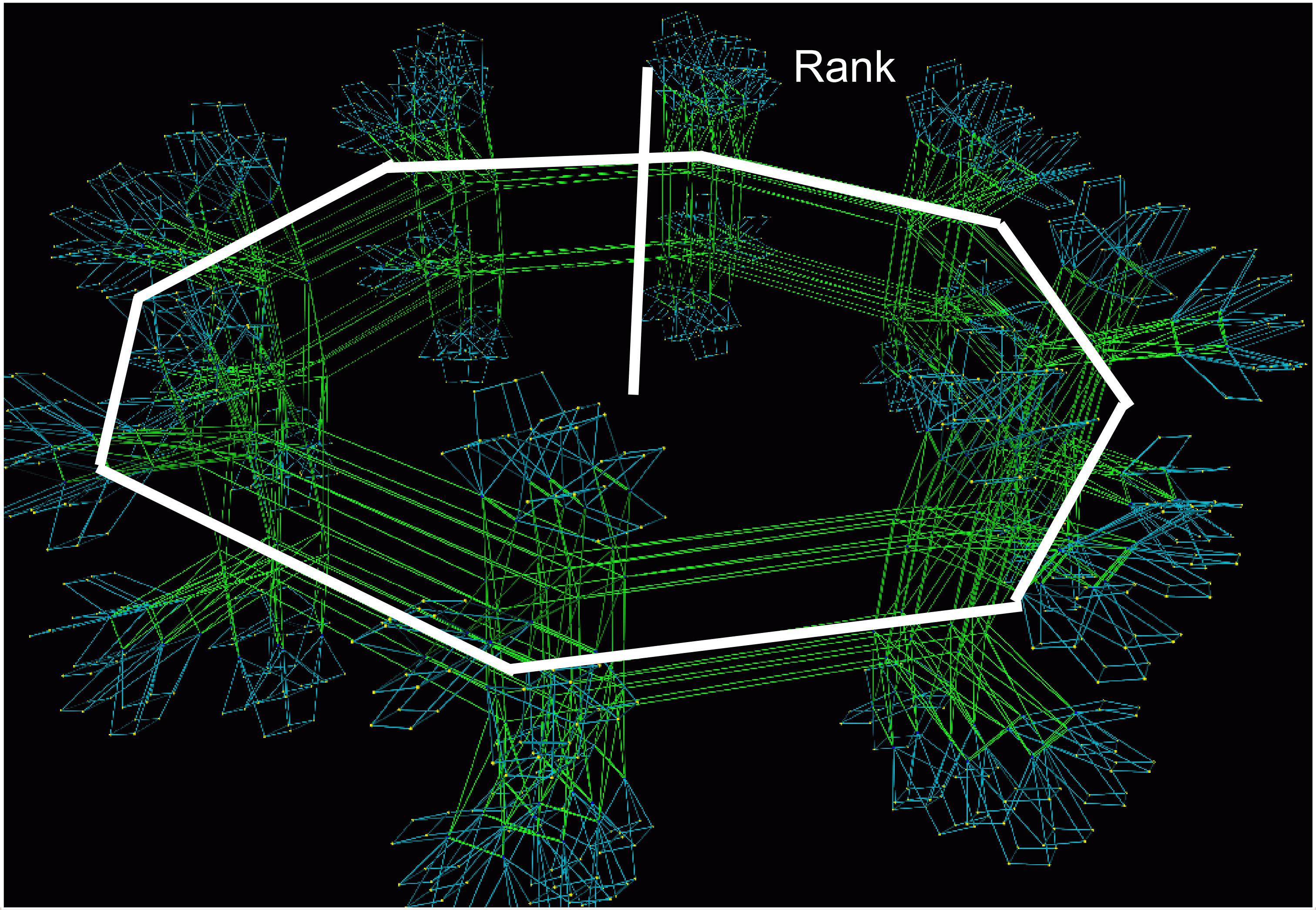


32*6.4 GByte/s = 204.8 Gbyte/s

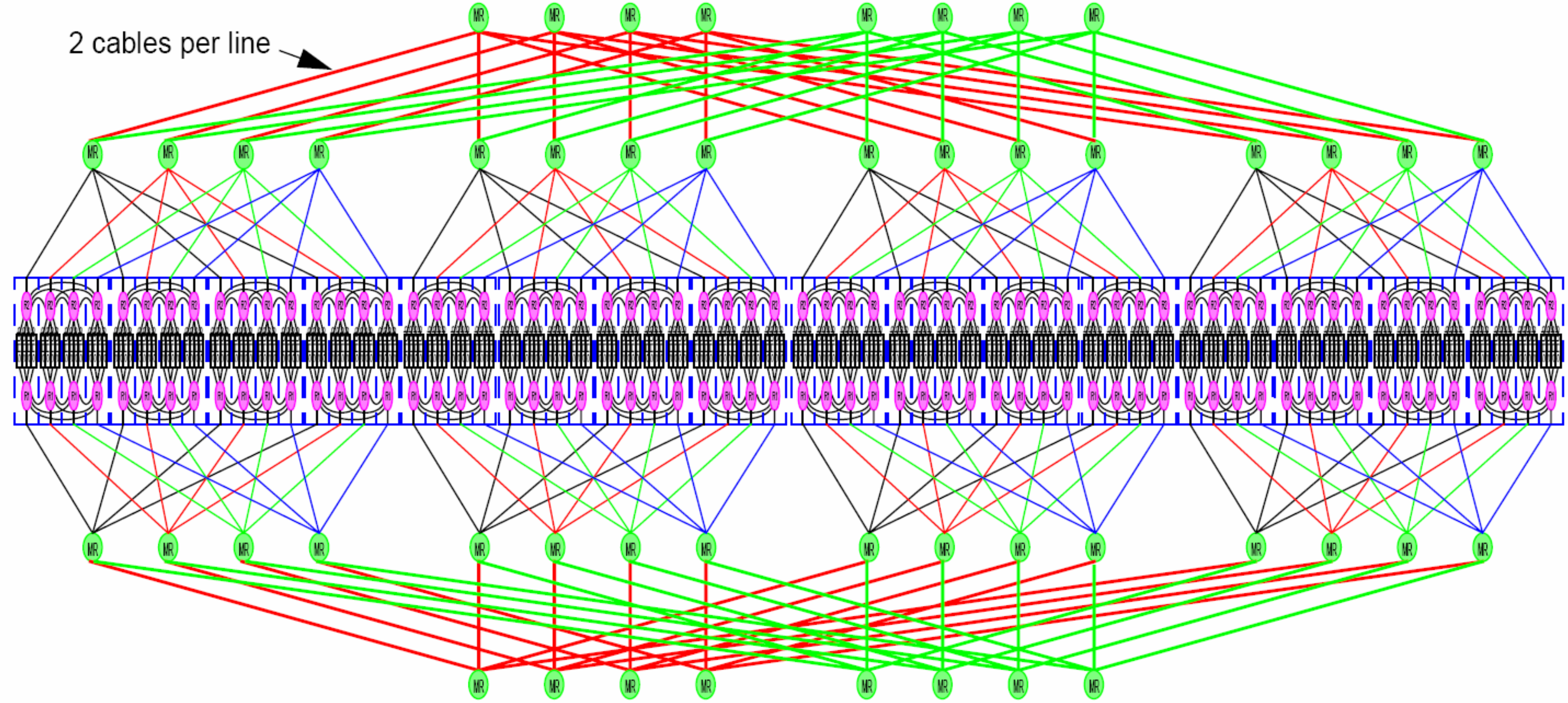
2*0.8 Gbyte/s per Socket pair

32*6.4 GByte/s = 204.8 Gbyte/s

2-D Torus Mesh topology.

