Defining Computer “Speed”: An Unsolved Challenge

Dr. John L. Gustafson, Director
Intel Labs

Thesis

The main reason we use computers is that they’re fast. Yet after 70 years of electronic digital computing, we can’t define the *speed of a computation* as a metric!

As a result, programmers choose the wrong methods, hardware designers optimize the wrong things, and customers often choose the wrong computers to buy.

Some Attempts to Find a Definition

1) Count arithmetic operations, divide by time.
2) Equate clock rate with speed.
3) Claim that “logical inferences” or “transactions” are units of computational work that can be standardized.
4) Use time ratios to measure *relative* speed.
5) Benchmark “representative kernels”

Let’s examine why these don’t work very well.

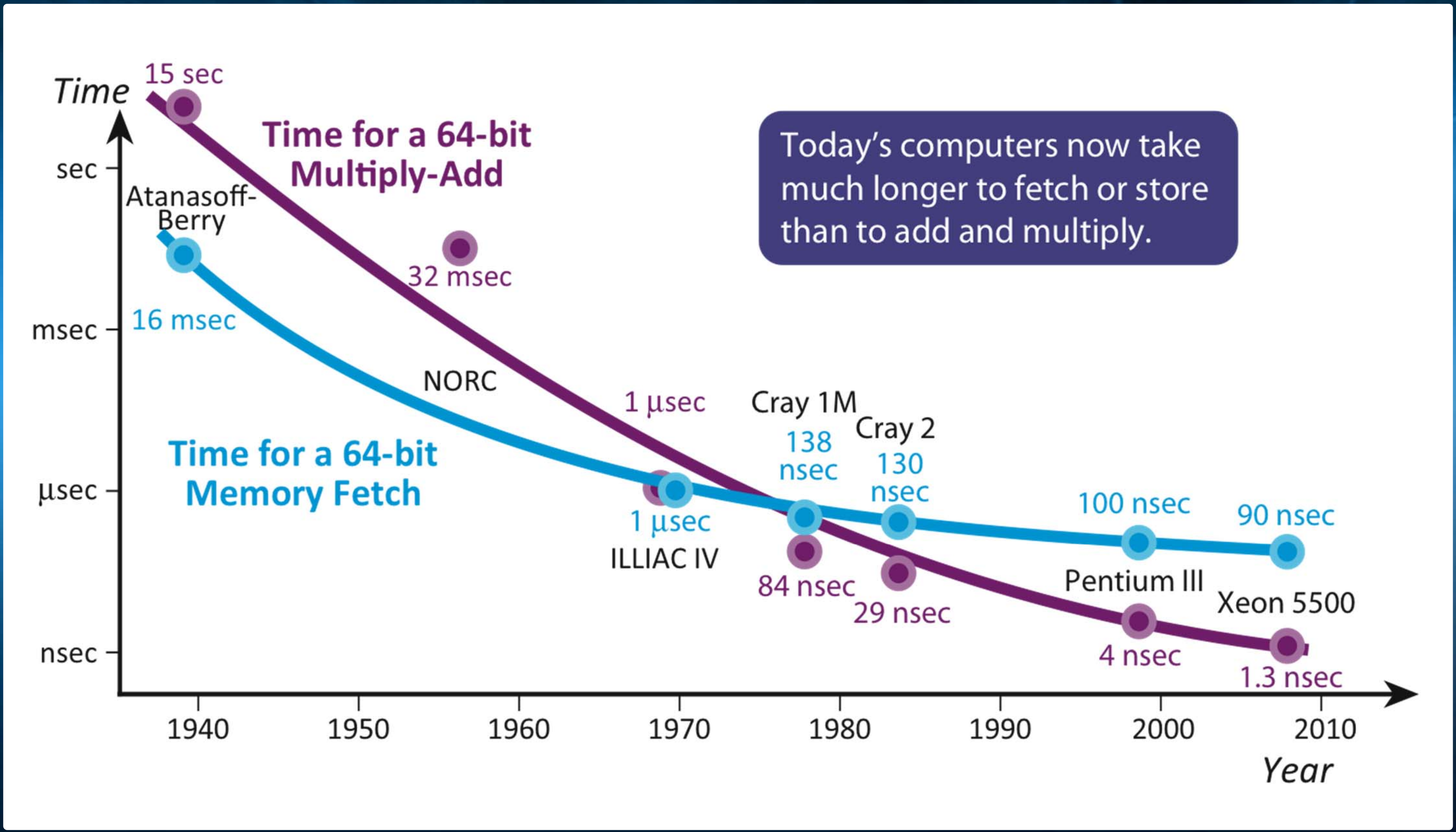
(1) The Good Old Days: Speed was (floating pt.) operations per second

If computer X has twice the peak floating-point rate of computer Y, a scientific code would take about half the time.

If an algorithm had half as many floating-point operations, it would take about half the time.

What happened?

Answer: The Memory Wall



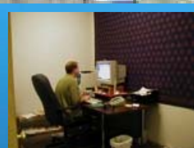
It’s so counterintuitive…

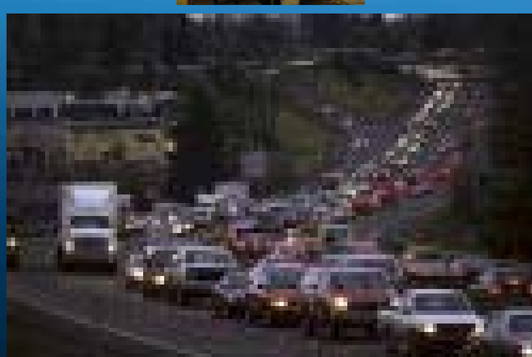
Copying numbers…

…is harder than doing THIS with them?

$$
\begin{array}{r}
0.583929181098938 \times 10^{12} \\
\times \quad 0.452284961938858 \times 10^{9} \\
\hline
4671433448791504 \\
2919645905494690 \\
4671433448791504 \\
4671433448791504 \\
1751787543296814 \\
5255362629890442 \\
583929181098938 \\
3503575086593628 \\
5255362629890442 \\
2335716724395752 \\
4671433448791504 \\
1167858362197876 \\
1167858362197876 \\
2919645905494690 \\
2335716724395752 \\
\hline
0.264102387448322 \times 10^{21}
\end{array}
$$

If Your Workday Were Like Current Computing Balance…





Spend 4 hours commuting from home to work.

Work for 2.4 minutes.

Take 4 hours to commute home.

What happened? Answer: The speed of light finally got to us.

1977:

Cray-1 was 80 MHz, or 12.5 nsec… speed of light in wire, in one clock cycle, was bigger than the machine.

