Datendeduplizierung mit Bacula

Erol Akman

http://www.dass-it.de

22. September 2010

Übersicht

Was ist Datendeduplizierung (DeDup)?

Dedup – search, destroy, replace!

- Datendeduplizierung ist ein Verfahren zur Eliminierung von identischen Kopien von Daten und Ersetzung der Kopien durch „Links“ auf ein Unikat.

Was ist Datendeduplizierung (DeDup)?

Dedup – search, destroy, replace!

- Datendeduplizierung ist ein Verfahren zur Eliminierung von identischen Kopien von Daten und Ersetzung der Kopien durch „Links“ auf ein Unikat.
- Im Wesentlichen gibt es zwei Arten der Datendeduplizierung

Was ist Datendeduplizierung (DeDup)?

Dedup – search, destroy, replace!

- Datendeduplizierung ist ein Verfahren zur Eliminierung von identischen Kopien von Daten und Ersetzung der Kopien durch „Links“ auf ein Unikat.
- Im Wesentlichen gibt es zwei Arten der Datendeduplizierung

File-Level- und Block-Level-Deduplizierung

Was ist Datendeduplizierung (DeDup)?

Dedup – search, destroy, replace!

- Datendeduplizierung ist ein Verfahren zur Eliminierung von identischen Kopien von Daten und Ersetzung der Kopien durch „Links“ auf ein Unikat.
- Im Wesentlichen gibt es zwei Arten der Datendeduplizierung

File-Level- und Block-Level-Deduplizierung

file-level einzelne ganze Dateien werden indiziert und miteinander verglichen: Datei-Ebene

Was ist Datendeduplizierung (DeDup)?

Dedup – search, destroy, replace!

- Datendeduplizierung ist ein Verfahren zur Eliminierung von identischen Kopien von Daten und Ersetzung der Kopien durch „Links“ auf ein Unikat.
- Im Wesentlichen gibt es zwei Arten der Datendeduplizierung

File-Level- und Block-Level-Deduplizierung

file-level einzelne ganze Dateien werden indiziert und miteinander verglichen: Datei-Ebene

block-level einzelne Dateien werden in Blöcke aufteilt, indiziert und miteinander verglichen: Block-Ebene

File-Level vs. Block-Level

Vor- und Nachteile

File-Level vs. Block-Level

Vor- und Nachteile

File-Level Dateien als Ganzes werden miteinander verglichen und werden ganz oder gar nicht gesichert

Vorteil schont die Systemressourcen

Nachteil bei großen Dateien, die sich im Verhältnis zur Gesamtgröße wenig verändern, sehr ineffektiv

File-Level vs. Block-Level

Vor- und Nachteile

File-Level Dateien als Ganzes werden miteinander verglichen und werden ganz oder gar nicht gesichert

Vorteil schont die Systemressourcen

Nachteil bei großen Dateien, die sich im Verhältnis zur Gesamtgröße wenig verändern, sehr ineffektiv

Block-Level Aufteilung in und Vergleich von Blöcken

Vorteil bei großen Dateien, die sich nur wenig verändern, sehr effektiv

Nachteile geht auf die Systemressourcen, Gefahr von falsch erkannten Duplikaten höher

Wo findet die Datendeduplizierung statt?

Quelle oder Ziel?

Wo findet die Datendeduplizierung statt?

Quelle oder Ziel?

Quelle (Source) Deduplizierung geschieht auf dem Client, dessen Daten gesichert werden sollen

Vorteil reduziert die Netzwerkauslastung

Nachteil fordert mehr Systemressourcen vom Client

Wo findet die Datendeduplizierung statt?

Quelle oder Ziel?

Quelle (Source) Deduplizierung geschieht auf dem Client, dessen Daten gesichert werden sollen

- **Vorteil** reduziert die Netzwerkauslastung
- **Nachteil** fordert mehr Systemressourcen vom Client

Ziel (Target) Dedup geschieht auf dem Storage, der die Daten vom Client empfängt und die Dedup durchführt

- **Vorteil** schont die Systemressourcen
- **Nachteil** höhere Netzwerkauslastung

Wann findet die Target-Datendeduplizierung statt?

Während oder nach dem Backup

Wann findet die Target-Datendeduplizierung statt?

Während oder nach dem Backup

während (in-line) Daten werden bevor Sie geschrieben werden dedupliziert

Vorteil erfordert weniger Festplattenspeicher

Nachteil fordert mehr Systemressourcen vom Storage und ein Backup dauert länger

Wann findet die Target-Datendeduplizierung statt?