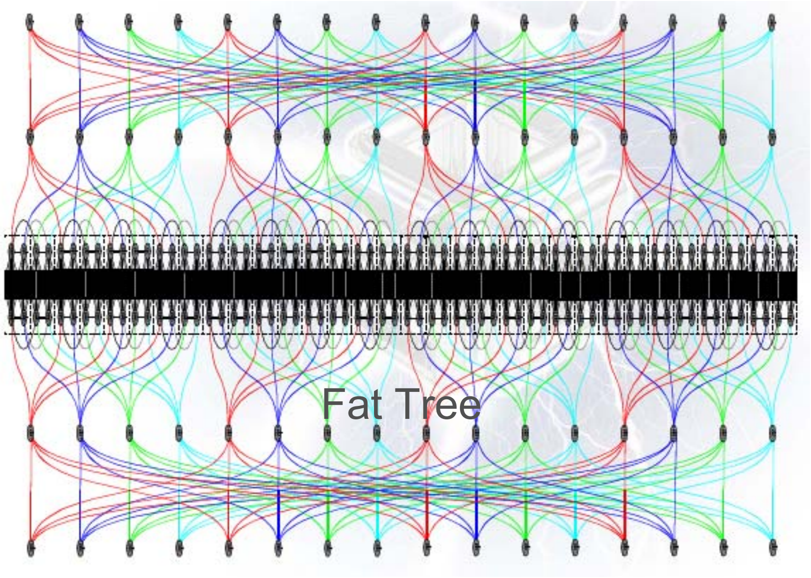


HLRB II Interconnect



Fat Tree Topology

Topology

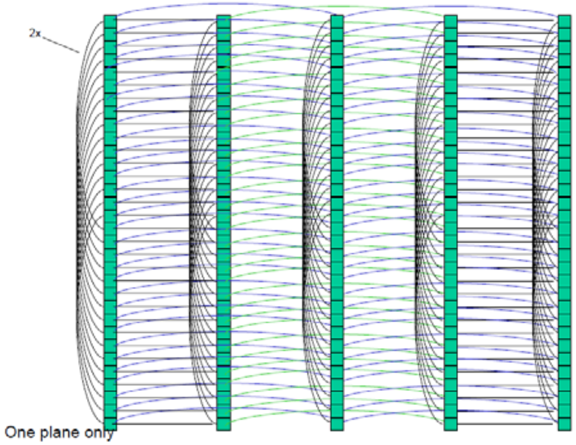


Batch Scheduler must be topology aware

PBSpro 8.0 early access

Placements of jobs is optimized according to topology

Topology is stored in placement sets



2D-Torus

Petascale Software

Key Issue is use of Excess Parallelism:

- Programming Model, Language and Compiler
  - Conventional
  - Parallelizing (Including JIT)
  - Parallelizing with Directives
  - New Parallel Languages (Transactional Memory, ...)
- Use of Threads
  - Functional
  - Speculative
  - Assist/Helper (Prefetching, Monitoring, Debugging, Tools, Virtualization, Security, FT-Lockstep, ...)
- Scalable Tool Models

You tell us!

- This is the Academic Forum, so you tell us.
- What are the solutions?
- Where are the new language ideas?
- Can you design a statically checked race-free language which is useful?
- Can naïve users really use functional languages?
- Any language which talks about threads is too low level for most users. So how do we raise the language level?
- Should we be doing message passing inside the node?
  - It's the only demonstrated way to achieve high scalability
  - Do we really need to bring back Occam? ☺

