

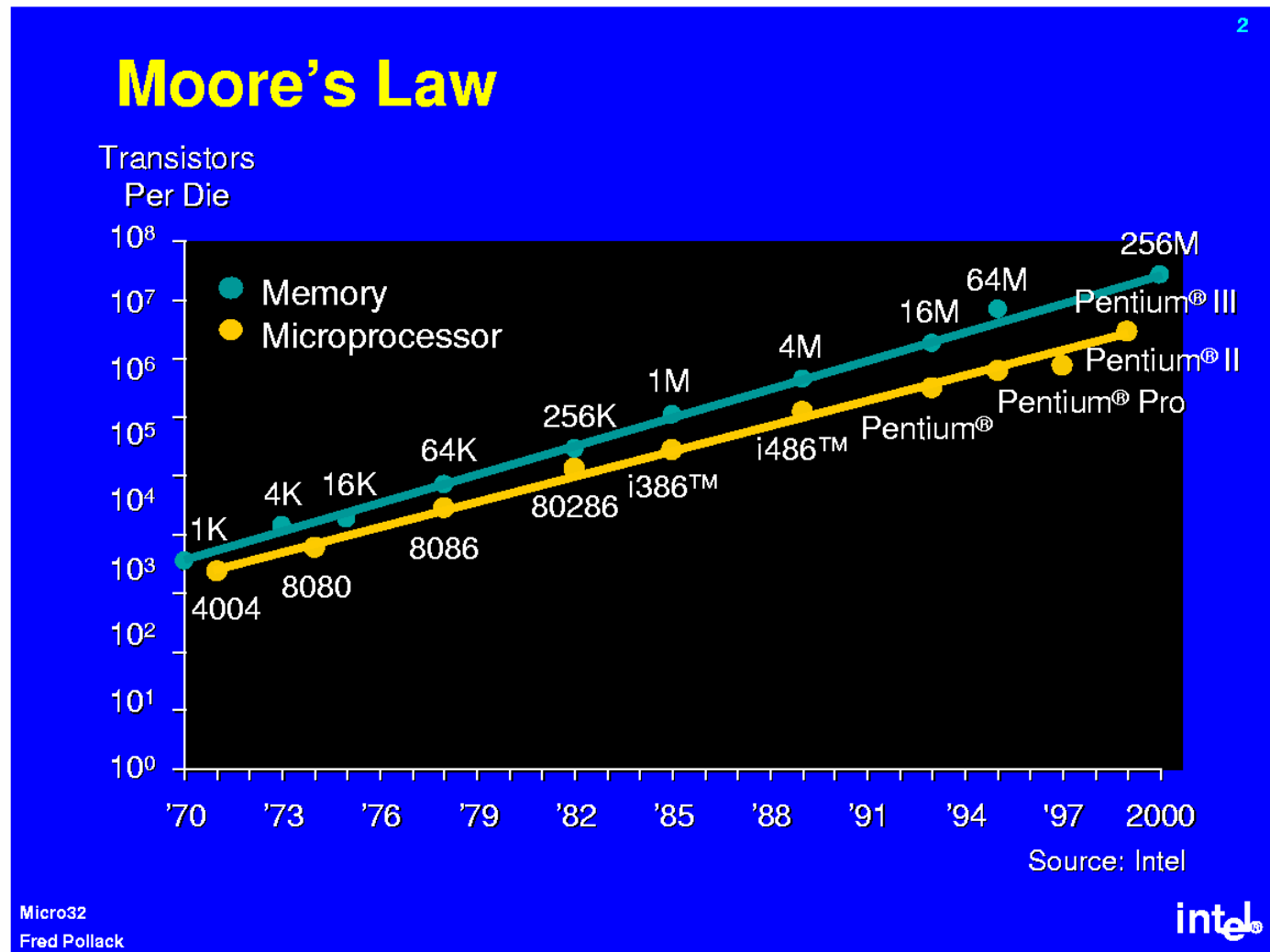
Parallel and Distributed Processing: Future Challenges

