

InformationWeek

Hochverfügbare Datenbanken

Die Hauptursachen von Ausfällen sind Fehler in der Software oder bei deren Bedienung durch Anwender und Administratoren. Um die Datenbanken dagegen zu wappnen, bieten die Hersteller verschiedene technische Möglichkeiten.

von martin hoermann * | werner.fritsch@informationweek.de

Immer mehr Anwendungen müssen sieben Tage die Woche und 24 Stunden am Tag verfügbar sein. Das fängt bei mittelständischen Unternehmen an, deren Geschäftsmodelle auf dem Internet basieren. In der Automobilbranche gibt es Anwendungen, die von Werkstätten rund um den Globus verwendet werden. Bei Banken sind die kontoführenden Applikationen und die Börsenlösungen extrem sensibel. Neben diesen Systemen, welche die Kerngeschäfte von Unternehmen abdecken, wachsen die Anforderungen an die Verfügbarkeit anderer operativer Anwendungen. Die fortschreitende Abhängigkeit der

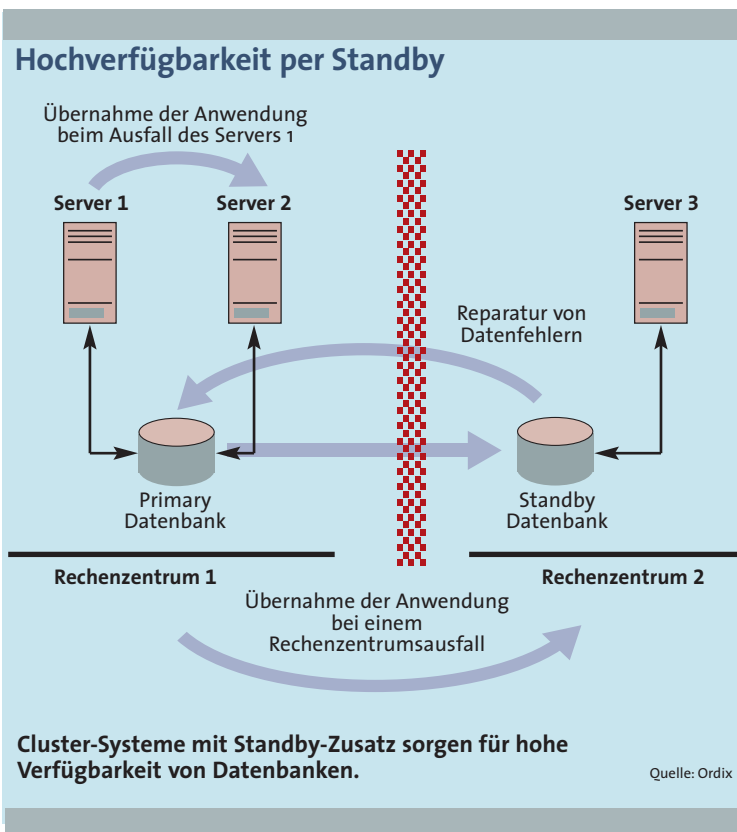
Arbeitsplätze von der Informationstechnologie wird zum Dreh- und Angelpunkt: nicht nur des geschäftlichen Erfolgs, sondern des Überlebens in einem immer härter werdenden Wettbewerb.

Deshalb wachsen die Anforderungen an die Rechenzentren, auf Fehler schnell zu reagieren und die Ausfallzeiten so gering wie möglich zu halten. Typische Cluster-Systeme, die eine Anwendung nach wenigen Minuten wieder zur Verfügung stellen, sind in vielen Fällen nicht mehr ausreichend. Forderungen nach Lösungen, die nahezu unterbrechungsfrei arbeiten, sind immer häufiger zu hören. Zwei Hauptforderungen, die damit einhergehen, sind die Wiederherstellung der Arbeitsfähigkeit nach dem Ausfall eines Rechenzentrums und die schnellstmögliche Behebung von Datenfehlern.

Ein weiterer Trend ist der Einsatz von Datenbanken für fast alle Anwendungen. Datenbanken stellen die Basis für immer mehr hochverfügbare Systeme dar und müssen daher selbst immer höheren Anforderungen genügen. Um diesen Anspruch zu erfüllen, bieten alle namhaften Datenbankhersteller Hochverfügbarkeitserweiterungen für ihre Produkte an.

