Break Free: General-Purpose Multi-Core Architectures

Justin Rattner
Intel Senior Fellow and Director
Corporate Technology Group

intel

Agenda

- Architectural crossroads
- Challenges and potential solutions

Will Cell Multiply?


BusinessWeek online

AUGUST 25, 2005

NEWS ANALYSIS

By Arik Hesseldahl

Cell: A Chip That's Going Public

The processor already powers PlayStation 3. But IBM, Sony, and Toshiba will release its technical details to stimulate new uses


THE ECONOMIC TIMES

IBM eyes more mkts for 'cell' chip

Clip

REUTERS [FRIDAY, SEPTEMBER 16, 2005 09:30:54 AM]

SAN FRANCISCO: Two years ago, analysts were calling on IBM to exit the microchip business after production problems and lower-than-expected orders led to losses of more than $1 billion since 2002 at the division.

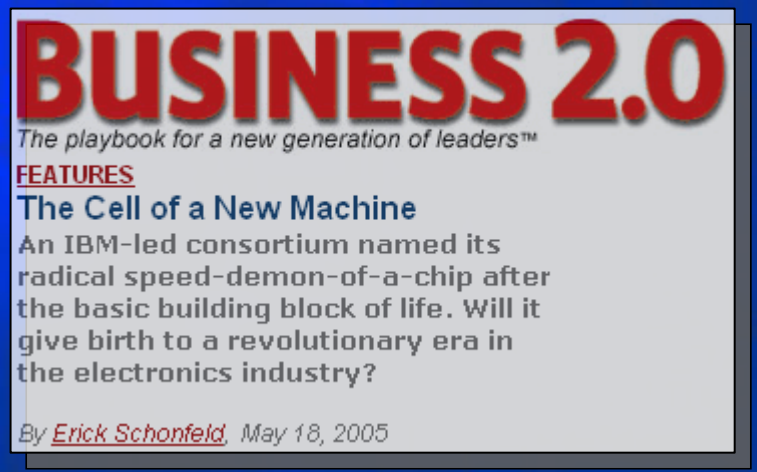

FORTUNE

GIVING TO GET MORE

The 9-in-1 Wonder Chip

Built for games, IBM's mighty Cell chip could help reshape all of computing.

By David Kirkpatrick


BUSINESS 2.0

The playbook for a new generation of leaders™

FEATURES

The Cell of a New Machine

An IBM-led consortium named its radical speed-demon-of-a-chip after the basic building block of life. Will it give birth to a revolutionary era in the electronics industry?

By Erick Schonfeld, May 18, 2005

intel

GPUs Branching Out

the INQUIRER
News, reviews, facts and friction

Nvidia warns multicore CPUs could stiff innovation

Is the GPU the real brain of the computer

By INQUIRER staff: Wednesday 17 August 2005, 15:07

GPGPU

apcmag.com: article

Mon

Hijacking the GPU

Thursday 11, August 2005 | By Dan Warne
From: Epinions > General

From APC Magazine: GPU hackers are turning the supe
video cards against Intel. Dan Warne investigates.

Example: Black-Scholes options pricing

- Widely-used model for pricing call/put options
- Implemented in ~15 lines of Cg, use combinations of input parameterss as separate simulations (fragments)
- Perf:
  - Fast (~3GHz) P4, good C++: ~3.0 MBSOPS (1X)
  - Quadro FX 3000, Cg: ~2.8 MBSOPS (~.9X)
  - Quadro FX 4400, Cg: ~14.4 MBSOPS (4.8X)
  - Quadro FX 4400, Cg, 100 runs: ~176.0MBSOPS (59X)
    (remove test/data transfer bandwidth overhead)
- How?
  - CPU: ~11GFLOPS, slow exp(), log(), sqrt(), fast mem access
  - GPU: ~65GFLOPS, fast exp(), log(), sqrt(), slow mem access
  - Black-Scholes has high ratio of math to memory access
  - GPU has Parallelism

