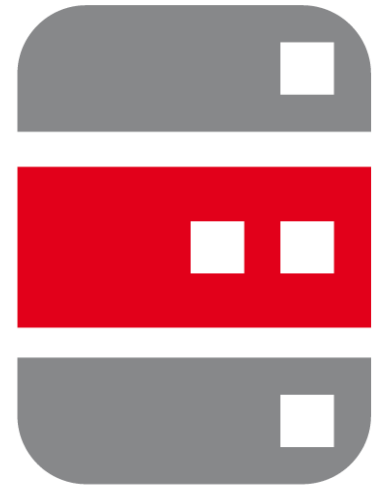


performing  
databases



Your reliability. Our concern.

# Maschinensizing

Martin Klier

Performing Databases GmbH  
Mitterteich

**DOAG**



# Referent

- Martin Klier
- Lösungsarchitekt und Datenbankspezialist
- Fachliche Schwerpunkte:
  - Performanceoptimierung / Tuning
  - hochverfügbare Systeme
  - Cluster und Replikation
- Linux seit 1997
- Oracle Database seit 2003

**ORACLE**<sup>®</sup>

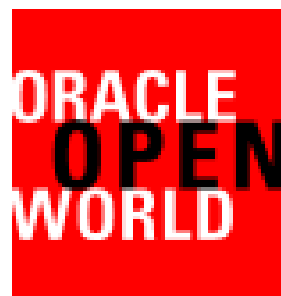


# Referent

- Vorträge



COLLABORATE12  
TECHNOLOGY AND APPLICATIONS FORUM  
FOR THE ORACLE COMMUNITY



COLLABORATE



- Kontakt: [martin.klier@performing-db.com](mailto:martin.klier@performing-db.com)
- Weblog: <http://www.usn-it.de>

# Unternehmen

- Spezialisten für Datenbanktechnik
  - Konzeptberatung und Vergabekompetenz
  - Architektur- und Systemplanung
  - Lizenzierung
  - Realisierung und Troubleshooting
- Kontakt
  - Performing Databases GmbH  
Wiesauer Straße 27  
95666 Mitterteich
  - Web: <http://www.performing-databases.com>
  - Mail: [info@performing-db.com](mailto:info@performing-db.com)
  - Twitter: @PerformingDB

**„Software ist ein Gas:**  
Sie füllt den zur Verfügung stehenden Raum  
vollständig aus.“

*(E.J., SW-Entwicklungsleiter)*

# So what?



+ €€€€ =



# So what?



- €€€€ =





„Die auf einen Tagesordnungspunkt verwendete Zeit ist umgekehrt proportional zu den jeweiligen Kosten.“

*(C.N. Parkinson, 1957)*

# Szenario

# Anforderungen I

Der neue DB-Server:

- (Ich weiß noch nicht was ich brauche, also:)  
Kann alles
- Kostet wenig
- Ist 100% ausfallsicher
- 10 Jahre investitionssicher

MfG, Ihre Fachabteilung

# Anforderungen II

Der neue DB-Server:

- Hardwarestandard: Doublemicro FunWare 19" 2HE
- Betriebssystem Maubux Version 7.3.223
- 4 Jahre Gewährleistung
- 2 CPUs / 12 Cores
- 2 HDDs 300GB 15.000rpm
- 2 NICs 1Gbit Fabrikat X Typ Y
- 1 FC HBA Fabrikat X Typ Y
- Storage über FibreChannel: 5 EUR / GB
- Abweichung nicht zulässig, sonst kein Hosting/Support

MfG, Ihr Rechenzentrum

# Anforderungen III

Der neue DB-Server:

- Oracle Database 11.2.0.2.8 ist Vorgabe  
11.2.0.3 evaluieren wir gerade,  
11.2.0.4 führen wir gar nicht mehr ein,  
wir gehen dann sowieso auf 12
- Enterprise Edition per ULA
- kein ASM (wg. Lizenz)
- kein Dataguard (kennen wir nicht)
- RAC würden wir unterstützen  
wir wollen in dem Bereich Know How aufbauen

MfG, Ihre Datenbanktechnik

# Architekt



Und noch hat nicht einmal jemand ermittelt, was wirklich NOTWENDIG ist ...