The speed of light limits of CMOS

For a 3 GHz processor
chip, in a 333 psec
clock cycle, light goes…

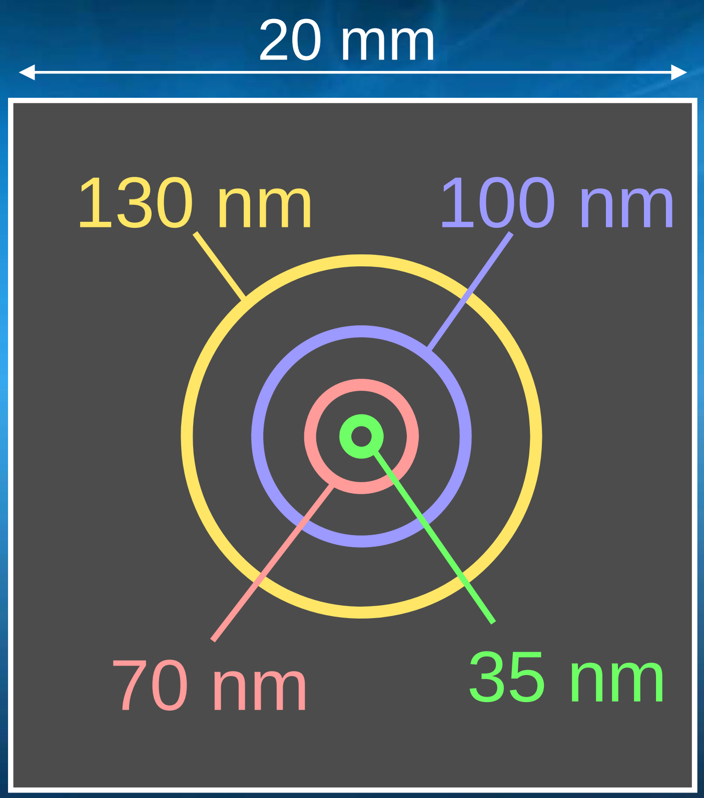
4 inches in a vacuum

2.5 inches in optical fiber

4 millimeters in an on-chip trace!

The Speed of Light Isn't What it Used to Be

- The clock can't make it across a 20 mm die at current GHz rates.
- "Speed of light" is only 4% of *c* on chip!
- Even fiber optics are only 60% of *c*. Latency of many nanoseconds for a big cluster.



[Source: Chuck Moore, UT-Austin]

Nature Understands the Memory Wall

Compute overlaps communication as the default behavior; only serialize it when there is no other way to get the correct answer.

Consider how biology does it:
100 billion processors in parallel
90% communicate, 10% compute
All computing is overlapped
…and the whole thing uses ~25 watts!

The Debate Still Rages…

System User

System Designer

Make the communication faster!

Can't. Near the speed of light now.

Then make the wires shorter.

If we do, the machine will melt.

What about liquid cooling?

Dead-end solution, and expensive.

How about optical interconnect?

Yeah, but the logic is electrical.

So use optoelectronic converters.

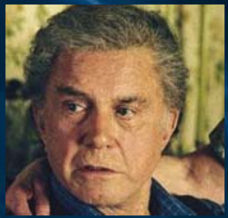
That adds latency everywhere.

Why don't you make the whole system optical?

*Why don't you just learn how to use machines with **distributed** memory?*

Because then we have to THINK.

I see.



With great power comes great responsibility

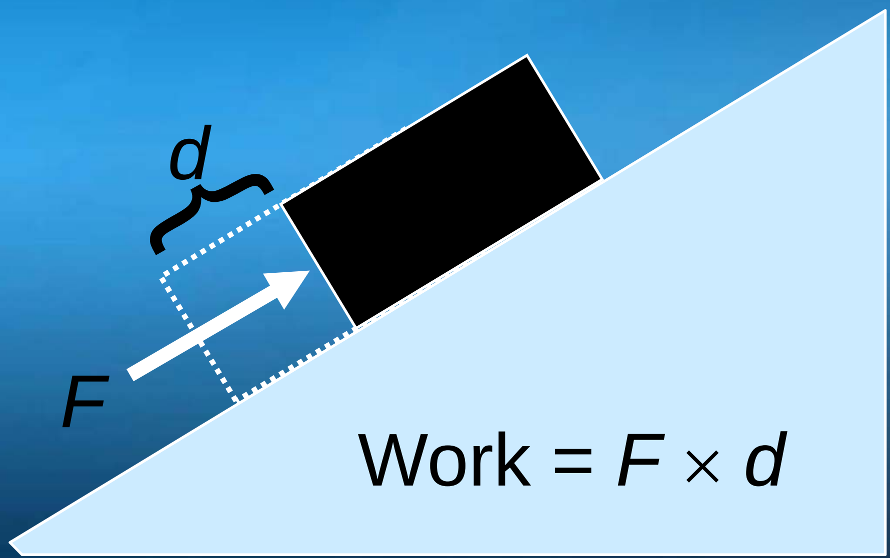
—*Uncle Ben*

Yes, and also some *really big* heat sinks.

—*John G.*

The Coming Weirdness-I

- Shared memory will *run hotter* than message passing, in general.
- Chip designers will try to conserve electrons and re-use them.
- Computational work might someday be best measured in… joules?



The Coming Weirdness-II

- Throttled processing. Drive a chip too hard, and it slows down to cool off.
- Algorithm “hot spots” …really are.

But back to measuring speed with arithmetic per second…

Floating Point Operation = FLOP, but tell me how to score these:

- X+Y
- X*Y
- X^Y
- If (X < Y)…
- |X|

- Sin(X)
- Log(X)
- Exp(X)
- Y / X
- 1 / X

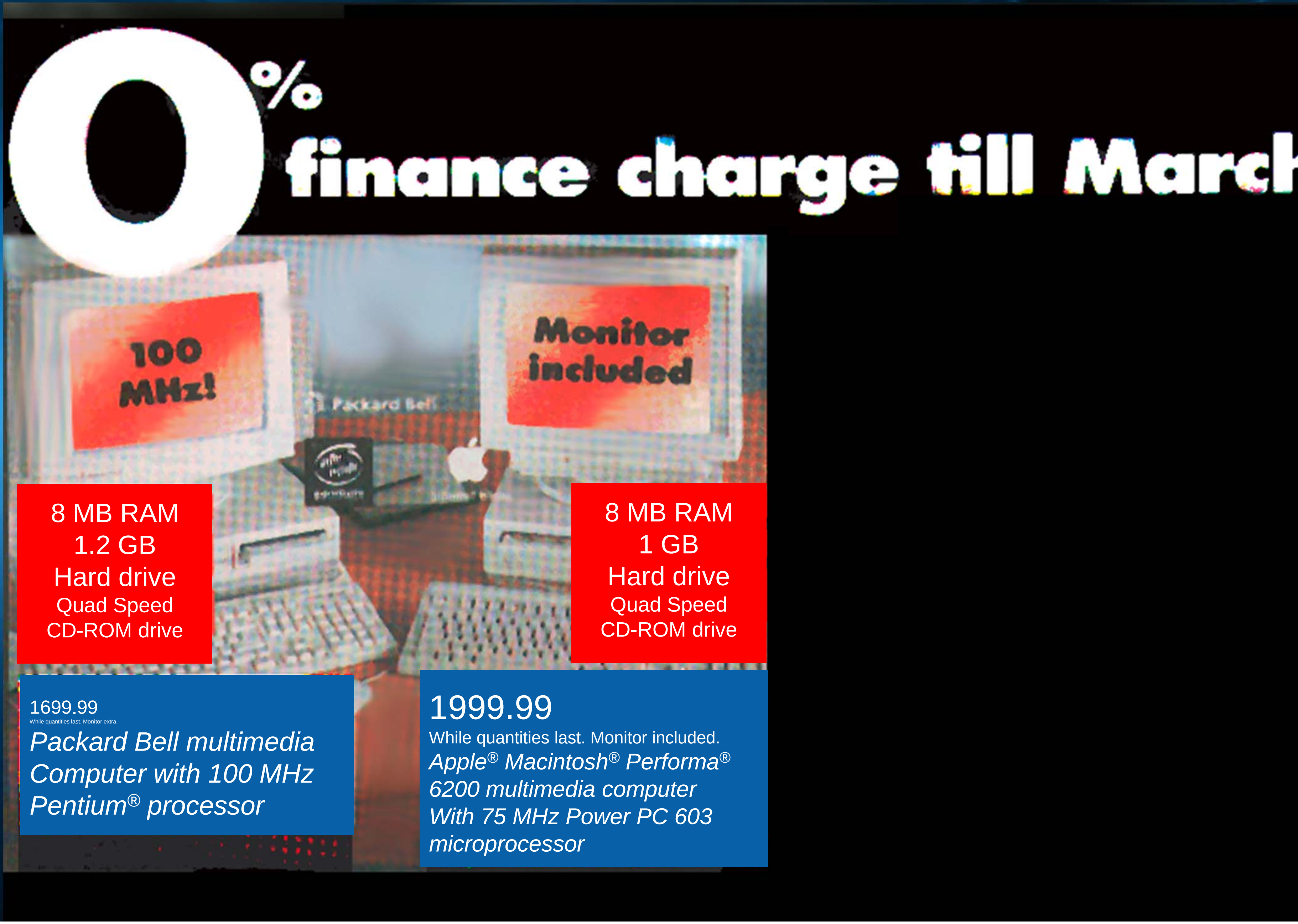
… and to what precision? What about rounding errors? IEEE standard does *not* solve this. And what about *integer* operations?

