



Optimizer Statistiken und Adaptive Features in 12.2

DOAG Konferenz + Ausstellung 2017

Dierk Lenz



Herrmann & Lenz Services GmbH

Herrmann & Lenz Solutions GmbH



- Erfolgreich seit 1996 am Markt
- Firmensitz: Burscheid (bei Leverkusen)
- Beratung, Schulung und Betrieb/Fernwartung rund um das Thema Oracle Datenbanken
- Schwerpunktthemen: Hochverfügbarkeit, Tuning, Migrationen und Troubleshooting / Monitoring
- Herrmann & Lenz Solutions GmbH
 - Produkte: Monitoring Module (NEU:Version 2.0!), Taskzone
 - Stand auf Ebene 2 (236)

Optimizer Statistiken



Optimizer Statistiken – wie war das nochmal?

- Ohne geht es nicht!
- Nur noch `DBMS_STATS` erlaubt
(`ANALYZE TABLE` schon lange nicht mehr...)
- Automatische Berechnung – da gab es einen Scheduler Job, oder?



DBA_SCHEDULER_JOBS

OWNER	JOB_NAME
SYS	PURGE_LOG
SYS	FILE_WATCHER
SYS	PMO_DEFERRED_GIDX_MAINT_JOB
SYS	CLEANUP_NON_EXIST_OBJ
SYS	CLEANUP_ONLINE_IND_BUILD
SYS	CLEANUP_TAB_IOT_PMO
SYS	CLEANUP_TRANSIENT_TYPE
SYS	CLEANUP_TRANSIENT_PKG
SYS	CLEANUP_ONLINE_PMO
SYS	FILE_SIZE_UPD
SYS	ORA\$AUTOTASK_CLEAN
SYS	HM_CREATE_OFFLINE_DICTIONARY
SYS	DRA_REEVALUATE_OPEN_FAILURES
SYS	ORA\$PREPLUGIN_BACKUP_JOB
SYS	BSLN_MAINTAIN_STATS_JOB
SYS	FGR\$AUTOPURGE_JOB
SYS	RSE\$CLEAN_RECOVERABLE_SCRIPT
SYS	SM\$CLEAN_AUTO_SPLIT_MERGE
ORACLE_OCM	MGMT_CONFIG_JOB
ORACLE_OCM	MGMT_STATS_CONFIG_JOB
SYS	LOAD_OPATCH_INVENTORY
SYS	XMLDB_NFS_CLEANUP_JOB




DBA_SCHEDULER_JOB_RUN_DETAILS

LOG_ID	LOG_DATE	OWNER	JOB_NAME	JOB_SUBNAME	STATUS	ERROR#	REQ_START_DATE	ACTUAL_START_DATE
7734	16.11.17 01:41:38,038022000	+01:00 SYS	ORA\$AT_OS_OPT_SY_457	(null)	SUCCEEDED	0	(null)	16.11.17 01:41:28,541650000
7706	16.11.17 01:31:36,085355000	+01:00 SYS	ORA\$AT_OS_OPT_SY_455	(null)	SUCCEEDED	0	(null)	16.11.17 01:31:26,758435000
7682	16.11.17 01:21:34,667728000	+01:00 SYS	ORA\$AT_OS_OPT_SY_453	(null)	SUCCEEDED	0	(null)	16.11.17 01:21:25,071082000
7650	16.11.17 01:11:32,365458000	+01:00 SYS	ORA\$AT_OS_OPT_SY_451	(null)	SUCCEEDED	0	(null)	16.11.17 01:11:23,204191000
7626	16.11.17 01:01:31,126990000	+01:00 SYS	ORA\$AT_OS_OPT_SY_449	(null)	SUCCEEDED	0	(null)	16.11.17 01:01:21,474164000
7598	16.11.17 00:51:30,590185000	+01:00 SYS	ORA\$AT_OS_OPT_SY_447	(null)	SUCCEEDED	0	(null)	16.11.17 00:51:19,850374000
7574	16.11.17 00:41:27,492399000	+01:00 SYS	ORA\$AT_OS_OPT_SY_445	(null)	SUCCEEDED	0	(null)	16.11.17 00:41:18,071771000
7550	16.11.17 00:31:25,576033000	+01:00 SYS	ORA\$AT_OS_OPT_SY_443	(null)	SUCCEEDED	0	(null)	16.11.17 00:31:16,281703000
7526	16.11.17 00:21:24,976003000	+01:00 SYS	ORA\$AT_OS_OPT_SY_441	(null)	SUCCEEDED	0	(null)	16.11.17 00:21:14,764161000
7496	16.11.17 00:11:22,535829000	+01:00 SYS	ORA\$AT_OS_OPT_SY_439	(null)	SUCCEEDED	0	(null)	16.11.17 00:11:13,078785000
7472	16.11.17 00:01:22,120244000	+01:00 SYS	ORA\$AT_OS_OPT_SY_437	(null)	SUCCEEDED	0	(null)	16.11.17 00:01:11,332825000
7444	15.11.17 23:51:19,409687000	+01:00 SYS	ORA\$AT_OS_OPT_SY_435	(null)	SUCCEEDED	0	(null)	15.11.17 23:51:09,702233000
7420	15.11.17 23:41:19,399960000	+01:00 SYS	ORA\$AT_OS_OPT_SY_433	(null)	SUCCEEDED	0	(null)	15.11.17 23:41:08,074054000
7396	15.11.17 23:31:22,302080000	+01:00 SYS	ORA\$AT_OS_OPT_SY_431	(null)	SUCCEEDED	0	(null)	15.11.17 23:31:06,055907000
7360	15.11.17 22:22:16,037973000	+01:00 SYS	ORA\$AT_OS_OPT_SY_411	(null)	SUCCEEDED	0	(null)	15.11.17 22:21:45,555700000