# Ressourcen

# Systemressourcen

- CPU
- RAM
- Massenspeicher
- Netzwerk





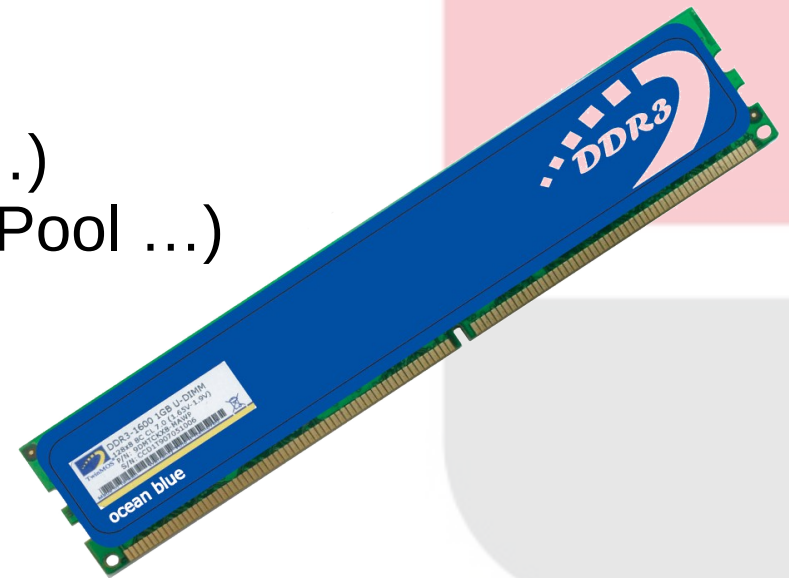
# CPU

- Relevant: „Leistung“
- Spezifikationen, gemessen in:
  - (Millionen) Instruktionen pro Sekunde (MIPs)
  - Anzahl Gleitkomma-Operationen (FLOPS)
  - Leerlauf-Maximaltakt (BogoMIPS)
  - NICHT: Taktrate (GHz)
- Für:
  - Speicherzugriffe
  - Hash/Sort/Merge
  - Berechnungen
  - Syscalls
  - => alles :)



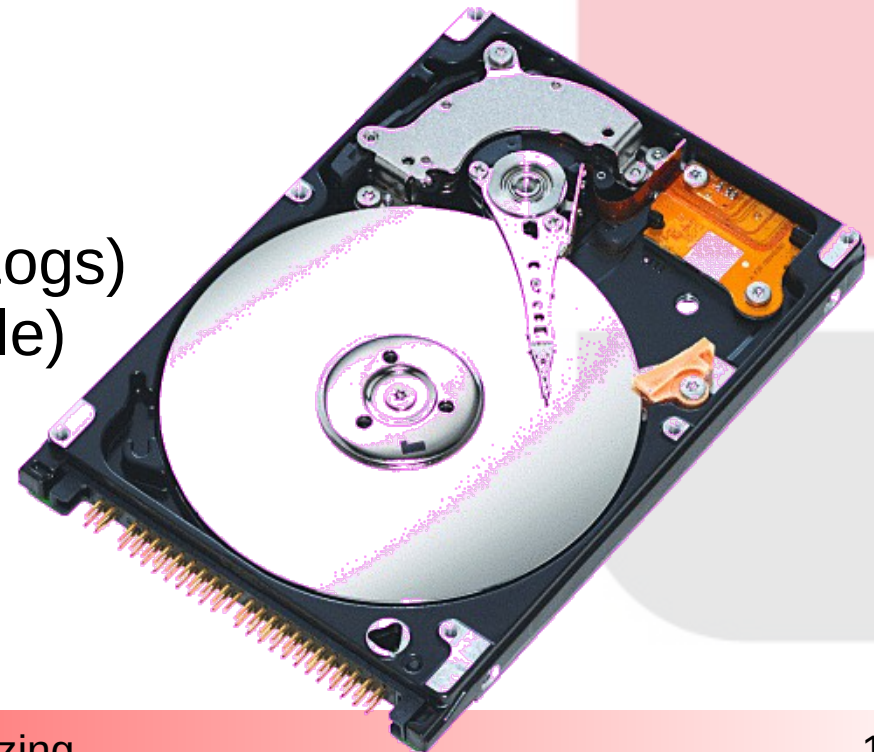
# RAM

- Relevant: Größe, Performance
- Spezifikationen, gemessen in:
  - Kapazität (Gigabytes)
  - Speichertransferrate (GB/s)
  - Latenzzeit (Nanosekunden)
- Für:
  - Caching (Buffer, Redo, Cursors ...)
  - Verwaltung / Metadaten (Shared Pool ...)
  - ausf. v. Binaries (Code Area)
  - Hash/Sort/Merge (PGA)
  - ...



