Fall 2019
EE 6633: Architecture of Parallel Computers
Lecture 1: Introduction to Parallel Processing

Avinash Karanth
Department of Electrical Engineering & Computer Science
Ohio University, Athens, Ohio 45701
E-mail: karanth@ohio.edu
Website: http://ace.cs.ohio.edu/~avinashk/classes/ee663/ee663.htm

1

Today's Goal:

- Go over the syllabus for EE 6633 with you
- Introduce you to Parallel Computer Architecture
- Answer your questions related EE 6633
- Start providing you a sense of the trends that shape the field

2

What will you get out of EE 6633?

- In-depth understanding of the design and engineering of modern parallel computers
  - technology forces
  - fundamental architectural issues
  - fundamental parallel computing models
  - fundamental issues in designing parallel programs
  - methods of evaluation
  - underlying engineering trade-offs
- from moderate to very large scale
- across the hardware/software boundary

3

Will it be worthwhile?

- Absolutely!
  - even though few of you will become PP designers
- The fundamental issues and solutions translate across a wide spectrum of systems and architectures.

4

What is Parallel Architecture?

- A *parallel computer* is a collection of processing elements that cooperate and communicate to solve large problems fast
- Some broad issues:
  - Resource Allocation:
    - how large a collection?
    - how powerful are the elements?
    - how much memory?
  - Data access, Communication and Synchronization
    - how do the elements cooperate and communicate?
    - how are data transmitted between processors?
    - what are the abstractions and primitives for cooperation?
  - Performance and Scalability
    - how does it all translate into performance?
    - how does it scale?

5

Why Study Parallel Architecture?

Role of a computer architect:

To design and engineer the various levels of a computer system to maximize *performance* and *programmability* within limits of *technology* and *cost*.

Parallelism:

- Provides alternative to faster clock for performance
- Applies at all levels of system design
- Is a fascinating perspective from which to view architecture
- Is increasingly central in information processing

6

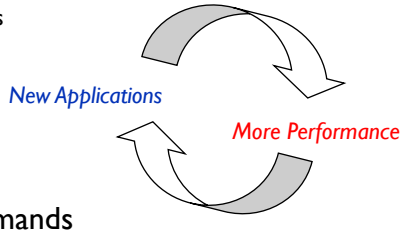
Is Parallel Computing Inevitable?

- Application demands: Our insatiable need for computing cycles
- Technology Trends
- Architecture Trends
- Economics
- Current trends:
  - Application-specific or domain-specific computing
  - Servers and workstations becoming are massively parallel (10+ processors)
  - Mobile computing is also parallel

7

Application Trends

- Application demand for performance fuels advances in hardware, which enables new appl'ns, which...
  - Cycle drives exponential increase in microprocessor performance
  - Drives parallel architecture harder
    - most demanding applications



- Range of performance demands
  - Need range of system performance with progressively increasing cost

8

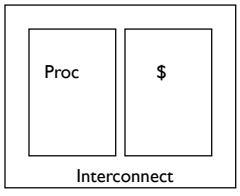
Speedup

- Speedup (p processors) = $\frac{\textit{Performance (p processors)}}{\textit{Performance (1 processor)}}$
- For a fixed problem size (input data set), performance = 1/time
- Speedup fixed problem (p processors) = $\frac{\textit{Time (1 processor)}}{\textit{Time (p processors)}}$

