Performance

time is money

lots of data ⇒ process ⇒ info insights

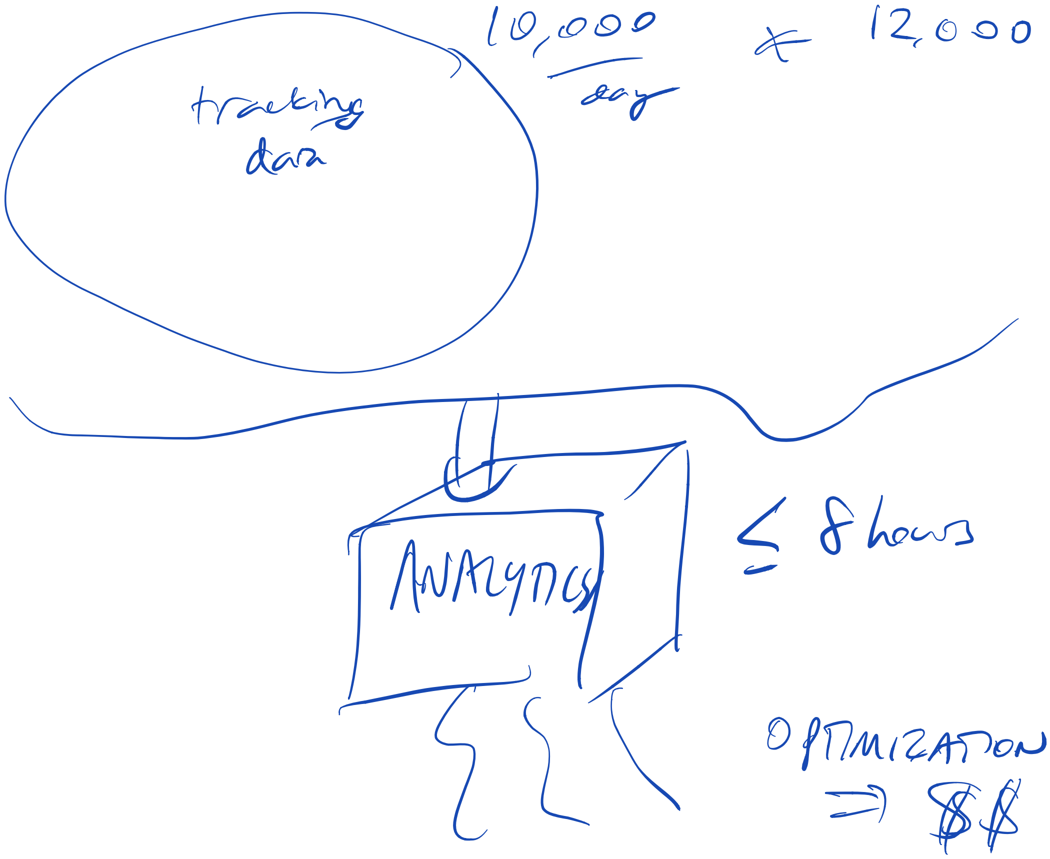
limited time budget – 1 day, 1 week

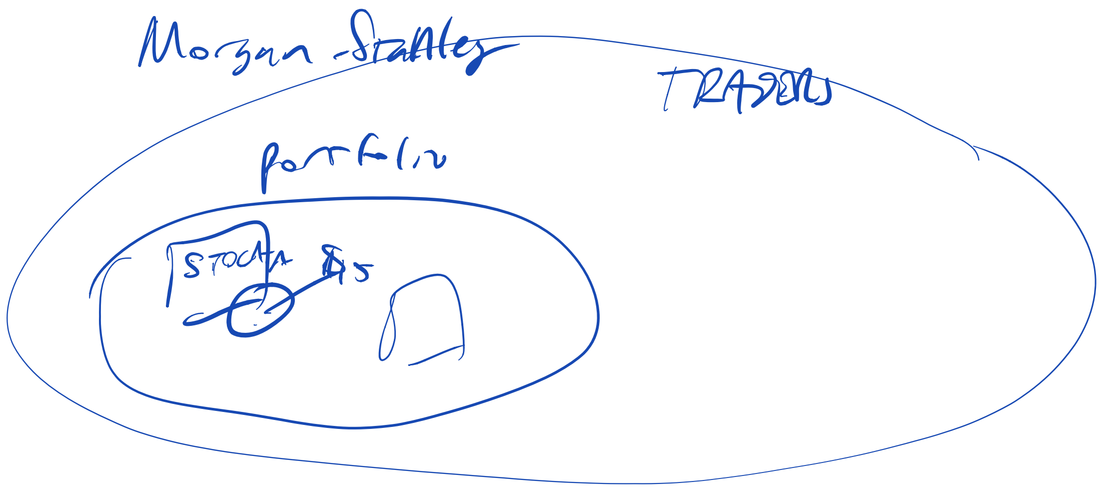
MORE data

⇒ better info

Wal-Mart – every single sale × # of WalMarts

10,000 / day × 12,000





risk models (overnight)

