# performing databases

Your reliability. Our concern.



## Performance is rarely an accident

Martin Klier Performing Databases GmbH



#### **Martin Klier**



- Managing Partner / Database Technology Performing Databases GmbH
- Focus on Performance and High Availability
- Linux 1997
- Oracle 2003



- Email: martin.klier@performing-db.com
- Weblog: http://www.usn-it.de

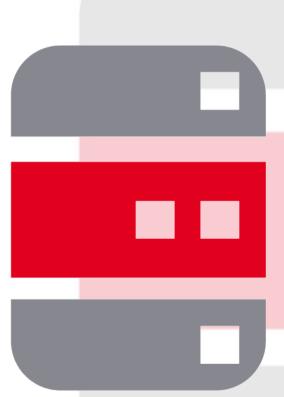




#### performing databases GmbH



- Experts for Database Technology
  - Consulting for concepts and sourcing
  - Plannung support = optimal architecture
  - Licensing and system design
  - Implementation, changes, troubleshooting
- Get in touch
  - Performing Databases GmbH
     Wiesauer Straße 27
     95666 Mitterteich, Germany
  - Web: http://www.performing-databases.com
  - Twitter: @PerformingDB





## Excellence



## Performance is rarely an accident



## Know it counts

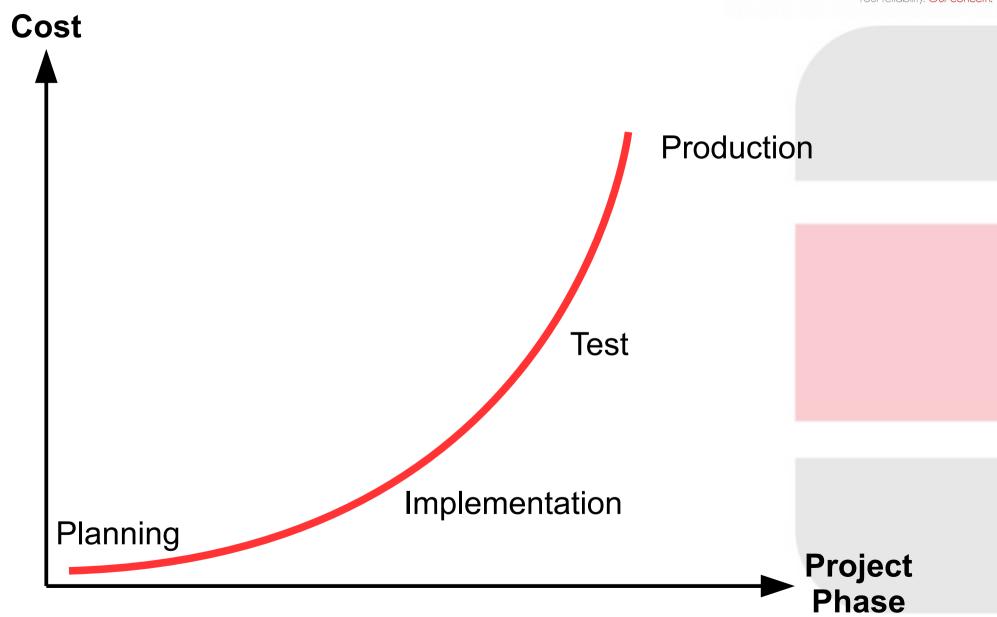
## Prove it counts more



## Boehm

#### **Costs of a Change**







## We need a plan ...



## Performance

#### **Performance**





#### **Performance**



### Work per Time

Kilometers per hour, tons per month ...

