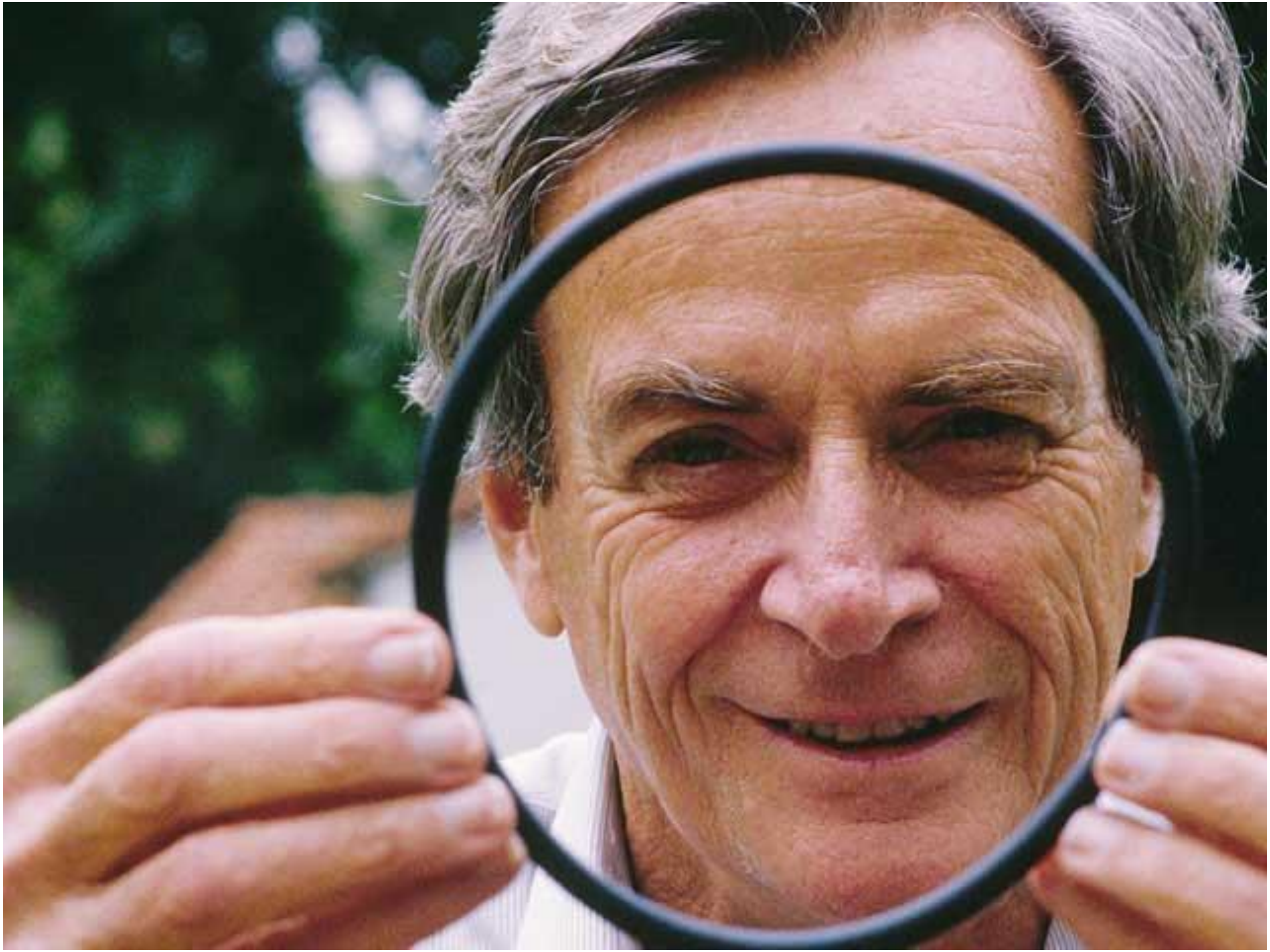
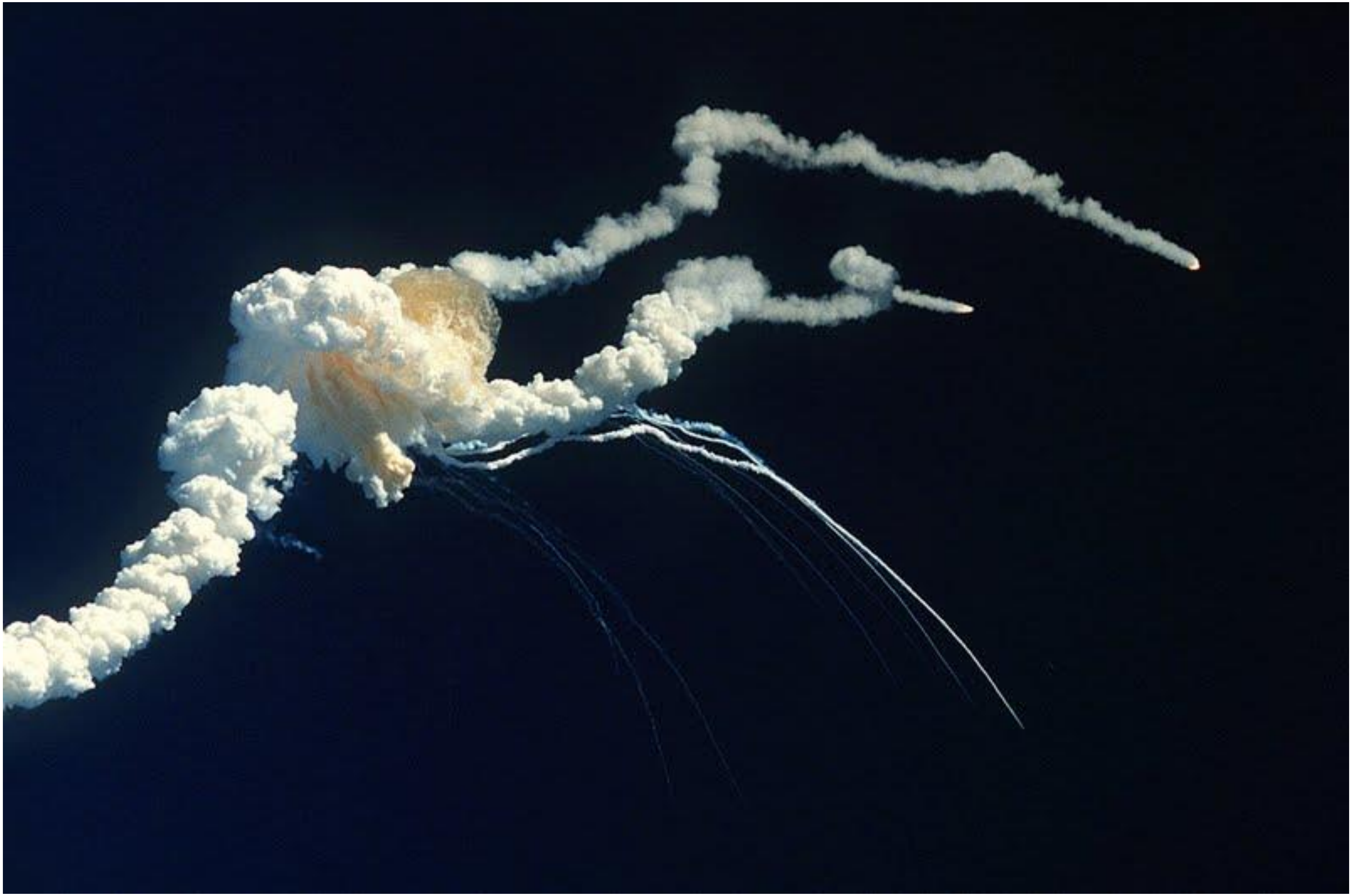
Real Logic

ACCELERATING SOFTWARE

Designing for Performance

Martin Thompson - @mjpt777





"Feynman is becoming a real pain."

"He has the greatest scientific honesty of anyone I've ever meet…"

- William P Rogers

“The impact of QED cannot be overestimated. It explains everything that is not explained by gravity.
It’s also the most accurate theory ever tested by experiments on Earth.”

- Freeman Dyson

“It does not matter how intelligent you are, if you guess and that guess cannot be backed up by experimental evidence – then it is still a guess.”

- Richard Feynman

How do we
Design for Performance?

1. What is Performance?
2. What is Clean & Representative?
3. Implementing efficient Models
4. How to Performance Test

Performance

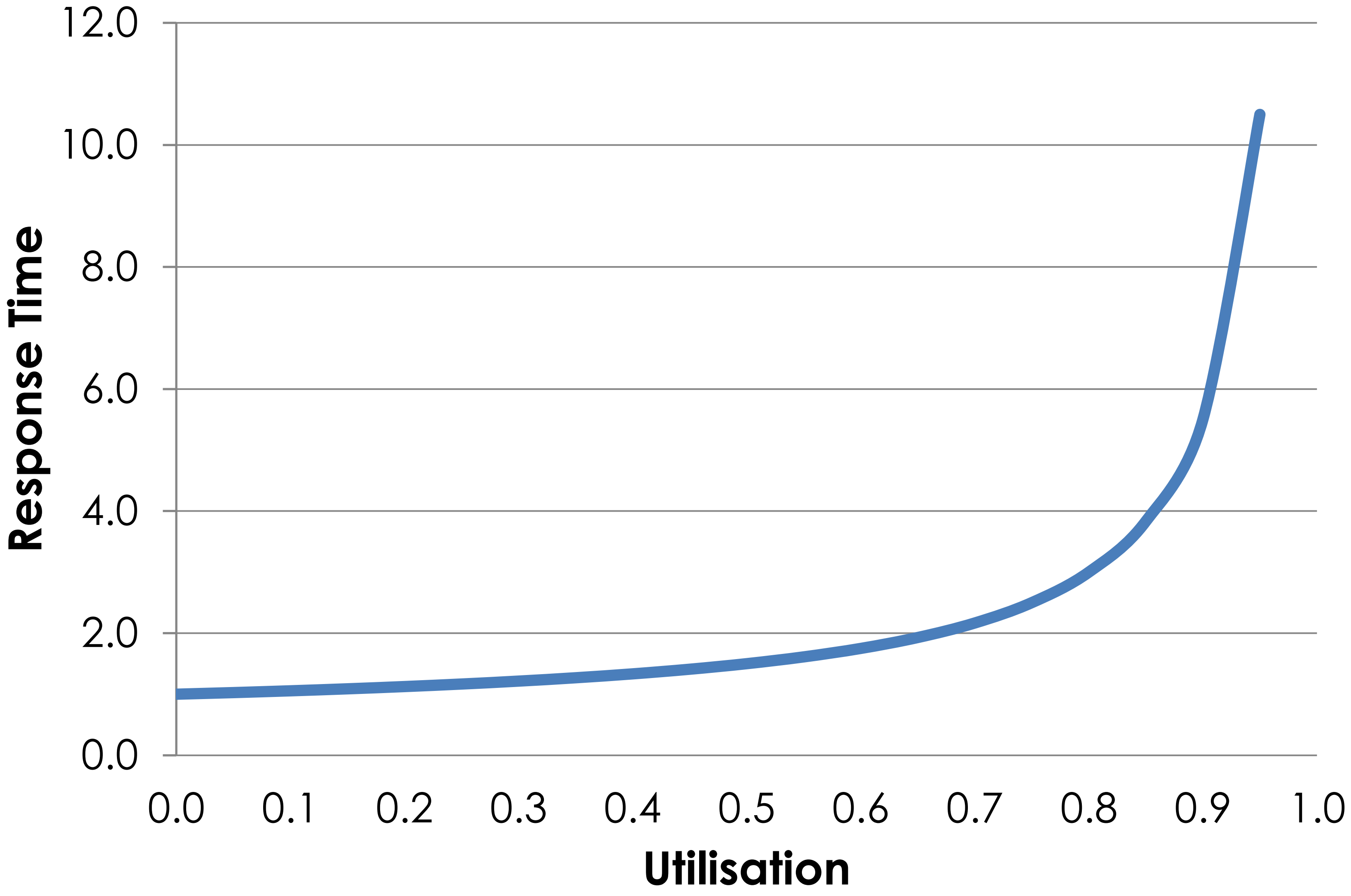
Throughput (aka Bandwidth)

Response Time (aka Latency)

