SYCL Academy

SYCL™

Think Parallel

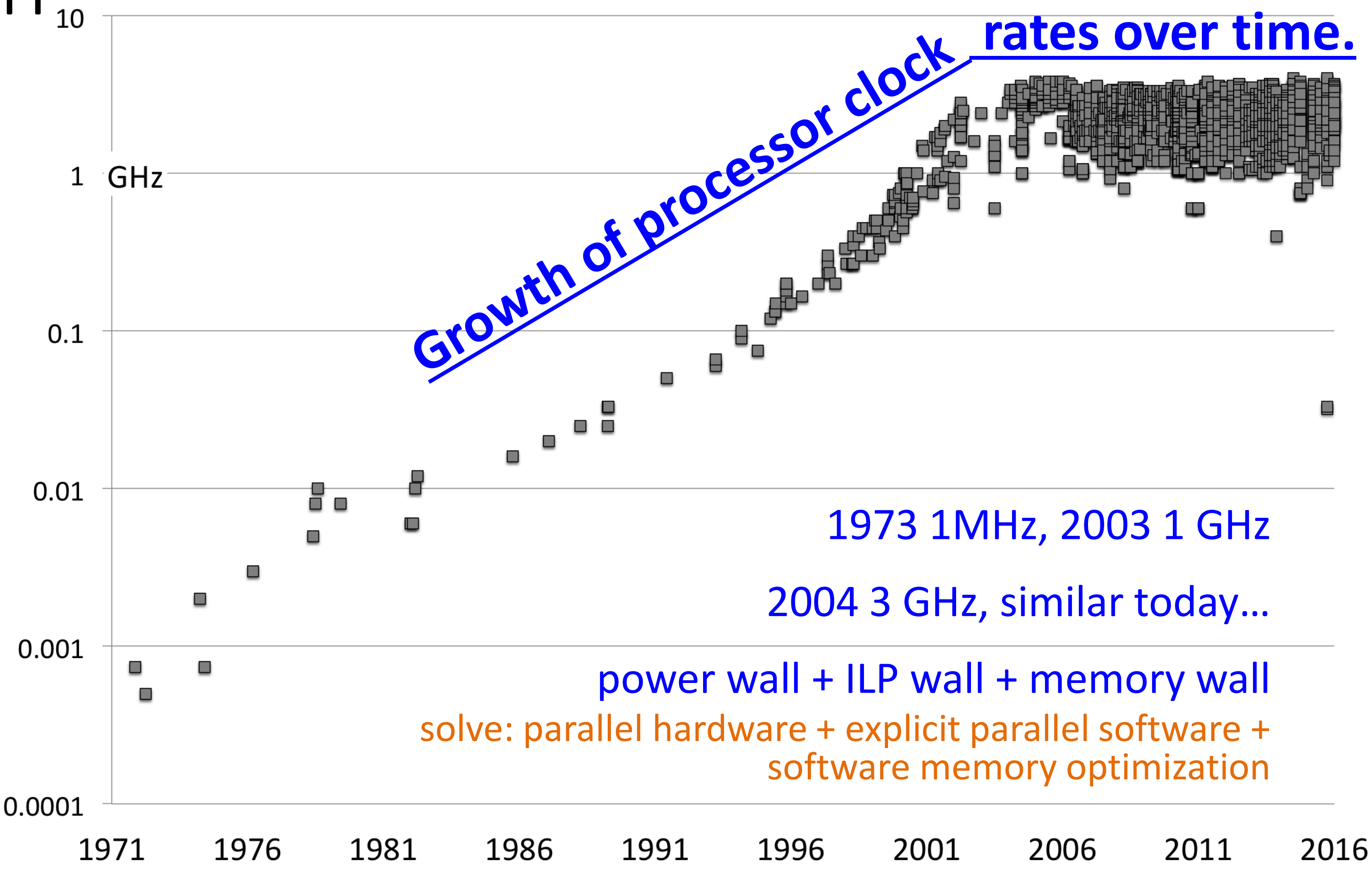
the Fundamentals of Parallel Programming

James Reinders (Intel)

Think Parallel

- Be motivated!
- Concurrency vs. Parallelism
- Our programming job: provide maximal *opportunity* for parallelism
  - Opportunity is different than required
- Parallelism is where the data is
  - Amdahl’s Law creates unnecessary pessimism
  - Gustafson encouraged use to rethink Amdahl’s Law
- Scaling is important
  - Optimal scalar is often different than ‘optimal ready to scale’

motivation



Growth of processor clock rates over time.

1973 1MHz, 2003 1 GHz

2004 3 GHz, similar today…

power wall + ILP wall + memory wall

solve: parallel hardware + explicit parallel software + software memory optimization

