

Die Grid De-Mystifizierung – oder: Grid ist überall

Ein Oracle White Paper
Mai 2006

INHALT

Zusammenfassung	2
1 Einleitung: Grid ist überall	3
2 Die Grid-Problematik in gewachsenen Umgebungen	5
2.1 Lösung bekannter Fragestellungen mittels Grid-Technologien	6
2.2 Vereinfachte Grid-Umsetzung durch Architekturanpassungen	7
3 Die Grid-Idee live – Praxisbeispiel	10
4 Fazit	12

Zusammenfassung

Zwei Jahre nach Einführung der Oracle Database 10g ist das Interesse an dem Thema Grid oder Grid Computing groß wie selten zuvor. Andererseits finden sich immer wieder Aussagen, dass eine tatsächliche Umsetzung der Grid-Idee erst in den kommenden Jahren gelingen kann. Ziel dieses Artikels ist es daher, beide Trends aufzunehmen und zu zeigen, dass gerade bei dem Thema Grid die Wahrheit in der Mitte liegt und auch eine zukünftige Grid-Strategie bereits heute berücksichtigt werden sollte. Hierbei wird davon ausgegangen, dass es auch Grid-Lösungen geben kann, die nicht auf Oracle Real Application Clusters (RAC) beruhen. Daher soll zunächst aufgezeigt werden, wie und vor allem unter welchen Voraussetzungen eine Grid-Lösung in bestehende Infrastrukturen integriert werden kann. Erst danach wird auf die Vorteile anderer Architekturen eingegangen, um abschließend in einem konkreten Beispiel zu zeigen, wie eine adäquate Berücksichtigung aller hergeleiteten Ideen bereits heute zu einer flexiblen und damit kostengünstigeren Grid-Infrastruktur im IT-Bereich führen kann.

1 EINLEITUNG: GRID IST ÜBERALL

Wird der Grid-Gedanke ungeachtet aller Definitionen auf die grundlegende Idee der Flexibilität unter Beachtung der bestehenden Geschäftsanforderungen reduziert, gibt es schon heute unzählige Anwendungsbereiche und -fälle für eine Grid-Lösung. Dies gilt natürlich nur unter der Annahme, dass Grid schwerpunktmäßig das Ziel verfolgt, die vorhandenen Ressourcen besser zu nutzen, Kosten zu sparen und dennoch dieselbe Qualität der angebotenen Dienstleistung sicherzustellen.

Ferner müssen grundsätzlich alle Ressourcen berücksichtigt werden. Im IT-Bereich bedeutet dies: Berücksichtigung von Server, Storage, Software und der sogenannten Human Resources, also Administratoren oder allgemeiner das Personal. Eine Grid-Lösung wird sich immer im Bereich dieser voneinander abhängigen Faktoren bewegen. Die Erwartung, dass eine Grid-Lösung hilft, alle diese Ressourcen effektiver zu nutzen, dürfte sich allerdings nicht erfüllen.

Hintergrund ist, dass im Grid eine Verbesserung der Ressourcen-Auslastung in großen Teilen durch eine Umverteilung vorhandener Ressourcen erzielt wird. Dabei ist es übrigens zunächst unerheblich, mit welcher Dynamik diese Umverteilung erfolgt! Zwar gilt immer: „Je schneller, desto besser“, dennoch gibt es durchaus gute Grid-Lösungen, die diesen Leitsatz nicht erfüllen.

Auf Basis dieser Annahmen findet sich eine Grid-Umsetzung im täglichen Leben zum Beispiel in einigen Postfilialen. Vielen dürften die kleineren Postfilialen bekannt sein, die zusammen mit einer Lotteriemaststelle betrieben werden. Hintergrund dieser zum Zeitpunkt der Einführung durchaus diskutierten Zusammenlegung beider Dienstleistungen war u. a. eine effektivere Ressourcen-Auslastung. So konnten seitens der Post Ladenlokale, die Einrichtung und z. T. Personal eingespart werden. Ähnliche Synergieeffekte fanden sich auf Seiten der Lotteriemaststellen.

