

**performing  
databases**

# Performance is Rarely an Accident

inspired by CARY MILLSAP's "PERFORMANCE IS A FEATURE"

Martin Klier 

Performing Databases GmbH  
Mitterteich / Germany



Blink  
and  
you  
miss  
it!

# Speaker

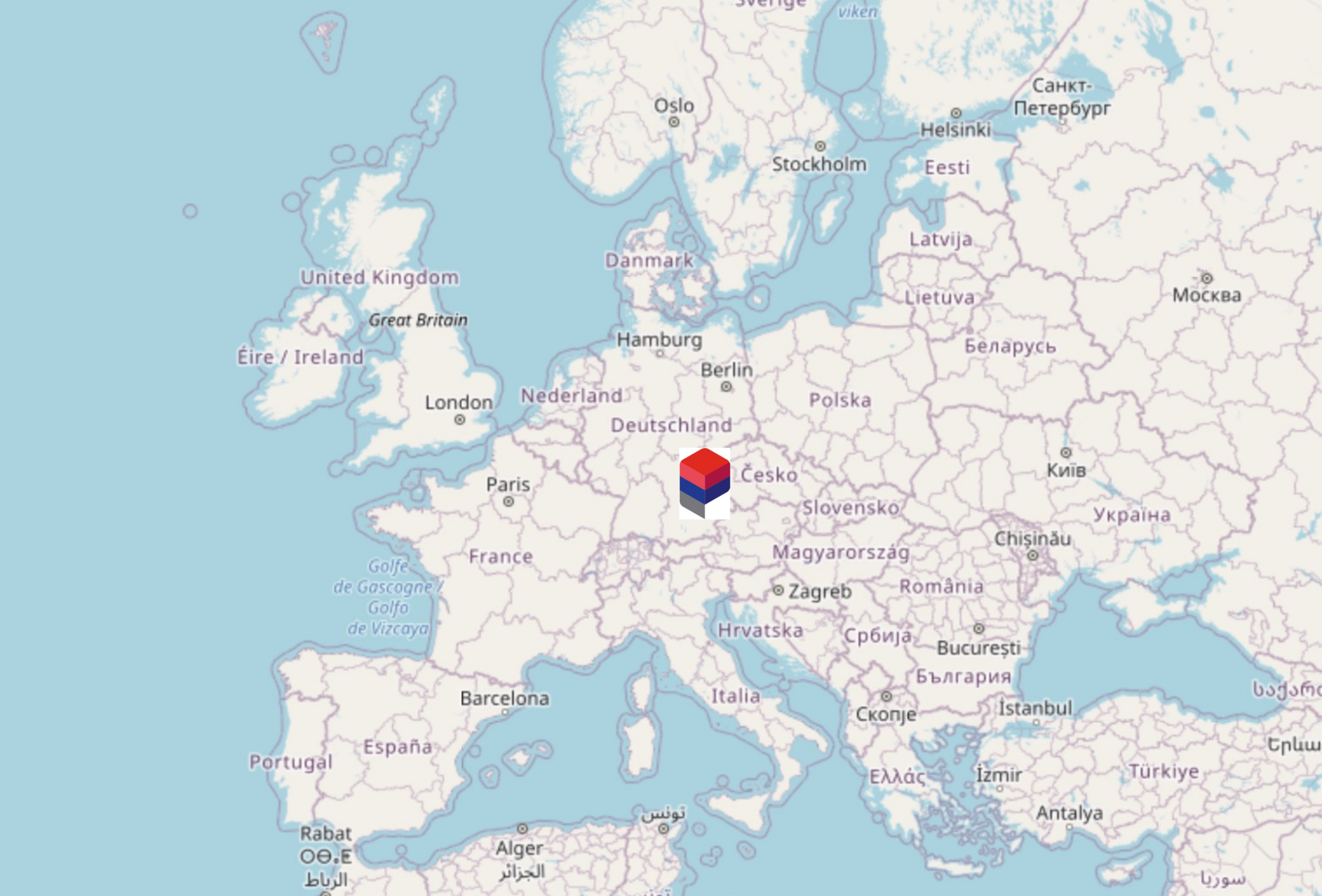
- Martin Klier
- Solution Architect and Database Expert
- My focus:
  - Performance + Tuning
  - Highly available systems
  - Cluster and Replication
- Linux since 1997
- Oracle Database since 2003



# Performing Databases

- Three Experts for Database technology
  - Concepts and Project Competence
  - Architecture- and System planning
  - Licensing
  - Implementation and Troubleshooting
- Get in touch
  - Performing Databases GmbH  
Wiesauer Strasse 27  
95666 Mitterteich // Germany
  - <http://www.performing-databases.com>
  - Twitter: @PerformingDB





# SYMPOSIUM 42

Created by the community, to support the community

Sharing of reliable knowledge

Supporting the various user groups and individuals



@sym\_42



<https://sym42.org/>

# Excellence



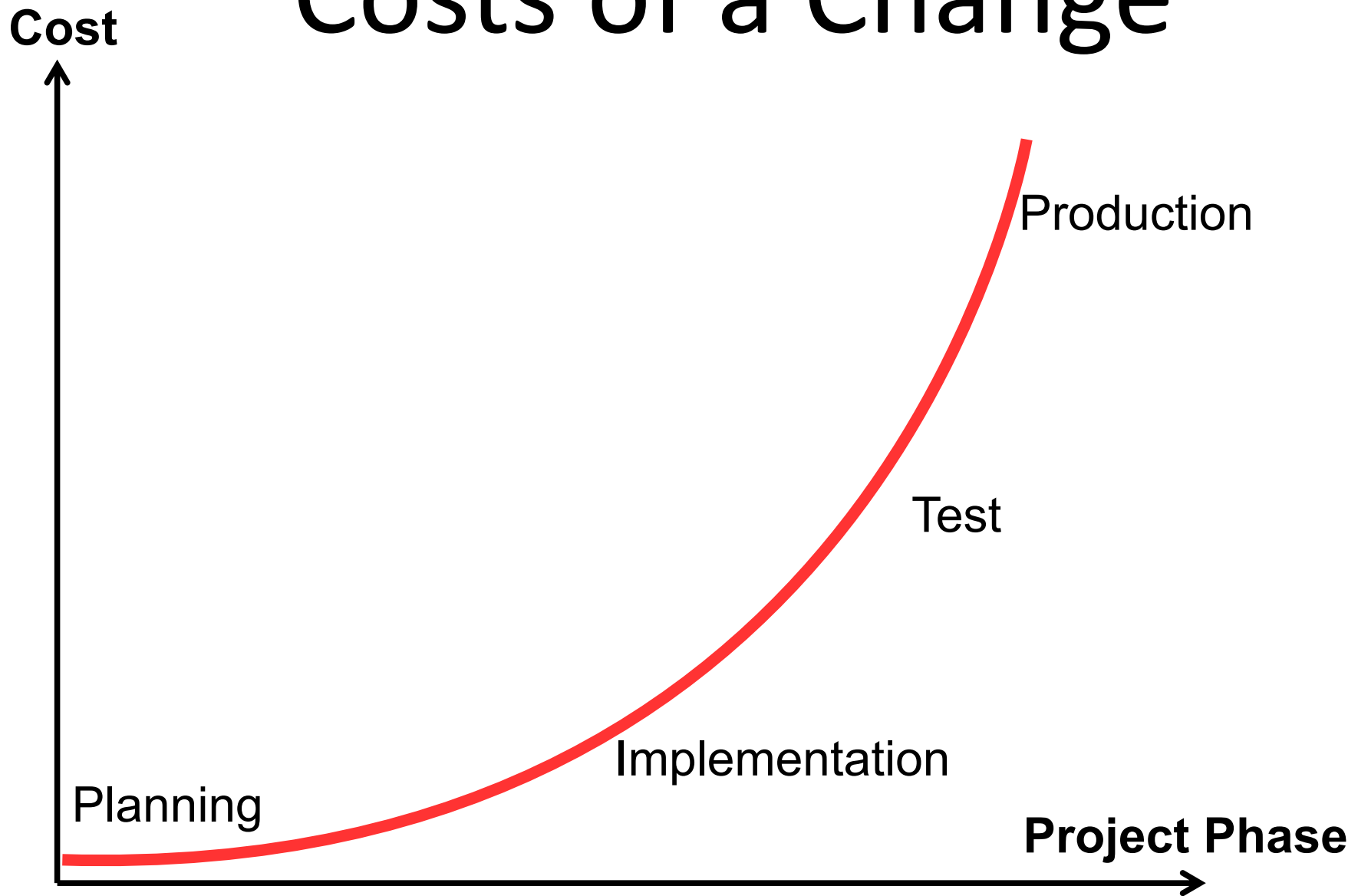
# Performance is rarely an accident

**Know it  
counts**

**Prove it  
counts more**

# Boehm

# Costs of a Change



# We need a plan ...

# PERFORMANCE

/pə'fɔrməns/

# Performance



# Performance

## Work per Time

Kilometers per hour, tons per month ...