concurrency vs. parallelism

concurrency

composition of independently executing processes

dealing with lots of things at once

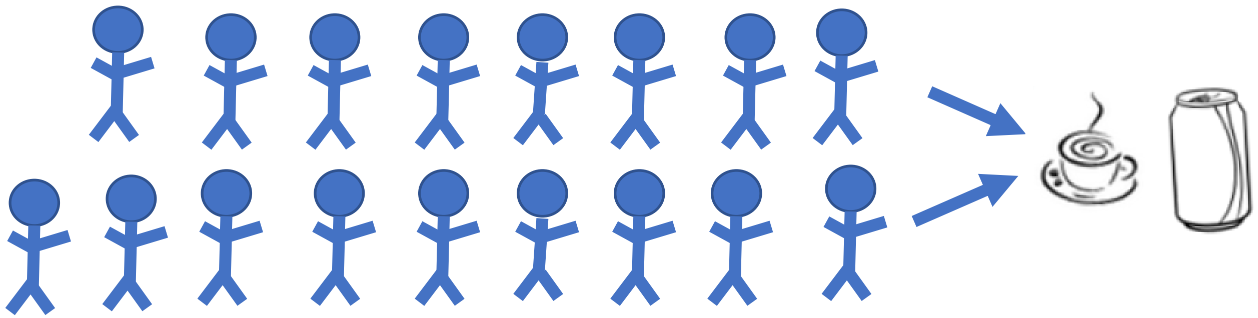
parallelism

simultaneous execution of (possibly related) computations

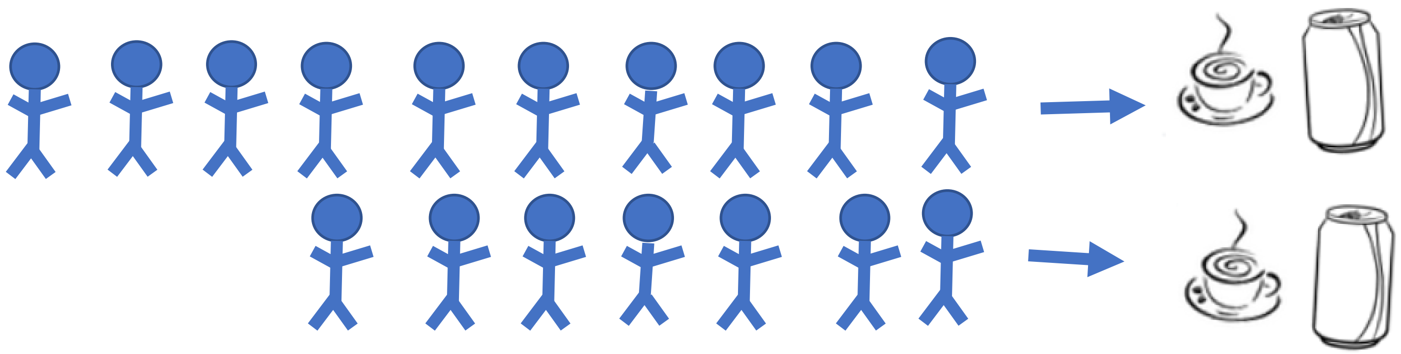
doing lots of things at once

concurrency vs. parallelism

Concurrent = two queues, one caffeine dispenser

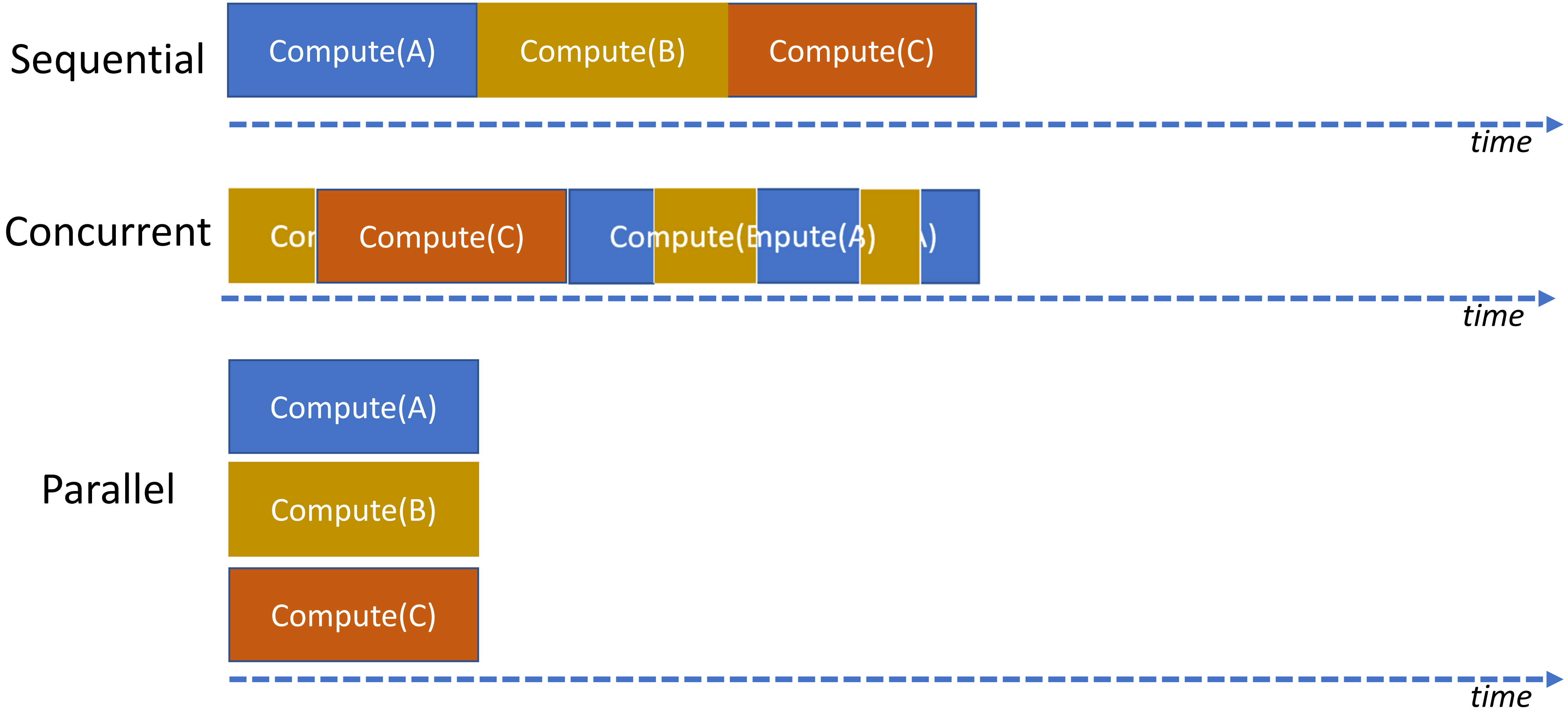


Parallel = two queues, two caffeine dispensers



concurrency vs. parallelism

Three independent computations.



Sequential

Compute(A)

Compute(B)

Compute(C)

time

Concurrent

Cor

Compute(C)

Compute(Enpute(A)

)

time

Parallel

Compute(A)

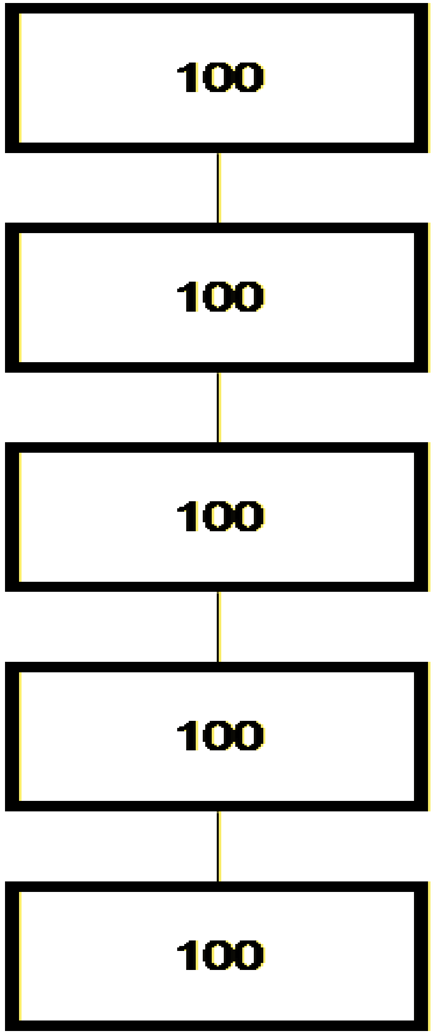
Compute(B)

Compute(C)

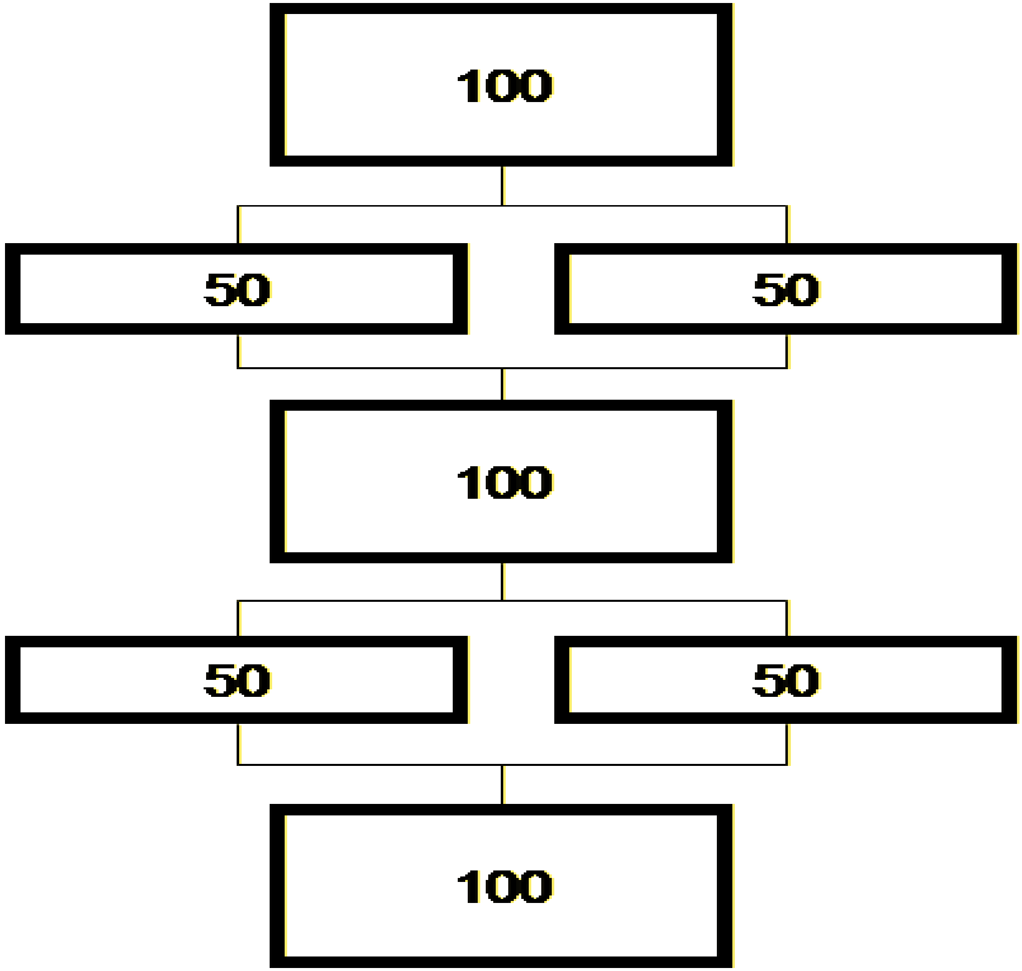
time

How much parallelism is there?