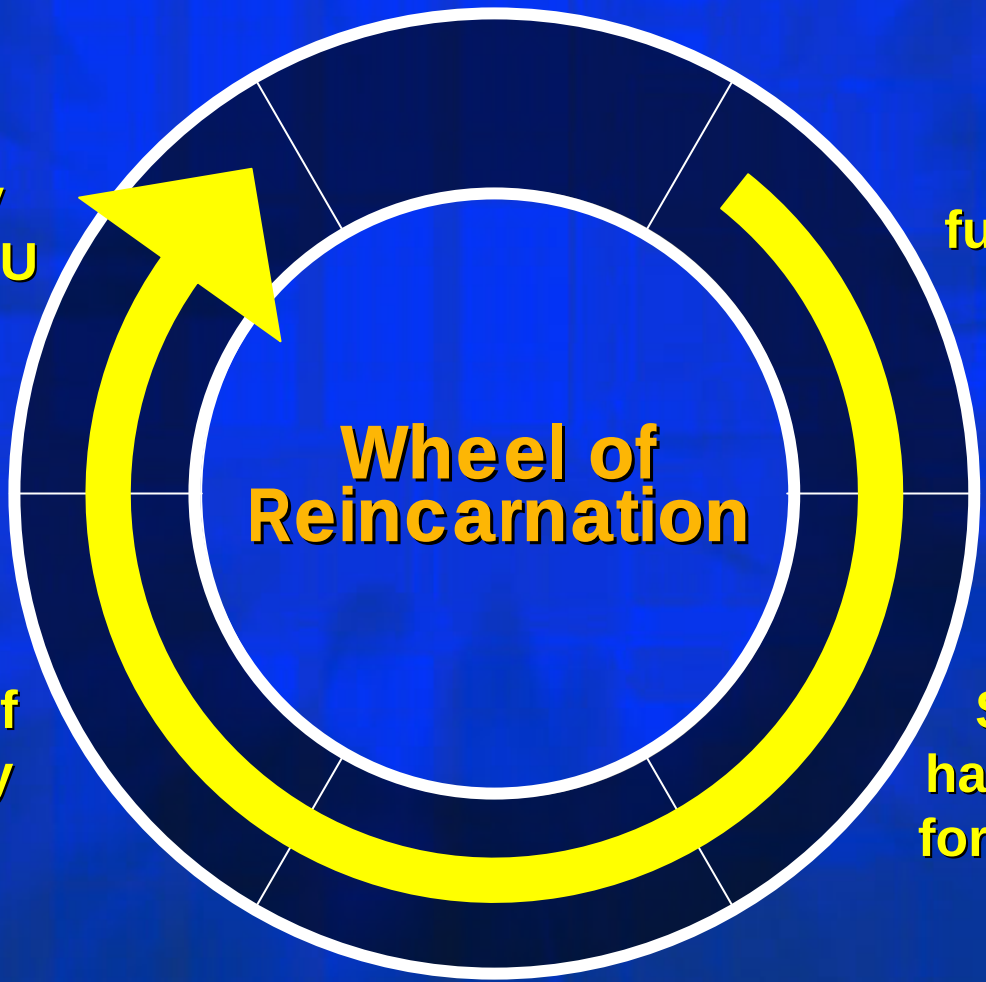
David B. Kirk, NVIDIA

IEEE Hot Chips 2005

intel

“We spent a long time on that wheel before we finally broke free”

Myer and Sutherland, 1968



Functionality folded into CPU

New functionality identified

Inefficiency of heterogeneity realized

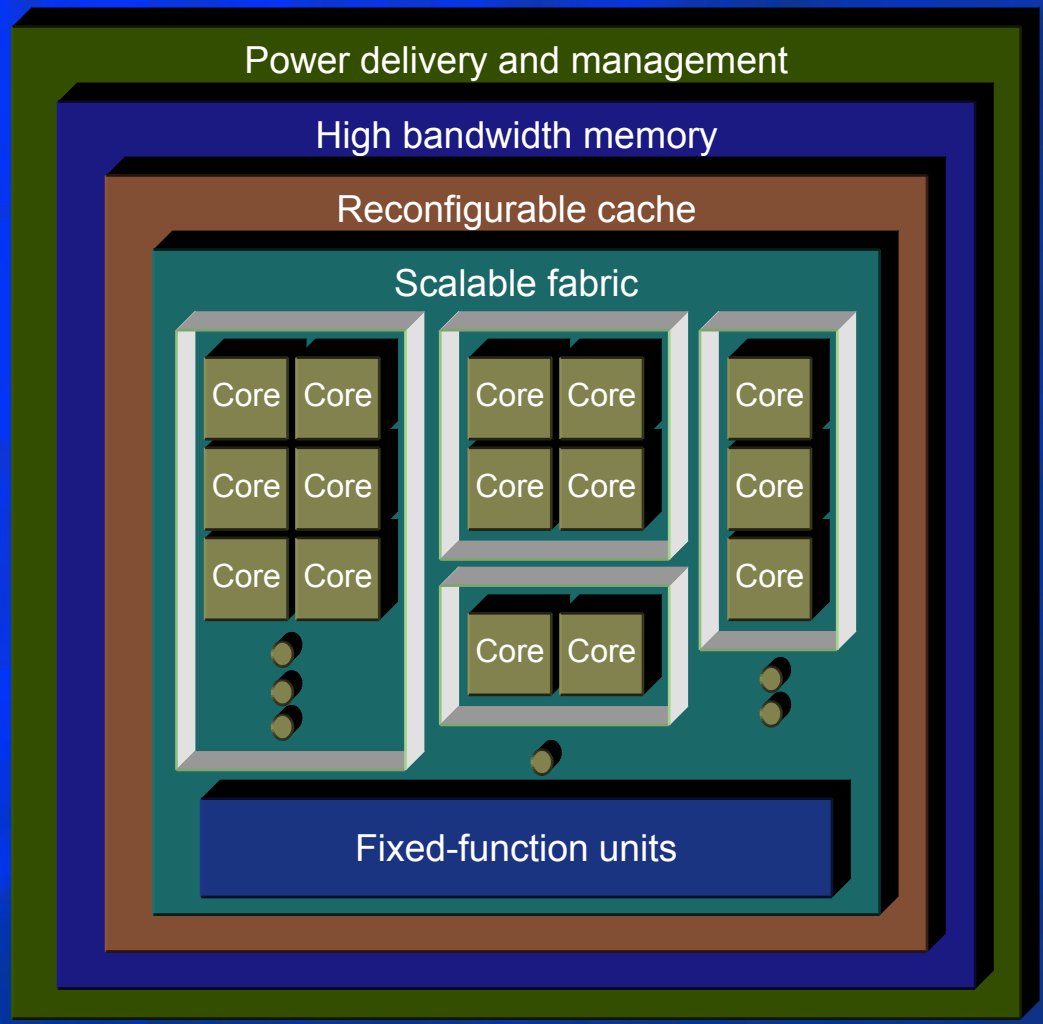
Special-purpose hardware developed for speed / efficiency

HW evolves to more general-purpose

intel

General-Purpose Multi-Core 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



Rethinking the Core

- Need efficient cores – “back to the future”
  - Reduced energy per instruction
  - More TLP, less ILP
- Potential benefit – full ISA plus:
  - 0.25 area and power
  - 0.5X legacy performance
  - 0.6-0.8X performance for MT apps
  - 1.0X FP performance

