
ORACLE®

ORACLE

1997

Effiziente Speicherung für SAP

Jörn Bartels
Architect
Oracle Database Server Technologies

ORACLE DATABASE 8.0

Effiziente Speicherungsformen

- Ziele
- Index Komprimierung
- Index Organized Tables
- Ergebnisse

Ziele

- Platz
  - Weniger Plattenplatz
  - Weniger Hauptspeicher
- Performance
  - Weniger IO
  - Besseres Caching
- Vor / Nachteile
  - Etwas weniger oder etwas mehr CPU Belastung

Index Komprimierung

- Komprimierung in den Blättern des Baumes
- Teilung des Schlüssels in:
  - Präfix Teil und
  - Suffix Teil
- Gleiche Präfixe werden nur einmal gespeichert

Beispiel

1	A	X	1	A	1	ROWID1
1	A	X	1	B	2	ROWID2
1	A	X	2	A	3	ROWID3
1	A	Y	1	B	4	ROWID4
1	A	Y	3	C	5	ROWID5
1	A	Y	3	C	6	ROWID6
1	A	Y	3	D	7	ROWID7
1	B	X	1	A	1	ROWID8
1	B	X	1	A	2	ROWID9
1	B	X	1	C	3	ROWID10
1	B	X	3	A	4	ROWID11
1	B	X	3	C	5	ROWID12
1	B	X	3	C	6	ROWID13

Beispiel – compress 1 – 12 Werte weniger

1	A	X	1	A	1	ROWID1
	A	X	1	B	2	ROWID2
	A	X	2	A	3	ROWID3
	A	Y	1	B	4	ROWID4
	A	Y	3	C	5	ROWID5
	A	Y	3	C	6	ROWID6
	A	Y	3	D	7	ROWID7
	B	X	1	A	1	ROWID8
	B	X	1	A	2	ROWID9
	B	X	1	C	3	ROWID10
	B	X	3	A	4	ROWID11
	B	X	3	C	5	ROWID12
	B	X	3	C	6	ROWID13

Beispiel – compress 2 – 22 Werte weniger

1	A	X	1	A	1	ROWID1
		X	1	B	2	ROWID2
		X	2	A	3	ROWID3
		Y	1	B	4	ROWID4
		Y	3	C	5	ROWID5
		Y	3	C	6	ROWID6
		Y	3	D	7	ROWID7
1	B	X	1	A	1	ROWID8
		X	1	A	2	ROWID9
		X	1	C	3	ROWID10
		X	3	A	4	ROWID11
		X	3	C	5	ROWID12
		X	3	C	6	ROWID13

Beispiel – compress 3 – 30 Werte weniger

1	A	X	1	A	1	ROWID1
			1	B	2	ROWID2
			2	A	3	ROWID3
1	A	Y	1	B	4	ROWID4
			3	C	5	ROWID5
			3	C	6	ROWID6
			3	D	7	ROWID7
1	B	X	1	A	1	ROWID8
			1	A	2	ROWID9
			1	C	3	ROWID10
			3	A	4	ROWID11
			3	C	5	ROWID12
			3	C	6	ROWID13

Beispiel – compress 4 – 28 Werte weniger

1	A	X	1	A	1	ROWID1
				B	2	ROWID2
1	A	X	2	A	3	ROWID3
1	A	Y	1	B	4	ROWID4
1	A	Y	3	C	5	ROWID5
				C	6	ROWID6
				D	7	ROWID7
1	B	X	1	A	1	ROWID8
				A	2	ROWID9
				C	3	ROWID10
1	B	X	3	A	4	ROWID11
				C	5	ROWID12
				C	6	ROWID13

