Lesson 3-2 Algorithmic Time, Energy, Power

Speed Trends

Processor speeds double every two years.

Speed Limits

To reach super high speeds a sequential computer would have to be really tiny.

Space Limits

At some point, the physical limits of size will be reached, so locality will have to be considered to increase speed.

Balance in Time

There is a growing gap between compute speed and communication speed.

$R \rightarrow$ [ops]/[time] $\rightarrow$ this is related to transistor density

Computation has gotten very fast.
Stream (β) is the speed of data transfer between slow and fast memory. This has doubled every 2.9 years.

Stream is much slower than computation.

$B == R/\beta$, this doubles every 5.5 years (this is the balance point)

Balance Principles

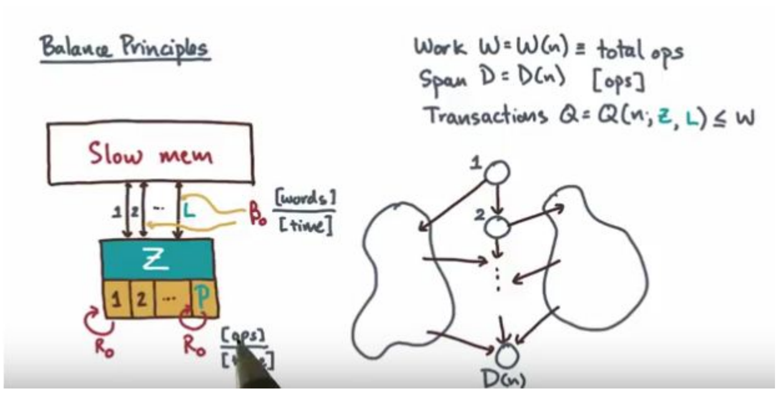
This implies, trading less communication for more computation.

Look at a DAG Model:
- Work $W = W(n)$ == total operations
- Span $D = D(n)$ [ops]
- $Q = Q(n;Z,L) \leq W$

The machine model:
P = # of processors

Transactions are the time it takes data to go from slow memory to fast memory.
Each transaction initiates a transfer of data over the wires, in parallel. The time it takes words to cross the wire is β_0.

Recall W, D, Q count the number of operations and ignore the costs of transportation.



.

The time for the process of the given DAG example:
$T_p \geq \max(D/R_0, W/(PR_0, QL/B_0)$

The right hand side is minimized when $W/P \gg D$
To benefit from transistor trends we want the compute time to dominate the communication time:
The Balance Principle → $W/PR_0 \geq QL/B_0$
Achieving balance is the best shot at scaling in the future.

$W/Q \geq (R_0/B_0)^* PL$
Therefore the left hand side (W/PR_0) should be as large as possible since the right hand side ($(R_0/B_0)^* PL$) will grow over time.

Double, Double, Toil and Trouble Quiz
Suppose a machine is perfectly balanced for sorting large arrays. If the number of cores doubles, how can balance be maintained?
For sorting: $W/Q \sim L \log(Z/L)$
Recall:
$W/Q \geq (R_0/B_0)^* PL$
$\rightarrow L \log(Z/L) \geq (R_0/B_0)^* PL$
$\rightarrow \log(Z/L) \geq (R_0/B_0)^* P$

Then double the cores:
$\rightarrow \log(Z/L) \geq (R_0/B_0)^* 2P$

The answer: Square Z and Square L or double the bandwidth

Squaring Z and L is a very expensive way to maintain balance.
But, bandwidth does not grow fast.

Power Limits
Power = Energy/Time
Increasing clock frequency makes the power consumption skyrocket, which is why multi-cores are prevalent.

Constant power = static power and idle power
Power = Constant Power + Dynamic Power
$P = P_0 + \Delta P$

Energy per Gate = CV^2
f = clock frequency, which is the maximum number of times a circuit can switch
a = activity factor, the number of switches per cycle

The Dynamic Power Equation
Dynamic Power = $CV^2 * f * a$
$f \propto$ Voltage $\rightarrow$ freq and voltage must change together to maintain the stability and reliability of the circuit.

Power Motivates Parallelism
Given the following processors:

f_1 = 4 GHz	f_2 = 1 GHz
ΔP = 64 watts	ΔP = 1 watts
T_1 = Time to run program	T_2 = Time to run program = $4T_1$

$CV^2 * f * a$

So by reducing the frequency by ¼ the power is reduced by 1/64.
This is a good argument for using parallelism to speedup instead of clock frequency.