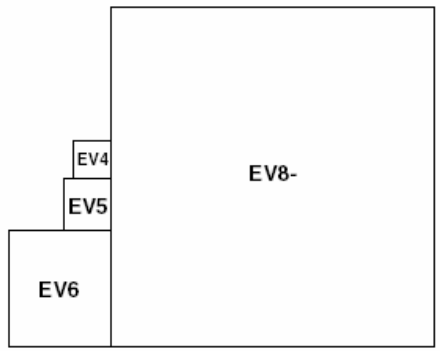
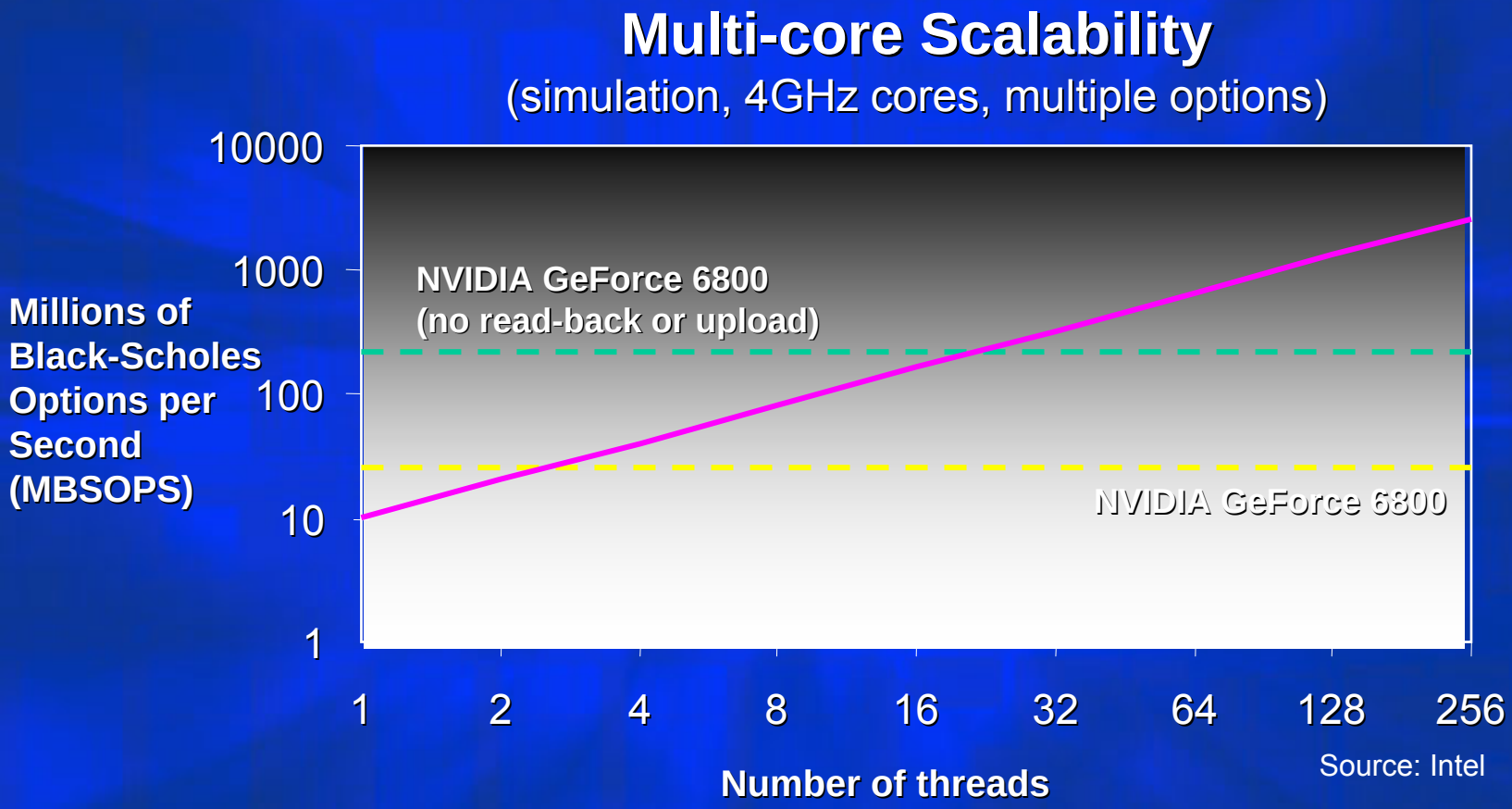


