

Willkommen zum Informix Newsletter

Inhaltsverzeichnis

Aktuelles.....	1
TechTipp: AGGREGATES (built-in).....	2
TechTipp: CREATE AGGREGATE.....	3
TechTipp: Aggregate mit verändertem Delimiter.....	5
TechTipp: Optionen des ONSTAT (onstat -g cac agg).....	5
TechTipp: SMI - Sysmaster Interface - sysaggregates.....	6
TechTipp: GSKIT.....	6
TechTipp: Encrypted Unload.....	7
TechTipp: onkstore.....	8
TechTipp: ONCONFIG - BAR_ENCRYPTION.....	8
TechTipp: ONCONFIG - BAR_DECRYPTION.....	9
TechTipp: ONCONFIG - BAR_IXBAR_PATH.....	9
TechTipp: ONCONFIG - BAR_PERFORMANCE.....	10
TechTipp: ONCONFIG - BAR_PROGRESS_FREQ.....	10
TechTipp: ONCONFIG - BAR_MAX_BACKUP.....	11
TechTipp: ONCONFIG - BAR_MAX_RESTORE.....	11
TechTipp: ONCONFIG - BAR_SIZE_FACTOR.....	11
Hinweis: Neuer Service – INFORMIX HEALTH CHECK.....	12
Hinweis: Gastbeiträge erwünscht.....	13
Newsletter Archive.....	14
Nutzung des INFORMIX Newsletters.....	14
Die Autoren dieser Ausgabe.....	14

Aktuelles

Liebe Leserinnen und Leser,

der Frühling hat Einzug gehalten und die Natur steht in voller Blüte.

Zu Informix gibt es zu jeder Jahreszeit interessante News, und so widmen wir uns diesmal den Schwerpunkten Verschlüsselung und Aggregate.

Haben Sie Themen, die Sie rund um Informix interessieren, und zu denen Sie gerne Hintergründe hier im Newsletter erfahren würden ? Dann schreiben Sie uns Ihre Themenvorschläge. Sicher sind viele dieser Themen auch für andere Leser von Interesse.

Bleiben Sie gesund !

Ihr TechTeam



TechTipp: AGGREGATES (built-in)

INFORMIX bietet eine Reihe an „built-in“ Aggregatfunktionen.

Da diese Funktionen als „Overloaded Functions“ erstellt wurden, können sie mit einer Vielzahl von Datentypen unter dem selben Funktionsnamen genutzt werden.

Die Built-In Funktionen sind:

- **COUNT(*)**
Das Zählen von Datensätzen (die den Bedingungen der WHERE-Clause entsprechen).
- **COUNT(<col>)**
Das Zählen von Datensätzen, deren Spalte <col> nicht NULL ist.
- **COUNT(DISTINCT <col>)**
Das Zählen von unterschiedlichen Datensätzen die in der Spalte <col> vorkommen.
- **AVG(<col>)**
Der Durchschnitt der Werte über die angegebene Spalte (ohne Berücksichtigung von NULL Werten). Dies ist nur auf numerischen Spalten erlaubt.
- **MAX(<col>)**
Der höchste Wert einer Spalte (ohne Berücksichtigung von NULL Werten).
- **MIN(<col>)**
Der kleinste Wert einer Spalte (ohne Berücksichtigung von NULL Werten).
- **SUM(<col>)**
Die Summe der Werte einer Spalte (ohne Berücksichtigung von NULL Werten). Dies ist nur auf numerischen Spalten erlaubt .
- **RANGE(<col>)**
Die Range der Werte einer Spalte (berechnet aus MAX(<col>) - MIN(<col>)).
- **STDEV(<col>)**
Die Standardabweichung der Werte einer Spalte. Dies ist nur auf numerischen Spalten erlaubt.
- **VARIANCE(<col>)**
Die Varianz der Werte einer Spalte. Dies ist nur auf numerischen Spalten erlaubt.

Einschränkungen:

- Aggregat Funktionen können meist nicht in der WHERE-Clause einer SQL Abfrage genutzt werden.
- Verschachtelte Aggregat Funktionen sind nicht erlaubt wie z.B. MAX(AVG(<col>))
Diese Einschränkung kann mittels temporärer Tabellen meist einfach umgangen werden.
- Aggregat Funktionen können nicht auf den folgenden Datentypen angewandt werden:
 - Large object (BLOB, BYTE, CLOB, TEXT)
 - Collection data types (LIST, MULTISSET, SET)
 - ROW data types (named oder unnamed)
 - OPAQUE data types (ausser es wurde eine UDR Funktion hierfür erstellt).