Wird dieses Beispiel abstrahiert und auf die Grid-Anwendung im IT-Bereich umgesetzt, ist es der Deutschen Post, die sich mit ihrer Tochter, der Deutschen Post IT Solutions, übrigens auch im IT-Bereich laut einer IDC-Studie von November 2004 mit dem Titel „Deutsche Post adopts Oracle Grid Computing“ klar zu Grid bekennt, gelungen, grundlegende Grid-Ideen umzusetzen.

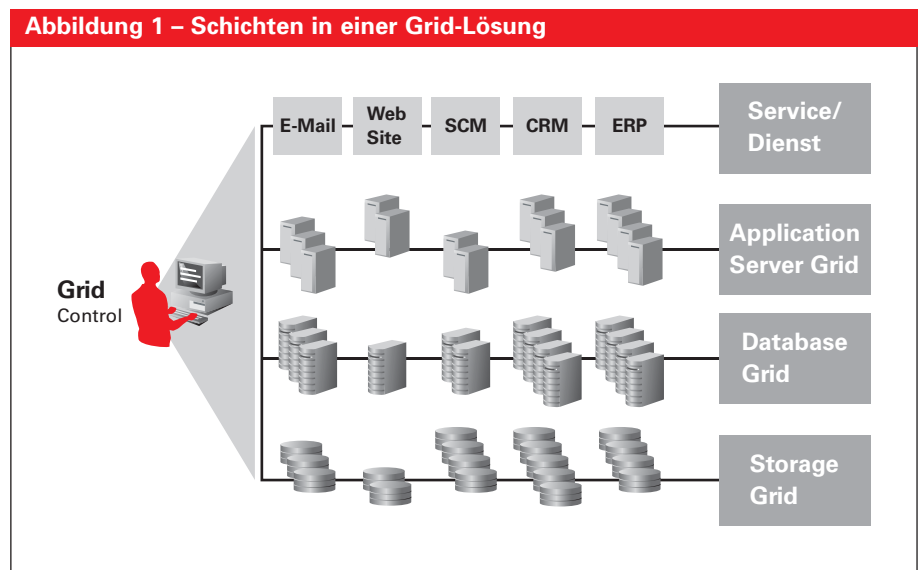
So kann eine Analogie z. B. dahingehend erfolgen, dass die Zusammenlegung der Ladenlokale als eine Storage-Konsolidierung angesehen wird; also der Zusammenfassung verschiedener kleiner Storages zu einem SAN- oder NAS-System. Die Bedienung der Kunden geschieht an entsprechenden Schaltern. Im übertragenen Sinne also von verschiedenen Servern aus. Das hierzu erforderliche Mobiliar – die hierzu erforderlichen Server – konnten reduziert werden.

Die eigentliche Ressourcen-Verlagerung findet nun immer dann statt, wenn die Bedienung für den Lotteriebetrieb und die Postabwicklung von ein und derselben Person erfolgt, die jeweils – wenn überhaupt – nur den Schalter zu wechseln braucht. Im Sinne der IT also unterschiedliche Anfragen wechselseitig von einem der beiden Server beantwortet werden (klassisches Loadbalancing im weiteren Sinne). Das ist Grid – im Prinzip.

2 DIE GRID-PROBLEMATIK IN GEWACHSENEN UMGEBUNGEN

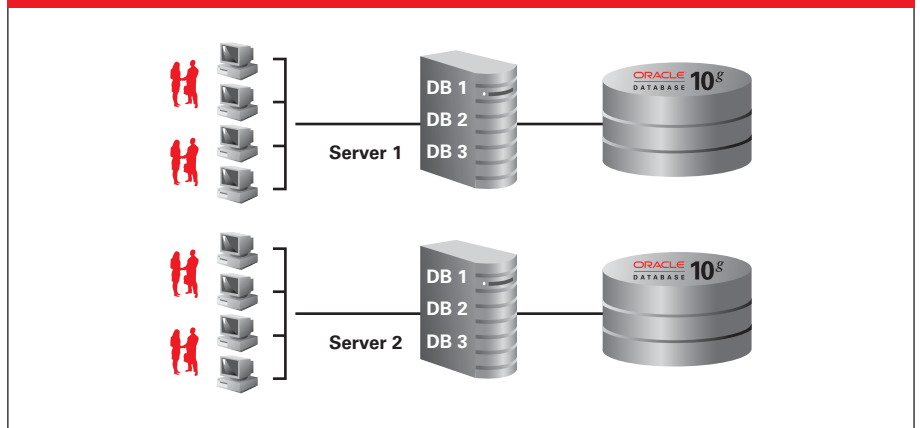
Obwohl der obige Vergleich ‚hinkt‘, kann er durchaus als ein Beispiel für die erfolgreiche Einführung einer Grid-Idee herangezogen werden. Vor allem zeigt seine Historie, dass eine Grid-Einführung immer mit Veränderungen einhergehen wird! Voraussetzung für eine erfolgreiche Grid-Umsetzung ist nämlich immer, dass eine generelle Veränderung bei der Verwendung von Ressourcen nicht von vornherein abgelehnt werden darf.