Prof. Margaret Martonosi
Dept. of Electrical Engineering
Princeton University

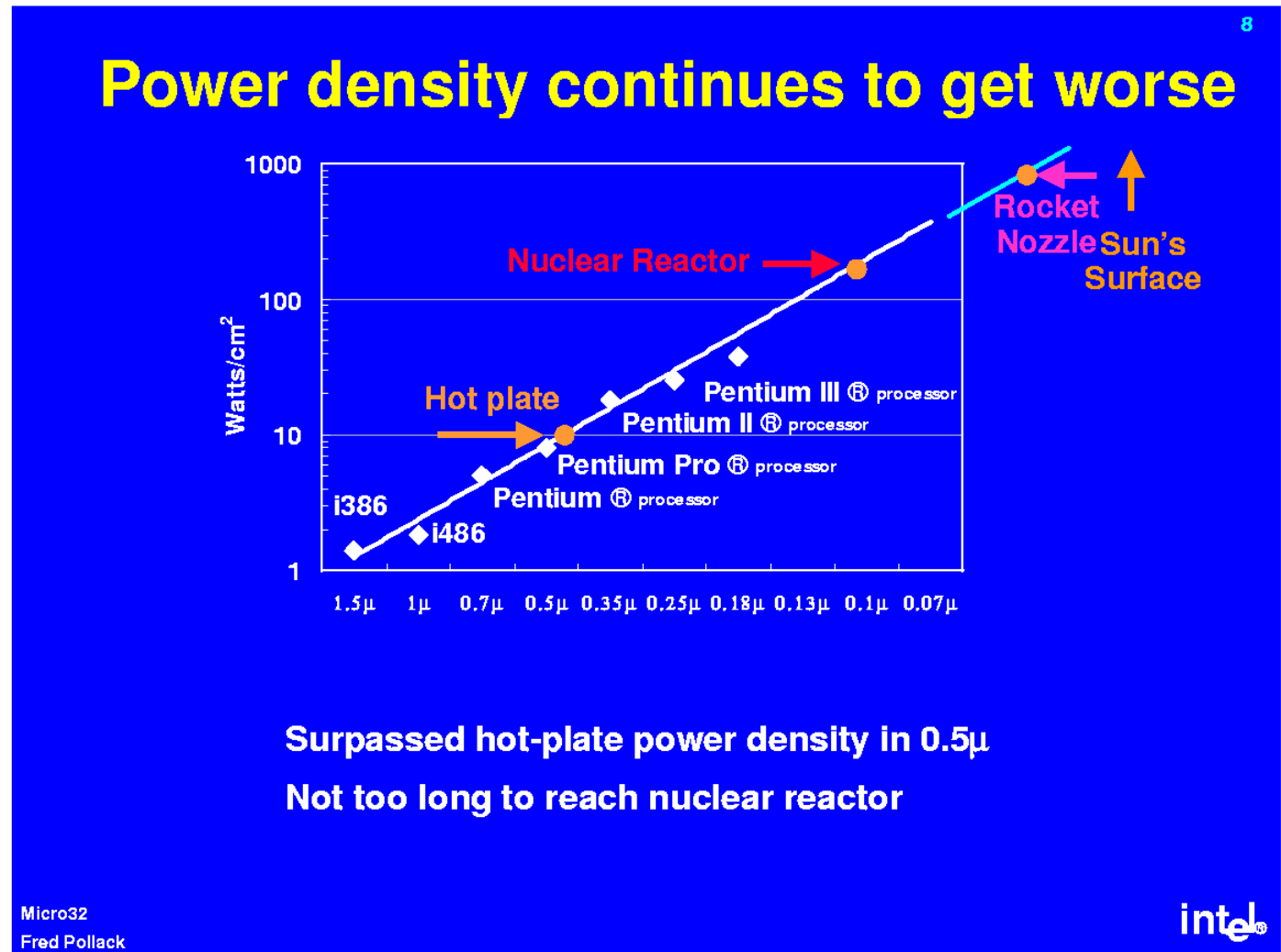
Or in other words... The Bait!