Scalability

UK Border

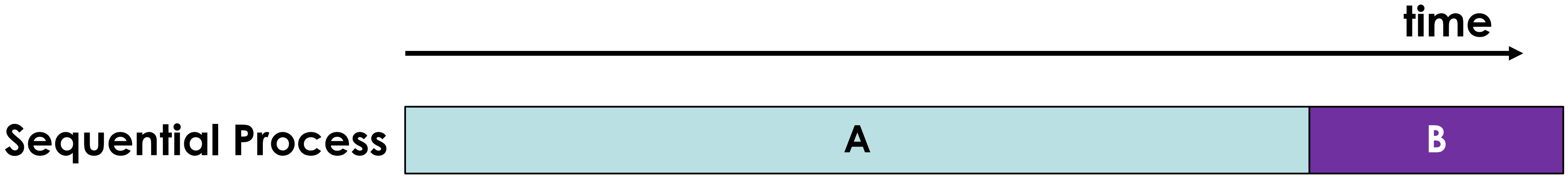
Queuing Theory



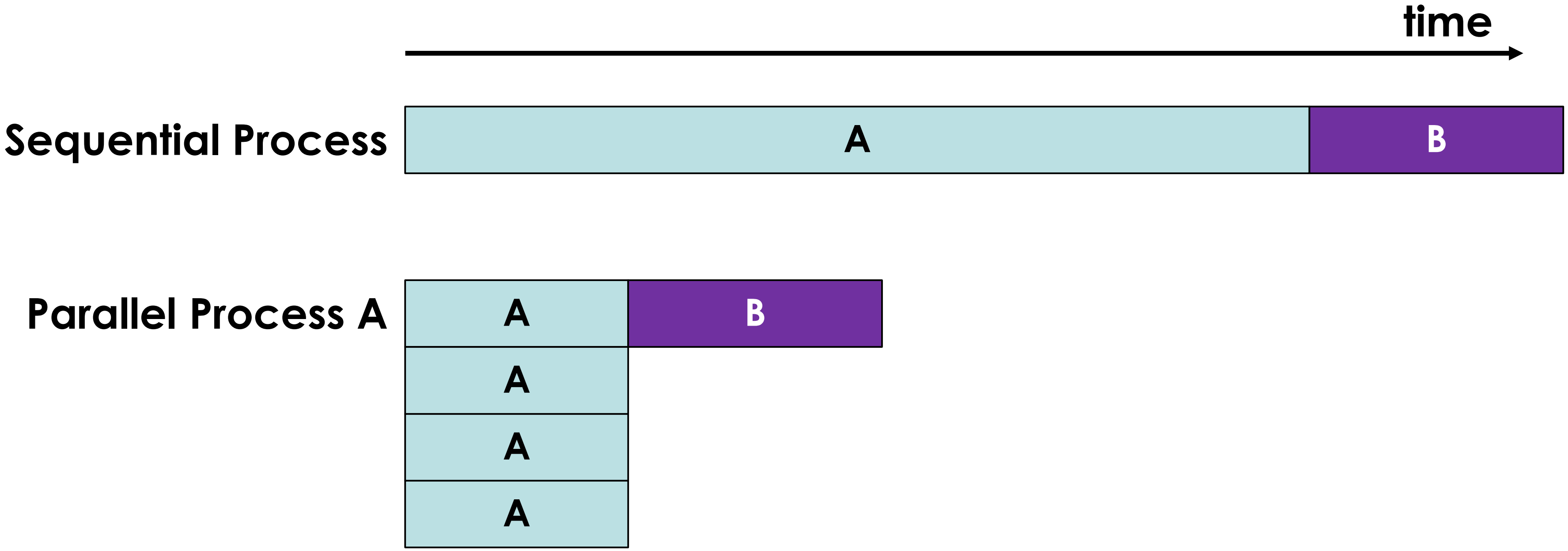
Pro Tip: Ensure you have sufficient capacity

Can we go parallel to speedup?

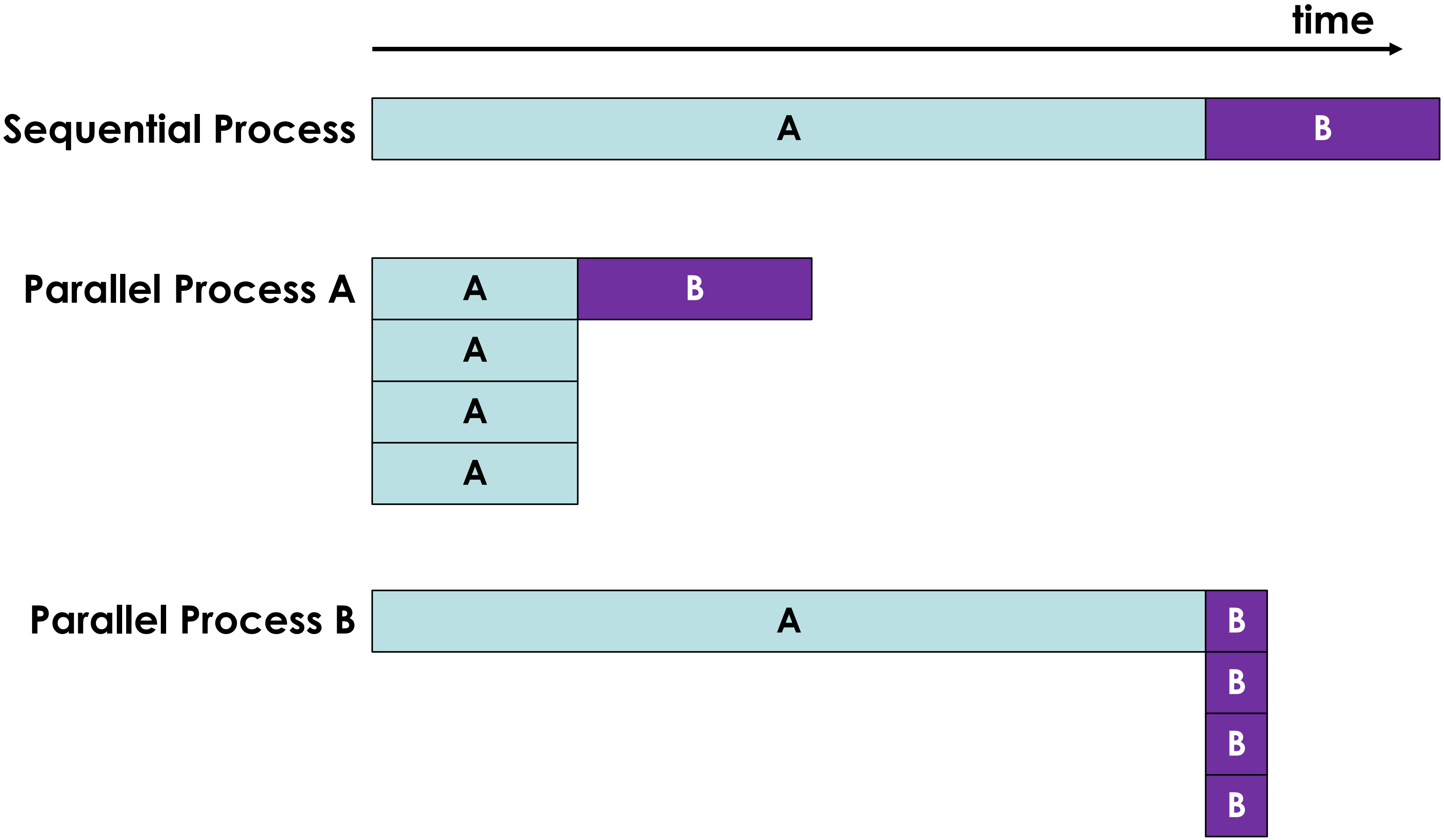
Amdahl's Law



Amdahl’s Law



Amdahl's Law



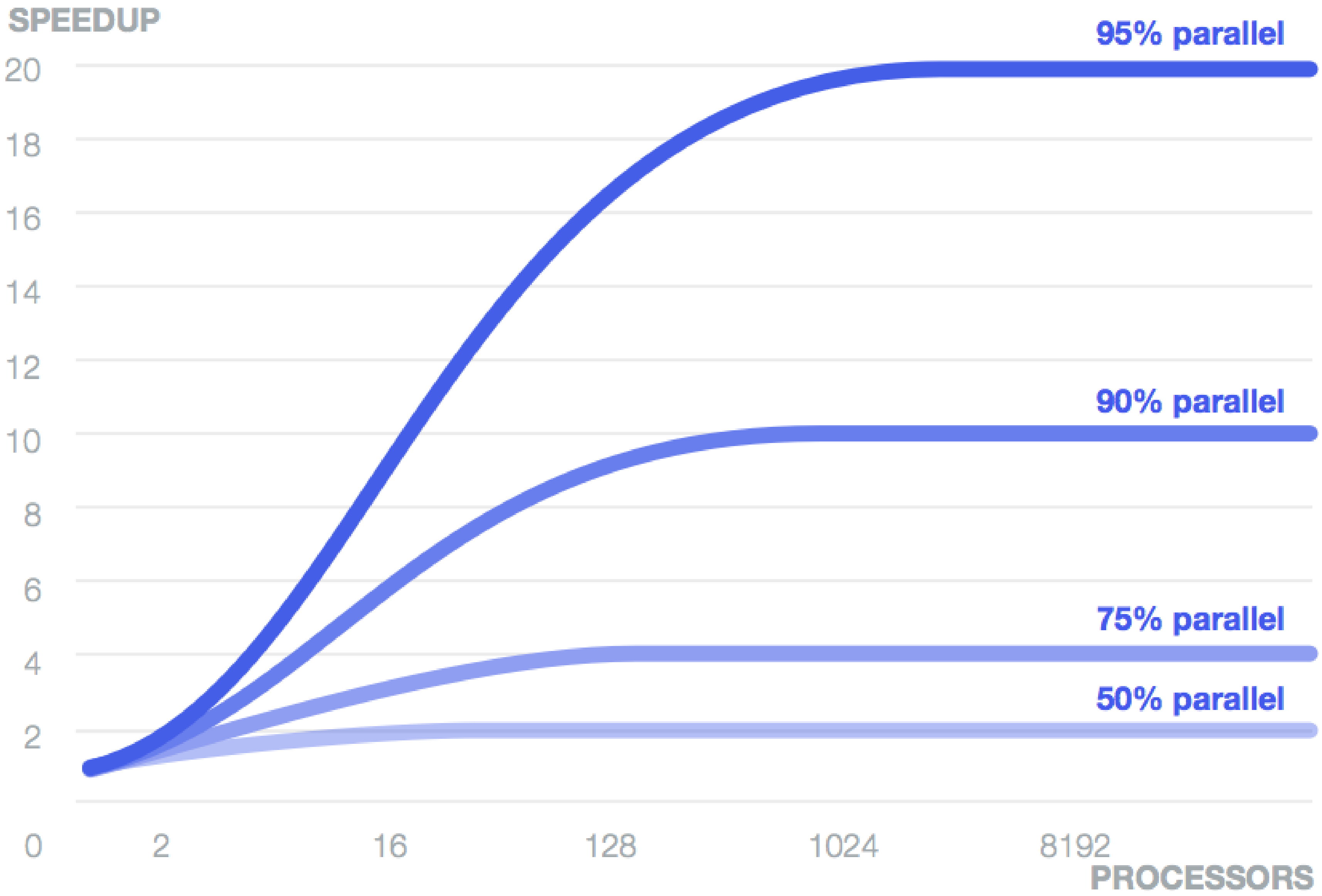
time

Sequential Process
A
B

Parallel Process A
A
B
A
A
A

Parallel Process B
A
B
B
B
B

Amdahl's Law


SPEEDUP
20
18
16
14
12
10
8
6
4
2
0
2
16
128
1024
8192
PROCESSORS
95% parallel
90% parallel
75% parallel
50% parallel

