

Datenbankadministration

8. Anfrageoptimierung II

AG DBIS University of Kaiserslautern, Germany

Karsten Schmidt kschmidt@informatik.uni-kl.de
(Vorlage TU-Dresden)

Wintersemester 2008/2009



- **Vorherige LV**

- Anfrageplan
- Indizes
- Materialisierte Sichten

- **Heute**

- Statistiken
 - zur Erstellung des optimalen Anfrageplanes notwendig
 - Runstats
- Statistikprofile
- Reorganisation der Daten

- **Statistiken**

- Basis der Kostenschätzung während der Anfrageoptimierung
- Erhebung für Datenbankobjekte, z.B.:
 - Anzahl Sätze einer Relation
 - Anzahl verschiedener Werte eines Attributes
 - Anzahl Knoten in einem B+-Baum
 - ...
- Hinterlegt im Data-Dictionary
 - SYSCAT.TABLES
 - SYSCAT.COLDIST
 - SYSCAT.COLUMNS
 - SYSCAT.INDEXES
 - etc.
- werden erstellt mit Hilfe des RUNSTATS-Kommandos

Statistikerstellung

- **Erstellen von Tabellenstatistiken**

- abgelegt in `SYSCAT.TABLES`
 - Erstellungszeit (`STATS_TIME`)
 - Kardinalität (`CARD`)
 - Gesamtzahl Seiten (`FPAGES`)
 - Gesamtzahl nicht-leerer Seiten (`NPAGES`)
 - Anzahl von Überlauftupeln (`OVERFLOW`)
- RUNSTATS-Aufruf
 - `RUNSTATS ON TABLE <tab-schema.tab-name>`
- Beispiel
 - `RUNSTATS ON TABLE DB2INST1.LINEITEM`
 - `SELECT TABNAME, STATS_TIME, CARD, NPAGES, FPAGES FROM SYSCAT.TABLES WHERE TABNAME = 'LINEITEM'`

TABNAME	STATS_TIME	CARD	NPAGES	FPAGES
LINEITEM	2008-11-30 10:17	60005	2083	2112

- **Erstellen von Attributstatistiken**

- abgelegt in `SYSCAT.COLUMNS`
 - Anzahl verschiedener Werte (`COLCARD`)
 - Zweithöchster Wert (`HIGH2KEY`)
 - Zweitniedrigster Wert (`LOW2KEY`)
 - Durchschnittliche Größe (`AVGCOLLEN`)
 - Anzahl Nullwerte (`NUMNULLS`)
- RUNSTATS-Aufruf
 - `RUNSTATS ON TABLE` <tab-schema.tab-name>
 - `ON COLUMN` (<col-name> [, <col-name>]*)
 - `ON {ALL|KEY} COLUMNS`
 - Standard: `ON ALL COLUMNS`
- Beispiel

COLNAME	COLCARD	HIGH2KEY	LOW2KEY	AVGCOLLEN	NUMNULLS
L_QUANTITY	50	49	2	8	0

- **Attributstatistiken mit Datenverteilung**

- Datenverteilung
 - häufige Werte (*frequent values*)
 - p-Quantile
 - p% der Werte sind kleiner oder gleich dem p-Quantil
 - Median: 50%-Quantil
- abgelegt als Histogramm in `SYSCAT.COLDIST`
 - derzeit keine Unterstützung für Attributkombinationen
- Metainformationen in `SYSCAT.COLUMNS`
 - Anzahl der gespeicherten häufigsten Werte (`NMOSTFREQ`)
 - Anzahl der Quantile (`NQUANTILES`)
- RUNSTATS-Aufruf
 - `RUNSTATS ... [ON ...]`
`WITH DISTRIBUTION ON COLUMNS (`
 `<col-name> [NUM_FREQUVALUES <int>]`
 `[NUM_QUANTILES <int>] [, ...]*`
`)`

Materialisierung der Datenverteilung

- **Datenverteilung** (`SYSCAT.COLDIST`)
 - Schema-, Tabellen-, Spaltenname (`TABSCHEMA, TABNAME, COLNAME`)
 - Typ (`TYPE`): **F** für häufiger Wert, **Q** für Quantil
 - Attributausprägung (`COLVALUE`)
 - Anzahl Tupel (`VALCOUNT`)
 - Beispiel (`LINEITEM.L_ORDERKEY`, 2 häufigste Werte, 5 Quantile)
 - (1) am häufigsten auftretender Wert; „10340“ (90-mal)
 - (2) 30003 Werte sind kleiner oder gleich „30151“ (Median)
 - hier: annähernd gleichverteilte Daten

