Oracle RAC virtualisiert

Eine Diskussion über Sinn und Unsinn im HA-Dschungel

Dr. Thomas Petrik

Juni 2016

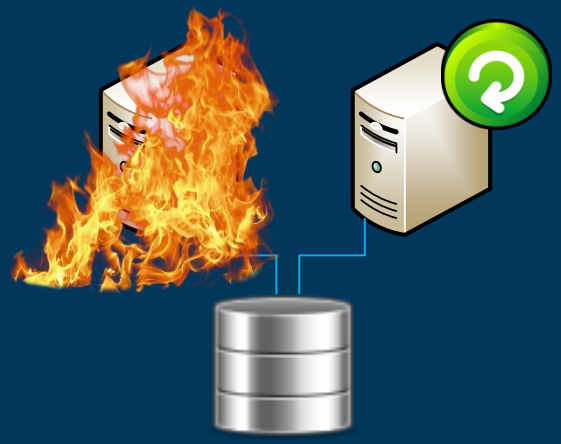
sphinx
MASTERING THE GAP

1. HA & Scalability
2. HA aus unterschiedlichen Blickwinkeln
3. RAC und Virtualisierung
4. Die Risiken der Cluster-Welt
5. Performance und RAC-Awareness
6. RAC & SE2

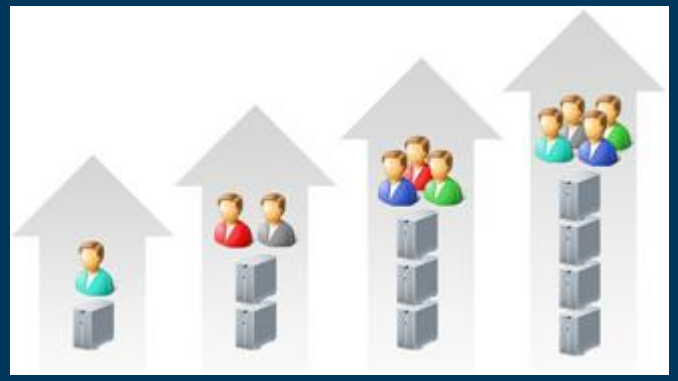
Warum RAC?

High

Availability



Scalability
(horizontal)



Was ist High Availability?

HRG-Klasse	Bezeichnung	Erklärung
AEC-0	*Conventional*	Funktion kann unterbrochen werden, Datenintegrität ist nicht essentiell
AEC-1	*Highly Reliable*	Funktion kann unterbrochen werden, Datenintegrität muss jedoch gewährleistet sein
AEC-2	***High Availability***	**Funktion darf nur innerhalb festgelegter Zeiten oder zur Hauptbetriebszeit minimal unterbrochen werden**
AEC-3	*Fault Resilient*	Funktion muss innerhalb festgelegter Zeiten oder während der Hauptbetriebszeit ununterbrochen aufrechterhalten werden
AEC-4	*Fault Tolerant*	Funktion muss ununterbrochen aufrechterhalten werden, 24/7-Betrieb (24 Stunden, 7 Tage die Woche) muss gewährleistet sein
AEC-5	*Disaster Tolerant*	Funktion muss unter allen Umständen verfügbar sein

https://de.wikipedia.org/wiki/Hochverfügbarkeit

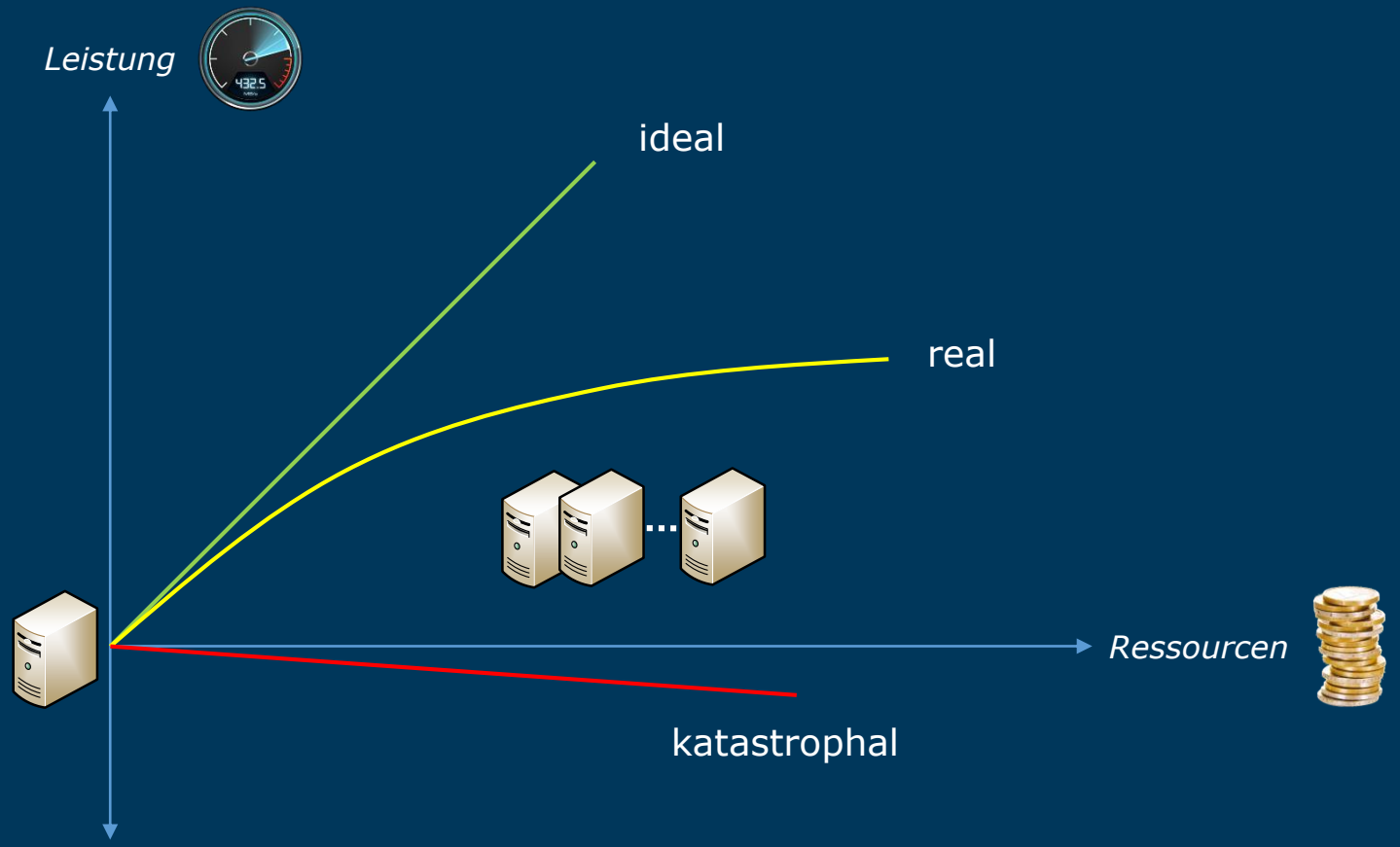
aus Sicht des SLA:

garantierte maximale Ausfallsszeit

aus Sicht des DBA:

gefühlte maximal mögliche Ausfallsszeit

Horizontale Skalierbarkeit



1. HA & Scalability
2. HA aus unterschiedlichen Blickwinkeln
3. RAC und Virtualisierung
4. Die Risiken der Cluster-Welt
5. Performance und RAC-Awareness
6. RAC & SE2