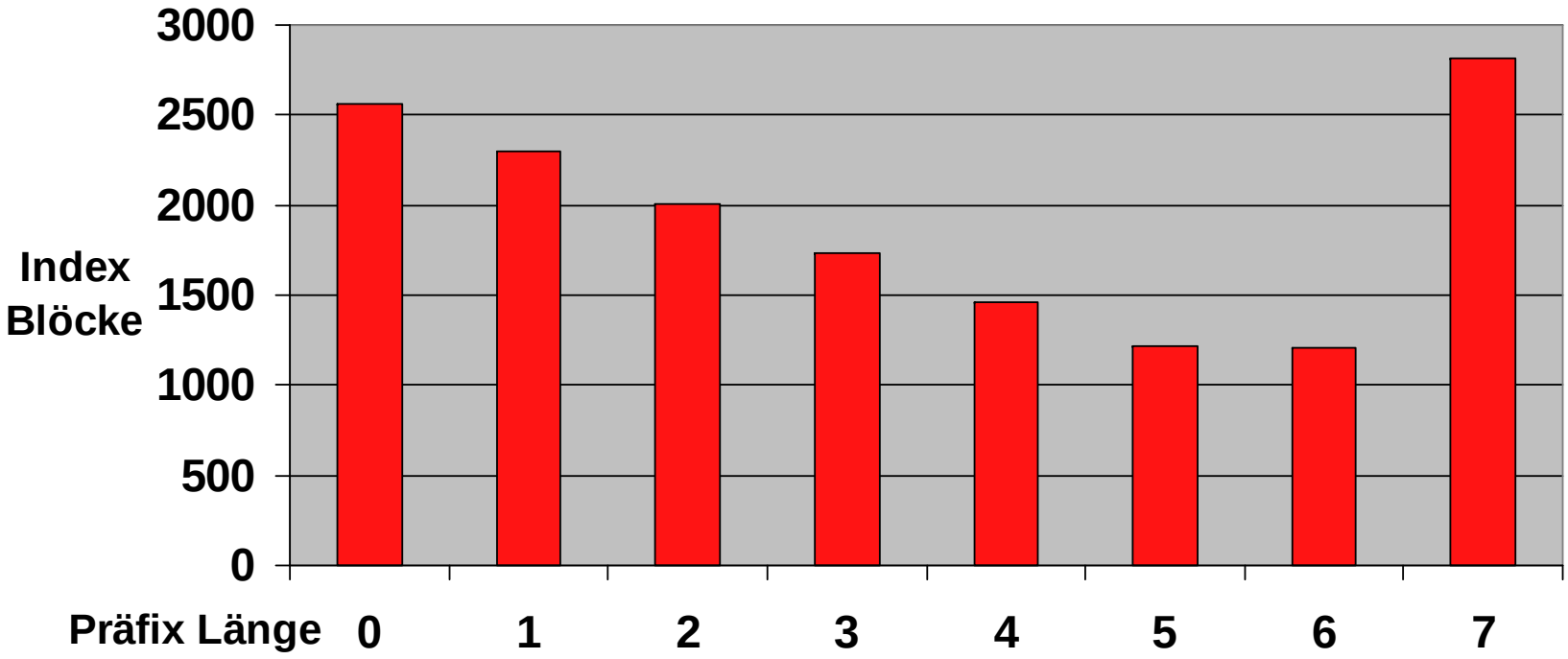
Beispiel – compress 5 – 15 Werte weniger

1	A	X	1	A	1	ROWID1
1	A	X	1	B	2	ROWID2
1	A	X	2	A	3	ROWID3
1	A	Y	1	B	4	ROWID4
1	A	Y	3	C	5	ROWID5
					6	ROWID6
1	A	Y	3	D	7	ROWID7
1	B	X	1	A	1	ROWID8
					2	ROWID9
1	B	X	1	C	3	ROWID10
1	B	X	3	A	4	ROWID11
1	B	X	3	C	5	ROWID12
					6	ROWID13

Beispiel – compress 6 – kein Wert weniger

1	A	X	1	A	1	ROWID1
1	A	X	1	B	2	ROWID2
1	A	X	2	A	3	ROWID3
1	A	Y	1	B	4	ROWID4
1	A	Y	3	C	5	ROWID5
1	A	Y	3	C	6	ROWID6
1	A	Y	3	D	7	ROWID7
1	B	X	1	A	1	ROWID8
1	B	X	1	A	2	ROWID9
1	B	X	1	C	3	ROWID10
1	B	X	3	A	4	ROWID11
1	B	X	3	C	5	ROWID12
1	B	X	3	C	6	ROWID13

Beispiel – Präfix Länge



Ermittlung des optimalen Präfix

- Offline
  - ANALYZE INDEX <xxx> VALIDATE STRUCTURE
  - SELECT opt_cmpr_count, opt_cmpr_pctsave FROM index_stats
- Online
  - Package IND_COMP aus Hinweis 1109743
  - EXEC ind_comp.get_column('SAPR3' , 'GLPCA')