(2) Clock rate = Computer speed?

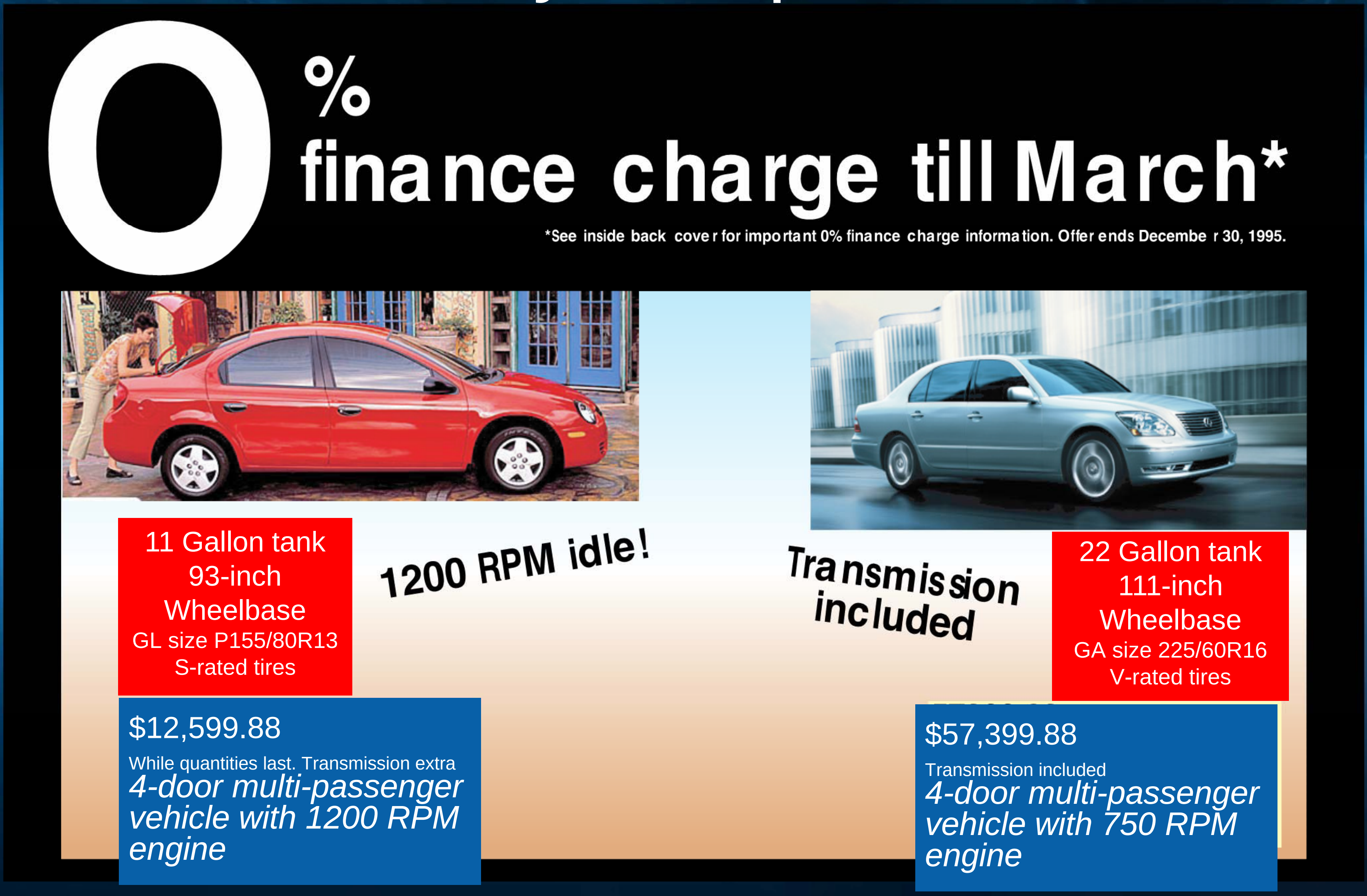
Marketing ploy,
like how many jewels
in a watch, or liters of
displacement in your engine.

Many consumers still think GHz is
how fast your PC is.

A Real Ad from the 1990s…



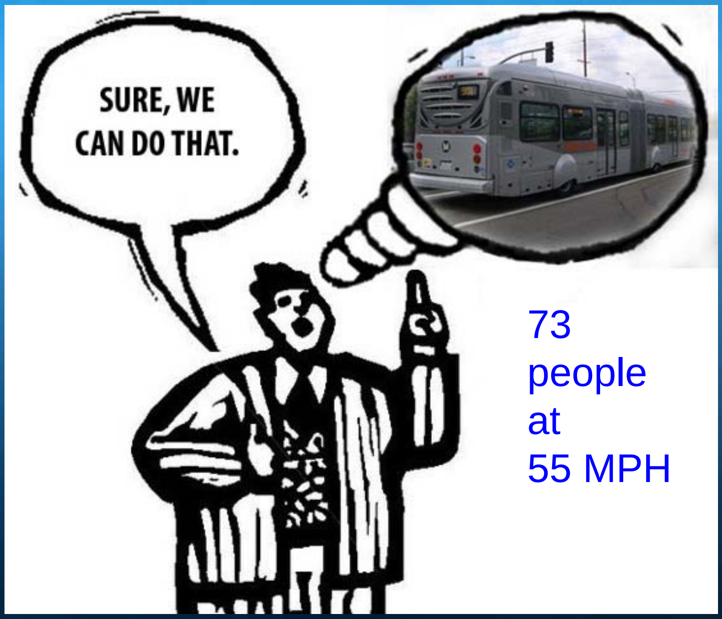
If We Bought Cars Like We Buy Computers…



(3) Define Transactions, Logical Inferences, etc. as units of "work"?

Those things cannot be made "standard."

…and some vendors don't understand difference between *capability* computing and *capacity* computing.