TechTipp: CREATE AGGREGATE

Mit der Möglichkeit Aggregate selbst zu erstellen, bietet sich die Option massgeschneiderte Funktionalitäten in der Datenbank bereitzustellen.

Die Syntax beim Erstellen eines Aggregates setzt sich zusammen aus:

```
CREATE AGGREGATE <agg_name>
  WITH (
    INIT = ... ,           Definiert den initialen Wert und Typ der Variable.
    ITER = ... ,          Fügt einen Wert dem Teilergebnis hinzu.
    COMBINE = ... ,       Fügt mehrere Teilergebnisse zusammen.
                          Kann identisch der Funktion ITER sein, falls die
                          Datentypen der Eingaben und Zwischenergebnisse
                          übereinstimmen.
                          COMBINE kann vom Optimizer parallel ausgeführt
                          werden.
    FINAL = ...           Rückgabe des Ergebnisses an die aufrufende Funktion.
  )
```

Für jeden dieser Bereiche muss die entsprechende Funktion erstellt werden.

Das folgende Beispiel zeigt die Erstellung der Funktion LISTAGG() als User Defined Aggregat (* siehe Anmerkung).

```
CREATE AGGREGATE listagg
  WITH (
    INIT = listagg_init,
    ITER = listagg_iter,
    COMBINE = listagg_combine,
    FINAL = listagg_final
  );
```

(* Anmerkung der Redaktion:

Dieser Artikel basiert auf der Anfrage eines Kunden im Rahmen unseres Consultings:
„Gibt es bei INFORMIX eine Funktion ähnlich LISTAGG() bei anderen Datenbanken, die alle Werte einer Spalte als einen Wert zurück gibt ?“

Die Funktion wurde in einer Applikation zur Anzeige auf einer Webpage benötigt.

Bevor wir jedoch das Aggregat wie oben erwähnt erstellen können, müssen wir die vier Funktionen bereitstellen.

Die Funktion für die Initialisierung INIT ist nichts anders als die Definition des Eingabe- und Rückgabewertes, sowie die Initialisierung der Variablen dafür.

```
CREATE FUNCTION listagg_init(in_val VARCHAR(255))
  RETURNING LVARCHAR;
  RETURN '' ;
END FUNCTION;
```

Wir geben also einen leeren String vom Typ LVARCHAR zurück.

Ebenso ist die Funktion FINAL mit sehr wenig Logik ausgestattet. Sie dient ausschliesslich der Rückgabe des ermittelten Ergebnisses:

```
CREATE FUNCTION listagg_final(in_val LVARCHAR)
  RETURNING LVARCHAR;
  RETURN TRIM(in_val);
END FUNCTION;
```

Nun haben wir die Initialisierung und die Rückgabe erstellt. Was noch fehlt ist die Iteration je Wert:

```
CREATE FUNCTION listagg_iter(in_val1 LVARCHAR, in_val2
  VARCHAR(255))
  RETURNING LVARCHAR;
  IF in_val1 = '' THEN
    RETURN TRIM(in_val2);
  ELSE
    RETURN TRIM(in_val1)||','||TRIM(in_val2);
  END IF;
END FUNCTION;
```

Diese Funktion gibt bei leerem Zwischenergebnis den neu gelesenen Wert zurück, ansonsten wird der neue Wert mittel CONCAT an das bisherige Teilergebnis angehängen, wobei zudem ein Trennzeichen (hier Komma) eingefügt wird.

Zur besseren Performance können Aggregate parallel ausgeführt werden, wenn der Optimizer dies für sinnvoll erachtet. Hierfür dient die Sektion COMBINE. Dabei werden Teilergebnisse zusammengeführt zu einem Gesamtergebnis. Falls Einzelwerte und Teilergebnis den selben Datentyp besitzen, kann COMBINE und ITER die selbe Funktion beinhalten.

```
CREATE FUNCTION listagg_combine(in_val1 LVARCHAR, in_val2 LVARCHAR)
  RETURNING LVARCHAR;
  IF in_val1 IS NULL OR in_val1 = '' THEN
    RETURN TRIM(in_val2);
  ELIF in_val2 IS NULL OR in_val2 = '' THEN
    RETURN TRIM(in_val1);
  ELSE
    RETURN TRIM(in_val1)||','||TRIM(in_val2);
  END IF;
END FUNCTION;
```

Nachdem wir nun alle Funktionen erstellt haben und das Aggregat definiert ist, können wir die Funktionalität in unseren SQL Statements nutzen:

```
select listagg(tabname)
from systables
where tabid > 99
;

listagg
customer,orders,manufact,stock,items,state,call_type,cust_calls,cat
alog,tab,warehouses,classes,employee,ext_customer
```

TechTipp: Aggregate mit verändertem Delimiter

Wer die Funktion LISTAGG von anderen Datenbankherstellern kennt, der kennt die Syntax dass der Delimiter mitgegeben werden kann. Da unser Aggregat nur ein Argument verarbeitet (die Spalte, über die aggregiert wird), haben wir hier nicht die Option, diesen Delimiter direkt zu beeinflussen.