Ferner verdeutlicht die oben gezogene Analogie, dass eine wichtige Grundlage zur späteren Grid-Einführung die Storage-Konsolidierung im weitesten Sinne ist. Abb. 1 macht diesen Zusammenhang deutlich, indem sie zeigt, dass letztendlich jeder Service auf Daten basiert. Wird nun ein Service auf einen anderen Server und damit auf eine andere Ressource verlagert, muss gewährleistet sein, dass die für den Service erforderlichen Daten von dem neuen Server aus zugänglich sind.



Im Bereich des Oracle Datenbank-Grids bedeutet dies (stark vereinfacht ausgedrückt), dass es einen direkten Zusammenhang zwischen Datenbank, Instanz und Session bzw. einer bestimmten Datenbankverbindung gibt. Aus dieser Tatsache und aus historischen Gründen hat sich bei vielen Anwendern leider ein Idealbild eines Datenbank-Servers entwickelt, der schematisch wie in Abb. 2 dargestellt werden kann und welcher der Grid-Idee eigentlich entgegensteht.

Abbildung 2 – historisch gewachsene Idealform eines Datenbank-Servers



Wie in Abb. 2 angedeutet, besteht der ideale Datenbank-Server für viele Anwender auch heute noch in einer mehr oder minder großen SMP (Symmetric Multiprozessor)-Maschine, der ein in sich geschlossener Storage zugewiesen wurde. Auf dieser Maschine laufen dann die Instanzen (in der Abb. 2 vereinfacht als „DB 1“ bis „DB 3“ dargestellt) verschiedener Datenbanken und zwar in der Regel eine Datenbank pro Applikation bzw. pro Verfahren.

Sind in einem Unternehmen mehr als ein Anwendungsfall vorhanden und übersteigen die dazu erforderlichen Ressourcen-Anforderungen die Kapazität eines dieser Rechner, werden die Systeme sukzessive ausgebaut oder mehr und mehr dieser Systeme angeschafft. Das letztere Verhalten wurde in der Abb. 2 mittels der zwei abgebildeten Server bereits vorweggenommen.

Trotz dieser scheinbar vorhandenen Flexibilität wird Oracle aus verschiedenen Gründen immer wieder gefragt, wie eine Datenbank am besten von dem einen auf den ansonsten unabhängigen zweiten Server transferiert werden kann. Mit anderen Worten und in Grid-Terminologie gesprochen, stellt sich oft die Frage, wie eine möglichst schnelle und transparente Ressourcen-Verlagerung mit Oracle vorgenommen werden kann. Das heißt, dass bereits seit Jahren indirekt bestimmte Grid-Anforderungen bestehen, sie aber bisher nicht als solche der Grid-Thematik zugewiesen wurden.

Hintergrund dieser Tatsache ist vermutlich, dass die Ausgangsfrage nach einer Verlagerung einer Datenbank, auch vor Grid und ohne Grid-Techniken zu benutzen, umfassend beantwortet werden konnte. So kann eine Datenbank unter der Annahme, dass auf beiden Systemen dasselbe Betriebssystem zum Einsatz kommt, z. B. einfach mittels Betriebssystembefehlen oder idealerweise mittels Oracle RMAN (Recovery Manager) kopiert bzw. ‚ge-clont‘ werden. Ein gleichzeitiger Wechsel des Betriebssystems ist dann allerdings nur noch mittels implizitem Export/Import möglich. Ein Seiteneffekt, der eine solche Art der Ressourcen-Verlagerung sicher nicht als ideale oder universell einsetzbare Lösung erscheinen lässt.