Figure 1. Relative sizes of the cores used in the study

N. Jouppi, Computer Architecture Letters, July 2003

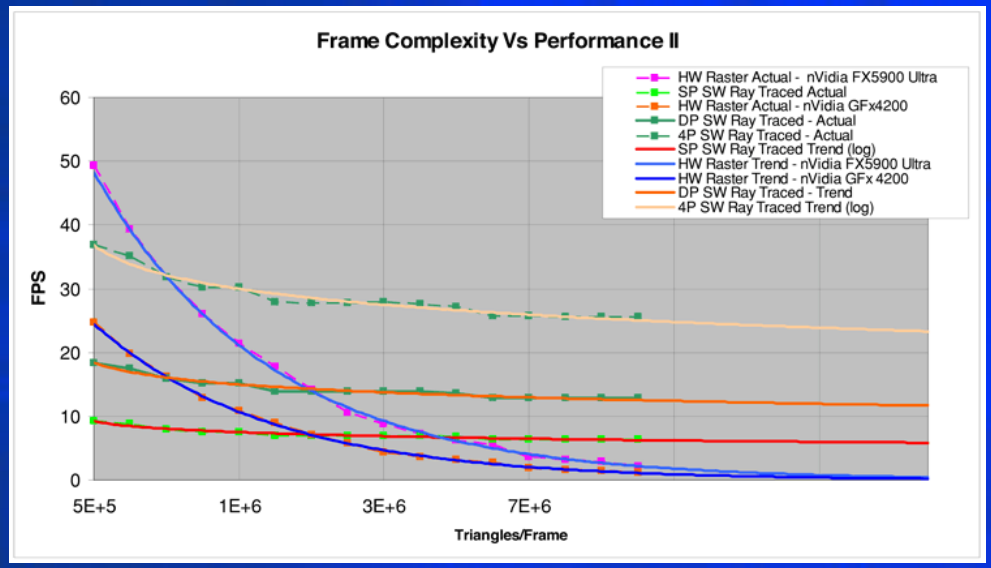
Black-Scholes Revisited



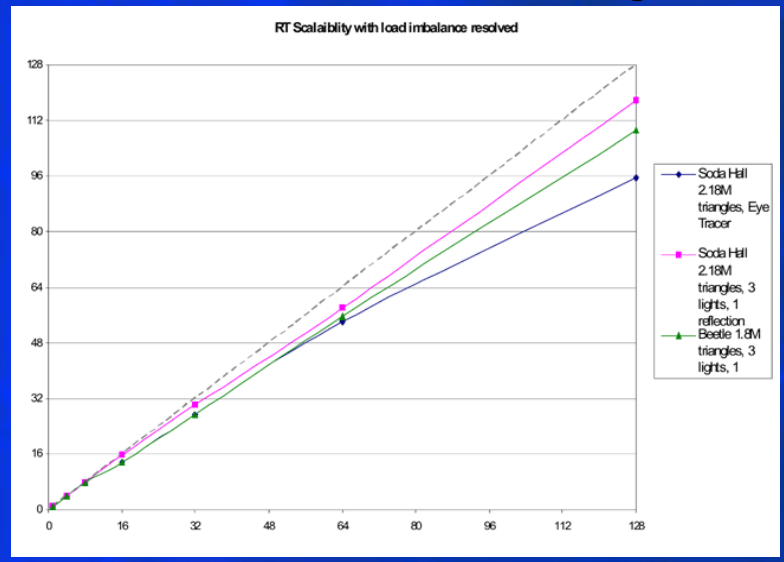
Surpasses GPU with just 1-4 cores

But Can It Do Graphics?
Real-Time Ray Tracing

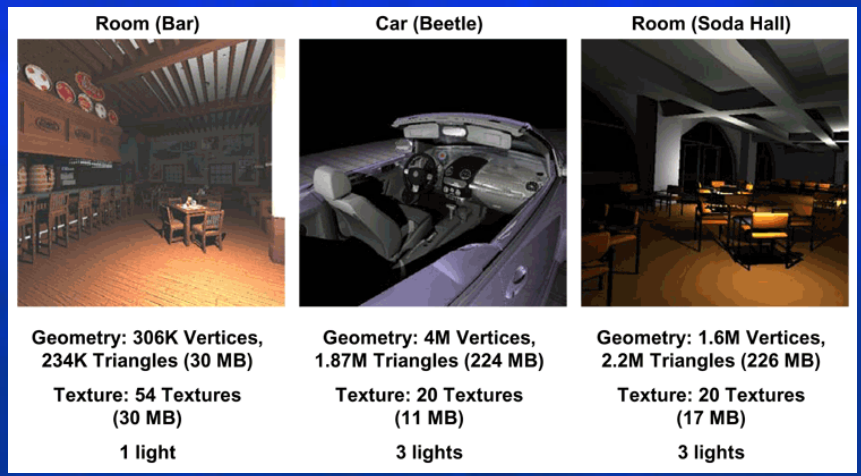
10X More Efficient Than GPUs



Great Scalability



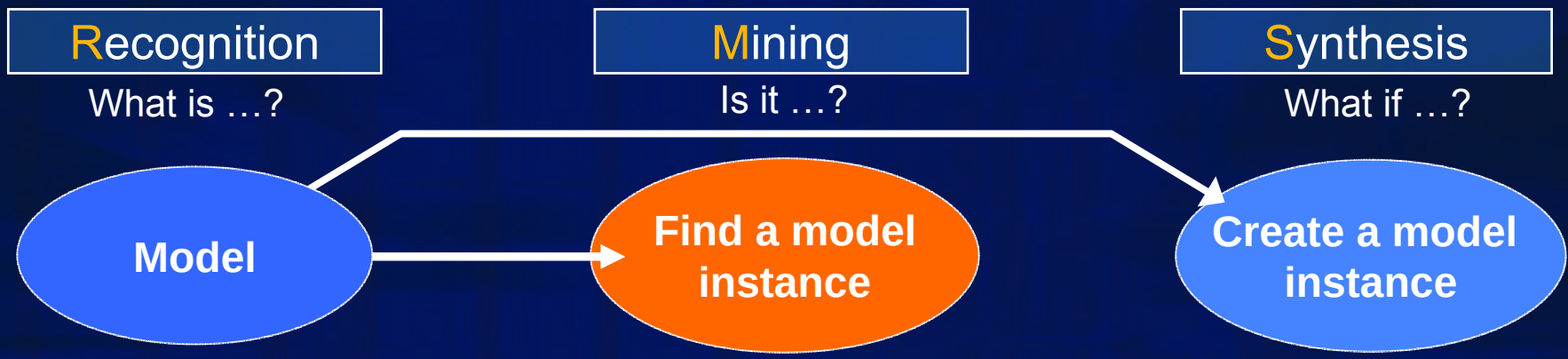
Amazing realism



Source: Intel

http://portal.acm.org/ft_gateway.cfm?id=1073329&type=pdf
ftp://download.intel.com/technology/itj/2005/volume09issue02/art01_ray_tracing/vol09_art01.pdf

Tomorrow's Killer Applications



Today

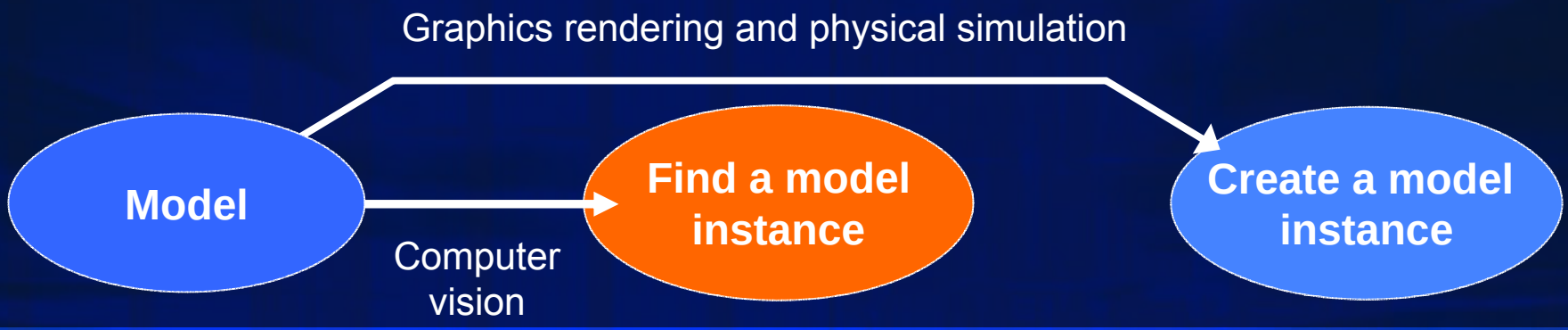
- Model-less
- Real-time streaming and transactions on static – structured datasets
- Very limited realism

Tomorrow

- Model-based multimodal recognition
- Real-time analytics on dynamic, unstructured, multimodal datasets
- Photo-realism and physics-based animation

intel

Example: Visual Computing



Human-body model

Multiple video streams

Synthetic world "Dancing lessons"

Source: Intel

Agenda

- Architectural crossroads
- Challenges and potential solutions

Parallel Programming Legacy

OpenMP

MPI

High Performance FORTAN

PThreads

Parallel C

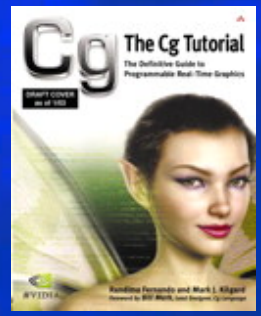
"I'm willing to stand back and say, 'Look, the way we've been doing it for half a century isn't very good.' "

Guy Steele, SUN Fellow on Fortress

intel

Recent Language Efforts

- Modern languages
  - Java, C#
  - No huge windfall
- Domain-specific languages
  - Cg, Baker
  - Limited success through restriction



Problem is deeply rooted in the model

intel

What's Do Parallel Programmers Need?