Universal Scalability Law

$$C(N) = N / (1 + \alpha(N - 1) + ((\beta * N) * (N - 1)))$$

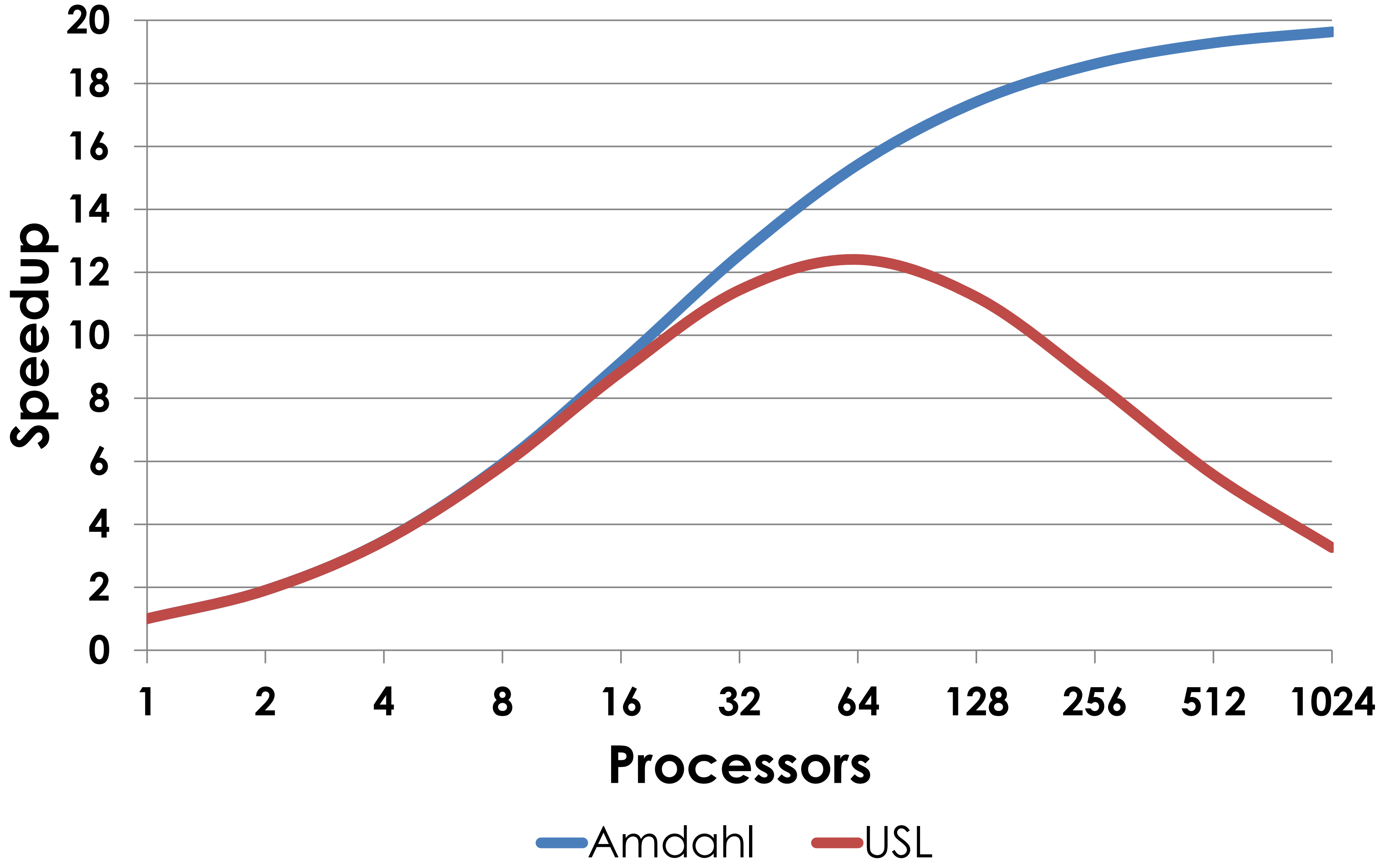
C = capacity or throughput

N = number of processors

α = ***contention*** penalty

β = ***coherence*** penalty

Universal Scalability Law



Clean & Representative

- Clean

“Morally uncontaminated; pure; innocent”

- Oxford English Dictionary

- Representative

“Serving as a portrayal or symbol of something”

- Oxford English Dictionary

- Representative

Code is the best place to capture current understanding of a model

Abstractions

Rules of Abstraction

1. Don’t use abstraction

2. Don’t use abstraction

3. Only consider abstracting when you see at least 3 things that ARE the same

4. Abstractions must pay for themselves

5. Beware DRY, the evil siren that tricks you into abstraction

Abstraction

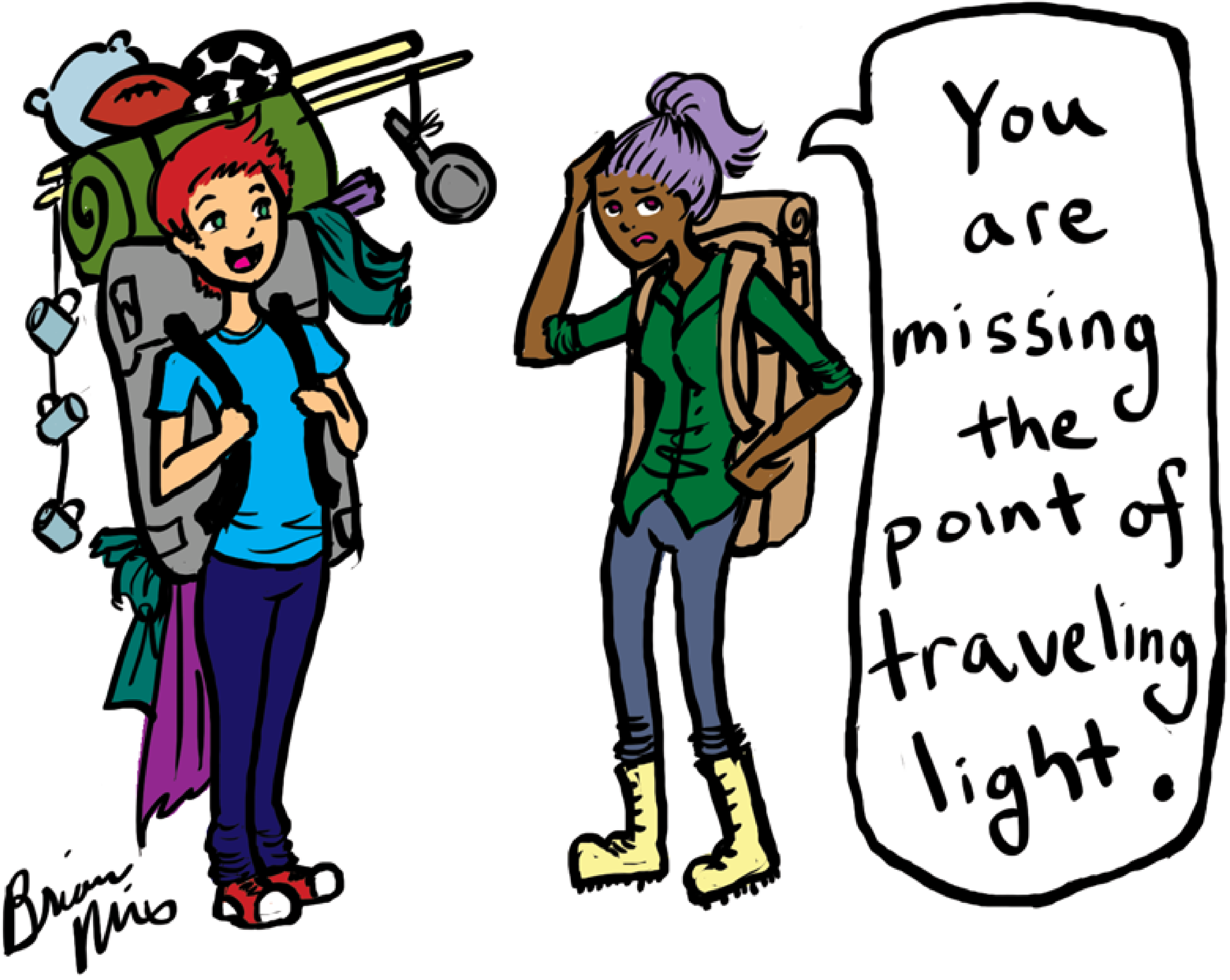
Megamorphism => Branch Hell

Abstraction

Not Representative => Big Smell

Abstraction

Say no to big frameworks!


You are missing the point of traveling light.

Pro Tip: Abstract when you are sure of the benefits

Law of Leaky Abstractions

> ***"All non-trivial abstractions, to some extent, are leaky."***

- Joel Spolsky

Law of Leaky Abstractions

“The detail of underlying complexity cannot be ignored.”

“the purpose of abstracting is not to be vague, but to create a new semantic level in which one can be absolutely precise”

- Dijkstra

How can we abstract memory systems?

- It’s about 3 bets!

1. *The Temporal Bet*

- It's about 3 bets!

1. ***The Temporal Bet***
2. ***The Spatial Bet***

- It's about 3 bets!

1. *The Temporal Bet*
2. *The Spatial Bet*
3. *The Striding Bet*

Model Implementation

Coupling vs Cohesion

Coupling vs Cohesion

```java
public class Queue
{
    private final Object[] buffer;
    private final int capacity;

    // Rest of the code

}
```

Coupling vs Cohesion

```
public class Queue
{
    private final Object[] buffer;
    private final int capacity;

    // Rest of the code

}
```