9

Technology: A Closer Look

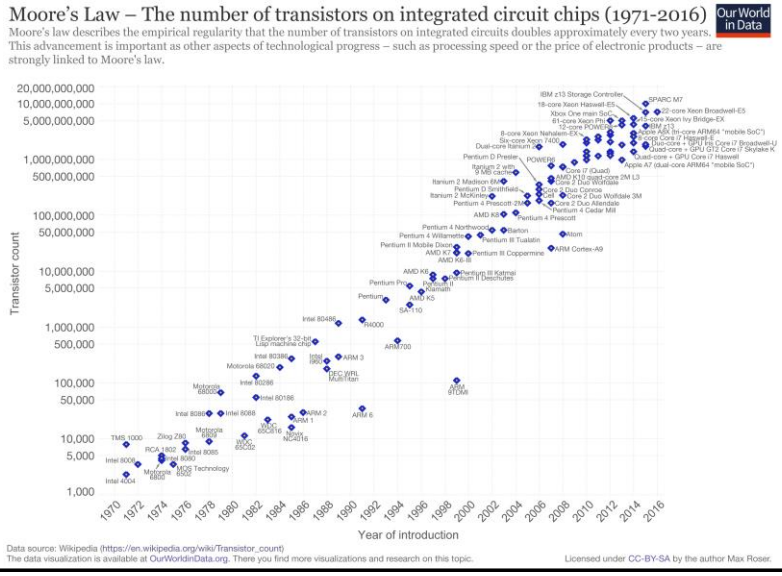
- Basic advance is *decreasing feature size* (λ)
  - Circuits become either faster or lower in power
- Die size is growing too
  - Clock rate improves roughly proportional to improvement in λ
  - Number of transistors improves like λ^2 (or faster)
- *How to use more transistors?*
  - Parallelism in processing
    - multiple operations per cycle reduces CPI
  - Locality in data access
    - avoids latency and reduces CPI
    - also improves processor utilization
  - Both need resources, so tradeoff



- *Fundamental issue is resource distribution, as in uniprocessors*

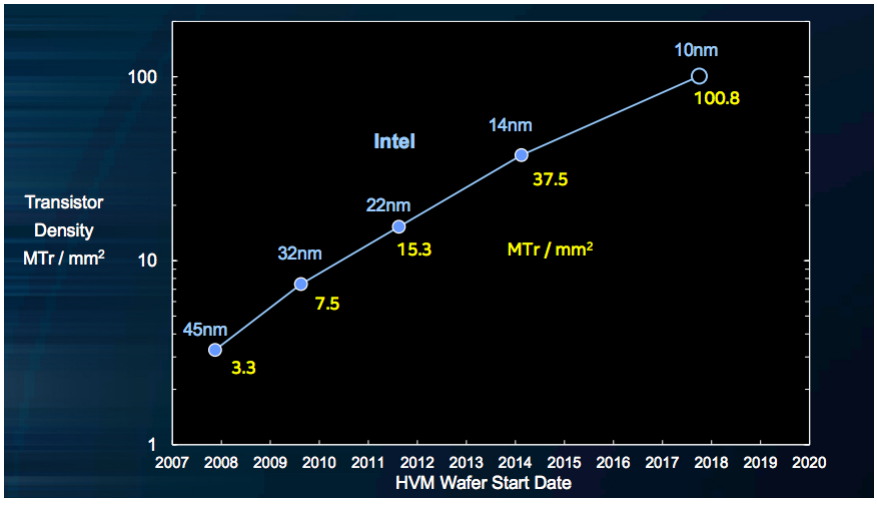
10

Moore's Law



11

Density per mm²



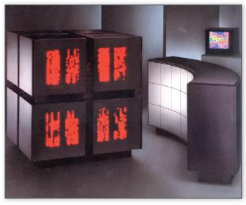
Courtesy: IEEE Spectrum, March 2017

12

Why parallel computing? Origins

- The answer 10-15 years ago: to realize improvements that exceeded what CPU performance could provide

For supercomputing applications



Thinking Machines (CM2)
(1987)
65,536 1-bit processors +
2,048 32 bit FP processors

SGI Origin 2000 — 128 CPUs
(1996)
Photo shows ASIC Blue Mountain
supercomputer at Los Alamos
(48 Origin 2000's)