## Time per work unit

Seconds per booking, days per project



Throughput:

$$p = \frac{\textit{task}}{\textit{time}}$$

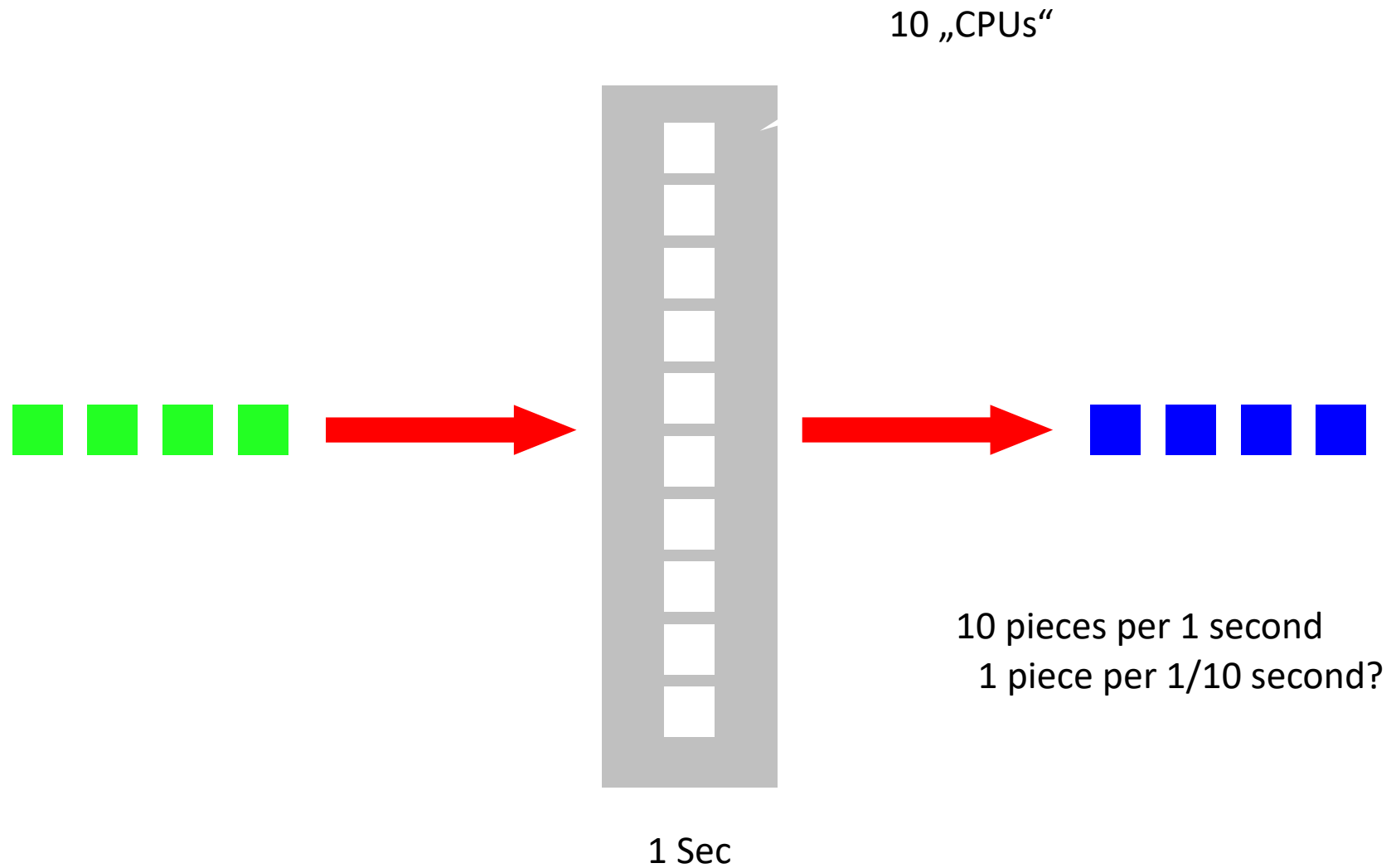
Response time:

$$R = \frac{\textit{time}}{\textit{task}}$$

BUT:

$$R \neq \frac{1}{p}$$

# Inverse Games



# The Mean

# Of the mean mean + Skew

0,9; 1,3; 0,9; 0,9;

0,5; 0,5; 0,5; 0,5; 0,5; 5,5; 0,5; 0,5; 0,5; 0,5;

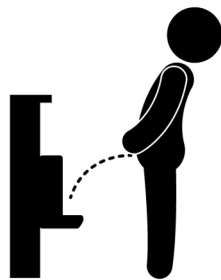
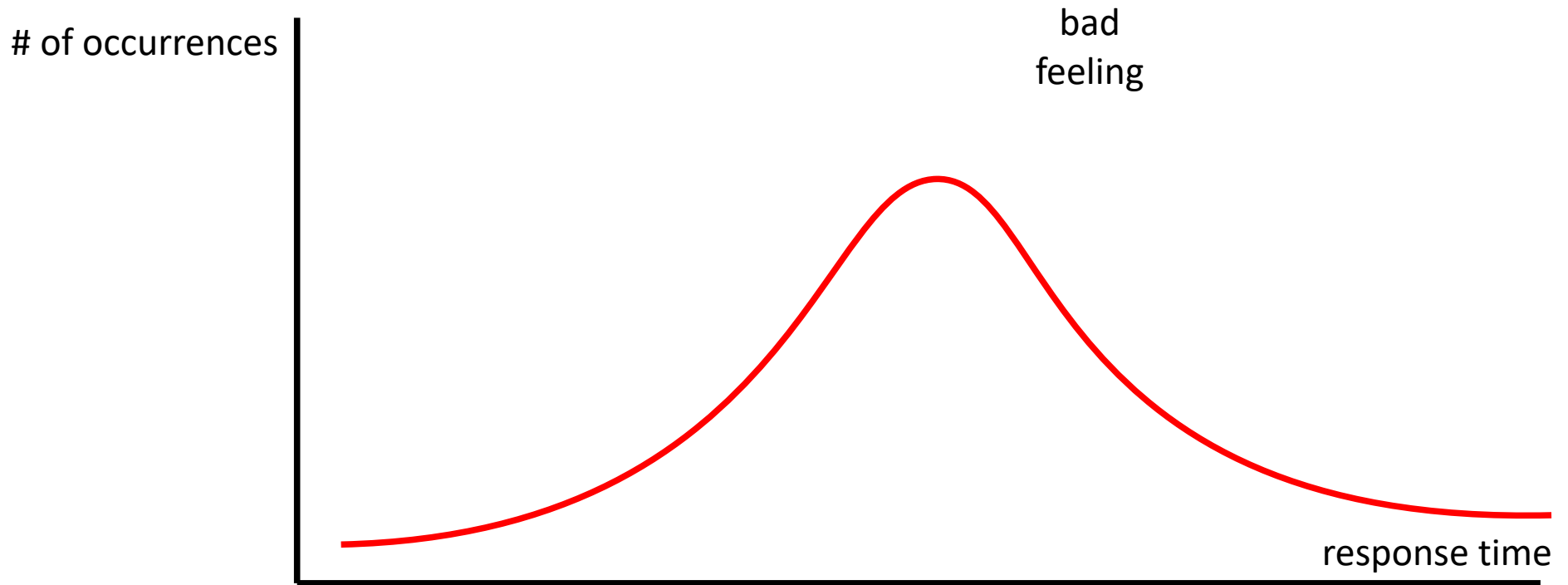
# Of the mean mean ...

Specs for response time:

1s in 99%

Not too bad, but ...

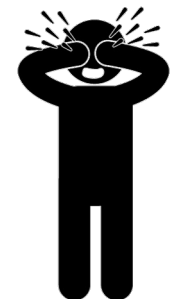
# Users feel the difference – not the mean