Während oder nach dem Backup

während (in-line) Daten werden bevor Sie geschrieben werden dedupliziert

- **Vorteil** erfordert weniger Festplattenspeicher
- **Nachteil** fordert mehr Systemressourcen vom Storage und ein Backup dauert länger

nach (post process) Daten werden erst geschrieben und nachher dedupliziert

- **Vorteil** Backupprozess wird nicht ausgebremst
- **Nachteil** erfordert mehr Festplattenspeicher

Zweck der Deduplizierung

Primär

- Reduzierung von Speicherplatz und
- Netzwerkauslastung

Zweck der Deduplizierung

Primär

- Reduzierung von Speicherplatz und
- Netzwerkauslastung

positive Nebenwirkungen

- spart Zeit beim Backup – Backup-Fenster wird weniger strapaziert
- Daten lassen sich länger vorhalten, mehr Daten lassen sich speichern
- weniger Speicherplatz, weniger Datenverkehr = geringere Kosten

Datendeduplizierung seit der 5.0.0 Version

Bacula und die Datendeduplizierung

- Datendeduplizierung ist sinnvoll
- Anfang 2010 in der Version 5.0.0 offiziell veröffentlicht
- bei dass IT GmbH getestet – es funktioniert, aber
- nicht in der Kombi Director 5.0.3 und Client 5.0.1!

Datendeduplizierung seit der 5.0.0 Version

Bacula und die Datendeduplizierung

- Datendeduplizierung ist sinnvoll
- Anfang 2010 in der Version 5.0.0 offiziell veröffentlicht
- bei dass IT GmbH getestet – es funktioniert, aber
- nicht in der Kombi Director 5.0.3 und Client 5.0.1!

Wie viele von Ihnen . . .

Datendeduplizierung seit der 5.0.0 Version

Bacula und die Datendeduplizierung

- Datendeduplizierung ist sinnvoll
- Anfang 2010 in der Version 5.0.0 offiziell veröffentlicht
- bei dass IT GmbH getestet – es funktioniert, aber
- nicht in der Kombi Director 5.0.3 und Client 5.0.1!

Wie viele von Ihnen . . .

- setzen Baculas Datendeduplizierung produktiv ein?

Datendeduplizierung seit der 5.0.0 Version

Bacula und die Datendeduplizierung

- Datendeduplizierung ist sinnvoll
- Anfang 2010 in der Version 5.0.0 offiziell veröffentlicht
- bei dass IT GmbH getestet – es funktioniert, aber
- nicht in der Kombi Director 5.0.3 und Client 5.0.1!

Wie viele von Ihnen . . .

- setzen Baculas Datendeduplizierung produktiv ein?
- haben Datendeduplizierung schon getestet?

Datendeduplizierung seit der 5.0.0 Version

Bacula und die Datendeduplizierung

- Datendeduplizierung ist sinnvoll
- Anfang 2010 in der Version 5.0.0 offiziell veröffentlicht
- bei dass IT GmbH getestet – es funktioniert, aber
- nicht in der Kombi Director 5.0.3 und Client 5.0.1!

Wie viele von Ihnen . . .

- setzen Baculas Datendeduplizierung produktiv ein?
- haben Datendeduplizierung schon getestet?
- vor diesem Vortrag davon gehört oder gelesen?

Implementierung von Datendeduplizierung (DeDup)

Deduplizierung auf Datei-Ebene

base job ist eine Vollsicherung (Full) eines FileSets

1. für Dedup geeignetes FileSet
2. ein Job, der eine Base-Sicherung macht
3. Client-Jobs benutzen eine oder mehrere Base Sicherungen als Basis

Konfiguration von Datendeduplizierung (DeDup)


```
Job {
 Name = BaseData
 JobDefs = DefaultJob
 Level = Base
}
Job {
 Name = client01
 JobDefs = DefaultJob
 Base = BaseData
 Accurate = yes
}
File Set {
 Name = Data
 Include {
  Options {
    Base Job = s5
    accurate = s5
   }
  File = /data
  }
 }
```

i compare the inodes
p compare the permission bits
n compare the number of links
u compare the user id
g compare the group id
s compare the size
a compare the access time
m compare the modification time (st mtime)
c compare the change time (st ctime)
d report file size decreases
5 compare the MD5 signature

```
Job {
 Name = BaseData
 JobDefs = DefaultJob
 Level = Base
}
Job {
 Name = client01
 JobDefs = DefaultJob
 Base = BaseData
 Accurate = yes
}
File Set {
 Name = Data
 Include {
  Options {
    Base Job = s5
    accurate = s5
   }
  File = /data
 }
}
```

i compare the inodes

p compare the permission bits

n compare the number of links

u compare the user id

g compare the group id

s compare the size

a compare the access time

m compare the modification time (st mtime)

c compare the change time (st ctime)

d report file size decreases

5 compare the MD5 signature

Kleines Szenario

Maschine 1: course (Base Job)

```
root@course:/# ls -gho /data
total 2.1M
-rw-r--r-- 1 1.0M 2010-09-19 15:10 a
-rw-r--r-- 1 1.1M 2010-09-19 15:13 a.tar
```