For database applications

Sun Enterprise 10000
(circa 1997)
64 UltraSPARC-II processors

13

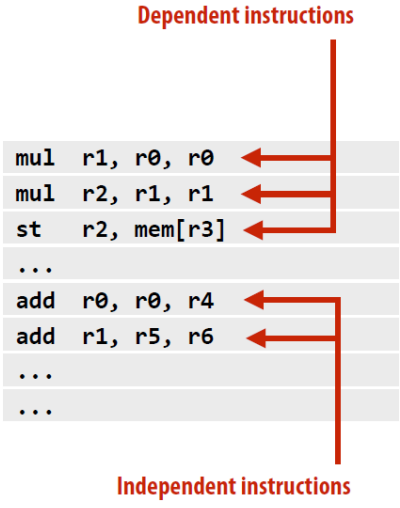
Where did the speedup originate?

- Exploiting instruction-level parallelism (uniprocessor)
- Data and thread-level parallelism (multiprocessor)
- Data locality
- Increasing CPU clock rates

14

Instruction Level Parallelism

- Processors leverages parallel execution, except that it is hidden from the programmer
- ILP
  - Instructions must appear to be executed in program order. But, Independent instructions can be executed simultaneously by a processor without impacting program correctness



15

ILP Example (1/2)

```
a = x*x + y*y + z*z
```

Consider the following program:

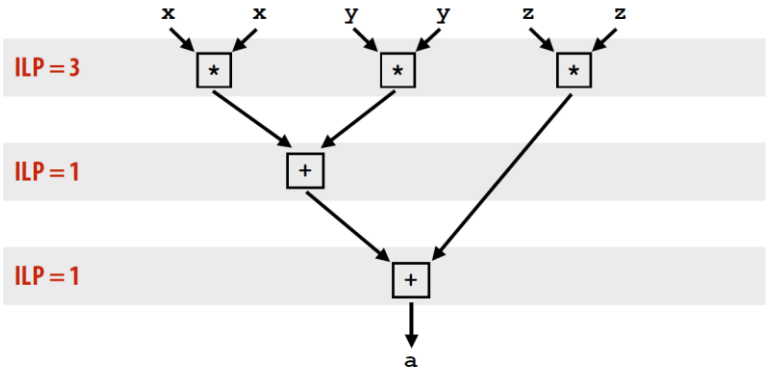
```
// assume r0=x, r1=y, r2=z

mul r0, r0, r0
mul r1, r1, r1
mul r2, r2, r2
add r0, r0, r1
add r3, r0, r2

// now r3 stores value of program variable 'a'
```

This program has five instructions, so it will take five clocks to execute, correct? Can we do better?

16

ILP Example (2/2)

```
a = x*x + y*y + z*z
```



17

Superscalar Execution

```
a = x*x + y*y + z*z
```

```
// assume r0=x, r1=y, r2=z

1. mul r0, r0, r0
2. mul r1, r1, r1
3. mul r2, r2, r2
4. add r0, r0, r1
5. add r3, r0, r2

// r3 stores value of variable 'a'
```

Superscalar execution: processor automatically finds independent instructions in an instruction sequence and executes them in parallel on multiple execution units!

In this example: instructions 1, 2, and 3 can be executed in parallel
(on a superscalar processor that determines that the lack of dependencies exists)

But instruction 4 must come after instructions 1 and 2

And instruction 5 must come after instruction 4

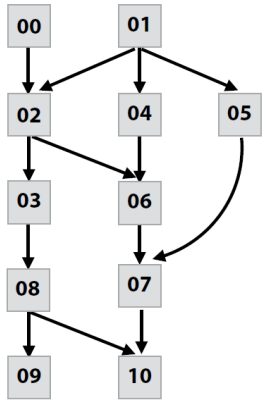
18

A more complex example

Program (sequence of instructions)