faster

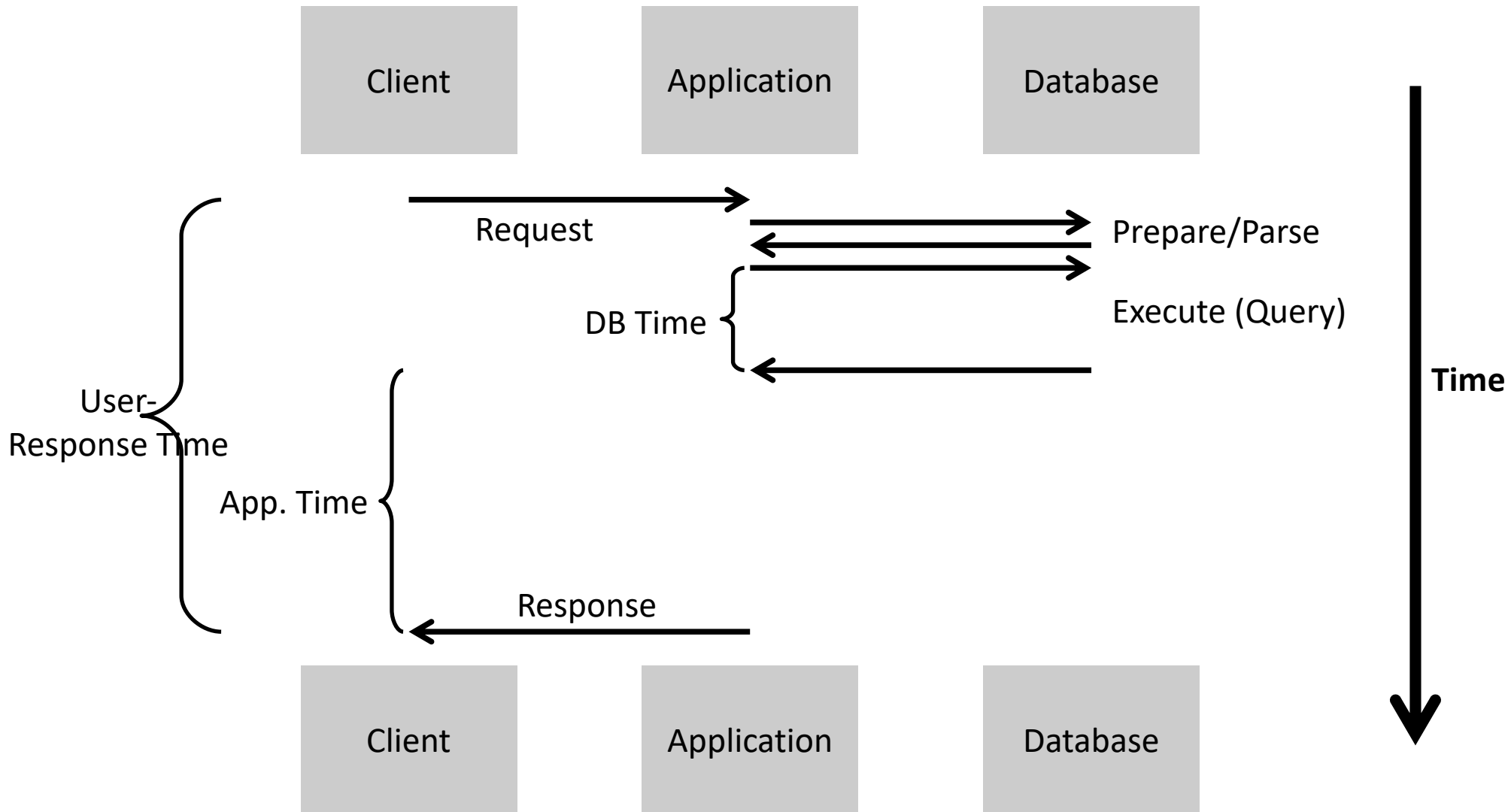


slower



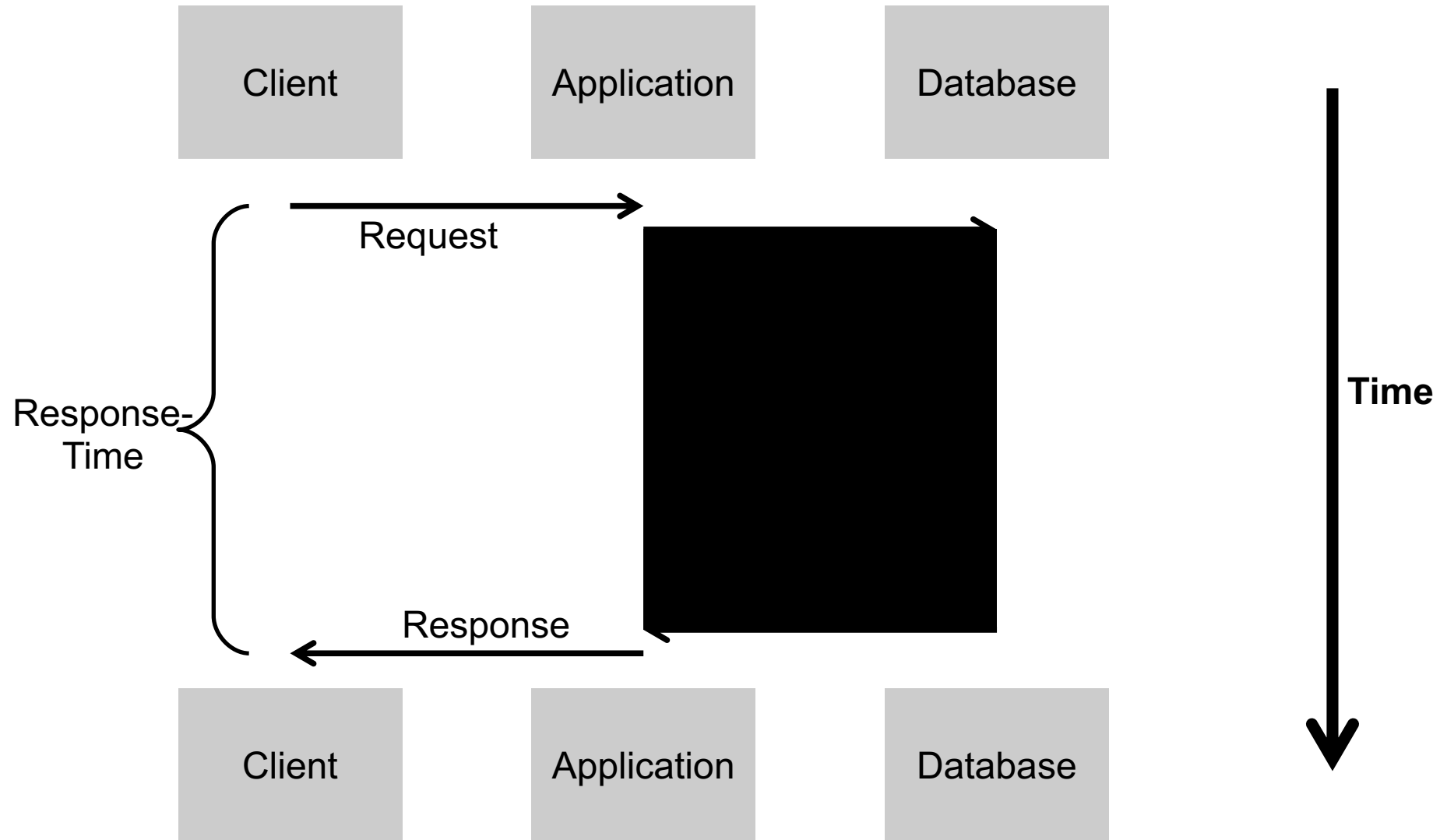
# Analysis

# UML Sequence Diagram





# UML Sequence Diagram



# Profiling

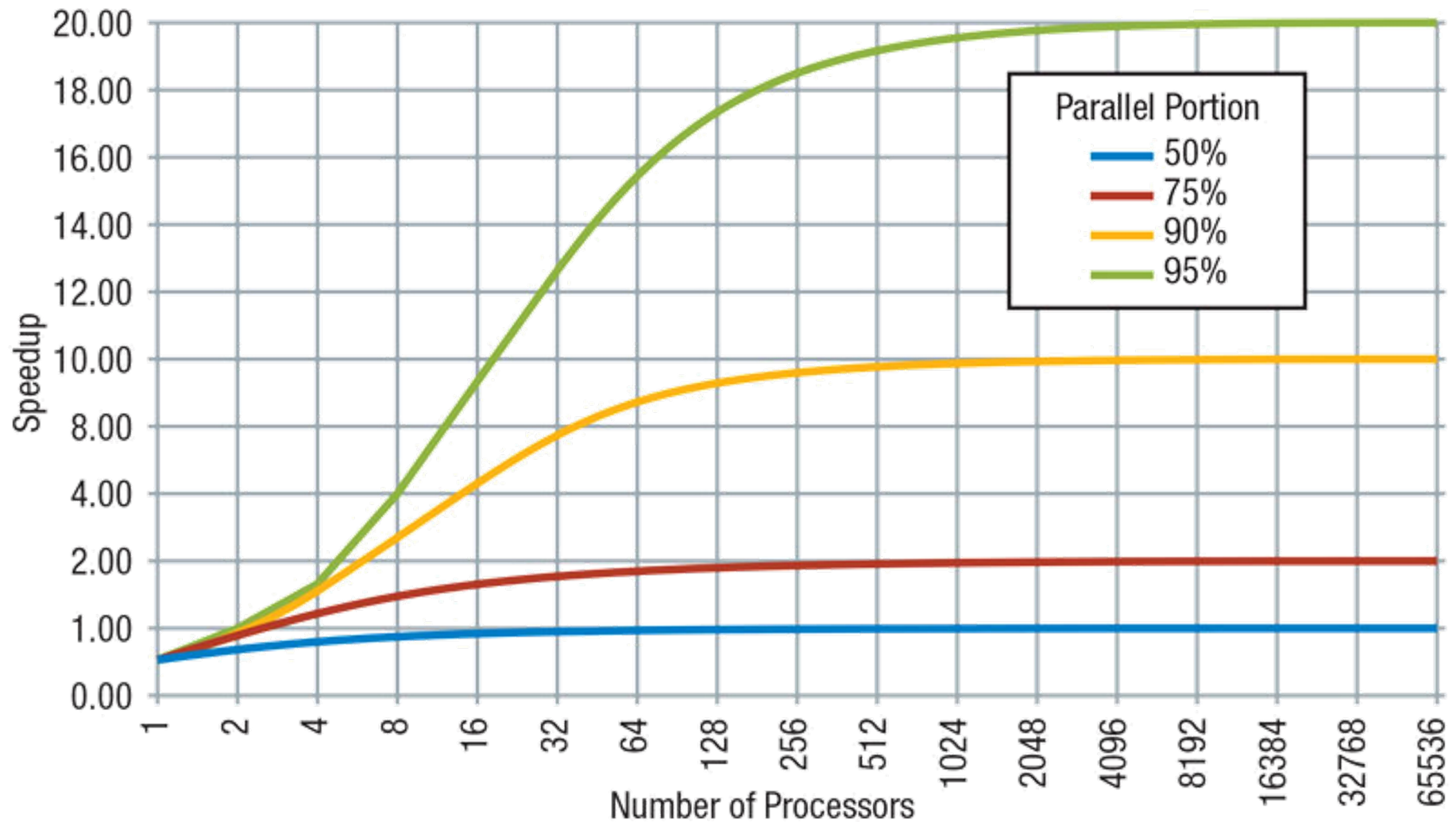
# Profile

Function	R	#
Frontend Page	1700s	35.000
Frontend DB-Access	300s	2.500.000
Background- Process XML Processing	100s	100.000
Background- Process DB-Access	97s	34.000

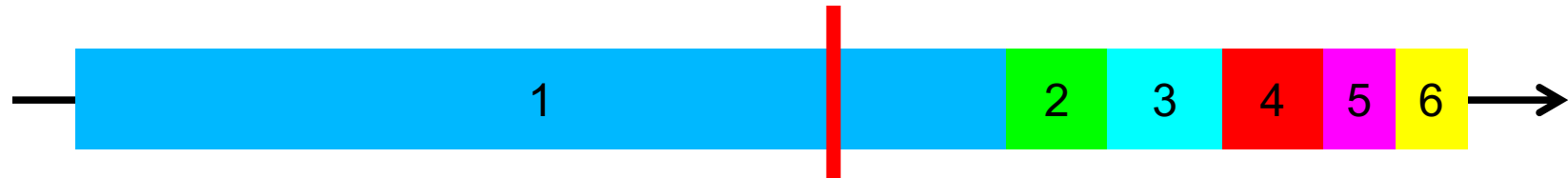
**Is the requested time REALISTIC?**

# Amdahl's Law

# Amdahl's Law - Special



# Upfront Classification



#	Poss. Improvement	Cost	R	R%
1	34%	€€€€	1.700s	70%
2	12 %	€	300s	13%
3	none	-	100s	6%
4	4 %	€	97s	4%
5	0,1%	€€€€	58s	3%
6	1,6%	€	48s	2%
...	...	...	...	...