### Time per work unit

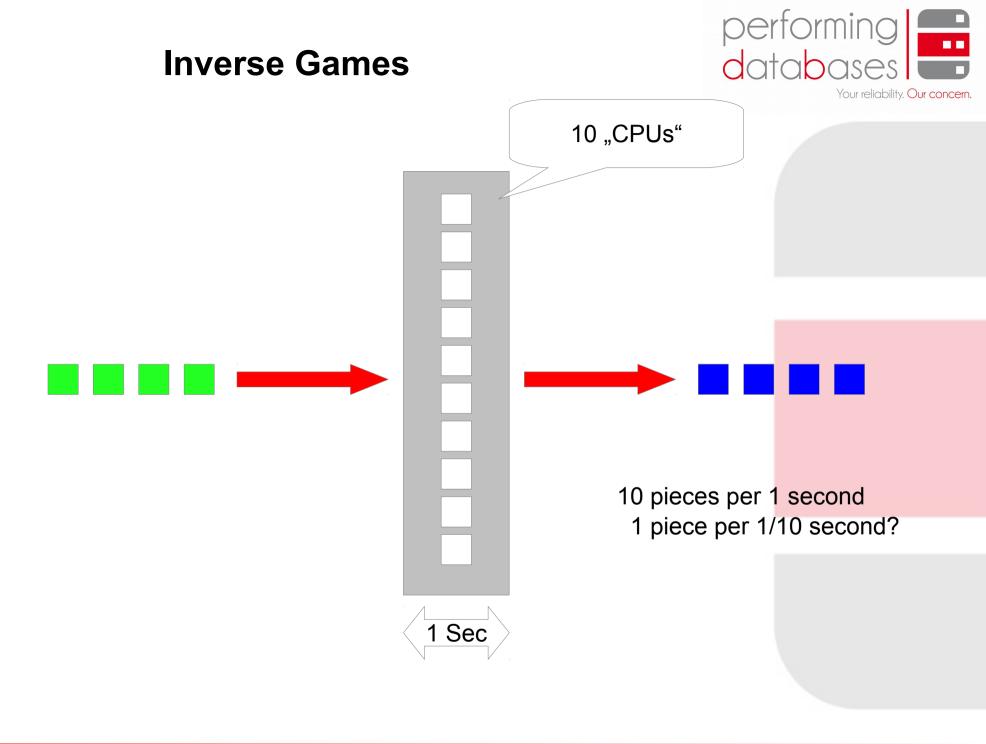
Seconds per booking, days per project



Throughput: 
$$p = \frac{task}{time}$$

Response time: 
$$R = \frac{time}{task}$$

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





## The Mean

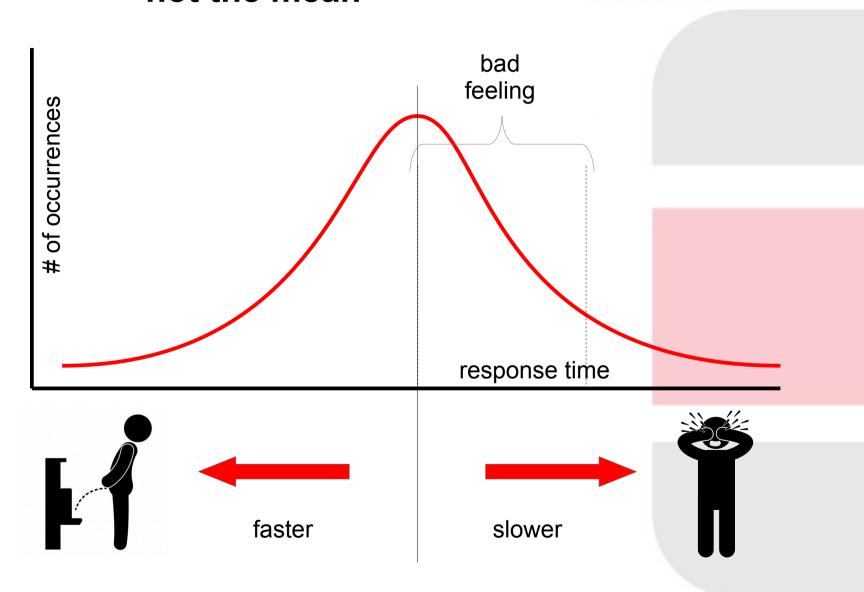
#### Of the mean ...