2.1 Lösung bekannter Fragestellungen mittels Grid-Technologien

Um es gleich vorwegzunehmen: Auch mit der Oracle Database 10g und Grid-Funktionalitäten wird es in der unter Abb. 2 dargestellten Konfiguration keine Möglichkeit geben, das eigentliche Problem, nämlich das Kopieren der Datenbank von Server 1 auf Server 2, zu vermeiden! Das Problem wird nämlich nicht von Oracle, sondern von der Grid-unfähigen Hardware-Architektur begründet. Folglich kann Oracle dieses Problem auch nicht allein lösen.

Ziel der mit der Oracle Database 10g angebotenen Grid-Technologie kann es daher zunächst nur sein, den für eine solche Ressourcen-Verlagerung erforderlichen Aufwand zu reduzieren. Hierzu können für das in Kapitel 2 hergeleitete Beispiel folgende Technologien eingesetzt werden:

- Anstelle von Export/Import kann das neue und schnellere DataPump verwendet werden.
- Beide Technologien können aus dem Oracle Enterprise Manager (EM) 10g angestoßen werden.
- Alternativ können beim Kopieren auch Cross Platform Transportable Tablespaces helfen.

Läuft auf beiden in Abb. 2 dargestellten Rechnern z. B. aufgrund einer zuvor vorgenommenen Betriebssystem-Konsolidierung dasselbe Betriebssystem, empfiehlt sich zudem folgende Technik: Oracle Home bzw. Oracle DB Cloning aus dem Oracle Enterprise Manager 10g heraus.

Das heißt, in beiden Fällen tragen mit Oracle 10g eingeführte Techniken dazu bei, die Dynamik für die Ressourcen-Verlagerung zu erhöhen oder diese komfortabler zu gestalten oder sogar beides. Dies ist der Anspruch, den Oracle mit seiner Grid-Strategie in gewachsenen Umgebungen verfolgt und der sich auch bei einer weitergehenden Betrachtung des Problems durchzieht.

Denn mit einer einmaligen Kopie der Datenbank wird es in den meisten Fällen nicht getan sein. Es muss zudem ein transparenter Benutzerzugriff auf beide Datenbanken ermöglicht werden. Für diesen Fall wurde bereits mit Oracle 9i der Gedanke der Service Names eingeführt, also die Idee, verschiedene Datenbanken mittels desselben Namens transparent anzusprechen.

Transparent ist dieser Zugriff natürlich nur dann, wenn sich auch permanent dieselben Daten in den beiden Datenbanken befinden. Daraus folgt, dass schreibende Zugriffe auf beiden Datenbanken abgeglichen werden müssen. Hierzu werden klassisch folgende Techniken empfohlen:

- Trigger-basierter Abgleich der Daten über Datenbank-Links mit manueller Konfliktlösung
- Aufsetzen einer Replikationsumgebung mittels EM oder Oracle Replication Commander

Im Sinne des Grid werden mit der Oracle Database 10g zusätzlich folgende Techniken angeboten:

- Aufbau einer Oracle Streams-Lösung mittels Enterprise Manager
- Aufbau einer Oracle Dataguard Logical Standby-Lösung (mit EM)

Neben den hier genannten Technologien werden mit der Oracle Database 10g mit einer ähnlichen Begründung zudem noch weitere Funktionen und Produkte für eine Grid-Lösung angeboten. Zu nennen sind hier z. B. Oracle Flashback, Oracle Transparent Gateways sowie weitere Technologien, die im Grid durchaus sinnvoll eingesetzt werden können.