Autotask Infrastruktur

- Generiert Scheduler Jobs für bestimmte Maintenance-Aufgaben, u.a. Statistiksammlung
- Einmal-Jobs!
- Eigene Views
 - `DBA_AUTOTASK_CLIENT`
(u.a. `CLIENT_NAME = 'auto optimizer stats collection'`)
 - `DBA_AUTOTASK_WINDOW_CLIENT` (welche Scheduler Windows für welchen Client)
- Eigene APIs
 - Package `DBMS_AUTO`  `TASK_ADMIN`



Scheduler Windows

- Montags bis Freitags jeweils um 22:00 Uhr, Dauer 4 Stunden
- Samstags und Sonntags jeweils um 6:00 Uhr, Dauer 20 Stunden
- **Bezeichnung** `MONDAY_WINDOW`, `TUESDAY_WINDOW`, ...



Präferenzen für die Statistiksammlung

- Parameter auf **Tabellenebene** (`DBMS_STATS.SET_TABLE_PREFS`) oder **global** (`DBMS_STATS.SET_GLOBAL_PREFS`) änderbar
 - `DBMS_STATS.SET_SCHEMA_PREFS` und `DBMS_STATS.SET_DATABASE_PREFS` für Präferenzen auf alle Tabellen im Schema (bzw. in der Datenbank)
 - Keine Voreinstellung für Schema- bzw. Datenbankebene



Präferenzen für die Statistiksammlung

- Beispiele (schon seit 11.2):
 - `STALE_PERCENT` (Wenn Prozentsatz an Daten in der Tabelle geändert wird neu berechnet; Default 10)
 - `NO_INVALIDATE` (Wann sollen Ausführungspläne invalidiert werden wenn neue Statistiken für abhängige Objekte vorhanden sind; `FALSE` für sofort; Default `AUTO_INVALIDATE`)
- Abfrage auf Tabellebene mit `DBA_TAB_STAT_PREFS`, globale Ebene mit `DBMS_STATS.GET_PREFS` (bzw. `SYS.OPTSTAT_HIST_CONTROL$`)



Neue Table-Präferenzen mit 12.2

- `AUTO_STAT_EXTENSIONS`: Automatische Erzeugung von Extended Statistics (Spaltengruppen); Default `OFF` (War mit 12.1 grundsätzlich `ON`!)
- `GLOBAL_TEMP_TABLE_STATS`: Statistiken auf temporären Tabellen für alle gleich (`SHARED`) oder für jede Session einzeln (`SESSION`, Default)
- `PREFERENCE_OVERRIDES_PARAMETER`: Parameter für `DBMS_STATS.GATHER_XXX_STATS` ignorieren, Präferenzen verwenden; Default `FALSE`

Neue Global-Präferenz in 12.2

- `TABLE_CACHED_BLOCKS`: Annahme für Anzahl Tabellenblöcke im Cache bei der Berechnung des Clustering Factors (Default 1)
- Sowie einige dokumentierte und nicht dokumentierte mehr... (`SYS.OPTSTAT_HIST_CONTROL$`)

Adaptive Features

Hintergrund: Adaptive Features

- Optimizer Grundsatz bis Oracle 10g:

Wir können den perfekten Plan berechnen, wenn die Statistiken gut genug sind!

