Computational energy, time, power and action

Robert W. Numrich and Robert M. Haralick

The Graduate Center
City University of New York

Frankfurt, 16 July 2015

The question

- How many joules does it take to perform a floating-point operation? To move a byte of data?
  - It depends. Depends on what?
  - How do algorithms interact with hardware?
  - What can programmers do to reduce energy dissipation?
  - What can hardware designers do to reduce energy dissipation?

Correspondence between computational and electrical quantities

electrical		↔	computational	
quantity	sybmol	↔	quantity	symbol
time (s)	t	↔	time (s)	t
charge (coulomb)	q	↔	length (byte)	x
energy (joule)	w	↔	energy (flop)	e
voltage	V	↔	force	f
capacitance	C	↔	spring	k
inductance	L	↔	mass	m
resistance	R	↔	dashpot	b

An electrical-computational model

Newton's Second Law	$MA =$	$-F_1$	$-F_2$	$+F_3$
electrical system	$L\ddot{q} =$	$-R\dot{q}$	$-q/C$	V
	inertia	current	capacitor	voltage
computational system	$m\ddot{x} =$	$-b\dot{x}$	$-x/k$	f
	latency	bandwidth	memory	force

- The electrical system

  $$L\ddot{q} + R\dot{q} + q/C = V \ , \quad q(0) = 0, \dot{q}(0) = 0$$

- The computational system

  $$m\ddot{x} + b\dot{x} + x/k = f \ , \quad x(0) = 0 \ , \dot{x}(0) = 0$$

Does a mechanical model of computation make any sense?

quantity	symbol	unit	dimension
time	t	s	T
length	x	byte	L
energy	e	flop	E
frequency	ν	Hz	T^{-1}
velocity (bandwidth)	v	byte $\cdot$ s^{-1}	LT^{-1}
power	r	flop $\cdot$ s^{-1}	ET^{-1}
action	s	flop $\cdot$ s	ET
force (intensity)	f	flop $\cdot$ byte^{-1}	EL^{-1}
spring (storage)	k	flop^{-1} $\cdot$ byte^2	$E^{-1}L^2$
mass (latency)	m	flop $\cdot$ s^2 $\cdot$ byte^{-2}	ET^2L^{-2}
dashpot (friction)	b	flop $\cdot$ s $\cdot$ byte^{-2}	ETL^{-2}

Energy dissipated by the resistor

- Initial energy

  $e(0) = 0 \ , \quad w(0) = 0$

- Final energy

  $e(t_*) = (1/2)x_*^2/k - x_* f \qquad x_* = \text{bytes moved}$

  $w(t_*) = (1/2)q_*^2/C - q_* V \quad q_* = \text{charge moved}$

- The answer to our question:

  $\mu(t_*) = w(t_*)/e(t_*) \ (\text{J/flop})$

- But we don't know the values for any of the quantities involved!

The forced pendulum with friction

- The Pi Theorem of dimensional analysis tells us how to scale the equations to dimensionless form.

  $\ddot{z} + \rho\dot{z} + z = 0 \ , \quad z(0) = -1 \ , \dot{z}(0) = 0$

  $\ddot{z} + \beta\dot{z} + z = 0 \ , \quad z(0) = -1 \ , \dot{z}(0) = 0$

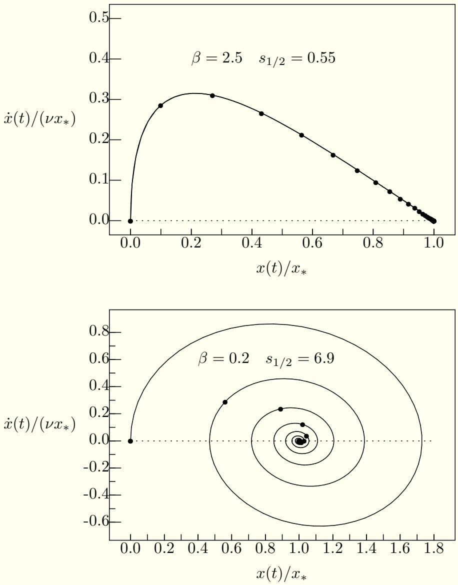
- In each case, the solution depends on the value of just one dimensionless parameter