Maschine 2: client01 (Backup Client1)

```
root@client01:/# ls -gho /data
total 2.1M
-rw-r--r-- 1 1.0M 2010-09-19 15:14 a
-rw-r--r-- 1 1.1M 2010-09-19 15:14 a.tar
-rw-r--r-- 1   10 2010-09-18 16:10 b
-rw-r--r-- 1  10K 2010-09-18 16:11 b.tar
```

Durchführung am kleinen Szenario

Maschine: course (Base Job)

```
/# echo "run job=BaseData yes" | bconsole
...
run job=BaseData yes
...
Job queued. JobId=77
/# echo "run job=client1 yes" | bconsole
...
run job=client1 yes
...
Job queued. JobId=78
```

Ausgabe zum Base Job (messages)


```
JobId 77: Start Backup JobId 77...
  ...
  FD Files Written:        3
  SD Files Written:        3
  FD Bytes Written:        2,103,296 (2.103 MB)
  SD Bytes Written:        2,103,510 (2.103 MB)
   ...
  Termination:             Backup OK
```

Ausgabe zur Deduplizierung (messages)


```
JobId 78: Start Backup JobId 78...
...JobId 78: Using BaseJobId(s): 77
...JobId 78: Sending Accurate information.
...
...JobId 78: Space saved with Base jobs: 2 MB
  ...
  FD Files Written:        5
  SD Files Written:        5
  FD Bytes Written:        10,250 (10.25 KB)
  SD Bytes Written:        10,609 (10.60 KB)
  ...
  Base files/Used files:   3/3 (100.00%)
  ...
  Accurate:                yes
  ...
  Termination:             Backup OK
```

Ergebnis des kleinen Szenario

list files jobid=77

```
+--------------+
| Filename     |
+--------------+
| /data/       |
| /data/a      |
| /data/a.tar  |
+--------------+
```

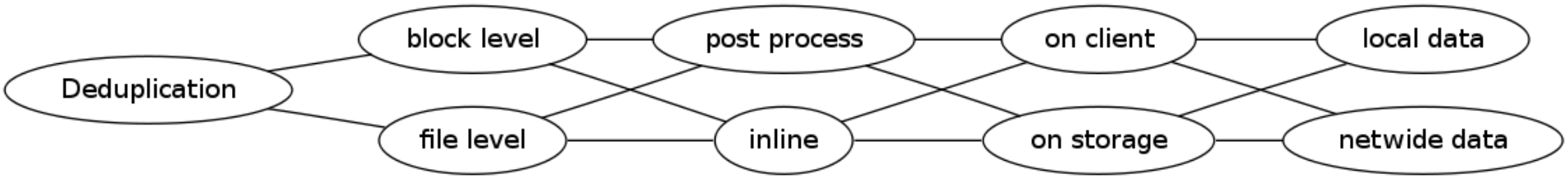
list files jobid=78

```
+--------------+
| Filename     |
+--------------+
| /data/b.tar  |
| /data/b      |
| /data/       |
| /data/a.tar  |
| /data/a      |
+--------------+
```

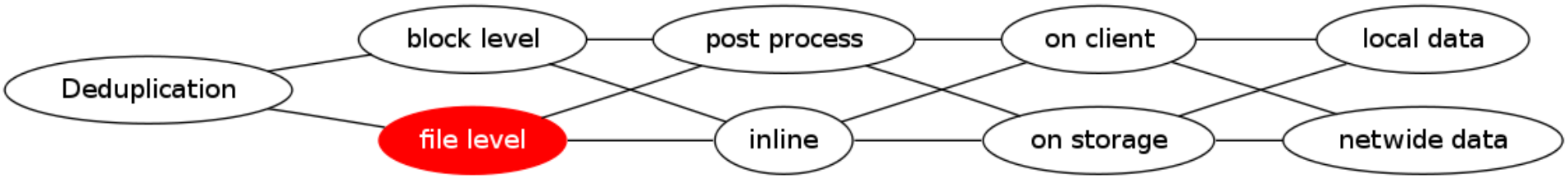
Datenrücksicherung bei Datendeduplizierung


```
restore client=client01-fd
...
Automatically selected FileSet: Full Set
+-------+-------+----------+----------+-----------+----
| JobId | Level | JobFiles | JobBytes | StartTime | Vol
+-------+-------+----------+----------+-----------+----
|    78 | F     |        5 |   10,250 |  14:04:31 | vol
+-------+-------+----------+----------+-----------+----
You have selected the following JobId: 82
The restore will use the following job(s) as Base
+-------+-------+----------+-----------+----------+----
| JobId | Level | JobFiles | JobBytes  | StartTime| Vol
+-------+-------+----------+-----------+----------+----
|    77 | B     |        3 | 2,103,296 | 14:04:20 | vol
+-------+-------+----------+-----------+----------+----
```