- Ab 11g:

Wenn wir erkennen, dass das nicht geklappt hat, dann...



||g Beispiel: Adaptive Cursor Sharing

- Statement hat (nicht allzu viele) Bindevariable
- Nutzung der Bindevariablen in Filtern mit Spalten, die Histogramme besitzen
- Mitverfolgung von Ausführungszeiten (Bind Sensitive)
- Bei stark unterschiedlichen Ausführungszeiten ggfs. Erzeugung von Child Cursors mit unterschiedlichen Ausführungsplänen (Bind Aware)
- Nachverfolgen in `V$SQL (BIND_SENSITIVE, BIND_AWARE)`
- Komponente `KGLH0` in `V$SGASTAT`



Zwei neue Adaptive Features in 12.1

- Adaptive Plans: Optimizer kann *on the fly* von Nested Loop Join zu Hash Join wechseln und umgekehrt
 - Nur beim ersten Join eines Ausführungsplans
 - Entscheidung basierend auf Abgleich der berechneten zur tatsächlichen Kardinalität
 - Verfügbar in der Enterprise Edition
- Adaptive Dynamic Statistics
 - Erweiterung des lange bekannten Dynamic Samplings
 - Basiert ebenfalls auf Cardinality Feedback
 - Aktiviert sich auch bei frischen Statistiken sowie
`optimizer_dynamic_sampling = 2`
- Beide gesteuert über einen Parameter:
`optimizer_adaptive_features` (Default TRUE)



Adaptive Plans...?

```
CREATE TABLE master (  
    mid NUMBER,  
    c VARCHAR2(100 CHAR)  
);  
  
CREATE TABLE detail (  
    mid NUMBER,  
    did NUMBER,  
    d VARCHAR2(100 CHAR)  
);  
  
ALTER TABLE master ADD CONSTRAINT master_pk PRIMARY KEY (mid);  
  
ALTER TABLE detail ADD CONSTRAINT detail_pk PRIMARY KEY (mid, did);  
  
ALTER TABLE detail ADD CONSTRAINT detail_r_master  
FOREIGN KEY (mid) REFERENCES master;  
  
EXEC dbms_stats.gather_schema_stats('DOAG');
```



Adaptive Plans...?

```
SELECT m.mid, d.did, m.c, d.d  
FROM master m JOIN detail d ON m.mid = d.mid  
WHERE m.mid BETWEEN 100 AND 150;
```



OPERATION	OBJECT_NAME	OPTIONS	CARDINALITY	COST
SELECT STATEMENT				52 14
HASH JOIN				52 14
Access Predicates M.MID=D.MID				
NESTED LOOPS				52 14
NESTED LOOPS				
STATISTICS COLLECTOR				
TABLE ACCESS	MASTER	BY INDEX ROWID BATCHED		52 3
INDEX	MASTER_PK	RANGE SCAN		52 2
Access Predicates AND M.MID >= 100 M.MID <= 150				
INDEX	DETAIL_PK	RANGE SCAN		1040 5
Access Predicates M.MID=D.MID				
Filter Predicates AND D.MID <= 150 D.MID >= 100				
TABLE ACCESS	DETAIL	BY INDEX ROWID		1 11
TABLE ACCESS	DETAIL	BY INDEX ROWID BATCHED		1040 11
INDEX	DETAIL_PK	RANGE SCAN		1040 5
Access Predicates AND D.MID >= 100 D.MID <= 150				

OBJECT_NAME=MASTER_PK



Adaptive Dynamic Statistics

- Nach Migration auf 12.1: CPU Auslastung auf 100%, riesige Run Queue (in mindestens 70% der Fälle)
- Zusatzmisere:
 - Erzeugte Dynamic Statistics landen im Result Cache
 - Wenn dieser nur Standardgröße hat: Contention! (Ja, es gibt ein Result Cache Latch!)

Parameter

- Neu in 12.1:
 - `optimizer_adaptive_features = TRUE | FALSE (12.1)`
- Desupported in 12.2, dafür neu
 - `optimizer_adaptive_plans = TRUE | FALSE`
 - `optimizer_adaptive_statistics = TRUE | FALSE`
- 12.2 Parameter auch für 12.1 als Backport verfügbar
 - Enthalten in Proactive Bundle Patch vom Oktober 2017



Empfehlung für 12.1

- MOS Doc 2312911.1:

Recommendations for Adaptive Features in Oracle Database 12c Release 1 And Changes to Adaptive Feature in Release 2 (Adaptive Features, Adaptive Statistics and 12c SQL Performance)



Fragen & Kontakt

- Mail: dierk.lenz@hl-services.de
- Web: www.hl-services.de
- Blog: blog.hl-services.de
- Twitter: @ora1578
- Live: Hier in der Ausstellung, Stand 236
Ebene 2 (gelb), 2. Stand links von der Rolltreppe

Vielen Dank für Ihre Aufmerksamkeit!

- Besuchen Sie auch unseren Vortrag:
 - **Hochverfügbarkeit mit Standard Edition - das geht!**
Johannes Kraus, 23.11.2017 12:00h, Foyer Tokio