(4) Compute *relative* speed by taking the ratio of times

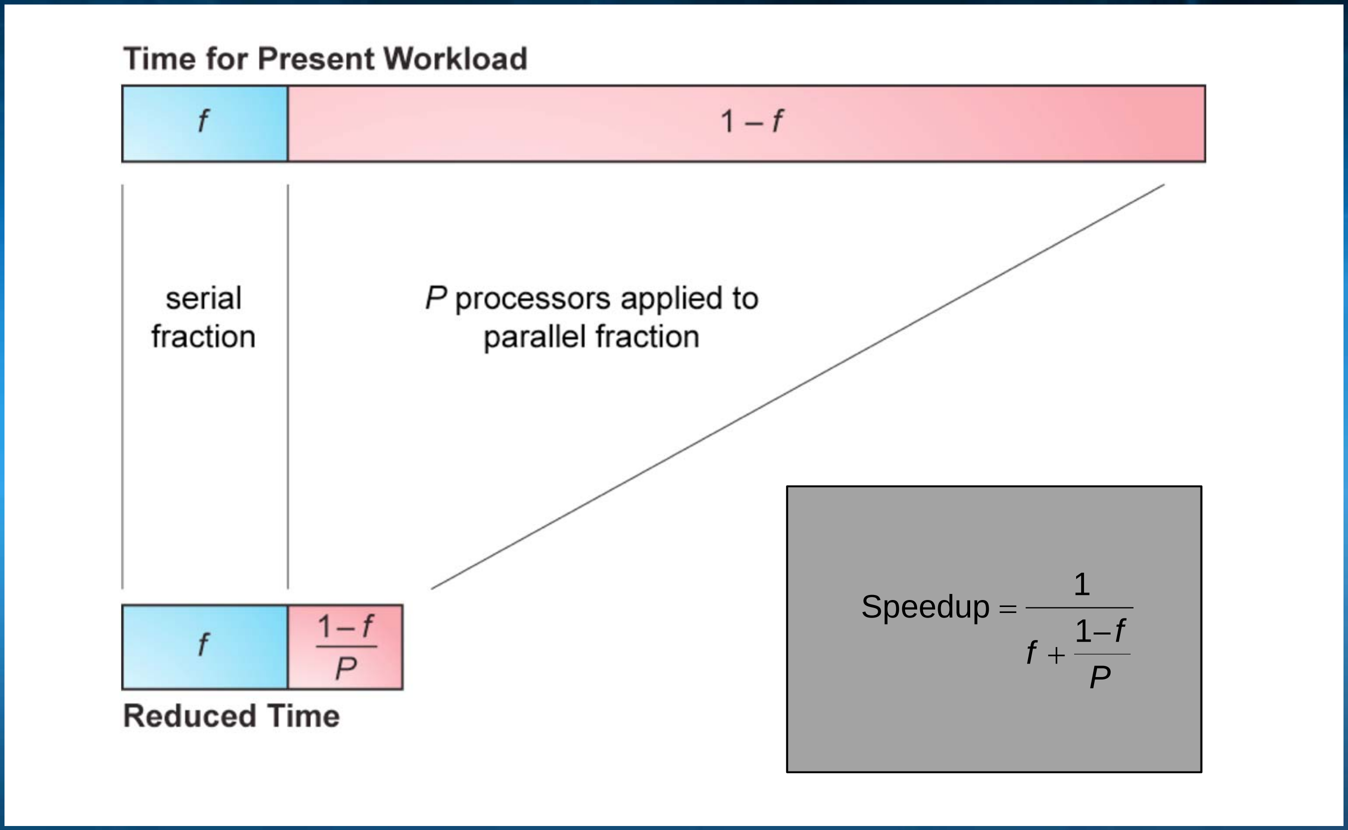
Speed1 = work1 / time1.

Speed2 = work1 / time2.

Speedup = Speed1 / Speed2 = time2 / time1.

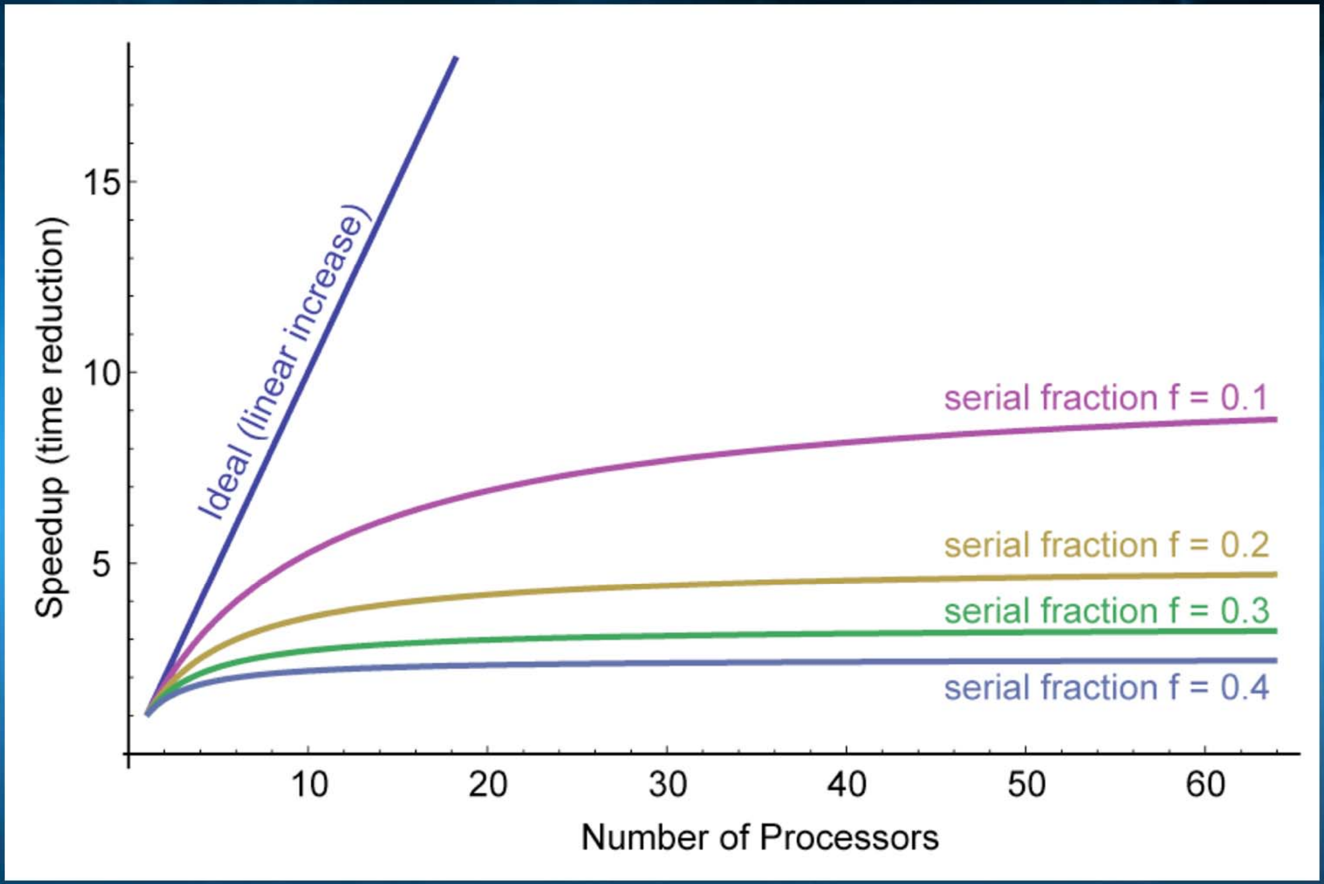
Sounds reasonable, right?
Yet, this approach held back parallel processing for 20 years!

Amdahl's law contains a hidden assumption about how people use computers.



…that people use computers to do what they do now, but in *less time*.

Amdahl's predictions were pessimistic



…so parallel processing was considered purely an academic fantasy, 1967-1988.