- Amdahl's Law
- Gustafson's observations on Amdahl's Law

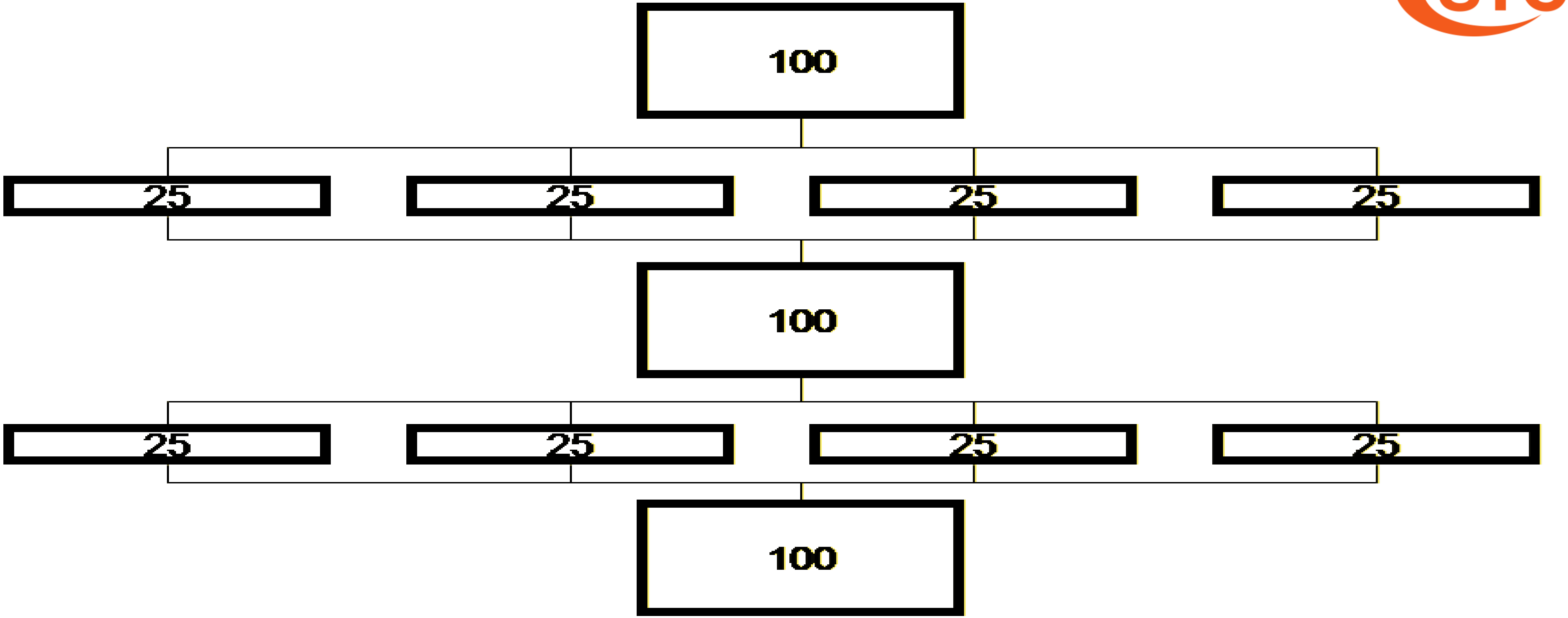


Work 500 Time 500
Speedup 1X



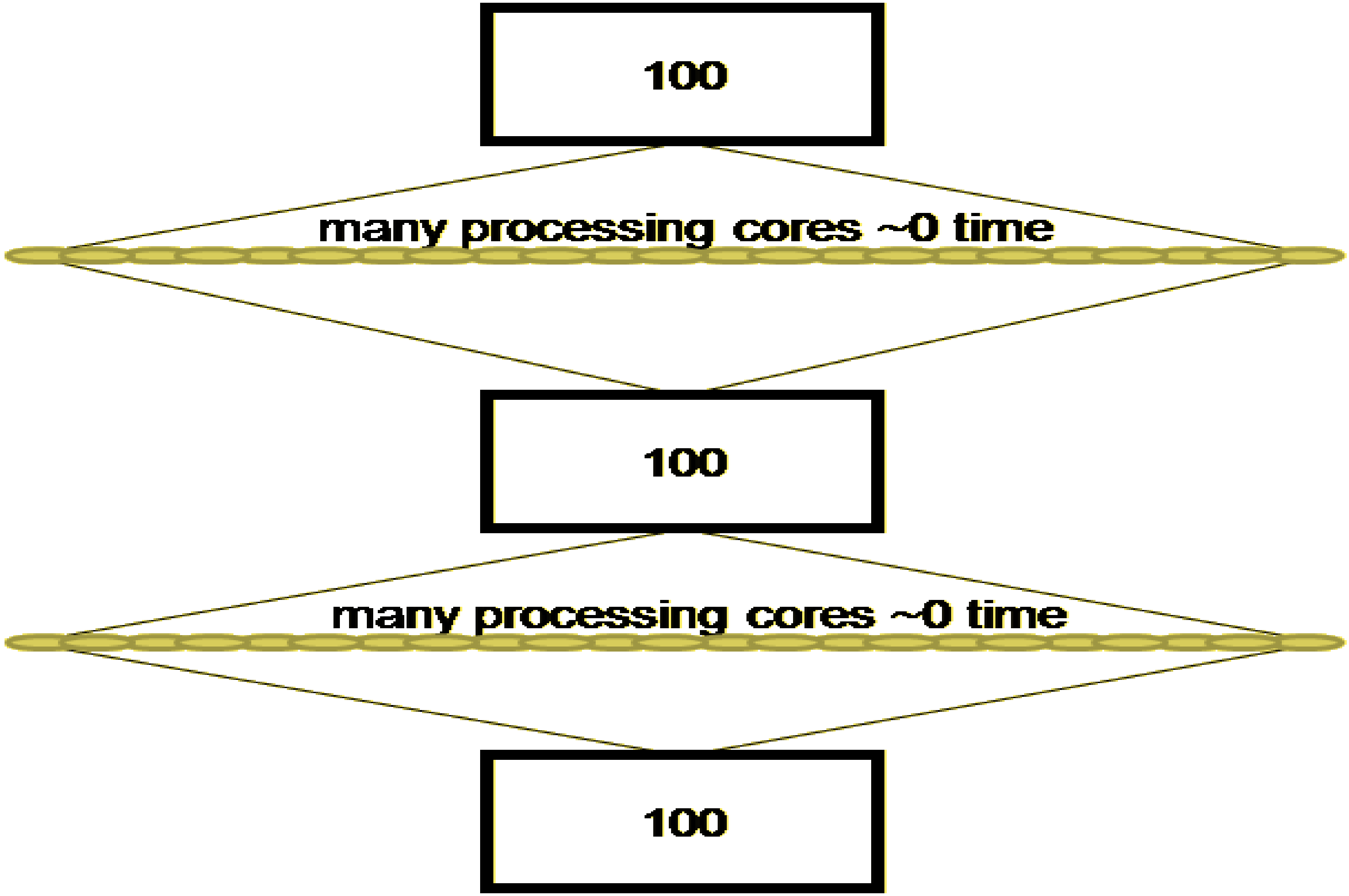
Work 500 Time 400

Speedup 1.25X

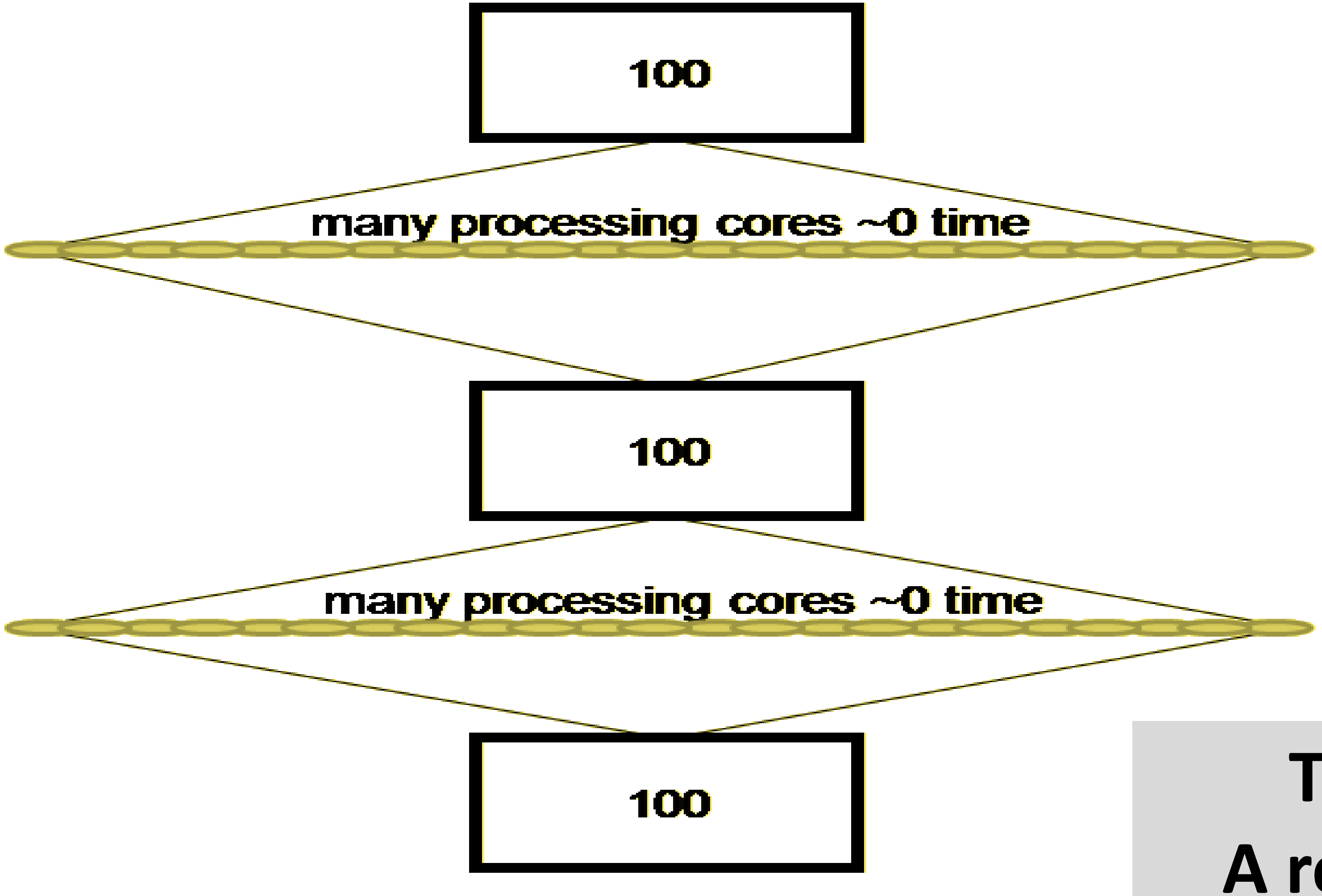


Work 500 Time 350

Speedup 1.4X



Work 500 Time 300
Speedup 1.7X



Work 500 Time 300
Speedup 1.7X

That's all?
A reduction of
only 40% percent
in time???

Amdahl's law

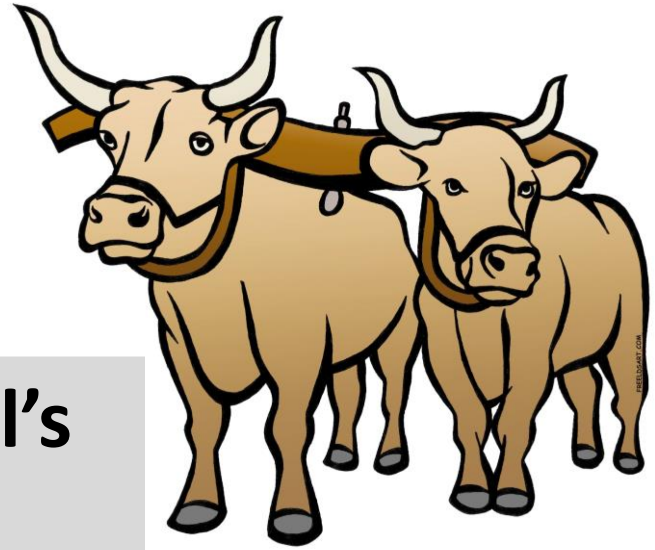