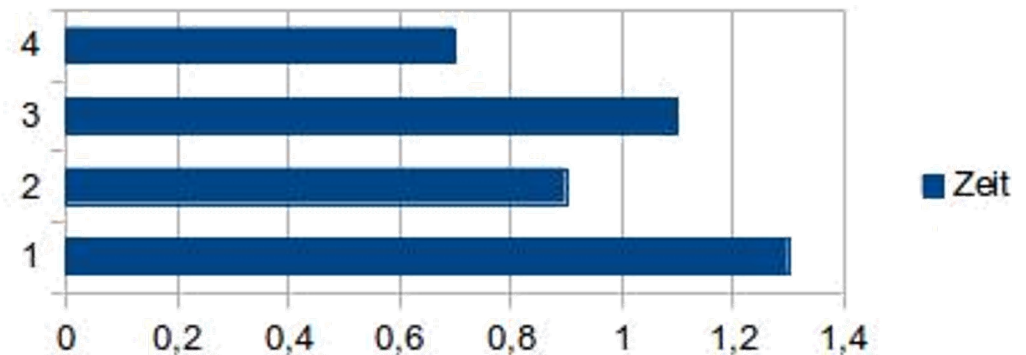
# Skew is Mean

# Catching Runaways (Skew!)

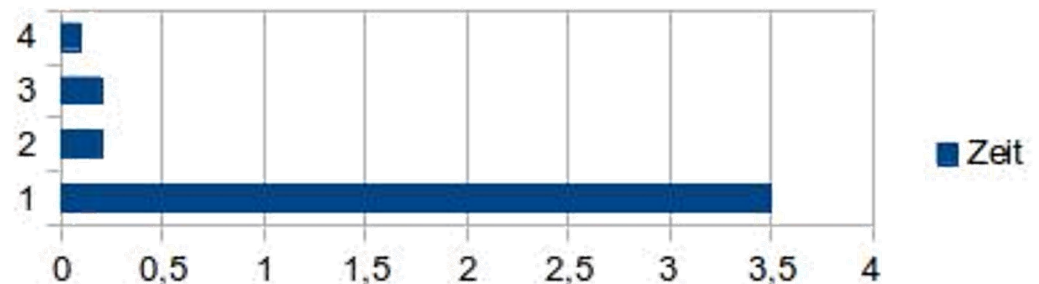
4 calls = 4 sec

2 calls = ? sec

Event	Duration [s]
4	0.7
3	1.1
2	0.9
1	1.3



Event	Duration [s]
4	0.1
3	0.2
2	0.2
1	3.5





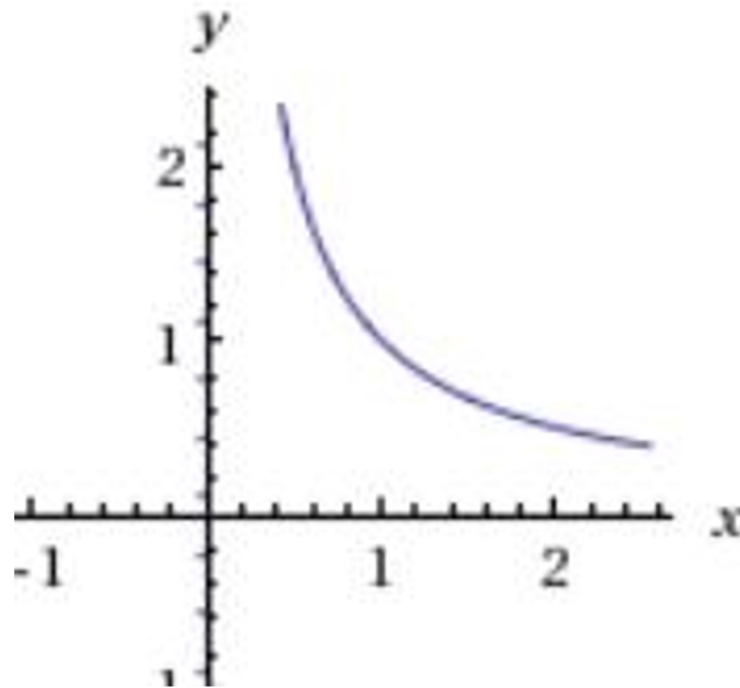
# Efficiency

Obviously, the highest type of efficiency is  
that which can utilize existing material  
to the best advantage

-- Jawaharlal Nehru

# Efficiency

$$\textit{efficiency} = \frac{1}{\textit{wastage}}$$

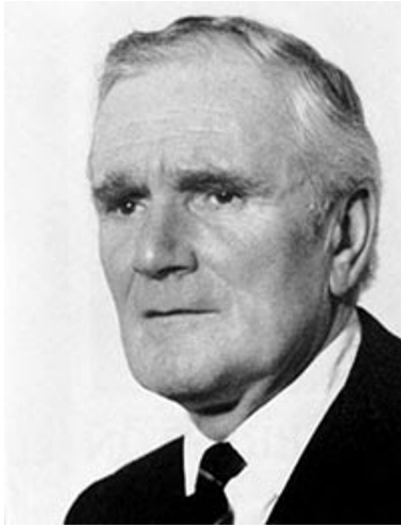


# Load

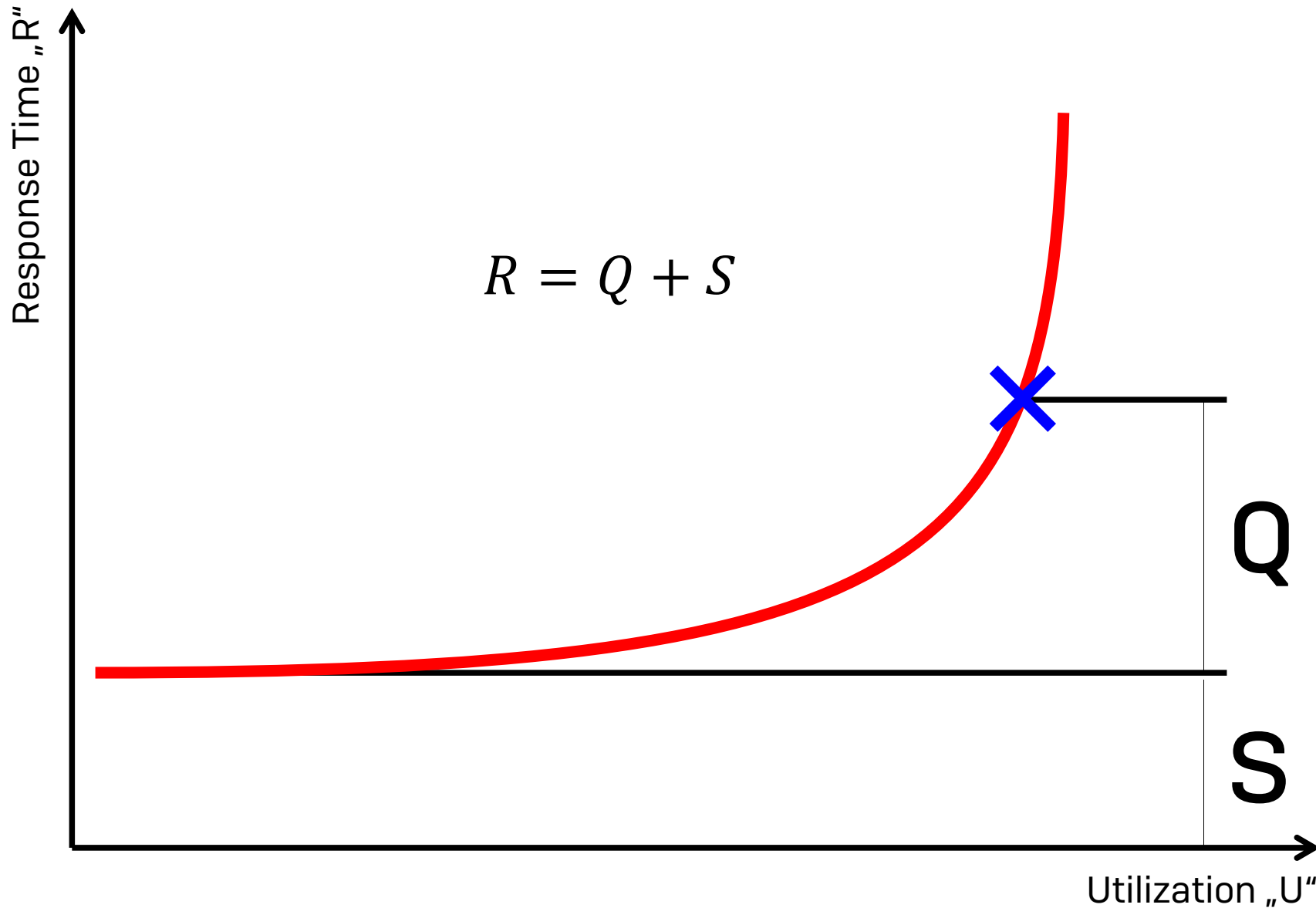
# What is “Q”?

Function	R	#	Type
Wait in line	0s - 600s	1	Q
Order menu incl. questions	30s	2	S
Fetching fries	45s	2	S
Filling drink	45s	2	S
Waiting for the Burger	0s-120s	1	Q
Fetching Burger	10s	2	S
...	...	...	...

