

Willkommen zum „IBM Informix Newsletter“

Inhaltsverzeichnis

Aktuelles.....	1
TechTipp: HASH ON - forest of trees index.....	2
TechTipp: onstat -g ses 0 – Snapshot für alle aktiven Sessions.....	3
TechTipp: Informix Warehouse Accelerator: CPU-Nutzung konfigurieren.....	4
TechTipp: Extent Size / Next Size beim Erstellen von Indices.....	6
TechTipp: Statement-Typ im onstat -k.....	6
TechTipp: INFORMIX Open Source Projekte.....	7
TechTipp: OFF_RECVRY_THREADS und HDR/RSS.....	8
TechTipp: ON_RECVRY_THREADS.....	8
WebTipp: Stress Tests auf Basis eines TCP-C Benchmarks unter Linux.....	8
Anmeldung / Abmeldung / Anmerkung.....	9
Die Autoren dieser Ausgabe.....	9

Aktuelles

Liebe Leserinnen und Leser,

der Maibaum hat uns inspiriert, Ihnen die Vorteile des Indexbaumes als „Forest of trees“ näher vorzustellen. Wichtig ist hierbei auch, dass nunmehr die Extent Size von Indizes unabhängig von der Extent Size der Tabelle eingestellt werden kann.

Beim INFORMIX Warehouse Accelerator tut sich so viel, dass wir (fast) jeden Monat über weitere Verbesserungen der Funktionalität oder der Benutzerfreundlichkeit berichten dürfen.

Ebenso erfreulich sind die Ergebnisse, die ein Stress Test auf Basis der Regeln des TCP-C Benchmarks für INFORMIX erbracht hat.

Tipp: Der INFORMIX Newsletter eignet sich hervorragend als spannende Lektüre im Biergarten bis die Kollegen eintreffen.



Wie immer haben wir für Sie eine Reihe an Tipps und Tricks zusammengestellt. Viel Spaß mit den Tipps der aktuellen Ausgabe.

Ihr TechTeam

TechTipp: HASH ON - forest of trees index

Indices werden meist angelegt, um sequentielle Zugriffe zu vermeiden, und somit die Performance bei Abfragen zu verbessern. Bei Tabellen mit vielen Einträgen können Indices selbst sehr groß werden, so dass diese mehrere Megabyte umfassen. Zudem besitzen diese Indices mehrere Ebenen (Level), die durchlaufen werden müssen, bevor die Information auf Blatt-Ebenen ausgewertet werden kann.

Ab Version 11.70 können große Indices durch einen „forest of trees index“ aufgeteilt werden. Dies erfolgt über die Angabe „hash on“, nach der die Aufteilung erfolgt.

Beispiel:

```
CREATE INDEX i1_ifx ON ifx(name,vorname);
```

Die Ausgabe von „oncheck -pT“ zeigt den Index mit drei Levels an:

Index Usage Report for index i1_ifx on test:informix.ifx

Level	Total	Average No. Keys	Average Free Bytes
1	1	6	1592
2	6	65	510
3	391	332	279
Total	398	327	286

Wird der Index mit einer Hash-Verteilung angelegt, so entstehen mehrere Teilindices mit jeweils einem Root-Knoten und nur noch einem zweiten Level:

```
CREATE INDEX i1_ifx ON ifx(name,vorname) HASH ON (name) WITH 20 BUCKETS;
```

Die Ausgabe von „oncheck -pT“ zeigt den neu angelegten forest of index trees:

Index Usage Report for index i1_ifx on test:informix.ifx

Level	Total	Average No. Keys	Average Free Bytes
1	20	21	1548
2	437	297	457
Total	457	285	505

Enthält das erste Feld eines Indexes zu viele Duplikate, so können im „hash on“ weitere Felder genutzt werden:

```
CREATE INDEX i1_ifx ON ifx(name,vorname) HASH ON (name,vorname) WITH 10 BUCKETS;
```

Die Felder, die im „hash on“ verwendet werden, müssen am Anfang des Indexes stehen, ansonsten wird eine Fehlermeldung ausgegeben:

```
CREATE INDEX i1_ifx on ifx(name,vorname) HASH ON (vorname) WITH 10 BUCKETS;
26361: Invalid 'hash on' list specified for FOT index.
```

TechTipp: onstat -g ses 0 – Snapshot für alle aktiven Sessions

Mittels „onstat -g ses <sid>“ kann für eine Session die detaillierte Information über der Ressourcenverbrauch, sowie deren aktuelles Statement und Isolation Level ausgegeben werden. Um solch einen Snapshot für alle aktuellen Sessions zu erstellen, kann als Session-ID die 0 angegeben werden.