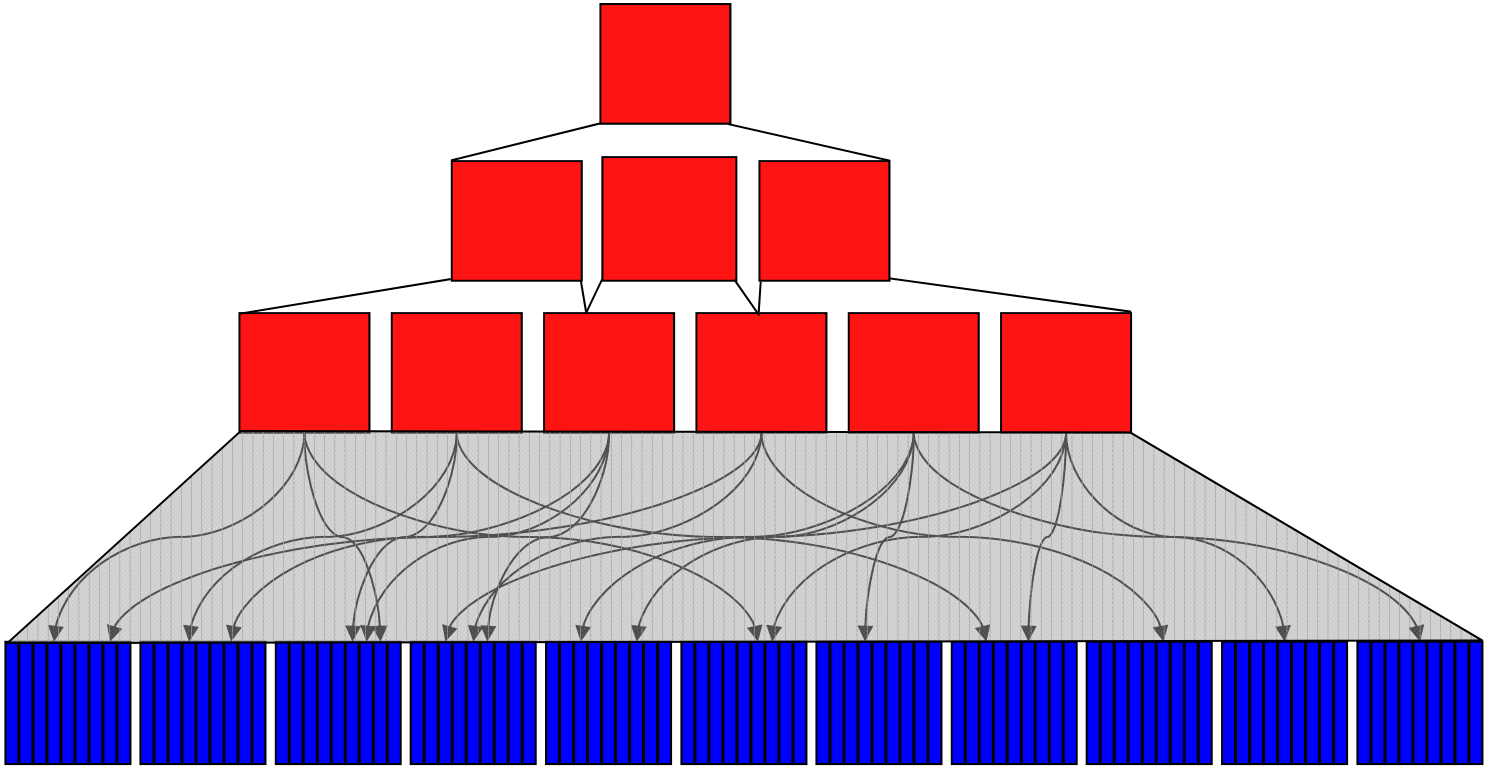
Ergebnisse der Index Komprimierung

- Index GLPCA~1 von 18 GB auf 4,5 GB reduziert.
  - 75% kleinerer Index
- Indices der 10 größten Tabellen von 224 GB auf 138 GB reduziert.
  - 40% kleinere Indices

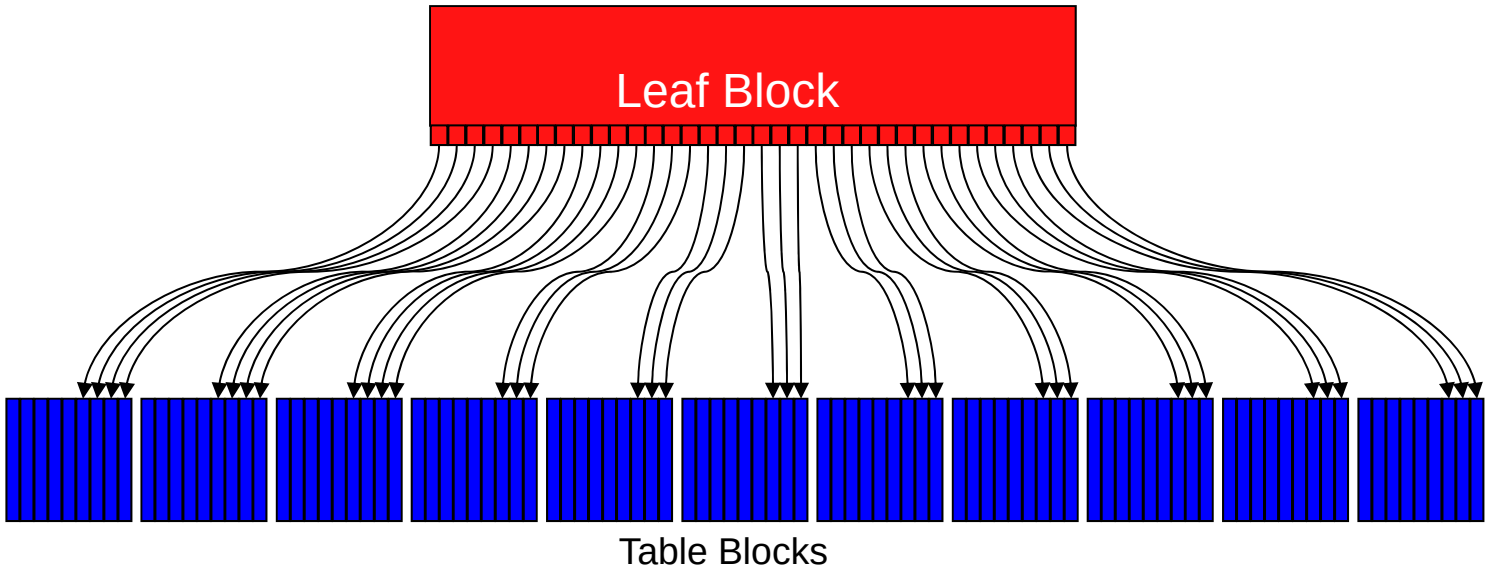
Index Organized Tables

- Speicherung aller Daten der Tabelle im B*Baum
- Kein Datensegment – bei kleinen Tabellen 64k gespart
- Schlüsselspalten nícht doppelt gespeichert
- Rowid eingespart