	COLNAME	TYPE	SEQNO	COLVALUE	VALCOUNT
	-----	-----	-----	-----	-----
1	L_ORDERKEY	F	1	10340	90
	L_ORDERKEY	F	2	17505	90
	L_ORDERKEY	Q	1	33	30
	L_ORDERKEY	Q	2	15778	15001
2	L_ORDERKEY	Q	3	30151	30003
	L_ORDERKEY	Q	4	45120	45004
	L_ORDERKEY	Q	5	59810	60005

- **Erstellen von Indexstatistiken**

- abgelegt in `SYSCAT.INDEXES`
 - Anzahl Blattknoten (`NLEAF`)
 - Höhe des B+-Baumes (`NLEVELS`)
 - Anzahl verschiedener Werte
(`[FIRST|...|FIRST4|FULL]KEYCARD`)
 - Clustering der Daten (`CLUSTERRATIO` oder `CLUSTERFACTOR`, 0-100)
 - ...
- RUNSTATS-Aufruf
 - `RUNSTATS ON TABLE <tab-schema.tab-name>`
`ON COLUMNS ...WITH DISTRIBUTION ...AND | FOR`
 - `[DETAILED] INDEX <tab-schema.idx-name>`
 - `[DETAILED] INDEXES ALL`
- **DETAILED**: verbesserte Index-Statistiken
 - z.B. `CLUSTERFACTOR` statt `CLUSTERRATIO`

Beispiel für Indexstatistik

- **Beispiel für Indexstatistik**

- Index `LINEITEM_PK`
 - Primärschlüsselindex der Relation `LINEITEM`
 - besteht aus 2 Attributen: `L_ORDERKEY`, `L_LINENUMBER`
- `RUNSTATS ON TABLE DB2INST1.LINEITEM FOR INDEX DB2INST1.LINEITEM_PK`
- `SELECT INDNAME, NLEAF, NLEVELS, FIRSTKEYCARD, FIRST2KEYCARD, FULLKEYCARD, CLUSTERRATION FROM SYSCAT.INDEXES WHERE INDNAME='LINEITEM_PK'`

INDNAME	NLEAF	NLEVELS	FIRSTKEY	FIRST2KEY	FULLKEY	CLUSTERRATIO
LINEITEM_PK	285	3	14957	60005	60005	100

- **Weitere Optionen**

- Konkurrerender Zugriff
 - Ziel: Datenbank nicht vollständig sperren
 - andere Transaktionen dürfen Lesen bzw. Lesen & Schreiben
 - `ALLOW {READ|WRITE} ACCESS`
- Statistikerstellung auf Basis einer Stichprobe
 - Ziel: möglichst kurze Blockierung des DBMS
 - Stichprobenerhebung auf Tupelebene (`BERNOULLI`) oder Seitenebene (`SYSTEM`)
 - `TABLESAMPLE BERNOULLI|SYSTEM(<percentage>)`
- Komplexes Beispiel
 - `RUNSTATS ON TABLE DB2INST1.LINEITEM
ON ALL COLUMNS
WITH DISTRIBUTION ON COLUMNS (L_EXTENDEDPRICE)
AND INDEXES ALL
ALLOW READ ACCESS
TABLESAMPLE SYSTEM(10);`

● Statistikprofile

- speichern Konfiguration für Statistikerhebung
- definiert pro Relation
- abgelegt in `SYSCAT.TABLES`, Attribut `STATISTICS_PROFILE`
- Definition eines Profils
 - `RUNSTATS ON TABLE <tab-schema.tabname> ...`
`{SET|UPDATE} PROFILE [ONLY] | SET PROFILE NONE`
 - `SET/UPDATE`: Setzen/Aktualisieren und Ausführen des Statistikprofils
 - `ONLY`: neues Profil nicht sofort Ausführen
 - `NONE`: Statistikprofil löschen
 - seit DB2 v8.2: auch automatische Erstellung möglich
- Nutzung eines Profils
 - `RUNSTATS ON TABLE <tab-schema.tabname> USE PROFILE`

- **Beispiel**

- Erstellung eines RUNSTATS-Profiles

- `RUNSTATS ON TABLE DB2INST1.LINEITEM
ON ALL COLUMNS
AND INDEXES ALL
SET PROFILE ONLY`

- Nutzung eines definierten Profils

- `RUNSTATS ON TABLE DB2INST1.LINEITEM USE PROFILE`

Reorganisation

- **Analyse der physischen Datenorganisation**

- Überprüft, ob Reorganisation von Relationen / Indizes notwendig
- basierend auf RUNSTATS Statistiken
- Analyse von Relationen mit ihren Indizes
 - **REORGCHK**
`ON TABLE <tab-schema.tab-name> | USER | SYSTEM | ALL`
 - **USER**: Analyse aller Relationen des aktuellen Nutzers
 - **SYSTEM**: Analyse aller Systemrelationen
 - **ALL**: Analyse aller Relationen
- Analyse aller Relationen eines Schemas
 - **REORGCHK ON SCHEMA <tab-schema>**
- Ergebnis
 - Vorschlag für Relationen / Indizes für Reorganisation
 - bestimmt durch eine Reihe von Heuristiken (z.B. F1)