James Cownie Intel Academic Forum, Budapest, 13.06.2007

Some Examples of Research at MMI

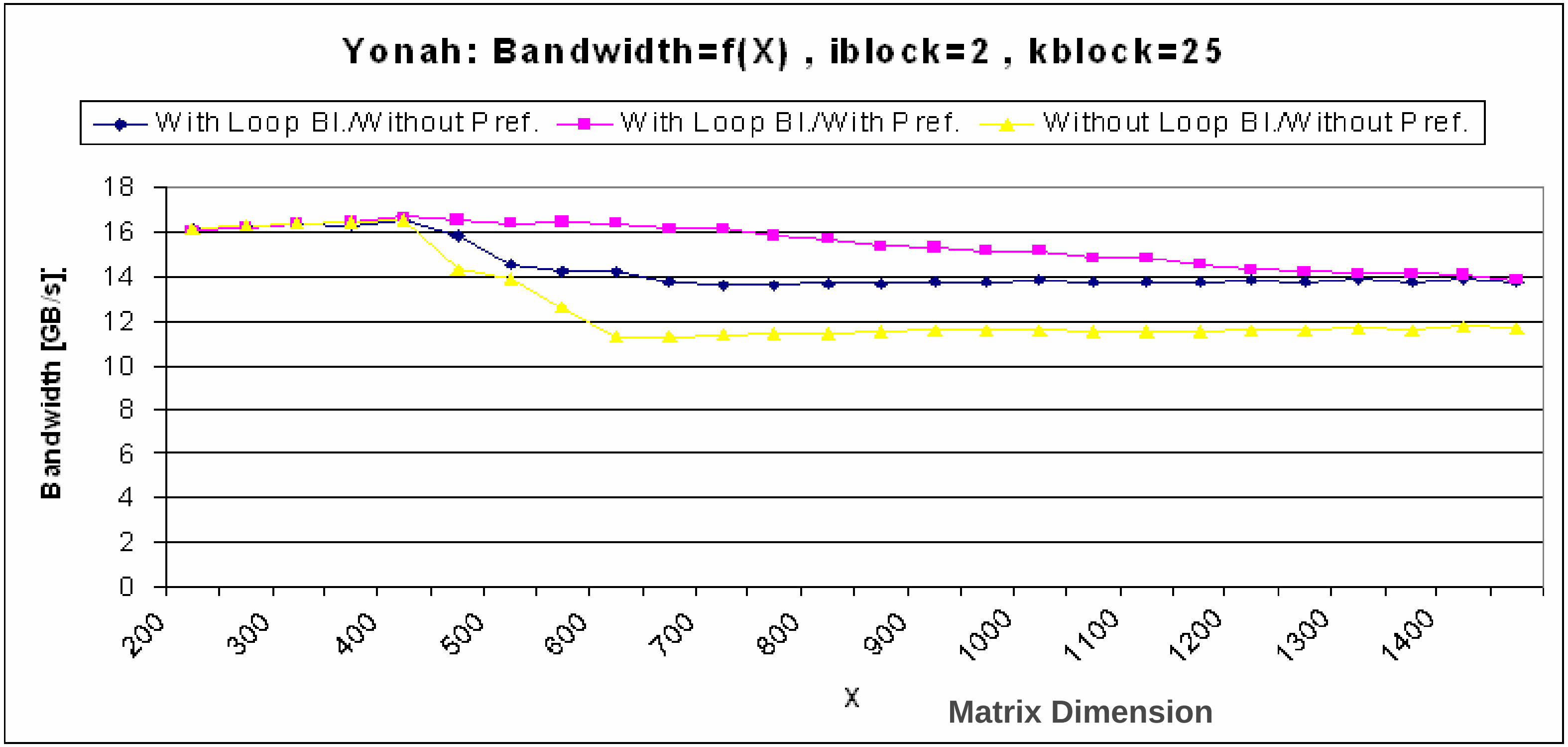
MUNICH
-CORE
INITIATIVE

- Cache-Prefetching with Helper-Core (up to 39 %)
- Cache-Behavior with shared/separate Caches
- Helper Cores for Fault Tolerance

