Willkommen zum „IBM DB2 Newsletter“

Liebe Leserinnen und Leser,

das erste Quartal von 2013 ist vorbei, Ostern auch. Ich hoffe Sie haben die gefärbten Ostereier im Schnee gut finden können. Auch war die Osterzeit schon eine gute Einstimmung in den April, mit all den Wetterkapriolen.

In diesem Jahr sind mit einigen DB2 Neuigkeiten zu rechnen.

Auch wenn wir es nicht mehr geschafft haben, die Ausgabe im ersten Quartal zu platzieren, wünschen wir wieder viel Spaß beim Lesen und ausprobieren.

Für Fragen und Anregungen unsere Kontaktadresse: db2news@de.ibm.com.

Ihr TechTeam

Inhaltsverzeichnis

NACHTRAG ZU DEN LETZTEN AUSGABEN....2

DIE GSE DB2 LUW WORKING GROUP TAGUNG IN HEIDELBERG....2

ARTIKEL: DB2 TROUBLESHOOTING – DB2STOP HANGS....2

TECHTIPP: MONITORING ARCHIVIERTER LOGFILES....4

AUTOMATISIERTE DATENBANK-REORGANISATION MIT DB2 FÜR LUW 10.1....7

CHATS MIT DEM LABOR....10

SCHULUNGEN / TAGUNGEN / INFORMATIONSVERANSTALTUNG....10

NEWSLETTER ARCHIV....10

ANMELDUNG/ABMELDUNG....11

DIE AUTOREN DIESER AUSGABE....11

Nachtrag zu den letzten Ausgaben

201203

Das Konstrukt wird nicht immer funktionieren. Sie machen gleichzeitig 2 Verbindungen zur Datenbank auf, das unter gewissen Umständen nicht funktioniert.

1. den select der die while Schleife fuellt
2. den alter Table….

Auch wenn die Lösung mit einer Zwischendatei nicht so elegant aussieht, ist dies jedoch die Möglichkeit dieses Problem zu umgehen.

201204

In der Ausgabe 201204 haben wir beschrieben, wie man innerhalb einer Select-Anweisung auch die Satzanzahl der Tabellen auszählen kann. Dies funktioniert leider nicht in einer Multipartitions-Datenbank.
In dieser Datenbank muss der Count der Tabelle weiterhin als seperaten Arbeitsschritt machen.

Die GSE DB2 LUW Working Group Tagung in Heidelberg

Im Dezember trafen sich über 30 Datenbankadministratoren der DB2 LUW Working Group in Heidelberg. Zu dem Treffen konnte der Chairman der Arbeitsgruppe, Herr Nils Kaden, dieses Mal als besonderen Gast Berni Schiefer aus dem DB2 Labor in Toronto begrüßen. Berni Schiefer ist verantwortlich für den Bereich Performance und Benchmarking der DB2-Datenbank für LUW. Berni hat deutsche Wurzeln und spricht sehr gut deutsch.

Themen der Jahrestagung waren unter anderen:

- DB2 und TSA – eine Herausforderung
- DB2 Troubleshooting
- DB2 und WLM – Umsetzung bei Filiadata

Neben dem Wissenstransfer und Informationsaustausch dient die Veranstaltung aber auch der Kontaktpflege unter den Teilnehmern. Als abendliches Rahmenprogramm hatte Herr Kaden für diesen Zweck eine Stadtführung organisiert , die die Teilnehmer in das Jahr 1904 zurückversetzte. Die gesamte Altstadt Heidelbergs war ein riesiges Eisstadion und stellte alle auch ohne Schlittschuhe vor eine Herausforderung der besonderen Art.

Die Tagungen der Arbeitsgruppe sind geprägt von Vorträgen und Diskussionen rund um die täglichen Problemstellungen eines DB2-Administrators. Verdeutlichen kann dieses der nachfolgende Auszug aus dem Vortrag von Thomas Kalb, ITGAIN Consulting GmbH.

Artikel: DB2 Troubleshooting – db2stop hangs

Kennen Sie die Situation? Sie planen eine kurze Downtime ihrer DB2 Instanz zur Aktivierung neuer Einstellungen. Die Frage nach der Downtime beantworten
Sie mit: Nicht mehr als 5 Minuten. Bereits wenige Sekunden, nachdem sie alle Verbindungen geforced, das Kommando „db2stop" eingegeben und noch keine Rückmeldung erhalten haben,

werden sie unruhig.