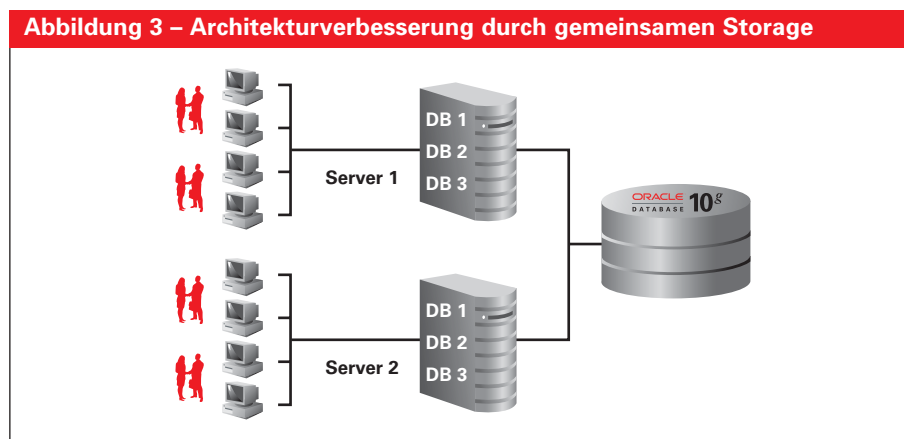
2.2 Vereinfachte Grid-Umsetzung durch Architekturanpassungen

Wie in Kapitel 2.1 gezeigt, gibt es auch in gewachsenen Strukturen durchaus einige Oracle (Grid-) Funktionalitäten für eine erfolgreiche Grid-Umsetzung. Allerdings sind diese Methoden in der Regel nicht sehr dynamisch. Diese Tatsache liegt jedoch nicht bei Oracle begründet. Vielmehr ist in der in Abb. 2 dargestellten Hardware-Architektur ein erheblicher Aufwand erforderlich, um die Daten von dem Storage des Server 1 auf den Storage des Server 2 zu transferieren. Zudem kommt in Folge der Aufwand, die Daten auf Basis zwei unterschiedlicher Storages abzugleichen.

Die Lösung dieses Problems wird in vielen Fällen darin gesehen, ein sogenanntes Storage-based Mirroring zwischen den beiden Storages zu verwenden. Das heißt, es wird ein intelligenter Mechanismus eingesetzt, um die Daten von Storage A auf Storage B zu übertragen, ohne dabei die Server mit der Übertragung zu belasten. Während mit dieser Vorgehensweise zwar das Problem der Datenübertragung gelöst wird und zudem sogar eine gewisse Absicherung gegen einen Storage-Ausfall gegeben ist, erweist sich dieser Ansatz aus Grid-Sicht eher als suboptimal.

Das Problem bei derartigen Lösungen ist nämlich, dass trotz der Verfügbarkeit der Daten auf beiden Storages dennoch kein simultaner Zugriff auf dieselben Daten ermöglicht wird. Mit anderen Worten, auch in diesem Fall kann die Datenbankinstanz nur auf Server 1 oder Server 2 laufen, aber nicht auf beiden gleichzeitig. Dennoch ist in diesem Fall zumindest dahingehend eine Verbesserung erzielt worden, dass für die Ressourcen-Verlagerung im Sinne einer Instanzverlagerung kein vorhergehendes Kopieren der Daten mehr erforderlich ist. Der Preis dafür ist allerdings, dass die Daten aller Datenbanken beider Server permanent auf beiden Storages vorgehalten werden müssen. Es kommt somit zu einer sehr hohen Redundanz bei der Datenhaltung.

Eine wirkliche Verbesserung dieses Problems ist daher nur durch eine Änderung der Storage-Architektur zu erreichen, wie sie in Abb. 3 dargestellt ist. Durch die Einführung eines Storage-Systems, welches von diversen Systemen aus erreicht werden kann, ergeben sich im Grid viele Vorteile. Der im Vergleich zu der auf Storage-Spiegelung basierenden Lösung augenfälligste Vorteil ist, dass die Daten nun nicht mehr zwingend doppelt vorgehalten werden müssen. Auch eine Ressourcen-Verlagerung im Sinne einer Instanz-Verlagerung kann nun wesentlich einfacher und effizienter vorgenommen werden; z. B. mittels eines Failover Cluster.



Während ein Failover Cluster allein zwar auch nicht in der Lage ist, den konkurrierenden Zugriff auf die nun in einem Storage gehaltenen Daten zu ermöglichen (dies ist aus Sicht der Datenbank nach wie vor der entscheidende Vorteil von Oracle RAC) und damit eine gleichzeitige Verwendung beider Server für eine Datenbank ausgeschlossen wird, kann er erheblich zu einer Beschleunigung bei der Instanz-Verlagerung beitragen. Grundlage ist natürlich auch hier der gemeinsame Storage.