Power Knobs

Recall the Dynamic power Equation: $CV^2 * f * a$
There are 4 factors that can change dynamic power:

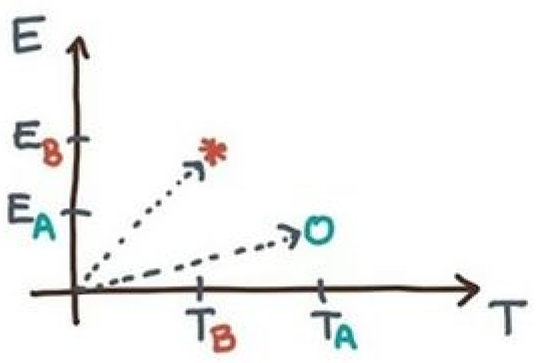
1. Capacitance
2. Supply Voltage

3. Clock Frequency
4. Activity factor

Which of the four factors can be controlled by software?
Voltage,frequency, activity

Capacitance - it is a geometric characteristic
Dynamic Voltage and Frequency Scaling (DVFS) - some processors allow this to be controlled by software.
Activity factor - if you know there are large portions of the algorithm that do not require certain hardware, that part of the processor can be turned off.

Powerless to Choose



System A has lower average power.
Avg Power = E/T.

So power corresponds to the slope of the lines.

Exploiting DVFS
Given two systems:
System A and System B

$E_B = 2\,E_A$ and $T_B = \frac{1}{3}\,T_A$

Suppose you use DVFS to rescale B, so that its power matches A.
Will B still be faster than A? Yes

$$\frac{P_B}{P_A} = \frac{E_B / T_B}{E_A / T_A} = \frac{E_B}{E_A} \cdot \frac{T_A}{T_B} = 6.$$

$$\text{DVFS} \Rightarrow \Delta P \propto f^3$$

B was re-scaled to match A, call it C.

$$\frac{P_B}{P_A} = 6, \quad P \propto f^3$$

$${}_B f_C^3$$

$$\frac{P_B}{P_A} = \frac{k_B f_B^3}{k_A f_A^3} = 6$$

$$\frac{f_B}{f_A} = \left(\frac{K_A}{K_B}\right)^{1/3} 6^{1/3}$$

$$\frac{P_C}{P_A} = \frac{k_B f_C^3}{k_A f_A^3} = 1$$

$$\frac{f_C}{f_A} = \left(\frac{K_A}{K_B}\right)^{1/3}$$

Recall time is inversely proportional to frequency.

$$\frac{T_A}{T_C} = \frac{T_A}{T_B} \cdot \boxed{\frac{T_B}{T_C}}$$

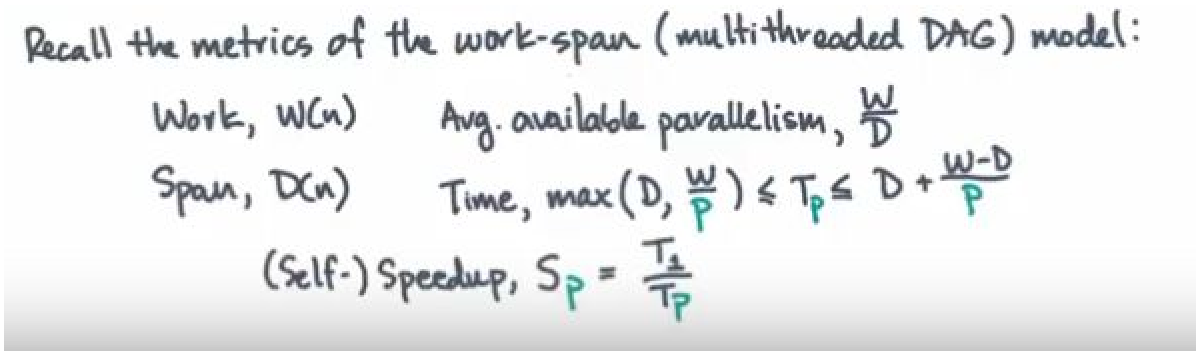
→

$$\frac{T_A}{T_C} = \frac{T_A}{T_B} \cdot \frac{f_C}{f_B}$$

$$\frac{T_A}{T_C} = \underbrace{\frac{T_A}{T_B}}_{=3} \cdot \underbrace{\frac{f_C}{f_A} \cdot \frac{f_A}{f_B}}_{=\frac{1}{6^{1/3}}} = \left(\frac{9}{2}\right)^{1/3}$$

Since the result is >1, system B is faster than system A.