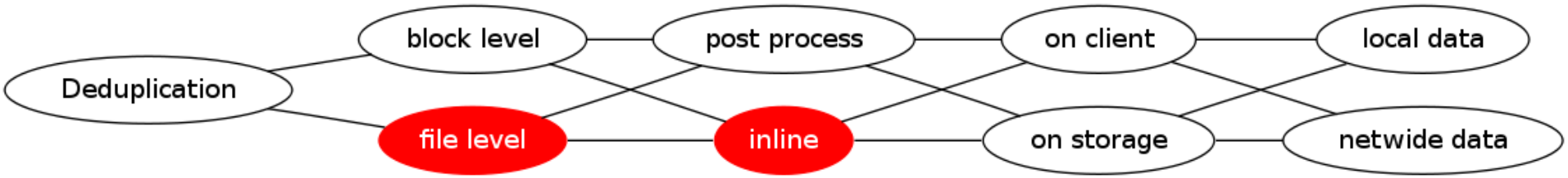
Charaterisierung der Deduplizierung bei Bacula



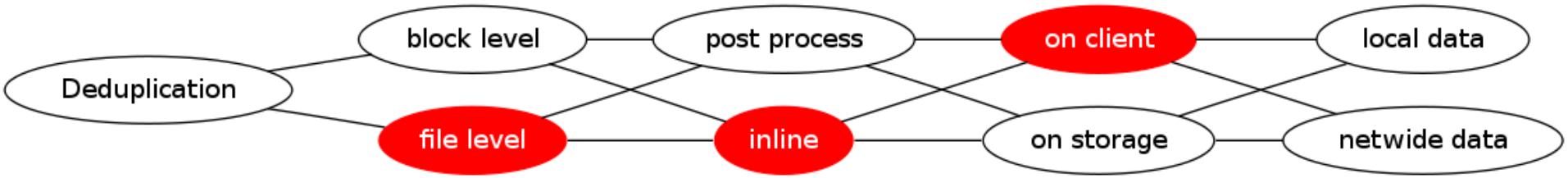
Charaterisierung der Deduplizierung bei Bacula



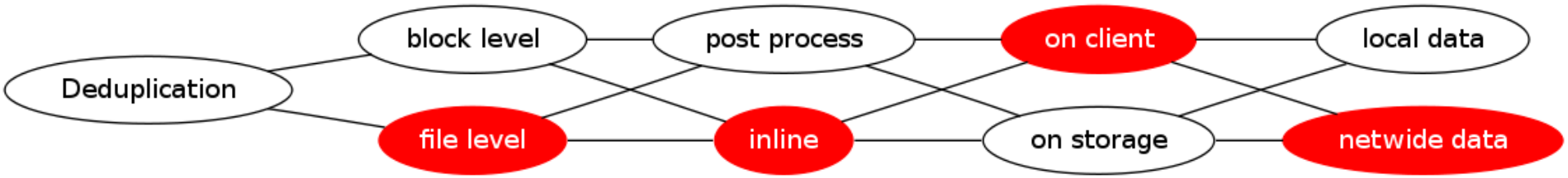
Charaterisierung der Deduplizierung bei Bacula



Charaterisierung der Deduplizierung bei Bacula



Charaterisierung der Deduplizierung bei Bacula



Fragen an Sie

Einsatz von Datenduplikation bei Ihnen?

- Sehen Sie den Nutzen von Datendup in Ihrer Umgebung?
- Wäre die Funktion, wie Sie in Bacula implementiert ist, für Sie sinnvoll?
- Sehen Sie die Möglichkeit Datendup in Ihrer Umgebung einzusetzen?
- Welche Bedingungen verhindern oder begünstigen den Einsatz von Baculas Deduplizierung in Ihrer Umgebung?
- Haben Sie schon Ideen, wie Sie es anwenden können?

Voraussetzungen für Datendeduplizierung (Dedup)

Bacula: file-level in-line on client netwide Deduplication

- der Admin kann einschätzen, welche Verzeichnisse sich für Dedup eignen
- Dateien, die sich selten ändern: Systemverzeichnisse
- keine großen Dateien, bei denen sich wenige Bytes am Tag ändern: Datenbankdateien
- großes Netz, in dem viele identische Dateien auf clients verteilt sind
- bei Dateien, die sich schlecht komprimmieren lassen und/oder sich für eine Block-Level-Dedup nicht eignen
- für Archivierung geeignet

Vielen Dank für Ihre Aufmerksamkeit!

Erol Akman
dass IT GmbH
erol.akman@dass-it.de