Prefetching

Bandwidth obtainable for Matrix-Multiplication

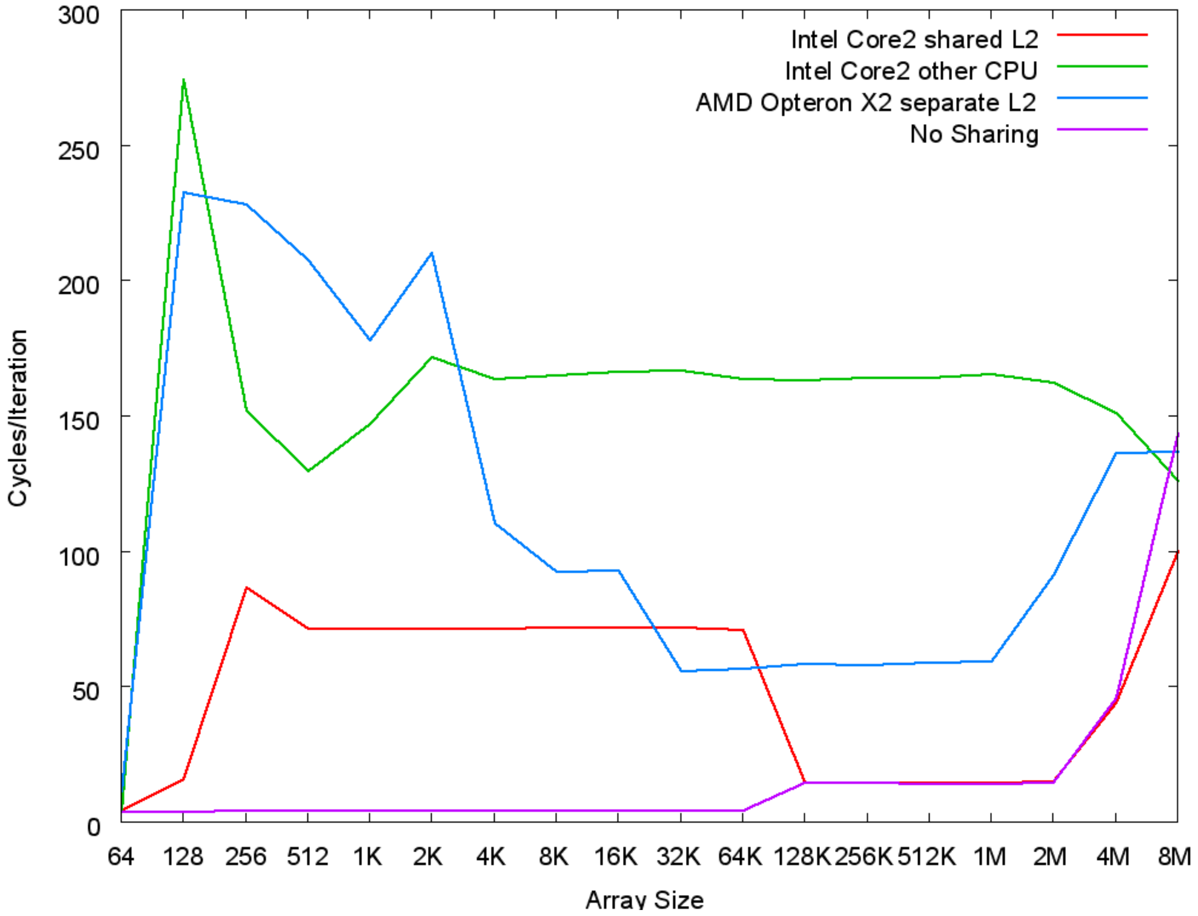
(Blocking with small amount of caching)