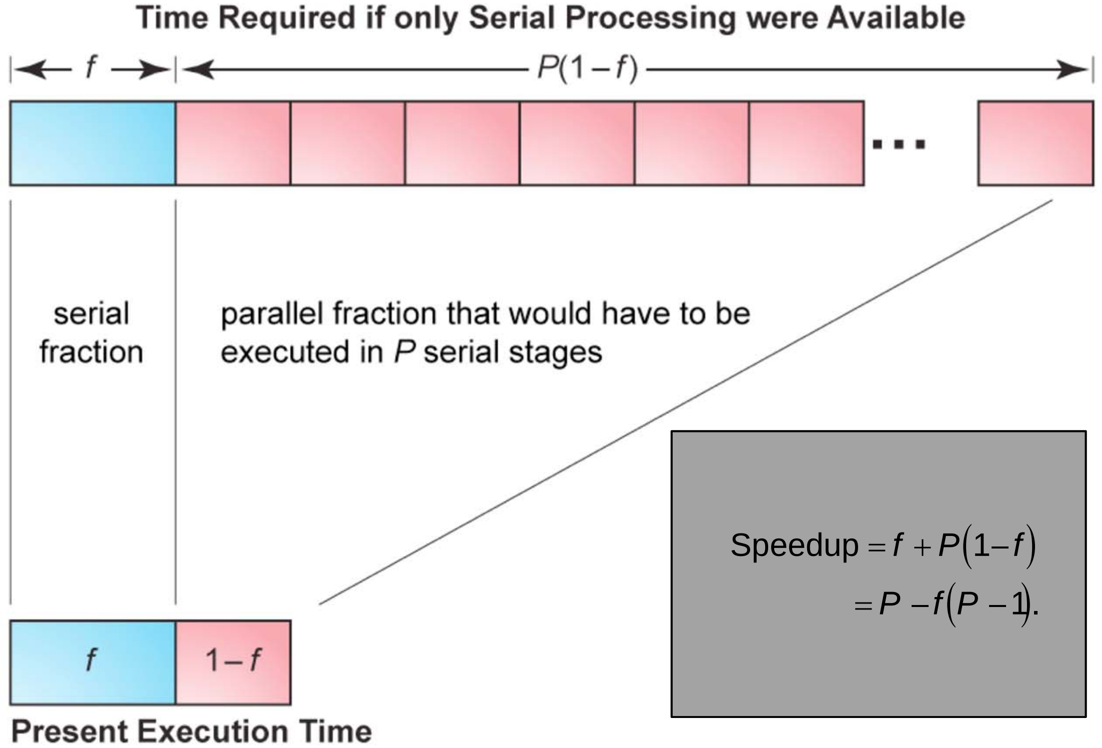
From Ambrose Bierce, *The Devil's Dictionary*

defined

Logic

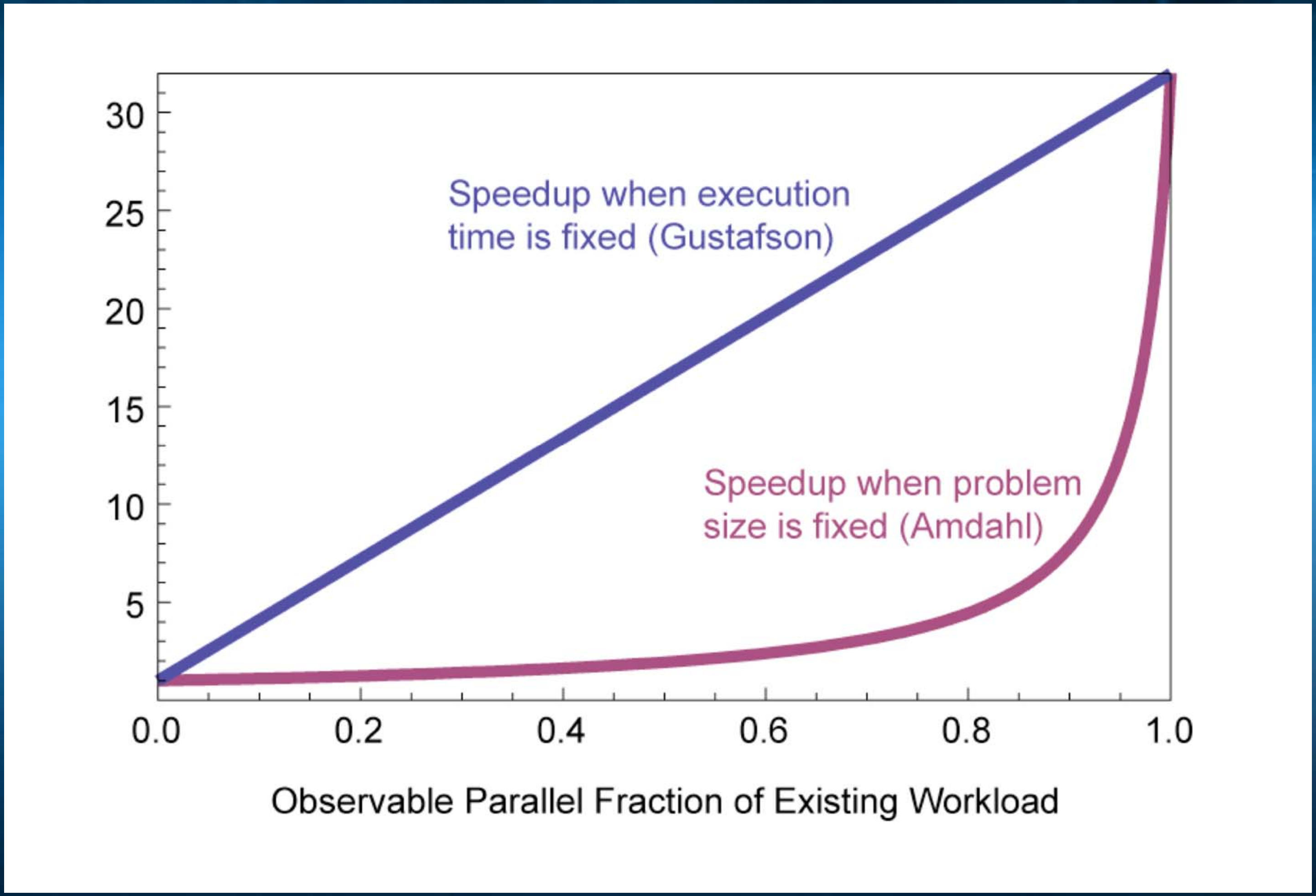
n. The art of thinking and reasoning in strict accordance with the limitations and incapacities of the human misunderstanding. The basic of logic is the syllogism, consisting of a major and a minor premise and a conclusion — thus: *Major Premise*: Sixty men can do a piece of work sixty times as quickly as one man. *Minor Premise*: One man can dig a posthole in sixty seconds; therefore — *Conclusion*: Sixty men can dig a posthole in one second. This may be called the syllogism arithmetical, in which, by combining logic and mathematics, we obtain a double certainty and are twice blessed.

What if $t_1 = t_2$, and *work* is what changes?



$$\text{Speedup} = f + P(1-f) = P - f(P-1).$$

Suddenly things look a lot better for parallel computing!