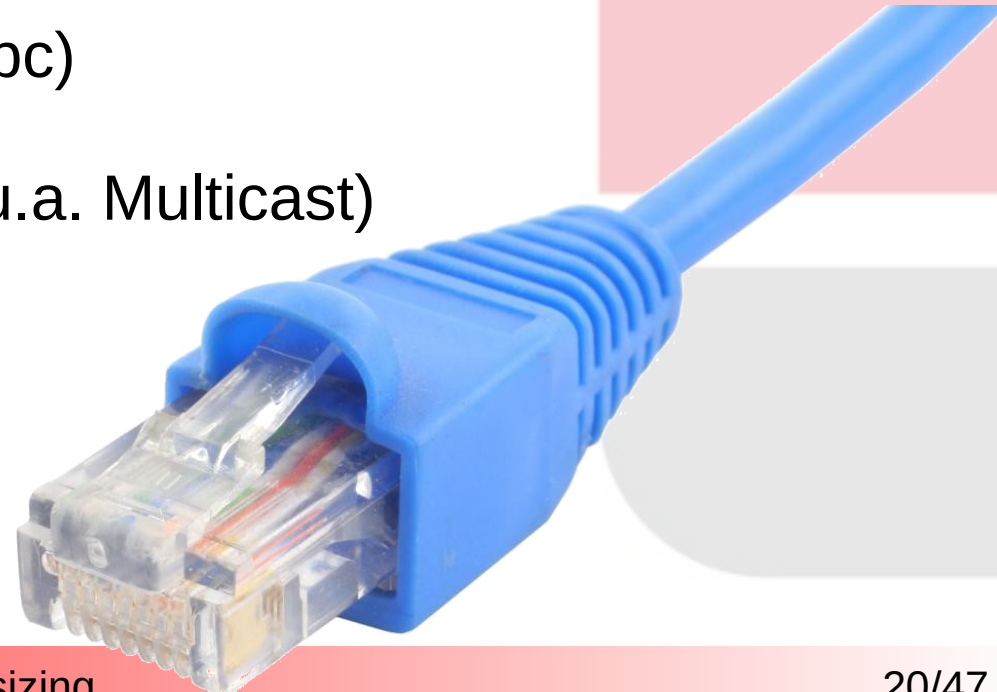
# Massenspeicher

- Relevant: Größe, Performance
- Spezifikationen, gemessen in:
  - Kapazität (Terrabytes)
  - Schreib-/Lese-Bandbreite (MB/s)
  - Latenzzeit (Millisekunden)
- Für:
  - Datenablage (Tablespaces)
  - Cache-Absicherung (Redo-Logs)
  - Metadaten (Control File, spfile)
  - Datensicherung (Backups)
  - ...



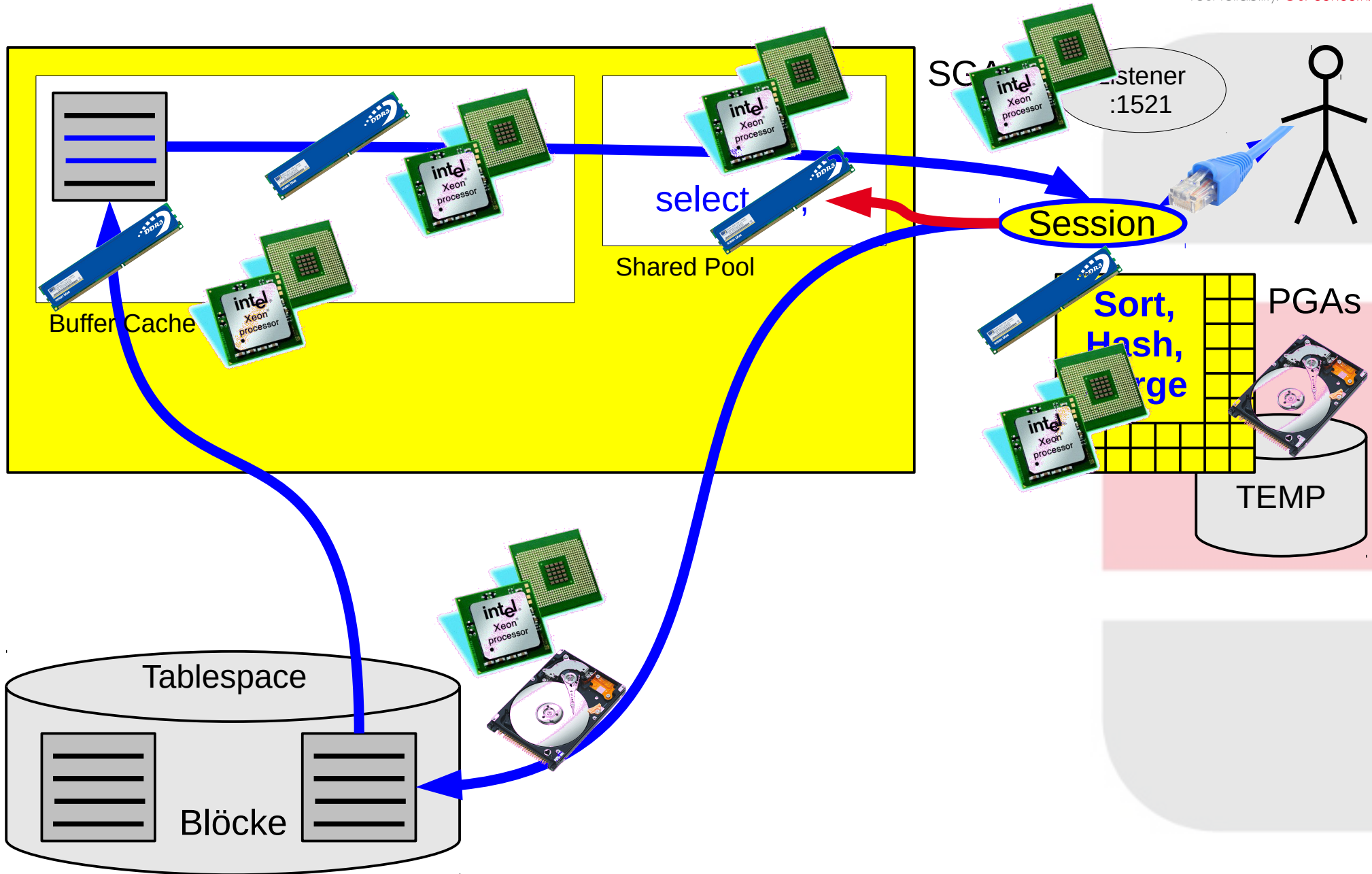
# Netzwerk

- Relevant: Durchsatz, Latenz
- Spezifikationen, gemessen in:
  - Bandbreite (Gigabit/s)
  - Latenz (Millisekunden)
- Für:
  - User-Zugriffe (TNS, OCI, jdbc)
  - Datenbank-Links (TNS)
  - RAC Cluster Interconnect (u.a. Multicast)
  - ...