- Versatile
- Intuitive
- Efficient
- Reliable
- Composable

Two major model options:
Shared memory and message passing

intel

Shared Memory Issues

- Latency
  - If off-chip, then huge (orders of magnitude)
- Coherency
  - Expensive off-chip
- Synchronization
  - Performance vs. correctness
  - Does not compose

Need a better synchronization model

intel

Transactional Memory (TM)

- Updates multiple memory locations atomically
- Simplifies design, provides composability

Thread 1

```
begin_xaction
A = A – 20
B = B + 20
A = A – B
C = C + 20
end_xaction
```

Thread 1's accesses and updates to A, B, C are atomic

Thread 2

```
begin_xaction
C = C - 30
A = A + 30
end_xaction
```

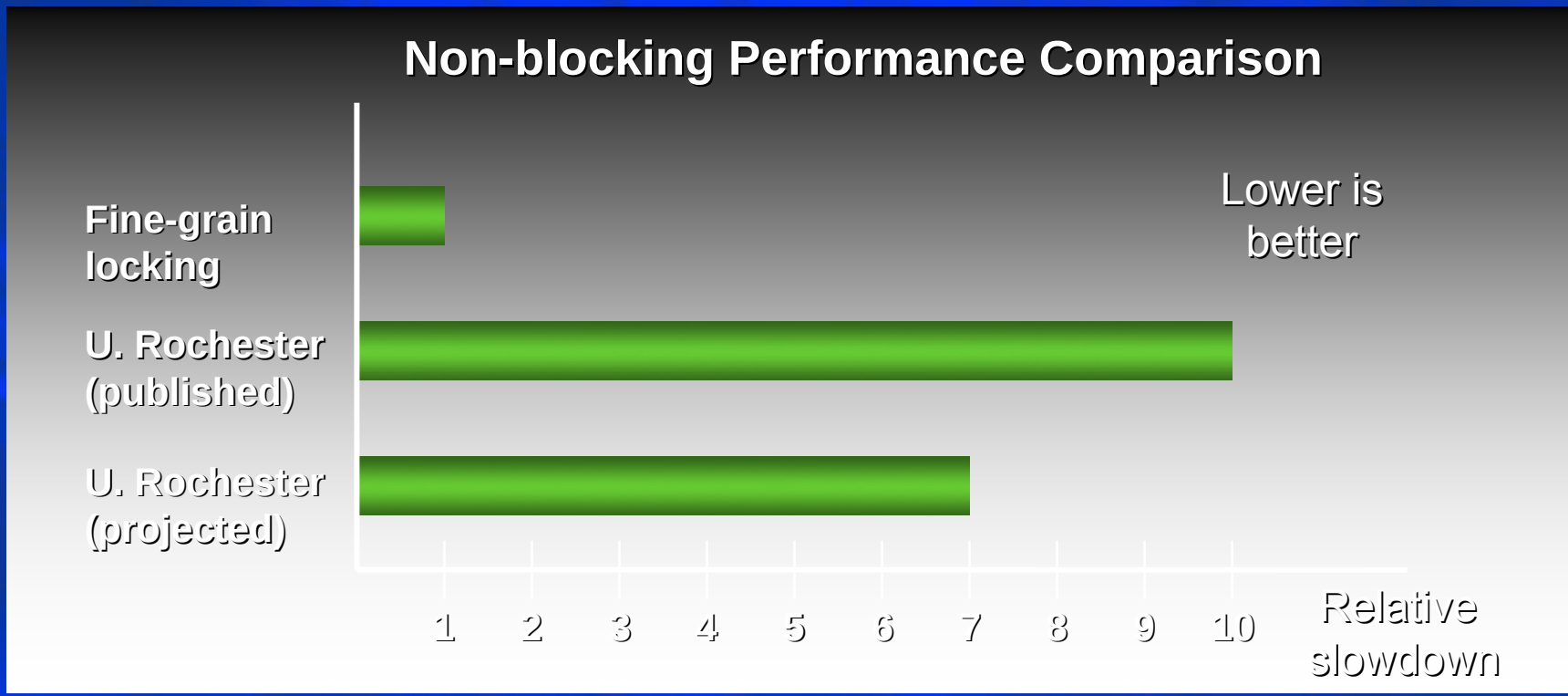
Thread 2 sees either "all" or "none" of Thread 1's updates

"Transactional memory is a ray of light in the darkness "