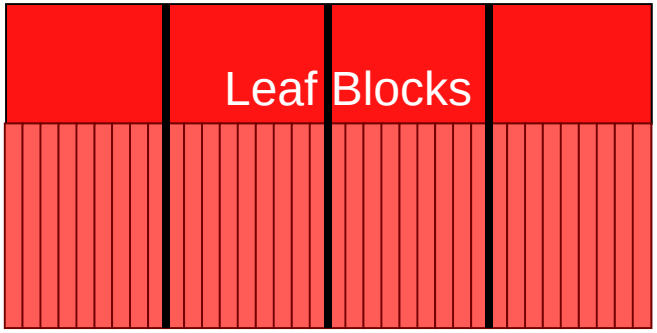
Index mit Heap Tabelle



Index mit Heap Tabelle

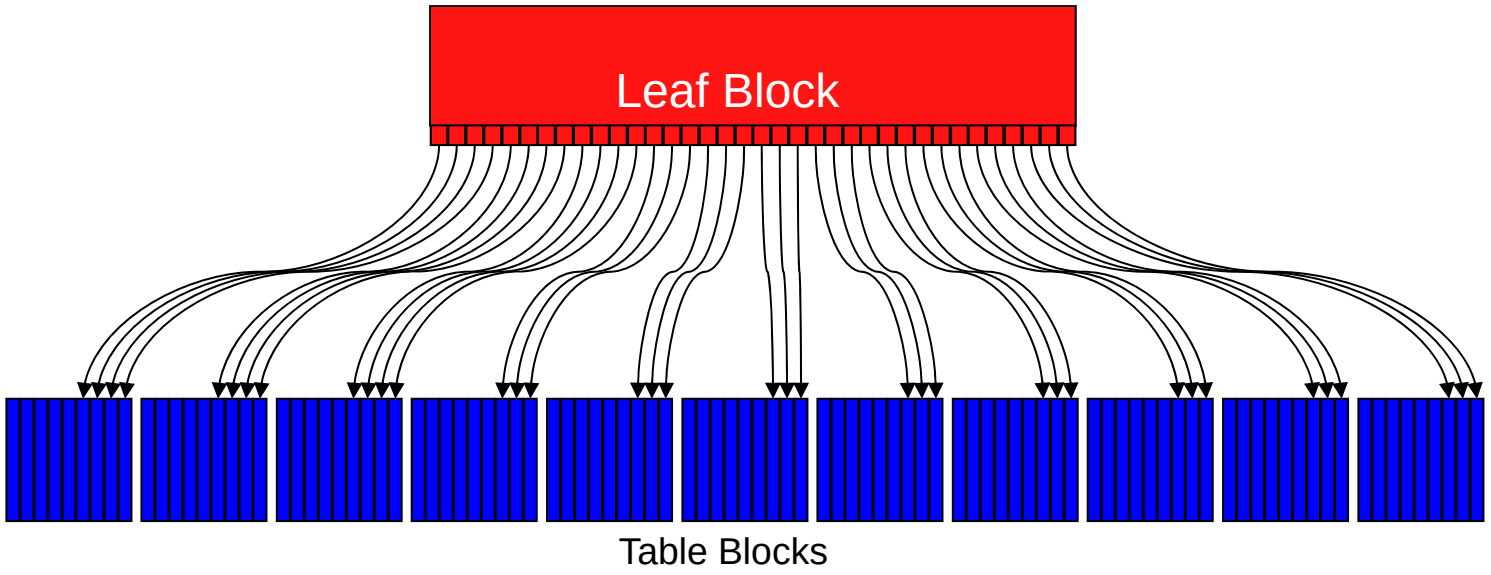


Index Organized Table



Index mit Heap Tabelle - Beispiele

Monoton wachsender Schlüssel



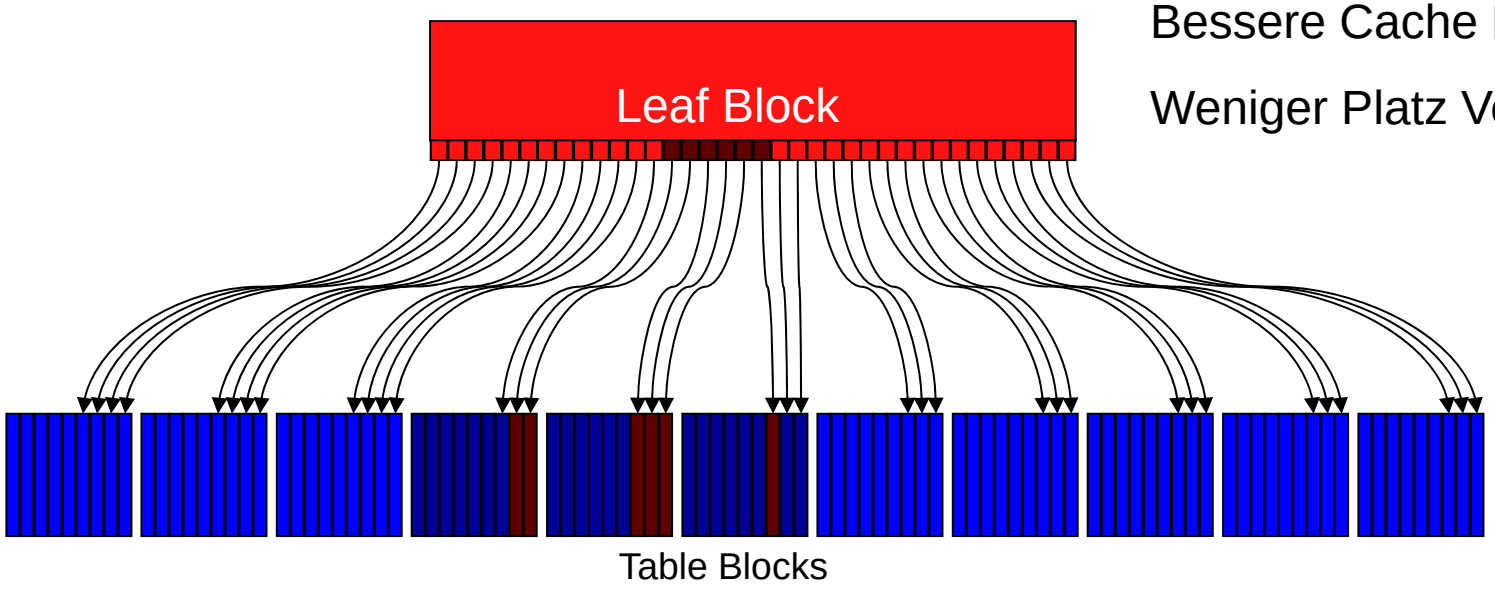
Index mit Heap Tabelle - Beispiele

Monoton wachsender Schlüssel

Weniger IO

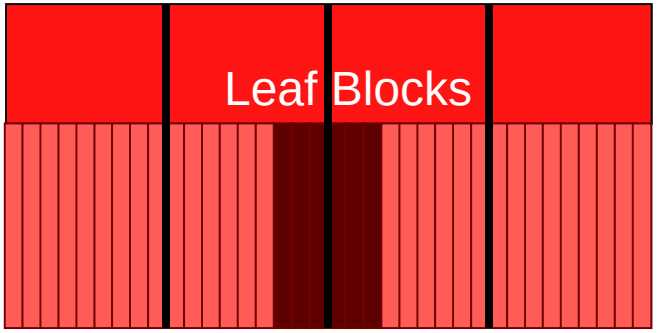
Bessere Cache Nutzung

Weniger Platz Verbrauch



Index Organized Table

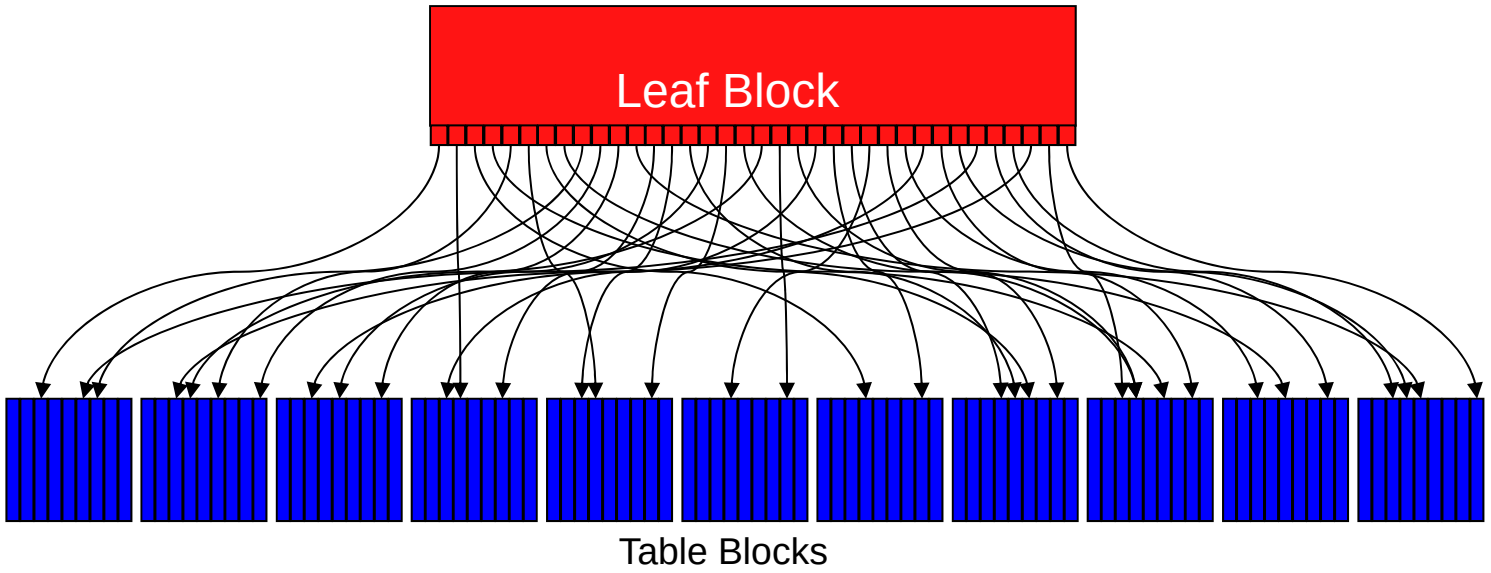
Monoton wachsender Schlüssel



Index mit Heap Tabelle - Beispiele

Buchungen pro Konto

Primary Key mit Foreign Key



Index mit Heap Tabelle - Beispiele

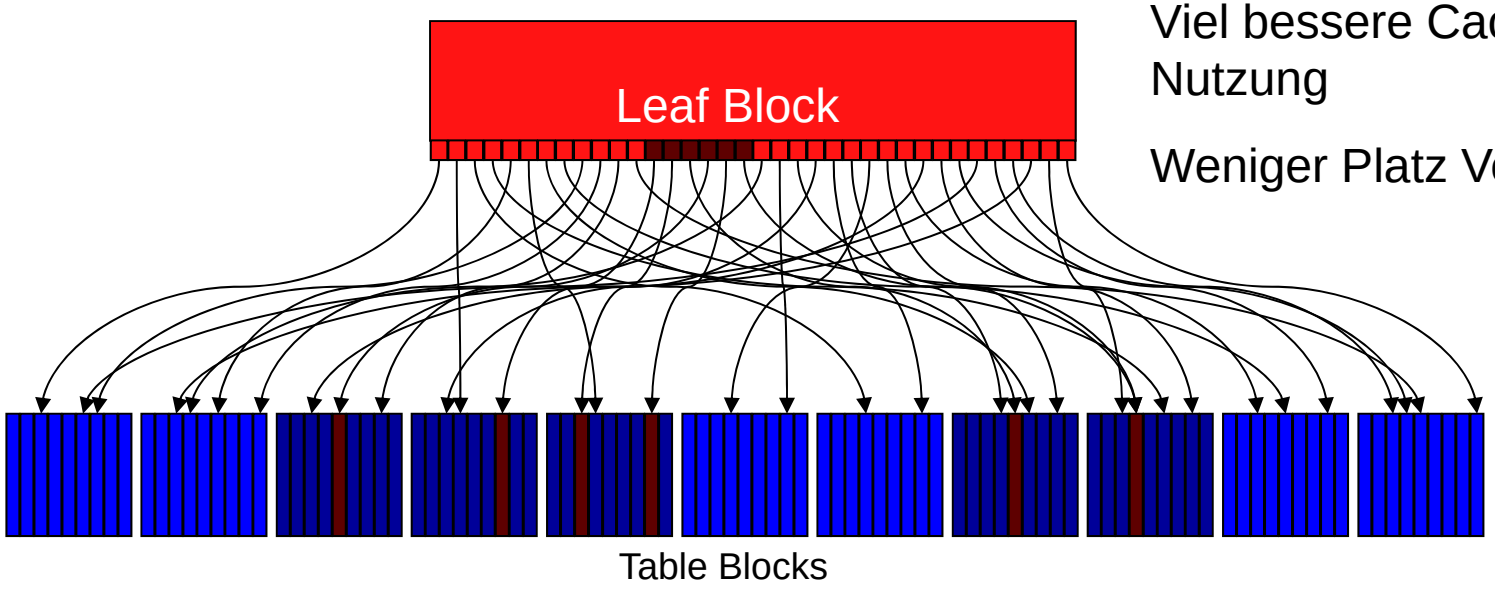
Buchungen pro Konto

Primary Key mit Foreign Key

Viel weniger IO

Viel bessere Cache Nutzung

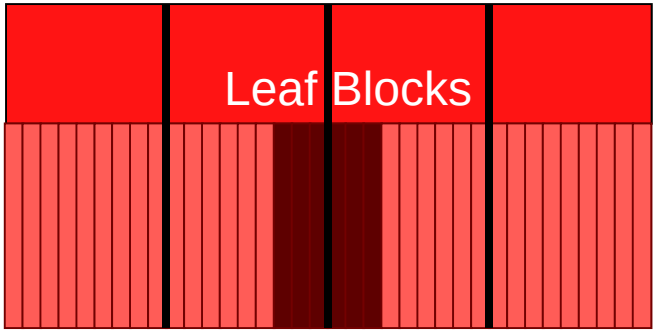