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

### Users feel the difference - not the mean



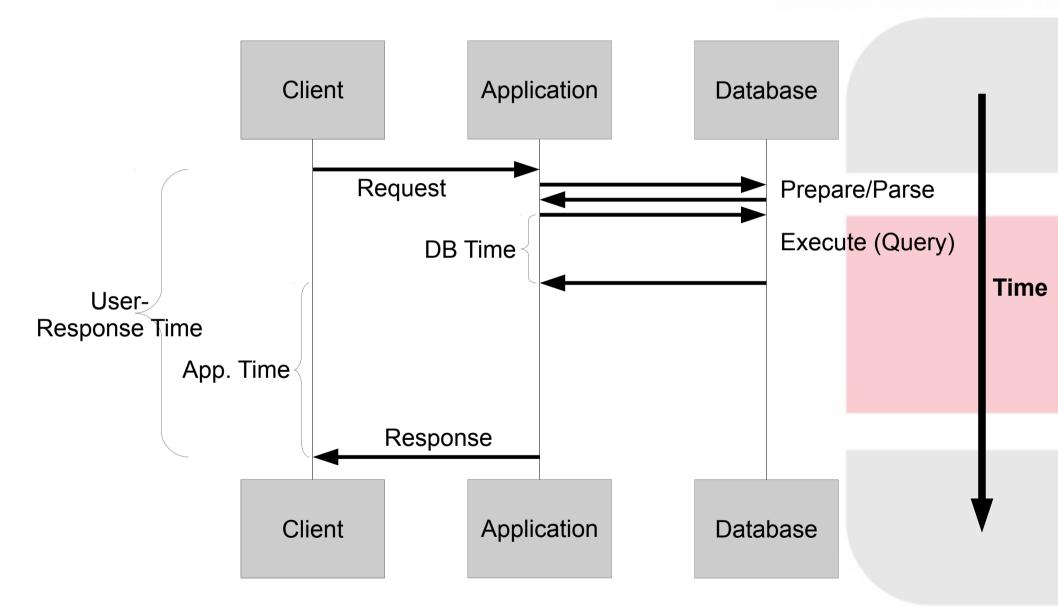




## Analysis

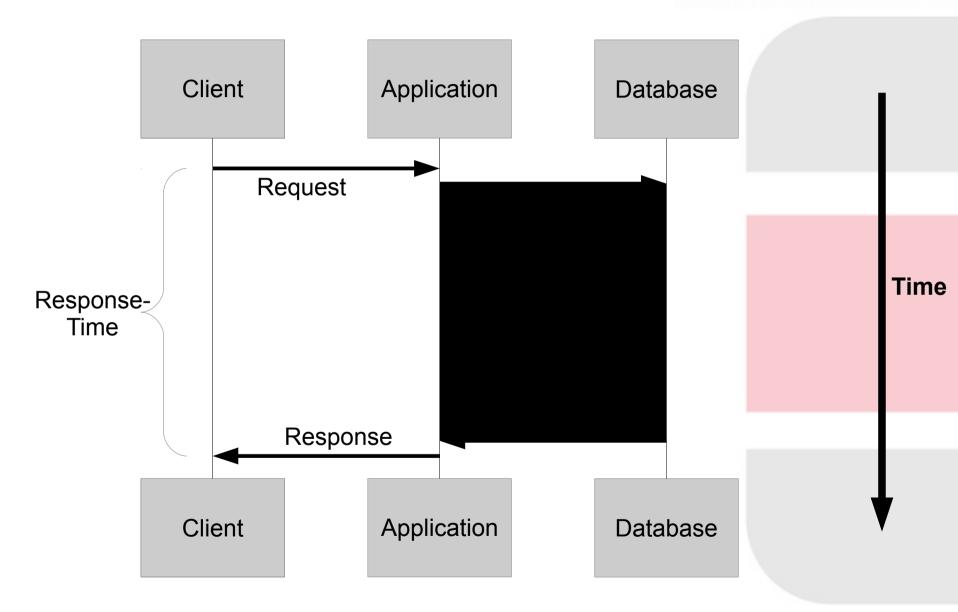