Warum dauert das Deaktivieren der Datenbank und das Stoppen der Instanz so lange? Welche Aktivitäten müssen noch vor Beendigung durchgeführt werden?

Häufig wird das längere Warten durch das „Aufräumen des Bufferpools" (IOCLEANING) verursacht.

Während der Aufräum-Aktion (IOCLEANING) werden die veränderten Pages („Dirty Pages) auf Platte geschrieben. Dieser Vorgang kann je nach Anzahl der „Dirty Pages" und Effizienz des DB2 IOCLEANINGs mehrere Minuten dauern.

Ein Teil des Vortrags „DB2 TROUBLESHOOTING" gibt Antworten auf folgende Fragestellungen:

- Wie kann ich die Anzahl der Dirty Pages bestimmen?
- Welche Faktoren bestimmen die Anzahl der „Dirty Pages"?
- Wie kann das Aufräumen des Bufferpools (Rausschreiben der Dirty Pages) manuell aktiviert werden?
- Welchen Einfluss hat die Anzahl der „Dirty Pages" und die damit verbundene minimale Bufferpool Log Sequenz Number (MINBUFLSN) auf die Crash Recovery Zeit ?
- Welche vorbereitenden Maßnahmen reduzieren die Zeit für das Deaktivieren einer Datenbank?

Die Anzahl der „Dirty Pages" pro Bufferpool kann mit dem Monitor Kommando „db2pd" mit der Option „-dirtypages" ermittelt werden:

```
~/minbuflsn> db2pd -d sample -dirtypages summary

Database Partition 0 -- Database SAMPLE -- Active -- Up 0 days 05:27:43 -- Date
12/24/2012 01:03:33


Bufferpool: 1
   Dirty pages %       : 0 / 1000 (0.00% dirty)

Bufferpool: 2
   Dirty pages %       : 3442 / 100000 (3.44% dirty)
   Bufferpool minbuflsn: 0000000043E3BFFA

Oldest page info:
DirtyLst   TspID PPNum      ObjID OPNum      Typ UFlag fixcount   wgt CPC LSN
pgLtch
n/a        6     6317       4     6189       0   3     0          1   0   000000
0043E3BFFA 0x00007FF617930E58

Recovery information:
   lowtranlsn              : 00000000451ED3F8
   minbuflsn               : 0000000043E3BFFA
   nextlsn                 : 00000000453120CC
   LFH lowtranlsn          : 00000000451ED3F8
   LFH minbuflsn           : 0000000043D24A66
   Active Log bytes in use : 21848274
   Current Softmax         : 21810380
```

Der erste Bereich der Ausgabe beinhaltet die Anzahl der Dirty Pages pro Bufferpool. Am Ende der Ausgabe werden die Recovery Informationen aufgelistet, die über das Volumen einer Crash Recovery

Die Aktivierung des Aufräumprozesses (IOCLEANING) wird durch drei Schwellwerte bestimmt:

- LSN GAP:
  Das LSN GAP ist die Größe des Protokollbereichs ab dem Protokollsatz mit der ältesten modifizierten Seite, die noch nicht auf Platte geschrieben worden ist. Diese Größe des LSN GAPs wird durch den Parameter SOFTMAX bestimmt. Softmax gibt die Anzahl von Logs als Prozentwert an:
  `(softmax : 100) x logfilsiz x 4096`

- DIRTY PAGE THRESHOLD
  Der Dirty Page Threshold startet das asynchrone Zurückschreiben.
  Hat der Anteil der veränderten Seiten im Bufferpool den Schwellwert des Datenbank-Konfigurationsparameters CHNGPGS_THRESHOD überschritten, so startet das System die IOCLEANER, die die veränderten Seiten asynchron auf die Platten schreiben.

- DIRY PAGE STEALS
  Reicht der Platz im Bufferpool für weitere Einlagerungen nicht aus, so müssen die veränderten Pages synchron auf Platte geschrieben werden, um Platz für neue Page Anforderungen zu schaffen.

Das synchrone Zurückschreiben (Dirty Page Steals) der Pages hat erhebliche negative Auswirkungen auf die Performance und sollte deshalb vermieden werden.

```
pool_drty_pg_steal_clns_ratio:
pool_drty_pg_steal_clns / (pool_drty_pg_steal_clns + pool_drty_pg_thrsh_clns + pool_lsn_gap_clns)
```

Das Aufräumen des Bufferpools (Rausschreiben der Dirty Pages) kann auch über folgende Kommandos aktiviert werden.
Ab der Version 9.1 kann mit dem Kommando „`db2pdcfg -d <DBNAME> –flushbp`“ das asynchrone Rausschreiben der Dirty Pages aktiviert werden. Mit „`db2pdcfg -d <DBNAME> –flushbp q`“ wird das
Rausschreiben überwacht.
Das Kommando „`flush bufferpool all`“ ab der Version 10.1 schreibt synchron die Dirty Pages auf Platte.