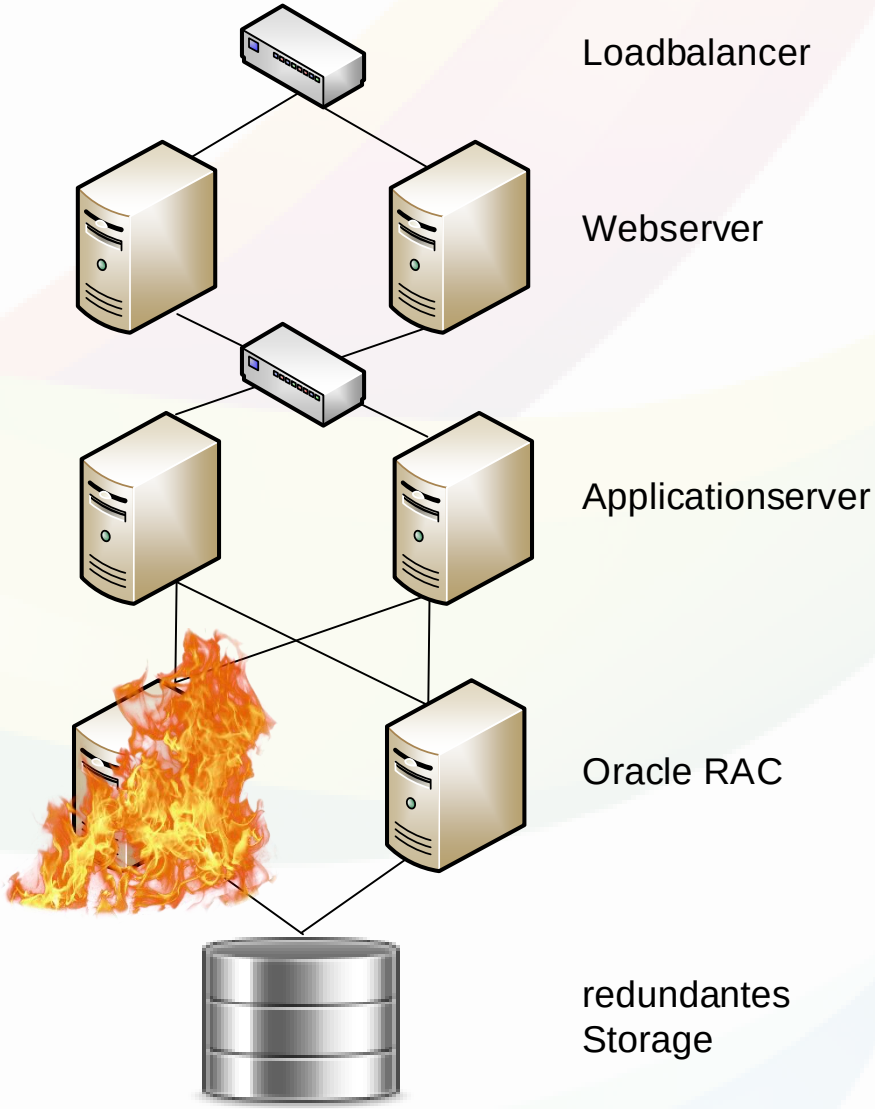
HA mit RAC: active/active Cluster

HA aus Sicht der DB

- 100% verfügbar
- bei „sauberem“ Knotenausfall

HA aus Sicht der Applikation

- aktive Connects werden unterbrochen
- automatischer Reconnect / Restart der Applikation
  - Ausfall < 1 Minute
- transparenter Failover
  - Migration der Session
  - 100% erreichbar
    - Anpassung der Applikation unerlässlich



RAC & Application Continuity

TAF

- nur OCI (TNS)
- transparent nur f. Selects
- Session State geht verloren
  - alter session set …
  - PL/SQL Variablen

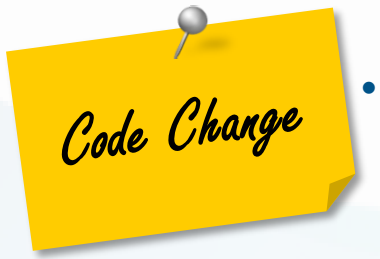
- Konfiguration: tnsnames.ora oder Dynamic DB-Service

tnsnames.ora

```
rac1_pdba=(description=(failover=on)
  (address=(protocol=tcp)(host=rac1_scan)(port=1521))
  (connect_data=(service_name=rac1_pdba)
    (failover_mode=(type=select)(method=basic))))
```

FAN & FCF

- Notification (FAN Events) via ONS (Default ab 12.1)
- automatischer Reconnect im Session Pool
  - 12.1 Client empfehlenswert
  - Weblogic, Oracle UCP (f. Drittprodukte)
- aktive Sessions terminieren
  - Error-Handling in der Applikation



Transaction Guard

- 12.1 Feature
- basiert auf LTXIDs
  - im Server- und Client-Context
  - at-most-once Semantik
- Anpassung der Applikationslogik