Coupling vs Cohesion

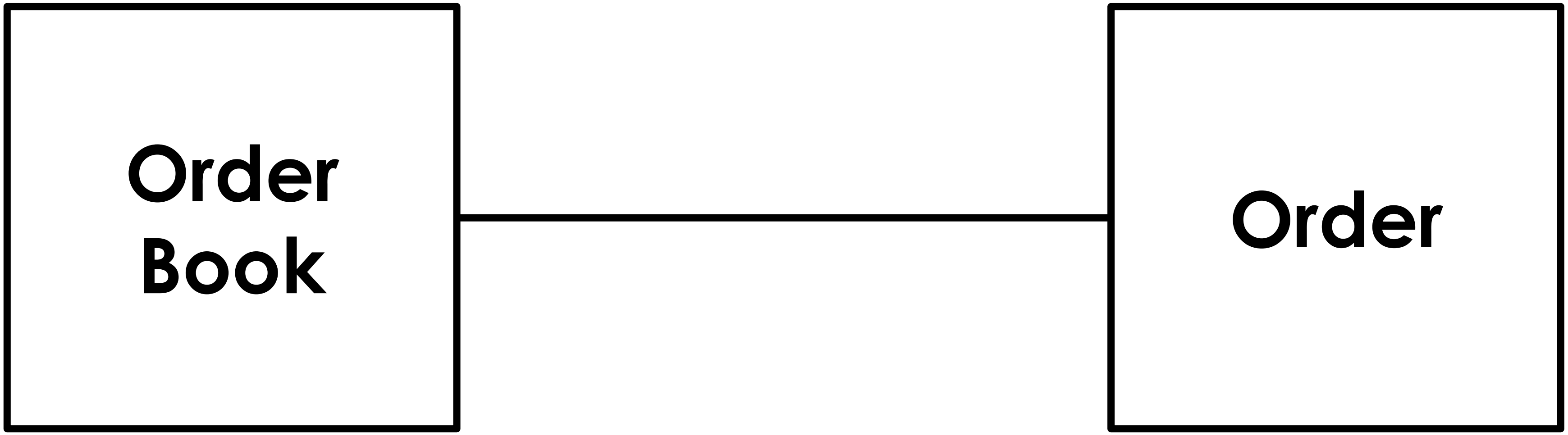
🚫

Properties Bag

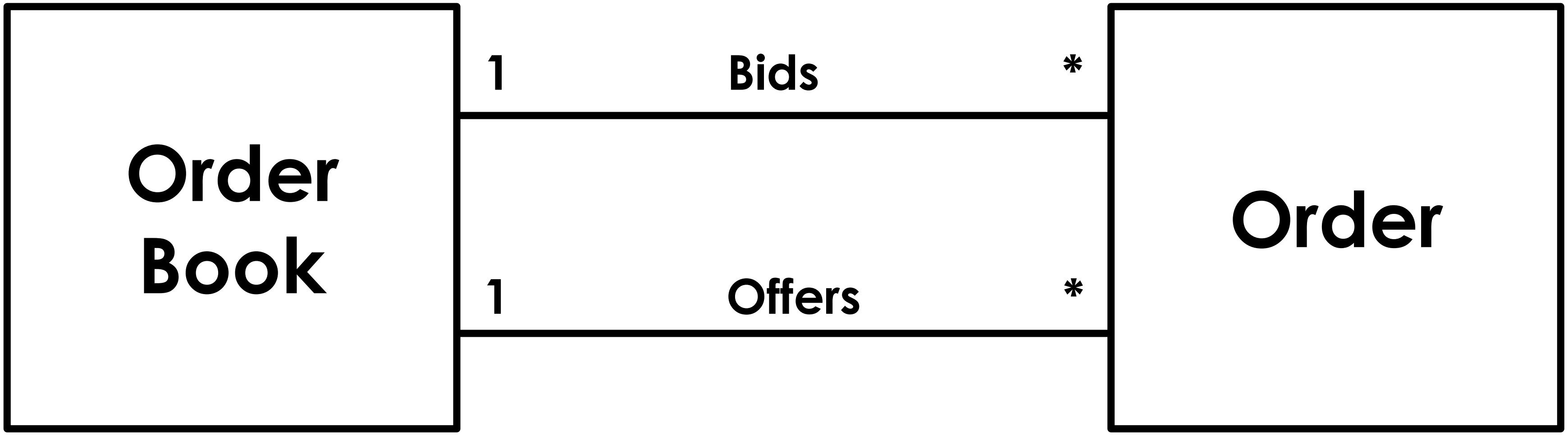
Pro Tip: Respect Locality of Reference

Relationships

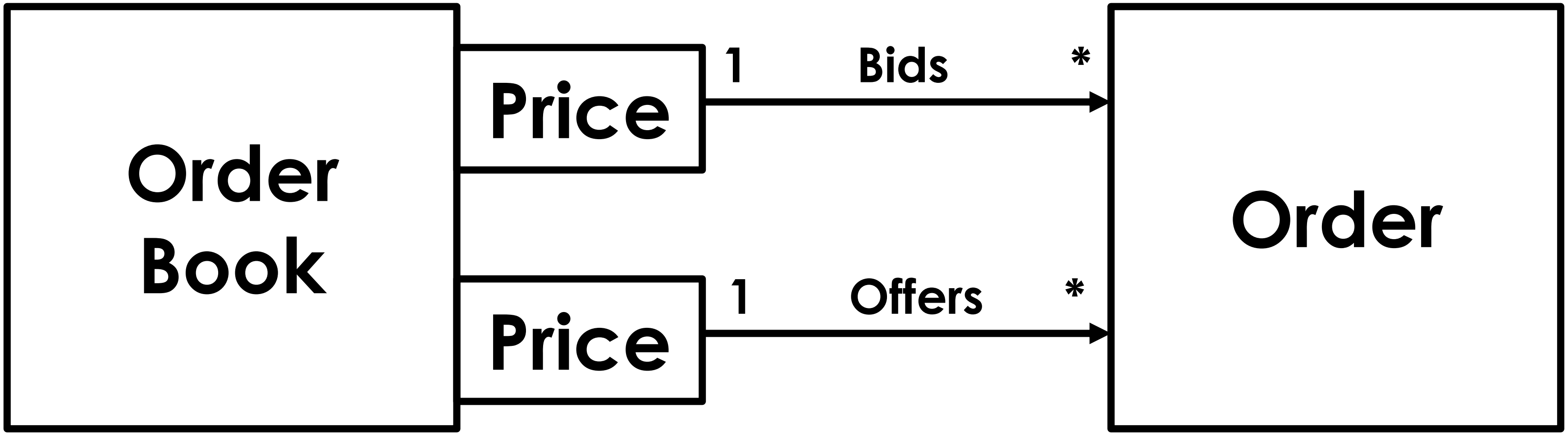
Relationships



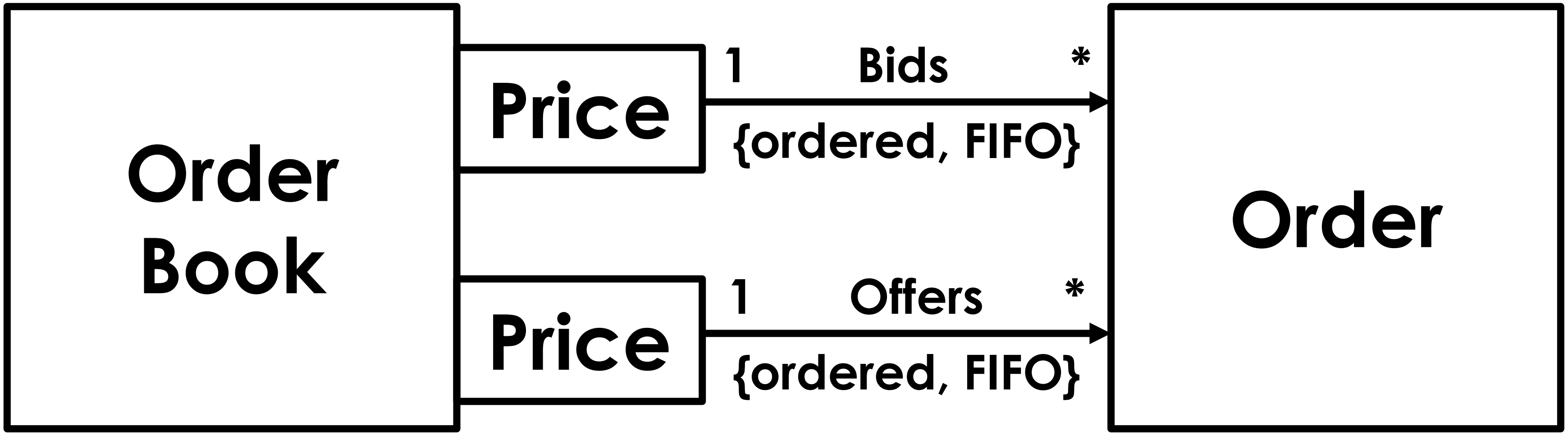
Relationships



Relationships



Relationships

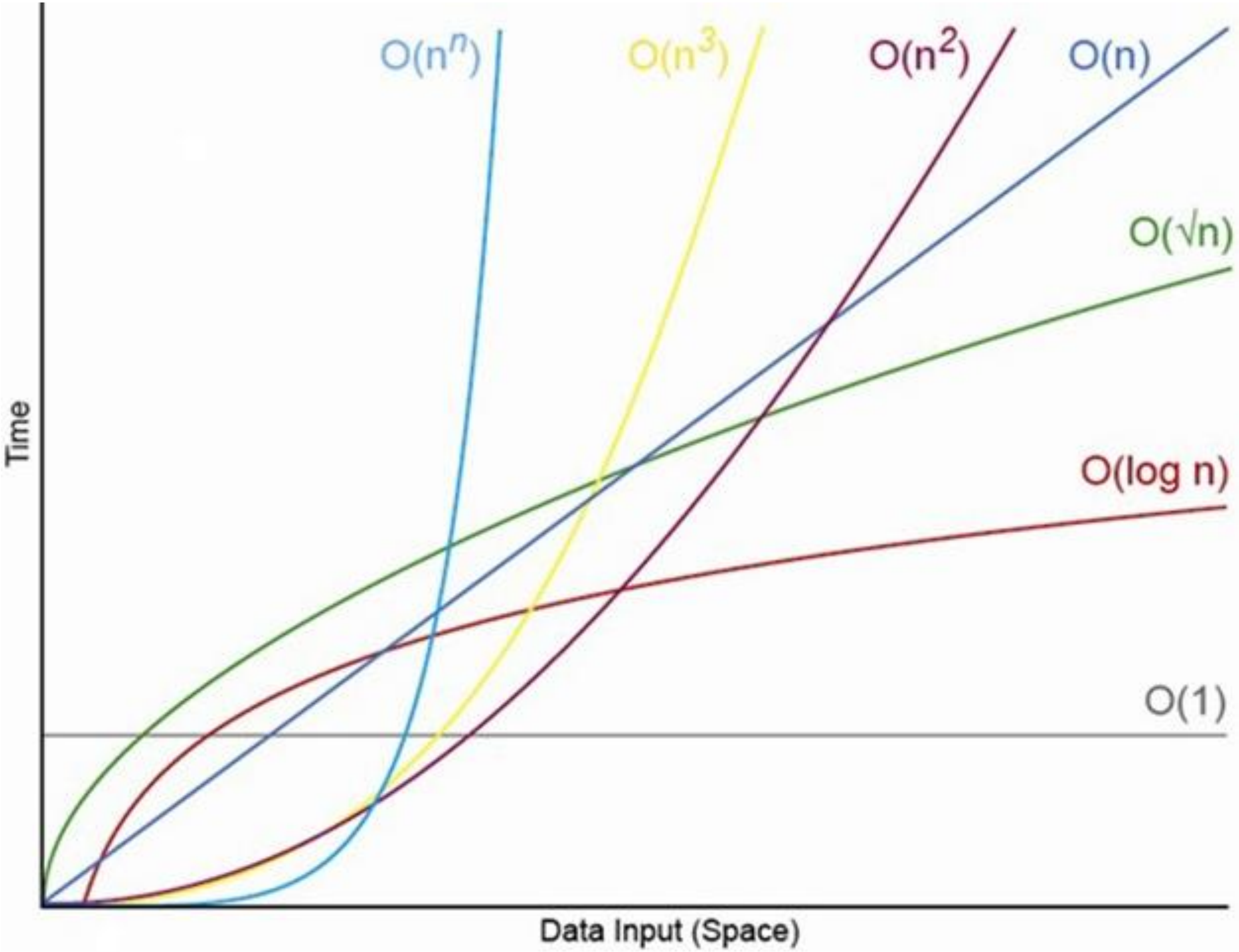


Pro Tip: Make friends with your Data Structures

Pro Tip: Document, discuss, design tests, before going to code

Algorithms

Order of Algorithms



Order of Algorithms

Magnitude of n?

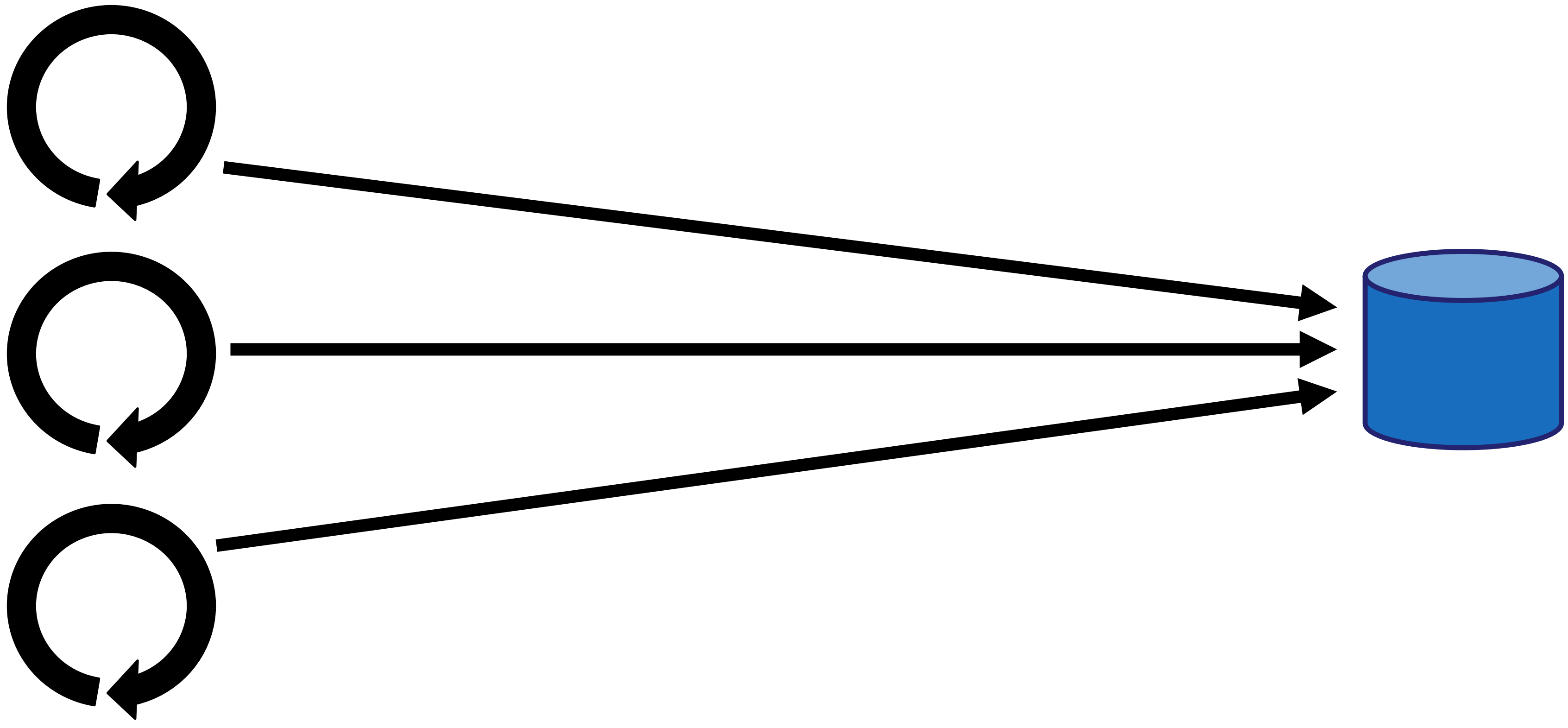
Pro Tip: Know the cardinality of all significant relationships

