

# **StorTrends Whitepaper**

## **“Die Asynchrone Replikation”**

Ajit Narayanan, © American Megatrends Inc.  
Winfried Proehl, © American Megatrends International GmbH  
12.12.2007

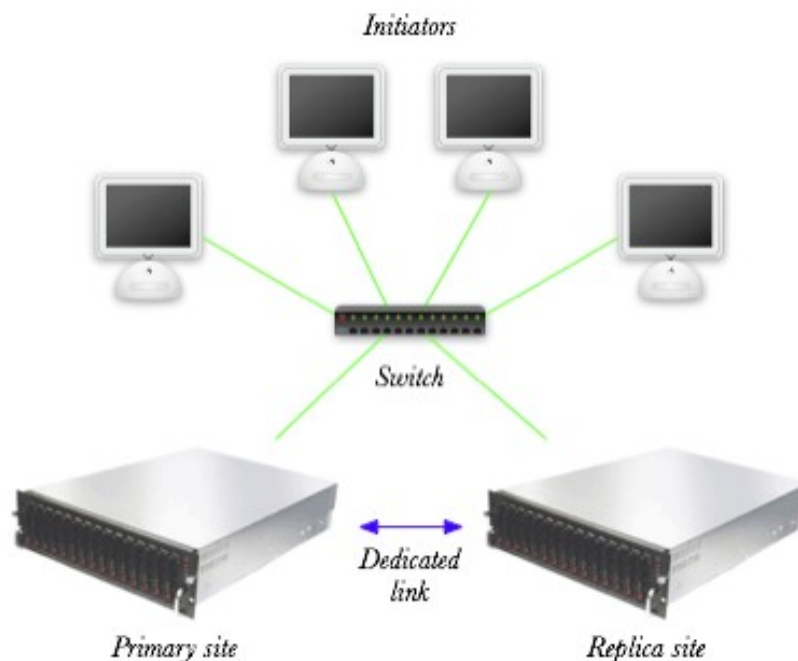
# INHALTSVERZEICHNIS

<b>KAPITEL 1: EINLEITUNG</b>	<b>3</b>
<b>KAPITEL 2: ASYNCHRONE REPLIKATION</b>	<b>6</b>
HERKÖMMLICHE METHODE FÜR DIE GEWÄHRLEISTUNG DER WRITE-ORDER-FIDELITY	7
BUCKETING FÜR ASYNCHRONE I/O	8
WECHSEL VON ASYNCHRONEN ZU SYNCHRONEN UND LOG-ASSISTED MODUS	10
<b>KAPITEL 3:</b>	<b>12</b>
TRADEMARKS AND COPYRIGHT ACKNOWLEDGEMENTS	12
FOR ADDITIONAL INFORMATION	12
LIMITATIONS OF LIABILITY	12
LIMITED WARRANTY	12
REVISION HISTORY	12

# 1. Einleitung

Der Wert der Daten übersteigt den Wert der Systeme auf denen Sie gespeichert, erzeugt und bearbeitet werden. Daher ist ihr Verlust schmerzvoller als der rein materielle Schaden im Falle einer Katastrophe. Viele Unternehmen sind so eng an ihre Daten gebunden, dass der Verlust das Überleben der Firma in der bestehenden Form unmöglich macht.