"…the effort expended on achieving high parallel processing rates is wasted unless it is accompanied by achievements in sequential processing rates of very nearly the same magnitude."

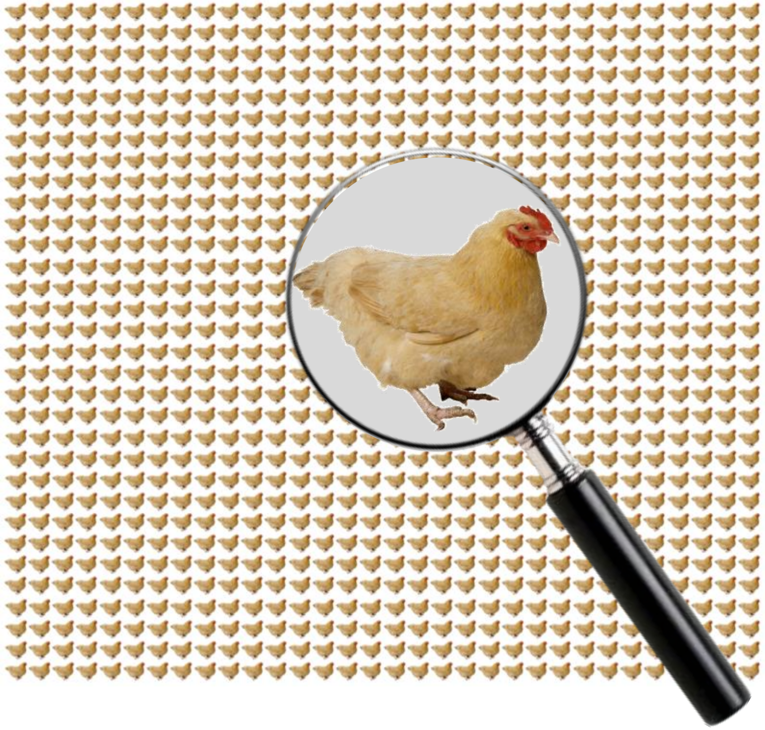
– Amdahl, 1967

If you were plowing a field, which would you rather use… two strong oxen, or 1024 chickens?

- Seymour Cray

Seymour had Amdahl’s Law on his mind. Too much? We’ll come back to that.