Die Crash Recovery einer Datenbank besteht aus der Backward Recovery Phase und der Forward Recovery Phase.
Innerhalb der Backward Recovery werden die noch nicht festgeschriebenden Transaktionen zurückgenommen. Der Wert LowTransLSN ist der Wert für die älteste noch nicht festgeschriebene Transaktion.
Während der Forward Recovery werden die vor dem Crash noch nicht zurückgeschriebenen Pages über die Log Informationen auf Platte geschrieben. Der Wert MinBufLSN ist der Wert für die älteste Dirty Page im Bufferpool.
Die Anzahl der Dirty Pages und die Position innerhalb der Logs (MinBufLSN) bestimmen die Zeit der Forward Recovery Phase.

Damit die Zeit zum Deaktivieren der Datenbank und zum Stoppen der
Instanzen möglichst gering und kalkulierbar gehalten werden kann, können folgende vorbereitende Aktivitäten durchgeführt werden:

1. Log Switch: archive logs for db <dbname>
2. Rausschreiben der Dirty Pages V9.1: `db2pdcfg -d <DBNAME> –flushbp`
   V10.1: `flush bufferpool all`
3. Forcen aller Verbindungen: `force application all`

TechTipp: Monitoring archivierter Logfiles

Die regelmäßige Sicherung von DB2 Datenbanken ist essentiell, um im Falle eines Hardwareproblems, Datenbankabsturzes oder Anwendungsfehler, eine Datenbank sicher wieder herstellen zu können. Um eine Rollforward Recovery to End Of Logs oder to Point in

Time durchführen zu können, benötigt man neben dem DB2 Backup Image auch die entsprechenden Logfiles.
Dazu stellt DB2 Möglichkeiten zur Verfügung, die aktuell nicht mehr benötigten Transaktionslogs zu sichern, genauer gesagt, zu archivieren. Es ist wichtig, nicht mehr benötigte Logfiles aus dem aktuellen Logpath in eine andere Lokation zu verschieben, um einen Overflow des Logpaths zu vermeiden. Der Datenbankmanager steuert dabei den Zeitpunkt der Archivierung automatisch.
Mit den zwei Datenbankparametern LOGARCHMETH1 und LOGARCHMETH2 kann der Datenbankadministrator (DBA) das Ziel der Archivierung definieren. Dabei sind drei Methoden erlaubt:

- DISK
- TSM
- VENDOR.

Die in früheren DB2 Versionen unterstütze Methode USEREXIT sollte nicht mehr verwendet werden.
Bei der Archivierung nach der Methode DISK wird in der Regel ein separates Filesystem, im Windows Betriebssystem ein Laufwerk, mit ausreichender Größe definiert, um Logfiles über einen längeren Zeitraum aufnehmen zu können.
Für die Sicherung nach TSM (Tivoli Storage Manager) wird eine eigene Management Klasse definiert, nach der archiviert wird.
Für das jeweilige Vendor-Produkt ist die Dokumentation des Herstellers zu beachten.

Nun kann es trotz aller Sorgfalt passieren, dass Logfiles nicht sofort archviert werden können. Die Ursachen können ein fehlerhaftes oder abgehängtes Filesystem oder Schnittstellenprobleme zum TSM bzw. Vendorprodukt sein. Der DB2 Datenbankmanager schreibt in diesem Fall eine Meldung in das db2diag.log File und wird weitere Versuche unternehmen, die Archivierung erfolgreich durchzuführen (siehe auch die Beschreibung der Parameter NUMARCHRETRY und ARCHRETRYDELAY).
Bei einer großen Anzahl von Datenbank Instanzen und Datenbanken wird es für den DBA zunehmend aufwändiger zu monitoren, ob auch wirklich alle Logfiles archiviert wurden. Man kann natürlich mittels des DB2-Tools db2diag bzw. durch geschicktes Parsen des db2diag.log Files (UNIX Systeme only) die entsprechenden Informationen gewinnen.
Dazu ein Beispiel:

```
db2diag -g 'funcname:=sqlpgArchiveLogFile' -l Severe,Error
```

```
2012-08-10-13.28.14.342820+120 E82946A557         LEVEL: Error
PID     : 10813560             TID  : 3086        PROC : db2sysc 0
INSTANCE: db2id004             NODE : 000
EDUID   : 3086                 EDUNAME: db2logmgr (SAMPLE) 0
FUNCTION: DB2 UDB, data protection services, sqlpgArchiveLogFile, probe:3160
MESSAGE : ZRC=0x86100025=-2045771739=SQLP_MEDIA_VENDOR_DEV_ERR
          "A vendor device reported a media error."
DATA #1 : <preformatted>
Failed to archive log file S0000015.LOG to TSM chain 0 from
/db2data/db2id004/NODE0000/SQL00011/SQLOGDIR/.
```