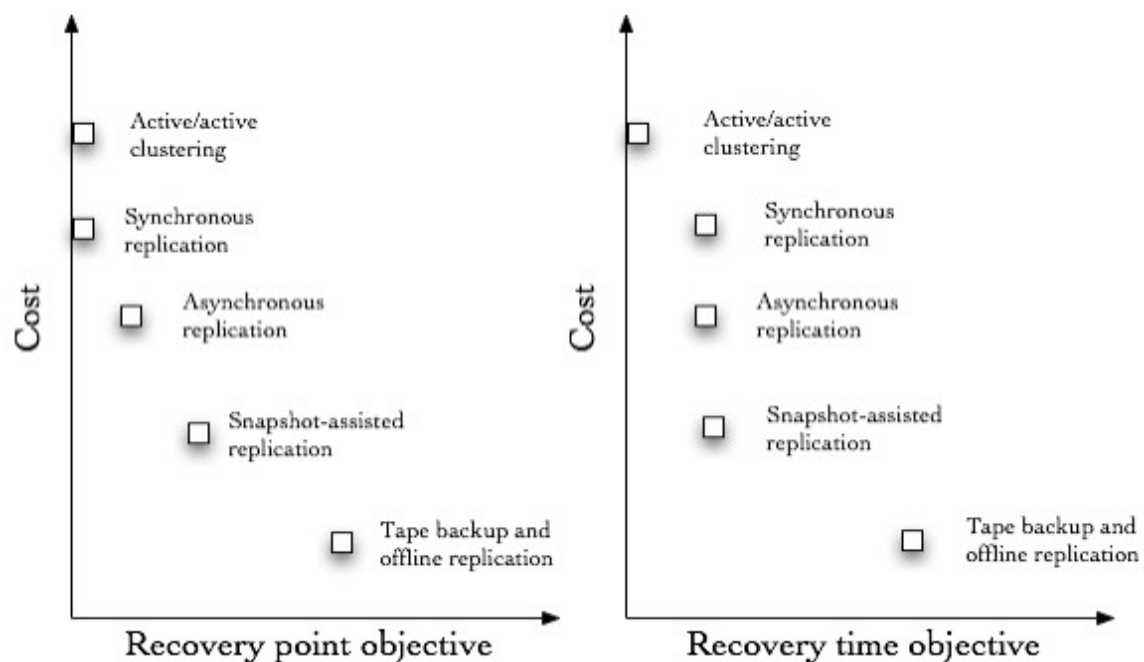
Um sich von dieser Katastrophe zu schützen, werden Daten lokal getrennt mehrfach gespeichert. Hierbei wird die Kontinuität der Geschäftsabläufe weiterhin gegeben, selbst wenn ein Standort vollständig ausfällt. Diesen Prozess des Übernehmens des duplizierten Datenbestandes von einem Server zu einem anderen nennt man Replikation oder Remote Mirroring. Die eingebundenen Server werden hierbei als 'Primary' und 'Secondary' Server, oder 'Primary' und 'Replica' Server bezeichnet.



**Abbildung 1: Typischer Aufbau der Replikation**

Die Replikation unterscheidet sich von üblichem Back Up (wie z.B. auf Bandlaufwerke) durch Gleichzeitigkeit, d.h. Online während der produktiven Zeit ohne Einwirkung auf die Produktivität.

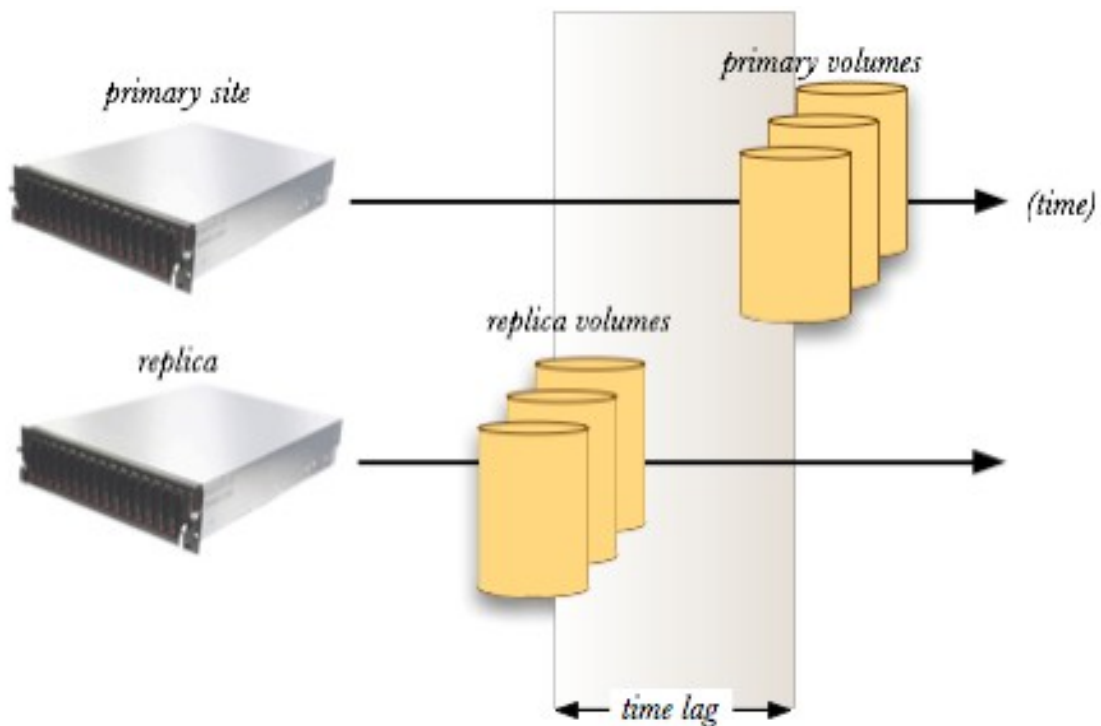
StorTrends Storage Appliance Server berücksichtigen die Bedeutung der Replikation und haben fortgeschrittene und neue Merkmale, welche ein Maximum an Datenschutz und höchste Flexibilität der Anwendung bieten. Dieses Dokument beschreibt die verschiedenen Konzepte, die für die Replikation angewandt werden, sowie die spezifischen Neuerungen in StorTrends, die diese Konzepte wirksam machen.