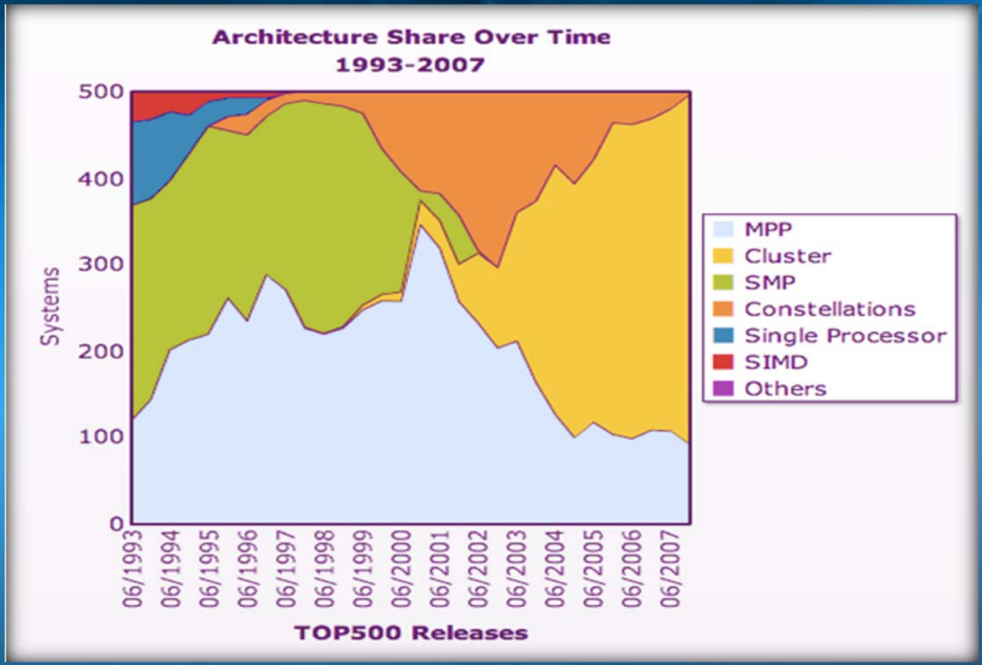


Back to the Memory Wall: Some Ideas

- Do much, much better arithmetic. Spend a few clocks to do 256-bit floating-point, say. Or use Interval Arithmetic.
- Add “freeways” to chips (carbon nanotubes?) for cross-chip communication closer to 100% of *c*
- Don’t distribute simulations across the Grid! Use local clusters, as tightly packed as the cooling system will permit.
- Stop using LINPACK Top 500 rankings. Please.

Looking Beyond LINPACK (HPL, TOP500)

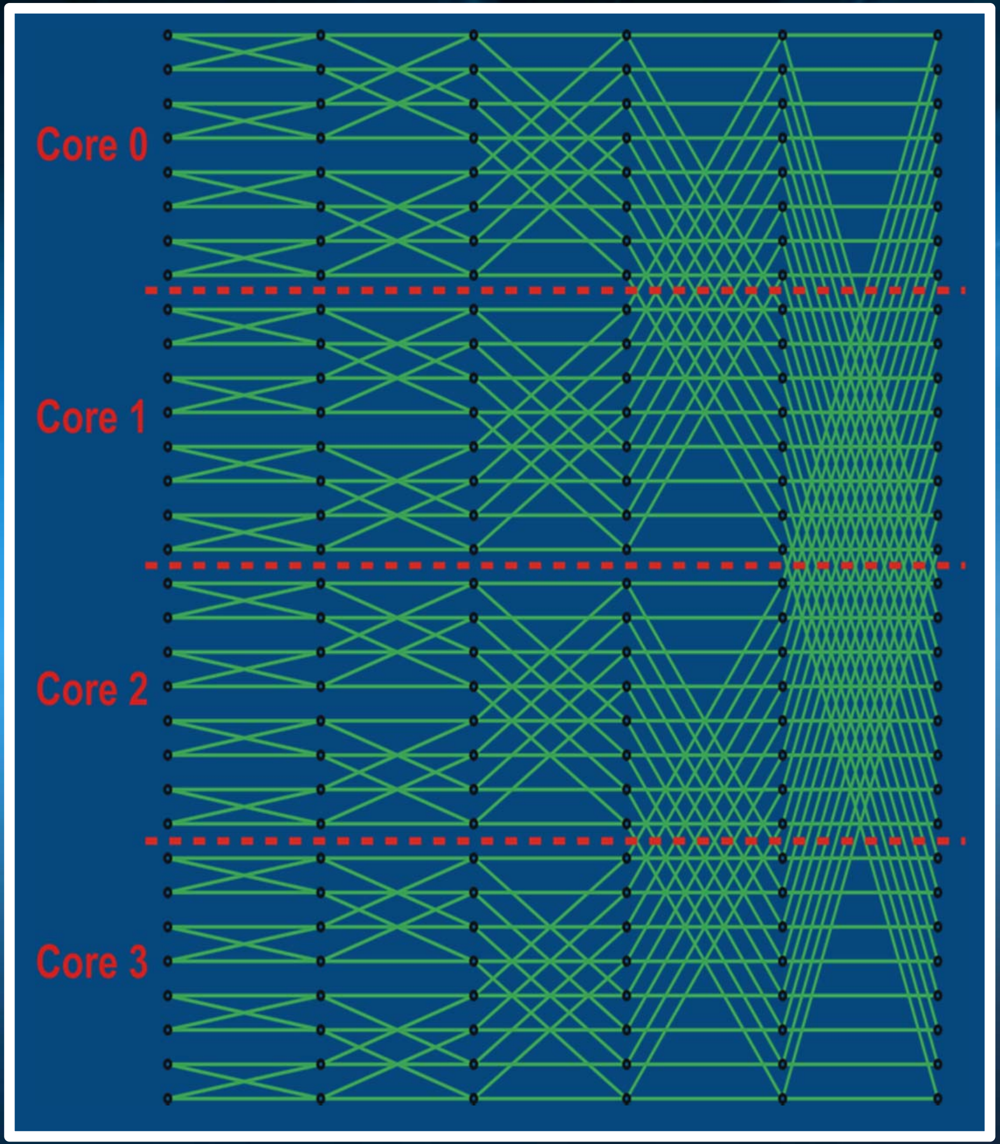
- The TOP500 ranking is based on a measure that is now no longer a pain point (64-bit multiply-add). *We need a ranking that aims at a pain point*.
- We're stuck in "LINPACK jail" because we hate to give up that wonderful database going back to 1992. Even though we know it's misleading.



But what can we use instead?

A Modest Proposal: How About Using… the FFT?

- Exercises every communication level in a system
- No way to cheat; all-to-all exchange *must* occur
- Data goes back to 1960s, even further than LINPACK!
- Already tracked by HPC Challenge, NAS Parallel Benchmarks
- Crucial for chemistry, CFD, image processing PDEs, oil exploration, DoD apps…



The Need for Benchmarks that Measure Productivity

- Current HPC benchmarks are grossly inadequate for guiding vendors to make the right choices.
- The myth persists that if a system has higher LINPACK, it means higher FLOPS on my app, which “typically gets *x*% of peak.”
- Instead of measuring machine activity, we should define and measure the quality of the answers we obtain.

Desirable Benchmark Attributes

- Should do something generally understandable by non-experts.
- The inevitable “single number” must be made as informative as possible.
- Should measure system, not cleverness of person running benchmark.
- Person who submits data is always identified.
- Published benchmarks are always dated.
- Inability to cheat by shifting computing work before the timing starts.

Purpose-Based vs. Activity-Based Benchmarks

An activity-based benchmark states computer operations to be performed, often defined by a source code or algorithm.