Mit der Funktion REPLACE lässt sie diese Anforderung allerdings sehr einfach bewerkstelligen, indem man den Aufruf von LISTAGG() in ein REPLACE einbindet:

```
select listagg(tabid) as csv_list,
       replace(listagg(tabid), ',', '-', 1) as dash_list
from systables
where tabid > 99
;

csv_list    100,101,102,103,104,105,106,107,108,109,110,111,112,113
dash_list  100-101-102-103-104-105-106-107-108-109-110-111-112-113
```

TechTipp: Optionen des ONSTAT (onstat -g cac agg)

Aggregate werden nach dem ersten Aufruf im Cache gehalten, daher werden diese im „onstat -g cac agg“ aufgelistet:

User Define Aggregate Cache:

```
Number of lists      : 31
DS_POOLSIZe         : 139
T1 count             : 1
T2 count             : 0
B1 count             : 0
B2 count             : 0
T1 target            : 0
B1 target            : 0
Short count          : 1
Long count           : 0
```

User Define Aggregate Cache Entries:

address	list id	ref	drop	type	hits	last_access
heap_ptr	item					
4601f4f8	17	13	0	0	T1	42
4601f4d8	stores@kalu:kalu.listagg (13)					2021-03-16 14:15:16
Total number of entries						: 1
Number of entries in use						: 0

Interessant an dieser Ausgabe ist der Wert „hits“, der anzeigt wie oft dieses Aggregat seit dem Rücksetzen der Statistiken verwendet wurde.

Eine komplette Übersicht über im Informix Server vorhandenen Caches wurde im INFORMIX Newsletter Mai 2013 aufgelistet.

TechTipp: SMI - Sysmaster Interface - sysaggregates

Die Tabelle sysaggregates in der Datenbank sysmaster beinhaltet die Information über alle vorhandenen User Defined Aggregates der Datenbank.

Hier findet sich auch unser Aggregat „listagg“ wieder:

```
name          listagg
owner         kalu
aggid         13
init_func     listagg_init
iter_func     listagg_iter
combine_func  listagg_combine
final_func    listagg_final
handlesnulls  f
```

Zudem sind hier Aggregate zu XML und anderen Tools zu finden.

TechTipp: GSKIT

Bevor wir mit unserer Informix Instanz irgendeine Variante der Verschlüsselung (SSL, DISK_ENCRYPTION, encrypt(), BAR_ENCRYPTION, ...) nutzen können, muss das mit der Installation mitgelieferte GSKIT installiert werden. Ja nach Version erfolgt die Installation automatisch mit der Installation des Servers und kann abgewählt werden.

Um zu überprüfen ob eine Version des GSKit installiert ist dient der folgende Aufruf:

```
gsk8capicmd_64 -version
```

In der aktuellen Installation (14.10.FC5) liefert dieser Version 8.0.55.17 zurück.

Wird versucht die Verschlüsselung ohne diese Installation aufzurufen, so wird im online.log folgender Fehler ausgegeben:

```
IBM Global Security Kit (GSKit) version 8.0.50.88 or higher,
which provides APIs for Secure Sockets Layer (SSL)
communication and for encryption, is not installed.
ICC library load failed
```

Das GSKIT ist zu finden im Verzeichnis \$INFORMIXDIR/gskit. Die Installation erfolgt als Benutzer „root“ mittels:

```
./installgskit
Performing GSKit installation for Linux ...
```

Mit dem Aufruf von „installgskit“ werden die für das jeweilige Betriebssystem notwendigen Aufrufe zur Installation und Konfiguration ausgeführt, die für die Nutzung einer Verschlüsselung notwendig sind. Das Vorhandensein der notwendigen Pakete des Betriebssystems wird bereits bei der Installation des Informix Servers abgeprüft (daher muss zur Installation z.B. auf Linux die „libcrypto“ in einer unterstützten Version vorhanden sein).