**Abbildung 2: RPO und RTO der wichtigen Replikationsverfahren**

Die Einbindung von Synchroner Replikation ist kostspielig, da diese eine schnelle Verbindung mit hohem Datendurchsatz zwischen dem Primary und Secondary erfordert um Störungen bei der Übermittlung auszuschließen. Bei Betrieb von Standorten mit hoher Entfernung, z.B. kontinentübergreifend, können solche Kosten für eine separate hochleistungsfähige Verbindung bis über eine Million Euro betragen. Daher mag man stattdessen der Asynchronen Replikation den Vorzug geben. Bei dieser Methode werden die I/Os nicht gleichzeitig von den Anwendungen zum Primary und zum Secondary gesandt, sondern Sie werden vom Primary erst für einige Sekunden zwischengespeichert und doppelte Schreibzugriffe werden bis zu einem

bestimmten Grad vermieden und dann gesammelt mit einer Verzögerung an den Secondary übertragen. Da das Zwischenspeichern von Daten eine bessere Nutzung der Übertragungsbreite ermöglicht und da zwischengespeicherte Daten komprimiert und für die Übertragung optimiert werden können, darf die Übertragungsleistung bei der Asynchronen Replikation erheblich geringer sein als bei der Synchronen Replikation. Dadurch ergibt sich auch eine wesentliche Reduzierung der Kosten durch moderate Anforderungen an die Übertragungsgeschwindigkeit. Andererseits zahlt man für diese geringeren Kosten einen höheren Preis durch gestiegene RPO: im Falle eines vollständigen Ausfalls des Primary werden alle zwischengespeicherten Daten verloren sein und der Secondary wird sich nach Inbetriebnahme mit einem Datenverlust von einigen Sekunden gegenüber den Anwendern darstellen.



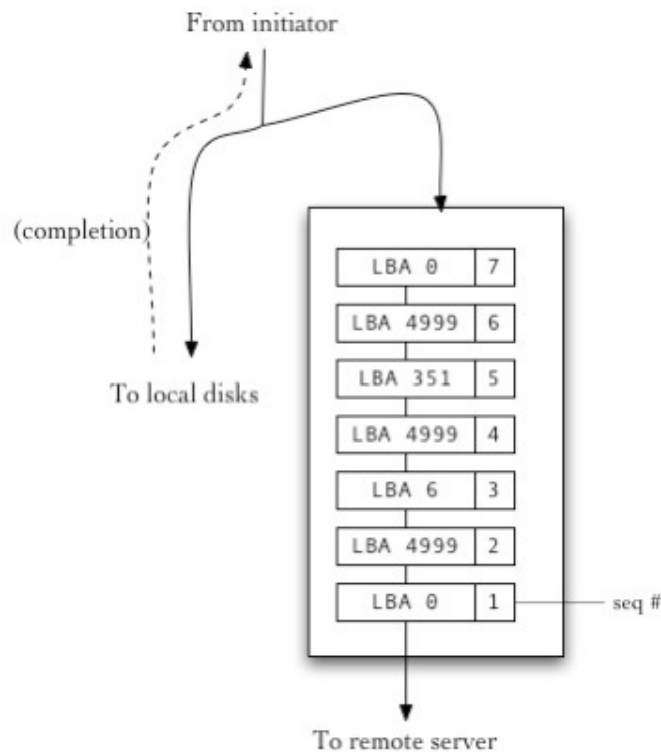
**Abbildung 3: Asynchrone Replikation**

## 2. Asynchrone Replikation

Die Asynchrone Replikation unterscheidet sich von der Synchronen indem die vom Primary Server erhaltenen I/Os umgehend nach Erhalt durch den Primary abgeschlossen werden und zusammengefasst werden für die Übertragung zum Secondary. Die Replikation der Daten erfolgt also nicht durch gleichzeitiges Schreiben der I/Os auf den Primary und Secondary, sondern zeitversetzt auf den Secondary.