Single Point of Failure

Das Kernproblem bei der Hochverfügbarkeit (HV) ist die Frage nach dem sogenannten Single Point of Failure. Das heißt: Es ist diejenige Komponente zu ermitteln, die bei einem Versagen den Komplettausfall des Systems nach sich zieht. Diesen Single Point of Failure gilt es dann mit einer HV-Lösung zu eliminieren. Bei Datenbanklösungen ist die Frage nach dem Single Point of Failure nicht immer leicht zu beantworten. Beispielsweise bietet Oracle den Real Application Cluster (RAC) an. Fällt ein Server aus, so kann die Anwendung in Sekundenschnelle auf einem anderen Server weiterarbeiten. Diese Möglichkeit bieten andere Hersteller bislang nicht. Werden jedoch versehentlich wichtige Daten gelöscht, so sind diese auf keinem Ser-



Komplexität

Jede Hochverfügbarkeitslösung führt zu einer höheren Anzahl der Software-Komponenten und erhöht dadurch die Komplexität. Bei der Kombination verschiedener HV-Lösungen steigt die Komplexität entsprechend. Dies führt in der Praxis dazu, dass HV-Anwendungen zunächst eine geringere Verfügbarkeit aufweisen als einfache Systeme. Umfangreiche Tests und kompetentes IT-Personal können dieses Paradoxon jedoch auflösen.

ver des Clusters mehr zugänglich. Für diesen Fehlerfall bietet Oracle die Standby-Datenbank-Software Data Guard an. RAC und Data Guard ergänzen sich, da sie verschiedene Single Points of Failure eliminieren.

Standby-Systeme

Der verbreitetste HV-Ansatz bei Datenbanken sind sogenannte Standby-Datenbanken. Die Standby-Lösungen der Datenbankhersteller beruhen meist auf demselben Prinzip. Änderungen auf der Primärdatenbank werden, gegebenenfalls zeitverzögert, auf einer Standby-Datenbank nachgefahren. Ein Datenfehler auf der Primärdatenbank lässt sich dann mit Hilfe einer Standby-Datenbank reparieren. Kommt es zu einem Totalausfall des Rechenzentrums, so übernimmt die Standby-Datenbank die Rolle der Primärdatenbank. Im Idealfall funktioniert dies ohne Datenverlust. Das garantiert aber nicht jede Lösung.

Im Wesentlichen gibt es zwei unterschiedliche Standby-Verfahren. Bei dem gängigen Vorgehen befindet sich die Standby-Datenbank in einem permanenten Recovery. Hierbei werden alle Änderungen an der Primärdatenbank physikalisch nachgefahren (physikalische Standby-Datenbank). Beim zweiten Verfahren werden alle Änderungen logisch extrahiert und auf einer geöffneten Standby-Datenbank nachgefahren. Dieses Vorgehen erlaubt es, die Datenbank im lesen-

den und bei einigen Herstellern auch im schreibenden Modus zu öffnen (logische Standby-Datenbank). Die Standby-Datenbank steht in diesem Fall ebenfalls für Analysen und Auswertungen zur Verfügung.

Logische Standby-Datenbanken eliminieren grundsätzlich dieselben Single Points of Failure. Je nach Implementierung eröffnet sich aber eine weitere Möglichkeit, die Verfügbarkeit zu erhöhen. Diese Möglichkeit heißt Rolling Upgrade. Damit ist es möglich, eine neue Datenbankversion ohne nennenswerte Auszeit zu installieren.

Alternative Ansätze

Neben dem Standby-Vorgehen (siehe Tabelle) gibt es andere Ansätze. Eine Variante zur Erhöhung der Verfügbarkeit sind geschickte Funktionalitäten zum Wiederherstellen von ganzen Datenbanken oder einzelnen Tabellen. Mit Hardware-Clustern lässt sich die einer Datenbank zugrunde liegende Hardware hochverfügbar gestalten. Bei einem Hardware-Defekt schaltet die Cluster-Software alle notwendigen Datenbankkomponenten transparent von einem Rechner auf einen anderen. ■

* **Martin Hoermann** ist Projektmanager bei dem IT-Dienstleister Ordix und Experte für Datenbanktechnologien.

Standby-Software für Datenbanken

Hersteller	Produkt	Read only	Read/Write	Synchron	Asynchron	Rolling Upgrade
IBM DB2	High Availability Disaster Recovery	nein	nein	ja	nein	nein
IBM Informix	High Availability Data Replication	ja	nein	ja	ja	nein
Microsoft SQL Server	Datenbankspiegelung	nein	nein	ja	nein	bedingt
Microsoft SQL Server	Protokollversand	bedingt	nein	ja	ja	bedingt
MySQL	Replikation	ja	ja	ja	nein	ja
Oracle	Physical Data Guard	nein	nein	ja	ja	nein
Oracle	Logical Data Guard	ja	ja	ja	ja	ja
Sybase	Mirror Activator	ja	nein	ja	ja	nein
Teradata	Dual Active Architecture	ja	ja	ja	ja	nein

Quelle: Ordix