#### **Sequence Diagram (UML)**





#### **Sequence Diagram (UML)**







## 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

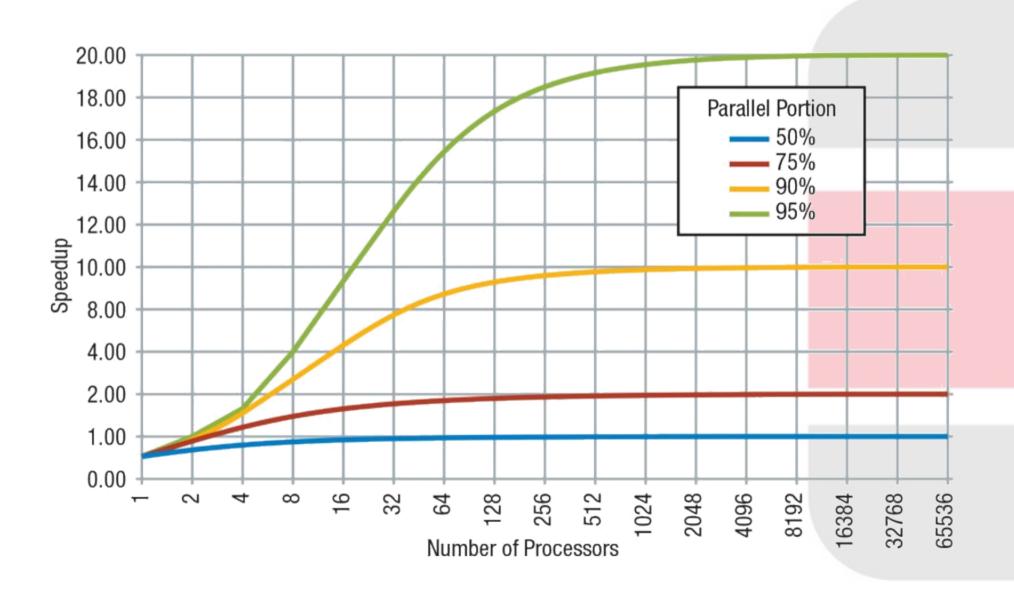
#### Is the requested time REALISTIC?



