



数据技术嘉年华

Data Technology Carnival

云 · 数据 · 智能 - 数聚价值智胜未来

关注公众号回复help
可获取更多经典学习
资料 and 文档，电子书



一次性能优化隐含的秘密

云和恩墨性能优化专家 罗海雄



第七届



数据技术嘉年华
Data Technology Carnival



WHO AM I ?

罗海雄(网名RollingPig)

2011 全国SQL大赛冠军得主

资深的架构师和性能优化专家

有超过15年的企业级系统设计和优化经验

**曾经服务于甲骨文公司亚太区开发者计划部门和
大型制造企业中芯国际**

现任云和恩墨性能优化总监

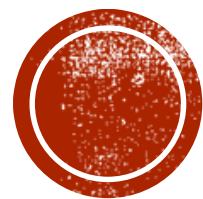


第七屆



数据技术嘉年华
Data Technology Carnival





SQL突然变慢，原因是？



第七届



数据技术嘉年华
Data Technology Carnival



SQL突然变慢

客户在某天日上午，突然发现应用异常缓慢

检查AWR/ASH, 发现有数条SQL执行时间超长，由平常的几十毫秒飙升到十几秒。

很简单的SQL, 单表访问

```
SELECT 。 。 。 FROM TC_ACCT... T  
WHERE  
REPORT_DATE BETWEEN TRUNC(:B4 ) - 3 AND TRUNC(:B4 ) + 1  
AND PROVINCE_CENTER_ID = :B3  
AND TERMINAL_ID = :B2 AND TID = :B1  
AND ACCOUNT_TRADE_TYPE_CODE <> 80
```



■ 执行计划突变

检查AWR历史记录中的执行时间和Plan_Hash_Value。发现有执行计划突变。

```
Select sample_time,sql_id,sql_plan_hash_value  
From dba_hist_active_sess_history where sql_id = '0z5z.....'  
Where sample_time between to_date('20170818-  
22','YYYYMMDD-HH24')  
and to_date('20170819-10','YYYYMMDD-HH24')  
Order by sample_time
```



执行计划突变

		SAMPLE_TIME		SQL_ID	SQL_PLAN_HASH_VALUE
	1	18-8月 -17 10.40.01.325 下午	...	0z5z	20469
	2	18-8月 -17 11.12.12.120 下午	...	0z5z	20469
▶	3	19-8月 -17 04.32.18.758 上午	...	0z5z	4259
	4	19-8月 -17 04.32.28.858 上午	...	0z5z	4259
	5	19-8月 -17 04.32.49.089 上午	...	0z5z	4259
	6	19-8月 -17 04.32.59.189 上午	...	0z5z	4259
	7	19-8月 -17 04.33.29.510 上午	...	0z5z	4259
	8	19-8月 -17 04.33.59.840 上午	...	0z5z	4259
	9	19-8月 -17 04.34.30.160 上午	...	0z5z	4259
	10	19-8月 -17 04.35.00.500 上午	...	0z5z	4259
	11	19-8月 -17 04.35.30.820 上午	...	0z5z	4259
	12	19-8月 -17 04.36.01.150 上午	...	0z5z	4259
	13	19-8月 -17 04.36.41.580 上午	...	0z5z	4259
	14	19-8月 -17 04.37.11.911 上午	...	0z5z	4259
	15	19-8月 -17 04.38.32.771 上午	...	0z5z	4259
	16	19-8月 -17 04.39.03.112 上午	...	0z5z	4259
	17	19-8月 -17 04.40.13.862 上午	...	0z5z	4259
	18	19-8月 -17 04.40.44.182 上午	...	0z5z	4259



之前的执行计划，很快

Stat Name	Statement Total	Per Execution	% Snap Total
Elapsed Time (ms)	1,515,433	6.20	0.22
CPU Time (ms)	277,340	1.13	0.24
Executions	244,577		
Buffer Gets	9,289,151	37.98	0.34
Disk Reads	160,137	0.65	1.67
Parse Calls	2,443	0.01	0.01
Rows	244,577	1.00	
User I/O Wait Time (ms)	1,230,484		
Cluster Wait Time (ms)	31,356		
Application Wait Time (ms)	0		
Concurrency Wait Time (ms)	42		
Invalidations	0		
Version Count	138		
Sharable Mem(KB)	1,192		

6.20
1.13
37.98

[Back to Plan 2\(PHV: 2046950722\)](#)