Praktischer ist es aber, die Table-Function PD_GET_DIAG_HIST zu benutzen. Mit dieser Function kann man gezielt Records aus DB2 Diagnose-Dateien, wie dem db2diag.log oder dem db2optstats.xxxx.log, auslesen. Ein weiterer Vorteil ist, dass man sich nur an eine Datenbank einer Instanz verbinden muss, um die Informationen aus allen Datenbanken der Instanz zu erhalten.

Anm.: Die Table-Function beinhaltet SQL, und SQL bedingt eine Verbindung zu einer Datenbank. Ausführliche Informationen zur Funktion sind hier zu findet.

Das SQL sieht wie folgt aus:

```
-- Find all Logfiles, that failed the first time for archiving
-- Get the Timestamp, WHEN they are archived successfuly
-- 31.08.2012 gm IBM
```

```
WITH PDDIAGHIST (DB2LOGFILE, PROCESS_NAME, PD_DBNAME)
AS (SELECT DISTINCT(CAST(SUBSTR(SUBSTR(MSG,1,52),39,12) AS CHAR(12))) AS DB2LOGFILE,
SUBSTR(PROCESS_NAME,1,20) AS PROCESS_NAME,
CAST(SUBSTR(PROCESS_NAME,12,
CASE
WHEN POSSTR (PROCESS_NAME,')' ) = 19 THEN 7
WHEN POSSTR (PROCESS_NAME,')' ) = 18 THEN 6
WHEN POSSTR (PROCESS_NAME,')' ) = 17 THEN 5
WHEN POSSTR (PROCESS_NAME,')' ) = 16 THEN 4
WHEN POSSTR (PROCESS_NAME,')' ) = 15 THEN 3
WHEN POSSTR (PROCESS_NAME,')' ) = 14 THEN 2
WHEN POSSTR (PROCESS_NAME,')' ) = 13 THEN 1
ELSE 8
END) AS VARCHAR(8)) AS PD_DBNAME
FROM TABLE (PD_GET_DIAG_HIST( 'MAIN', 'ALL', '', CAST (NULL AS TIMESTAMP), CAST (NULL AS
TIMESTAMP) ) ) AS T
WHERE T.FUNCTION = 'sqlpgArchiveLogFile'
AND PROCESS_NAME LIKE 'db2logmgr%'
AND LEVEL = 'W'),
DBSNAPDB (DBSNAPDB)
AS (SELECT SUBSTR(DB_NAME,1,8) AS DB_NAME FROM SYSIBMADM.SNAPDB)
SELECT T.PD_DBNAME AS DATABASE,
S.EID,
SUBSTR(S.FIRSTLOG,1,12) AS FIRSTLOG,
CASE S.OPERATIONTYPE
WHEN '1' THEN 'ArchiveM1'
WHEN '2' THEN 'ArchiveM2'
WHEN 'N' THEN 'Forced Archive'
WHEN 'P' THEN 'Primary LogPath'
END AS OPERATIONTYPE,
S.START_TIME,
S.END_TIME,
S.DEVICETYPE,
T.PROCESS_NAME
FROM SYSIBMADM.DB_HISTORY S
JOIN PDDIAGHIST T
ON SUBSTR(S.FIRSTLOG,1,12) = T.DB2LOGFILE
JOIN DBSNAPDB P
ON P.DBSNAPDB = T.PD_DBNAME
WHERE S.OPERATION = 'X'
ORDER BY EID;
```