Die asynchrone Replikation erhöht die I/O Performance des Primary, da die Latency wesentlich verringert wird. Asynchrone Replikation kann ebenfalls die Anforderungen an die Übertragungsbreite verringern, da die I/Os in Gruppen zusammengefasst werden können, um größere I/Os zu bilden, die dann an den Secondary weitergeleitet werden, auch entfallen hierbei doppelte Schreibzugriffe zur gleichen LBA.

Jedoch kommt in der asynchronen Replikation der Write Order Fidelity eine besondere Bedeutung zu. Da der Initiator nicht mehr die direkte Kontrolle über die Anordnung der I/Os hat, die zum Secondary gesendet werden, kann er auch nicht mehr sicherstellen, dass abhängige Schreibzugriffe in der korrekten Folge auf den Secondary repliziert wurden. Die Aufgabe des Replikationsmechanismus ist daher noch schwerwiegender: jetzt ist die Verantwortung für die sichere Funktion der Applikationen im Falle des Failover auf den Storage Server übergegangen. In den meisten zeitgemäßen Anwendungen von asynchroner Replikation, sammelt diese die I/Os die in dem Primary Server ankamen und sendet diese I/Os zu dem Secondary Server in exakt derselben Anordnung ohne irgendeine Art von Rahmen oder Buffer. Hierdurch wird die Write Order Fidelity gewährleistet; jedoch dieses verhindert weitgehend jedweden zusätzlichen Gewinn, der durch die asynchrone Replikation hinsichtlich der genutzten Bandbreite und der Leistung ersichtlich wird.