# Evolution of “Q” 😊



# Queueing Theory – the Q Diagram



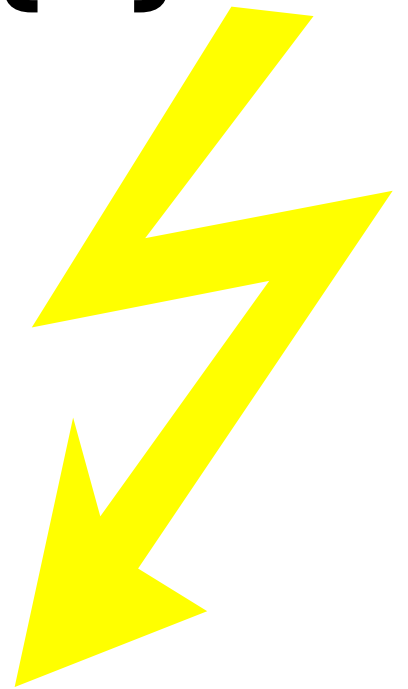
# The Elbow



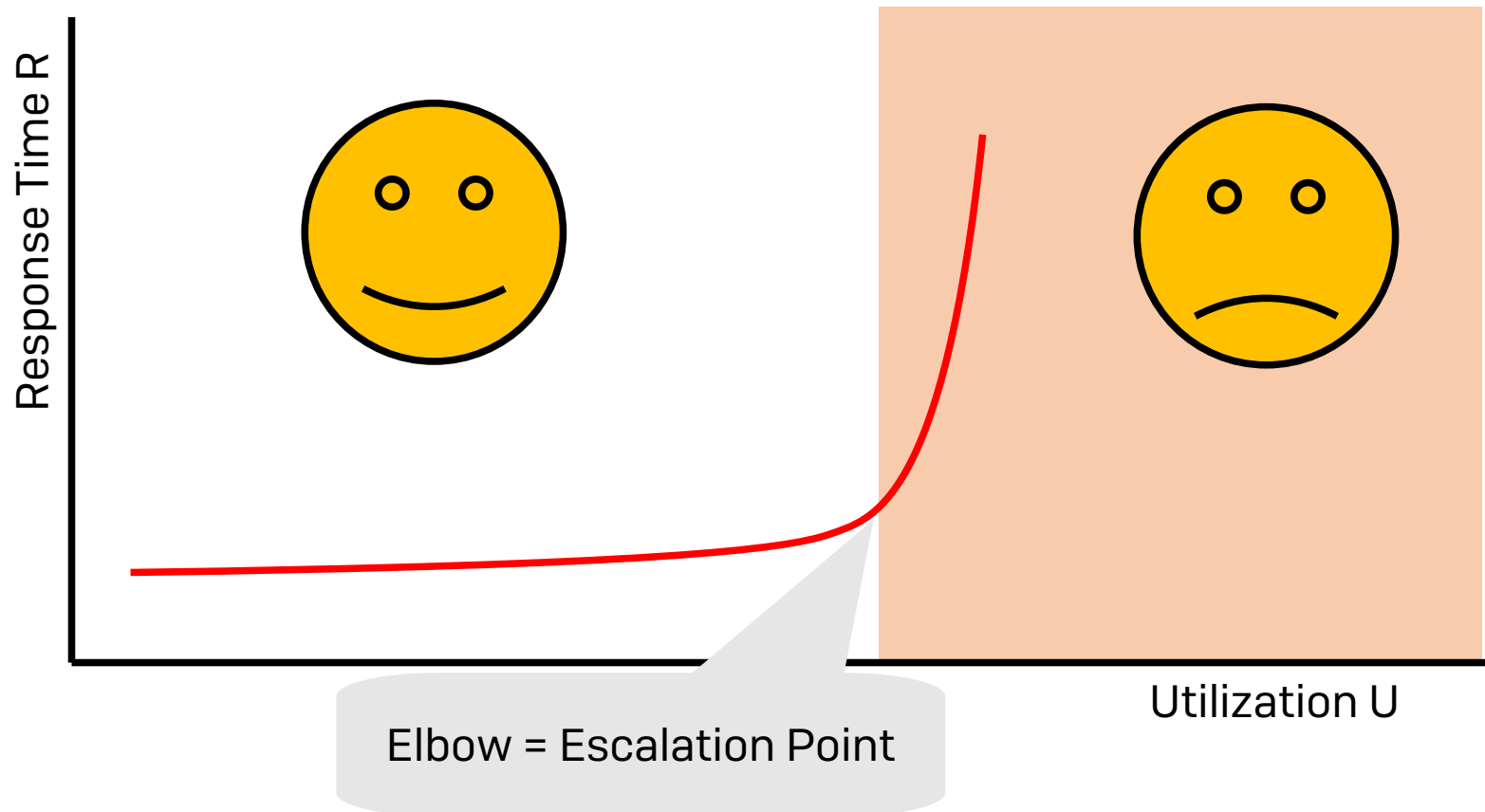
# Fast Response Times (R)

vs.