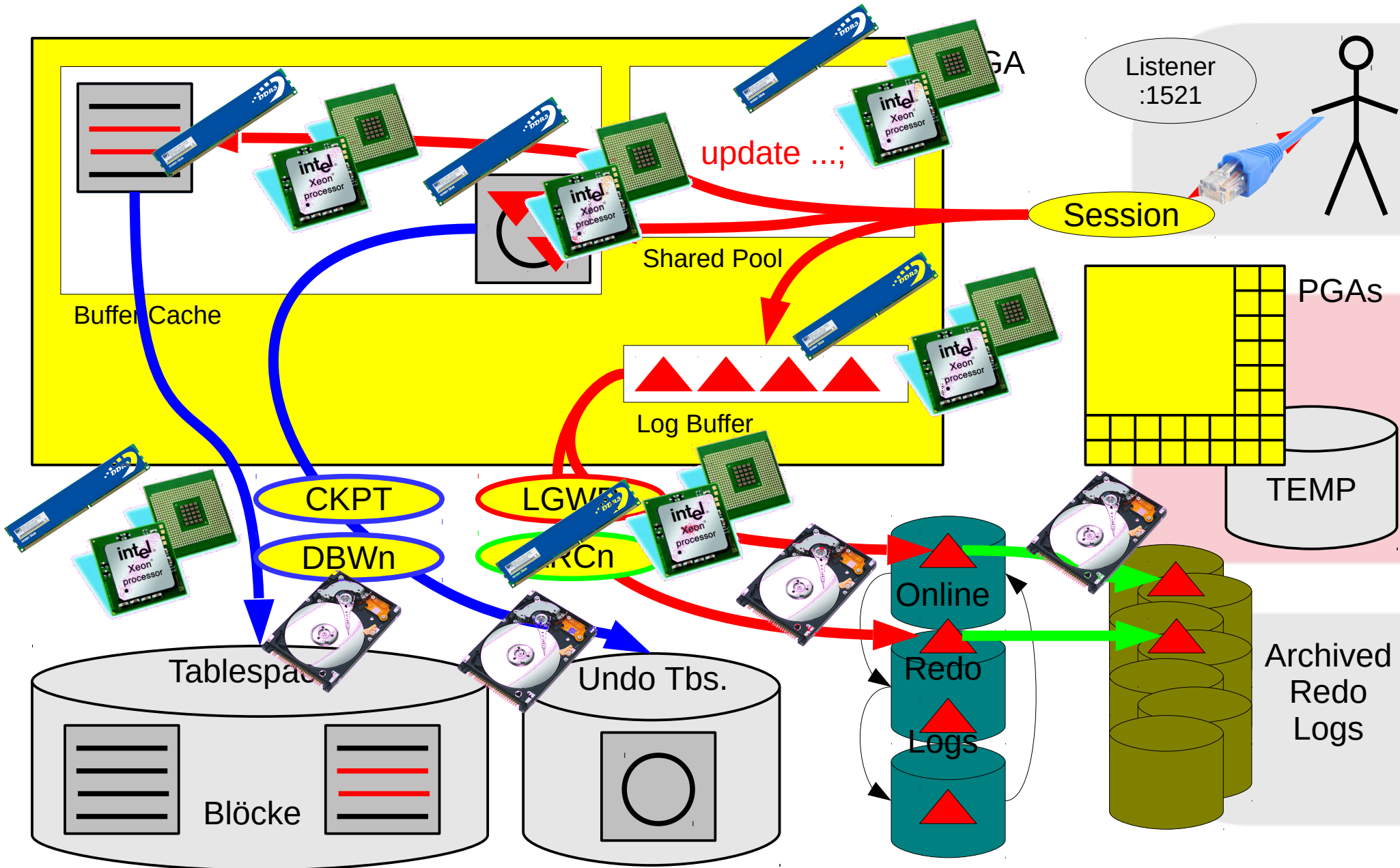


# Ressourcen- Nutzung

# Lesevorgang



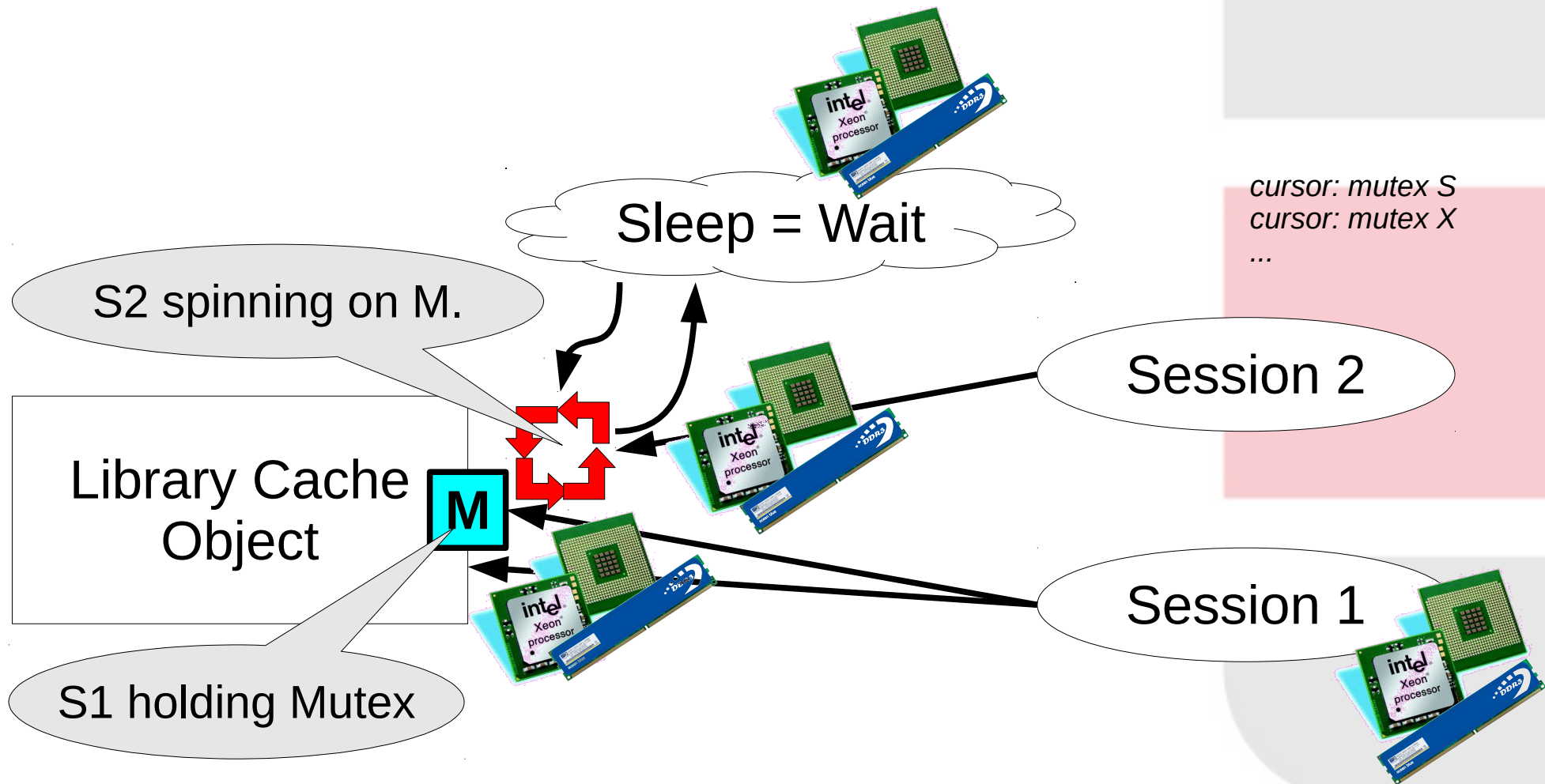
# Daten-Änderung



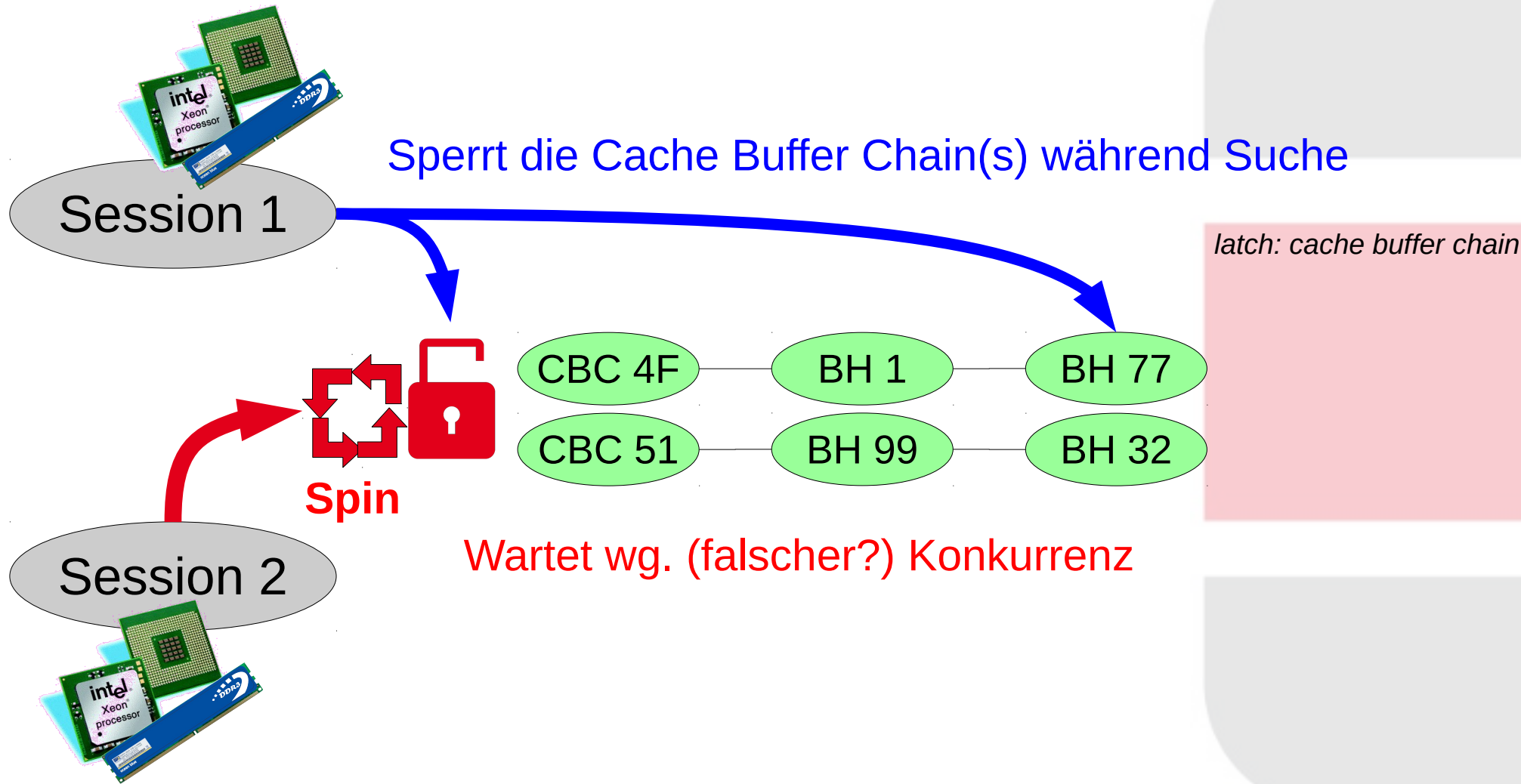
# Detail-Beispiele



# Bsp. 1: Parsing



# Bsp. 2: Find Free Buffer



# Herangehensweise

# 1. Schritt

## Taktische und Strategische Orientierung

- Wieso gibt es das Projekt?
- Wer wird das neue System
  - nutzen?
  - betreuen?
  - warten?
- An was wird es gemessen?



# 2. Schritt

## Abfragen der geplanten Last

- In abstrakten Begriffen
- Direkt beim Verursacher (=Fachabteilung?)
- Natur der Belastung erforschen

## Beispiele:

- „Wie oft wird der Dialog ausgeführt?“
- „Wie viele Daten bilden die Basis für den Report?“

# 3. Schritt

make techspeak I

- 1 Dialogaufruf  
= wieviele Abfragen?
- 1 „OK“-Button  
= 1 Transaktion??  
= wieviele Änderungen?
- 1 Massen-Ladevorgang  
= wieviele GB (pro Sekunde?)



# 3. Schritt

make techspeak II

- 1 Dialogaufruf
  - = wieviele Parsing-Vorgänge?
  - = wieviele Read IOs / Write IOs?
- 1 „OK“-Button
  - = wieviele Write-IOs?
  - = wieviele Read-IOs?
- 1 Massen-Ladevorgang