## Amdahl's Law

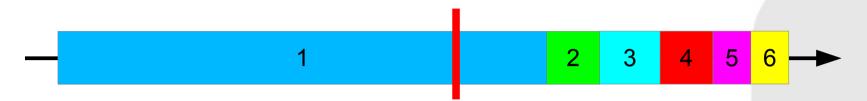
#### **Amdahl's Law - Special**





#### **Amdahl's Law - General**





#	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

#### **Catching runaways**

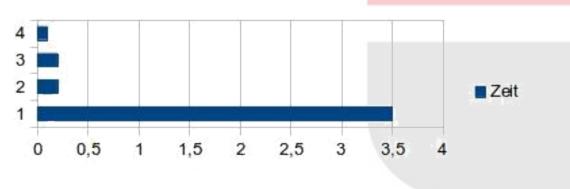


### 4 calls = 4 sec2 calls = ? sec

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	





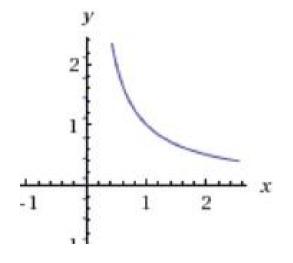


## Efficiency

#### **Efficiency**



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





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

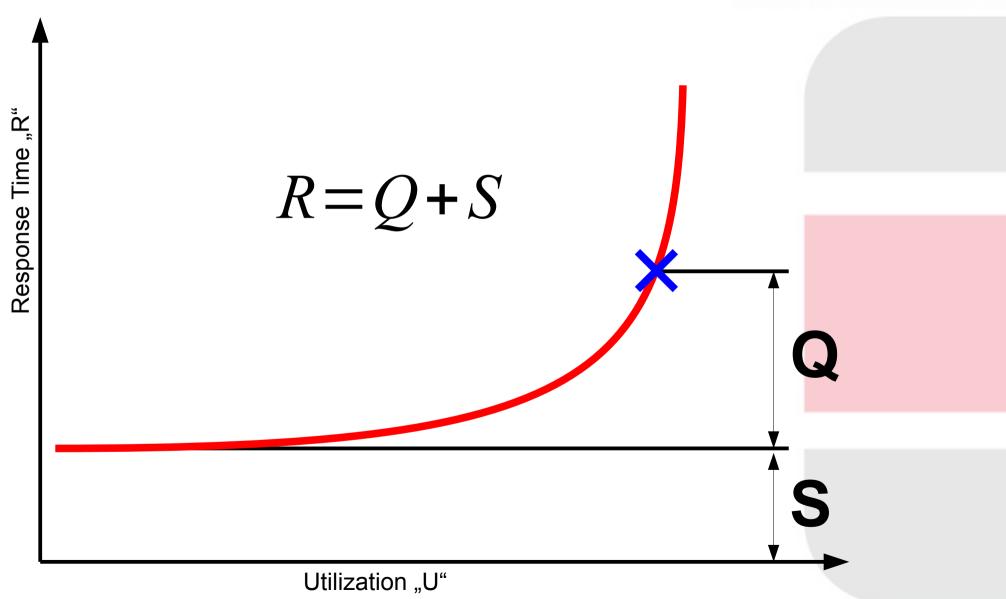
-- Jawaharlal Nehru



## Load

#### "Q" - Diagram







## The Elbow

#### The Elbow



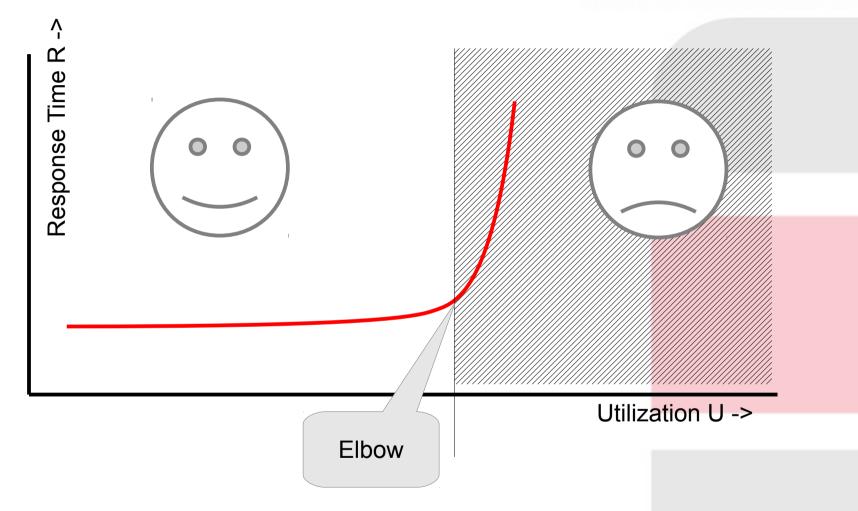
### Fast Response Times (R)

VS.