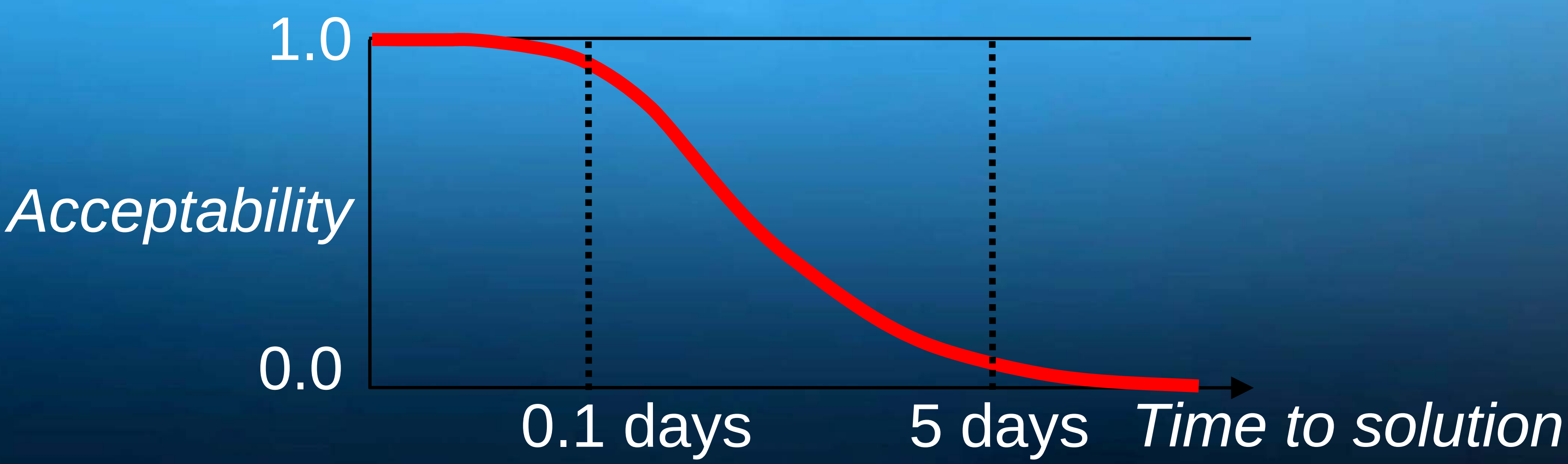
A purpose-based benchmark (PBB) states an objective function that is of *direct interest to humans*.

What a PBB Consists Of:

- Human-language description of purpose and constraints.
- Mathematics to show what needs to be calculated, but not *how* to calculate it. (Like “solve this system of equations” or “perform a Fourier transform”).
- “Acceptability functions” definitions.
- For those not testing programming productivity, reference implementations provided as starting points

Acceptability Functions

- Acceptability varies from 0 (unacceptable by everyone) to 1 (acceptable by everyone) as a given metric scales.
- Different for each market. Slight differences per user.
- Rigorous definition: The probability that a customer will accept a solution having that measured property.



Acceptability Curves Can Apply To…

Reliability (fraction of runs that complete)
Availability (fraction of time system can solve problem)
Accuracy (relative or absolute)
Time to boot system from a cold start
Time to load system with application
Time to execute application
Acquisition cost of system
Operational cost per hour of system, including admin. costs
Space required (footprint or volume)
Power required and heat dissipated

Warning: Paradigm Shift Ahead

For HPC benchmarks, replace "speed" with Acceptability as the measure of what customers want to know about a system.

Benchmarkers can choose their run times, guided by the A_t function. This eliminates "Amdahl's law" misguidance.

Net Acceptability is the product of all the *A* functions. (Assumes independence of variables)

Example: Mechanical Engineering

Generic problems in MCAE:

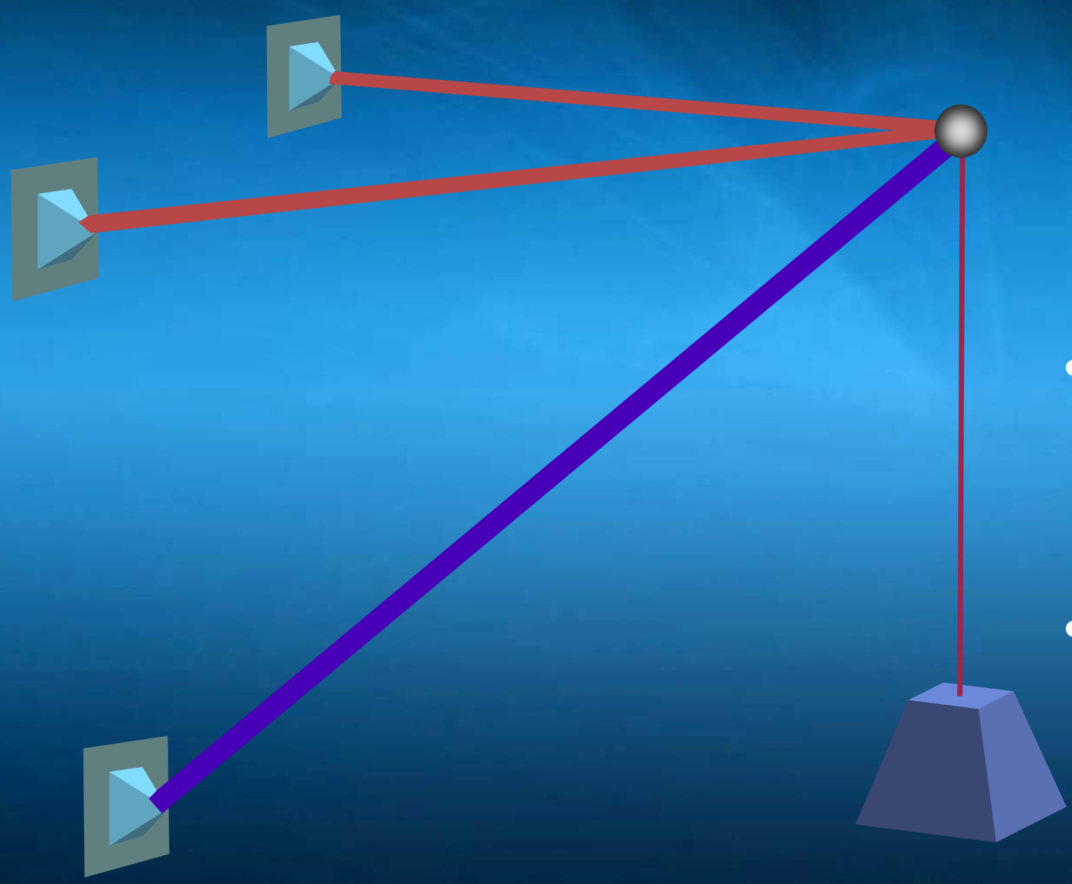
- Find stresses and strains of design to see if it meets requirements
- Optimize design for a particular feature, like low weight or low cost or high efficiency
- Find out if design has resonant modes that will cause failure (buildings, bridges, engines, etc.)