time = money

---

performance factors

h/w — # CPUs fast/slow — cache coherence AMD vs. Intel vs ARM

COMPUTE

GPUs

(esoteric) accelerators

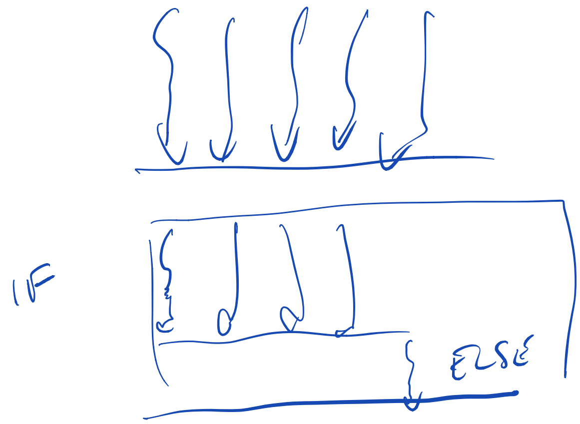
FPGAs (Azure) } ML

TPUs (Google) }

massively ||

thousands of threads

GPGPU — general-purpose GPU prog.



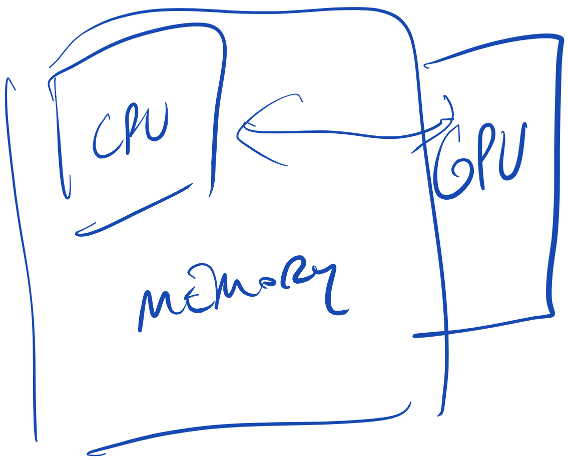
divergence

ASICs

application specific IC

CUDA Nvidia

OpenCL

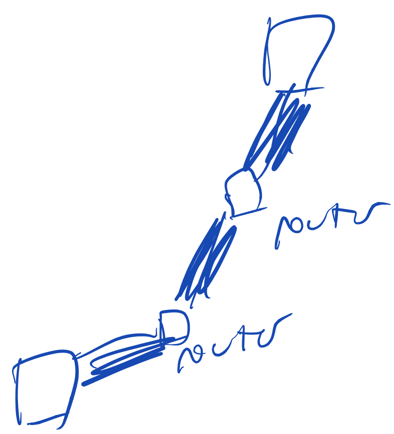


COMPUTE

— NETWORK

topology

(hops)



— latency

— throughput high bandwidth

InfiniBand

— DISK

- HDD latency seeks
- SSD capacity bandwidth

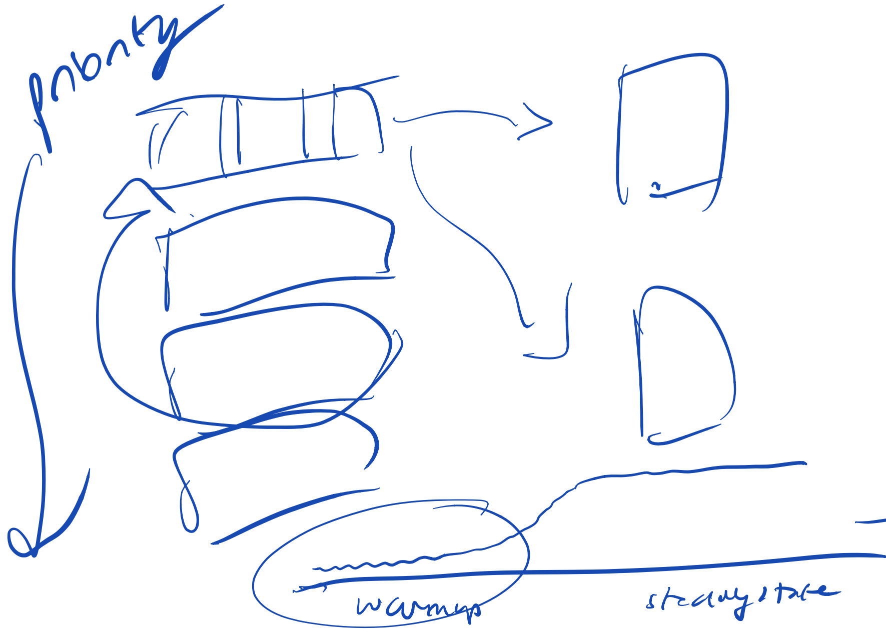
RAID increase b/w

- MEMORY
  - speeds bus ~~b/w~~ freq.
    - ≈ b/w
  - capacity — MORE = BETTER
  - cache (SRAM) bigger

lowend small cores 2.4 GHz big cores ~ 4 GHz

SW → perf.
- algorithms — $O(n) \to O(1)$
- PL — GC ~ "managed" Java / JavaScript
  - JIT compilation — warmup period