**High Throughput** 

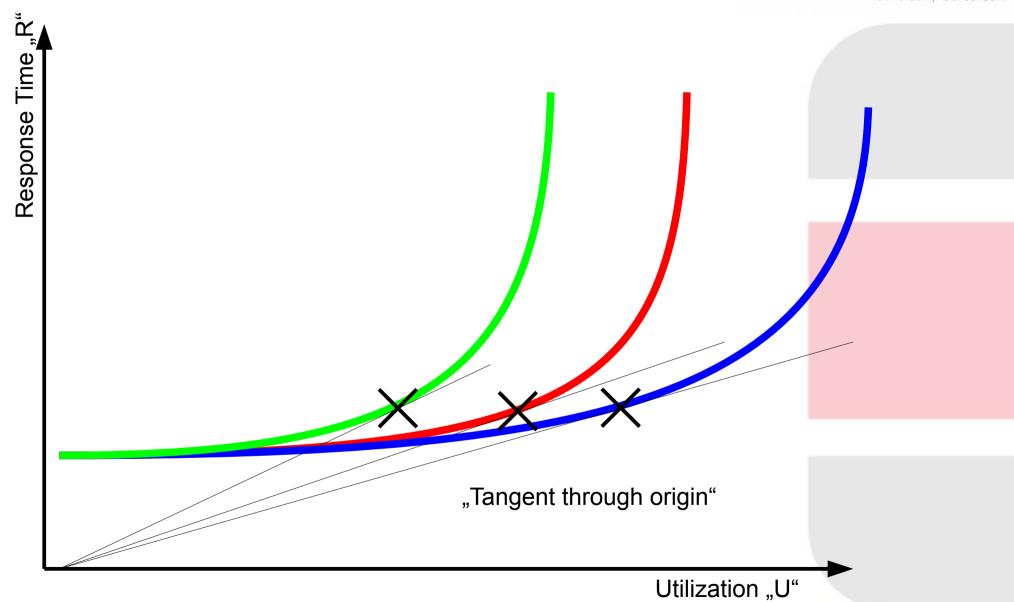
#### Relevance of the elbow





#### Where is the elbow?





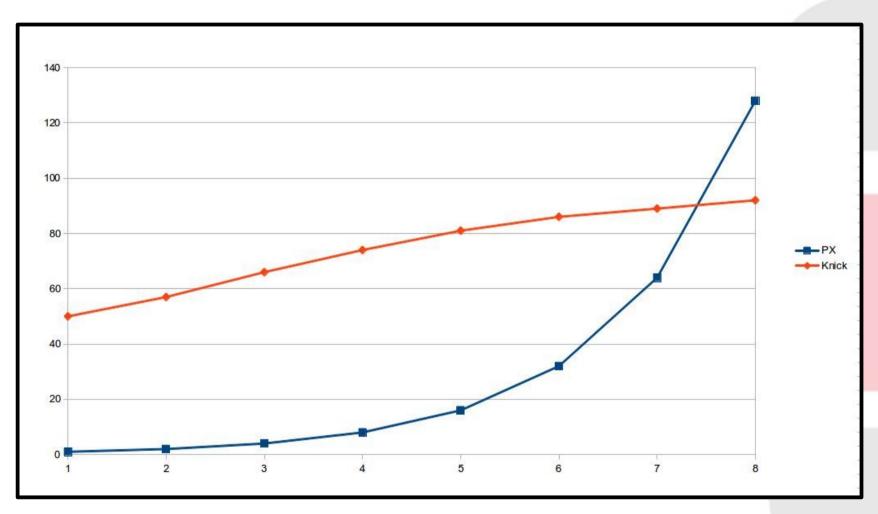


## Scalability: Elbow follows delayed

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%

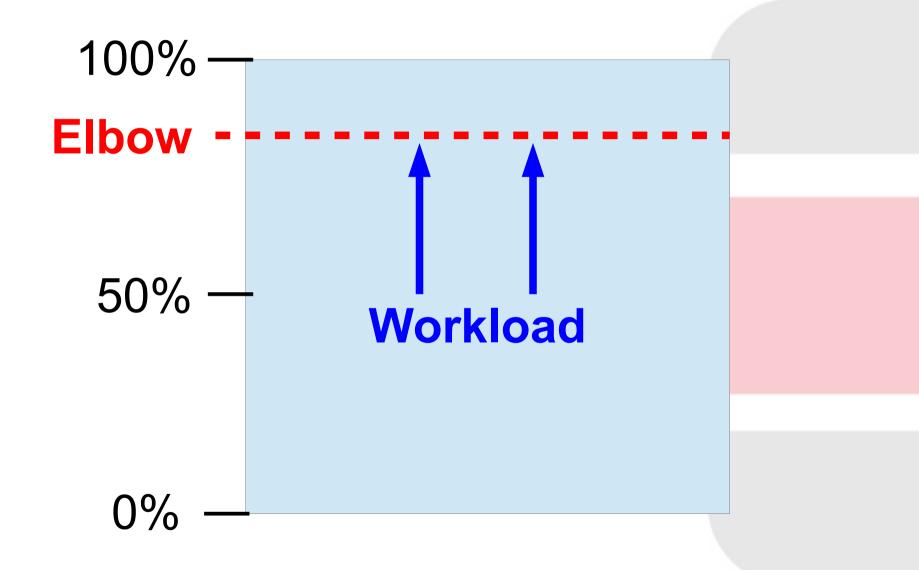






#### **Capacity planning**



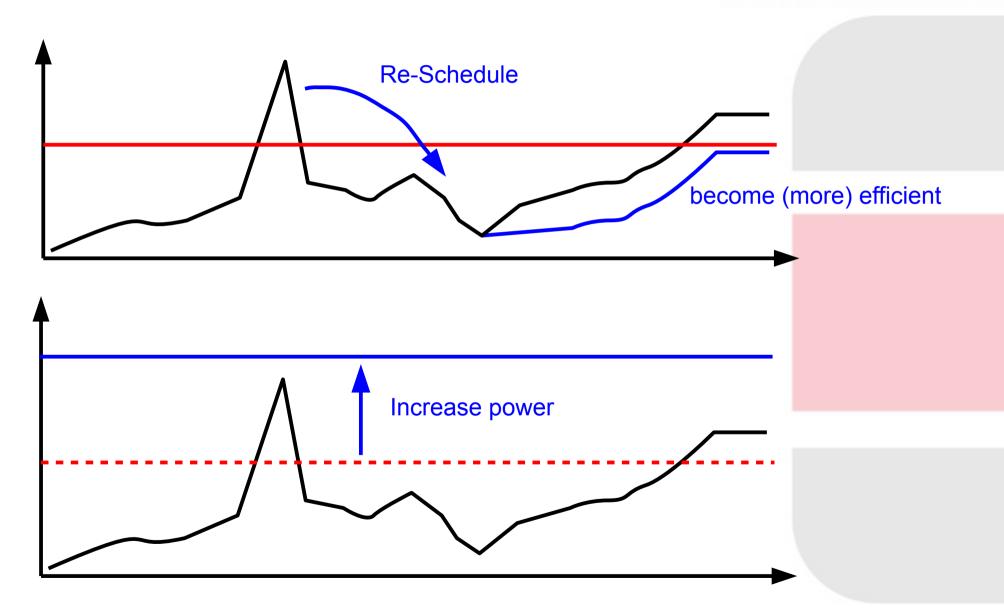