Pro Tip: Algorithms are your key to service time

Batching

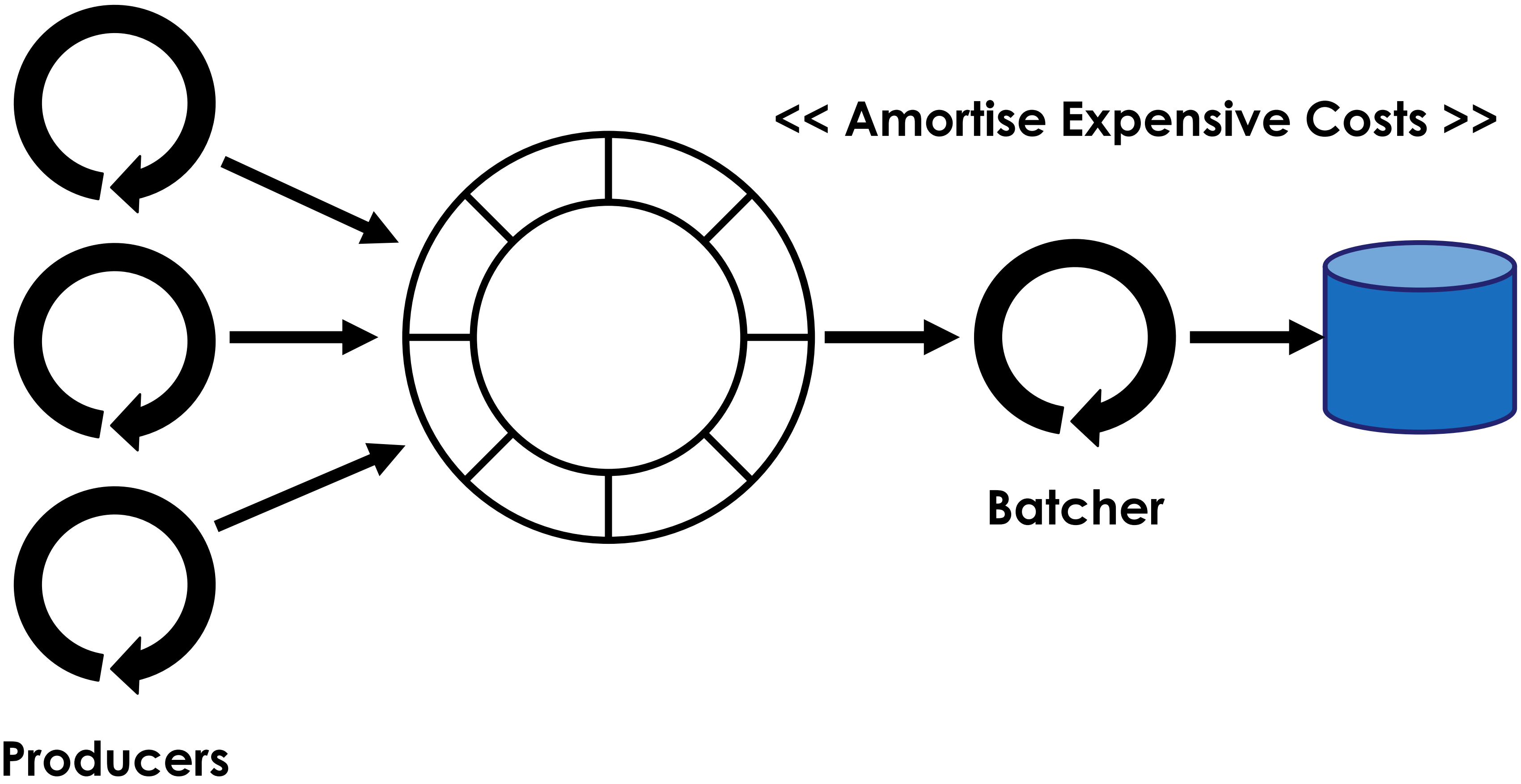
Amortise* *the expensive costs

Natural Batching

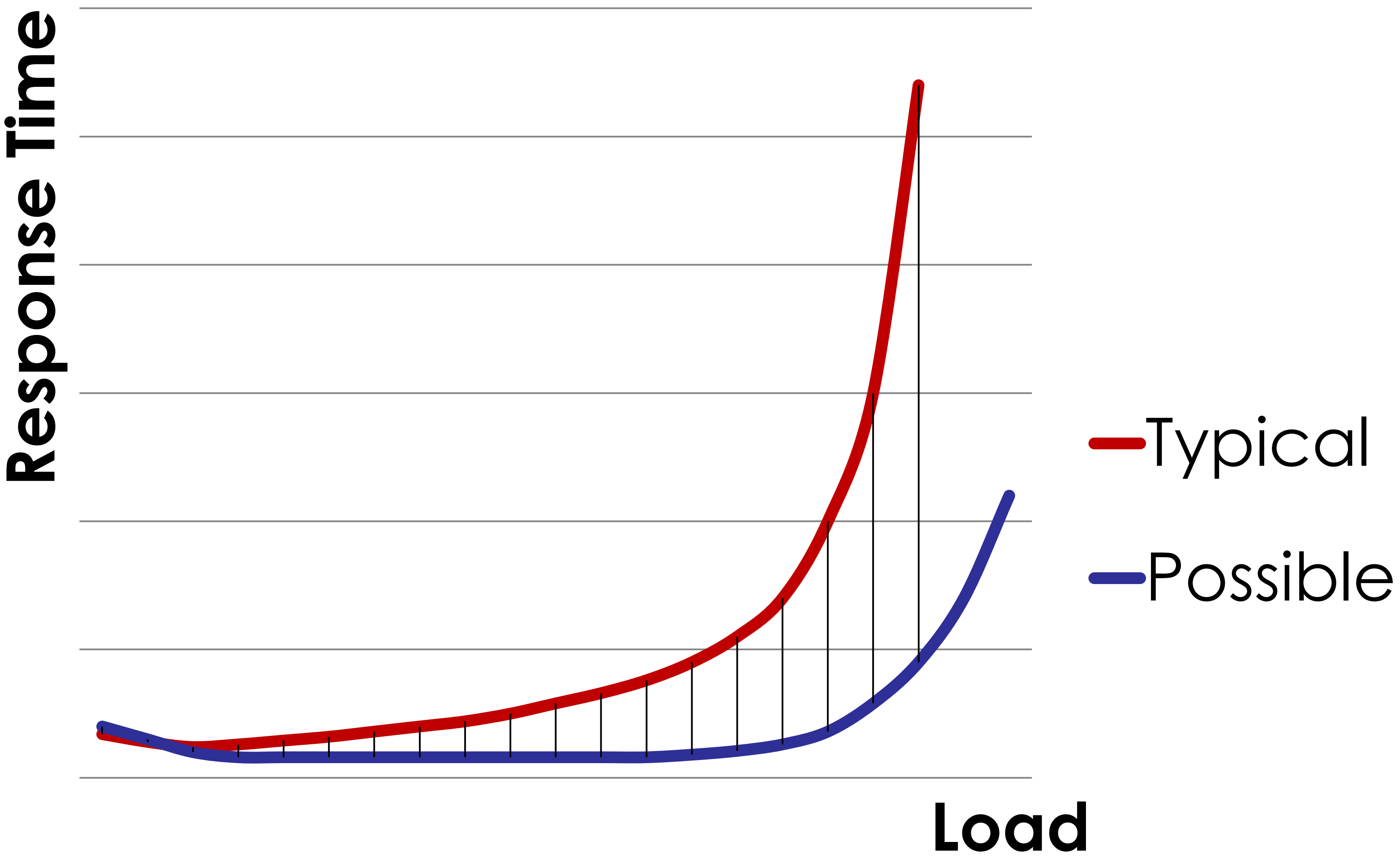


Producers

Natural Batching



Natural Batching



Pro Tip: Batch processing is not just for offline

Branches, branches, branches…

Branches

```
public void doStuff(List<String> things)
{
    if (null == things || things.isEmpty())
    {
        return;
    }

    for (String thing : things)
    {
        // Do useful work
    }
}
```

Branches

```
public void doStuff(List<String> things)
{
    if (null == things || things.isEmpty())
    {
        return;
    }

    for (String thing : things)
    {
        // Do useful work
    }
}
```

Branches

```
public void doStuff(List<String> things)
{
    for (String thing : things)
    {
        // Do useful work
    }
}
```

Pro Tip: Respect the Principle of least surprise

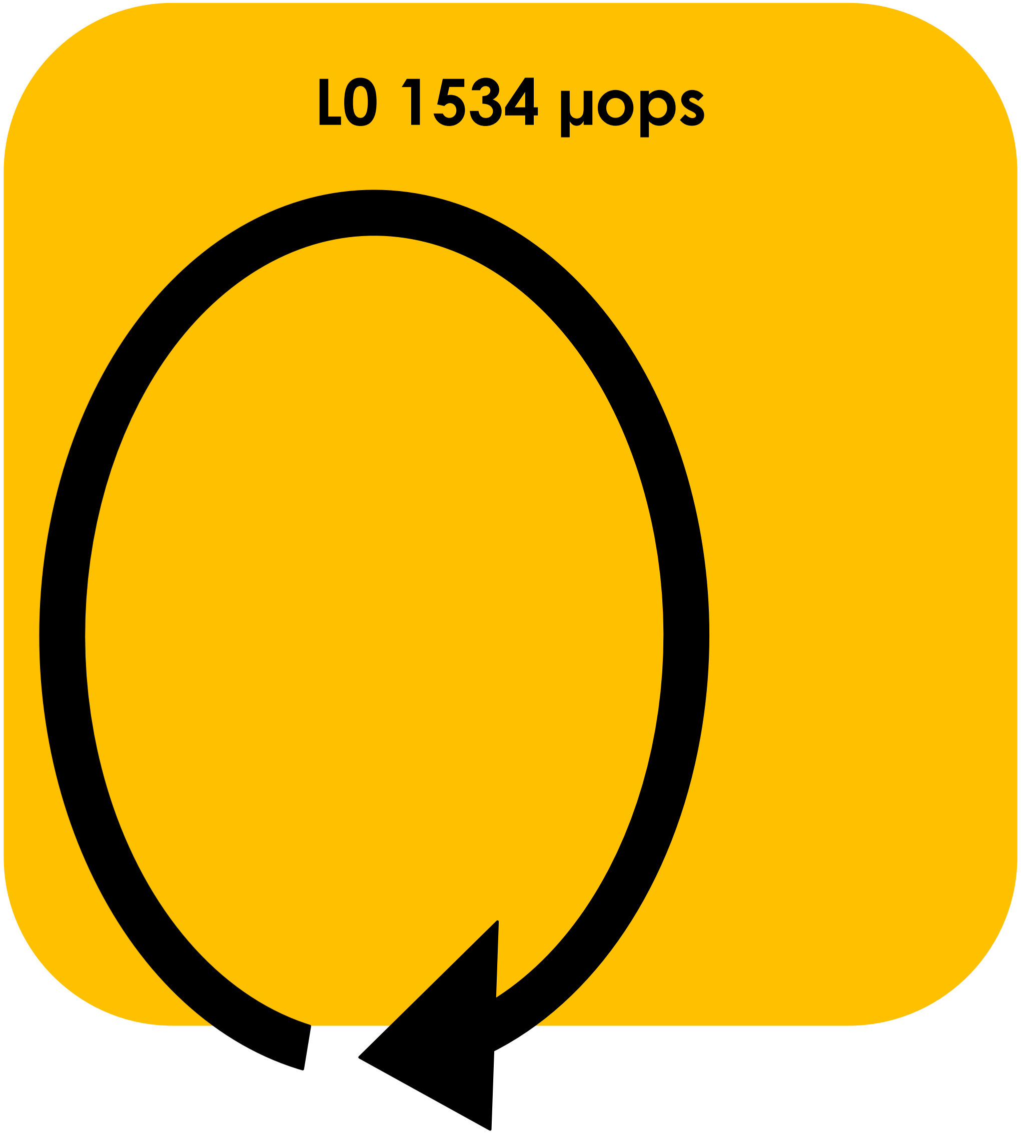
Loops

Loops

“If I had more time, I would have written a shorter letter.”