TechTipp: Encrypted Unload

Die Frage „Lassen sich Unload Daten verschlüsseln?“ hat uns zu diesem Artikel inspiriert. Die Antwort ist einfach: „JA, mit einem kleinen Trick“.

Das Entladen der Daten mittels „encrypt“ lässt sich über folgenden Befehl erreichen:

```
set encryption password "Kalu knows the solution";

unload to /tmp/kalu_test
select customer_num, lname, encrypt_aes(fname)
from customer;
```

In der Unload Datei sind die Vornamen nun nicht mehr im Klartext zu finden:

```
101 | Pauli | 0CvX/AAAAEAavcxIyZcJjsmQLIVvE+r4Y9ZL4S0dHC/ |
102 | Sadler | 0wj3/AAAAEALeFCYLF4kByXNGvWYLBQfqbRmFxY9+3T |
103 | Currie | 0vUL/AAAAEA+bZc/GQCyjsdF40lzk7dDYv0oPRfZRp |
104 | Higgins | 02iX/AAAAEA1MayLOWGrbSrQF25Xblqnpncyk0XRozd |
105 | Vector | 0g3z/AAAAEAF+U4dGXs3+yRH8NW7I7I6DSy3HmOU6aC |
```

Das Laden der Daten mit Entschlüsselung geht leider nicht direkt, da „LOAD“ keinen Decrypt in der Syntax erlaubt.

Der folgende Workaround mit einer temporären Tabelle hilft hier weiter.

```
create temp table customer_enc(
customer_num int,
lname varchar(128),
fname_enc varchar(128)
) with no log;

create table customer2(
customer_num int,
lname varchar(128),
fname varchar(128)
);

load from /tmp/kalu_test
insert into customer_enc;

insert into customer2
select customer_num, lname, decrypt_char(fname_enc, "Kalu knows the
solution") from customer_enc;
```

Achtung !

INFORMIX beachtet nicht, dass beim Encrypt der gesamte Key, der basierend auf das Password erstellt wird, in die entsprechende Spalte passen muss. Es wird keine Warnung ausgegeben falls dies nicht der Fall ist, sondern die Zeichen, die nicht in die entsprechende Spalte passen werden verworfen.

Es muss daher unbedingt für ausreichende Dimensionierung gesorgt werden beim Aufbau der temporären Tabelle, da ansonsten ein Teil vom Key angeschnitten wird.

Damit wären die Daten zwar sicher, aber leider nicht zu entschlüsseln.

TechTipp: onkstore

Im Informix Newsletter Q1/2017 hatten wir die Option vorgestellt, die Datenablage in den DBSpaces bzw. Chunks zu verschlüsseln. Dazu musste ein Keystore mittels der Funktion `gsk8capicmd_64` angelegt werden.

Hierbei handelte es sich um einen lokalen Keystore.

In aktuellen Versionen ist hierfür der Befehl „onkstore“ im Verzeichnis `$INFORMIXDIR/bin` vorhanden, mit dessen Hilfe der Keystore erstellt werden kann.

Der Parameter „type“ gibt an, welche Art von Keystore erstellt werden soll:

<code>local</code>	Lokaler Keystore wie bisher mit dem Kommando aus dem GSKIT.
<code>AWS-EAR</code>	für Storage Spaces im Amazon Web Services Key Management Service (AWS-KMS).
<code>AWS-BAR</code>	für das Amazon Web Service Integrated Backup.
<code>AZURE-EAR</code>	für Microsoft Azure Key Vault Encryption der Storage Spaces.
<code>AZURE-BAR</code>	für das Microsoft Azure Integrated Backup.
<code>KMIP</code>	für Server bzw. Cluster die einen KMIP Service nutzen. Dieser kann auch für die Verschlüsselung des Backups genutzt werden.

Für die folgenden Beispiele nutzen wir einen lokalen Keystore.

Diesen erstellen wir mittels:

```
onkstore -create -file keystore_ifx -type local -cipher aes256
```

Der Keystore wird automatisch im Verzeichnis `$INFORMIXDIR/etc` abgelegt.

TechTipp: ONCONFIG - BAR_ENCRYPTION

Mit Version 14.10 wurde die Möglichkeit geschaffen, dass binäre Sicherungen direkt beim Schreiben verschlüsselt werden. Bisher war dies nur über die Verwendung eines `BACKUP_FILTER` möglich, der als Pipe den Stream der Sicherung verschlüsselt hat.