Weniger Platz Verbrauch



Index Organized Table

Buchungen pro Konto

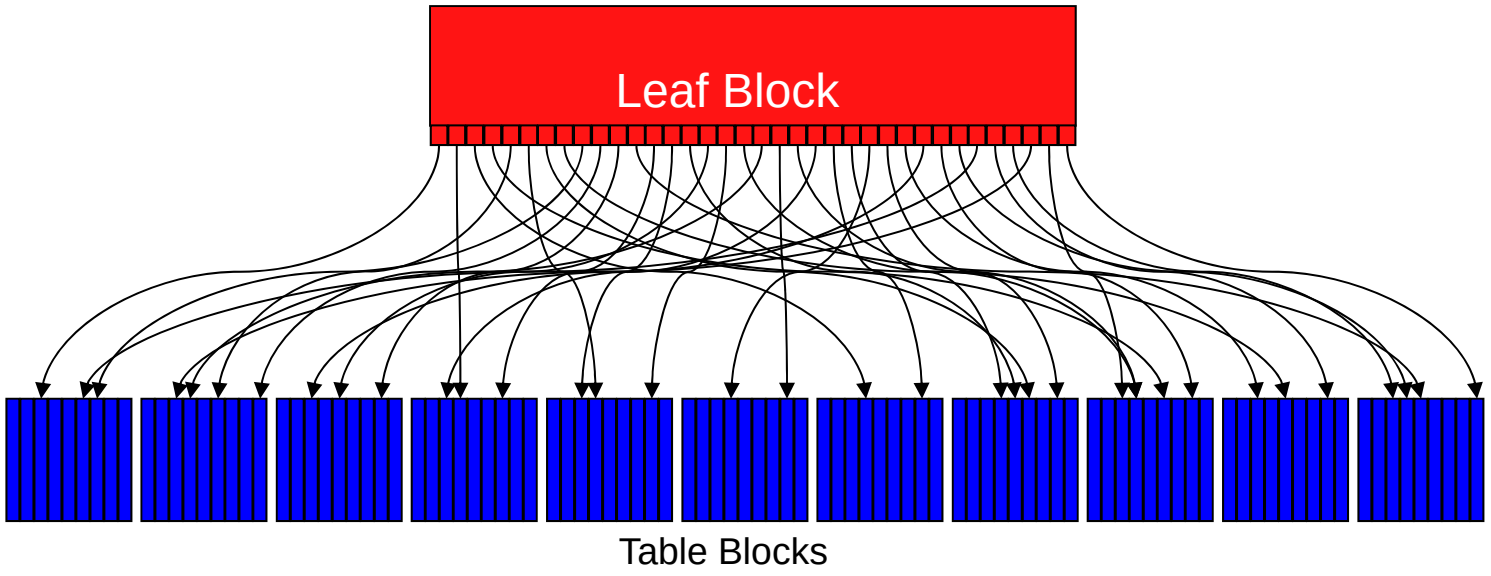
Primary Key mit Foreign Key



Index mit Heap Tabelle - Beispiele

GUID

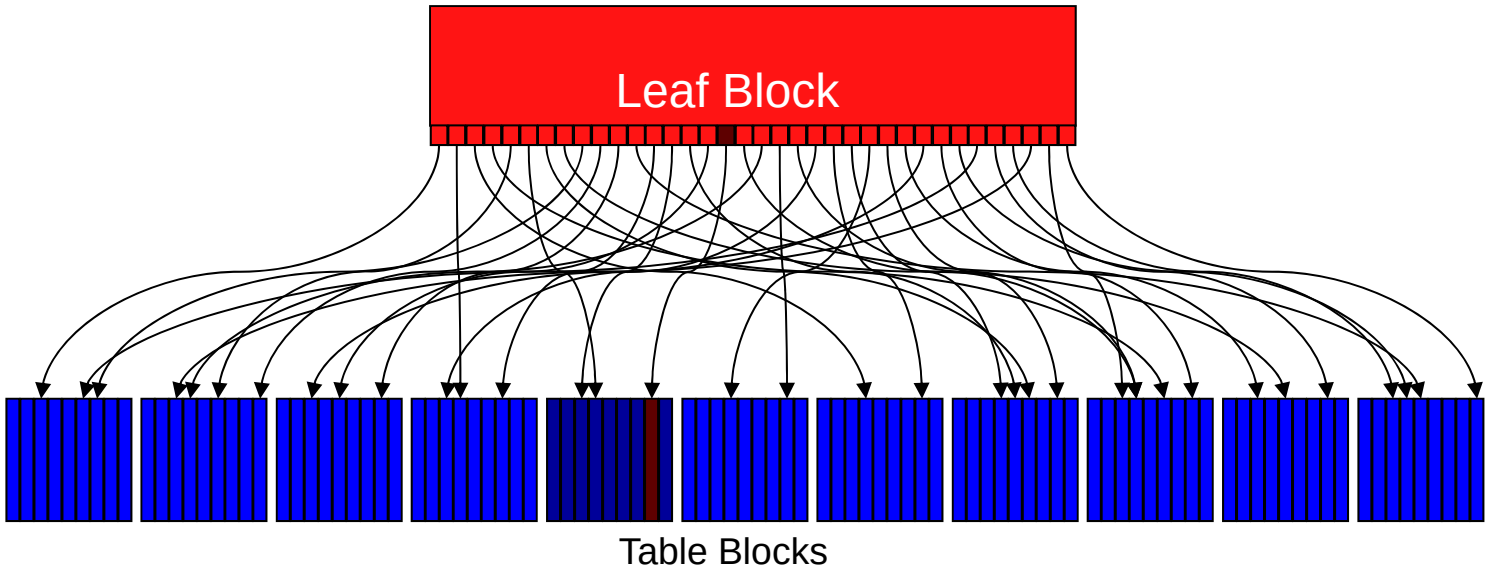
Weniger Platz Verbrauch



Index mit Heap Tabelle - Beispiele

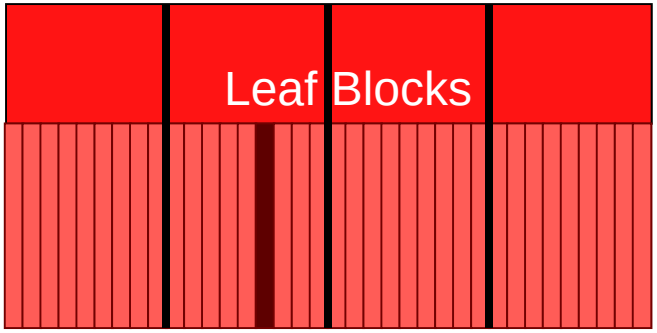
GUID

Weniger Platz Verbrauch



Index Organized Table

GUID



Zweiter Index für IOT

- Speichert zusätzlich die Primary Key Columns
- Zugriff über Guess Rowid – wenn möglich
- Sonst Zugriff über Primay Key
- Braucht mehr Platz
- Meist jedoch genau so schnell, wie bei Heap Tabelle durch Guess Rowid

Syntax für IOT

```
CREATE TABLE  "VBOX$iot"
(  "MANDT",  "KAPPL",  "KOTABNR",  "VAKEY",  "FBUDA",  "VBELN",  "POSNR"
, CONSTRAINT  "VBOX~0$iot"  PRIMARY KEY