  $\rho < 2 < \rho$ friction in the electrical system

  $\beta < 2 < \beta$ friction in the computational system

  - V.I. Arnol'd. *Ordinary Differential Equations*, pp. 174-176; 191-192, Springer-Verlag, 3rd edn (1992)

Phase portrait: global attractor at (1,0)



Imposing final conditions: Magic happens

$$x(t) = (kf) \cdot (z_\beta(\nu t) + 1) \quad \lim_{t\to\infty} z_\beta(\nu t) = 0$$
$$\lim_{t\to\infty} x(t) = kf = x_*$$

- The mysterious quantities k and m are determined by measureable quantities

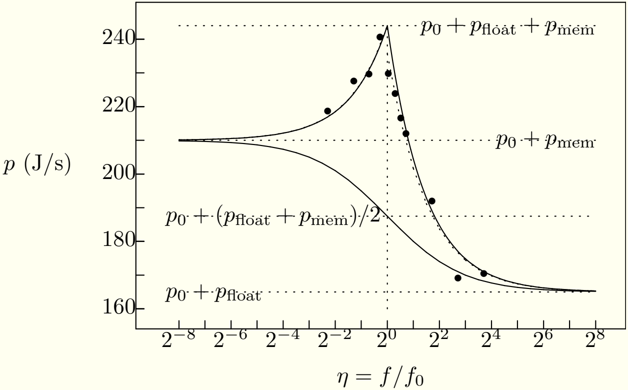
$$k = x_*/f$$
$$m = f/(x_* \nu^2)$$

- A better answer to our question:

$$\mu_*(f) = \lim_{t_*\to\infty} \mu(t_*) = (q_*/x_*)(V/f) \text{ (J/flop)}$$

- But we still don't know what values to use for the quantities involved.

The power envelope



- Choi, Bedard, Fowler, Vuduc, IPDPS 2013.
- Measurements on Nvidia 580

A formula that represents Choi's data

$$p(\eta) - p_0 = \begin{cases} \eta(p_{\text{float}} + p_{\text{mem}}/\eta)/(1+\beta\eta) & \eta \le 1 \\ \eta(p_{\text{float}} + p_{\text{mem}}/\eta)/(\alpha+\eta) & \eta > 1 \end{cases}$$

$\eta = f/f_0$

$f_0 = r_0/b_0$

$\alpha = \beta = 0 \implies$ total overlap

$\alpha = \beta = 1 \implies$ no overlap

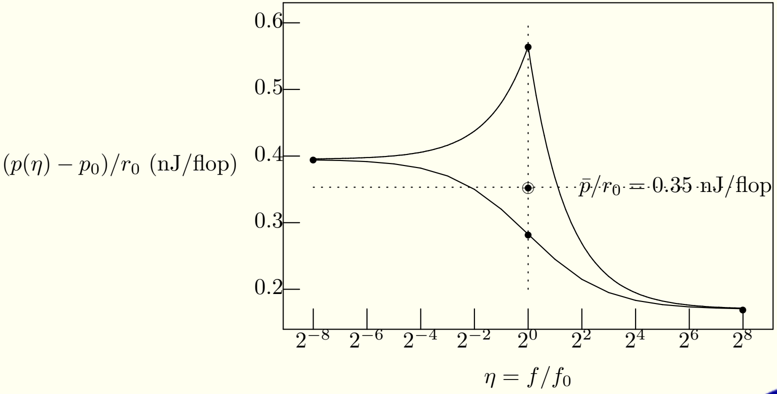
$\lim_{\eta\to\infty}(p(\eta) - p_0) = p_{\text{float}}$