## Reducing Peaks

#### **Peaks**



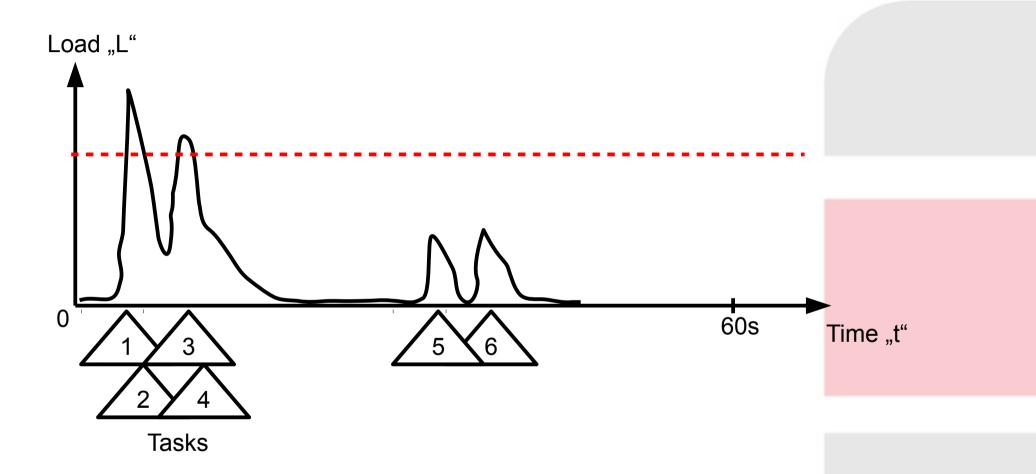




### Random Arrivals

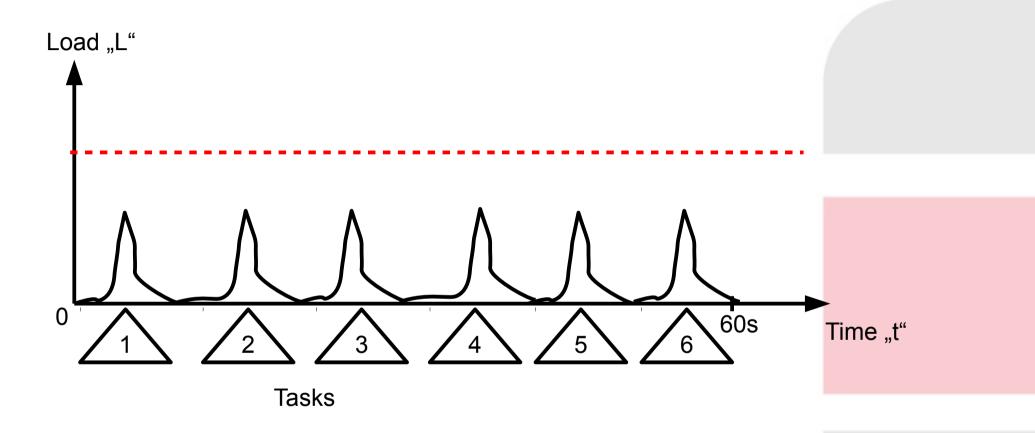
#### Random





#### **Deterministic**



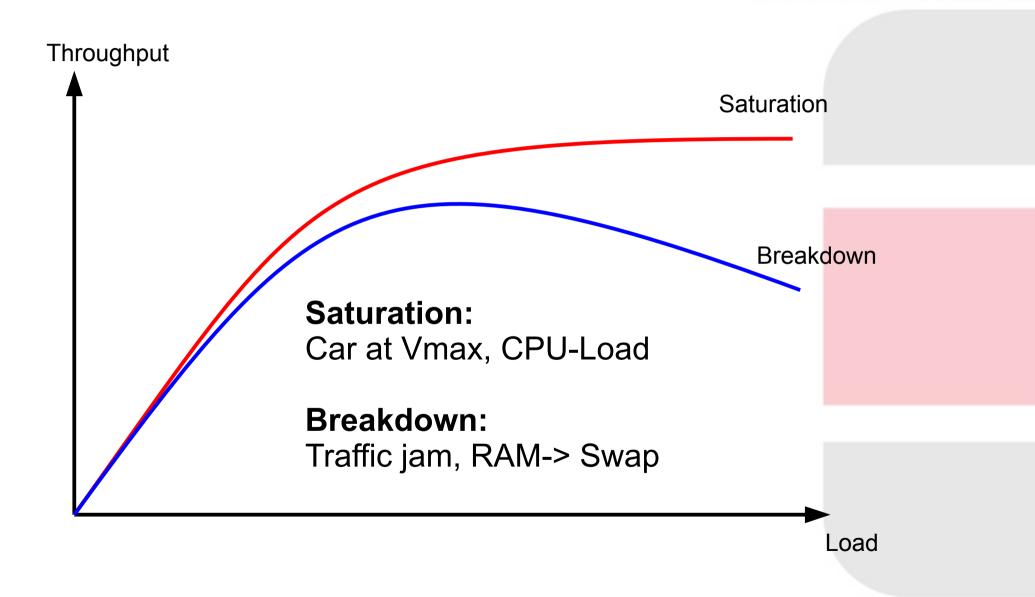




## The delay in the bigger picture

#### **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: 'meʒə/

#### Measure what?



# Throughput! and Response time!



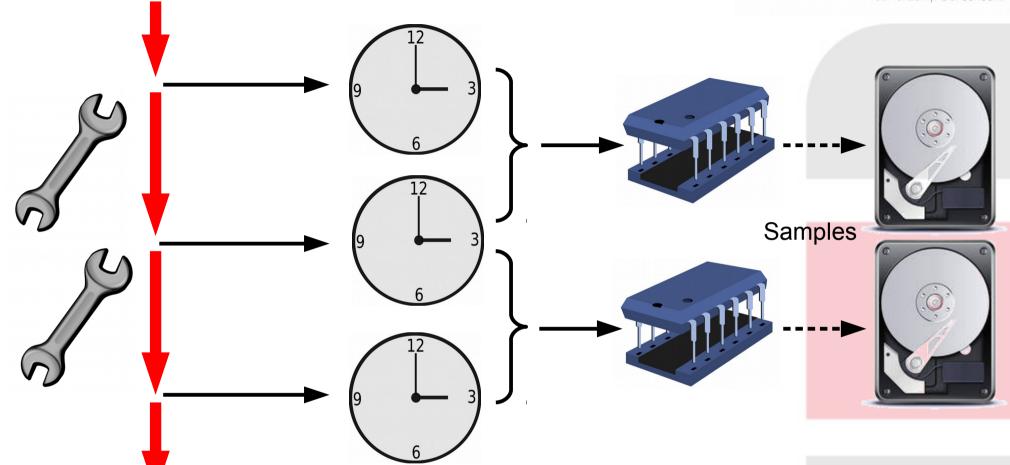
S.M.S.

Seiner Majestät Schiff? Short Message Service?

# Surrogate Measure Sucks

#### **Measure everything**





## Code Instrumentation



## 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.





## Thank you very much!

### Performance



is rarely an accident



Download: http://www.performing-databases.com

# performing databases

Your reliability. Our concern.