  - = wieviele Write-IO / Sekunde?
  - = wieviele Transaktionen / Sekunde?



# Meine Faustregeln

Damit „kann man anfangen“:

- Buffer Cache = 7x Segmentgröße-to-be-cached  
<http://arup.blogspot.de/2011/04/can-i-fit-80mb-database-completely-in.html>
- Anzahl Cores = 2x (Anzahl PX-Worker + HG-Prozesse)
- OLTP: pro 200GB Tablespace => 10GB Undo TBS
- OLTP: pro 1000 User Calls/s => 5GB Temp TBS
- OLTP: pro Transaktion => 5 IOs x Anzahl Tabellen



# Sizing

# Sizing Mantra

- Kenne ich meine Applikation? Sonst bin ich hier falsch.
- Kenne ich meine Applikation? Sonst bin ich hier falsch.
- Kenne ich meine Applikation? Sonst bin ich hier falsch.
- Kenne ich meine Applikation? Sonst bin ich hier falsch.
  
- Aber was, wenn trotzdem ... ?
  - Forschen
  - Forschen
  - Forschen
  
  - Beobachten  
(AWR/ASH ...)
  - Messen  
(top, iostat ...)



<http://www.haus-der-kleinen-forscher.de>

# Sizing Mantra

- Keine Standardkomponenten selbst benchmarken
  - Extrem schwierig, die korrekte Workload zu definieren
  - Vergleich zwischen Systemen nur mit extremem Aufwand machbar
- Komponenten blockweise testen:
  - CPU + RAM (Whetstone etc.)
  - Massenspeicher + SAN (ORION)
  - Netzwerk end-to-end
  - Datenbank (HammerDB, SwingBench ...)
- Optimal: Komplet-Benchmark



# Benchmarking

# Aufbaumodell

- 1) Bestandsaufnahme  
(funktionierendes und beanstandetes System)
  - a) Rahmenbedingungen festhalten  
(Systemumgebung, Lastsituation)
  - b) Kriterien definieren  
(Auf was kommt es mir an?)
  
- 2) Maßstäbe ableiten
  - a) User experience?  
(Dialogperformance, Batchdauer ...)
  - b) Systembelastung?  
(Cloud-Kosten, Energieverbrauch ...)



# Aufbaumodell

- 3) Erstellen eines eigenen Benchmarks
  - a) künstlicher Lastgenerator?
  - b) reale Belastung der Applikation?
  
- 4) Vermessung einer Referenzarchitektur
  - a) „Typisches System“ als Baseline
  - b) „Typische Lastsituation“ nachstellen
  
- 5) Skalierungsfaktoren feststellen
  - a) Systemgröße variieren  
(nur Extreme abdecken)
  - b) Belastung mögl. feingranular verändern  
(hohe Datenqualität als Basis)



# Alternative 1

- 1) Standard-Benchmark
  - a) mögl. Passenden aussuchen
  - b) Vermessung eines Referenzsystems
  