Beispiel:

```
onstat -g ses 0
```

```
IBM Informix Dynamic Server Version 11.70.UC4 -- On-Line -- Up 06:58:27 ...
```

session id	user	tty	pid	hostname	#RSAM threads	total memory	used memory	dynamic explain
277	kalu	3	19100	kalu2	1	90112	81192	off

tid	name	rstcb	flags	curstk	status
314	sqlxec	4eb53004	Y--P---	5216	cond wait netnorm -

Memory pools count 2

name	class	addr	totalsize	freesize	#allocfrag	#freefrag
277	V	54d3b028	86016	6504	136	9
277*00	V	5490b028	4096	2416	1	1

name	free	used	name	free	used
overhead	0	3360	scb	0	96
opentable	0	2392	filetable	0	456
log	0	16512	temprec	0	14432
keys	0	360	ralloc	0	12024
gentcb	0	1240	ostcb	0	2608
sqscb	0	14832	sql	0	40
hashfiletab	0	280	osenv	0	1920
buft_buffer	0	2144	sqtcb	0	6872
fragman	0	1544	sapi	0	40

sqscb info

scb	sqscb	optofc	pdqpriority	optcompind	directives
54374018	54ef7018	0	0	2	1

Sess Id	SQL Stmt type	Current Database	Iso Lvl	Lock Mode	SQL ERR	ISAM ERR	F.E. Vers	Explain
277	SELECT	ibm	CR	Not Wait	0	0	9.24	Off

Current statement name : slctcur

Current SQL statement (2) :
select * from firma

Last parsed SQL statement :
select * from firma

1024 byte(s) of memory is allocated from the sscpool

```
-----
```

session id	user	tty	pid	hostname	#RSAM threads	total memory	used memory	explain
358	carmen	8	19940	kalu2	1	73728	67032	on

...

TechTipp: Informix Warehouse Accelerator: CPU-Nutzung konfigurieren

Dieser TechTipp gilt für Informix Warehouse Accelerator ab der Version 11.70.FC4.

Mit der Defaultkonfiguration des Informix Warehouse Accelerator kann es sein, dass die im System vorhandenen Prozessorressourcen bei der Ausführung von Abfragen nicht vollständig ausgenutzt werden. Eine Möglichkeit, die Ausnutzung der Prozessorleistung zu erhöhen, ist das Konfigurieren vom mehr Workerknoten. Dies wurde im sog. „IWA-Blog“ in einem frühen Eintrag mit dem Titel „[Configuration Tips \(1\): CPU Resources](#)“ schon ausführlich beschrieben.

Auf einem Einzelsystem (im Gegensatz zu einem Clustersystem) hat die beschriebene Vorgehensweise jedoch den Nachteil, dass mehr Workerknoten einen erhöhten Speicherbedarf verursachen, bedingt durch das Duplizieren der Dimensionstabellen für jeden Workerknoten (wie in einem weiteren Blog-Eintrag mit dem Titel „[Estimating the Size of a Data Mart](#)“ schon beschrieben). In einem Clustersystem, wo jeder Workerknoten des Accelerators auf seinem eigenen Clusterknoten läuft, ist dieses Duplizieren der Dimensionstabellen unvermeidbar, denn aufgrund der Architektur eines Clusters kann der Speicher von den Clusterknoten nicht gemeinsam genutzt werden. Wenn aber der Accelerator auf einem einzelnen (SMP) System läuft, dann ist dies ziemlich ungünstig, denn alle Workerknoten haben ihren Teil vom Speicher ja auf demselben System. Damit kann der Speicher nicht nur von den Threads eines einzelnen Workerknoten gemeinsam genutzt werden, sondern wäre auch für die verschiedenen Workerknoten untereinander gemeinsam nutzbar. Das Duplizieren der Dimensionstabellen im Speicher für jeden Workerknoten ist daher auf einem Einzelsystem nicht effizient.

Deshalb gibt es nun eine Möglichkeit, unter Vermeidung von mehreren Workerknoten und der damit verbundenen Duplizierung der Dimensionstabellen im Speicher, dennoch mehr Parallelität und damit eine bessere Prozessorauslastung auch auf einem Einzelsystem zu erreichen. Jeder Workerknoten ist in sich schon parallelisiert durch die Nutzung von Threads zur parallelen Ausführung jeder einzelnen Abfrage. Mit mehr parallelen Threads für die Abfrageausführung kann auch ein einzelner Workerknoten mehr Prozessorleistung ausnutzen und somit eine Abfrage schneller abarbeiten. Dasselbe gilt auch für das Laden von Daten in einen Data Mart, jedoch ist der Leistungsgewinn dafür geringer, da beim Laden die Aufgaben besser an verschiedene Workerknoten verteilt werden können, als ein einzelner Workerknoten sie auf seine Threads aufteilen kann.