- OS — File systems
  - Networking stack
  - Scalability of OS
  - optimized OS
  - Scheduler



quantum
~ 1-5ms

design choices

optimize for throughput
OR latency

(disk I/O) interactive

batching

MT node.js 1 thread concurrent I/O

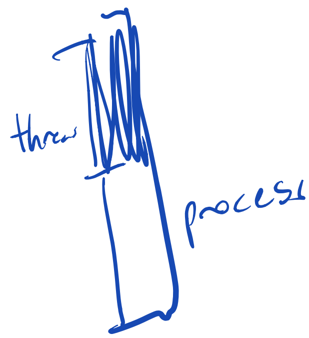
MP
node.js
node.js
⋮

— fault tolerance

MP good ⌊ if no "shared state"

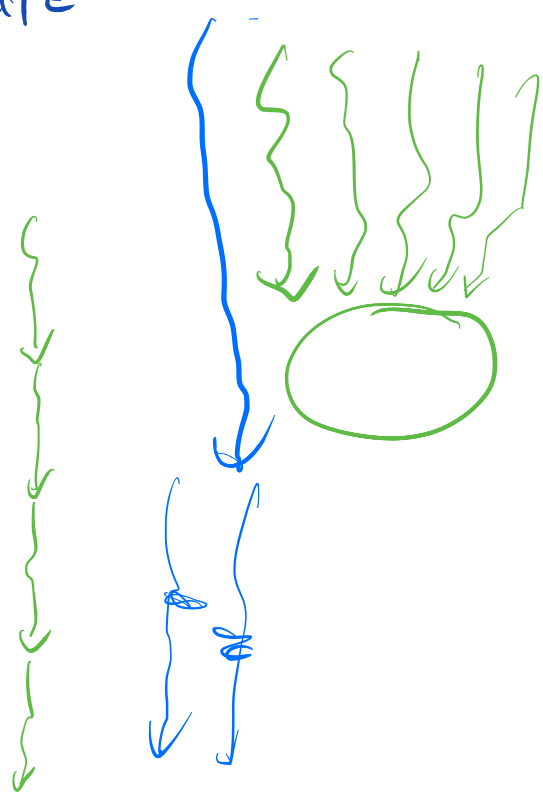
MT good — (lot) of shared state

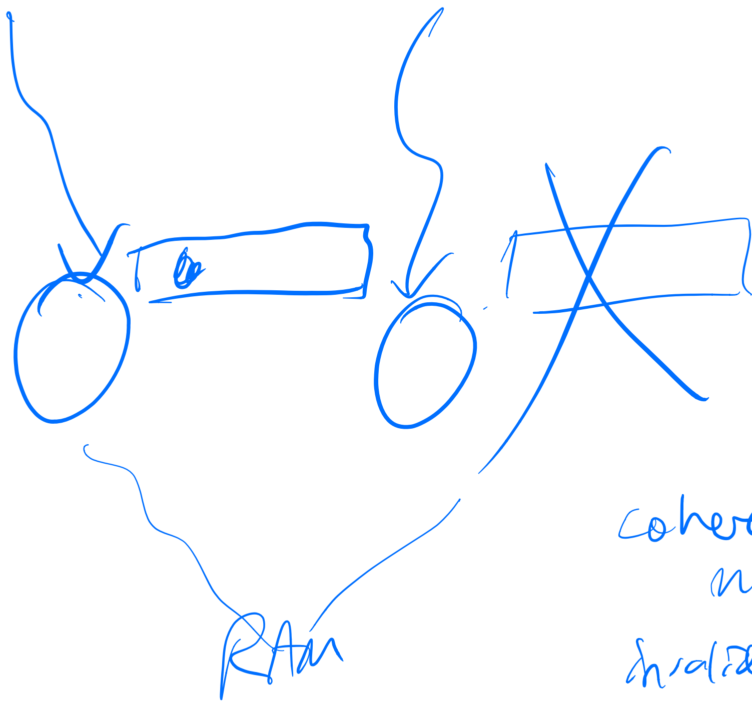
MT { Apache — a patchy server
nginx



MT
painful
(correctness)

CACHE

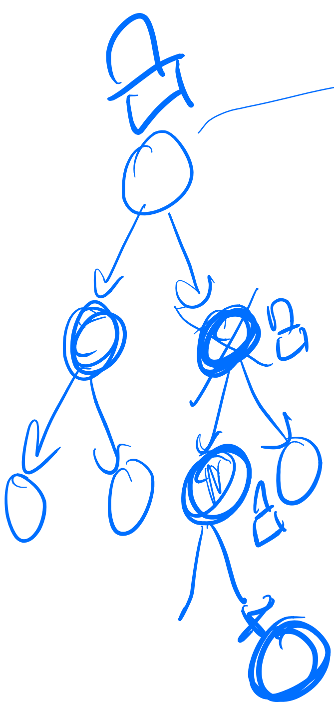




coherence

MESI

invalidation



hand-over-hand locking

CONTENTION

abstractions

APIs

perf. ultimate leaky abstraction

→ workload - dependent