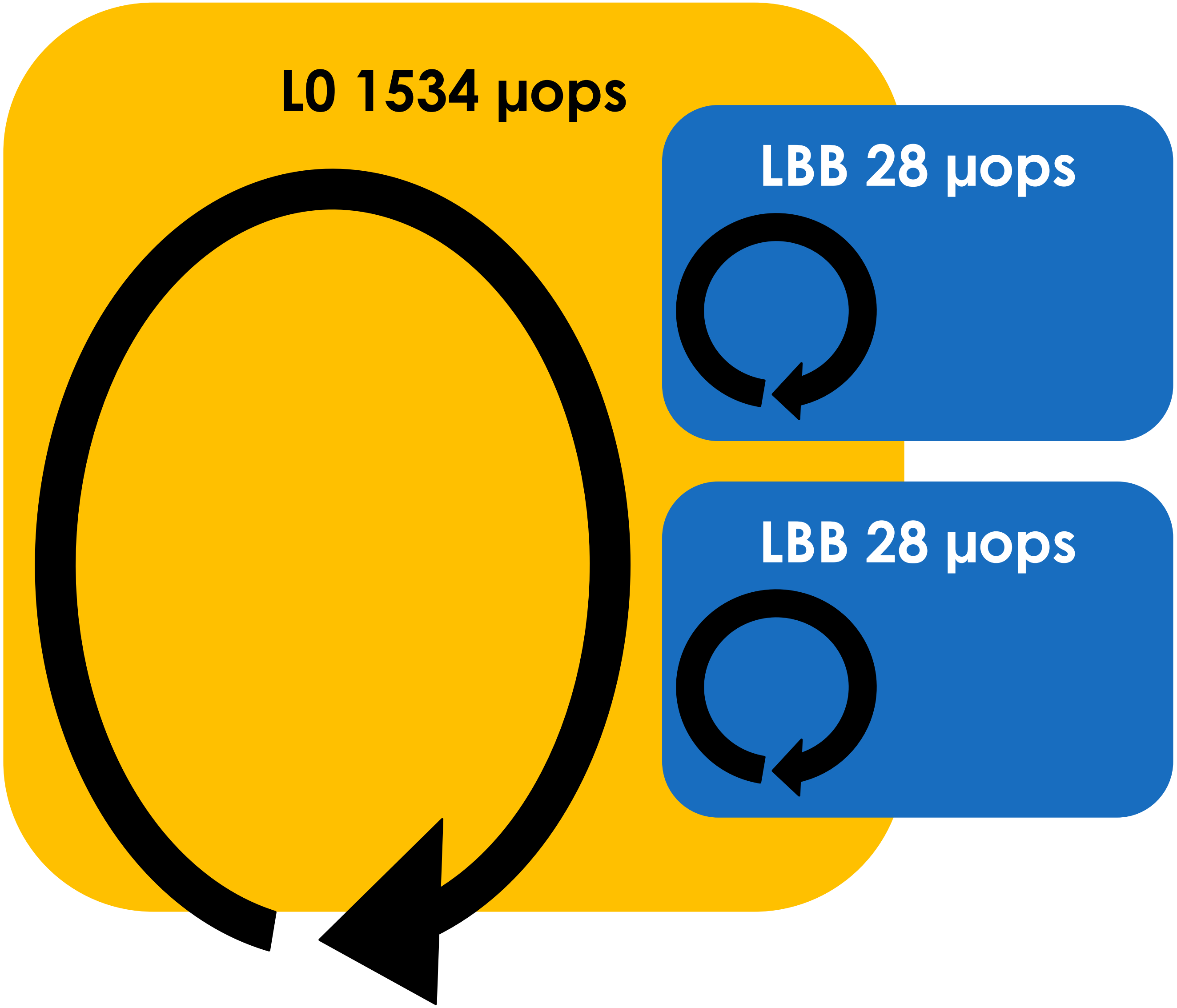
- Blaise Pascal

Loops



L0 1534 µops

Loops



Pro Tip: Craft major loops like good prose

Composition

Composition

Size matters

Composition

“Inlining is THE optimisation.”

- Cliff Click

Composition

Single Responsibility

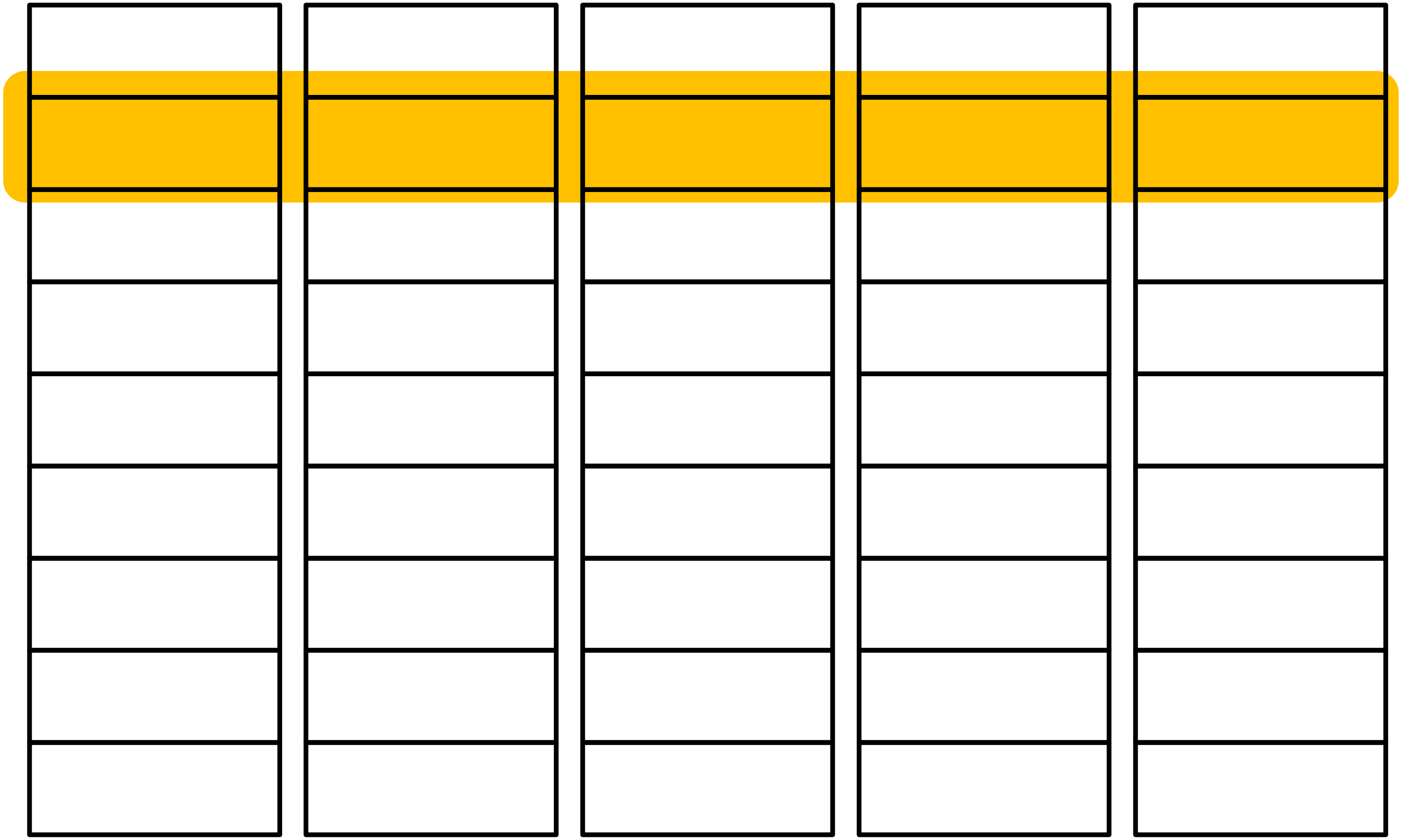
Pro Tip: Small atoms can compose to build anything