Es gibt zwei Konfigurationsparameter für die Prozessorauslastung durch jeden einzelnen Workerknoten, sowohl für die Ausführung von Abfragen als auch das Laden von Data Marts:

- **CORES_FOR_SCAN_THREADS_PERCENTAGE**

Bestimmt den Prozentsatz vorhandener Prozessorleistung, der für die Ausführung von Abfragen benutzt werden soll. (Eine Hauptaufgabe bei der Ausführung von Abfragen ist das Lesen der Tabellendaten, daher das „SCAN“ im Parameternamen.)

- **CORES_FOR_LOAD_THREADS_PERCENTAGE**

Bestimmt den Prozentsatz vorhandener Prozessorleistung, der für das Laden von Data Marts benutzt werden soll. (Aufbau von Datenhistogramm und -wörterbuch, sowie Datenkompression).

Beide Parameter sind optional und können unabhängig voneinander in der Accelerator Konfigurationsdatei „dwainst.conf“ spezifiziert werden. Sind die Parameter nicht spezifiziert, so gilt für beide derselbe Defaultwert von 17 (Prozent). Sobald die Parameter und ihre Werte in die Konfigurationsdatei „dwainst.conf“ eingetragen sind, müssen sie an die einzelnen Worker- und Koordinatorknoten übertragen werden, um sie wirksam zu machen. Dafür muss der Accelerator gestoppt werden (mit dem Kommando „ondwa stop“), die neue Konfiguration muss verteilt werden (mit dem Kommando „ondwa setup“) und schliesslich muss der Accelerator wieder gestartet werden (mit dem Kommando „ondwa start“).

Der Accelerator benutzt die Werte dieser Parameter, um die Anzahl der parallelen Threads für die Ausführung der jeweiligen Aufgabe zu bestimmen. Dies geschieht anhand einer internen Berechnung basierend auf empirischen Werten und dem gegebenen Prozentsatz der vorhandenen Prozessorkerne. Vorhandene Prozessorkerne bezieht sich hierbei auf die Anzahl der Prozessorkerne, die dem Linux Betriebssystem zur Verfügung stehen. Dies kann durchaus verschieden sein von zwei anderen Begriffen, der Anzahl installierter Prozessorkerne und der Anzahl verfügbarer Prozessorkerne. Besonders in einer virtualisierten Umgebung ist die Anzahl der vorhandenen Prozessorkerne oft nicht gleich der Anzahl physikalisch installierter Prozessorkerne, da die virtuelle Maschine evtl. nur einen Teil der physikalisch existierenden Hardware nutzen darf. Andererseits kann die Anzahl vorhandener Prozessorkerne verschieden sein von der Anzahl verfügbarer Prozessorkerne, wenn einige Prozessorkerne dauerhaft von anderen Prozessen besetzt sind (wie z.B. von einem Informix Server mit konfigurierter Prozessoraffinität).

Beim Festlegen der Konfigurationsparameter sollte beachtet werden, dass im Allgemeinen die Ausführungsgeschwindigkeit von Abfragen recht gut mit der Anzahl paralleler Threads skaliert. Auf einem großen System (z.B. 64 oder mehr Prozessorkerne) ist es jedoch durchaus möglich, den Wert zu hoch zu konfigurieren, wodurch zu viele Threads einen merklichen Overhead erzeugen, und daraus resultierend die Leistung nicht verbessert wird. Auf einem kleineren System, z.B. mit 8 oder 16 Prozessorkernen, können beide Parameter meist ohne Probleme mit 80 oder 90 Prozent konfiguriert werden.

Das bisher Beschriebene bezieht sich auf einen Accelerator installiert auf einem einzelnen (SMP) System. Bei einer Clusterinstallation läuft normalerweise jeder individuelle Koordinator- oder Workerknoten des Accelerators exklusiv auf seinem eigenen Clusterknoten. Für solche Clusterinstallationen wird daher empfohlen, beide Parameter auf 100 Prozent zu setzen.

TechTipp: Extent Size / Next Size beim Erstellen von Indices

Die Größe der Extents bei Indices war bisher durch die Extentsize und Nextsize der Tabelle bestimmt. Über das Verhältnis Rowsize zu Indexsize wurde intern die Größe für die Extents der Indices festgelegt.