1. We do not need to develop new parallel programming paradigms since the old shared-memory and message-passing paradigms will be good enough.
2. Multicore systems are irrelevant to application software developers because they will be used only for increasing overall system throughput, not for accelerating individual application programs.
3. Cooling and power consumption will not be a major concern since most of the cores will be idle most of the time.
4. The system designer does not need to worry about reliability since the chip and circuit designers will guarantee that the cores are reliable. While an adaptive system sounds like a great idea for adjusting to changes in the system environment, real systems will have no need for adaptation since the environment for any given application will remain relatively constant.
5. The new multicore computing systems will not need any performance tools since each core will be fast enough for typical applications (which will be single-threaded).

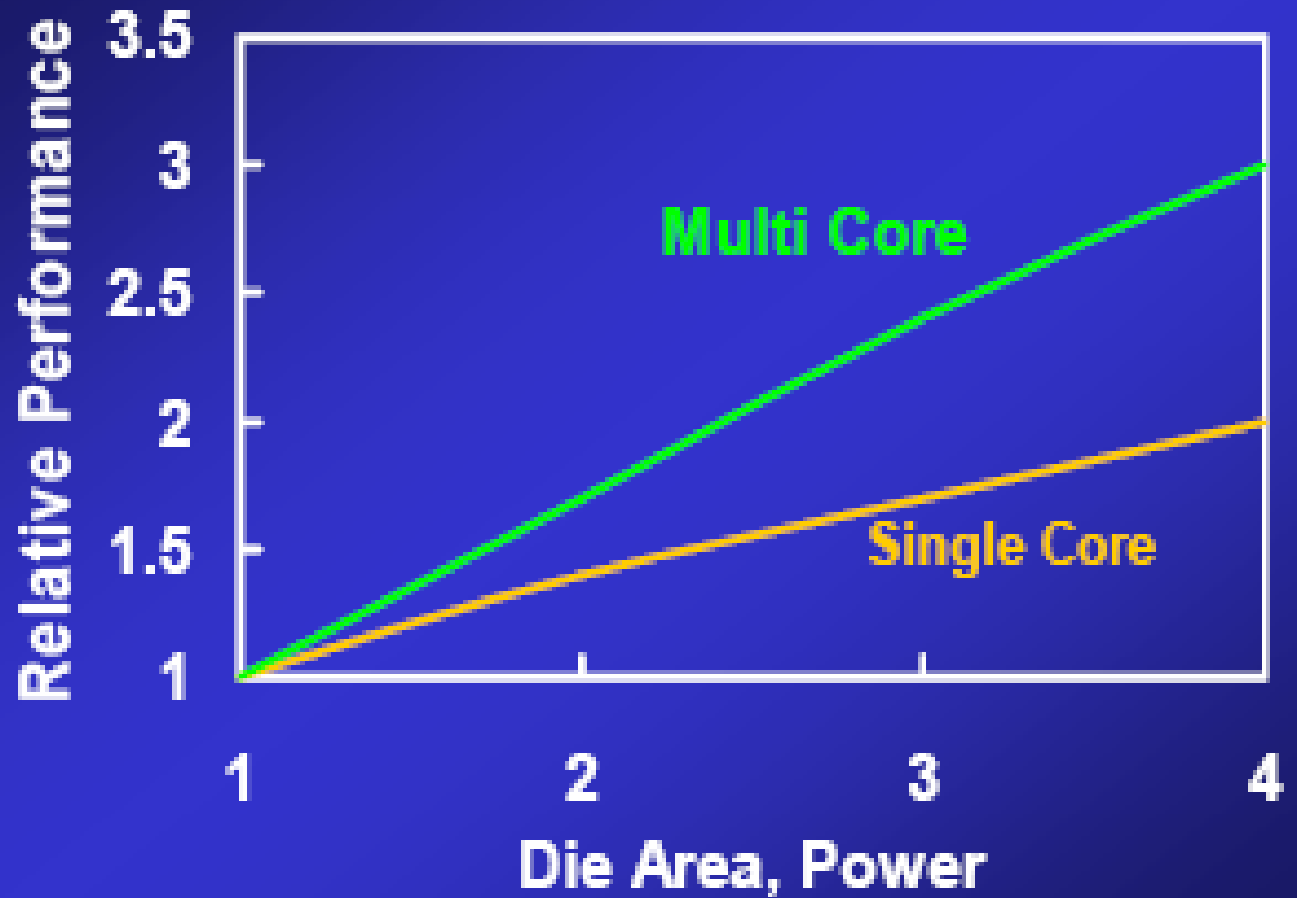
Moore's Law gives us transistors...