Amdahl's law

"…the effort expended on achieving high parallel processing rates is wasted unless it is accompanied by achievements in sequential processing rates of very nearly the same magnitude."

– Amdahl, 1967

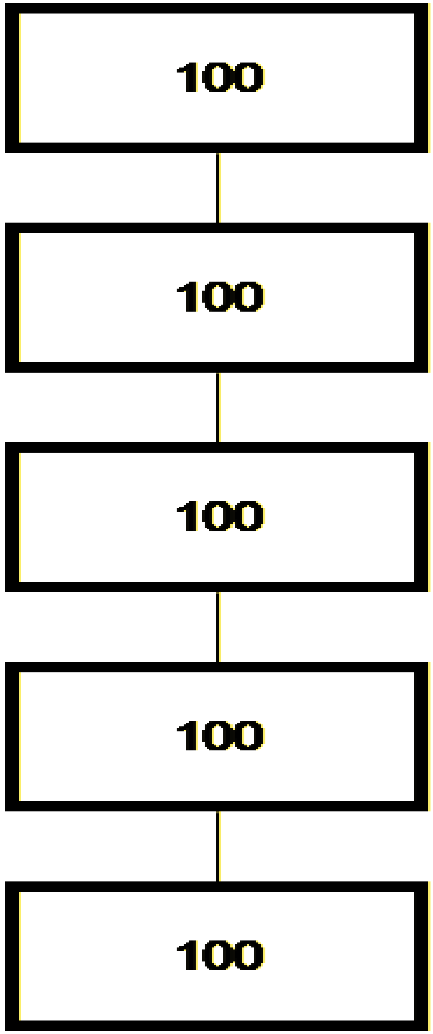
**Amdahl was not wrong,
but his law cast enormous doubt
on any hope for parallel computing.**

Rethinking Amdahl's law

"…speedup should be measured by scaling the problem to the number of processors, not by fixing the problem size."

– Gustafson, 1988

Undaunted, parallel computing enthusiasts continued – and John Gustafson offered an explanation why we might not be crazy.