PC	Instruction	
00	a = 2	
01	b = 4	
02	tmp2 = a + b	// 6
03	tmp3 = tmp2 + a	// 8
04	tmp4 = b + b	// 8
05	tmp5 = b * b	// 16
06	tmp6 = tmp2 + tmp4	// 14
07	tmp7 = tmp5 + tmp6	// 30
08	if (tmp3 > 7)	
09	print tmp3	
	else	
10	print tmp7	

Instruction dependency graph

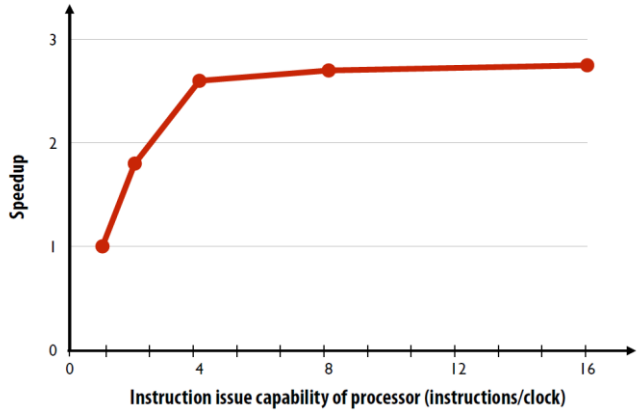


What does it mean for a superscalar processor to "respect program order"?

19

Diminishing Benefits of Superscalar

Most available ILP is exploited by a processor capable of issuing four instructions per clock (Little performance benefit achieved from building a processor that can issue more)



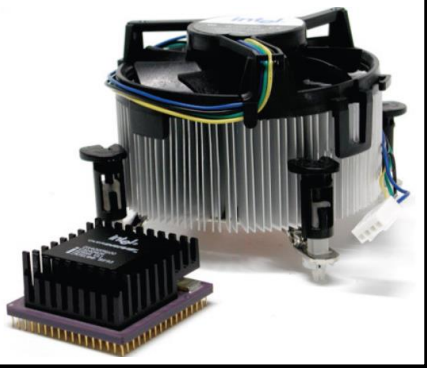
Source: Culler and Singh

20

The Power Wall

- Power is consumed by a transistor α C x Vdd^2 x Freq
- Static power is consumed even when the transistor is inactive due to leakage
- Power is a critical factor in design for all computing systems

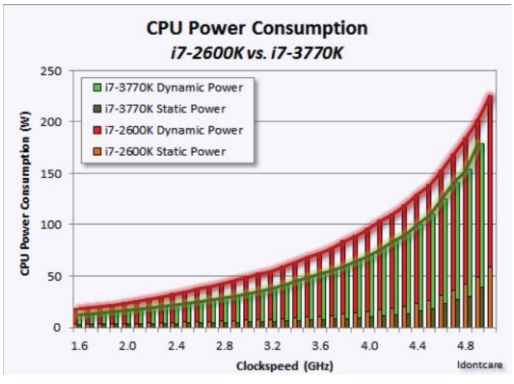
	TDP
Intel Core i7 (in this laptop):	45W
Intel Core i7 2700K (fast desktop CPU):	95W
NVIDIA GTX 780 GPU	250W
Mobile phone processor	1/2 - 2W
World's fastest supercomputer	megawatts
Standard microwave oven	700W



21

Static and Dynamic Power

- With billions of transistors, power consumption both static and dynamic will be major impediment

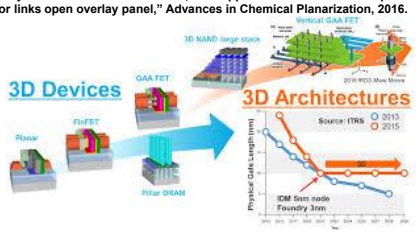


22

New Devices? New Architectures?