But... Power density is exponential too...



CMPs offer superior power-performance...



Software's Future

- Implications of Moore's Law are changing
 - Old Moore's Law:
 - Doubling transistors => Double performance
 - New Moore's Law:
 - Double transistors => Double # cores
- Performance: If you want it, you can have it, but so far it ain't easy...

The Realities

- If you build it, they will come.
 - Hardware is multicore
 - Therefore software will be parallel
- That which does not kill us makes us stronger.
 - Parallel software is difficult
- Curiouser and curiouser...
 - Emerging technologies will make everything worse, not easier...

Reduced to a previously-unsolved problem

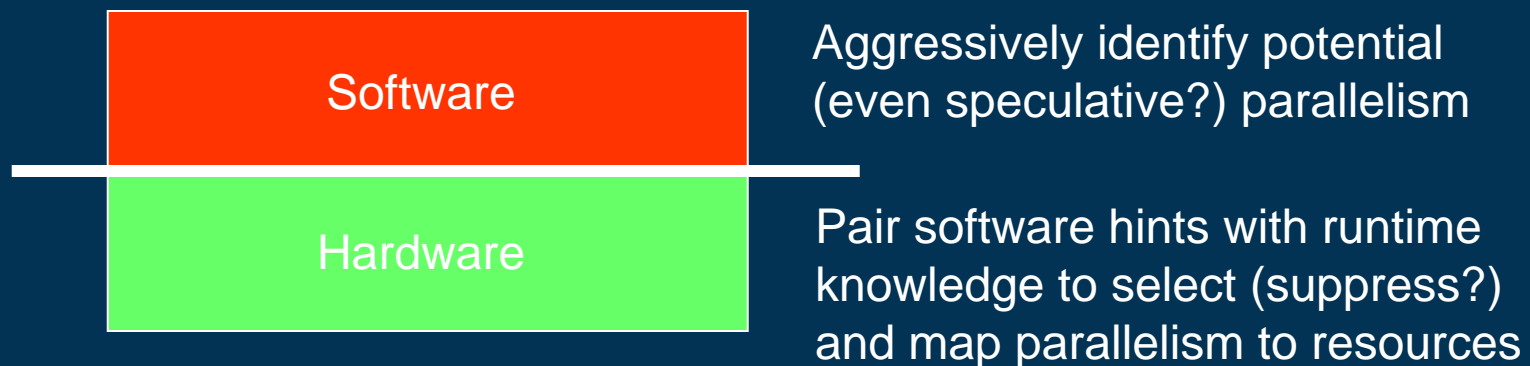
- Multicore is for real
 - IBM Power5, Power6, Intel Itanium, Sun Niagara, ARM MPCore
- Multicore makes a lot of (hardware) sense
 - Moore's Law gives us enough transistors
 - Replicating cores mitigates design complexity
 - Running out of easy tricks for single-core performance
- But, parallelism is hard...

Key challenges: Map workloads to CMPs to maximize performance... and also handle power, thermal, reliability limits?

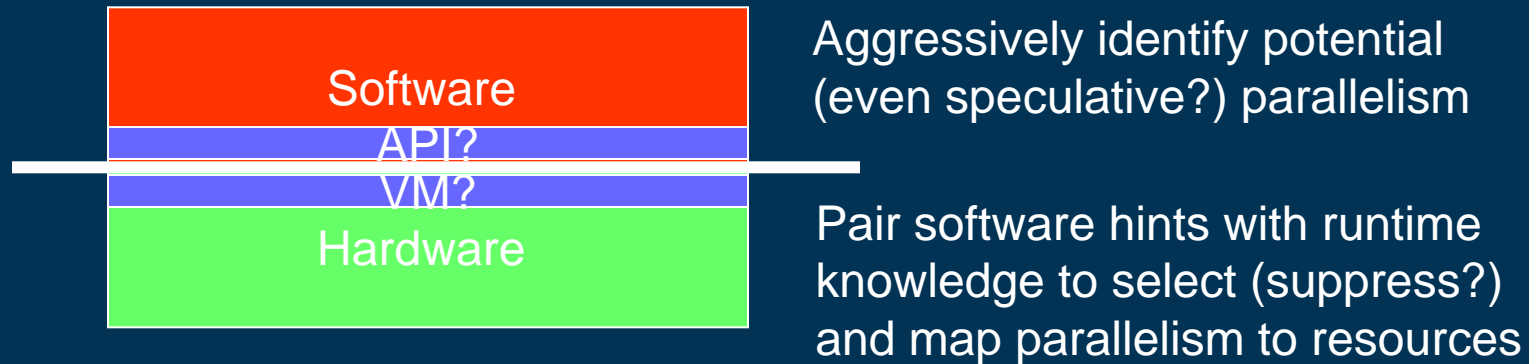
Problems with parallelism...