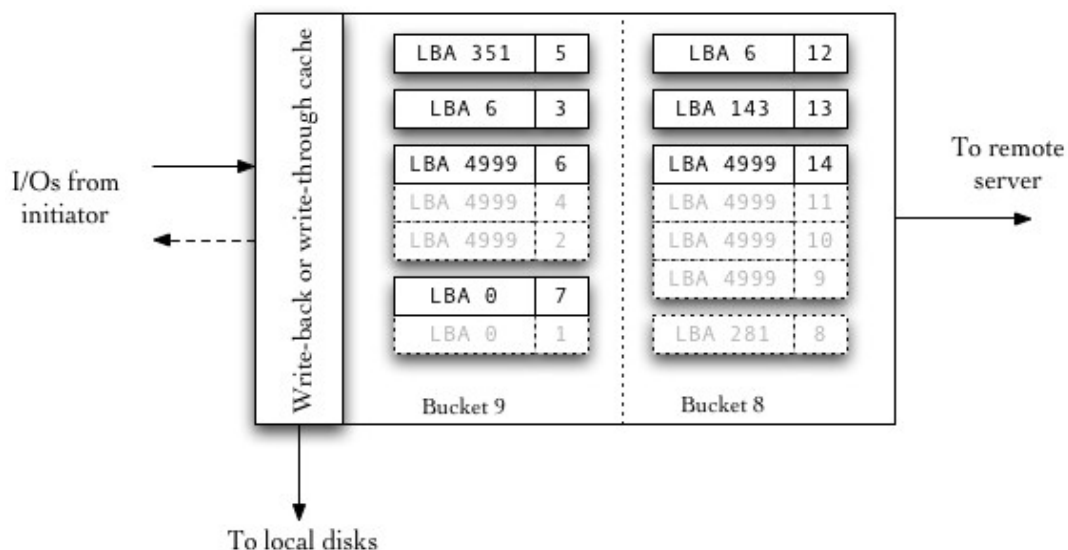


**Abbildung 4: Herkömmliche Methode für die Gewährleistung der Write-Order Fidelity**

StorTrends behandelt die asynchrone Replikation anders. Der asynchrone Replikationsmechanismus ist nahe an einen Zwischenspeicher angebunden. Der Cache sortiert alle I/Os, die vom Initiator kommen, in *Buckets*. Ein Bucket ist eine vollständige Sammlung von I/Os die die Eigenschaft haben, zwischen zwei Zeitpunkten auf dem Server eingetroffen sind. Wenn das System nun drei Buckets hat, die jeweils zum Zeitpunkt nach 10, 20 and 30 Sekunden gebildet wurden, garantieren wir, dass alle I/Os, die auf dem Server zwischen 0 und 10 Sekunden eintrafen in dem ersten Bucket vorhanden sind; alle I/Os die zwischen der 10ten und 20ten Sekunde ankamen befinden sich im zweiten Bucket und alle von der 20ten bis 30ten Sekunde im dritten. Das Ende eines jeden „Körbchens“ ist notwendigerweise ein I/O – konsistenter Punkt – so lange wie der Secondary Server jeweils mit einem Bucket synchronisiert wurde, ist es gewährleistet, das sie niemals einen I/O aufzeichnet, der außerhalb der Ordnung liegt und die Bestände bleiben konsistent und die Replik kann sich jederzeit wiederherstellen.

Dieses Bucketing ist der Grundstein von StorTrends. Zusätzlich zur gegebenen Konsistenz der Daten und der Write Order Fidelity bietet das Bucketing eine

größtmögliche Steigerung der Performance durch Reduktion der doppelten I/Os, die auf das Volumen geschrieben werden. Sich überschneidende I/Os innerhalb eines Buckets werden jeweils abgeglichen, so dass nur der letzte I/O innerhalb des Buckets an den Remote Server übertragen wird. Dies reduziert merklich die benötigte Bandbreite für bestimmte Anwendungen. Auch deshalb, da die Buckets im Memory gespeichert werden, das auch als Cache genutzt wird, entfällt das Kopieren der Daten innerhalb des Memory vom Cache Memory zum Bucket Memory und zurück und so fort. Dies führt zu signifikant niedrigeren Anforderungen an CPU und Memory – Nutzung.



**Abbildung 5: Bucketing für asynchrone I/O**

Zusätzlich zur Senkung der Bandbreite erhöht diese Methode des Bucketing die Nutzung der Bandbreite durch eine weitgehend gleiche Last auf dem Netzwerk. Dies beruht darauf, dass Buckets auch vom Primary zum Secondary übertragen werden dürfen, wenn sie nicht abgeschlossen wurden - jedoch sind sie erst dann auf dem Secondary Volumen nutzbar, wenn sie vollständig übertragen wurden. Diese innovative 'Open-Bucket Transition' Logik erlaubt eine konstante niedrige Ausnutzung der Verbindung.

Die Leistung des Primary wird weiter gesteigert durch die Übertragung eines Buckets, der in einen Rahmen kleine I/Os zusammenfasst und an ihrer Stelle größere I/Os zum Secondary überträgt. Im Ergebnis haben die Prozesse auf dem Primary weniger