CMOS logic technology roadmap							
Year	2012	2014	2016	2018	2020	2022	2024
Technology node (nm)	22	14	10	7	5	3.5	2.5
Interconnect width (nm)	44	28	20	14	10	8	6
Interconnect pitch (nm)	88	56	40	28	20	16	12
Devices	FinFET						
				Nanowire/tunnel FETs			
		SiGe		High moblity channel materials			
					2D materials		
Interconnect	Dual damascene						
				Alternatives—Cu subtractive etch			
Memory	SRAM and eDRAM						
				MRAM			
Lithography	Immersion						
					EUV		
Patterning	Multi-patterning						
							Self-assembly

Courtesy: M.Krishnan and M.F.Lofaro, "2- Copper chemical mechanical planarization (Cu CMP) challenges in 22 nm back-end-of-line (BEOL) and beyond Author links open overlay panel," Advances in Chemical Planarization, 2016.



Courtesy: Perspective: New Process Technologies required for future devices

23

CMP Scaling Trends

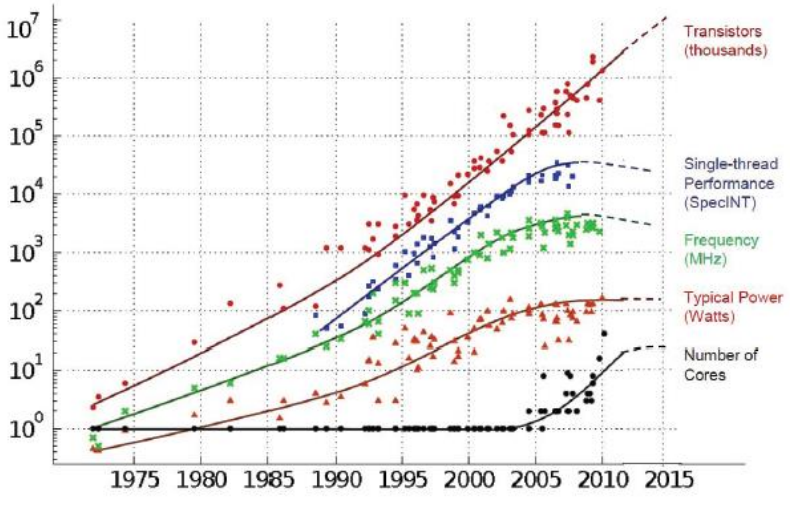
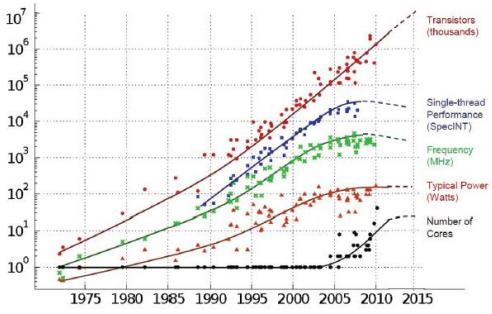


Image credit: "The free Lunch is Over" by Herb Sutter, Dr. Dobbs 2005

24

Limits of Single Core Performance Scaling

- Frequency scaling limited by power
- ILP scaling tapped out
- Architects are building faster processors by adding more execution units that can run in parallel (or units that are specialized for a specific task such as an accelerator – graphics, video, etc)
- Software written must be parallel to see such performance, no more free lunch for software developers!