Static (programmer/compiler) Dynamic (OS, runtime sys)

- Oblivious to other apps (multiprogramming)
 - Hard to handle run-time variable latencies (memory, I/O)
 - Harder to handle broken or overheated cores...
- High thread overhead forces coarse-grained partitioning
 - Not responsive enough to underlying hardware: power, thermal hotspots, network congestion...

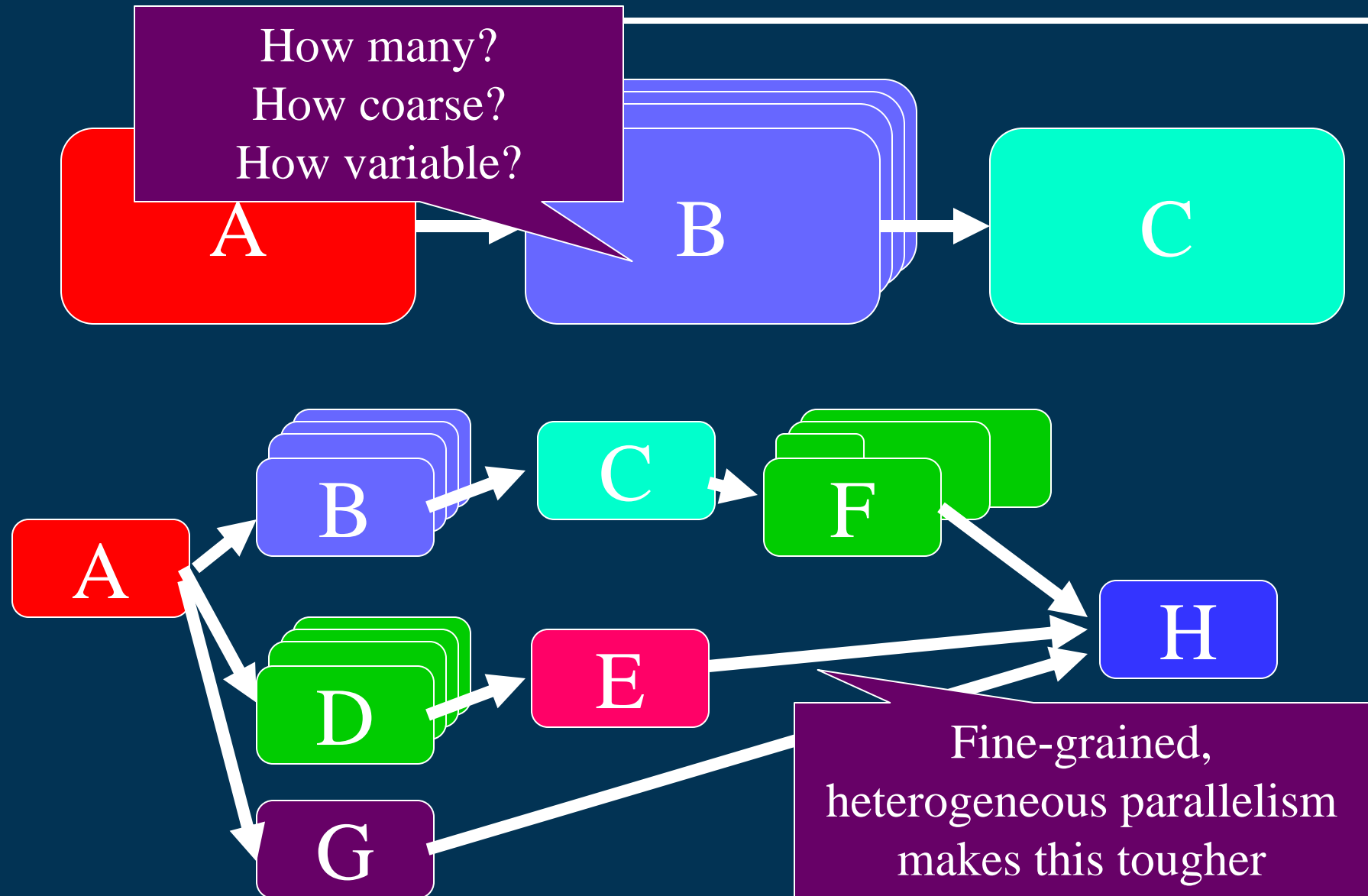


New Hardware/Software Contract

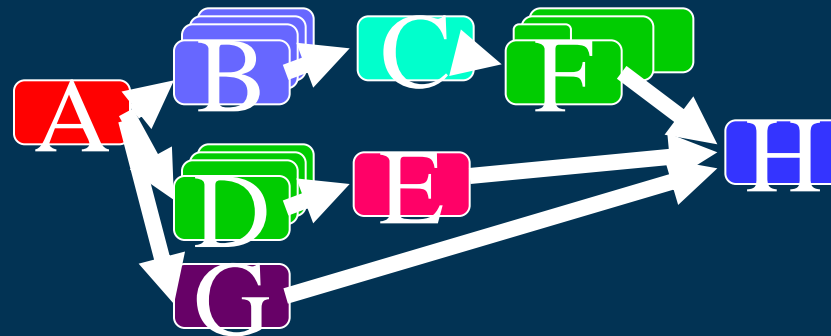