Hier ist ein Beispiel für das Ergebnis

```
DATABASE EID FIRSTLOG OPERATIONTYPE START_TIME END_TIME DEVICETYPE PROCESS_NAME

SAMPLE 67 S0000033.LOG ArchiveM1 20120829104121 20120829105007 A db2logmgr (SAMPLE) 0
SAMPLE 69 S0000035.LOG ArchiveM1 20120829114349 20120829114619 A db2logmgr (SAMPLE) 0
SAMPLE 70 S0000036.LOG ArchiveM1 20120829114429 20120829114620 A db2logmgr (SAMPLE) 0
SAMPLE 72 S0000038.LOG ArchiveM1 20120829152647 20120829164532 A db2logmgr (SAMPLE) 0
SAMPLE 73 S0000039.LOG ArchiveM1 20120829152858 20120829164533 A db2logmgr (SAMPLE) 0
SAMPLE 74 S0000040.LOG ArchiveM1 20120829155013 20120829164533 A db2logmgr (SAMPLE) 0
SAMPLE 76 S0000042.LOG ArchiveM1 20120830110807 20120830113827 A db2logmgr (SAMPLE) 0
SAMPLE 81 S0000045.LOG Primary LogPath 20120831102854 -         D db2logmgr (SAMPLE) 0


DATABASE EID FIRSTLOG OPERATIONTYPE START_TIME END_TIME DEVICETYPE PROCESS_NAME

SAMPLE99 60 S0000018.LOG ArchiveM1 20120829162522 20120829164612 A db2logmgr (SAMPLE99)
SAMPLE99 63 S0000020.LOG ArchiveM1 20120830111057 20120830114011 A db2logmgr (SAMPLE99)
SAMPLE99 65 S0000021.LOG ArchiveM1 20120830112637 20120830114012 A db2logmgr (SAMPLE99)
SAMPLE99 67 S0000022.LOG Primary LogPath 20120831102948 - D db2logmgr (SAMPLE99)

DATABASE EID FIRSTLOG OPERATIONTYPE START_TIME END_TIME DEVICETYPE PROCESS_NAME

GERD 18 S0000011.LOG ArchiveM1 20120830111322 20120830114210 A db2logmgr (MARC) 0
GERD 20 S0000012.LOG ArchiveM1 20120830112833 20120830114210 A db2logmgr (MARC) 0
GERD 26 S0000016.LOG Primary LogPath 20120831103040 - D db2logmgr (MARC) 0
```

Die Spalte OPERATIONTYPE gibt an, ob das Logfile archiviert wurde, entsprechend der Logarchive Methode 1 oder 2. Beinhaltet die Spalte 'Primary Logpath' bedeutet das, dass das Logfile noch nicht archiviert wurde, es steht noch im Logpath. In diesem Beispiel gab es für die Instance db2inst1 ein Schnittstellenproblem zum TSM. Der Type 'Forced Archive' bedeutet, dass das Logfile durch ein `db2 "ARCHIVE LOG FOR DATABASE SAMPLE"` geschlossen wurde.

Durch einen weiteren Aufruf des SQL Scripts kann dann jederzeit geprüft werden, ob ein noch

nicht gesichertes Logfile inzwischen archiviert wurde.
Da die Table Function PD_GET_DIAG_HIST die Informationen aus den Diagnosedateien der Instance liest, kann sie auch nur Daten verarbeiten die noch in diesen Dateien vorhanden sind.

Automatisierte Datenbank-Reorganisation mit DB2 für LUW 10.1

Datenbank-Objekte wie Tabellen oder Indizes mit häufigen INSERT-, UPDATE- und DELETE-Operationen fragmentieren im Laufe der Zeit. Reorganisation optimiert das physikalische Layout der Datenbank-Objekte wie Tabellen oder Indizes.
Folgende ursächlichen Ziele werden bei einer klassischen Reorganisation verfolgt:

- Speicherplatzfreigabe
- Optimierung von I/O
- Komprimierung von Tabellen und Indizes (Neukomprimierung und/oder Optimierung der Objekte

Reorganisationen sollten nur durchgeführt werden, wenn dadurch messbare und damit auch signifikante Vorteile zu erkennen sind. Reorganisationen belasten das System durch erhöhtes I/O Aufkommen, CPU-Bedarf und durch eine eingeschränkte Verfügbarkeit. Unvermeidbar ist auch ein erhöhtes Protokoll-Aufkommen durch die Reorganisation und kurzfristiger benutzter Speicherplatzbedarf.