100

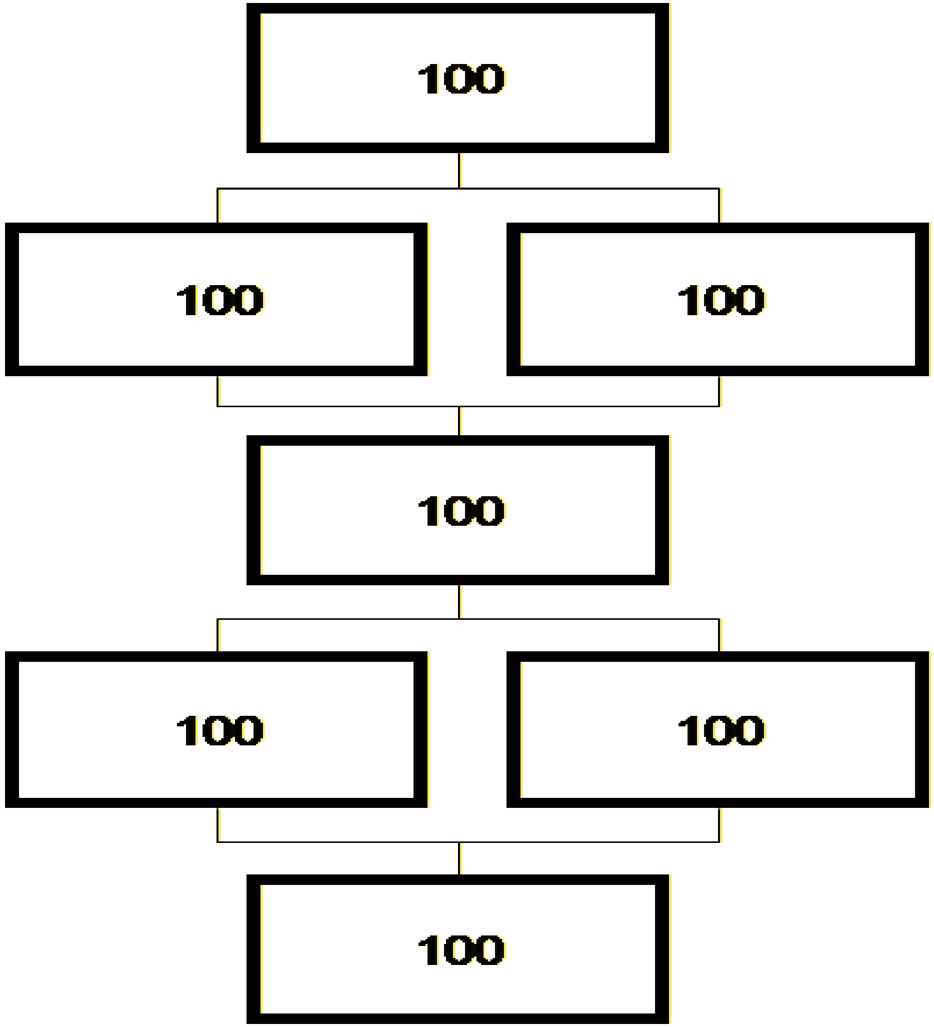
100

100

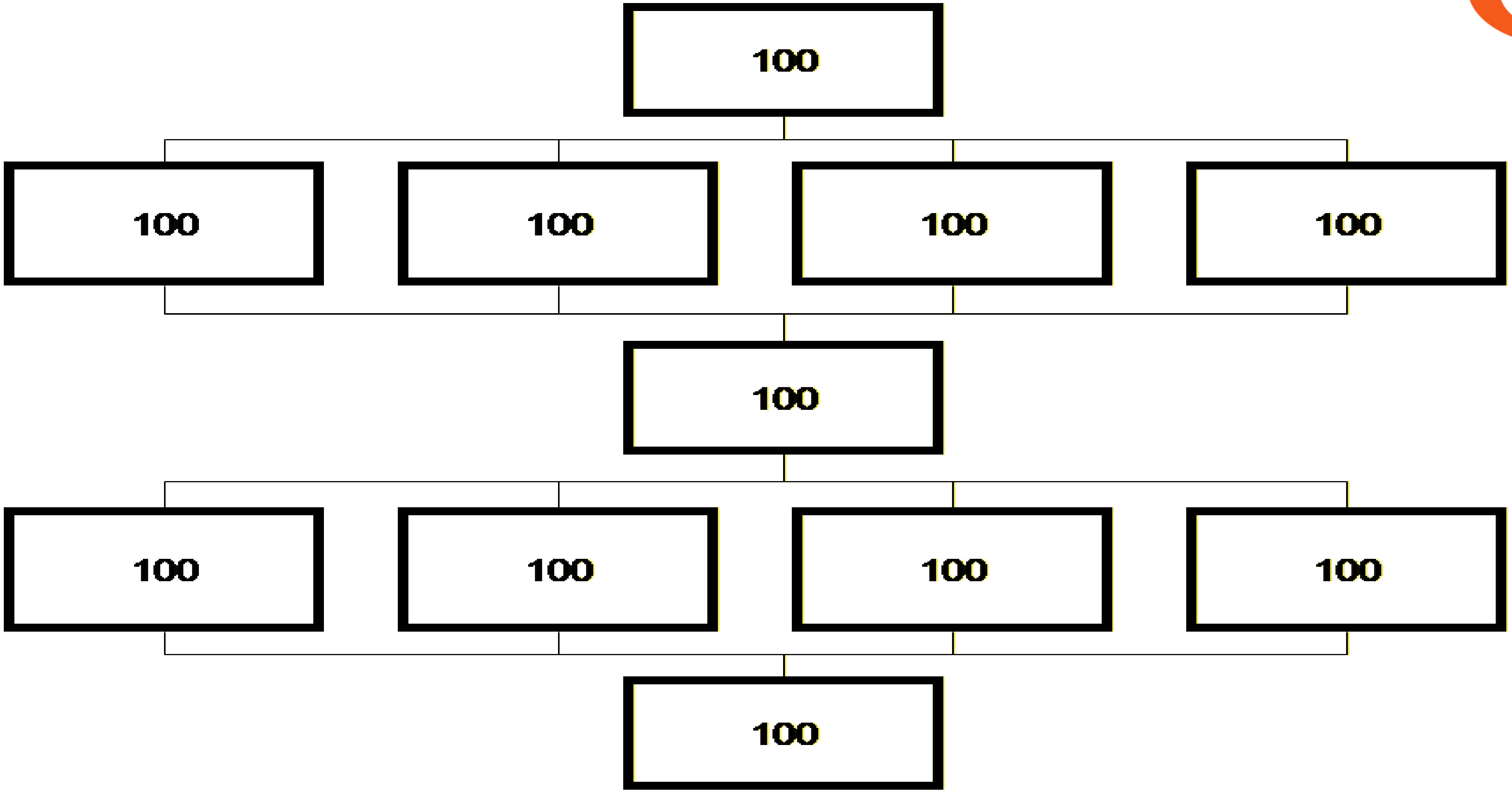
100

100

Work 500 Time 500
Speedup 1X

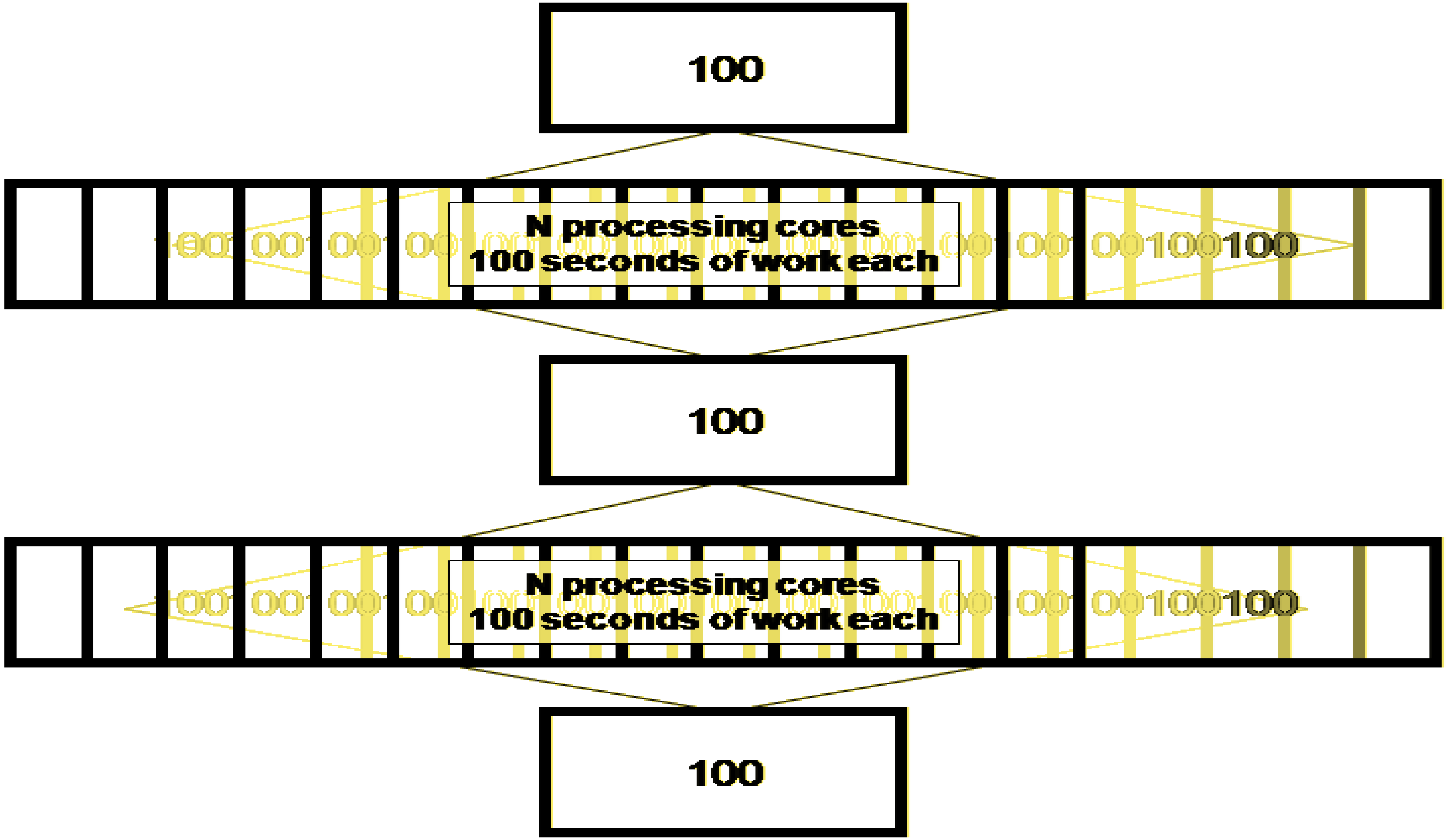


Work 700 Time 500
Speedup 1.4X



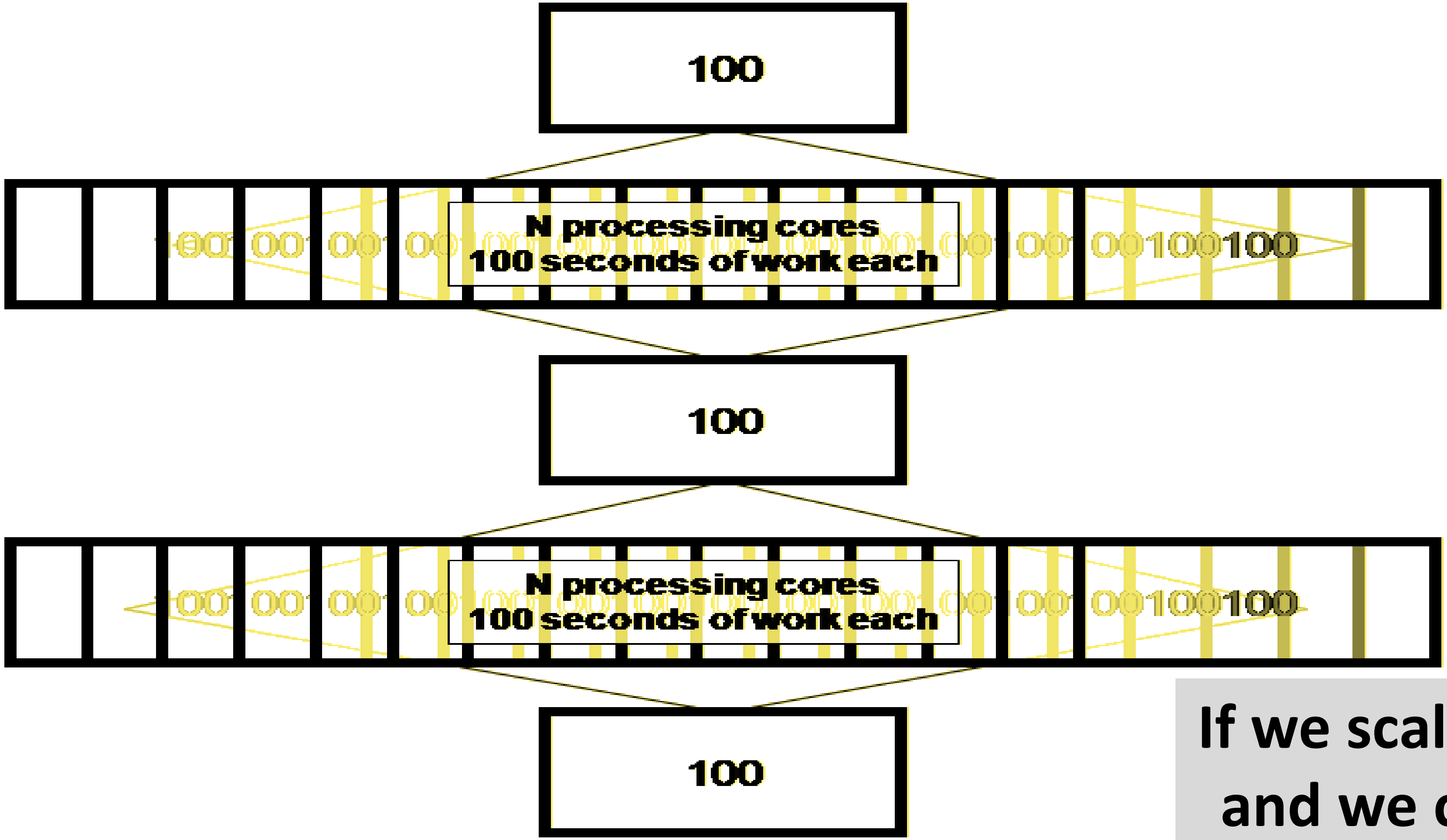
Work 1100 Time 500

Speedup 2.2X



Work 2*N*100+300 Time 500

Speedup O(N)



Work 2*N*100+300 Time 500

Speedup O(N)

If we scale with data, and we can process data in parallel, we have what we need to scale overall.