$\lim_{\eta\to 0}(p(\eta) - p_0) = p_{\text{mem}}$

Choi's measurements

quantity	value	units
r_0	197	Gflop/s
b_0	162	Gbyte/s
$f_0 = r_0/b_0$	1.3	flop/byte
p_0	131	J/s
p_{float}	34	J/s
p_{mem}	79	J/s
p_0/r_0	0.66	nJ/flop
p_{float}/r_0	0.17	nJ/flop
p_{mem}/r_0	0.40	nJ/flop

Centroid of the power envelope



$$\bar{p} = (5/8)(p_{\text{mem}} + p_{\text{float}})$$

Correlating our model with Choi's measureents

- Recall our formula for joules per flop:

  $\mu_*(f) = (q_*/x_*)(V/f)$

  $\mu_*(f_0) = (q_*/x_*)(V/f_0)$

- What happens if we equate this quantity with the power at the centroid?

  $\mu_*(f_0) = \bar{p}/r_0$

- We can calculate the unknown quantity

  $(q_* V/x_*) = f_0(\bar{p}/r_0)$

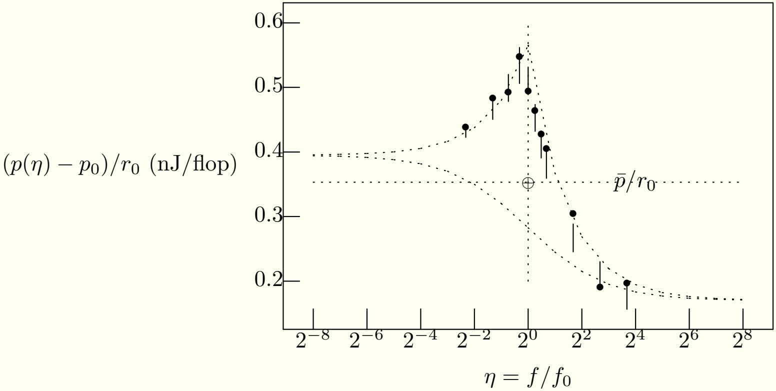
  $(q_* V/x_*) = (1.3 \text{ flop/byte})(0.35 \text{ J/flop}) = 0.46 \text{ nJ/byte}$

- On the other hand, if we know the value of $(q_* V/x_*)$, we can compute the value of the quantity $\bar{p}/r_0$.

Does the value of $(q_* V / x_*)$ make any sense?

quantity	value	units
V_{chip} (chip volume)	10^{-1}	cm^3 per chip
N_{chip} (chip capacity)	10^9	byte per chip
ρ_{byte} (byte density)	10	Gbyte/cm^3
ρ_{e} (electron density in Si)	2.8×10^{19}	$\mathrm{electron}/cm^3$
$\sigma = \rho_{\mathrm{e}} V_{\mathrm{chip}} / N_{\mathrm{chip}}$	2.8×10^9	electron/byte
e^-	1.6×10^{-19}	coulomb/electron
$\sigma e^- = (q_* / x_*)$	4.5×10^{-10}	coulomb/byte
$(q_* V / x_*)$	$0.45 V$	nJ/byte
Kogge Exa-Report		
ρ_{byte}	8-18	Gbyte/cm^3
$(q_* V / x_*)$	0.1-1.0	nJ/byte
V	0.2-1.4	J/coulomb

Sanity check

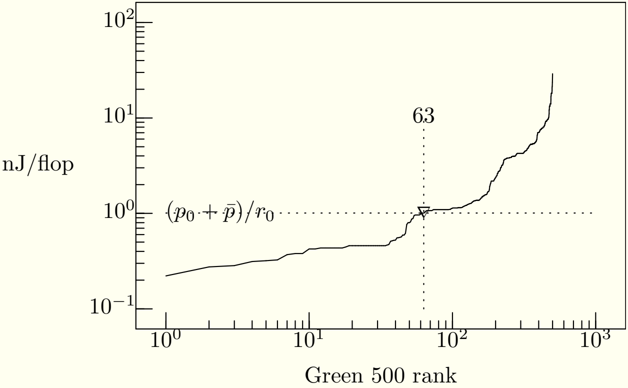


$$\eta = \mu_*(f_0)/\mu_*(f) = f(\bar{p}/r_0)/(q_* V/x_*) = 0.78(f/V)$$

$$V = 1.0 \pm dV$$

(•) measurement; (|) theory

The Green 500 (November 2013)



What have we learned?