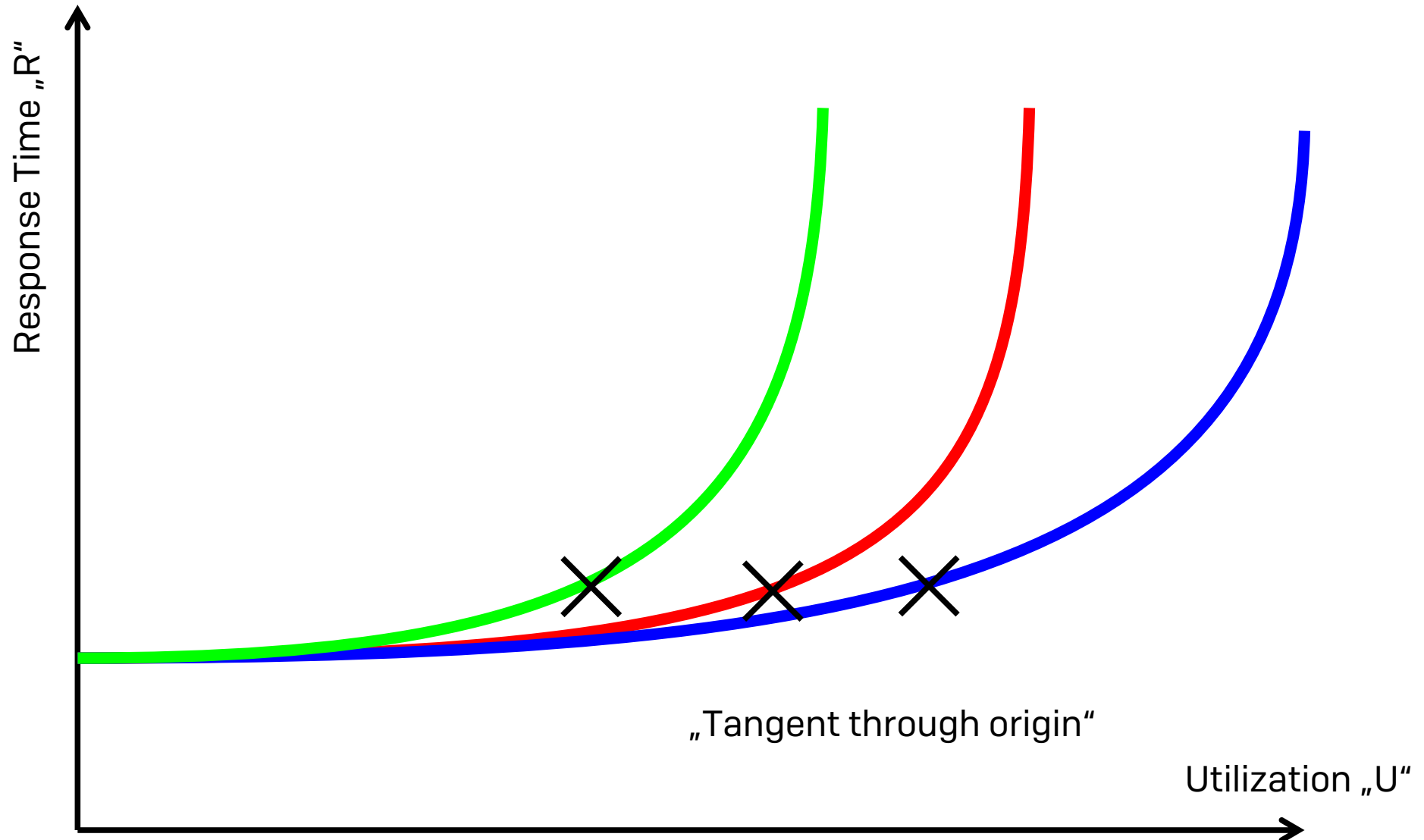
# High Throughput



# Relevance of the Elbow



# Where is the Elbow?

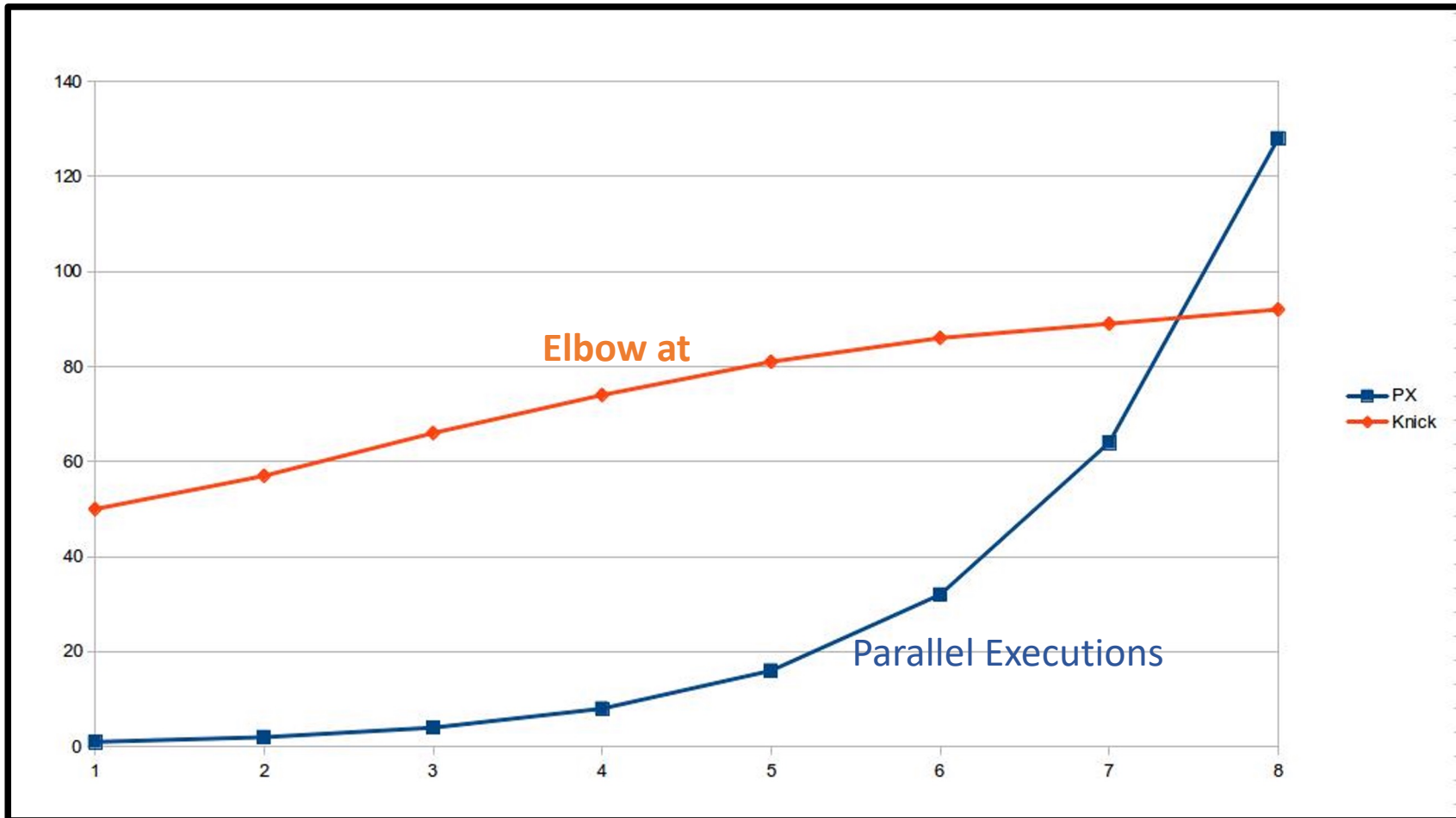


# (CPU) Scalability

Service channels / degree of parallelism	Elbow at % of overall utilization
1	50%
2	57%
4	66%
8	74%
16	81%
32	86%
64	89%
128	92%

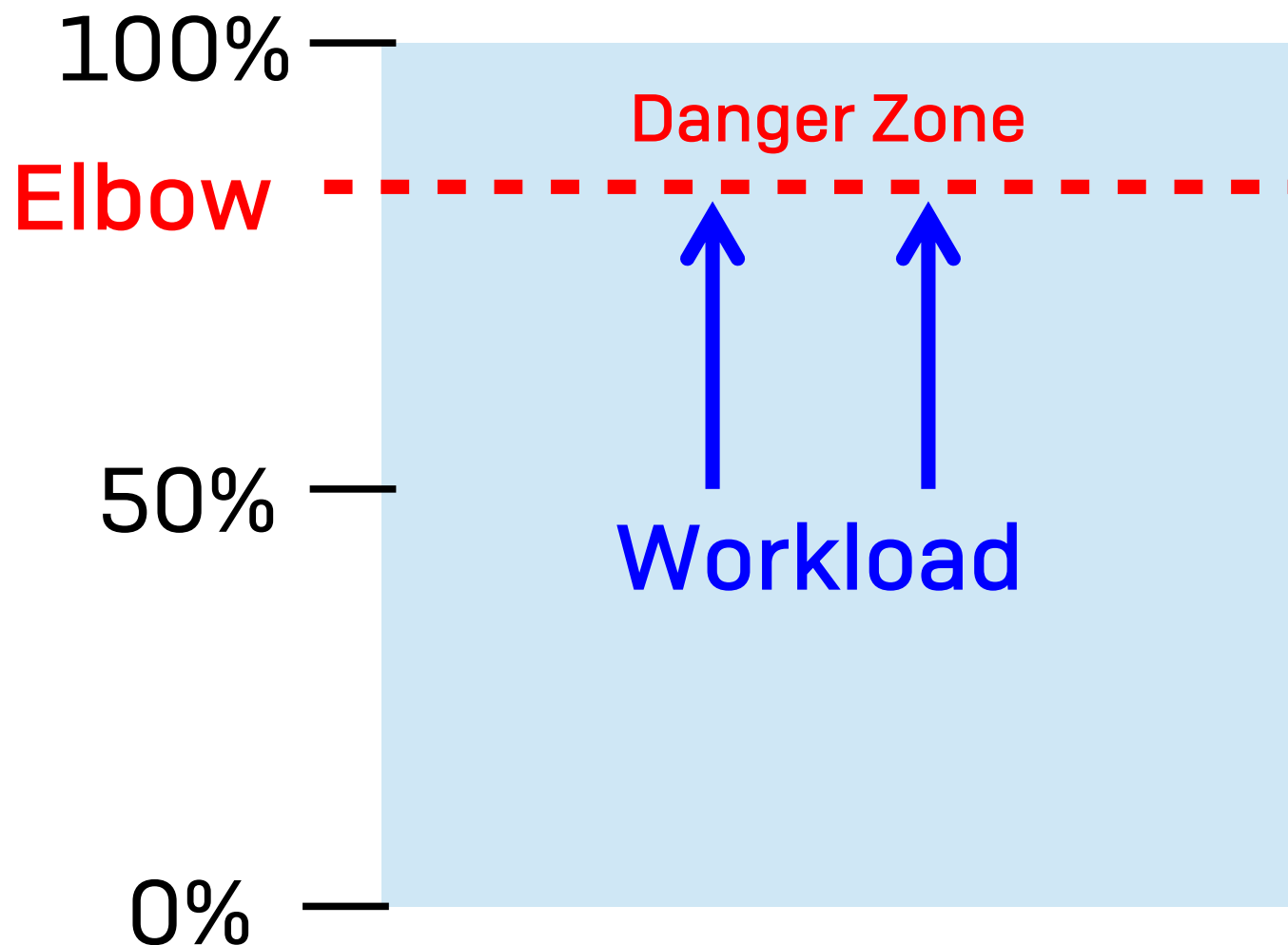
**=> The Elbow follows delayed**

# Scalability: Elbow follows delayed



# Capacity Planning

# Capacity Planning



# Capacity Planning

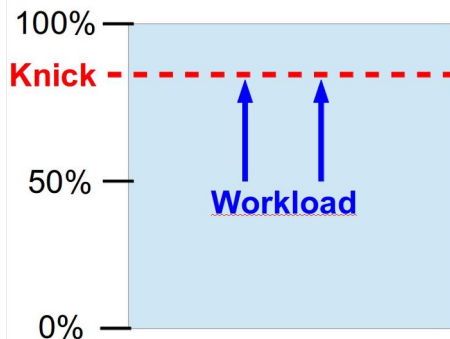
## Capacity planning

How big has it to be?

## Utilization management:

How small can it be?

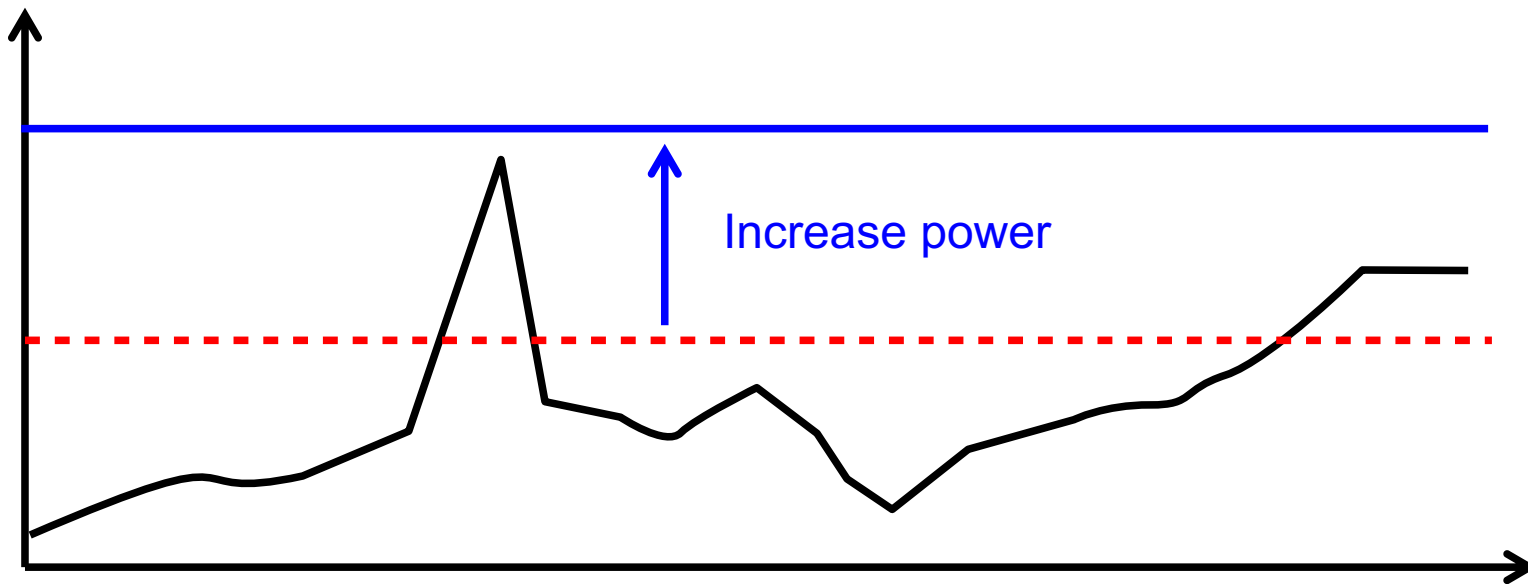
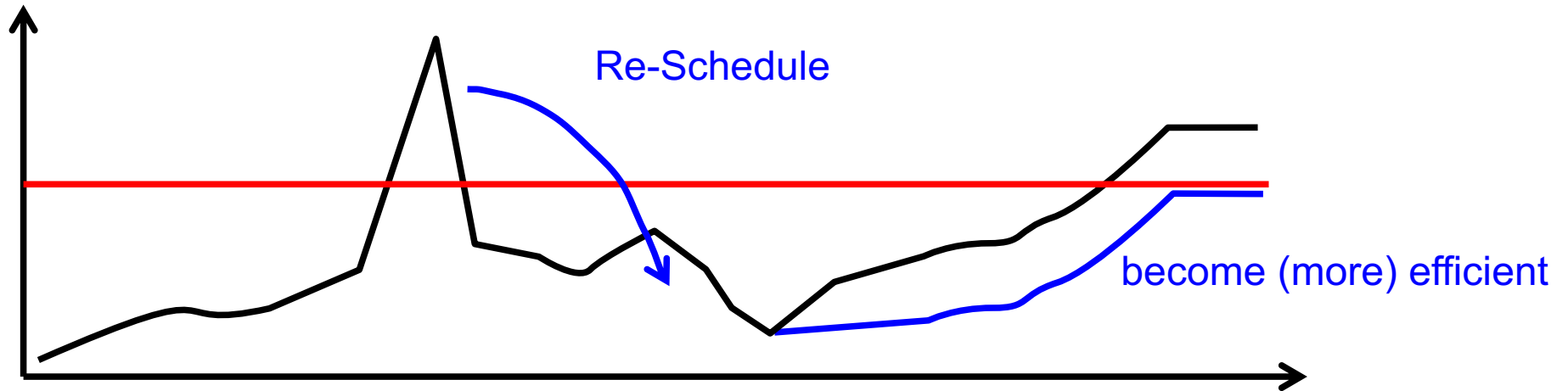
=> The elbow is the new 100%