Hieraus folgt, dass nicht, wie von vielen Anwendern angenommen, allein RAC die Notwendigkeit eines gemeinsam zu erreichenden Storage begründet, sondern bereits die Grid-Grundidee einer möglichst dynamischen Ressourcen-Verlagerung. Es verwundert daher auch nicht, dass sie die Grundlage klassischer Failover Cluster ist.

Eine weitere Steigerung der Effizienz bei der Ressourcen-Nutzung kann auf dieser Grundlage dadurch erreicht werden, dass bereits bei der Auswahl des Storage aus Grid-Überlegungen heraus darauf geachtet wird, einen multi-protokoll-fähigen Storage zu verwenden. Das heißt, der Storage sollte mittels möglichst vieler Protokolle erreichbar sein. Zu nennen wären hier z. B. diverse Fiber-Channel-Protokolle sowie netzwerkbasierte Zugriffe, wie z. B. NFS, SMB, iSCSI.

Wird diese Überlegung zudem auf die Ebene der Logical Volume Manager erweitert, zeigt sich, dass die richtige Wahl des Volume Manager zu einer weiteren Flexibilitätssteigerung beitragen kann. Aus Sicht von Oracle ist es z. B. unter Verwendung der Automatic Storage Management (ASM)-Technik möglich, die Dynamik, mit der die Daten zwischen den unterschiedlichen Servern ausgetauscht werden, weiter zu erhöhen.

Hintergrund ist, dass ungeachtet des nun physikalisch vorhandenen gemeinsamen Storage ein gemeinsamer Zugriff aufgrund weiterer logischer Ebenen oft nach wie vor nicht ermöglicht werden kann. Probleme können hier z. B. bestimmte Logical Volume Manager oder lokale File-Systeme sein, die einen konkurrierenden Zugriff verhindern oder einen schnellen Wechsel aufgrund von Mount- und Unmount-Operationen erschweren.

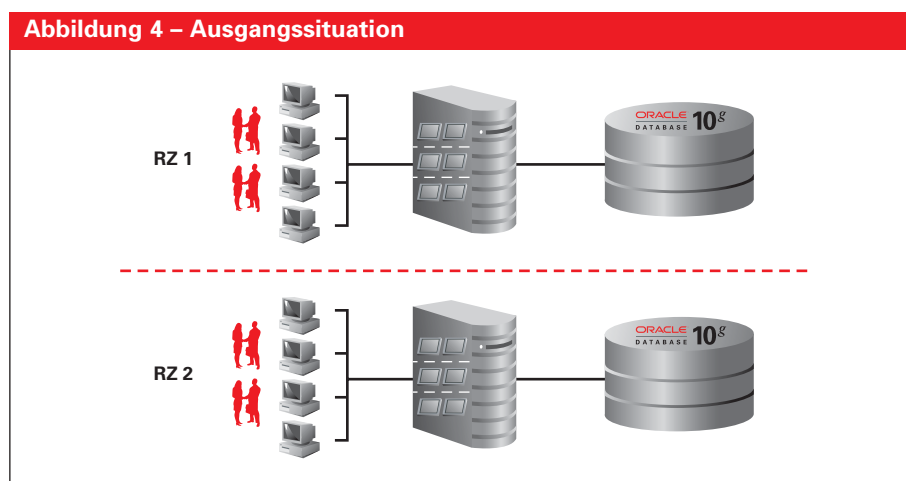
Hier hilft ASM. Mit ASM ist es sehr schnell und einfach möglich, Diskgruppen von einer ASM-Instanz in eine andere zu übernehmen. Voraussetzung ist auch hier ein gemeinsamer Storage. Dies gilt übrigens sowohl für ASM im Single-Instance-Modus als auch im RAC! ASM ist damit ebenfalls eine wichtige Grid-Technologie.

Der größte Vorteil dieser relativ kleinen Architekturanpassung liegt jedoch in der Flexibilität für zukünftige Änderungen oder Erweiterungen. So ist es nicht nur möglich, weitere Storage-Einheiten zur Absicherung des nun zentralen Speicherelements einzuführen (eine entsprechende Spiegelung vorausgesetzt), um damit beide der hier diskutierten Ansätze miteinander zu kombinieren. Sondern es ist auch jederzeit denkbar, einzelne Ressourcen-Gruppen aus Skalierbarkeitsgründen sukzessive zu erweitern. Dies gilt wiederum sowohl für den Storage als auch für den Bereich der Server.