Bei fragmentierten/partitionierten Tabellen, deren Indices eine abweichende Fragmentierungsregel nutzen, kann es sehr sinnvoll sein, die Extentsize und Nextsize von Indices individuell anzugeben. Ebenso kann es Sinn machen, bei bestehenden Tabellen einen neuen Index mit einer ausreichenden Größe für die Extentsize und Nextsize anzulegen.

Die Syntax entspricht derjenigen beim Anlegen einer Tabelle.

Beispiel:

```
create index i1_ifx on ifx(name,vorname) extent size 1000 next size 200;
```

Bzw. bei einem Forest of index trees:

```
create index i1_ifx on ifx(name,vorname) hash on (name) with 10 buckets extent size 1000 next size 200;
```

Ein Index, dessen Ablage aus vielen Extents besteht, kann unnötigerweise die Performance negativ beeinflussen.

Ein Ändern der Extent- bzw. Nextsize von Indices ist derzeit im Gegensatz zur Extent- und Nextsize von Tabellen nicht möglich. Der Index müsste hierfür neu angelegt werden.

TechTipp: Statement-Typ im onstat -k

In der Ausgabe vom „onstat -k“ ist ab Version 11.70 zusätzlich zu den bisherigen Informationen der Typ des Statements zu sehen, das die Sperre verursacht. Der Typ ist mit einem Großbuchstaben hinter dem Key angegeben.

I steht für Insert.

U steht für Update.

Beispiel:

```
IBM Informix Dynamic Server Version 11.70.UC4 -- On-Line -- Up 03:55:05 ...
```

Locks

address	wtlist	owner	lklist	type	tblsnum	rowid	key#/bsiz	
442ef2a4	0	4eb506c8	442f0564	HDR+SR	100284	108	K- 1	
442ef724	0	4eb506c8	442ef8a4	HDR+X	1002a7	301	K- 1	I
442f01a4	0	4eb506c8	442f0624	HDR+X	100284	301	K- 1	I
442f0504	0	4eb506c8	442f04a4	HDR+IS	100259	0	0	
442f0564	0	4eb506c8	442f0444	HDR+SR	100284	107	K- 1	
442f05c4	0	4eb506c8	442f0504	HDR+X	100283	104	0	U

TechTipp: INFORMIX Open Source Projekte

Immer wieder erreicht uns die Frage, ob INFORMIX in OpenSource Projekten eingesetzt werden kann. Die IIUG hat hierzu eine Liste erstellt, die zeigt, welche Projekte bereits mit INFORMIX eingesetzt werden können, und welche Projekte in der Planung sind. Zum Teil findet man hier auch gleich die Links für einen Download der benötigten Treiber.

Aktive Projekte (in alphabetischer Reihenfolge)

=====		
Alfresco	wiki.alfresco.com/wiki/Main_Page	
Bugzilla	www.bugzilla.org	
Drupal	drupal.org	Download V6.16 und V7.7
Geronimo	geronimo.apache.org	JDBC-Support
Hibernate	www.hibernate.org	Download
iBATIS	ibatis.apache.org	JDBC-Support
Mediawiki	www.mediawiki.org/wiki/MediaWiki	Download V1.15.1 bis V1.15.3
OpenJPA	openjpa.apache.org	
PDO_IBM	pecl.php.net/package/PDO_IBM	Download
Tomcat	tomcat.apache.org	JDBC-Support
XWiki	www.xwiki.org/xwiki/bin/view/Main/WebHome	Download 2.2.5
Zabbix	www.zabbix.com	Download

Project Candidates (in alphabetischer Reihenfolge)

=====		
Atlassian	www.atlassian.com/software/	Commercial open-source Jira and Confluence
Jboss	www.jboss.com	Commercial open-source. Application server, portal, etc.
Joomla	www.joomla.org	Content management system
Mono	www.mono-project.com	Driver already available, may require attention
OpenBravo	www.openbravo.com	

Die Liste, so wie die Links für den Download finden Sie unter:

<http://www.iiug.org/opensource/>

Sollte der von Ihnen gesuchte Treiber nicht in der Liste stehen, dann wenden Sie sich bitte an opensource@iiug.org um das Projekt in die Wunschliste aufnehmen zu lassen.

TechTipp: OFF_RECOVERY_THREADS und HDR/RSS

Der Parameter OFF_RECOVERY_THREADS wird meist nur im Zusammenhang mit dem Restore einer Datenbank betrachtet.