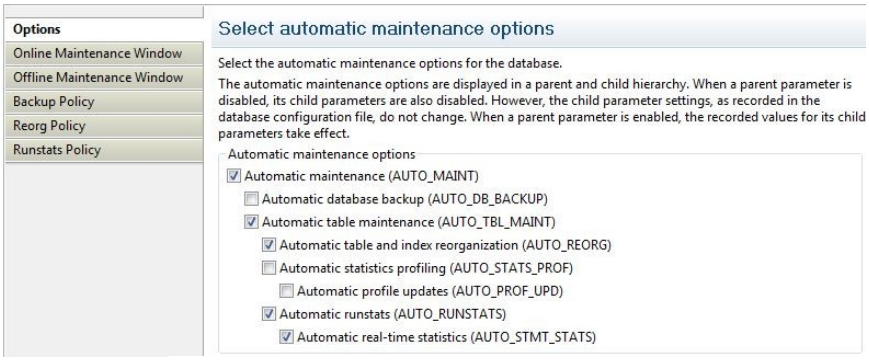
Reorganisationsmethoden:

- Der klassische OFFLINE REORG (REORG TABLE …)
- Der ONLINE REORG (REORG TABLE … INPLACE)
- Der ONLINE INDEX REORG (REORG INDEX … ALLOW WRITE ACCESS)
- Stored Procedure (ADMIN_MOVE_TABLE), neu ab DB2 V10.1 FP2 mit RI

Neben der klassischen Reorganisation gibt es auch sogenannte Pflegemaßnahmen für DB2-Objekte:

- Entfernen von committed pseudogelöschten Schlüsseln in Indizes (REORG … CLEANUP)
  Dabei wird der Index nicht wieder neu aufgebaut, sondern alle freiwerdenden pages stehen den Indizes dieser Tabelle wieder zur Verfügung.
- Freigabe leerer Extents nach Delete-Operationen (REORG … RECLAIM EXTENTS)
  Dieser Prozess reorganisiert und erstellt leer werdende Blöcke durch verschieben von pages, um diese dem Tablespace wieder zur Verfügung zu stellen.

Diese Pflegemaßnahmen können unter Verwendung von (DB2 Auto Reorg) automatisiert werden. Folgende Voraussetzungen müssen erfüllt sein:
Das Online/Offline Fenster ist entsprechend den Vorgaben im Automatic Maintenance mittels Data Studio oder einer Policy definiert bzw. konfiguriert und die in der Abbildung gezeigten Datenbankkonfigurationsparameter wurden aktiviert. Wichtig ist hierbei, dass ein 24 Stunden online Fenster definiert ist:



Automatic Reorg Policy(Autonomic Feature)

Die Automatic reorg policy regelt, in welchem Betrieb (online/offline), die DB2-Objekte behandelt werden. Die Policy selbst ist ein XML-Dokument und wird mithilfe von DB2 Werkzeugen eingerichtet (Data Studio oder DB2 CLP).
Es gibt Beispiel XML Dateien im /sqllib/samples/automaintcfg Verzeichnis, welche als Referenz verwendet werden um an die Wartung angepasste XML Dateien zu erstellen.

```
DB2MaintenanceWindowPolicySample.xml
DB2AutoReorgPolicySample.xml
DB2AutoRunstatsPolicySample.xml
DB2AutoBackupPolicySample.xml
```

Die unten aufgeführten Elemente in diesem XML-Dokument steuern entsprechend der Werte das Verhalten des DB2 Auto Reorg.
Wichtig: Die Beschreibung der Elemente stehen in den entsprechenden Beispiel XML Dateien, aber nicht im Information Center.

DB2AutoReorgPolicy.xml:

```
<?xml version="1.0" encoding="UTF-8"?>
<DB2AutoReorgPolicy
   xmlns="http://www.ibm.com/xmlns/prod/db2/autonomic/config">
  <ReorgOptions
   dictionaryOption="Rebuild"
   indexReorgMode="Online"
   useSystemTempTableSpace="true"
   reclaimExtentsSizeForTables="10000"
   reclaimExtentsSizeForIndexObjects="1000"/>
 <ReorgTableScope
   maxOfflineReorgTableSize="1000000">
    <FilterClause>
     TABSCHEMA NOT LIKE 'SYS%' and (tabschema, tabname) not in
     (select tabschema, tabname from syscat.eventtables)
    </FilterClause>
 </ReorgTableScope>
</DB2AutoReorgPolicy>
```

Beschreibung der Elemente:

- dictionaryOption="Rebuild": Neuaufbau des compression dictionary bei offline table reorg
- indexReorgMode="Online": Index reorg nur online
- useSystemTempTableSpace="true": Bei einem offline table reorg wird der temporary tablespace verwendet
- reclaimExtentsSizeForTables="10000": reclaim für MDC,- ICT Tabellen nur wenn mindestens 10MB Speicherplatz frei wird
- reclaimExtentsSizeForIndexObjects="1000": reclaim für Indizes nur wenn mindestens 1MB Speicherplatz frei wird
- maxOfflineReorgTableSize="1000000": offline reorg nur bis max. 1GB Tabellengröße und online reorg nur für Tabellen bis 1GB. CLEANUP wird aber

ausfgeführt.
- FilterClause: Systemtabellen und Eventmonitor Tabellen sind davon ausgeschlossen