[Back to Top](#)

Execution Plan

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time	Pstart	Pstop
0	SELECT STATEMENT				71 (100)			
1	FILTER							
2	PARTITION RANGE ITERATOR		1	78	71 (91)	00:00:01	KEY	KEY
3	TABLE ACCESS BY LOCAL INDEX ROWID	TC_ACCOUNT	1	78	71 (91)	00:00:01	KEY	KEY
4	INDEX SKIP SCAN	PK_TC_ACCOUNT	1		70 (92)	00:00:01	KEY	KEY

出问题的执行计划，很慢

Stat Name	Statement Total	Per Execution	% Snap Total
Elapsed Time (ms)	36,395,675	20,481.53	5.18
CPU Time (ms)	21,971,910	12,364.61	19.13
Executions	1,777		
Buffer Gets	391,047,037	220,060.23	14.43
Disk Reads	63	0.04	0.00
Parse Calls	478	0.27	0.00
Rows	1,760	0.99	
User I/O Wait Time (ms)	1,677		
Cluster Wait Time (ms)	1		
Application Wait Time (ms)	0		
Concurrency Wait Time (ms)	2,443,489		
Invalidations	0		
Version Count	24		
Sharable Mem(KB)	123		

Per Execution	
	20,481.53
	12,364.61
	220,060.23

[Back to Plan 1\(PHV: 425993148\)](#)

[Back to Top](#)

Execution Plan

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time	Pstart	Pstop
0	SELECT STATEMENT				5 (100)			
1	FILTER							
2	PARTITION RANGE ITERATOR		1	79	5 (0)	00:00:01	KEY	KEY
3	TABLE ACCESS BY LOCAL INDEX ROWID	TC_ACCOUNT	1	79	5 (0)	00:00:01	KEY	KEY
4	INDEX RANGE SCAN	PK_TC_ACCOUNT	1		4 (0)	00:00:01	KEY	KEY

两者的区别

Execution Plan

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time	Pstart	Pstop
0	SELECT STATEMENT				71 (100)			
1	FILTER							
2	PARTITION RANGE ITERATOR		1	78	71 (91)	00:00:01	KEY	KEY
3	TABLE ACCESS BY LOCAL INDEX ROWID	TC_ACCOUNT	1	78	71 (91)	00:00:01	KEY	KEY
4	INDEX SKIP SCAN	PK_TC_ACCOUNT	1		70 (92)	00:00:01	KEY	KEY

Execution Plan

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time	Pstart	Pstop
0	SELECT STATEMENT				5 (100)			
1	FILTER							
2	PARTITION RANGE ITERATOR		1	79	5 (0)	00:00:01	KEY	KEY
3	TABLE ACCESS BY LOCAL INDEX ROWID	TC_ACCOUNT	1	79	5 (0)	00:00:01	KEY	KEY
4	INDEX RANGE SCAN	PK_TC_ACCOUNT	1		4 (0)	00:00:01	KEY	KEY



两者的区别

第一步，先对比Cost

好的执行计划Cost=71, 不好的是5

一个Cost = 5 的执行计划实际执行20秒，显然是评估错误

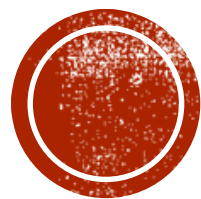
第二步，看执行计划异同之处

好的执行计划是index skip scan

不好的是index range scan

同一个索引





INDEX RANGE SCAN vs INDEX SKIP SCAN



第七屆



数据技术嘉年华
Data Technology Carnival



SKIP SCAN 为什么比 RANGE SCAN快？

索引是一个复合索引，正好包含了SQL中的5个条件
(REPORT_DATE, PROVINCE_CENTER_ID, TERMINAL_ID,
ACCOUNT_TRADE_TYPE_CODE, TID)

语句条件：

```
REPORT_DATE BETWEEN TRUNC(:B4 ) - 3 AND TRUNC(:B4 ) + 1  
AND PROVINCE_CENTER_ID = :B3  
AND TERMINAL_ID = :B2 AND TID = :B1  
AND ACCOUNT_TRADE_TYPE_CODE <> 80
```