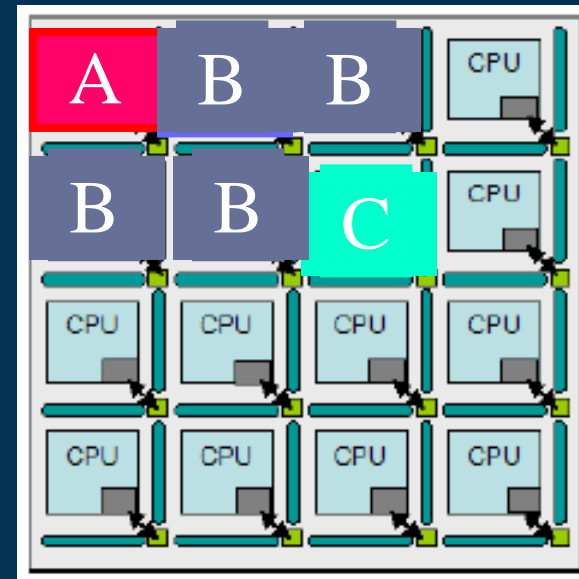
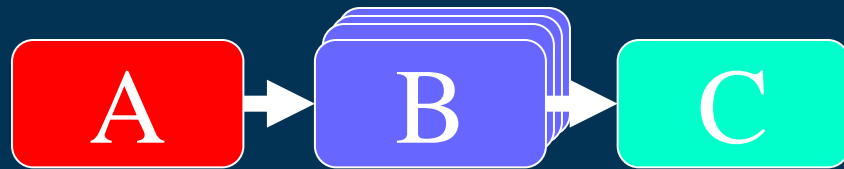


- Give “hardware” more control over aspects it knows best: memory latency, dynamically-varying parallelism
 - But also new tough issues like thermal, reliability...
- Empower software just to find potential parallelism
 - “Worry” less about load balancing, granularity, ...
- Clean execution model: smoothly add/remove parallelism to respond to energy/thermal concerns