Bei der Verwendung der `BAR_ENCRYPTION` bleibt die Funktionalität des `BACKUP_FILTER` erhalten, so dass zusätzlich eine Komprimierung oder ähnliche Aktionen möglich sind.

Die Verschlüsselung, die mittels `BAR_ENCRYPTION` eingerichtet wird, verschlüsselt nicht nur Sicherungen mittels „onbar“, sondern auch die Sicherungen mittels „ontape“.

Als Keystore wird empfohlen nicht den Keystore zu nutzen, der für die `DISK_ENCRYPTION` verwendet wird.

Als Wert für `BAR_ENCRYPTION` kann entweder ein Keystore, oder ein Keyfile genutzt werden.

Die Definition eines Keystores wurde im vorherigen Artikel beschrieben. Ein Keyfile wird erstellt mittels externer Tools die nicht mit dem Informix Server installiert werden, z.B.:

```
openssl rand -base64 24 > $INFORMIXDIR/etc/backup_key
```

In der \$ONCONFIG wird dann entweder der Keystore, oder das Keyfile eingetragen in der Form:

```
BAR_ENCRYPTION keystore=$INFORMIXDIR/etc/backup_keystore,cipher=aes192  
BAR_ENCRYPTION keyfile=$INFORMIXDIR/etc/backup_key,cipher=aes192
```

Der Parameter kann nicht dynamisch gesetzt werden mittels „onmode -wf“.

Eine Sicherung mit „ontape“ gibt zu Beginn die Information aus:

```
ontape -s -L 0  
The backup volume will be encrypted.
```

...

Eine Sicherung mit „onbar“ gibt keine Hinweise direkt zurück, hier ist die Verschlüsselung im Debug File zu sehen:

```
2021-04-22 11:36:35 45883 45881 encryptionSetupForBackup: enter  
2021-04-22 11:36:35 45883 45881 buildEncryptionEnvelope: enter  
2021-04-22 11:36:35 45883 45881 buildEncryptionEnvelope: return 0  
(0x00)  
2021-04-22 11:36:35 45883 45881 encryptionSetupForBackup: return 0  
(0x00)
```

TechTipp: ONCONFIG - BAR_DECRYPTION

Der Parameter BAR_DECRYPTION wird ausschliesslich benötigt, falls die Werte der Verschlüsselung mittels BAR_ENCRYPTION beim Restore verändert werden sollen. Andernfalls werden die identischen Werte genutzt. BAR_DECRYPTION erlaubt kein Angabe des cipher, da dies unverändert aus der Verschlüsselung übernommen werden muss.

TechTipp: ONCONFIG - BAR_IXBAR_PATH

Dieser Parameter ist nicht in der onconfig.std enthalten. Per Default wird die Datei „ixbar.<servernum>“, in der sich die Informationen über alle mit dem Befehl „onbar“ erstellten Sicherungen befinden, im \$INFORMIXDIR/etc“ erstellt.

Der Parameter „BAR_IXBAR_PATH“, der dynamisch mittels „onmode -wf“ änderbar ist, kann dazu genutzt werden, um diese Datei in ein anderes Verzeichnis zu schreiben.

Der Parameter ist jederzeit änderbar und wird beim nachfolgenden Aufruf von „onbar“ wirksam.

TechTipp: ONCONFIG - BAR_PERFORMANCE

Wer sich durch das Setzen von BAR_PERFORMANCE eine schnellere Sicherung erwartet, wird leider enttäuscht.

Der Parameter bietet die Option, im Logfile der Sicherung (BAR_ACT_LOG) zusätzliche Statistiken über die Sicherungsschritte auszugeben.

Als Parameter sind gültig:

- 0 Deaktiviert (Default)
- 1 Zeit für die Übertragung der Daten zwischen Instanz und StorageManager ausgeben
- 2 TimeStamps in Microsekunden anzeigen
- 3 TimeStamps und Übertragungszeit ausgeben