- 2) In Relation zu eigener Erfahrung mit der Applikation setzen



# Alternative 2

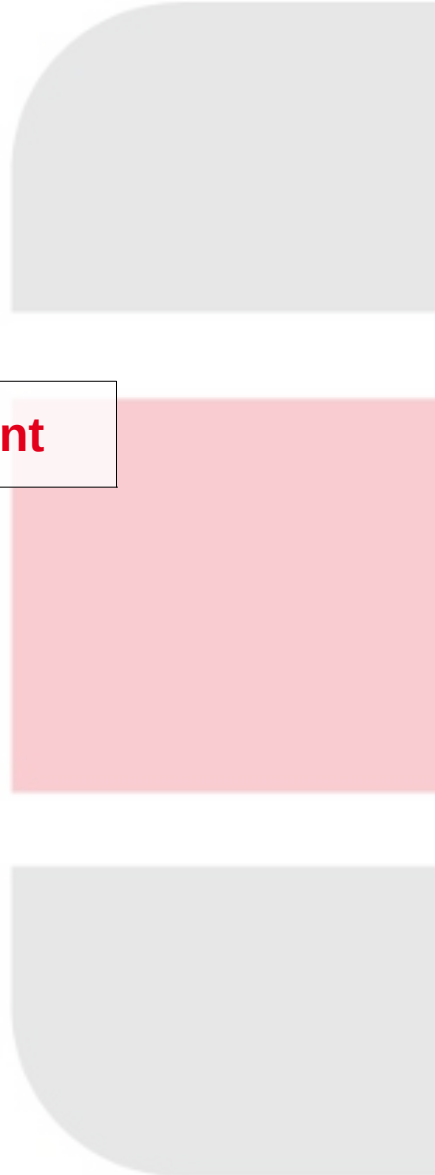
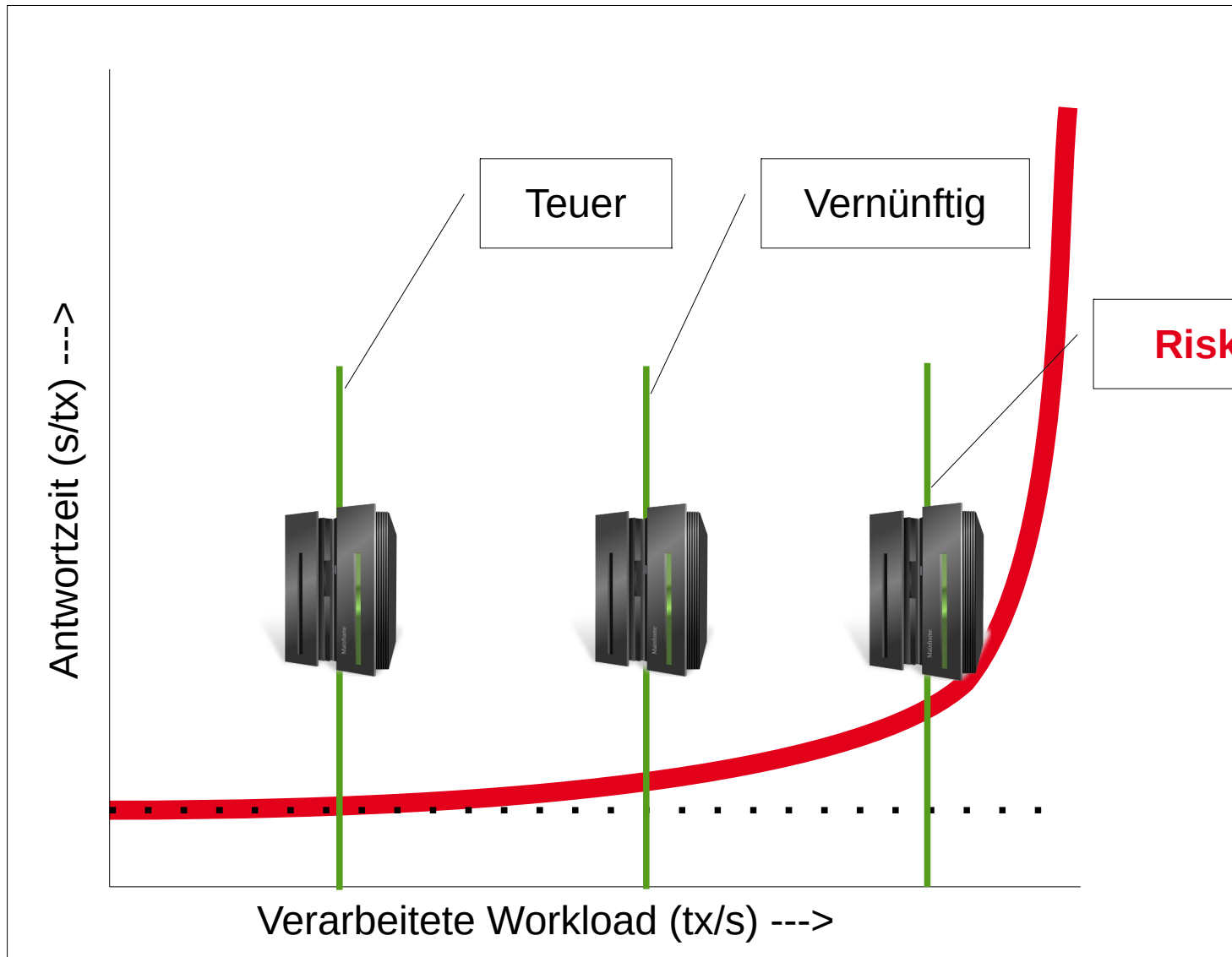
- 1) Aufzeichnen Real-Last
  - a) Oracle Real Application Testing (EE Option)
  - b) Applikationsmittel (Journal erstellen)
  