Algorithmic Energy
Time: can be reduced or hidden by overlap (parallelism)
Energy: You must pay energy cost for every operation


Recall the metrics of the work-span (multithreaded DAG) model:

Work, $W(n)$ Avg. available parallelism, $\frac{W}{D}$

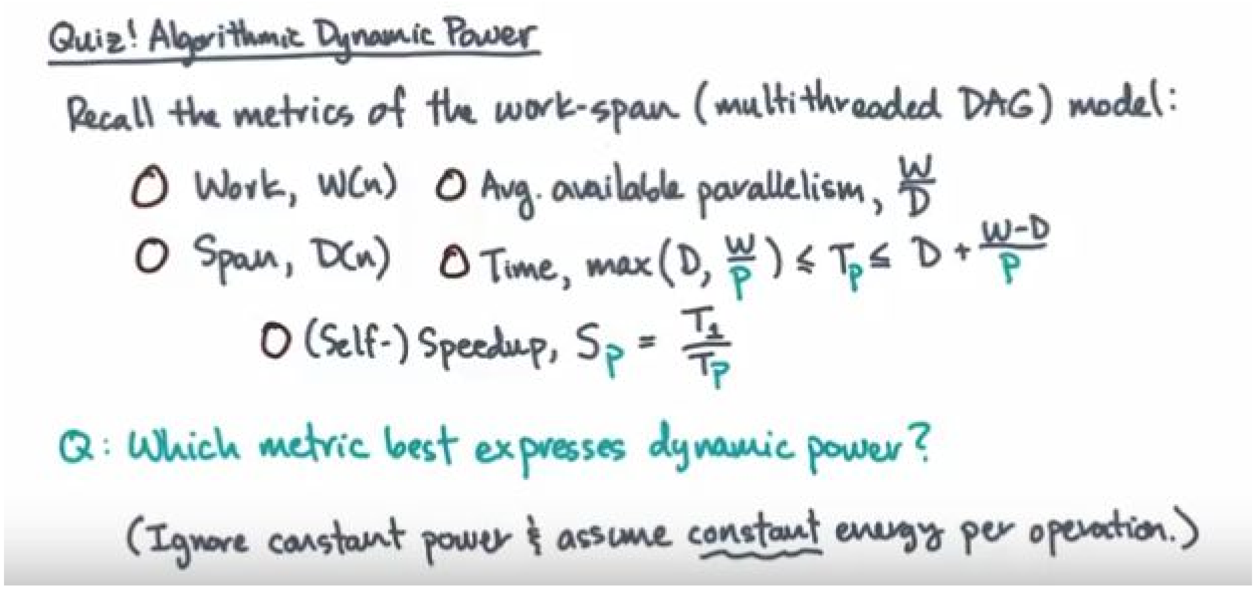
Span, $D(n)$ Time, $\max(D, \frac{W}{P}) \le T_P \le D + \frac{W-D}{P}$

(Self-) Speedup, $S_P = \frac{T_1}{T_P}$

T_1= the time of the algorithm when run in parallel on a single processor.

Work W(n) best quantifies energy. For energy - you must account for EVERY operation. For work you must account for every operation.

Algorithmic Dynamic Power


Quiz! Algorithmic Dynamic Power

Recall the metrics of the work-span (multithreaded DAG) model:

O Work, $W(n)$ O Avg. available parallelism, $\frac{W}{D}$

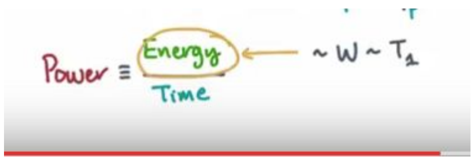
O Span, $D(n)$ O Time, $\max(D, \frac{W}{P}) \le T_P \le D + \frac{W-D}{P}$

O (Self-) Speedup, $S_P = \frac{T_1}{T_P}$

Q: Which metric best expresses dynamic power?

(Ignore constant power & assume constant energy per operation.)

Self-Speedup best expresses dynamic power.



$$\text{Power} \equiv \frac{\boxed{\text{Energy}}}{\text{Time}} \longleftarrow \sim W \sim T_1$$