Beispiel:

```
2021-04-22 12:43:26.665059 47728 47726 PERFORMANCE INFORMATION
TRANSFER RATES:
-----
| OBJECT NAME | xfer-kbytes | XBSA API xfer-time | RATIO(kb/s) | API-TIME | xfer-kbytes | SERVER API xfer-time | RATIO(kb/s) | API-TIME |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 13          | 0           | 0.003     | 0           | 0.007     | 48          | 0.043     | 1112        | 0.044     |
| datadbs    | 0           | 0.001     | 0           | 0.001     | 62          | 0.000     | 805099     | 0.737     |
| rootdbs    | 0           | 0.023     | 0           | 0.026     | 17422       | 0.015     | 1161569    | 0.544     |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| PID =      | 47728      | 0         | 0           | 0.035     | 17532       | 0.058     | 301114     | 1.324     |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| TOTAL XFER | 0         | 0.026     | 0           | 0.035     | 17532       | 0.058     | 301114     | 1.324     |
-----

2021-04-22 12:43:26.665069 47728 47726 PERFORMANCE INFORMATION
PROCESS CLOCKS:
-----
| PID | CLOCK DESCRIPTION | TIME SPENT (s) |
|-----|-----|-----|
| 47728 | To execute the master OnBar daemon. | 1.833 |
| 47728 | To update 'sysutils' for Dbspace datadbs. | 0.036 |
| 47728 | To update 'sysutils' for Dbspace rootdbs. | 0.043 |
| 47728 | To update 'sysutils' for Log 13. | 0.033 |
-----

2021-04-22 12:43:26.665133 47728 47726 /opt/informix/bin/onbar_d complete, returning 0 (0x00)
2021-04-22 12:43:27.367630 47764 47762 /opt/informix/bin/onbar_d -b -l
2021-04-22 12:43:27.401587 47764 47762 Using IBM's PSM version 14.10.FC5 as the Storage Manager.
XBSA API version is 1.0.3.
2021-04-22 12:43:27.437663 47764 47762 PERFORMANCE INFORMATION
TRANSFER RATES:
-----
| OBJECT NAME | xfer-kbytes | XBSA API xfer-time | RATIO(kb/s) | API-TIME | xfer-kbytes | SERVER API xfer-time | RATIO(kb/s) | API-TIME |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| TOTAL XFER | 0         | 0.000     | -nan        | 0.000     | 0           | 0.000     | -nan        | 0.000     |
-----

2021-04-22 12:43:27.437693 47764 47762 PERFORMANCE INFORMATION
PROCESS CLOCKS:
-----
| PID | CLOCK DESCRIPTION | TIME SPENT (s) |
|-----|-----|-----|
| 47764 | To execute the master OnBar daemon. | 0.082 |
-----

2021-04-22 12:43:27.437947 47764 47762 /opt/informix/bin/onbar_d complete, returning 0 (0x00)
```

TechTipp: ONCONFIG - BAR_PROGRESS_FREQ

Der Prozess „onbar“ schreibt üblicherweise weder beim Backup noch beim Restore den geschätzten Fortschritt in % in die Datei BAR_ACT_LOG (Default 0).

Wird der Parameter auf einen ganzzahligen Wert grösser als 5 gesetzt, so wird alle x Minuten der Fortschritt in % zusätzlich in die Logdatei geschrieben.

Werte zwischen 0 und 5 werden als 5 (Minimum) interpretiert und genutzt.

TechTipp: ONCONFIG - BAR_MAX_BACKUP

Die Sicherungen mit Hilfe von „onbar“ können im Gegensatz zur Sicherung mittels „ontape“ parallel durchgeführt werden. Als Default steht dieser Parameter in onconfig.std auf 0, was eine dynamische Anpassung zwischen der Informix Instanz und dem StorageManager bewirkt. Obwohl der Wert 0 aufgelistet ist, wird intern ein Default Wert von 4 verwendet.

Um die Parallelisierung zu deaktivieren muss der Parameter explizit auf 1 gesetzt werden. Als mögliche Werte sind ganze Zahlen bis 2047 zugelassen. Hierbei sollten die Anzahl der DBSpaces und die Möglichkeiten des StorageManagers berücksichtigt werden.

TechTipp: ONCONFIG - BAR_MAX_RESTORE

Neu hinzu kam der Parameter BAR_MAX_RESTORE, der es zulässt, die Parallelität des Restore unabhängig von der Parallelität des Backups zu bestimmen. Als Werte sind wie beim BAR_MAX_BACKUP die Zahlen 0=dynamisch über 1=sequentiell bis 2047 erlaubt. Ist der Parameter nicht gesetzt, so wird der Wert von BAR_MAX_BACKUP übernommen.

TechTipp: ONCONFIG - BAR_SIZE_FACTOR

Der Vollständigkeit halber sei noch der Parameter BAR_SIZE_FACTOR erwähnt. Dieser ist im Default auf 0 und sollte nur gesetzt werden, falls während des Backups erhebliche Datenmengen neu gespeichert werden, und der StorageManager nicht mit veränderbaren Grössen des Backup umgehen kann.