表的栏位信息

REPORT_DATE - 分区条件，每天一个分区，保留90天数据，每分区大约1000万条记录，且是纯DATE,不包含time信息，唯一值 90个

PROVINCE_CENTER_ID -- 唯一值 14个

TERMINAL_ID -- 唯一值 4万

ACCOUNT_TRADE_TYPE_CODE - 唯一值3个

TID - 唯一值50万



第七届



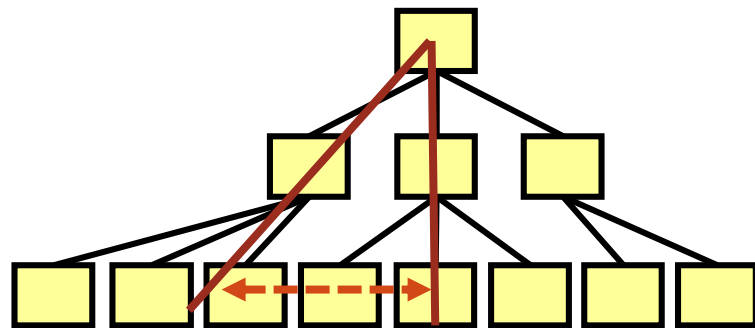
数据技术嘉年华
Data Technology Carnival



复合索引：RANGE SCAN止于区间条件

复合索引中，如果在前面的是区间条件（如 $>$ 、 $<$ 、between and），后面的条件无法使用ACCESS的方式直接获取对应索引块，RANGE SCAN形式使用到这个条件，然后把索引块取到内存，再进行FILTER操作。

ACCESS, 根据KEY定位Block, 效率很高
FILTER, 取出值到内存做计算, 效率较差



SKIP SCAN:遍历引导列

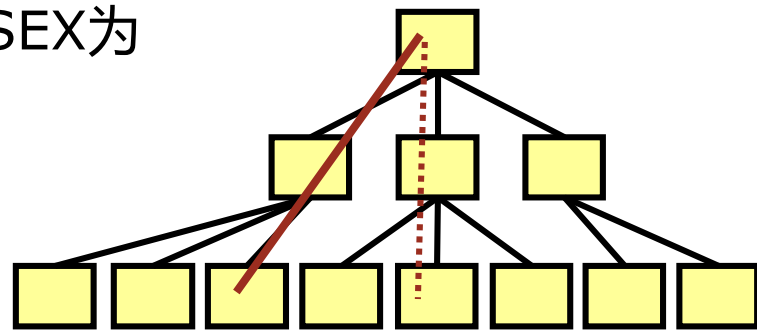
复合索引中，前导列不在条件中（或者因区间条件被舍弃），但前导列唯一值不多，可以走SKIP SCAN. 简单理解SKIP SCAN, 就是把不在条件中的前导列通过遍历补全。

Create index .. On person (sex,name); 其中SEX为
MALE/FEMALE

Select * from person where name='罗海雄' ;

Skip Scan 相当于：

Select * from person where name='罗海雄'
and SEX in ('MALE','FEMALE');



RANGE SCAN vs SKIP SCAN

```
REPORT_DATE BETWEEN TRUNC(:B4 ) - 3 AND TRUNC(:B4 ) + 1  
AND PROVINCE_CENTER_ID = :B3  
AND TERMINAL_ID = :B2 AND TID = :B1  
AND ACCOUNT_TRADE_TYPE_CODE <> 80
```

Range Scan取出3天约3000万数据对应的索引块，再进行Filter, 速度很慢

```
REPORT_DATE BETWEEN TRUNC(:B4 ) - 3 AND TRUNC(:B4 ) + 1  
AND PROVINCE_CENTER_ID = :B3  
AND TERMINAL_ID = :B2 AND TID = :B1  
AND ACCOUNT_TRADE_TYPE_CODE <> 80
```

SKIP遍历90个REPORT_DATE值，每次通过PROVINCE_CENTER_ID和TERMINAL_ID ACCESS 10条记录，相当于总共1000条记录，性能很好。



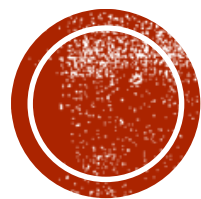
最简单的优化方式

语句条件：

```
REPORT_DATE BETWEEN TRUNC(:B4 ) - 3 AND TRUNC(:B4 ) + 1  
AND PROVINCE_CENTER_ID = :B3  
AND TERMINAL_ID = :B2 AND TID = :B1  
AND ACCOUNT_TRADE_TYPE_CODE <> 80
```

建立复合索引(**PROVINCE_CENTER_ID , TERMINAL_ID ,TID ,
REPORT_DATE**)