Nur eine wie in Abb. 3 dargestellte Architektur ist von dem Grundaufbau her also in der Lage, neben einer im Grid eher ineffektiven vertikalen Skalierbarkeit auch eine angemessene horizontale Skalierbarkeit sicherzustellen. Diese kann dann natürlich nicht mehr allein auf den bisher diskutierten Failover Clusters basieren, sondern erfordert den Einsatz des Oracle Real Application Clusters. Hintergrund ist, dass nur im RAC die Nutzung mehrerer Datenbankinstanzen auf Basis einer gemeinsamen Datenbank (in einem gemeinsamen Storage) sichergestellt ist. Das ist der Grund, warum RAC im Hause Oracle als wichtige Grundlage der neuen Grid-Technologie gesehen wird. RAC erfüllt derzeit als einzige Technologie die Anforderungen hinsichtlich einer möglichst dynamischen und transparenten Ressourcen-Verlagerung.

3 DIE GRID-IDEE LIVE – PRAXISBEISPIEL

Ungeachtet der in Kapitel 2.2 hergeleiteten Ideen, die im Übrigen problemlos auf weitere Anwendungsfälle ausgedehnt werden können, verfallen viele Anwender sehr häufig immer noch klassischen Fehlern, die eine optimale Grid-Lösung von vornherein ausschließen. Der folgende Fall soll daher aufzeigen, wie eine Grid-Lösung bereits im Frühstadium geplant werden kann.



Ausgangssituation der Grid-Beratung seitens Oracle war eine klassische Fehlarchitektur, die in Abb. 4 stilisiert ist. Hintergrund war der Aufbau einer hochverfügbaren Datenbanklösung über 2 Rechenzentren (RZ) hinweg. Klassisch wurden dazu zwei sehr große SMP Maschinen (> 36 CPUs) geplant, die auf jeweils einem eigenen Storage aufsetzen. Die Storages selbst sollten mittels Storage-based Mirroring abgeglichen werden. Die Ausfallsicherung der Server sollte mittels eines Failover Cluster erfolgen. Die Server sollten zur Sicherstellung der Unabhängigkeit des Tests, Integrations- und Produktionssystems Hardware-partitioniert werden. In jeder Partition sollte ein eigenes Solaris Betriebssystem laufen. Als Randparameter wurden definiert:

- Das Integrationssystem muss dasselbe Layout haben wie das Produktionssystem.
- Bei einem Desasterausfall des RZ 1 muss dieselbe Leistung in RZ 2 bereitstehen.

Der erste Schritt der Oracle Beratung bestand in der einfachen Anwendung grundlegender Grid-Ideen mit dem Ziel einer höheren Flexibilität bei gleichzeitiger Kostenreduktion. Dazu wurde zunächst festgelegt, dass jede Hardware-Partition durch einen entsprechenden Hardware-Cluster auf Basis kleinerer Maschinen ersetzt wird. Dies hatte nicht nur eine erhebliche Flexibilitätssteigerung, sondern auch eine nicht unerhebliche (Hardware-) Kostenreduktion zur Folge.

Erst danach wurde versucht, den eklatanten Nachteil dieser Architektur zu beheben: Aufgrund des Failover Cluster zwischen beiden RZ werden nur 50% der insgesamt verfügbaren Hardware genutzt; ein herber Verstoß gegen die Grundideen für eine optimale Grid-Lösung, obgleich mit dem Failover Cluster bereits eine relativ gute Ressourcen-Verlagerung gegeben war. Für eine optimierte Hardware-Nutzung wurde daher der Einsatz von Oracle RAC empfohlen.