Satnam Singh, Microsoft

intel

SW Transactional Memory

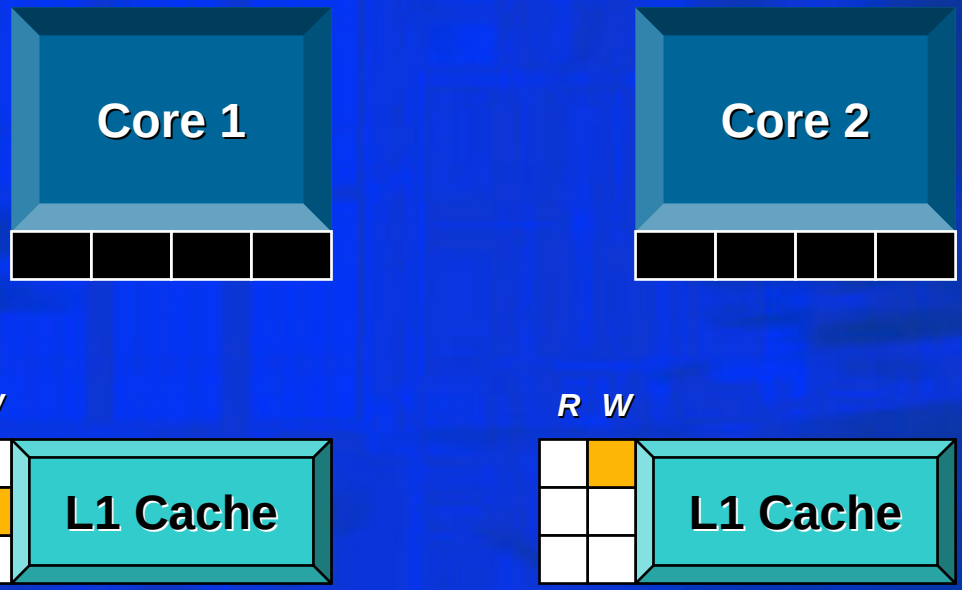


Source: Michael Scott, U. Rochester

Platform agnostic, but significant overhead even for common, non-blocking case

HW Support for TM Performance

Per-thread stack register
Zero-overhead
Checkpoint/restore



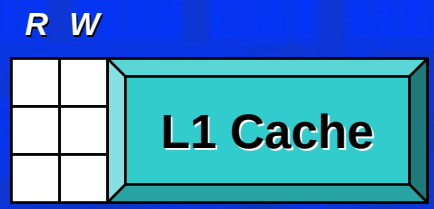
Read/write bit (per block)
Zero-overhead
Buffer/commit data