谓词越界与校正



第七届



数据技术嘉年华
Data Technology Carnival



为什么选择错误？

Column (#2): PROVINCE_CENTER_ID(NUMBER)

AvgLen: 3.00 NDV: 14 Nulls: 0 Density: 0.0058835 Min: 11 Max: 65

Histogram: Freq #Bkts: 14 UncompBkts: 5354 EndPtVals: 14

Column (#3): TERMINAL_ID(NUMBER)

AvgLen: 6.00 NDV: 39775 Nulls: 0 Density: 2.5141e-05 Min: 22 Max: 387650760

Column (#6): TID(NUMBER)

AvgLen: 6.00 NDV: 507757 Nulls: 0 Density: 1.9694e-06 Min: 100115 Max: 3330929

Column (#1): REPORT_DATE(DATE)

AvgLen: 8.00 NDV: 90 Nulls: 0 Density: 0.0036421 Min: 2457804 Max: 2457893

Histogram: Freq #Bkts: 90 UncompBkts: 5354 EndPtVals: 90

Using prorated density: 3.8321e-10 of col #1 as
selectivity of out-of-range value pred

Column (#5): ACCOUNT_TRADE_TYPE_CODE(NUMBER)

AvgLen: 3.00 NDV: 3 Nulls: 0 Density: 0.0085917 Min: 10 Max: 80

Histogram: Freq #Bkts: 3 UncompBkts: 5354 EndPtVals: 3



谓词越界

原来是谓词越界了，REPORT_DATE 超出了直方图记录的信息？

检查统计信息，发现REPORT_DATE直方图的确有问题。

```
select h.*,  
       to_date(to_char(endpoint_value, 'FM99999999') || '.' || to_char(mod(endpoint_value, 1), 'FM999999'), 'J.sssss')  
from user_histograms h  
where table_name =                   
       and column_name = 'REPORT_DATE';
```



谓词越界

COLUMN_NAME	ENDPOINT_NUMBER	ENDPOINT_VALUE	ENDPOINT_ACTUAL_VALUE	TO_DATE(TO_C
REPORT_DATE	3441	2457861		2017/4/17
REPORT_DATE	3509	2457862		2017/4/18
REPORT_DATE	3565	2457863		2017/4/19
REPORT_DATE	3616	2457864		2017/4/20
REPORT_DATE	3674	2457865		2017/4/21
REPORT_DATE	3746	2457866		2017/4/22
REPORT_DATE	3821	2457867		2017/4/23
REPORT_DATE	3891	2457868		2017/4/24
REPORT_DATE	3940	2457869		2017/4/25
REPORT_DATE	4004	2457870		2017/4/26
REPORT_DATE	4056	2457871		2017/4/27
REPORT_DATE	4108	2457872		2017/4/28
REPORT_DATE	4178	2457873		2017/4/29
REPORT_DATE	4231	2457874		2017/4/30
REPORT_DATE	4315	2457875		2017/5/1
REPORT_DATE	4365	2457876		2017/5/2
REPORT_DATE	4434	2457877		2017/5/3
REPORT_DATE	4488	2457878		2017/5/4
REPORT_DATE	4532	2457879		2017/5/5
REPORT_DATE	4600	2457880		2017/5/6
REPORT_DATE	4663	2457881		2017/5/7
REPORT_DATE	4716	2457882		2017/5/8
REPORT_DATE	4765	2457883		2017/5/9
REPORT_DATE	4844	2457884		2017/5/10
REPORT_DATE	4895	2457885		2017/5/11
REPORT_DATE	4954	2457886		2017/5/12
REPORT_DATE	5029	2457887		2017/5/13
REPORT_DATE	5089	2457888		2017/5/14
REPORT_DATE	5149	2457889		2017/5/15
REPORT_DATE	5198	2457890		2017/5/16
REPORT_DATE	5259	2457891		2017/5/17
REPORT_DATE	5309	2457892		2017/5/18

谓词越界

谓词越界了，但是，越得似乎有点多？
5/19的最大值，为何 8/18才出问题？



第七届



数据技术嘉年华
Data Technology Carnival



谓词越界的矫正

谓词越界：条件中的值超出了直方图的最大最小值

由于统计信息不可能实时更新，不能避免发生谓词越界，Oracle优化器针对谓词越界做了校正。

- Height Balanced直方图谓词外越界
- Frequency 直方图谓词外越界
- Frequency 直方图谓词内部未命中



谓词越界的矫正探索

- 10053 看 out-of-range selectivity

Using prorated density: 3.8321e-10 of col #1 as selectivity of **out-of-range** value pred