Bei Nutzung der Replikation vom Typ HDR oder RSS kommen die Offline Recovery Threads jedoch ebenfalls zum Einsatz. Um eine optimale Performance zu erzielen, sollte der Parameter für die OFF_RECOVERY_THREADS auf eine Primzahl gesetzt werden, die in der Größenordnung von 3x CPU-VPs des Primary Servers liegt.

Der Default-Wert ist 10 und sollte nicht unterschritten werden (*).

Da ein schwacher Secondary Server auch die Performance auf dem Primary Server beeinflussen kann, gilt zudem die Empfehlung, den Secondary Server im Bereich CPUs und Hauptspeicher nicht schwächer als den Primary Server auszustatten.

(*) Anmerkung der Redaktion: Da der Default-Wert 10 keine Primzahl ist, haben wir einen Changerequest veranlasst, um den Default zu ändern.

TechTipp: ON_RECOVERY_THREADS

Der Parameter ON_RECOVERY_THREADS beeinflusst die Performance bei einem „warm restore“. Dies ist der Fall, wenn ein DBSpace zurück gesichert wird, während in der Instanz weiter gearbeitet wird.

Der Defaultwert dieses Parameters ist 1.

Um den „warm restore“ zu beschleunigen, kann der Parameter auf die Anzahl der Tabellen bzw. Fragmente gesetzt werden, die wiederhergestellt werden müssen. Die Erfahrung zeigt, dass Werte zwischen 10 und 20 die Zeiten des Restores deutlich verbessern. Zu hohe Werte belasten das System ggf. durch den notwendigen Overhead.

Auch beim Rollback einer Transaktion werden diese Recovery Threads aktiv.

Da hierbei jedoch meist nur wenige Tabellen betroffen sind, reichen niedrigere Werte aus.

WebTipp: Stress Tests auf Basis eines TCP-C Benchmarks unter Linux

INFORMIX Version 11.70.FC4 wurde auf Basis der Vorgaben eines TCP-C Benchmarks auf Linux einem Stresstest unterzogen. Es wurde nicht der offizielle Sourcecode eines TCP-C Benchmarks verwendet, die Applikation entsprach jedoch den offiziellen Vorgaben und Anforderungen des TPC Council.

Der Testrechner war ausgestattet mit einem Quadcore Prozessor und 16 GB Hauptspeicher. Ziel war es zu zeigen, dass die Datenbank Instanz im Rahmen der vorhandenen Ressourcen des Rechners sehr gut skaliert.

Die überaus erfreulichen Ergebnisse und weitere Details zum Test können Sie unter folgender Adresse nachlesen:

http://en.vercelletto.com/wp-content/plugins/downloads-manager/upload/Benchmark%20Informix%20innovator_EN.pdf

Anmeldung / Abmeldung / Anmerkung

Der Newsletter wird ausschließlich an angemeldete Adressen verschickt. Die Anmeldung erfolgt, indem Sie eine Email mit dem Betreff „**ANMELDUNG**“ an ifmxnews@de.ibm.com senden.

Im Falle einer Abmeldung senden Sie „**ABMELDUNG**“ an diese Adresse.

Das Archiv der bisherigen Ausgaben finden Sie zum Beispiel unter:

<http://www.iiug.org/intl/deu>

http://www.iug.de/index.php?option=com_content&task=view&id=95&Itemid=149

<http://www.informix-zone.com/informix-german-newsletter>

<http://www.drap.de/link/informix>

<http://www.nsi.de/informix/newsletter>

http://www.bytec.de/de/software/ibm_software/newsletter/

<http://www.cursor-distribution.de/index.php/aktuelles/informix-newsletter>

<http://www.listec.de/Newsletter/IBM-Informix-Newsletter/View-category.html>

<http://www.bereos.eu/software/informix/newsletter/>

Die hier veröffentlichten Tipps&Tricks erheben keinen Anspruch auf Vollständigkeit. Da uns weder Tippfehler noch Irrtümer fremd sind, bitten wir hier um Nachsicht falls sich bei der Recherche einmal etwas eingeschlichen hat, was nicht wie beschrieben funktioniert.

Die Autoren dieser Ausgabe

Gerd Kaluzinski IT-Specialist Informix Dynamic Server und DB2 UDB
 IBM Software Group, Information Management
gerd.kaluzinski@de.ibm.com +49-175-228-1983

Martin Fuerderer IBM Informix Entwicklung, München
 IBM Software Group, Information Management
martinfu@de.ibm.com

Sowie unterstützende Teams im Hintergrund.

Fotonachweis: Gerd Kaluzinski (Maibaum Lindau-Oberreitnau)