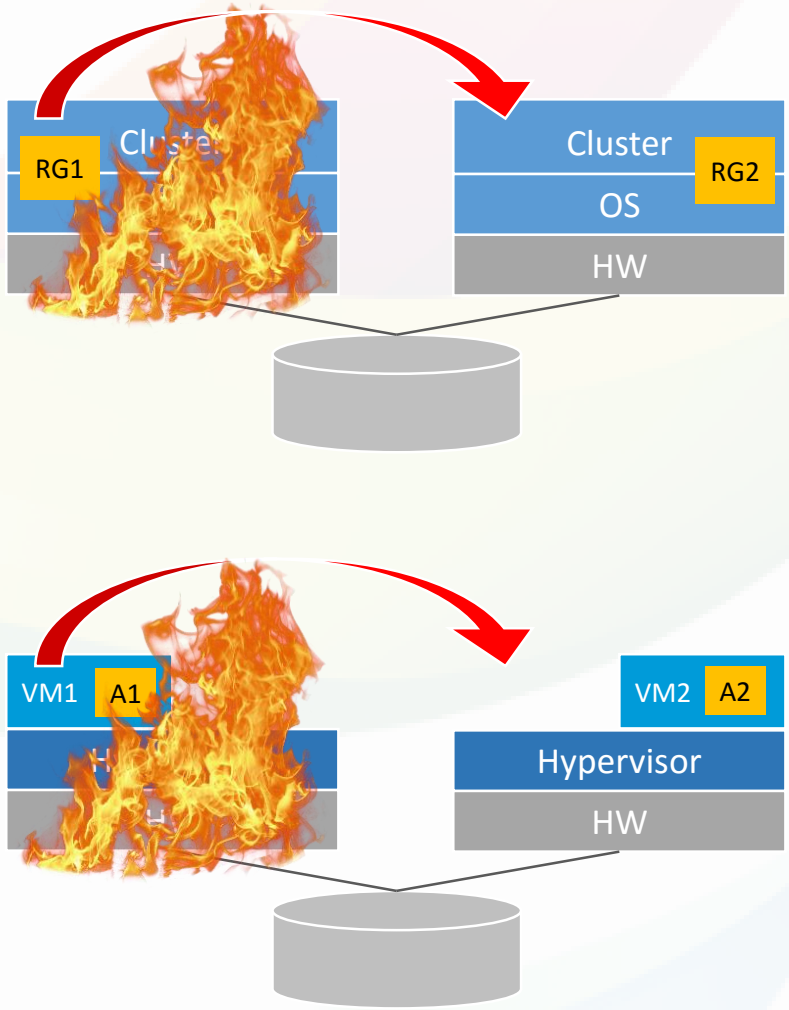
HA mit active/passive Clustern

Cluster-Software am OS

- z.B. Veritas Cluster, Power HA, GI, RAC One Node
- eigener Layer am OS
- Applikationen müssen Cluster-aware sein
- Filesysteme
  - mit failover Mount
  - concurrent (GPFS, OCFS2, ACFS, …)

Cluster als Teil des Hypervisors

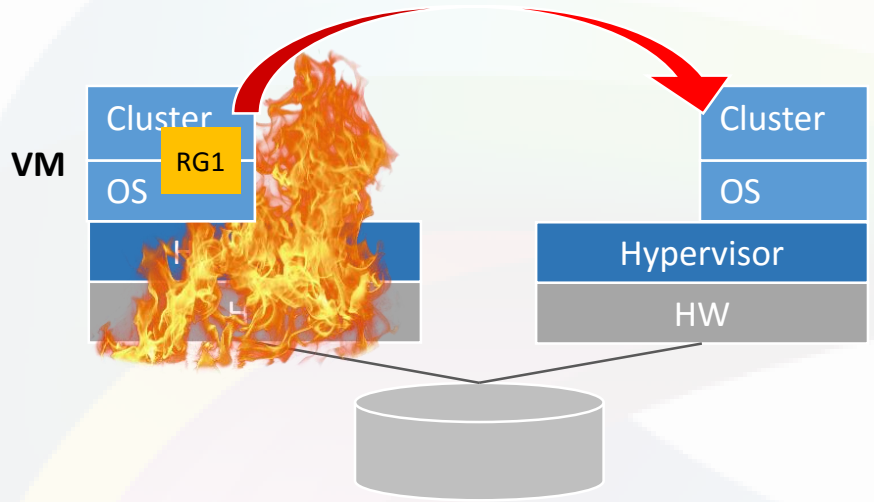
- Oracle VM, VMware, AIX LPAR, Hyper-V, …
- integrierter Bestandteil der Virtualisierung
- zusätzliche HA-Features
  - VM Live Migration
  - DRS
  - DPM



HA mit active/passive Clustern

Failover-Cluster virtualisiert

- Clusterfunktionalität der Virtualisierung ungenutzt
- Failover der Resource Group
  - von einer VM zur anderen



Knotenausfall

- Timeout bis zum Failover: ca. 30 sec.
  - Heartbeat-Logik
- Cold Failover der VMs oder Resourcegroups
  - seriell oder parallel
  - Dauer: im Minutenbereich
    - abhängig vom Filesystem (Repair Time)
      - concurrent / non-concurrent
      - Größe
    - abhängig von der Applikation
      - Startdauer