Hintergrund:

Einige Storage Manager fragen zu Beginn der Sicherung die Grösse ab und stellen daraufhin das passende Medium zur Verfügung. Werden nun während der Sicherung erhebliche, zusätzliche Datenmengen erfasst, so steigt das Datenvolumen und ggf. bricht die Sicherung mit einem Fehler ab, der vom StorageManager veranlasst wird.

Mögliche Werte sind ganze Zahlen, die den Prozentsatz des Wachstums der Sicherung darstellt. So entspricht 10 der Erwartung dass die Sicherung vom Start bis zum Abschluss um 10% mehr Plattenplatz einnimmt.

Hinweis: Neuer Service – INFORMIX HEALTH CHECK

Da immer mehr Kunden nach einem Health Check für ihre INFORMIX Instanz nachgefragt haben, wurde ein neues Angebot erstellt, das den „Gesundheitszustand“ der INFORMIX Instanz weitgehend automatisiert analysiert.

Hierbei werden alle wichtigen Informationen gesammelt und Remote qualifiziert ausgewertet, so dass am Ende ein übersichtliches PDF mit den markanten Informationen zur Verfügung steht.

Geprüft werden dabei unter Anderem

- Die Cache Rates beim Lesen von Daten
- Der Füllungsgrad der DBSpaces
- Die Informationen über Backup und Restore
- Deadlocks, Performance Kennzahlen
- Fragmentierung der Tabellen
- Aktualität von Statistikdaten
- Warnungen, Fehlermeldungen, geänderte Parameter
- Limits für die Datenhaltung
- Replikation
- Entwicklung des Speicher- und Plattenverbrauchs seit Initialisierung
- sowie viele weitere Faktoren

Sollten Sie Bedarf an einem derartigen Check haben, so kontaktieren sie unsere Redaktion unter Mail: gerd.kaluzinski@de.ibm.com

INFORMIX Health Check Report for INFORMIX System: KALU

VERSION: IBM Informix Dynamic Server Version 14.10.FC5DE

Check Zeit: 2021-04-28 12:54:39

gerd.kaluzinski@de.ibm.com
Erstellt: 2021-04-28
check_version 1.01

CHECK SYSTEM STATUS

DBSERVER Information			
DBSERVER	kalu		
DBSERVER ALIASES			
TIME Information			
Start Time:	2021-04-28 11:28:16		
UP-Time:	0 02:19:02.872		
Reset Time	2021-04-28 11:28:16		
CACHE (Schwellenwerte: 90%, 95%)			
BUFFSIZE:2048	READ CACHE	98,8224	OK
BUFFSIZE:4096	READ CACHE	63,6363	CHECK: CACHE RATE TOO LOW
BUFFSIZE:2048	OVERFLOWS	0	OK
BUFFSIZE:4096	OVERFLOWS	0	OK
BUFFSIZE:2048	FG WRITES	0	OK
BUFFSIZE:4096	FG WRITES	0	OK
DBSPACES (Schwellenwerte: 30%, 15%)			
DBSPACE1 - rootdbs	FREE %	70,8780	OK
DBSPACE2 - logdbs	FREE %	0,9894	OK
DBSPACE3 - sdbdbs	FREE %	5,0099	OK
DBSPACE4 - tmpdbs	FREE %	99,9878	OK
DBSPACE5 - datadbs	FREE %	66,4023	OK
DBSPACE6 - sdbdbs2	FREE %	5,3800	OK
DBSPACE7 - data4k	FREE %	26,2000	WARN
DBSPACE BACKUP (Schwellenwerte: 1 Tag, 5 Tage)			
DBSPACE1 - rootdbs	LAST BACKUP	27.04.2021	WARNING
DBSPACE2 - logdbs	LAST BACKUP	27.04.2021	WARNING
DBSPACE3 - sdbdbs	LAST BACKUP	27.04.2021	WARNING
DBSPACE5 - datadbs	LAST BACKUP	27.04.2021	WARNING
DBSPACE6 - sdbdbs2	LAST BACKUP	27.04.2021	WARNING
DBSPACE7 - data4k	LAST BACKUP	27.04.2021	WARNING
DEADLOCKS		Anzahl Tabellen:	0
SEQUENTIAL SCANS (Schwellenwert: 200 bei Tabellen > 200.000 Rows und 100.000 Pages)			0
EXTENTS (Schwellenwert: 80 und 160)			0
STATISTICS (Schwellenwert: 10 Tage / 30 Tage)			
Database: db_diff	NO LOG - created: 24.02.2021	StatTime: 28.04.2021	OK
Database: health_check	NO LOG - created: 10.03.2021	StatTime: 28.04.2021	OK
Database: ibm	LOGGING - created: 27.04.2021	StatTime: 28.04.2021	OK
Database: ibm_old	NO LOG - created: 16.10.2018	StatTime: 28.04.2021	OK
Database: informix_hq	LOGGING - created: 21.07.2020	StatTime: 28.04.2021	OK
Database: kalu	LOGGING - created: 16.07.2020	StatTime: 28.04.2021	OK
Database: kalu_202007	LOGGING - created: 23.04.2019	StatTime: 28.04.2021	OK