- We have an *a priori* estimate for the energy dissipated as a function of computational force.

  $\mu_* = (q_*/x_*) \cdot (V/f)$ J/flop

- The energy used to store a byte of data depends on a particular machine.

  $(q_* V/x_*) = 0.46V$ nJ/byte

What can hardware designers do?

- Rewrite our basic relationship with hardware terms in red and software terms in blue:

  $\mu_*(f) = (1/f_0)(q_*/x_*)V \cdot (f_0/f)\ , \quad f_0 = r_0/b_0$
- Chip designers can reduce energy dissipation by:
  - Increasing the value of the hardware force f_0
  - Reducing the charge used to represent a byte of data, for example, by packing more bytes on a chip.
  - Lowering the voltage

What can programmers do?

- Now look at the blue piece:
  $\mu_*(f) = (1/f_0)(q_*/x_*)V \cdot (f_0/f)$, $\quad f_0 = r_0/b_0$
- Programmers can reduce energy dissipation by increasing the software force f until it is larger than the hardware force f_0
- That's all a programmer can do.
- It's in direct opposition to what the hardware designers are doing!

Have we really learned anything?

- Computational force has always been recognized as a very important quantity for performance analysis.
  - The name *computational intensity* is a misnomer
- Computational force is the same quantity that must be increased to optimize numerical algorithms, regardless of the amount of energy dissipated.
  - A.W. Burks, H.H. Goldstine, J. von Neumann (ca. 1946) Preliminary discussion of the logical design of an electronic computing instrument. In: *John von Neumann collected works*, vol V, p.38, Pergamon (1963)
  - R.W. Hockney and I.J. Curington. f-half: a Parameter to Characterise Memory and Communication Bottlenecks. Parallel Computing, 10:277-286 (1989)

References

- Peter M. Kogge (editor), ExaScale Computing Study: Technology Challenges in Achieving Exascale Systems. Technical Report CSE TR-2008-13, University of Notre Dame, September 28, 2008.
- James Demmel, Andrew Gearhart, Benjamin Lipshitz, and Oded Schwartz, Perfect strong scaling using no additional energy. IPDPS, pp 649-660 (2013)
- Jee Whan Choi, Daniel Bedard, Robert Fowler, and Richard Vuduc, A Roofline Model of Energy. IPDPS, pp. 661-672 (2013)
- R.W. Numrich, Computer performance analysis and the Pi Theorem, Comput Sci Res Dev, 29:45-71 (2014)
- R.W. Numrich, Computational force, mass, and energy, Int. J. Mod. Phys. C 8(3):437-457 (1997)

Dimensional analysis

- G. Birkhoff, *Hydrodynamics: a study in logic,fact and similitude*, 2nd edn. Princeton University Press, Princeton (1960)
- G.I. Barenblatt, *Scaling, self-similarity, and intermediate asymptotics*, Cambridge University Press, Cambridge (1996)
- P.W. Bridgman, *Dimensional analysis*, 2nd edn. Yale University Press, New Haven (1931)

Dimensional analysis

- A. Einstein, Elementare Betrachtungen uber die thermische Molekularbewegung in festen Korpern. Ann Phys 35:679694 (1911)

  "dimensionless parameters of physical systems ought to have values of order unity "

- M. Schechter, *Operator Methods in Quantum Mechanics*, Dover, New York (2002)

  "Planck's constant (a quantity physicists will have no difficulty remembering and mathematicians will have no difficulty forgetting)"