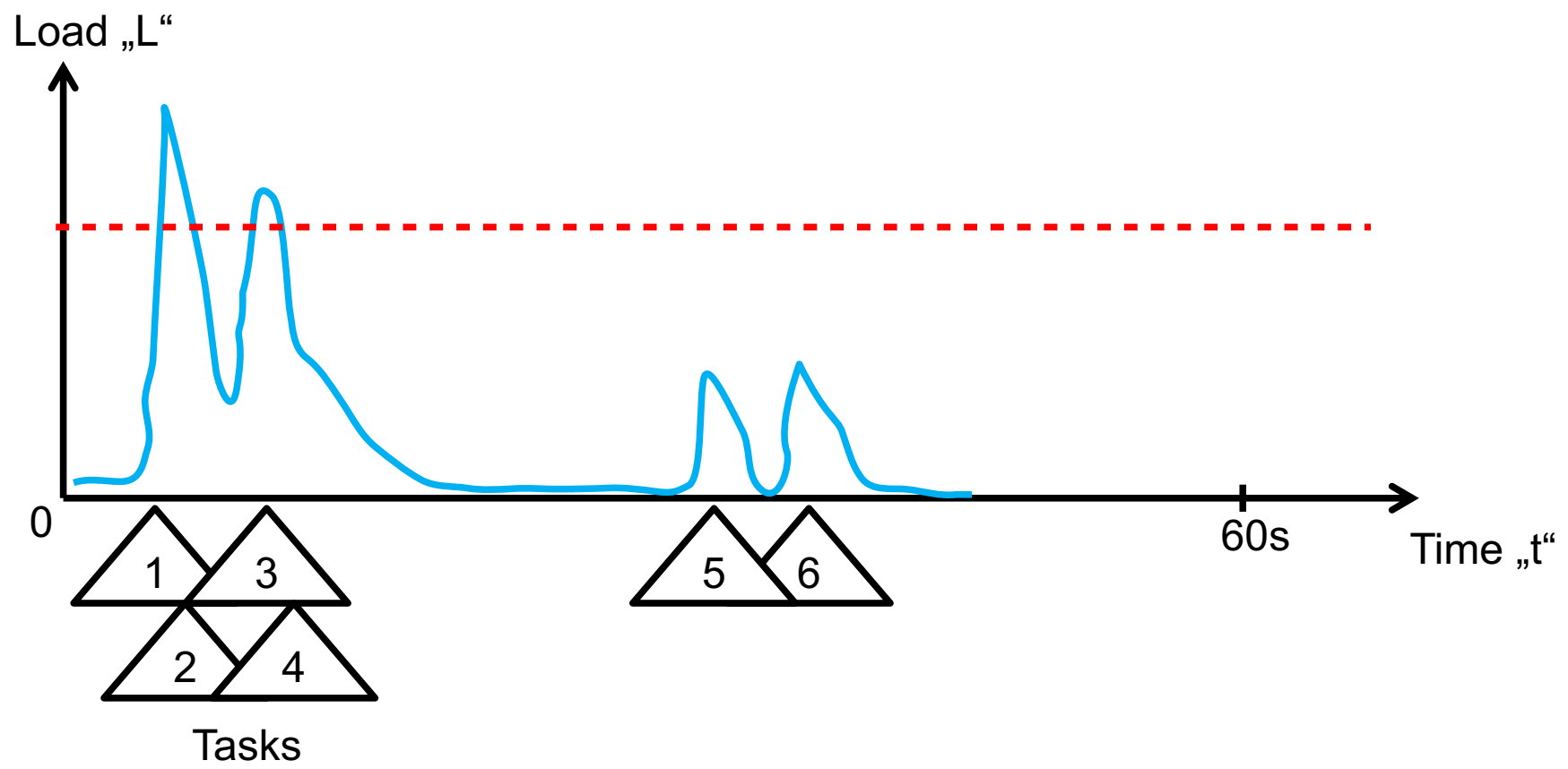
# Reducing Peaks

# Magic Peaks and where to find them

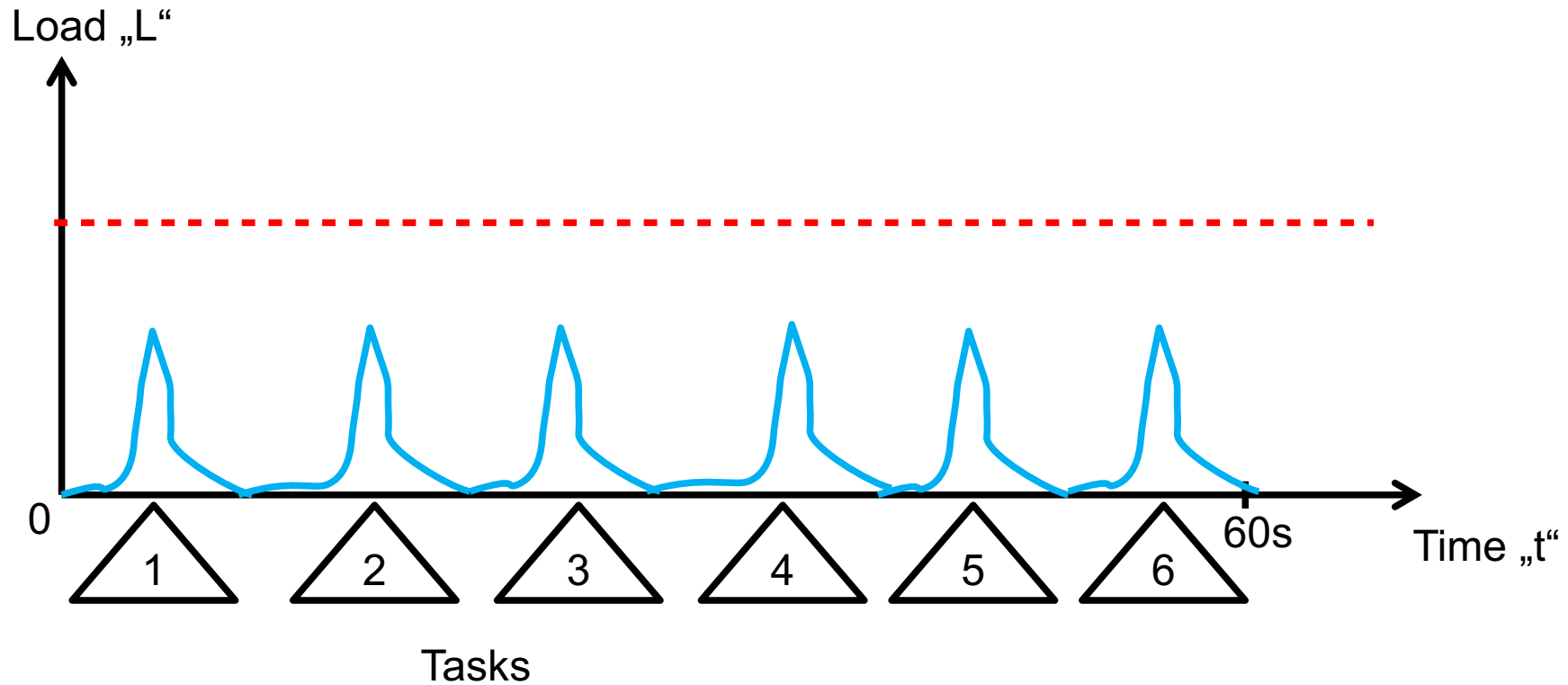


# Random Arrivals

# Random

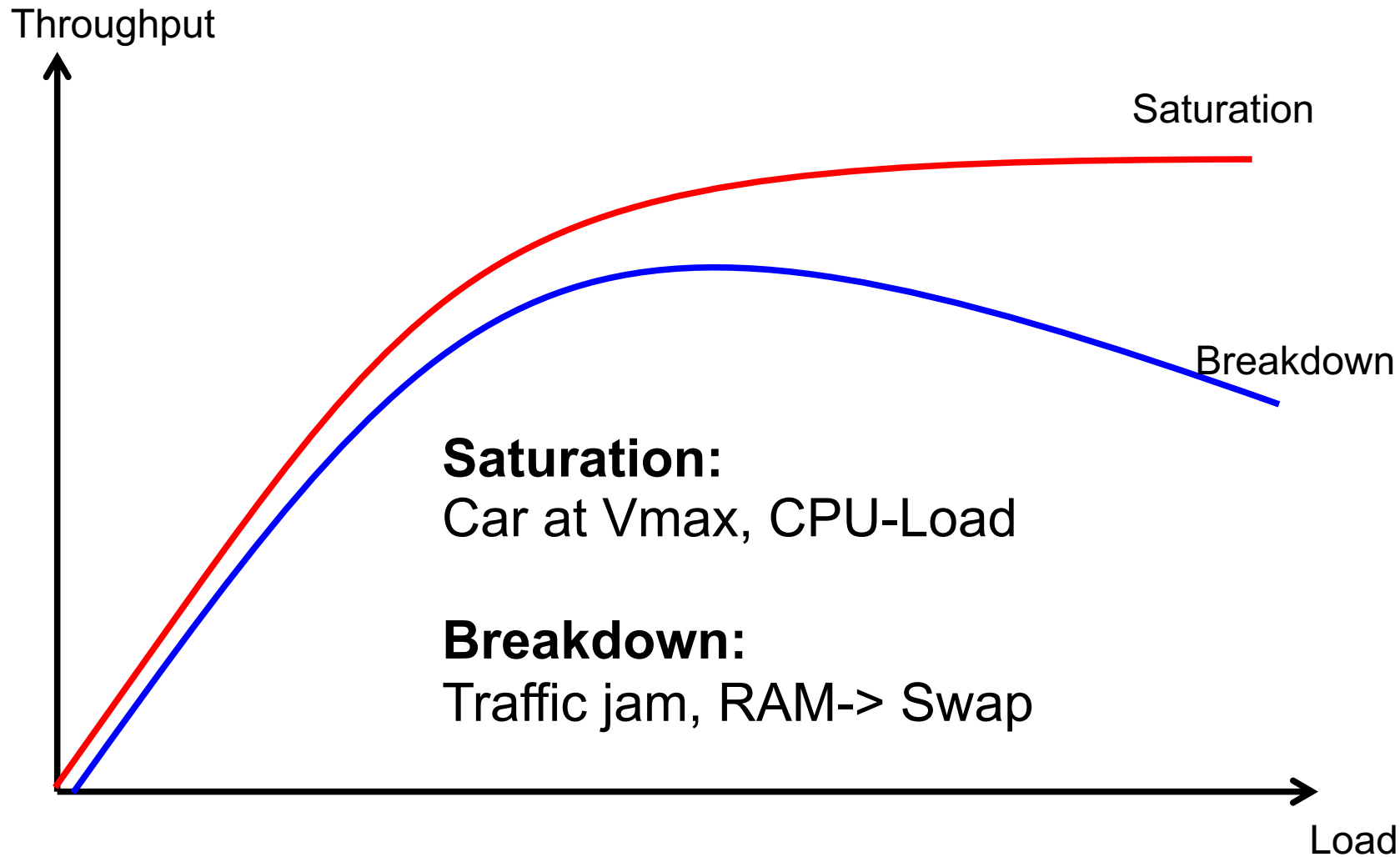


# Deterministic



# The Delay in the Bigger Picture

# Two-Class Society







# Performance Testing

# Performance Testing

**Just do it!**

You'll find more if you try ...

**You'll never find everything.**

# to measure

*/'tu: 'meɜə/*

# Measure ... what?

## Throughput!

and

## Response time!

# S.M.S.

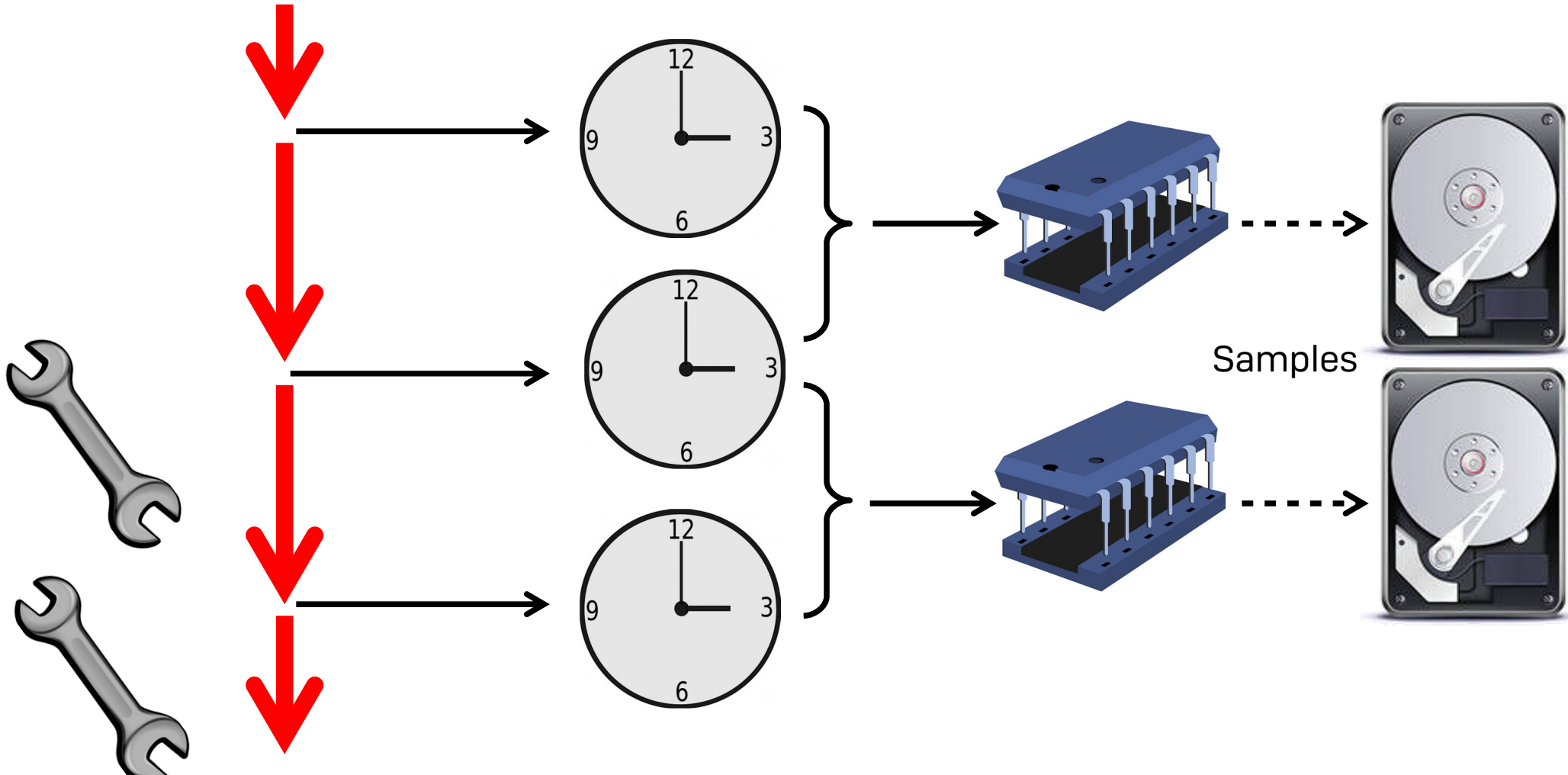
Seiner Majestät Schiff?  
Short Message Service?

# Surrogate

# Measure

# Sucks

# Code Instrumentation



Measure everything - Reduce data

# Performance is a Feature

# Performance is a feature

**Planning <- Cost**

**Implement -> Use**

Specify in same detail