1 HA & Scalability

2 HA aus unterschiedlichen Blickwinkeln

3 RAC und Virtualisierung

4 Die Risiken der Cluster-Welt

5 Performance und RAC-Awareness

6 RAC & SE2

Virtualisierung

Partitioning

Lizenzen
Ressourcen

Overprovisioning

gleichmäßige Auslasung
höhere Packungsdichte

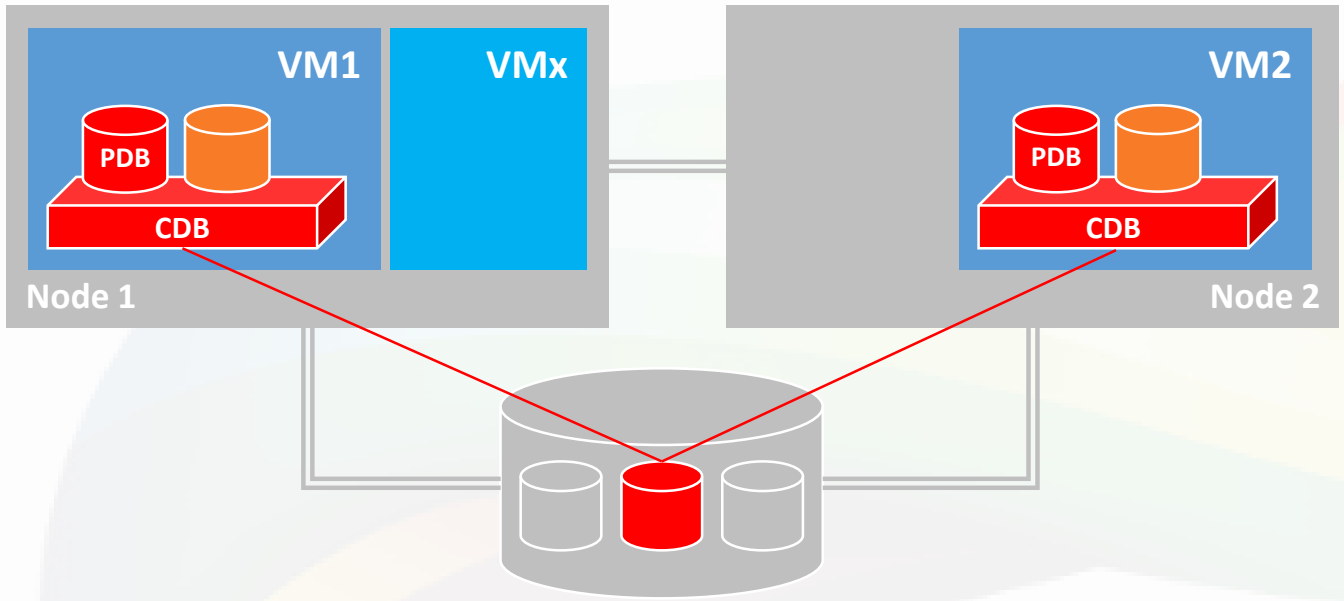
Rapid Provisioning

Templates
flexible Umschichtung
Cloud

HA-Cluster

Cold Failover
Live Migration

RAC 12c virtualisiert



HA

Failover: < 1 Min.
Live Migr.: online HW Repair

Skalierung

horizontal & vertikal

Snapshots

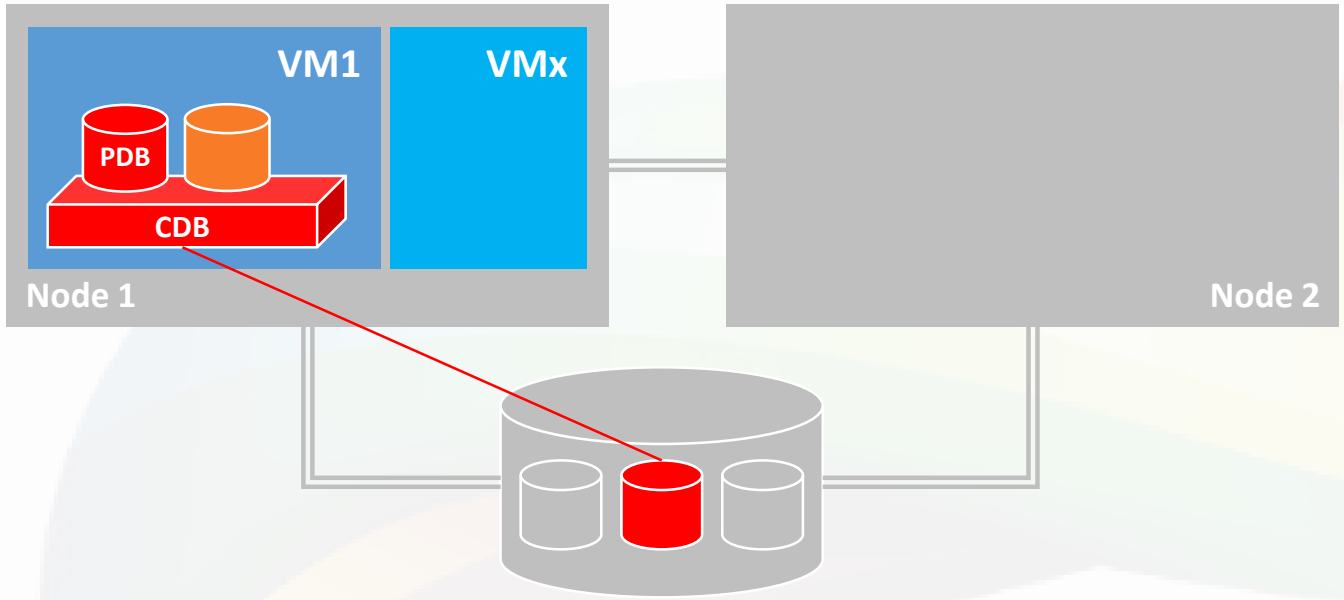
VM (nur VDisks)

Partitioning
Overprovisioning

Rapid Provisioning

CDB: VM Deployment
PDB: Multitenancy

Single Node 12c virtualisiert



HA

Failover: > 1 Min.
Live Migr.: online HW Repair

Skalierung

vertikal

Snapshots

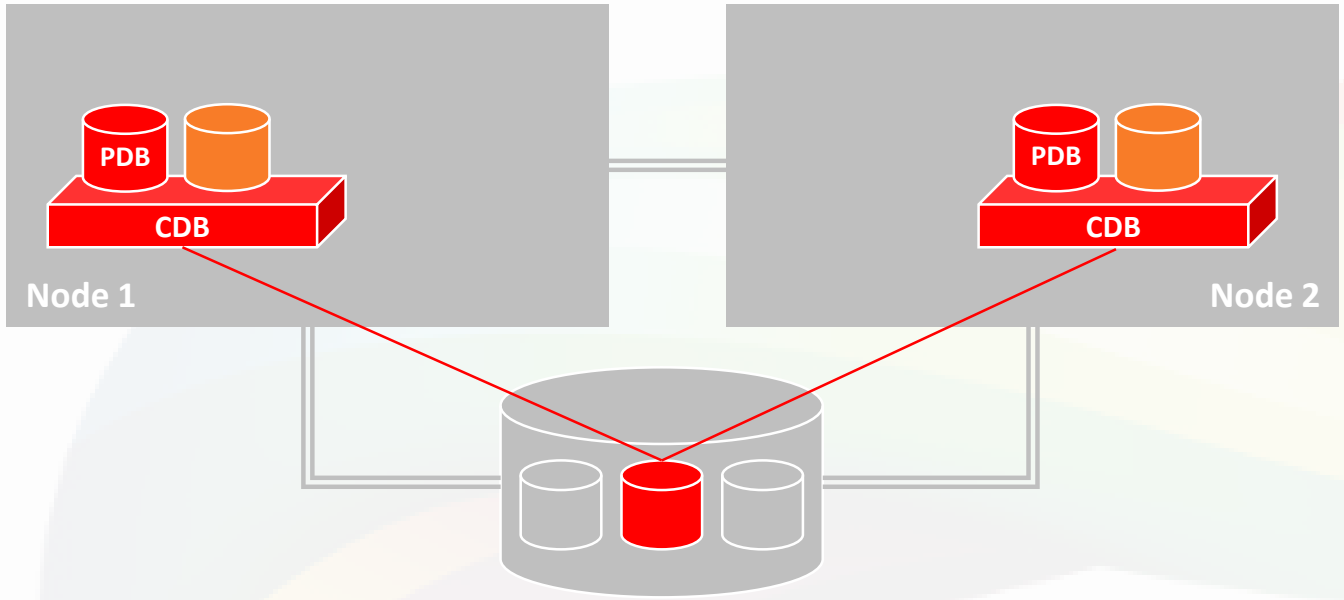
VM (nur VDisks)
PDB (snapshot copy)