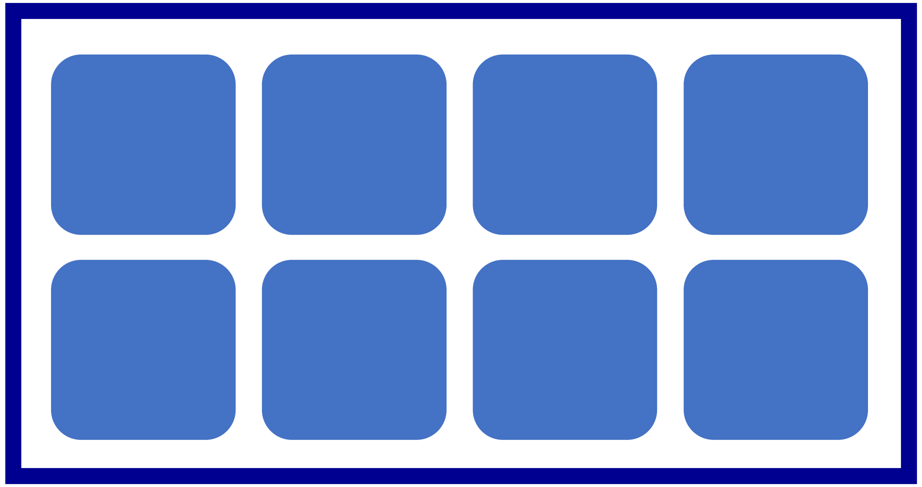
How much parallelism is there?

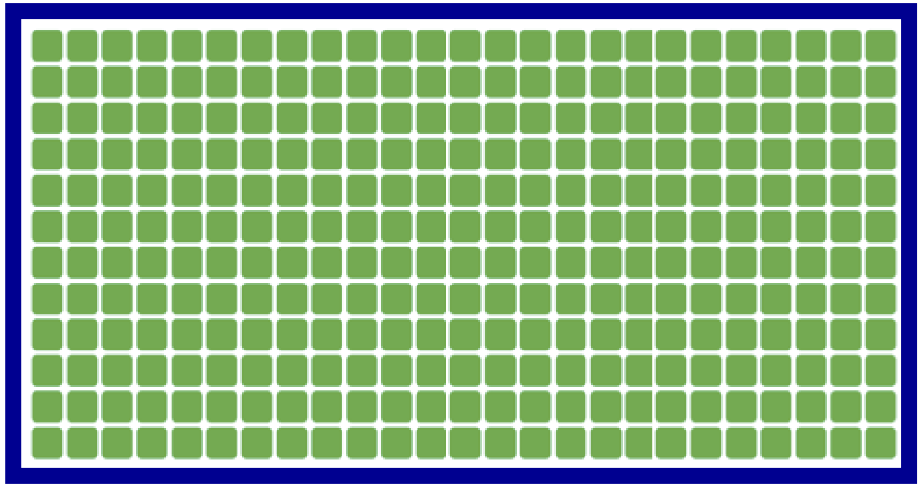
- Amdahl’s Law
- Gustafson’s observations on Amdahl’s Law

Plenty –

but the workloads need to continue to grow !

Design Question



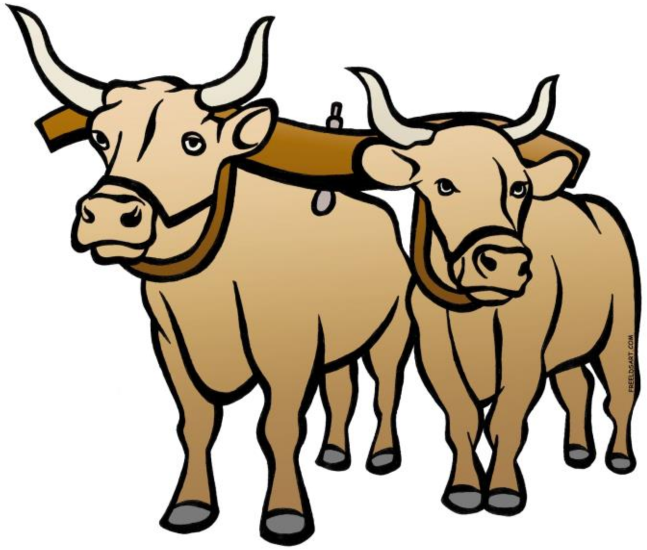


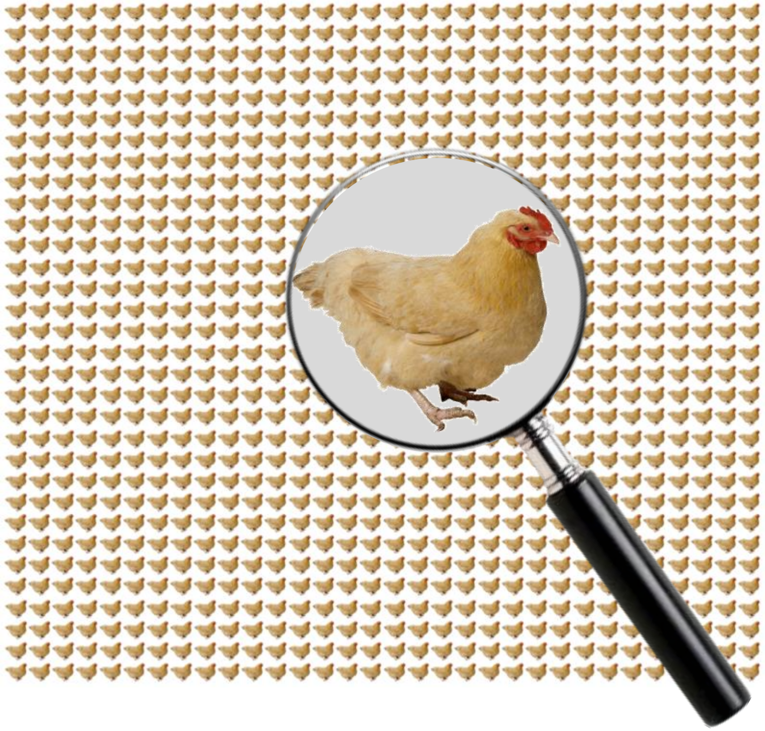
A few powerful vs. Many *much* less powerful *and very restrictive*.