(  "MANDT",  "KAPPL",  "KOTABNR",  "VAKEY",  "FBUDA",  "VBELN",  "POSNR")
)  ORGANIZATION INDEX COMPRESS 6
 TABLESPACE PSAPSR3 parallel
 AS SELECT * FROM  "VBOX"
```

Beispiel CDHDR

- Spalten:
  - MANDANT
  - OBJECTCLAS
  - OBJECTID
  - CHANGENR
  - USERNAME
  - UDATE
  - UTIME

Spalten:

TCODE
PLANCHNGNR
ACT_CHNGNO
WAS_PLANND
CHANGE_IND
LANGU
VERSION

- Index:
  - MANDANT
  - OBJECTCLAS
  - OBJECTID
  - CHANGENR

Beispiel CDHDR

- Heap Tabelle
  - Tabelle CDHDR: 31 GB
  - Index CDHDR~0: 18 GB
  - ----------
  - Summe: 49 GB

IOT

IOT CDHDR: 28 GB

Faktor: 1,75

Beispiel VBOX

- Spalten:
  - MANDT
  - KAPPL
  - KOTABNR
  - VAKEY
  - FBUDA
  - VBELN
  - POSNR

Index VBOX~0
- MANDT
- KAPPL
- KOTABNR
- VAKEY
- FBUDA
- VBELN
- POSNR

Index VBOX~A
- MANDT
- KAPPL
- VBELN

Beispiel VBOX

- Heap Tabelle
  - Tabelle VBOX: 122 GB
  - Index VBOX~0: 135 GB