Partitioning
Overprovisioning

Rapid Provisioning

CDB: VM Deployment
PDB: Multitenancy

RAC 12c nicht virtualisiert



HA

Failover: < 1 Min.
kein online HW Repair

Skalierung

horizontal & vertikal

~~**Snapshots**~~

Partitioning
Overprovisioning
nur in der CDB

Rapid Provisioning

PDB: Multitenancy

1 HA & Scalability

2 HA aus unterschiedlichen Blickwinkeln

3 RAC und Virtualisierung

4 Die Risiken der Cluster-Welt

5 Performance und RAC-Awareness

6 RAC & SE2

Real World Cluster ...

schleichender Tod einzelner Komponenten

active / active Cluster: 1 Knoten mit Performance-Problemen blockiert meist den gesamten Cluster

selten fällt ein Knoten komplett und sofort aus

Cluster können keine Mehrfachfehler abhandeln

Availabilty vs. Risk

Software = Features + Bugs

No. Bugs ~ (Lines of Code)^n

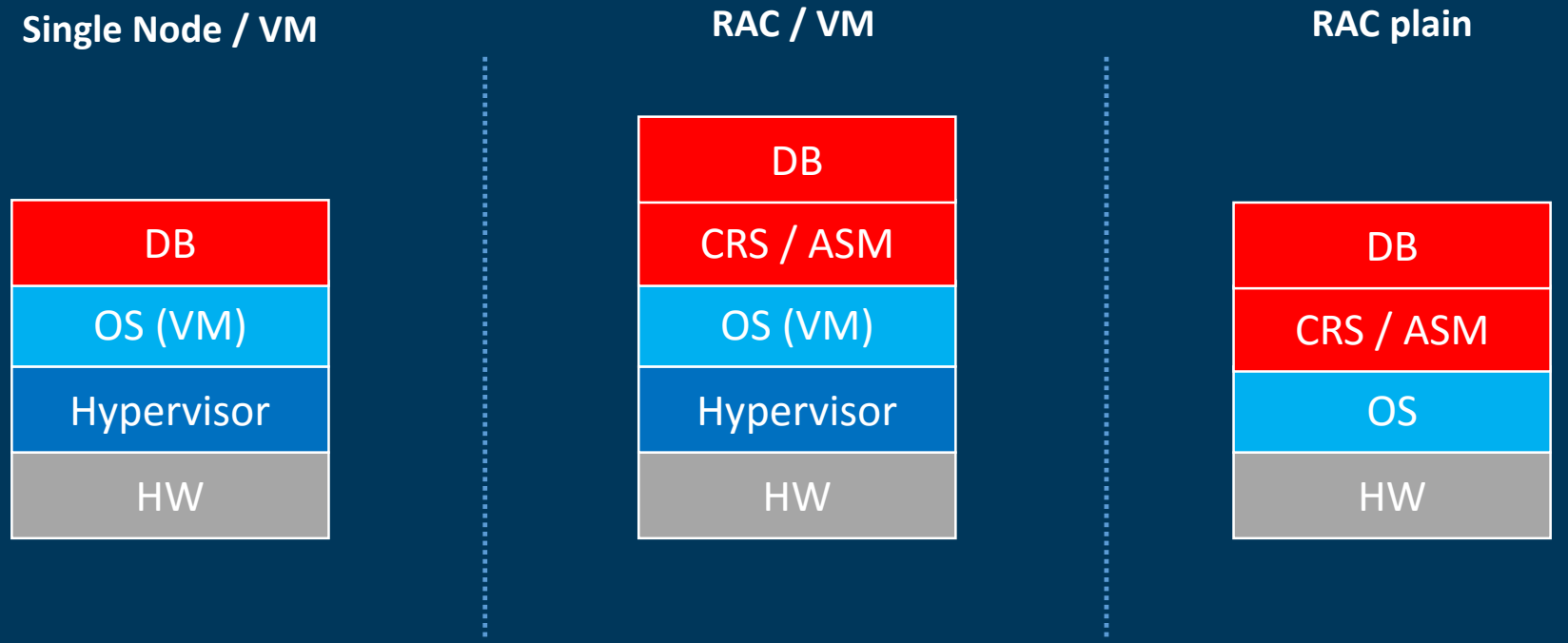
$n \geq 1$

n = f(Lines of Code)

Steve McConnecll, *Code Complete, 2nd Edition.* Redmond, Wa.: Microsoft Press, 2004.
http://www.stevemcconnell.com/articles/art06.htm

Keep it simple!
Smarte Infrastruktur = max. Erfüllungsgrad bei minimaler Komplexität

Risikoabschätzung



Fragen über Fragen ...

- Wie groß ist das Risiko der einzelnen Schichten tatsächlich?
- Wie groß ist das Risikoverhältnis von Hypervisor zu Grid Infrastructure?
- Ist das doppelte Cluster-Risiko mit RAC/VM gerechtfertigt?
- Wieviel Risikominderung bringt eine Appliance („engineered together“)?

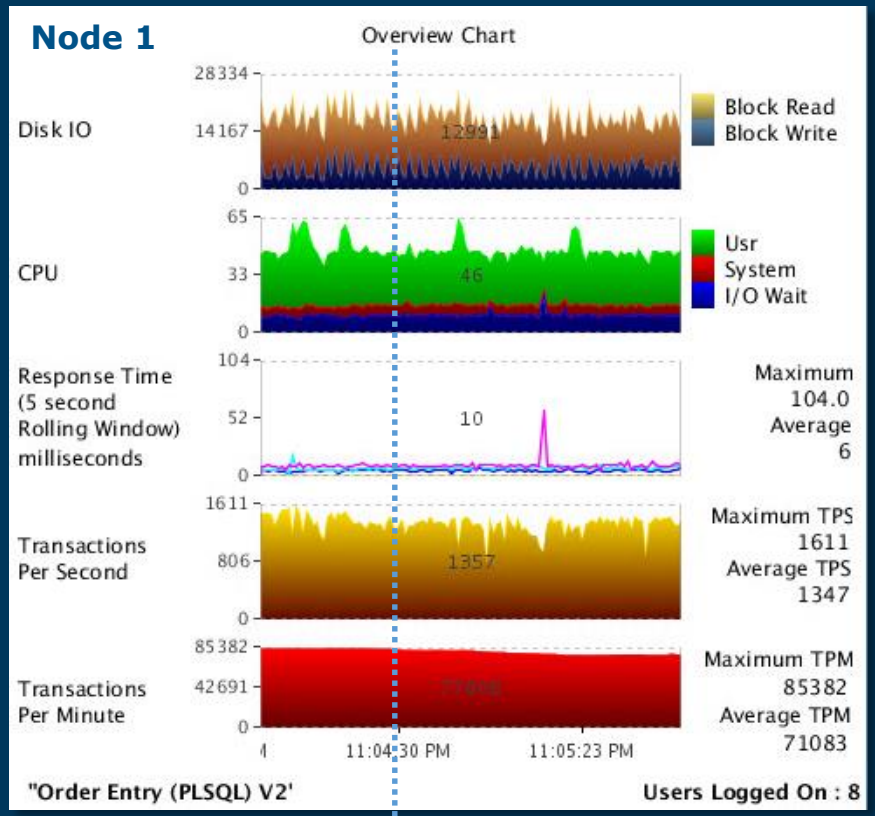
1. HA & Scalability
2. HA aus unterschiedlichen Blickwinkeln
3. RAC und Virtualisierung
4. Die Risiken der Cluster-Welt
5. Performance und RAC-Awareness
6. RAC & SE2