Um die Policy zu aktivieren, muss das angepasste XML Dokument nach /sqllib/tmp kopiert und anschließend mit der sysproc.automaint_set_policyfile Prozedur aufgerufen werden:

```
cp $HOME/autoreorg_policy.xml $HOME/sqllib/tmp/.
db2 "call sysproc.automaint_set_policyfile( 'AUTO_REORG', ' DB2AutoReorgPolicy.xml ')"
```

Nach der erfolgreichen Ausführung findet sich folgender Eintrag im db2diag.<n>.log und im <instance>.<n>.nfy:

```
2013-01-28-12.40.05.938275+060 E8297983A1627        LEVEL: Warning
PID     : 13238476             TID : 772            PROC : db2fmp (C) 0
INSTANCE: db2bet               NODE : 000           DB   : BET
APPID   : *LOCAL.db2bet.121221153811
HOSTNAME: r32l06
EDUID   : 1286                 EDUNAME: db2fmp (C) 0
FUNCTION: DB2 UDB, Policy, sqlpiAutomaintSetPolicySpCommon, probe:10
MESSAGE : ADM10515I  The automatic maintenance policy "AUTO_REORG" has been
          updated in database "BET". The policy options have been updated from
          "Table Scope:     TABSCHEMA NOT LIKE 'SYS%' and (tabschema, tabname)
          not in    (select tabschema, tabname from syscat.eventtables)
          Reorganization options: Rebuild dictionary , Use temporary tablespace
          Index Reorganization Mode: Online  Reclaim Extents Size for Tables:
          10000 KB  Reclaim Extents Size for Index Objects: 1000 KB  Table size
          limit: 1000000 KB" to "Table Scope:     TABSCHEMA NOT LIKE 'SYS%' and
          (tabschema, tabname) not in    (select tabschema, tabname from
          syscat.eventtables)     Reorganization options: Rebuild dictionary ,
          Use temporary tablespace  Index Reorganization Mode: Online  Reclaim
          Extents Size for Tables: 10000 KB  Reclaim Extents Size for Index
          Objects: 1000 KB  Table size limit: 1000000 KB".
```

Analog dazu liest sysproc.automaint_get_policyfile Stored Procedure die automatic maintenance configuration der Datenbank und schreibt diese in das /sqllib/tmp/ Verzeichnis :

```
db2 "call sysproc.automaint_get_policyfile( 'AUTO_REORG', ' DB2AutoReorgPolicy_Get.xml ')"
```

Anhand eines Beispiels wird der DB2 Auto Reorg überprüft:

```
SELECT TABSCHEMA, TABNAME, INDNAME, INDEXTYPE
  FROM SYSCAT.INDEXES
  WHERE TABSCHEMA = 'DB2BET' AND TABNAME = 'STUDENT';

TABSCHEMA TABNAME INDNAME      INDEXTYPE
 --------- ------- ------------ ---------
 DB2BET    STUDENT STUDENT_IDX  REG
 DB2BET    STUDENT STUDENT_IDX1 REG
```

Dabei stellen wir fest, dass nach einem erfolgreichen Löschvorgang, der Speicherplatz freigegeben wurde. Benutzt wird dabei die ADMIN_GET_INDEX_INFO Stored Procedure:

```
DELETE FROM STUDENT WHERE PROGRAM_ID = 3;

Updated 2.000.066 rows.
```

```
SELECT RTRIM(INDSCHEMA) || '.' || RTRIM(INDNAME) AS INDEX, INDEX_OBJECT_P_SIZE, RECLAIMABLE_SPACE
  FROM TABLE (SYSPROC.ADMIN_GET_INDEX_INFO('T', 'DB2BET', 'STUDENT')) AS T;

INDEX               INDEX_OBJECT_P_SIZE RECLAIMABLE_SPACE
 ------------------- ------------------- -----------------
 DB2BET.STUDENT_IDX               194560              7680
 DB2BET.STUDENT_IDX1              194560              7680
```

Wichtig: Die Anzeige ist in diesem Fall irreführend. Beide Indizes zusammen sind 190 MB groß und 7MB können wieder freigegeben werden.

