

Cost

Caches, performance, energy. Landauer's floor, and Goodhart's law for everything you measure.

THREE CURRENCIES

Time, space, energy. Every design in this class was a trade among them.

TIME

Instructions, as the profiler counts them. Emulation $\times 2,593$, virtualization $\times 208$, a context switch, a page fault, a system call: each a number selfie prints.

Optimisations preserve meaning and change this.

SPACE

Bytes mapped. A flat page table at 8 MB against a tree; a bump allocator that never reuses against a free list; a collector that keeps dead memory because it is reachable. Each a bound with a footprint.

ENERGY

Joules from the wall. The one currency with a law of physics attached, and the one a data centre actually pays. Comes at the end.

Decidable is not doable: the halting problem is not what stops a kernel from scheduling optimally or a collector from being precise; the budget is. Station IV, for a systems engineer, is the budget.

Memory is slow. A cache bets on **locality**, and the bet usually pays.

```
$ ./selfie -c selfie.c -L1 1
./selfie: summary: 85754 executed instructions in total
./selfie: L1 caches: accesses,hits,misses
./selfie: data: 34517,34283(99.32%),234(0.67%)
./selfie: instruction: 85755,85225(99.38%),530(0.61%)
```

Lines, associativity, eviction, write policy: the chapter's parameters. Change them with selfie's options and watch the miss rate; that is the exercise.

A processor cycle is a nanosecond; a memory access is a hundred. A cache is a small fast memory that keeps what was used recently, on the bet that it will be used again. Selfie simulates L1 instruction and data caches and reports the hit rate: over 99 percent for selfie compiling itself, which is why the bet is the whole architecture.

Same meaning, different cost: the cache is invisible to the program's semantics, like the page table's data structure. Which is why it can be tuned freely, and why it leaks: timing differs, and a program that measures its own timing can read what another program cached. Isolation of meaning is not isolation of cost.

The profiler is the instrument. What it does not count did not happen.

```
./selfie: summary: 85754 executed instructions in total  
./selfie: 14 syscalls, 1 page faults, 0 timer interrupt  
./selfie: memory: ld: 21307(24.84%), sd: 13210(15.40%)  
./selfie: calls: ... loops: ... loads: ... stores: ...
```

Instruction mix, nops, system calls, page faults, timer interrupts, the hottest procedures and loops: selfie's profile of every run. Every number a design decision of this semester changed. Look at the nop share: the compiler class's optimisation week is about that line.

Measure before you optimise, and measure what the user pays, not what is convenient to count. A kernel that minimises context switches and maximises latency has optimised the wrong line. Which is the next slide.

A metric decays the moment it becomes a **target**.

Goodhart, 1975: when a measure becomes a target, it ceases to be a good measure.