Advantages of Shared Caches

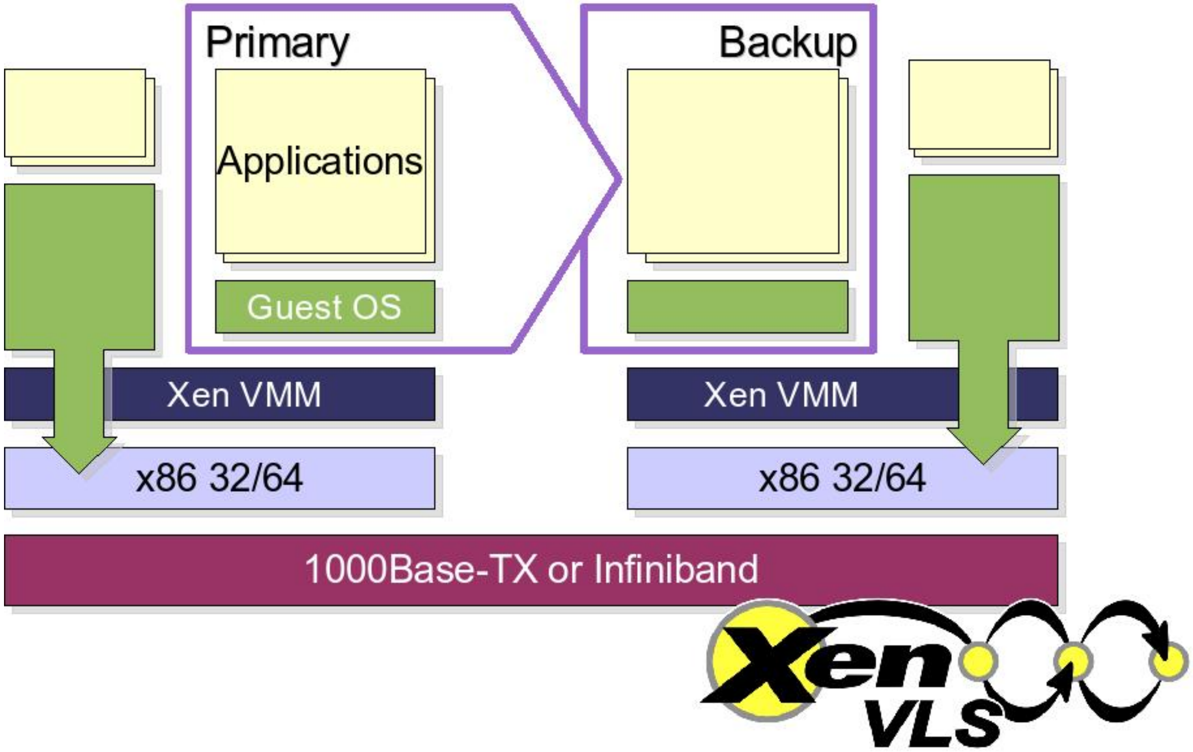
Synthetic Benchmark:

Latency for continuous
Writes onto the same
Memory area (2 cores)