25

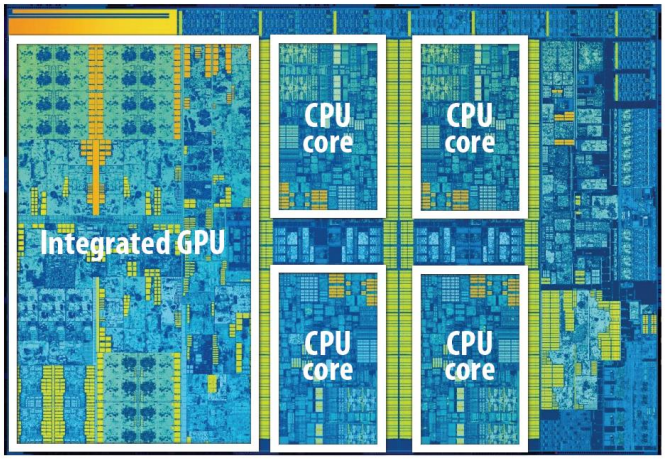
So Why Parallelism

- 15 years ago –
  - To realize performance improvements that exceeded single-thread performance
  - Because if you waited until next year, your code will run faster on the next generation of CPU
- Today –
  - Because it is the primary way to achieve higher application performance for the foreseeable future!

26

Intel Skylake (2015)

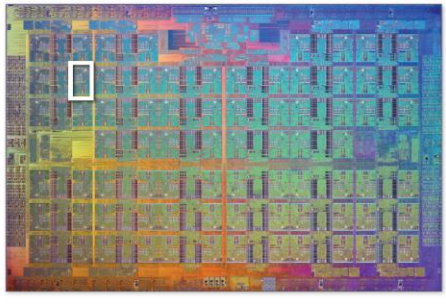
- Quad-core CPU + multicore CPU



27

Intel Xeon Phi 7290

72 cores (1.5 Ghz)



28

NVIDIA Maxwell GTX 1080 GPU (2016)

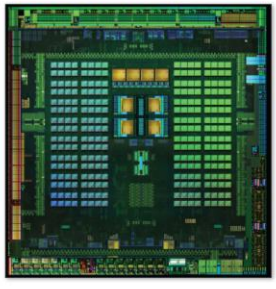
20 major processing blocks
(but much, much more parallelism available… details coming in a future class)

29

Mobile Parallel Processing

Power constraints heavily influence design of mobile systems

Apple A10: (in iPhone 7)
2 "big" CPU cores + 2 "small" CPU cores + GPU + image processor (and more!) on one chip



NVIDIA Tegra X1:
4 ARM A57 CPU cores +
4 ARM A53 CPU cores +
NVIDIA GPU + image processor…

30

Mobile Parallel Processing

Raspberry Pi 3

Quad-core ARM A53 CPU

31

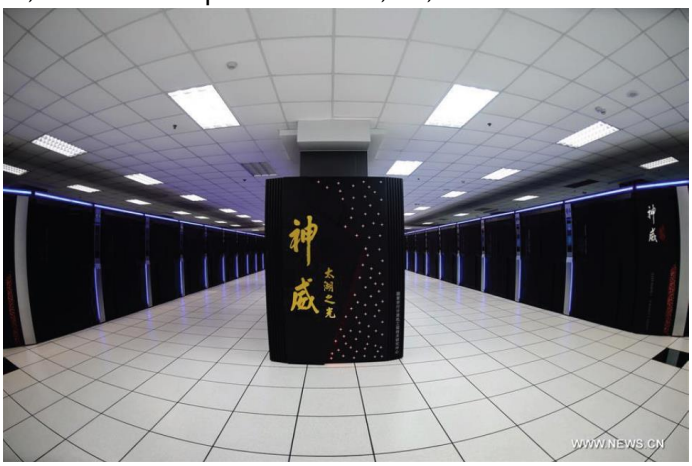
Supercomputing

- Today: clusters of CPUs + GPUs
- Oak Ridge National Labs: Titan
  - 18,688 x 16 core AMD CPUs + 18,688 NVIDIA K20X GPUs

32

Supercomputing

- Sunway TaihuLight
  - 40,960 x 256 core processors = 10,485,760 CPU cores



33

Summary

- Today, single-thread performance is improving slowly (lack of parallelism, power wall, etc)
  - To run programs significantly faster, programs must utilize multiple processing elements
- Writing parallel programs can be challenging
  - Requires problem partitioning, communication and synchronization
  - Knowledge of machine characteristics is important
- Modern computers have tremendously more processing power than you might realize, if you can just use it!

34