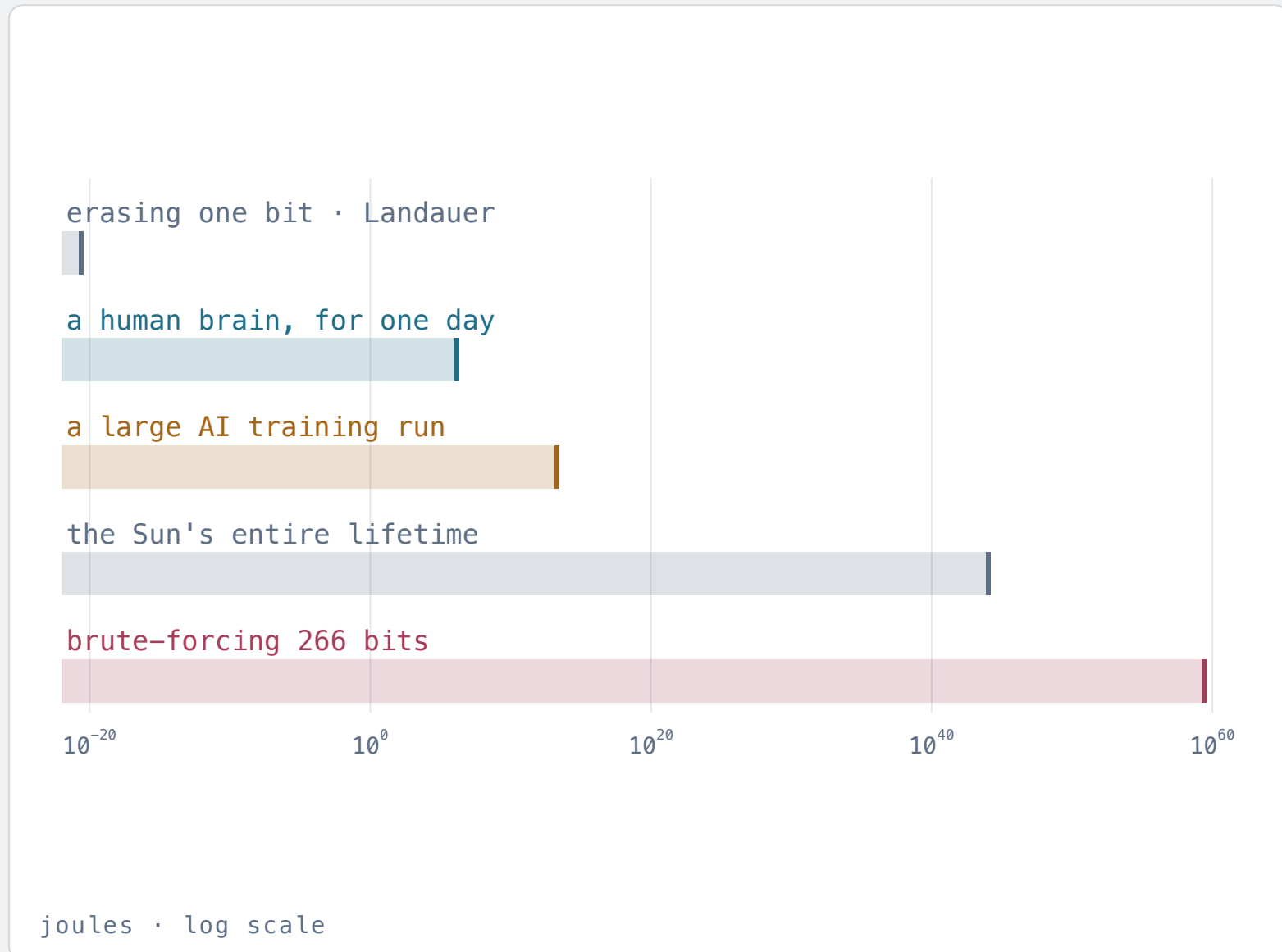
Benchmarks that compilers detect. Schedulers tuned for a throughput number until the interactive user leaves. Energy accounting that moves the joules to a different meter. Training to the benchmark.

A mature field needs notation, semantics, and a metric, and it needs to keep replacing the metric. The systems engineer's version: the profile is where the argument starts, not where it ends.

When a measure becomes a target, it ceases to be a good measure.

CHARLES GOODHART, 1975

Every step that forgets **costs energy**. Every context switch forgets.



Landauer, 1961: erasing one bit costs at least $kT \ln 2 \approx 3 \times 10^{-21}$ joules, because two states become one and the lost distinction leaves as heat. Every overwrite is an erasure: every store, every register restore, every page copied and dropped.

A brute-force search over the introduction class's 266-bit space just counting would cost 10^{59} joules, 10^{15} Suns. Real hardware sits orders of magnitude above the floor, and the floor does not move for any device that overwrites. A data centre is a machine for forgetting at scale.

A human brain runs on 20 watts. Whatever intelligence is, it can be done far more cheaply than it is currently done, and a systems engineer is the person whose job that is.

Exercises.

- 1 Profile your kernel with four processes and a small timeout. Which lines of the profile did processes change, and which did the timeout change?
- 2 Vary selfie's cache parameters and plot the miss rate for selfie compiling itself. Find the smallest cache that keeps it above 99 percent.
- 3 Estimate, from the profile and Landauer, a lower bound on the energy of one context switch in your kernel. Compare with what a real processor spends.

NEXT WEEK

Universality for systems: the halting problem in the scheduler, Rice on everything the kernel wants to know, and deadlock detection as the approximation it uses instead.