Virtual TM for Completeness

Per-thread stack register
Zero-overhead
Checkpoint/restore

Timer interrupts,
Context switches,
Exceptions,…

Virtual TM



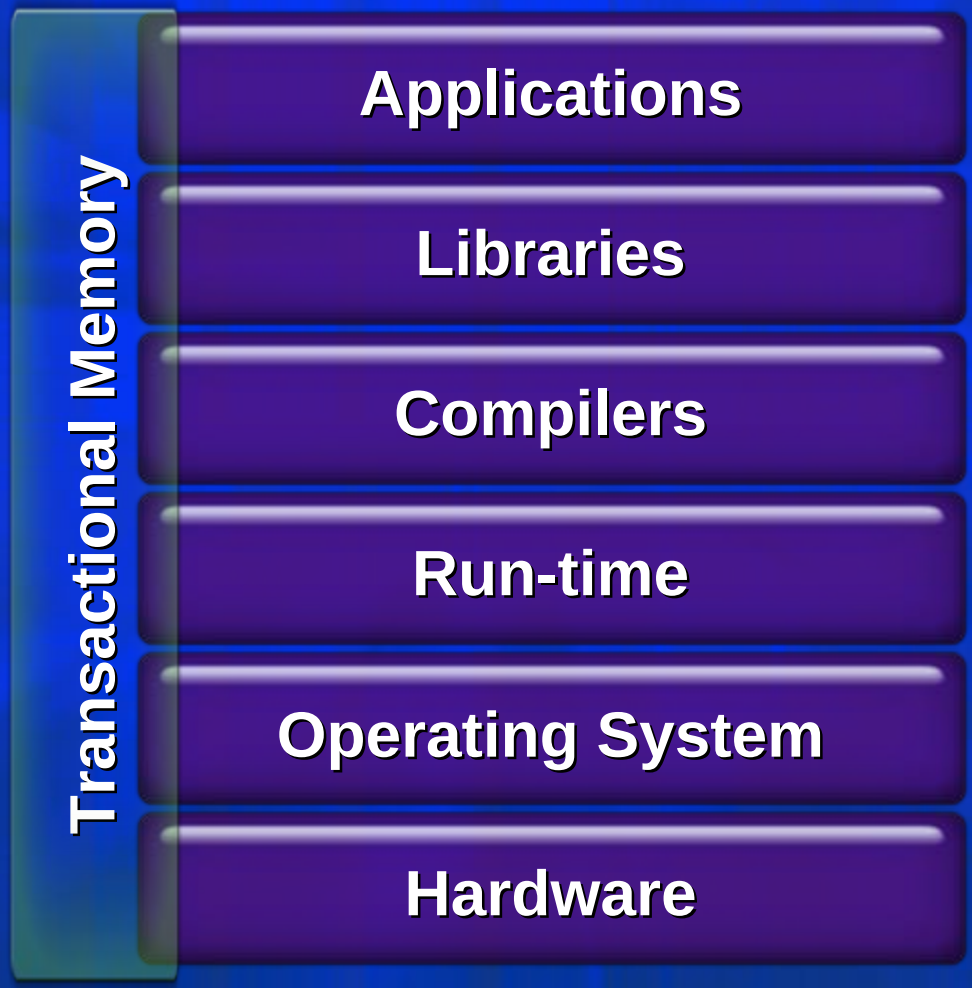
Limited buffers

Read/write bit (per block)
Zero-overhead
Buffer/commit data

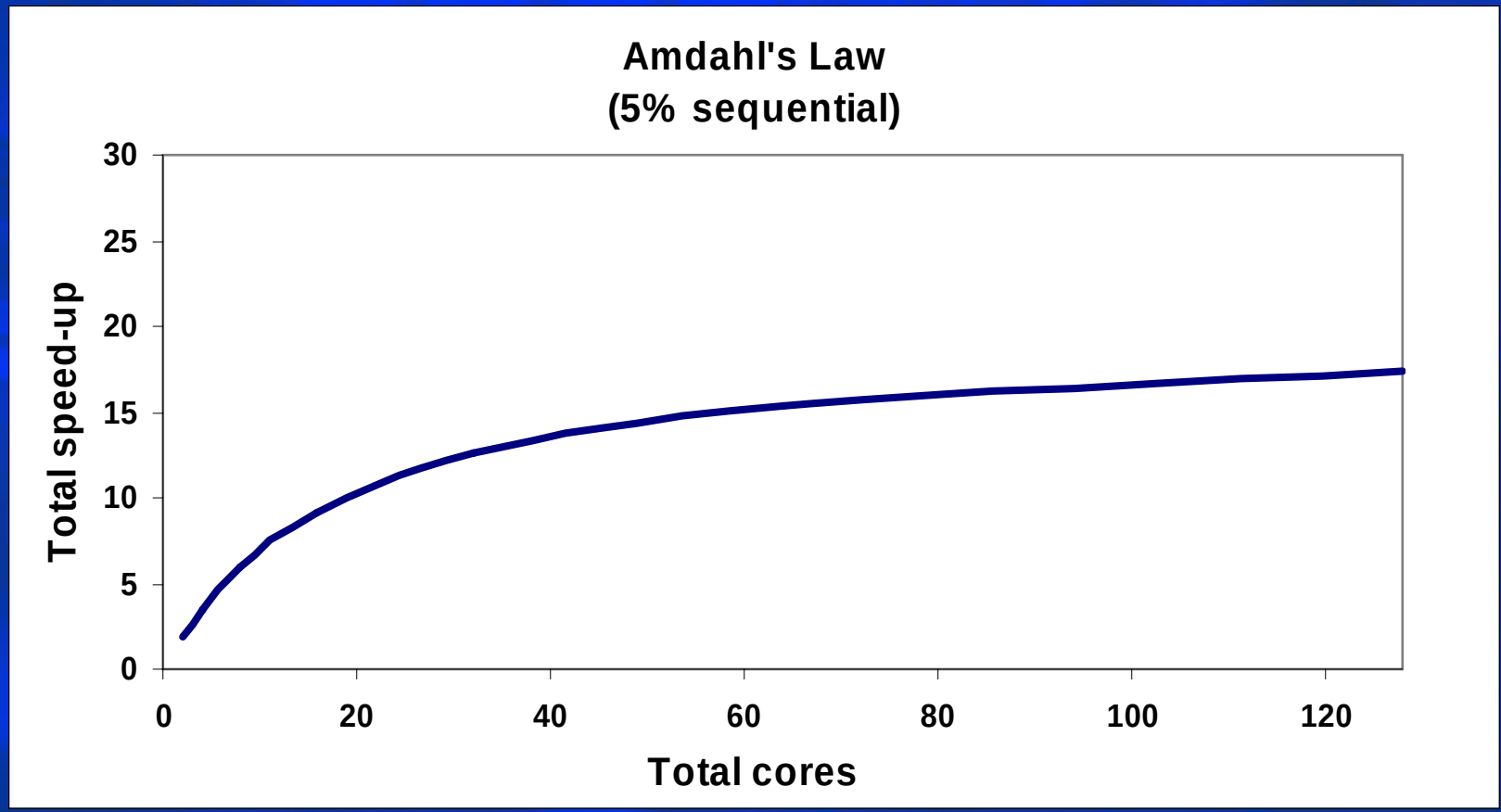
Memory System /
Coherence Network

http://www.cs.wisc.edu/~rajwar/scripts/download.cgi?file=http://www.cs.wisc.edu/~rajwar/papers/isca05-vtm.pdf