... je nach Anzahl der Details zu aufgetretenen Meldungen umfasst der Report üblicherweise zwischen 3 und 8 Seiten.

Hinweis: Gastbeiträge erwünscht

Oft haben Sie wohl schon in einem unserer Newsletter Anregungen oder Tipps gefunden, die Sie im Alltag als DB Admin oder Entwickler gut gebrauchen konnten. Allerdings schlummern bei Ihnen am Rechner sicher noch interessante Scripts und Lösungen, die für viele andere Leser interessant sein können.

Daher unsere Bitte an Sie:

Senden Sie uns Ihre Scripts, Tipps und Tricks, damit wir diese hier vorstellen können. Sie haben die Wahl, ob Sie als Autor genannt werden wollen, oder der Beitrag anonym als Gastbeitrag eingestellt wird.

Entsprechend unserem Budget können wir als Prämie leider nur diesen Newsletter als PDF anbieten und viele weitere Tipps und Tricks in den kommenden Ausgaben.

Haben Sie eine Lösung mit INFORMIX erstellt und wollen diese hier beschreiben ? Auch dafür bieten wir Ihnen an, eine kurze Vorstellung der Lösung und Ihrer Kontaktinformation für die Leser, die sich näher dafür interessieren.

Newsletter Archive

Vielen Dank dass uns auch nach der langen Pause einige Archive wieder aufgenommen haben.

Hier eine Übersicht der Archive, von denen wir wissen:

- <https://iug.de/informix-newsletter>
- <https://www.cursor-distribution.de/de/ibm-software/informix-fuer-administratoren/informix-newsletter-ibm>
- <https://bereos.eu/informix-newsletter.html>
- <https://www.nsi.de/informix/informix-newsletter>

Sie haben auch den Newsletter in einem öffentlichen Archiv abgelegt ? Dann senden Sie uns den Link, damit wir diesen hier mit einpflegen können.

Nutzung des INFORMIX Newsletters

Die hier veröffentlichten Tipps&Tricks erheben keinen Anspruch auf Vollständigkeit.

Die IUG hat sich dankenswerterweise dazu bereit erklärt, den INFORMIX Newsletter auf ihren Web Seiten zu veröffentlichen.

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.

Rückmeldungen hierzu sind herzlich Willkommen !

Die gefundenen Tippfehler dürfen zudem behalten und nach Belieben weiterverwendet werden.

Eine Weiterverbreitung in eigenem Namen (mit Nennung der Quelle) oder eine Bereitstellung auf der eigenen HomePage ist ausdrücklich erlaubt. Alle hier veröffentlichten Scripts stehen uneingeschränkt zur weiteren Verwendung zur Verfügung.

Die Autoren dieser Ausgabe

Gerd Kaluzinski

DataOps Application Consultant

IBM Cloud and Cognitive Software

gerd.kaluzinski@de.ibm.com

+49-175-228-1983

Besonderer Dank geht an **Martin Fuerderer** der HCL, der zuverlässig die Korrektur und Überprüfung der Inhalte und der Tippfehler übernimmt, sowie an **Andreas Legner** (ebenfalls HCL), der bei der Recherche neuer Features zuverlässig unterstützt.

Dank auch an die vielen Helfer im Hintergrund.

In dieser Ausgabe finden sich nicht nur unzählige Kommata, die Martin bei der Korrektur gespendet hat, sondern auch eine Reihe an Punkten. DANKE ! Ich hatte mich an den Leitsatz gehalten, dass Satzzeichen keine Herden-Tiere sind. Martin verhindert nun auch ein Aussterben.

Nicht zu vergessen der Dank an die Informix User Group, ohne die es keinen Neuanfang des INFORMIX Newsletters gegeben hätte und die dankenswerter Weise die Verteilung übernimmt.

Foto Nachweis:

Apfelblüte - Redaktionsgarten

(Gerd Kaluzinski)