RAC: horizontale Skalierbarkeit

Benchmark 1 (Select-lastig)

- Swingbench „Order Entry“
- Simulation eines Hochlastsystems
- wenige Updates
- 8 concurrent User
- 2 x 10 Gbit Interconnect

Ressourcen:	1 : 2
Performance (TPM):	**1 : 1,7**



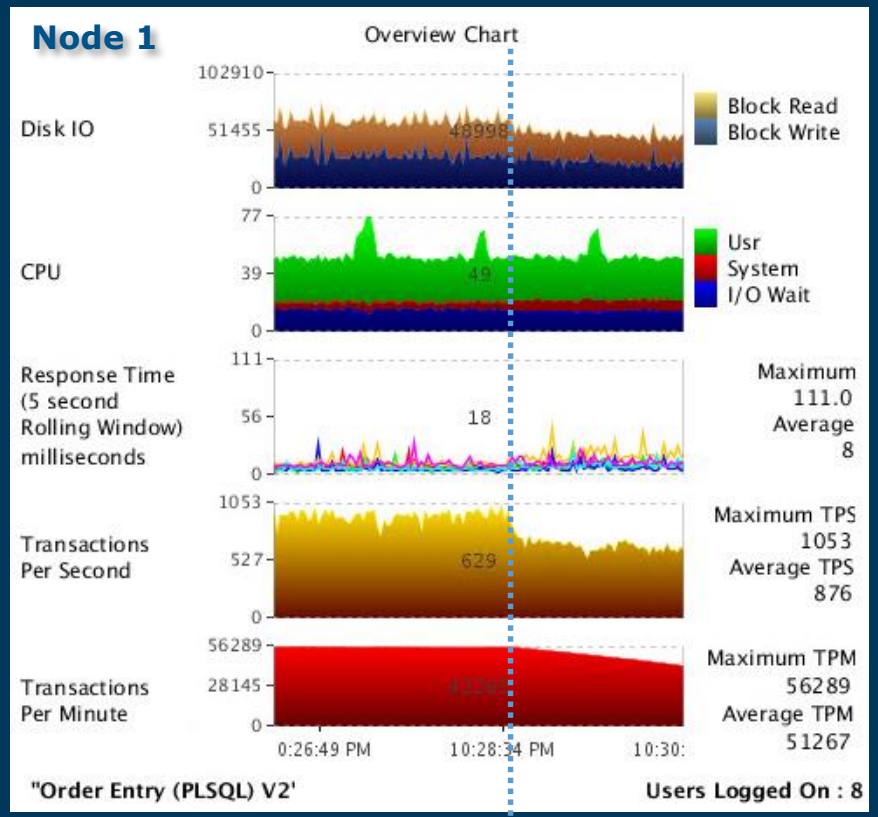
Node 2 mit gleicher Last

RAC: horizontale Skalierbarkeit

Benchmark 2 (Update-lastig)

- Swingbench „Order Entry“
- Simulation eines Hochlastsystems
  - typisch OLTP (Web-Shop, etc.)
- 8 concurrent User
  - 8 CPUs pro Knoten
  - 2 x 10 Gbit Interconnect

Ressourcen:	1 : 2
Performance (TPM):	**1 : 1,2**



Node 2 mit gleicher Last

	Wait		Event		Wait Time			Summary Avg Wait Time (ms)				
I#	Class	Event	Waits	%Timeouts	Total(s)	Avg(ms)	%DB time	Avg	Min	Max	Std Dev	Cnt
*	Commit	log file sync	73,008	0.00	334.26	4.58	29.84	4.61	3.60	5.61	1.42	2
		DB CPU			298.50		26.65					2
	User I/O	db file sequential read	193,772	0.00	232.71	1.20	20.77	1.24	1.03	1.45	0.30	2
	System I/O	log file parallel write	56,862	0.00	121.52	2.14	10.85	2.17	1.89	2.45	0.39	2
	Cluster	gc current block 2-way	133,239	0.00	84.80	0.64	7.57	0.64	0.63	0.64	0.00	2
	Cluster	gc cr block 2-way	112,348	0.00	66.07	0.59	5.90	0.59	0.58	0.59	0.01	2
	Other	target log write size	27,512	1.22	46.18	1.68	4.12	1.77	1.66	1.88	0.15	2
	Cluster	gc cr grant 2-way	39,298	0.00	23.27	0.59	2.08	0.60	0.59	0.61	0.01	2
	Cluster	gc current grant 2-way	33,838	0.00	20.51	0.61	1.83	0.60	0.60	0.61	0.01	2
	Cluster	gc cr block busy	605	0.00	16.95	28.01	1.51	31.86	27.19	36.53	6.60	2

RAC & Parallel Query

Select mit Aggregat

- Swingbench SH-Schema
- 8 CPUs pro Knoten
  - Bottleneck!
- 2 x 10 Gbit Interconnect
- parallel_force_local
  - true – lokal auf Knoten 1
  - false – verteilt auf 2 Knoten
- optimizer_dnymic_sampling = 1
  - massive Probleme bei Werten >1
  - Bug 20636003