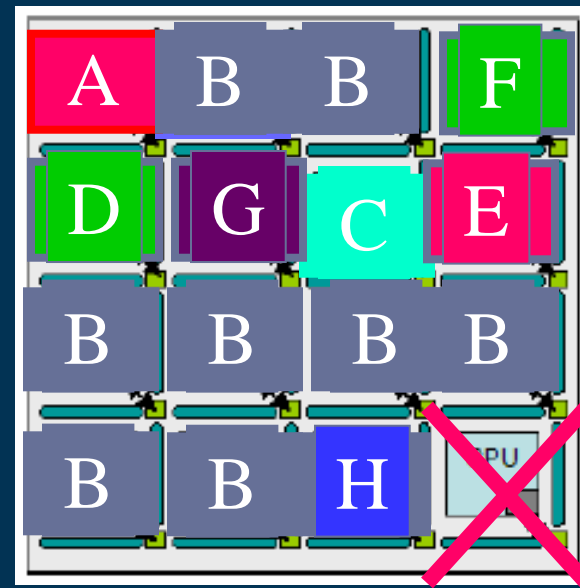
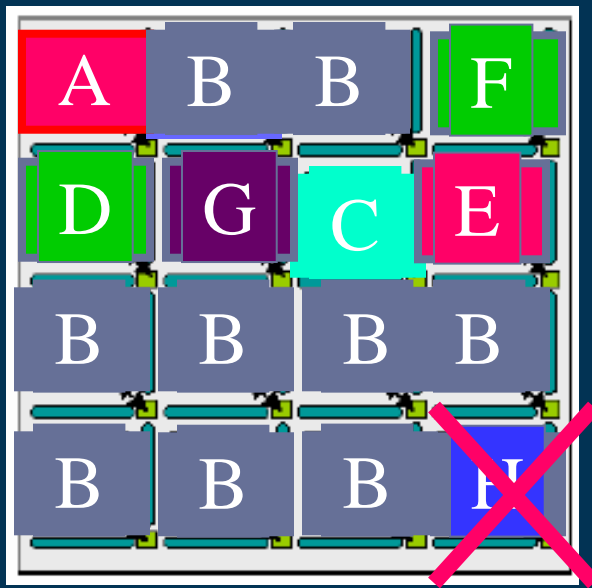
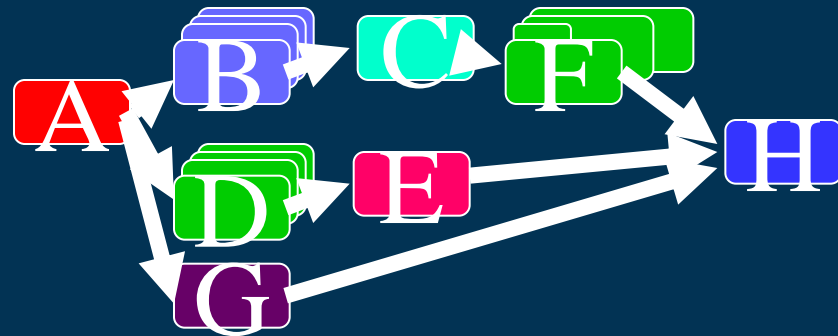
Managing and Mapping Parallelism



Dynamic Mapping with Hardware Support



Dynamic Mapping with Hardware Support



Adaptive Power/Performance Management

- Tradeoff parallelism/speed for performance
- Single clean model to handle:
 - Core on/off due to reliability issues
 - Core speedup/slowdown/on/off due to thermal throttling
 - Core speedup/slowdown/on/off due to dynamic energy management.
- Thus far... Adaptive control methods for
 - DVFS management in single-core systems (ASPLOS 2004, HPCA 2005)
 - DVFS management in multi-core systems (ISLPED 2005)
 - Dynamic compiler-driven energy management (Micro 2005)

The Bait

1. We do not need new programming paradigms or hardware paradigms. **New hw/sw contract needed. Build programming models on top of this** passing
2. Multicore systems are irrelevant to application software development because they will be used only for increasing overall system throughput, not for accelerating individual application programs. **No!**
3. Cooling and power are a major concern since most of the cores are mostly pointless the time.
4. The system designer does not need to worry about reliability since the chips are reliable. **No! Environment is time and workload dependent, so adaptation is inevitable** the cores idea systems
5. The new multicore computing systems will not need any performance tools since each core will be fast enough for typical applications (which will be single-threaded).