- 生成执行计划看rows(cards)

- 生成执行计划查询PLAN_TABLE里的CARDINALITY

Id	Operation	Name	Rows
0	SELECT STATEMENT		1
1	SORT AGGREGATE		1
* 2	TABLE ACCESS FULL	TEST_OOR	100



谓词越界的矫正

构建测试表：表内0-200的偶数，10000个值

```
Create Table Test_OOR(id number);
```

```
Insert into Test_OOR
```

```
select a.r from
```

```
(select (rownum-1)*2 r from dual
```

```
Connect by rownum <= 100) a ,
```

```
(select rownum from dual
```

```
Connect by rownum <= 100) b;
```



HEIGHT BALANCED直方图的矫正

Begin

```
Dbms_stats.gather_table_stats(user,'TEST_O  
OR', METHOD_OPT=>'for all columns size  
30');  
End;
```

Size 30, 小于distinct key, 因此生成Height Balance直方图



HEIGHT BALANCED直方图的矫正

```
delete from plan_table;  
begin  
for i in -300..500 loop  
execute immediate 'explain plan set statement_id="||i||  
"' for select count(*) from TEST_OOR where id = '||i|;  
end loop;  
end;  
/
```



HEIGHT BALANCED直方图的矫正

```
select statement_id, CARDINALITY  
from plan_table  
where object_name = 'TEST_OOR'  
order by to_number(statement_id);
```



第七届

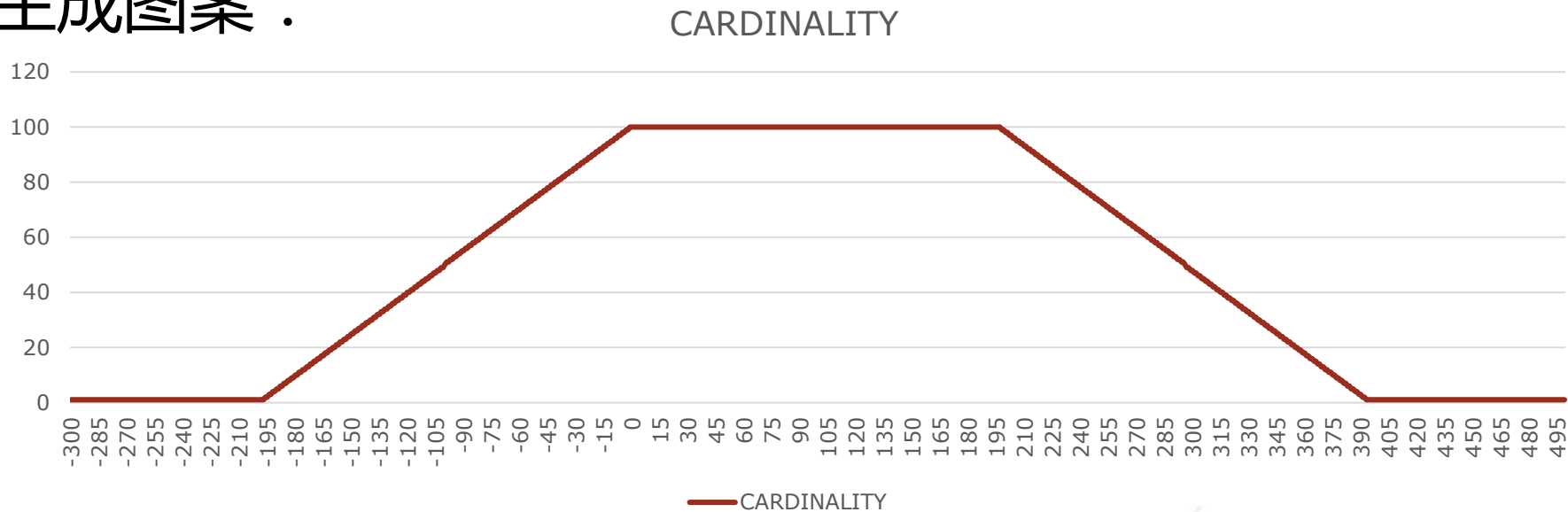


数据技术嘉年华
Data Technology Carnival



HEIGHT BALANCED直方图的矫正

生成图案：



第七届



数据技术嘉年华
Data Technology Carnival



FREQUENCY直方图的矫正

为了做图好看，减少表内的唯一值（30-50的偶数）

```
truncate table Test_OOR;
```

```
Insert into Test_OOR
```

```
select a.r from
```

```
(select (rownum+14)*2 r from dual
```

```
Connect by rownum <= 10) a ,(select
```

```
rownum from dual
```

```
Connect by rownum <= 100) b;
```