TM Requires Complete Stack



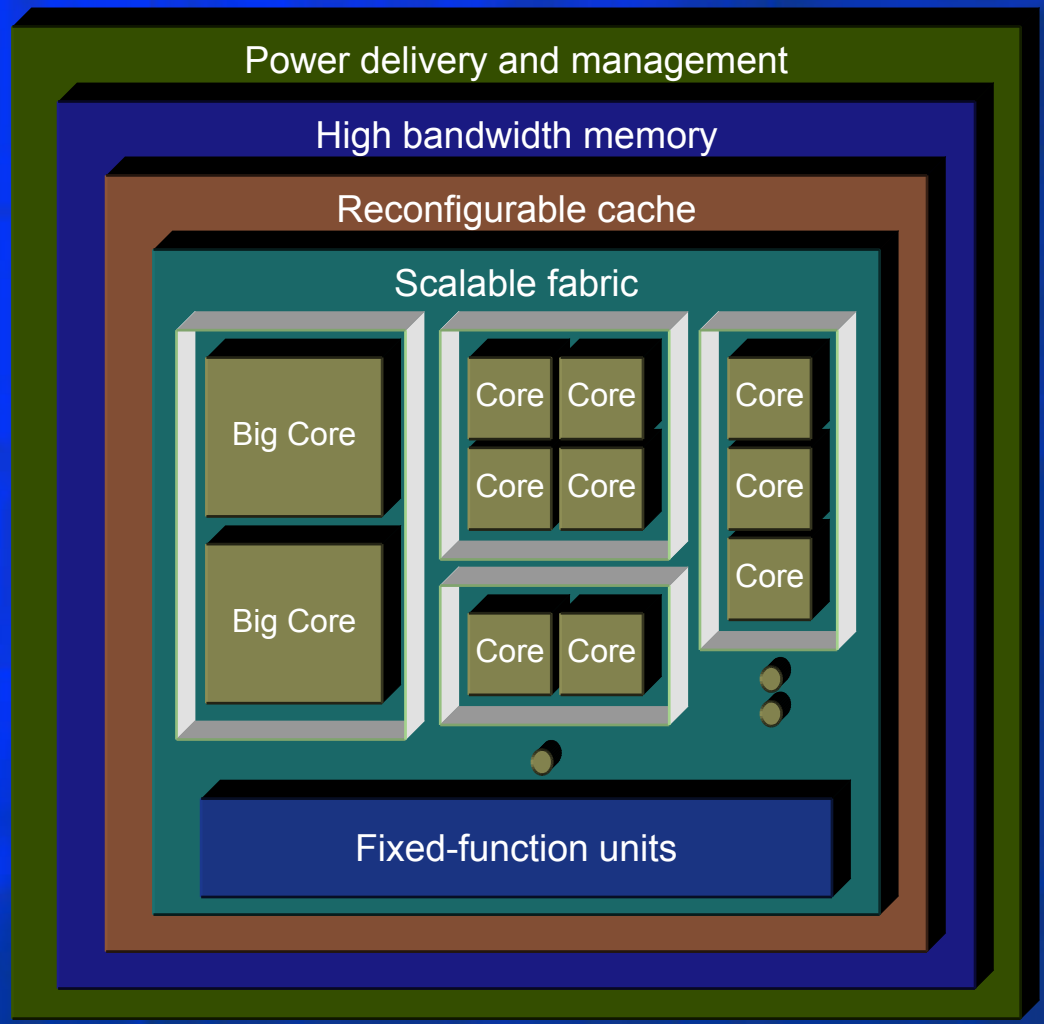
The Importance of ST Performance



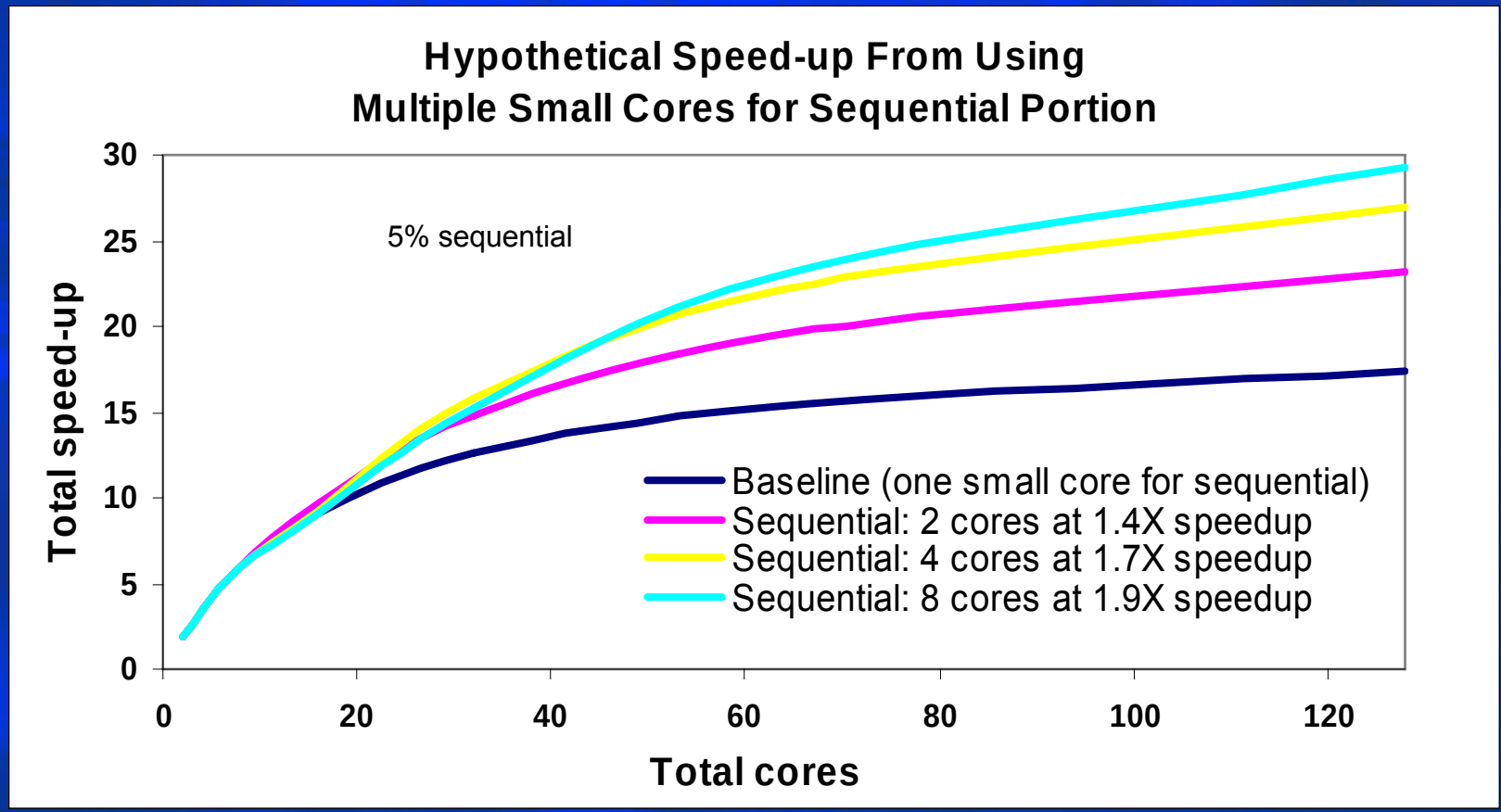
Asymmetric Solution to ST Performance

- Big cores for ST
- Small cores for MT

- Pros
  - Best for legacy
  - Best speedup
- Cons
  - Compromised MT
  - Limited flexibility

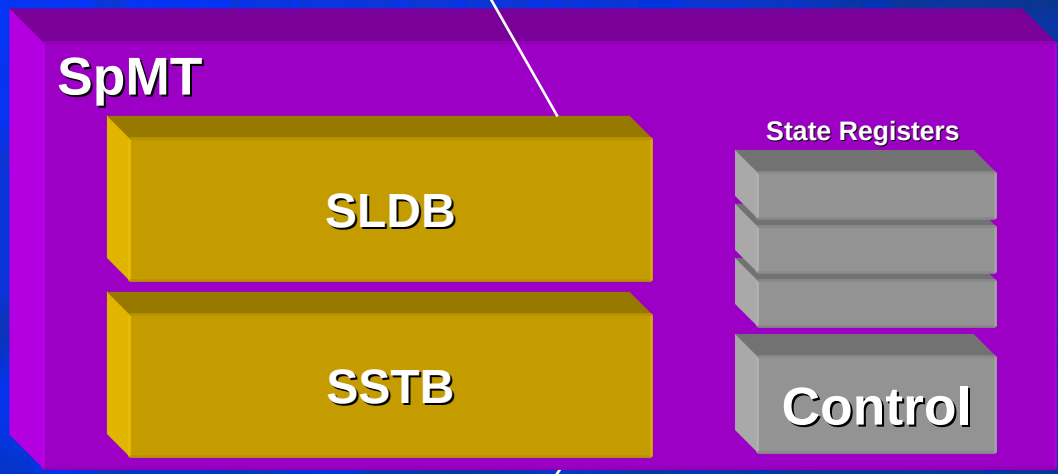


The Potential Improvement



HW Support for Speculative MT

- **Speculative Load Buffer**
  - Keeps the addresses accessed speculatively
    - Read by spec thread, not produced by it
  - Checked by non-speculative stores
    - Hit ≡ Misspeculation



- **Speculative Store Buffer**
  - Keeps the values generated by the spec thread
  - When validated, values are committed

intel

Call to Action

- Break free from the HW wheel of reincarnation
  - General-purpose is where we want to go
  - Bring in the desired functionality where needed
- Focus on the fundamental programming challenges
  - Don’t create any new ones
  - One size won’t fit all
- Remember the importance of ST performance
  - Avoid the complexity of asymmetric solutions
  - Make clever use of small cores

Build balanced HW-SW solutions