DB2 löscht nicht sofort die korrespondierenden Schlüsseleinträge in den Indizes. Diese Indizes werden als „pseudo-deleted" markiert.

Manuell ist es möglich, mit dem u.a. DB2-Command die „pseudo-deleted" Schlüssel zu entfernen und den Speicherplatz wieder freizugeben(CLEANUP ALL RECLAIM EXTENTS):

```
CALL SYSPROC.ADMIN_CMD ('REORG INDEXES ALL FOR TABLE DB2BET.STUDENT ALLOW WRITE ACCESS CLEANUP ALL
RECLAIM EXTENTS');
```

Hier setzt aber der DB2 Auto Reorg an:

```
2013-01-28-14.38.08.415738+060 I8319073A373         LEVEL: Event
PID     : 15401058             TID : 1974           PROC : db2acd 0
INSTANCE: db2bet               NODE : 000
HOSTNAME: r32106
EDUID   : 1974                 EDUNAME: db2acd 0
FUNCTION: DB2 UDB, Health Monitor, db2HmonEvalReorg, probe:10
START   : Automatic reorg evaluation has started on database BET

2013-01-28-14.38.08.572923+060 I8319447A389         LEVEL: Event
PID     : 15401058             TID : 1974           PROC : db2acd 0
INSTANCE: db2bet               NODE : 000
HOSTNAME: r32106
EDUID   : 1974                 EDUNAME: db2acd 0
FUNCTION: DB2 UDB, Health Monitor, db2HmonEvalReorg, probe:1230
STOP    : Automatic reorg evaluation has finished successfully on database BET
```

```
SELECT RTRIM(INDSCHEMA) || '.' || RTRIM(INDNAME) AS INDEX, INDEX_OBJECT_P_SIZE, RECLAIMABLE_SPACE
  FROM TABLE (SYSPROC.ADMIN_GET_INDEX_INFO('T', 'DB2BET', 'STUDENT')) AS T;

INDEX               INDEX_OBJECT_P_SIZE RECLAIMABLE_SPACE
 ------------------- ------------------- -----------------
 DB2BET.STUDENT_IDX               185344                 0
 DB2BET.STUDENT_IDX1              185344                 0
```

DB2 Auto Reorg ist ein probates Mittel um automatisierte Pflegemaßnahmen auf DB2 Objekte durchzuführen. Klassische und arbeitsintensive Tätigkeiten entfallen und werden nach und nach durch autonome Funktionen innerhalb der Datenbank durchgeführt.
IBM DB2 10.1: self-configuring | self-healing | self-optimizing | self-protecting.

Quelle:

- IBM DB2 Version 10.1 Information Center
- SAP Community Network(SCN)
- SAP Hinweis 975352

Auch im Blog von Henrik Loeser wurde dieses Thema aufgegriffen.

Chats mit dem Labor

Eine Liste der bereits durchgeführten Chats ist hier zu finden.
Die Präsentationen der Chats können dort angeschaut und heruntergeladen werden.

Schulungen / Tagungen / Informationsveranstaltung

Eine Liste der anstehenden Konferenzen ist hier zu finden.

Newsletter Archiv

Wir haben ein weiteres Archiv für den DB2 Newsletter bei ORDIX.

Alte Ausgaben vom DB2-NL sind nun zum Nachlesen in den Archiven zu finden von:

- Lis.Tec
- Bytec
- Drap
- Cursor Software AG
- ids-System GmbH

Anmeldung/Abmeldung

Sie erhalten diesen Newsletter bis zur 3ten Ausgabe ohne Anmeldung. Wenn Sie weiterhin diesen Newsletter empfangen wollen, schicken Sie Ihre Anmeldung mit dem Subject „ANMELDUNG" an db2news@de.ibm.com.

Die Autoren dieser Ausgabe

Sollten Sie Anfragen zu den Artikeln haben, können Sie sich entweder direkt an den jeweiligen Autor wenden oder stellen Ihre Frage über den DB2 NL, denn vielleicht interessiert ja die Antwort auch die anderen DB2 NL Leser.

Doreen Stein	Master Certified IT-Spezialist für DB2 LUW, IBM SWG; Chief-Editor DB2NL, djs@de.ibm.com
Thomas Kalb	CEO, ITGAIN #Artikel: DB2 Troubleshooting – db2stop hangs|outline
Gerhard Müller	Senior IT-Spezialist

Reviewer und Ideenlieferanten:

Wilfried Hoge	IBM SWG
Dirk Fechner	IBM SWG