Data

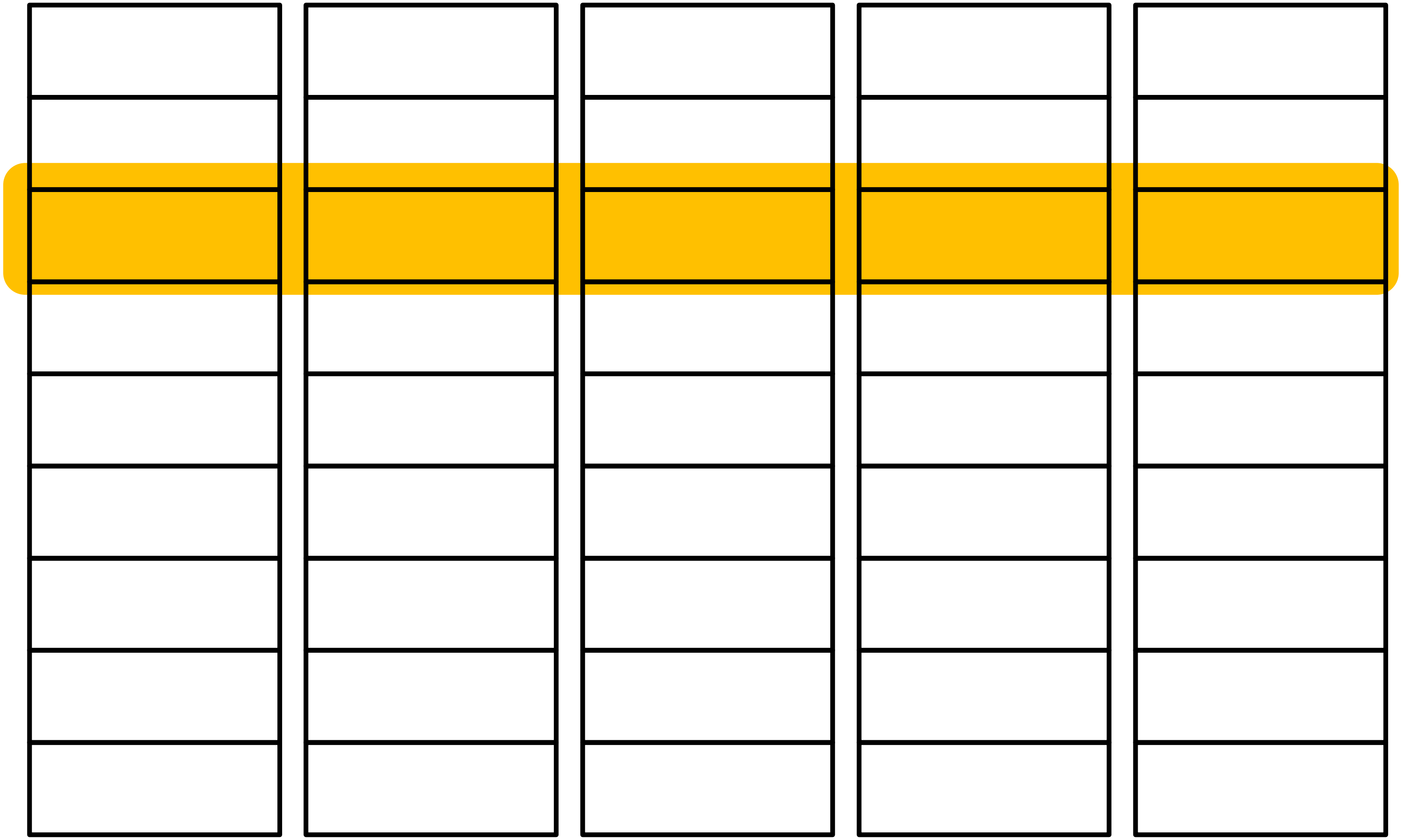
Data

Data

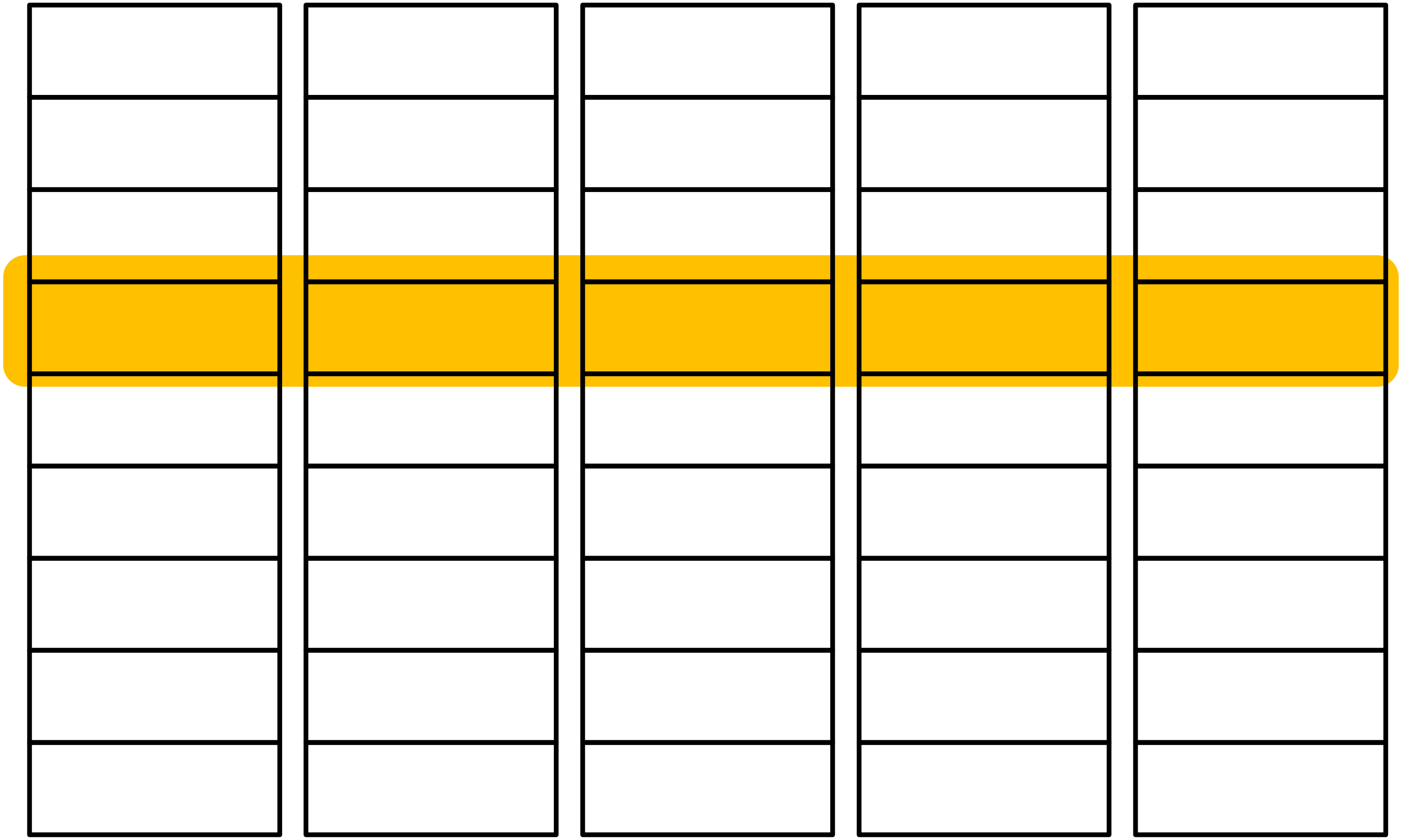
Data



Data



Data



Pro Tip:

Embrace Set Theory and FP techniques

Performance Testing

Define Performance Goals

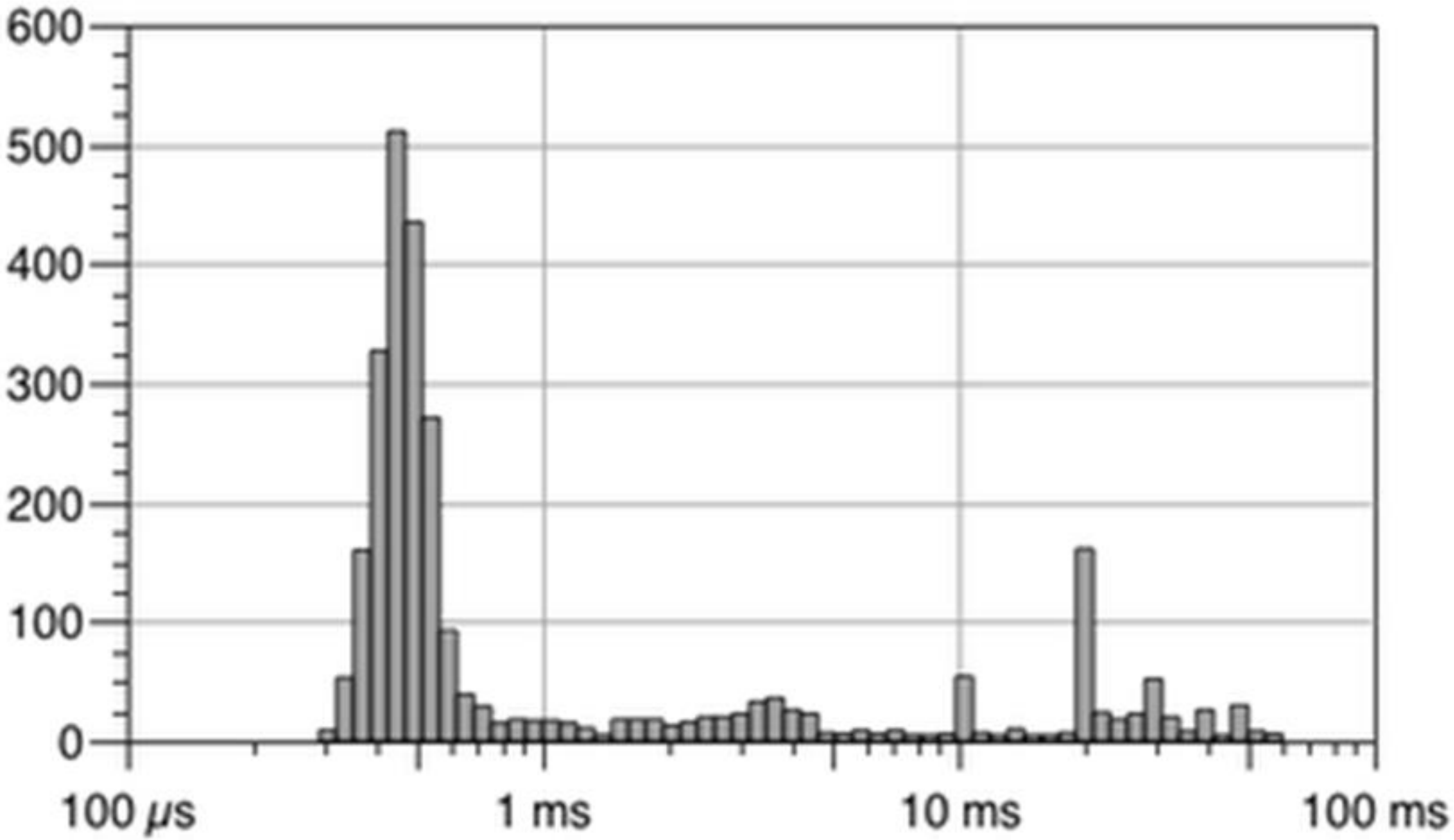
Establish Design Principles

Aeron Design Principles

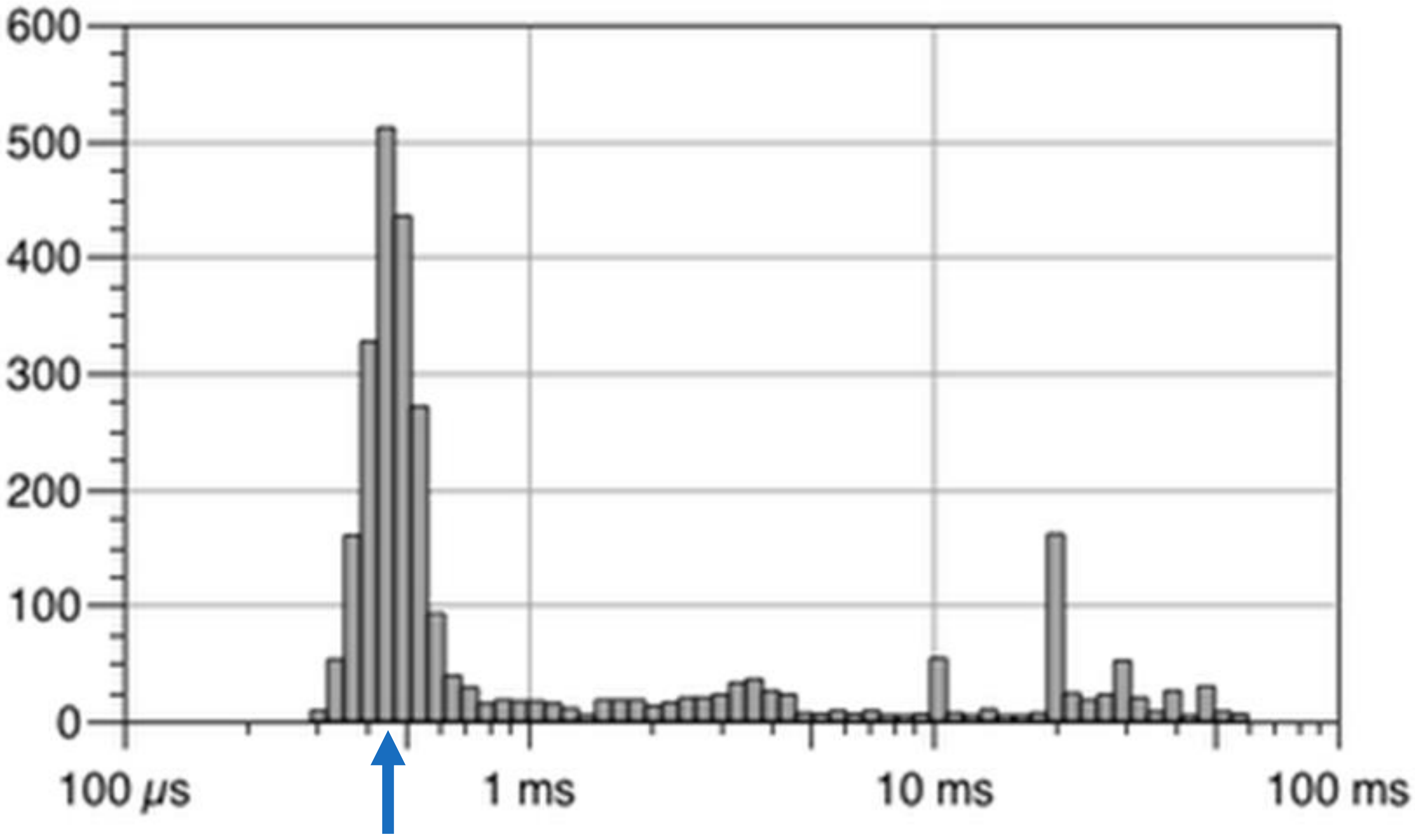
1. Garbage free in steady state running
2. Smart Batching in the message path
3. Lock-free algos in the message path
4. Non-blocking IO in the message path
5. No exceptional cases in message path
6. Apply the *Single Writer Principle*
7. Prefer unshared state
8. Avoid unnecessary data copies

How to measure response time?