Answer validity EXTREMELY important

Time scales of hours to days tolerated

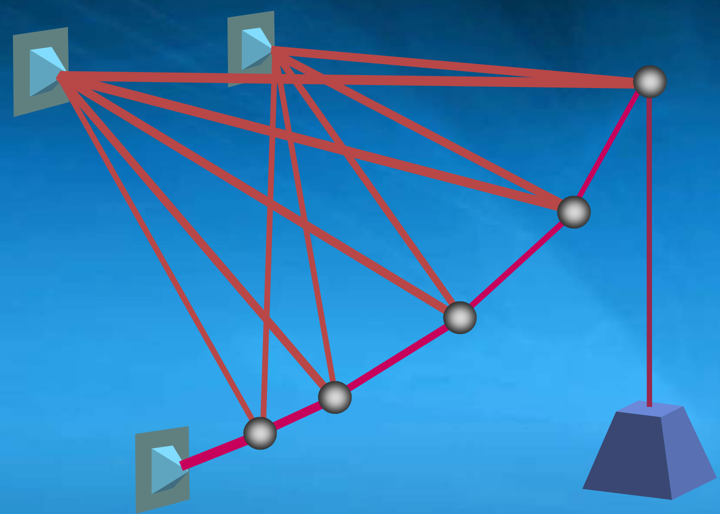
Cost of run comparable to savings in design + reduction of litigation risk

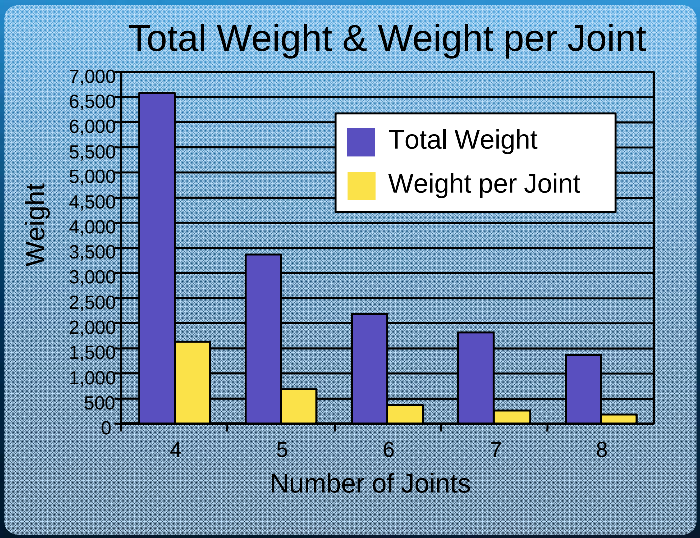
Truss Benchmark Definition



- Given a wall with 3 attachment points and a load at a distance from the wall, *find the pin-connected structure that uses the least steel* to bear the load.
- Structure must support its own weight; each joint adds steel.
- The weight for no extra joints (shown) is w_0; the purpose is to reduce the weight w_0 as much as possible, so the figure of merit is $(w_0 – w) / w_0$.

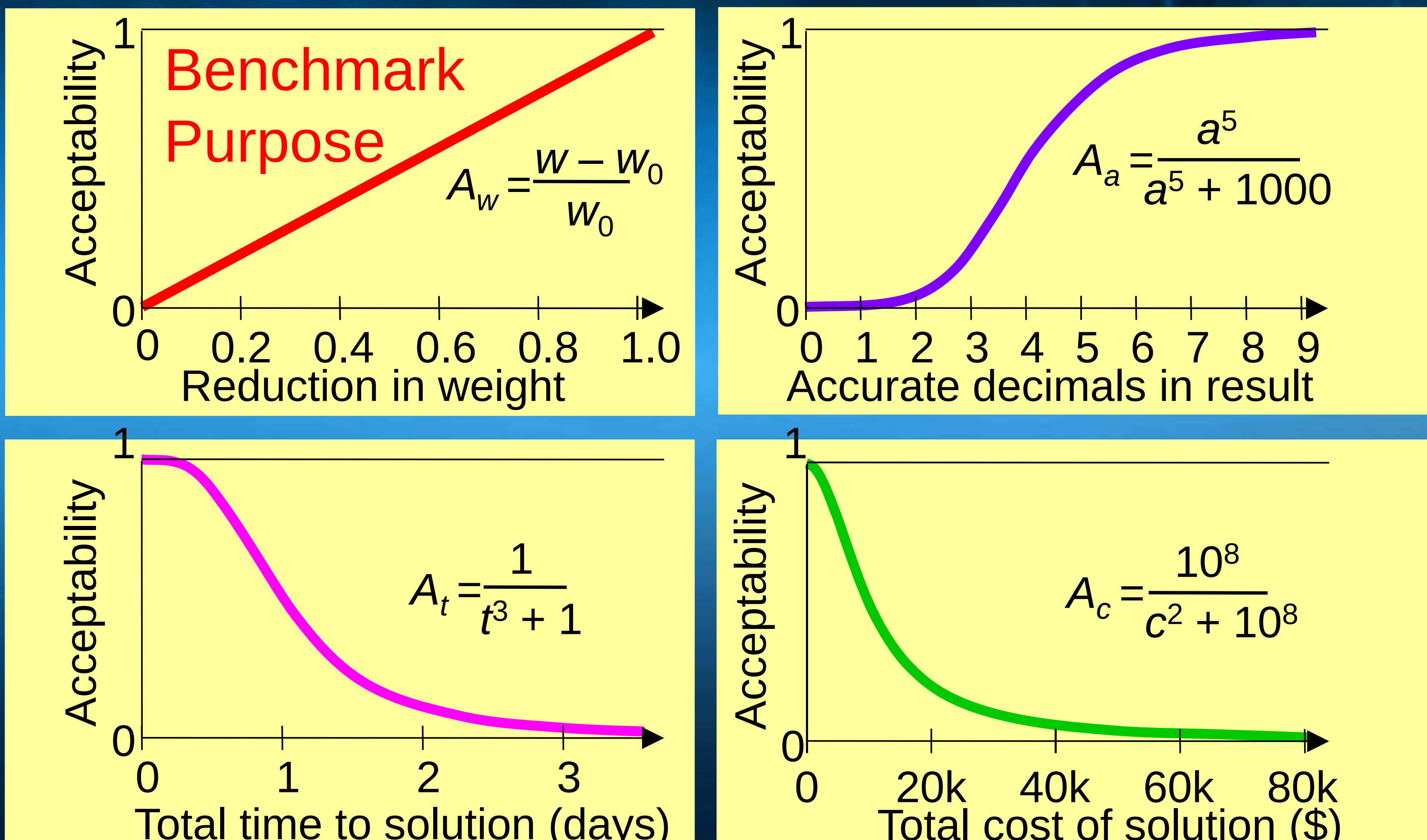
Truss Design





- More complex structures have higher strength and thus lower weight...
- ...until weight of joints exceeds the savings.
- Benchmark requires exploration of large number of topologies and weight as a function of joint position, to determine the optimum
- Imitates actions of a human engineer exploring a design space

Productivity is multidimensional, but reducible to a single number



HPC Benchmarks - I

HPC Segment	Task	Objectives
Electronic Design	Design and test an *N*-bit adder in a given time	Maximize speed, minimize area
Nuclear Applications	Optimize radiation transfer	Minimize radiation variation
Mechanical Engineering	Design truss to bear given load	Minimize weight of truss
Crash Simulation	Minimize deformation of elastic body past crush zone	Minimize deformation

HPC Benchmarks - II

HPC Segment	Task	Objectives
Fluid Dynamics	For shape-constrained vehicle, minimize drag coefficient	Minimize air drag
Weather/climate modeling	Predict weather for *N* days	Minimize forecast error
Life Sciences	Fold chain of *N* peptides in given amount of time	Maximize length of peptide
Financial Modeling	Predict and act on actual stream of financial events	Maximize profit
Petroleum	Manage petroleum reservoir	Maximize total petroleum extracted

SUMMARY

Computer Science must make the painful switch from counting operations to measuring communication when assessing complexity. This will not be easy, because of tiered memory.

Perhaps the best thing is to assess not “speed” but *productivity*, using a fuzzy-logic product (AND function) of the acceptability criteria!