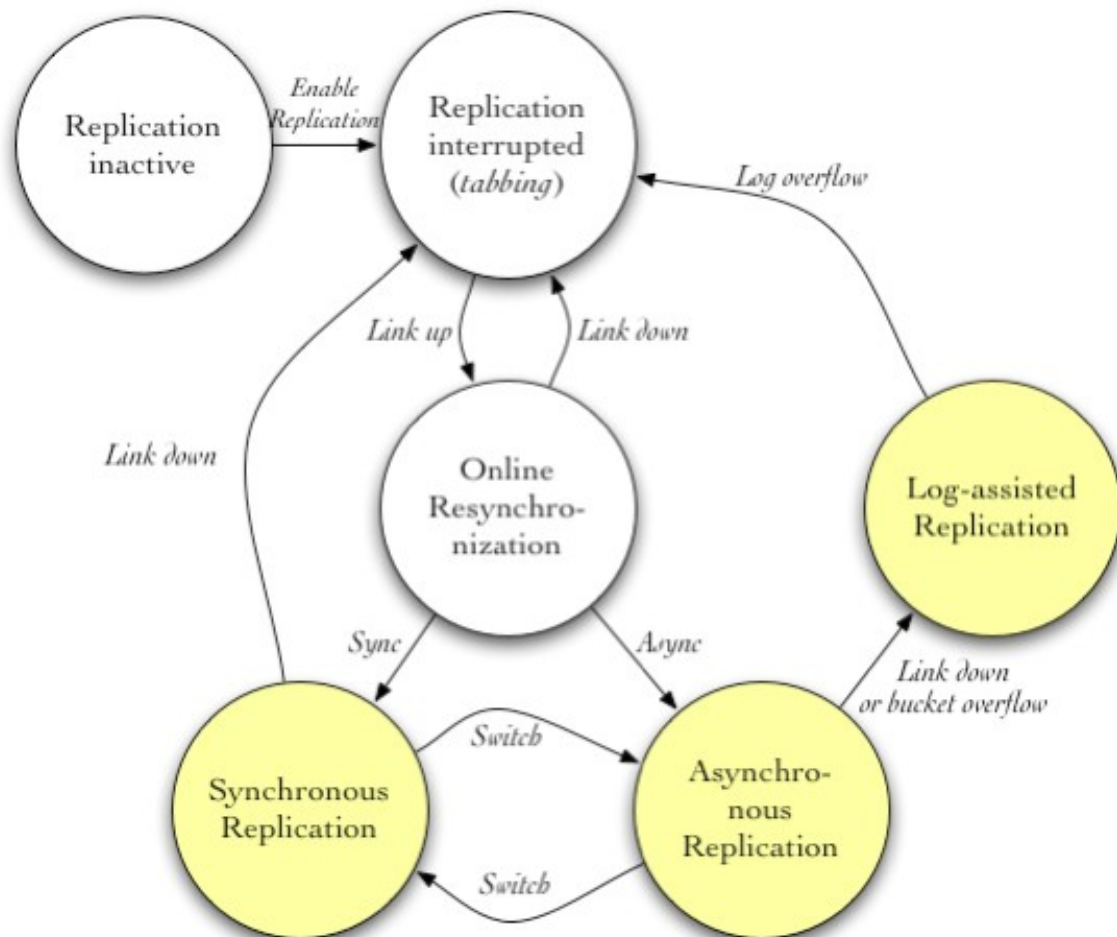
Interrupts als bei einer synchronen Replikation oder anderen Realisierungen der asynchronen Replikation. Eine Reduzierung der Interrupts bewirkt eine erhöhte Effizienz der CPU; wiederum führt eine geringere Last auf der CPU zu einem Ansteigen der Leistung.

Ebenso wie in der Synchronen Replikation kommt in der asynchronen Replikation der Behandlung von Fehlern in der Spannungsversorgung wie auch in der Übertragungsverbindung eine besondere Beachtung zu. Auch hier bietet StorTrends mehrere Innovationen. Eine besondere Aufgabenstellung ist hierbei der „constant lag“, die ständige Verzögerung, der Asynchronen Replikation beim Wiederherstellen des synchronisierten Zustandes nach Fehlern in der Verbindung. Diese tritt immer dann auf, wenn ein „normaler“ asynchroner Modus erneut begonnen wird, nachdem der gesamte Anteil der nicht resynchronisierten Daten übertragen wurde, jedoch die I/Os weiterhin übertragen und angenommen werden. Wenn dies passiert und die Rate der ankommenden I/Os höher ist als die der Resynchronisation, dann wird die Resynchronisation niemals vervollständigt. Eine Alternative der Problemlösung ist es, alle I/Os anzuhalten bis die Resynchronisation abgeschlossen wurde; jedoch kann dies bei großen Volumen dazu führen, dass es zu einer Downtime von mehreren Tagen führt – ein schwer vermittelbare Lösung.

Bei StorTrends wird die Resynchronisation auf eine Art erbracht, die es sicherstellt, dass sie in einer absehbaren Zeit abgeschlossen wird und dass trotzdem I/Os während dieses Prozesses übertragen werden können. Realisiert wird dies durch einen Wechsel in den synchronen Modus während die Resynchronisation läuft. Nach Abschluss der Resynchronisation wechselt das Replikationspaar zurück in den Zustand der asynchronen Replikation. Die Resynchronisation und die Behandlung von Link-Fehlern bei der asynchronen Replikation entsprechen also der bei der synchronen Replikation – mit Beibehalten aller verbundener Vorteile.

Jedoch ist es nicht immer notwendig bei jedem aufgetretenen Link - Failure in einen Link-Fail – Modus zu wechseln. Vorübergehende kurzzeitige Link Failures sind in Netzwerken, die große Flächen und weite Entfernungen abdecken geradezu üblich. Daher ist es möglich diese kurzen sporadischen Störungen in der asynchronen Replikation zu behandeln ohne den Modus zu wechseln. Ein bestimmtes Maß an Störungen muss gegeben sein bevor eine Verbindung als unterbrochen erklärt wird, z.B. nur dann, wenn eine bestimmte Zeitspanne der Verbindungsunterbrechung gegeben ist, wechselt die Replikation in den Link-Fail Modus. Andererseits füllen sich weiterhin die Buckets bis die Verbindung wieder hergestellt wurde und dann kann die

Übertragung der Buckets fortgesetzt werden als wäre die Verbindung nie unterbrochen gewesen.