Response Time Histograms

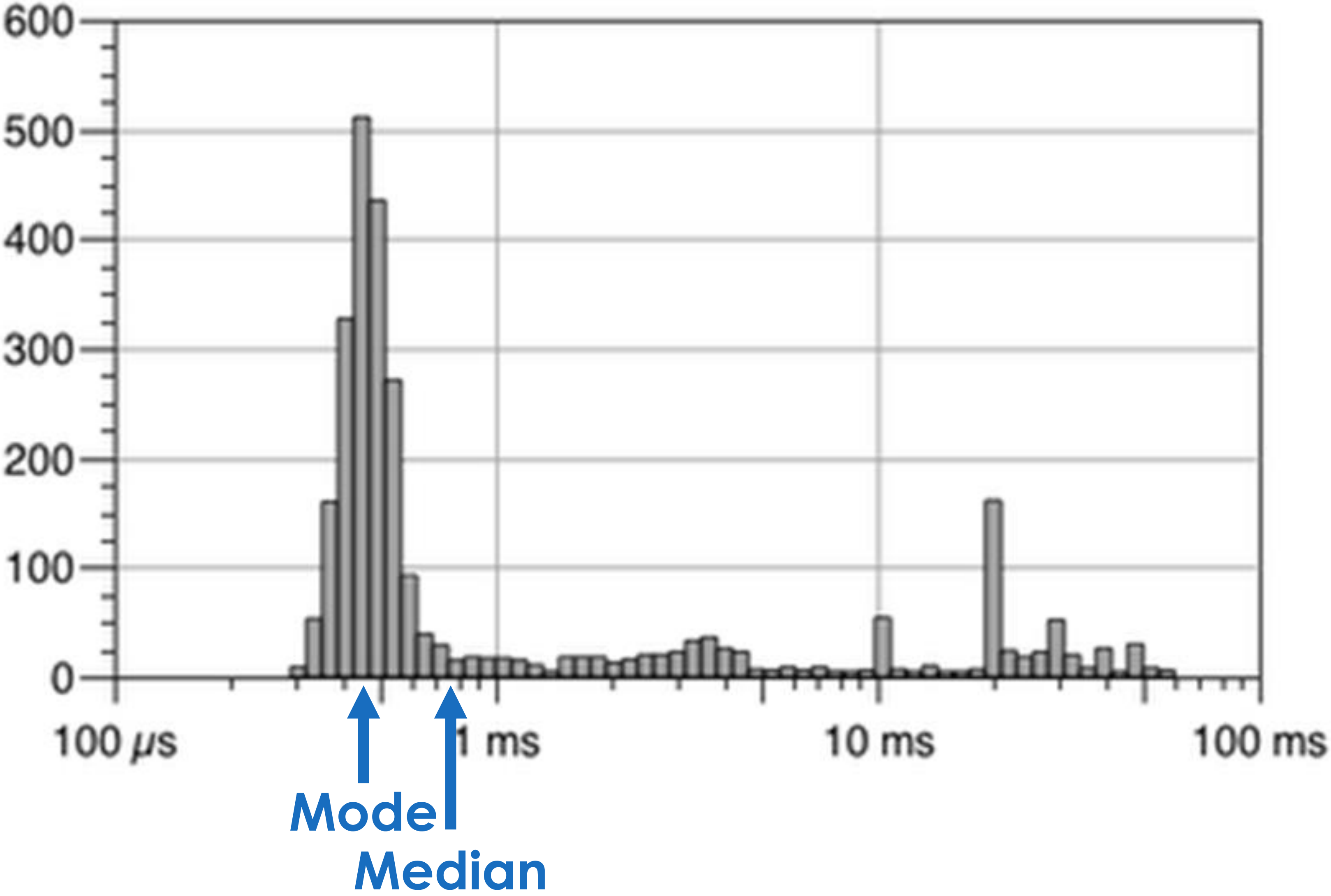


Response Time Histograms



Mode

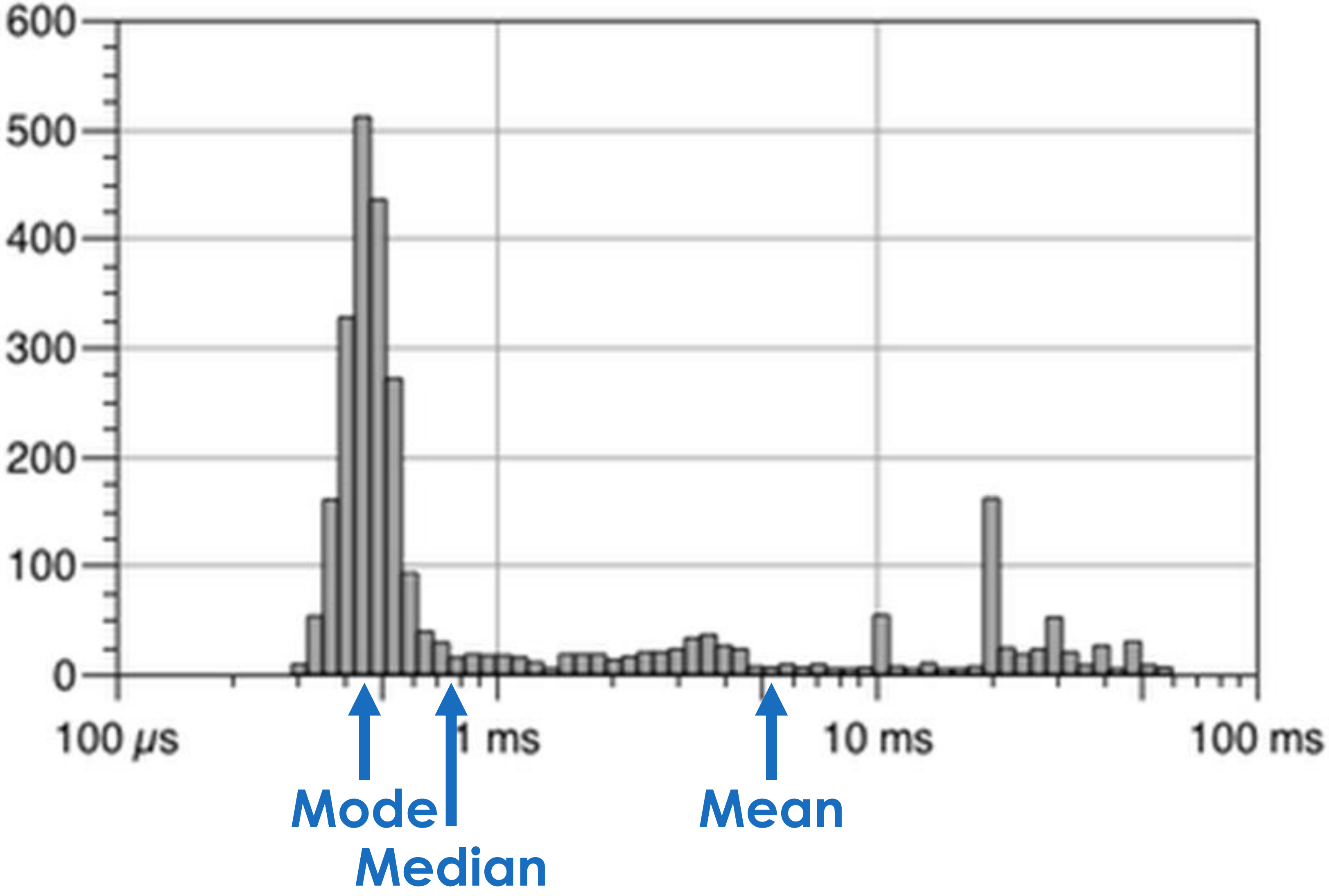
Response Time Histograms



Mode

Median

Response Time Histograms

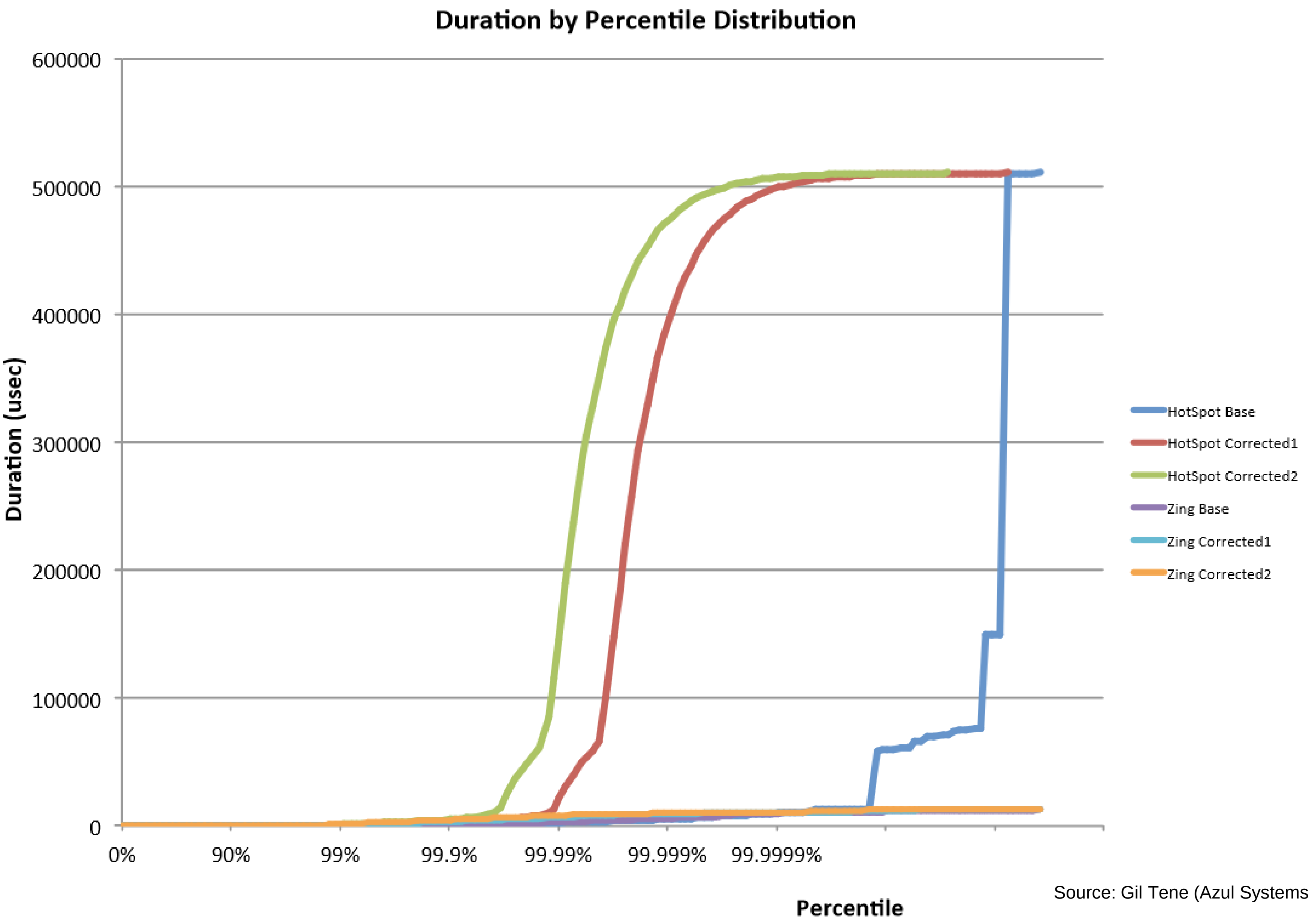


Mode

Median

Mean

Coordinated Omission



Source: Gil Tene (Azul Systems)

HdrHistogram

Java Microbenchmark Harness

CPU Performance Counters

Performance test as part of Continuous Integration

Can your acceptance tests run as performance tests?

Build telemetry into production systems

AGAIN!!!

Build telemetry into production systems

Counters of:

- Queue Lengths
- Concurrent Users
- Exceptions
- Transactions - orders, trades
- Etc.

Histograms of:

- Response Times
- Service Times
- Queue Lengths
- Concurrent Users
- Etc.

In closing…

Clean => Uncontaminated

Representative => True Portrayal

Does it pass the “Out Loud” test?

Measure – Don’t Guess!!!

BOMBARDIER
LEARJET 45
LEARJET 45
LEARJET
45

Questions?

http://mechanical-sympathy.blogspot.com/

Twitter: @mjpt777

> *"It does not matter how intelligent you are, if you guess and that guess cannot be backed up by experimental evidence –*
> *then it is still a guess."*

- Richard Feynman