Ressourcen:	1 : 2
Performance (Runtime):	1 : 1,9

```
| Id  | Operation                          | Name       | Rows  | Bytes |
|   0 | SELECT STATEMENT                   |            |   312 | 11856 |
|   1 |  PX COORDINATOR                    |            |       |       |
|   2 |   PX SEND QC (RANDOM)              | :TQ10006   |   312 | 11856 |
|   3 |    HASH GROUP BY                   |            |   312 | 11856 |
|   4 |     PX RECEIVE                     |            |   312 | 11856 |
|   5 |      PX SEND HASH                  | :TQ10005   |   312 | 11856 |
|   6 |       HASH GROUP BY                |            |   312 | 11856 |
|*  7 |        HASH JOIN                   |            |   312 | 11856 |
                                                          |    23 |   345 |
                                                          |    23 |   345 |
                                                          |       |       |
                                                          |    23 |   345 |
                                                          |    23 |   345 |
                                                          |   312 |  7176 |
                                                          |   312 |  7176 |
                                                          |   312 |  7176 |
                                                          |   312 | 12792 |
                                                          |   312 | 12792 |
                                                          |   312 | 12792 |
                                                          |   312 | 12792 |
                                                          |  146M|  5737M|
                                                          |  7305 | 94965 |
                                                          |  7305 | 94965 |
                                                          |  146M|  3917M|
                                                          |   16M|   170M|
|  25 |                    PX SEND HASH LOCAL  | :TQ10000   |   16M|   170M|
|  26 |                     PX BLOCK ITERATOR  |            |   16M|   170M|
|  27 |                      TABLE ACCESS FULL | CUSTOMERS  |   16M|   170M|
|  28 |                   PX RECEIVE           |            |  147M|  2387M|
|  29 |                    PX SEND HASH LOCAL  | :TQ10001   |  147M|  2387M|
|  30 |                     PX BLOCK ITERATOR  |            |  147M|  2387M|
|  31 |                      TABLE ACCESS FULL | SALES      |  147M|  2387M|
```

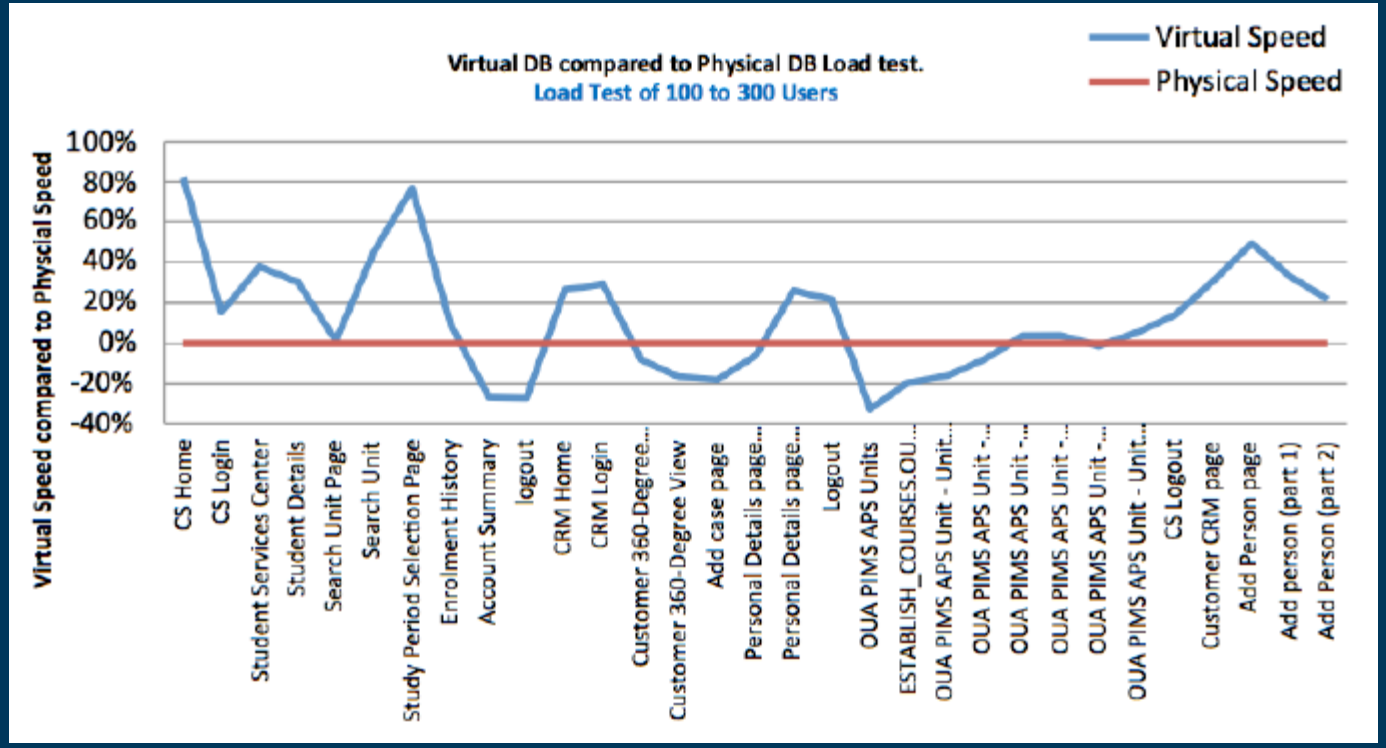
```
oracle@rac12:~/admin/C008E
nmon-14i------[H for help]---Hostname=rac12--------Refresh= 2secs ---14:23.21-----
CPU Utilisation
CPU  User%  Sys% Wait% Idle|0          |25         |50         |75       100|
  1  94.5   5.0   0.0   0.5|UUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUss>
  2  94.2   5.8   0.0   0.0|UUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUss>
  3  93.8   6.2   0.0   0.0|UUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUsss>
  4  93.0   7.0   0.0   0.0|UUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUsss>
  5  91.0   9.0   0.0   0.0|UUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUssss>
  6  92.5   7.0   0.0   0.5|UUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUsss>
  7  91.1   7.9   0.5   0.5|UUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUsss>|
  8  94.0   6.0   0.0   0.0|UUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUss>
Avg  93.0   6.7   0.1   0.3|UUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUUsss>
Network I/O
I/F Name Recv=KB/s Trans=KB/s packin packout insize outsize Peak->Recv Trans
    eth0      0.1       1.0      1.5      2.0    50.0   504.0       1.3    15.7
    eth1  85201.6   92918.3  75205.1  79277.3  1160.1  1200.2   85201.6 92918.3
    lo      887.0     887.0   5252.1   5252.1   172.9   172.9    3803.5  3803.5
```

RAC Awareness

Komponente	Anpassung
Application	Transaction Guard, FCF
Application Server	FAN / FCF, (TAF)
Restart-Logik	Autorestart der Applikation
Data Partitioning	funktionale Unterteilung der Applikation no order Sequences
Connection Partitioning	loadbalance = off (oder Service Failover)
Job Affinity	internes Application Partitioning
DD-Zugriffe	GV$-Views verwenden

OLTP

RAC virtualisiert: Performance



Open University Australia, http://www.oracle.com/us/technologies/virtualization/oracle-vm-for-oracle-database-2155841.pdf

1. HA & Scalability
2. HA aus unterschiedlichen Blickwinkeln
3. RAC und Virtualisierung
4. Die Risiken der Cluster-Welt
5. Performance und RAC-Awareness
6. RAC & SE2