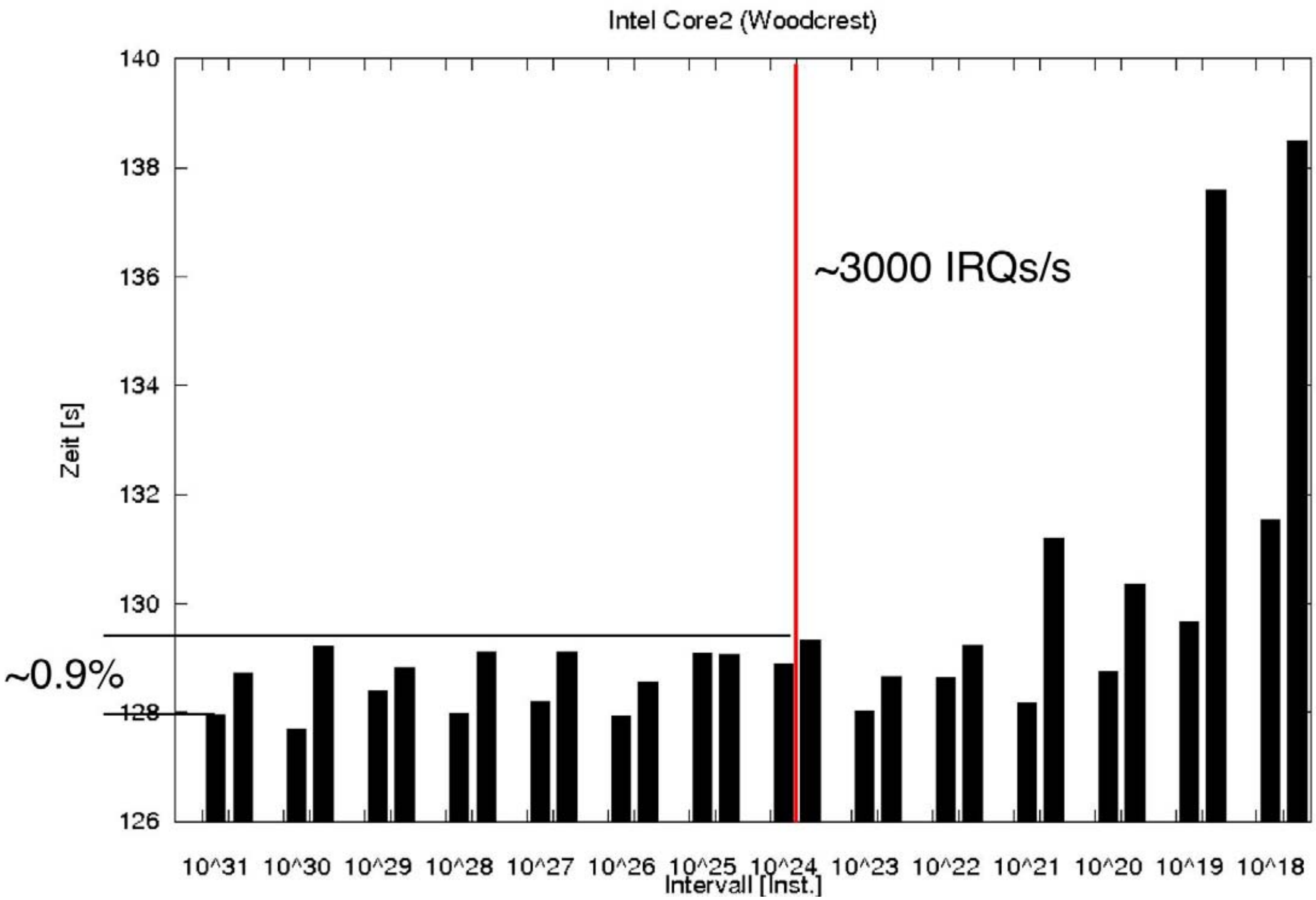


Lockstepping of Virtual Machines

- Synchronized Identical Execution of Applications on 2 Processors
- Virtual Environment allows for Monitoring and Control by Second Core
- Minimal Performance Loss



Lockstepping of Virtual Machines



Bavarian Graduate School of Computational Engineering
Elitenetzwerk Bayern

Welcome to the Bavarian Graduate School of Computational Engineering

Computational Engineering ...

refers to all activities in engineering that use computers as their main tool. Typical tasks in Computational Engineering are the solution of differential equations that model certain physical phenomena, the optimization of processes in engineering, or the stochastic simulation of a complex system.

The Bavarian Graduate School ...

of Computational Engineering is an association of three Master programs:

- **Computational Engineering (CE)** at the Friedrich-Alexander-Universität Erlangen-Nürnberg,
- **Computational Mechanics (COME)**, and
- **Computational Science and Engineering (CSE)** at the Technische Universität München.

Our goal is to push forward the rapidly growing field of Computational Engineering by offering high-quality master programs for students who are interested in the field.

The Elitenetzwerk Bayern ...

is an initiative of the state of Bavaria to support the education and advancement of highly talented students. With the help of the Elitenetzwerk Bayern, we are able to offer an "elite" degree program for the best students in our master programs. Outstanding performance in one of the three Master's programs will be honoured by the newly formed academic degree of a **"Master of Science with Honours"**.

For further information about these elite program, please read on in our section **"Infomation for students".**

IGSSE

BGCE is a partner of **IGSSE**, TUM's International Graduate School of Science and Engineering

Summary

- Multicore (and Heterogeneous): Disruptive Technologies
- Many Choices in System Architecture and Programming Models
- HPC Systems will be massively parallel: Scalability is Challenge for Application-Algorithms, System Software, Tools and Architectures
- Interesting Times Ahead!