- 2) Abspielen auf anderer Umgebung
  - a) Realitätsgrad kann Herausforderung sein (Zeitliche Verschränkung vs. Datenintegrität)
  - b) Logische Änderungen nicht abbildbar ohne Applikation zu ändern



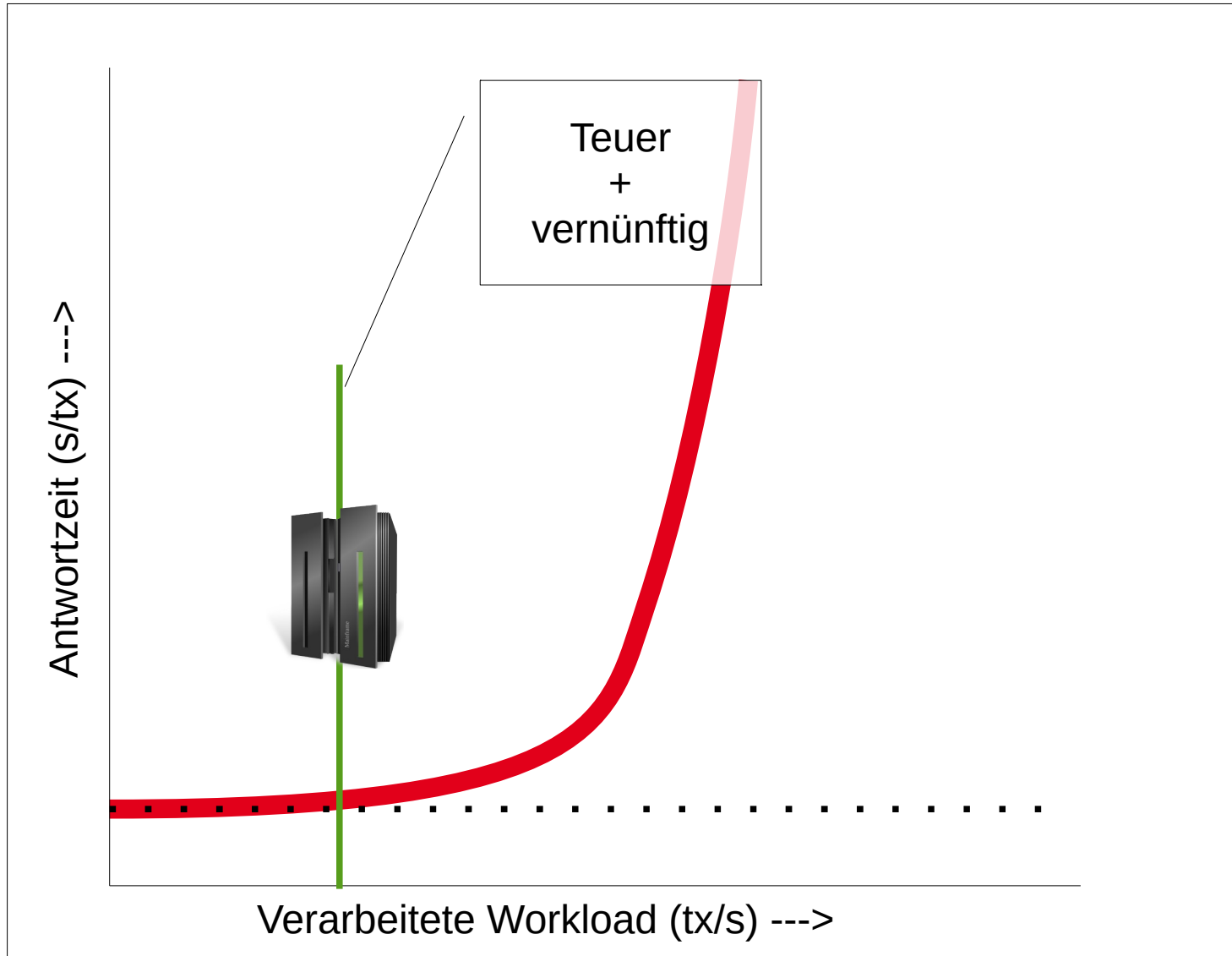


# Skalierung

# Antwortzeit



# Antwortzeit



# Abschluß

# Weiterdenken

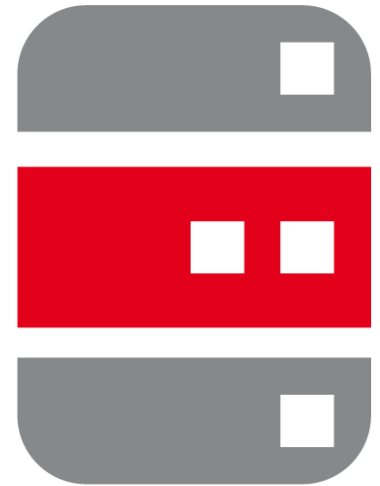
- 1) Gefühl
  - a) schlechter Ratgeber beim Übertragen von Ergebnissen (Schmetterlingseffekt)
  - b) guter Indikator beim Beurteilen von Messwerten (Realistisch?)
- 2) Durchschnitt rulez
  - a) Eine Messung ist keine Messung
  - b) Ein Erfolg ist noch kein Erfolg
- 3) Theorie ist gut, Praxis ist aber auch wichtig
- 4) Life Long Learning: immer besser werden



# Q & A



performing  
databases



Your reliability. Our concern.