  - Index VBOX~A: 57 GB
  - ----------
  - Summe: 364 GB
- IOT
  - IOT VBOX: 72 GB
  - Index VBOX~A: 139 GB
  - ----------
  - Summe: 211 GB

Faktor: 1,7

Faktor ohne 2. Index: 3,6 !!!!!

Kunden Beispiel

- Tabelle /SAPAPO/MATLSPP
  - Index (MANDT, MATID, LOCID, SIMID)
  - 112 weitere Spalten
- Query
  - SELECT *
    FROM "/SAPAPO/MATLSPP"
    WHERE
    "MANDT" = :A0 AND "SIMID" = :A1 AND "MATID" = :A2 AND "LOCID" = :A3 OR
    "MANDT" = :A4 AND "SIMID" = :A5 AND "MATID" = :A6 AND "LOCID" = :A7 OR
    …. (47)
    "MANDT" = :A196 AND "SIMID" = :A197 AND "MATID" = :A198 AND "LOCID" =
    :A199

Kundenbeispiel

	Tabelle mit einzelnem Index	Komprimiertes IOT
Elapsed time per Exec (s)	0,4	0,04
Reads per exec	17,91	5,36
Executions	429.564	429.705

Simulations Job mit einer Mischung aus select, update, insert und delete

Tabelle mit Index: 44 Minuten

Komprimiertes IOT: 21 Minuten

Package IOTC

- Automatisiert für einzelne oder die n größten Tabellen die Umstellung auf ein komprimiertes IOT
- Erstellt SQL Script zur automatischen Umstellung
- EXEC IOTC.CONVERT (<n>, <parallel>)
- EXEC IOTC.CONVERT (<tab>, <parallel>)

Zusammenfassung

- Queries bis zu 5 mal schneller
- Insert und Deletes bis zu 2 mal schneller
- Ca. 25% Platz Ersparnis
- Wesentlich mehr „Hot“ Daten im Cache

Weitere Infos

- Proceedings of the 26th International Conference on Very Large Databases, Cairo, Egypt, 2000
- SAP Hinweise
  - Note 1109743 –
    Komprimierung von Indexschlüsseln für Oracle-Datenbanken
  - Note 1856270 –
    Performanceverbesserung für Tabellen mit einem einzigen Unique-Index

Hardware and Software

ORACLE

Engineered to Work Together


ORACLE®