Analyse der physischen Datenorganisation

● Heuristiken für Tabellen

- F1: weniger als 5% Überlauf tupel
- F2: mind. 70% Speicherauslastung
- F3: mind. 80% der Seiten belegt

● Beispiel

- `REORGCHK ON TABLE DB2INST1.LINEITEM`

F1: $100 * \text{OVERFLOW} / \text{CARD} < 5$

F2: $100 * (\text{Effektive Speicherauslastung von Datenseiten}) > 70$

F3: $100 * (\text{Erforderliche Seiten} / \text{Seiten insgesamt}) > 80$

SCHEMA.NAME	CARD	OV	NP	FP	ACTBLK	TSIZE	F1	F2	F3	REORG
Tabelle: DB2INST1.LINEITEM	60005	0	2083	2084	-	8220685	0	98	100	---

Kardinalität # Überlauf tupel # belegte Seiten # Seiten gesamt geschätzte Relationengröße Empfehlung
- Kein REORG nötig
* REORG nötig

Analyse der physischen Datenorganisation

● Heuristiken für Indizes

- F4: Cluster Rate mind. 80%
- F5: max. 50% freier Speicher in Blattknoten
- F6: neu generierter Index ist nicht flacher
- F7: weniger als 20% der Tupel als gelöscht markiert
- F8: weniger als 20% der Blätter leer (ohne Zählung gelöschter Tupel)

● Beispiel

F4: CLUSTERATIO oder normalisierter CLUSTERFACTOR > 80
F5: $100 * (\text{Für Blattseiten verwendeter Speicherbereich} / \text{Verfügbare Speicherbereich für nicht leere Blattseiten}) > \text{MIN}(50, (100 - \text{PCTFREE}))$
F6: $(100 - \text{PCTFREE}) * (\text{Verfügbare Speicherbereich in einem Index mit einer Ebene weniger} / \text{Erforderlicher Speicherbereich für alle Schlüssel}) < 100$
F7: $100 * (\text{Anzahl pseudogelöschter Satz-IDs} / \text{Gesamtzahl Satz-IDs}) < 20$
F8: $100 * (\text{Anzahl pseudoleerer Blattseiten} / \text{Gesamtzahl Blattseiten}) < 20$

SCHEMA.NAME	INDCARD	LEAF	ELEAF	LVLS	NDEL	KEYS	LEAF RECSIZE	NLEAF RECSIZE	LEAF_PAGE OVERHEAD	NLEAF_PAGE OVERHEAD	F4	F5	F6	F7	F8	REORG

Tabelle: DB2INST1.LINEITEM																
DB2INST1 LINEITEM_PART	60005	136	0	2	0	420	4	4	710	710	100	92	-	0	0	-----
DB2INST1 LINEITEM_PARTSUPPKEY	60005	134	0	2	0	734	8	8	566	566	89	91	-	0	0	-----
DB2INST1 LINEITEM_PK	60005	318	0	3	0	60005	8	8	566	566	100	91	49	0	0	-----
DB2INST1 LINEITEM_SHIPMODE	60005	130	0	2	0	7	10	10	516	516	79	90	-	0	0	*-----

Kardinalität

Index Seiten

Index Höhe

Indexschlüssel

Empfehlung
- Kein REORG nötig
* REORG nötig

- **Reorganisation mit REORG**

- Relation organisieren
 - Neuordnung der Tupel
 - `REORG TABLE <tab-schema.tab-name>`
- Index reorganisieren
 - Neuaufbau von Indizes
 - `REORG TABLE <tab-schema.tab-name>`
`INDEX <tab-schema.idx-name>`
 - `REORG INDEXES ALL FOR TABLE <tab-schema.tab-name>`
- Beispiel
 - `REORG TABLE DB2INST1.LINEITEM`
 - `REORG TABLE DB2INST1.LINEITEM`
`INDEX DB2INST1.LINEITEM_SHIPMODE`

- **Statistiken**

- Einflussgrößen der Anfrageoptimierung
- Pflege mit Hilfe des RUNSTATS-Kommandos

- **Reorganisation**

- Analyse mit REORGCHK
- Reorganisation mit REORG