**Abbildung 6: Wechsel vom asynchronen zum synchronen und log-assisted Modus**

Wie in der synchronen Replikation kann auch in den Modus der Snapshot-Assisted Replikation gewechselt werden, falls eine Verbindung über eine längere Zeitspanne unterbrochen bleibt und die Resynchronisation und die Wiederherstellung vereinfacht werden soll. Zusätzlich zu diesen beiden Verfahren der Wiederherstellung gibt es ein Drittes, das in der asynchronen Replikation angewandt wird, das als „Log-Assisted Replication“ bezeichnet wird. In diesem Verfahren werden ganze „Buckets“ in sequentielle Daten gewandelt und lokal auf der Festplatte gespeichert. Ist die

Verbindung wieder hergestellt, dann werden diese Daten von der lokalen Festplatte gelesen und an den Secondary übertragen. Dieser Vorgang kann einer spezifischen Anwendung angepasst werden, so dass weitere Innovationen wie z.B. Kompression, Batch Processing etc., durchgeführt sind. In der Beschreibung der Snapshot-Assisted Replikation werden die Vorteile dieser Methode detailliert dargestellt.

Ein weiteres wichtiges Merkmal der "Buckets" ist "*Event Bucketing*". Ein Bucket enthält nicht nur eine vollständige Sammlung von I/Os über eine bestimmte Zeit – er beinhaltet auch Ereignisse, „Events“, wie z.B. die Erstellung von Snapshots, das Löschen von Snapshots, etc.. Daher ist es möglich synchrone Snapshots auf beiden, dem Primary und Secondary, Volumen zu erstellen und eine Vielzahl an Management Operationen in einer gleichen synchronisierten Art durchzuführen. Dies erlaubt dem User einen Roll Back nicht nur zu einem I/O konsistenten Zustand, sondern auch zu einem Anwendungs-konsistenten Zustand mit der Garantie einer Wiederherstellung der Anwendung.

Somit liefert StorTrends eine asynchrone Replikation, die nicht nur einige Hürden nimmt, die hier gegeben sind, sondern auch innovative Wege diese zu vereinfachen und auf eine sichere Weise als leistungsfähiges Werkzeug zu nutzen.

© Copyright 1998-2007 American Megatrends, Inc.  
All rights reserved  
American Megatrends, Inc.  
6145-F Northbelt Parkway  
Norcross, GA 30071

© Copyright 1998-2007 American Megatrends International GmbH  
All rights reserved  
American Megatrends International GmbH  
D 81825 München , Wardeinstrasse 3 a

## **TRADEMARK AND COPYRIGHT ACKNOWLEDGMENTS**

This publication contains proprietary information that is protected by copyright. No part of this publication can be reproduced, transcribed, stored in a retrieval system, translated into any language or computer language, or transmitted in any form whatsoever without the prior written consent of the publisher, American Megatrends, Inc.

Trademarks and trade names may be used in this document to refer to either the entities claiming the marks and names or their products. American Megatrends, Inc. disclaims any proprietary interest in trademarks and trade names other than its own.

## **FOR ADDITIONAL INFORMATION**

Call American Megatrends at 1-800-246-8600 for additional information. You can also visit us online at [ami.com](http://ami.com).

## **LIMITATIONS OF LIABILITY**

In no event shall American Megatrends be held liable for any loss, expenses, or damages of any kind whatsoever, whether direct, indirect, incidental, or consequential, arising from the design or use of this product or the support materials provided with the product.

## **LIMITED WARRANTY**

No warranties are made, either express or implied, with regard to the contents of this work, its merchantability, or fitness for a particular use. American Megatrends assumes no responsibility for errors and omissions or for the uses made of the material contained herein or reader decisions based on such use.

## **REVISION HISTORY**

28.03.2007 Preliminary release  
12.12.2007 Revised version

Bitte nehmen Sie für weitere Informationen Kontakt auf mit:  
[europa@ami.com](mailto:europa@ami.com) - [www.ami.de](http://www.ami.de) - [www.ami.com](http://www.ami.com)