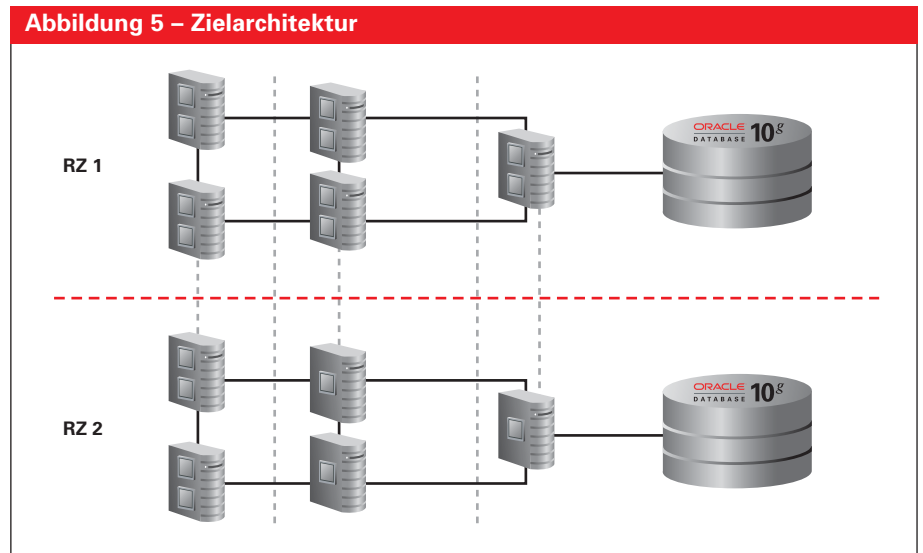
Die direkte Einführung hat zur Folge, dass zwar eine 100%ige Hardware-Nutzung erfolgt, in dem hier vorliegenden Falle jedoch bei einem regulären Betrieb ein Kapazitätsüberhang besteht. Mit diesem kann nun verschiedentlich umgegangen werden: Freie Kapazitäten können mittels Oracle Enterprise Manager anderen Verwendungen zugeführt werden oder der Überhang wird abgebaut.

An dieser Stelle beginnt die Grid-Diskussion, die von vielen Anwendern leider nicht geführt werden mag, da eine einheitliche Lösung erwartet wird. Hier kann aber allenfalls ein Blueprint weiterhelfen, da es keine einheitliche Grid-Lösung gibt! Es gibt nur einheitliche Techniken bzw. Standard-Komponenten, die in einer Lösung „out of the box“ verwendet werden können. Inwieweit diese zum Einsatz gebracht werden, hängt von dem jeweiligen Anwendungsfall ab.

In dem vorliegenden Fall entschied der Grad der Grid-Umsetzung über ein Einsparungspotenzial für das Gesamtprojekt aus Hard- und Oracle Software von bis zu 25% und zwar unter Beachtung der oben gemachten Annahmen. Das bedeutet insbesondere:

- Die verwendete Plattform (Sun Solaris) wurde beibehalten. Ein Wechsel auf Linux hätte ein weiteres Einsparungspotenzial von ca. 10% zur Folge gehabt, die nicht berücksichtigt werden.
- Das Integrationssystem muss dasselbe Layout haben wie das Produktionssystem. Hier wurde allerdings angenommen, dass dies nur für den regulären Betrieb gilt. Im Falle des Desaster war es das Ziel, dieselbe Rechenleistung zunächst allein für das Produktionssystem vorzuhalten und die anderen Systeme ggf. sukzessive nachzufahren. Auch erschien es hinnehmbar, während des Desasterfalls kein Testsystem zu haben.
- Der Administrationsaufwand darf nicht proportional zu der Anzahl der Maschinen ansteigen.

Gerade diese Einschränkungen sind der Verhandlungsraum, der im Grid gegeben ist. Inwieweit hierbei eine Änderung zu der Ausgangssituation hinnehmbar ist, kann nicht allgemeingültig festgelegt werden. Die Zielarchitektur im vorliegenden Fall ist daher wie folgt stilisierbar:



4 FAZIT

Grid ist überall. Es gibt zwar nicht die allumfassende Grid-Funktion, aber es gibt sehr viele Grid-Technologien und -Ideen. Der Anwender entscheidet daher immer selbst, ob er ‚Grid macht‘!

ORACLE®

**Die Grid De-Mystifizierung –
oder: Grid ist überall**
Mai 2006
Autor: Markus Michalewicz

ORACLE Deutschland GmbH
Riesstrasse 25
D-80992 München

Tel.: 0800-1810111
oracle.com/de

Oracle, JD Edwards and PeopleSoft
are registered trademarks of Oracle
Corporation and/or its affiliates. Other
names may be trademarks of their
respective owners.

May 2006.