RAC mit SE2

Lizenzierung der SE2

- max. 2 CPU Sockel pro Server
  - es gilt die maximale Ausbaufähigkeit
  - VM Hardpartitioning von 4- oder 8-Sockel-Servern ist unzulässig
- max. 16 User Threads (CPU Capping)
  - Background-Prozesse sind unlimitiert

RAC: spezielle Bestimmungen

- max. 1 Sockel pro Server
  - Sockel ausbauen
    - nur bei 2-Sockel-Servern zulässig
  - Hardpartitioning (Oracle VM, AIX LPAR, Solaris Capped Zones, etc.)
    - http://www.oracle.com/us/corporate/pricing/partitioning-070609.pdf
    - OVM Live Migration ist in diesem Fall untersagt
- max. 8 User Threads pro Node
  - horizontale Skalierung spielt daher mit SE2 keine Rolle mehr

Conclusio

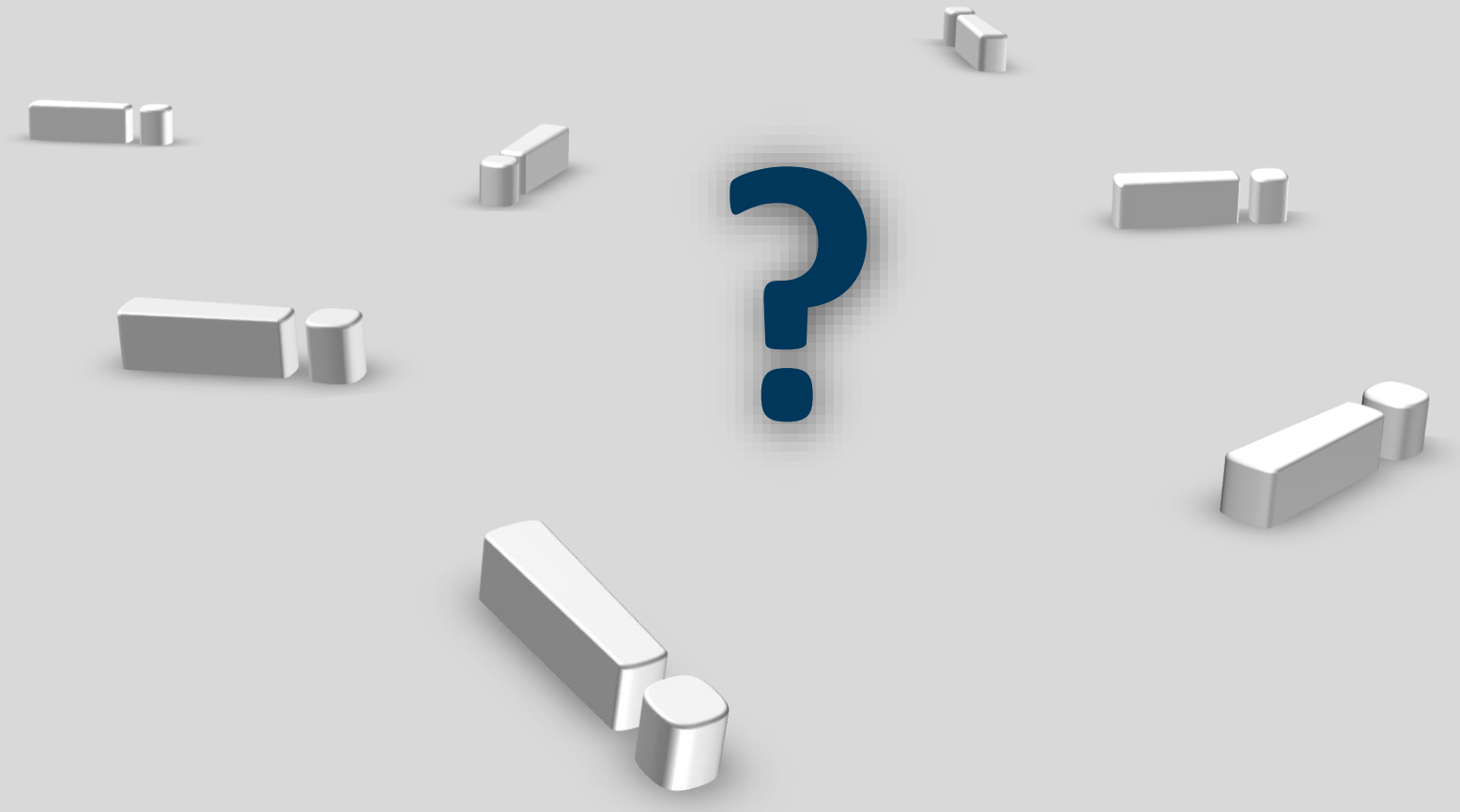
- Virtualisierung bietet massiven Mehrwert
  - integrierte HA-Funktionalität
  - rapid Deployment
  - optimierte Ressourcen-Auslastung
  - Flexibilität
    - Snapshots
    - Live Migration
  - Lizenz-Mamagement
- RAC ist in bestimmten Fällen ein sinnvolles Add-On
  - höchste Verfügbarkeit
    - Risiko abwägen
    - Applikationskette anpassen
  - horizontale Skalierung
    - Reporting-Systeme
    - Vorsicht bei OLTP Hochlast

Was ist High Availability?

HRG-Klasse	Bezeichnung	Erklärung
AEC-0	*Conventional*	Funktion kann unterbrochen werden, Datenintegrität ist nicht essentiell
AEC-1	*Highly Reliable*	Funktion kann unterbrochen werden, Datenintegrität muss jedoch gewährleistet sein
AEC-2	***High Availability***	**Funktion darf nur innerhalb festgelegter Zeiten oder zur Hauptbetriebszeit minimal unterbrochen werden**
AEC-3	*Fault Resilient*	Funktion muss innerhalb festgelegter Zeiten oder während der Hauptbetriebszeit ununterbrochen aufrechterhalten werden
AEC-4	*Fault Tolerant*	Funktion muss ununterbrochen aufrechterhalten werden, 24/7-Betrieb (24 Stunden, 7 Tage die Woche) muss gewährleistet sein
AEC-5	*Disaster Tolerant*	Funktion muss unter allen Umständen verfügbar sein

https://de.wikipedia.org/wiki/Hochverfügbarkeit

sphinx
MASTERING THE GAP



Dr. Thomas Petrik

E thomas.petrik@sphinx.at
M +43 664 155 8304
T +43 1 599 31- 0

Sphinx IT Consulting GmbH
Aspernbrückengasse 2
1020 Wien

www.sphinx.at
www.blueboxx.at