as you do with other features



# Overhead

- 30%

By qualified Instrumentation (=Feedback) you'll improve quality.  
This leverages the additional cost by far.

# (More) Ressource Consumption

# Makes you (more) vulnerable

# Meet & Greet

[martin.klier@performing-db.com](mailto:martin.klier@performing-db.com)

[www.performing-databases.com](http://www.performing-databases.com)

Many national // international events

# DOAG

Deutsche ORACLE-Anwendergruppe e.V.

DOAG Database Conference Düsseldorf

May 15-16, 2024

CALL FOR PAPERS OPEN till Feb 7



UTOUG Training Days

March 21-22, 2024

Salt Lake City, UT, USA



AIOUG OCYatra Replay Webinar

March 15, 2024

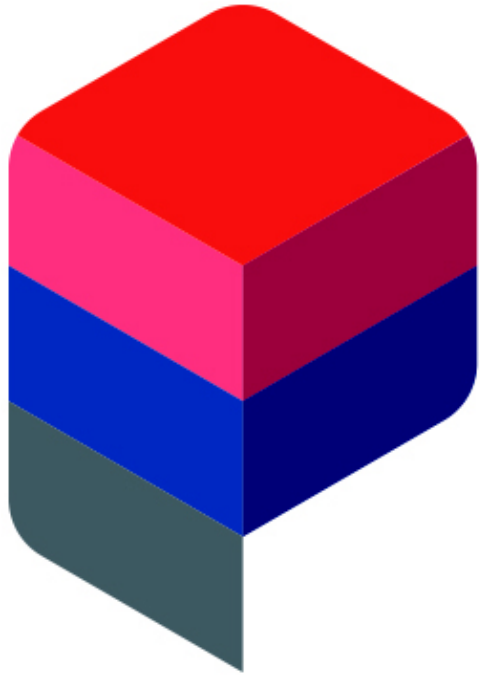


performing  
databases

Q & A



Download my Presentations and Whitepapers  
<http://www.performing-databases.com/>



**performing  
databases**