



**MICHAEL FUNKE**  
BERATER FÜR SYSTEMSOFTWARE

## Performancesteigerung bei Dateisystembackups

### **Zielgruppe:**

Erfahrene DP-Administratoren.

### **Inhalt:**

Dieses Dokument zeigt Wege auf, wie man bei Dateisystembackups die Laufzeit verkürzen kann.

### **Methodik:**

Auf eine umfassende Analyse des Flaschenhalses wird bewusst verzichtet, da in der Regel bekannt ist wo die Schwachstellen in der Umgebung sind. Hier werden die bekannten Flaschenhälse beschrieben und Lösungsansätze präsentiert.

## Bekannte Flaschenhalse:

### 1. Lesegeschwindigkeit

Die Lesegeschwindigkeit lässt sich mit DP-Bordmitteln wie folgt ermitteln:

Syntax für Linux (Generisch):

```
/opt/omni/sbin/vbda -vol <MOUNTPOINT> -tree <DIRECTORY> -out /dev/null -profile
```

Syntax für Windows (Beispiel):

```
vbda -vol e: -tree e:\test -out nul -profile
```

Dies liest die Festplatte mit der Blockgröße von 8 kB und einem Disk Agent-Buffer und schiebt es nach NULL. Es entsteht keine DP-Session. Die ermittelte Geschwindigkeit liegt etwas unter dem, was bei einem richtigen Backup erreicht wird, da dort eine andere Blockgröße und 8 Buffer genutzt werden.

Wenn man es genauer haben möchte:

Subtrahieren Sie 160 kB von der Blockgröße des Medienagenten, damit BackupRead() mit der gleichen Größe wie der Medienagent ausgeführt wird.

Beispiel:

Eine typische Media Agent-Blockgröße für LTO-Laufwerke beträgt 256 kB.

$256 \text{ kB} - 160 \text{ kB} = 96 \text{ kB} = 98304 \text{ Byte}$

Daraus resultieren dann folgende omnirc Variablen, die temporär zu setzen sind:

Blockgröße: OB2DABLOCKSIZE=98304

Buffer Anzahl: OB2DABLOCKSPERDA=8

In der Praxis stellt man fest, dass der Unterschied zwischen HDD und SSD signifikant ist. Bei einem Faktor 5 bis 10 ist oben genannte Ungenauigkeit nicht relevant, zumal Vergleichsmessungen ohne Anpassung der Parameter erfolgen, da es primär um den direkten Vergleich geht.

Eine Steigerung der Lesegeschwindigkeit kann man erreichen, wenn die Grenzen der Leistungsfähigkeit der Hardware noch nicht erreicht sind, [indem man mehrere Disk Agenten Prozesse parallel startet](#).

## 2. Viele kleine Dateien – Administrativer Overhead

Bei vielen kleinen Dateien ist der Verwaltungsoverhead enorm. Eine kleine Datei zu öffnen und zu schließen, erfordert die gleiche Zeit wie bei einer großen Datei. Dasselbe trifft auf die Sicherung der Dateiattribute zu.

Um diesen Overhead kommen wir nicht drumherum, außer man zieht eine [Disk Image](#)- bzw. [Blockweise](#) Sicherung in Betracht.

Der Unterschied ist:

Feature	Block Based	Disk Image (RAW)
Backup Device	Only StoreOnce / Data Domain	All
Backup Full	✓	✓
Backup Incremental	✓	✗
Backup Used Capacity	✓	✗
Online Backup (VSS)	✓	✓
Object Copy	✓	✓
Single Item Restore	✓	✗
No Tree Walk Needed	✓	✓
Operating System	Windows	Windows / Linux

Einrichten eines Disk Image Backups über „Backup Object Summary“ und „Manual Add“:

Disk image object

Client system: win2022en.fritz.box

Device:

Description: [Disk Image F]: win2022en.fritz.box

Windows:

Disk image sections

\\\\.F:

Linux:

Disk image sections

/dev/raw/raw1

Einrichten eines Block Based Backup:

Show: All systems

- win2022en.fritz.box
  - C:
  - E: [Dedup]
  - F: [Files]
    - \$RECYCLE.BIN
    - dir 1
    - dir 2
    - dir 3

Restore-Ansicht:

Disk Image:

- [[Disk Image F]: win2022en.fritz.box]
  - \\\\.F:

Block Based:

- \\
  - \$RECYCLE.BIN
  - dir 1
  - dir 2
    - 2.txt
  - dir 3

### 3. Viele kleine Dateien – Katalogisieren der Dateinamen

Das Eintragen der Dateinamen in die DP-Datenbank dauert seine Zeit.

Falls eine Rücksicherung von einzelnen Dateien nicht erforderlich, kann man darauf verzichten diese in die DP-Datenbank einzutragen.

Logging Level einstellen:

Options - File System Options - Avanced - Other - Logging

No Log (Nur Mountpoint bzw. Laufwerksbuchstabe)

Log All (Dateinamen inclusive Attribute, Standardwert)

Log Directories (Nur Verzeichnisamen)

Log Files (Dateinamen ohne Attribute)

### 4. Viele kleine Dateien – Statistikinformationen

Das Einsammeln von Statistikinformationen (Treewalk) führt bei vielen kleinen Dateien zu erheblichen Verzögerungen. Sind diese Statistikinformationen nicht erwünscht, kann man den Treewalk für Windows über eine omnirc Variable abschalten.

Size	Done	Done (%)
...	...	...

```
# OB2NOTREEWALK=0|1
```

```
# Default: 0
```

```
# Disables the first treewalk (Windows only). This is only applicable if  
# hard links are not enabled. Setting this variable may improve  
# performance on filesystems with many small files. If set, the progress  
# is no longer displayed in percentage.
```

Unter Linux greift dieser Parameter nicht, dort reicht es die Option „Backup POSIX hard links as files“ auszuwählen. Man sollte sich aber darüber im Klaren sein, was das bedeutet, und die Folgen abschätzen können.

Egal ob Windows oder Linux, die Option „Display statistical info“ darf nicht aktiviert werden, weil es sonst wieder zu einem Treewalk kommt.