Diagrams for discussion purposes only, not a precise representation of any product of any company.

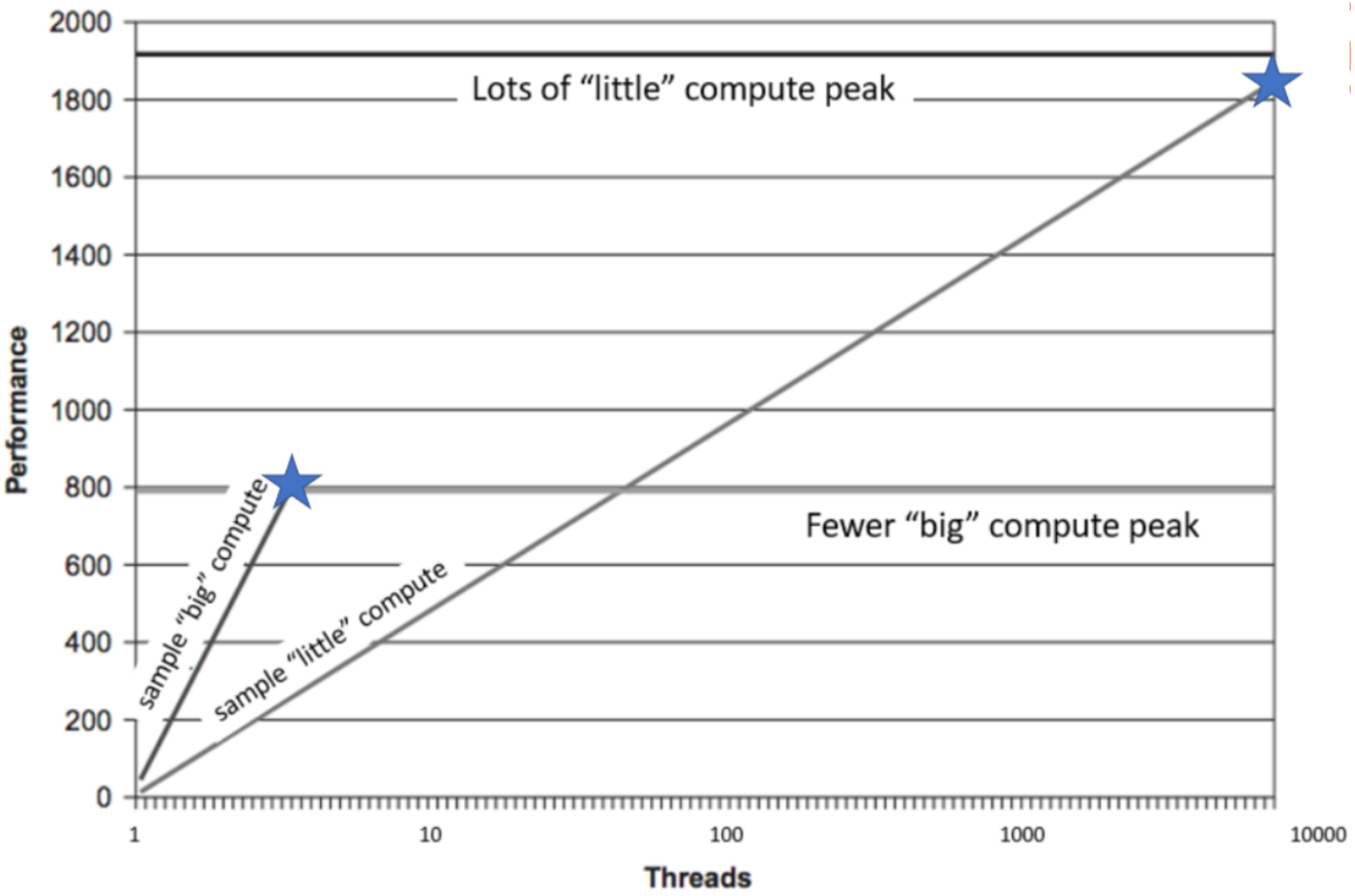
If you were plowing a field, which would you rather use… two strong oxen, or 1024 chickens?

- Seymour Cray



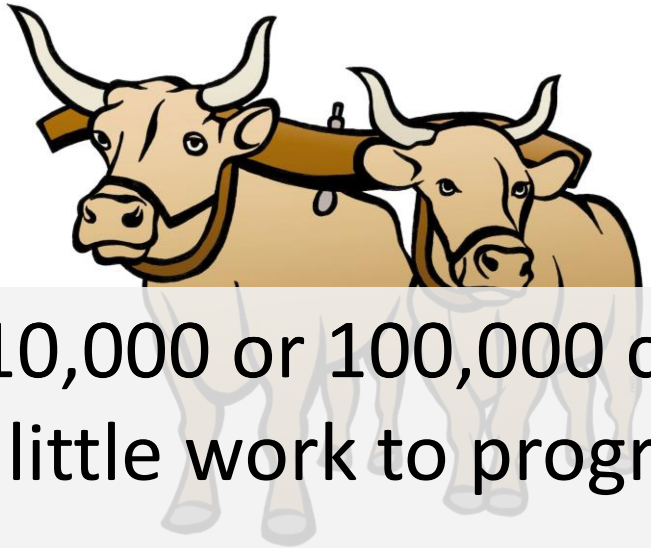


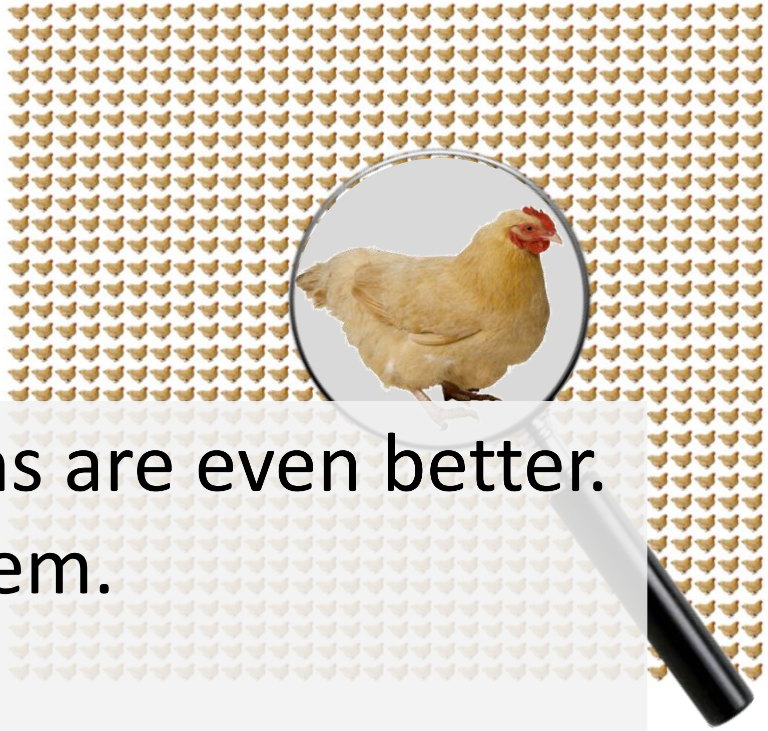
Scaling is key:
Lots of little computes can win the race



If you were plowing a field,
which would you rather use…
two strong oxen, or
1024 chickens?

- Seymour Cray





The answer: 10,000 or 100,000 chickens are even better.
It just takes a little work to program them.
Think SYCL !

Think Parallel

- Be motivated!
- Concurrency vs. Parallelism
- Our programming job: provide maximal *opportunity* for parallelism
  - Opportunity is different than required
- Parallelism is where the data is
  - Amdahl’s Law creates unnecessary pessimism
  - Gustafson encouraged use to rethink Amdahl’s Law
- Scaling is important
  - Optimal scalar is often different than ‘optimal ready to scale’