FREQUENCY直方图的矫正

Begin

```
Dbms_stats.gather_table_stats(user,'TEST_O  
OR', METHOD_OPT=>'for all columns size  
200');
```

End;

/



第七届

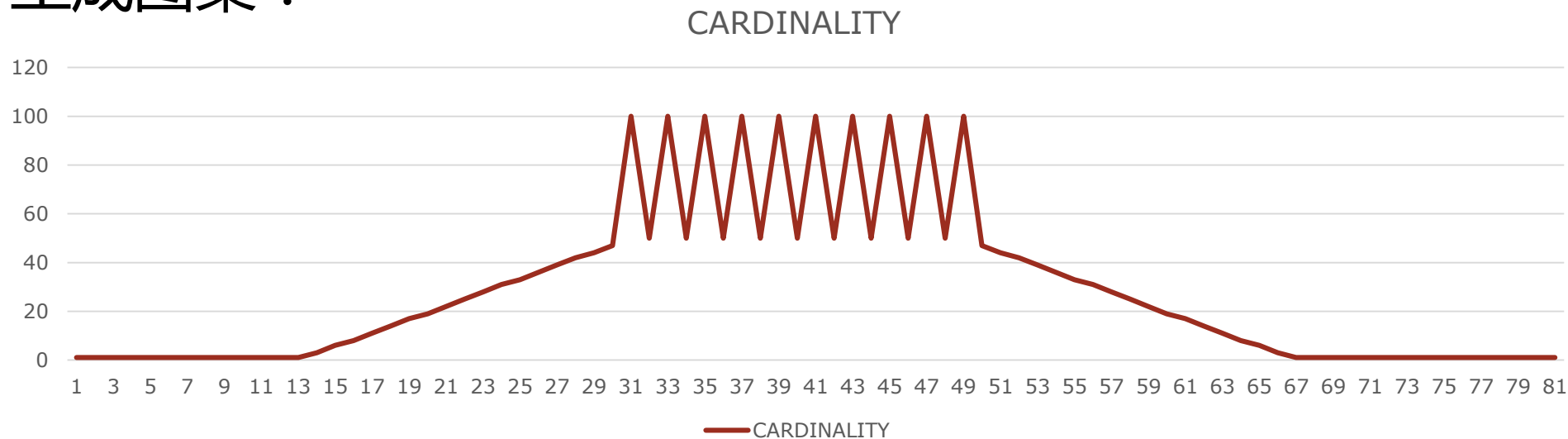


数据技术嘉年华
Data Technology Carnival



FREQUENCY直方图的矫正

生成图案：



第七届



数据技术嘉年华
Data Technology Carnival



直方图的矫正公式

确定非命中值的密度(density),为越界基准密度

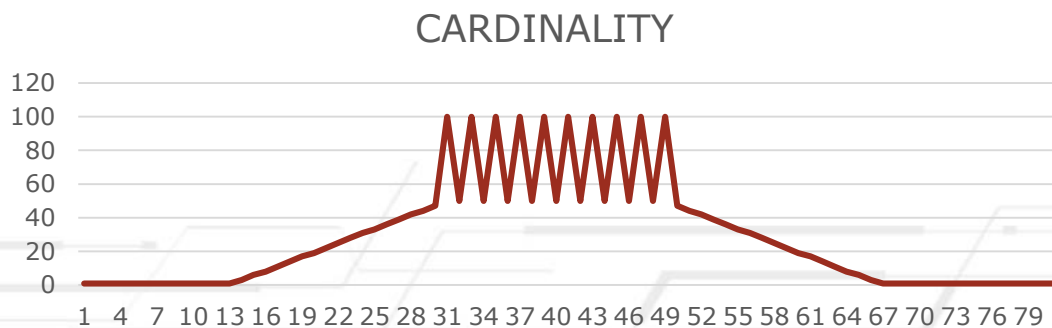
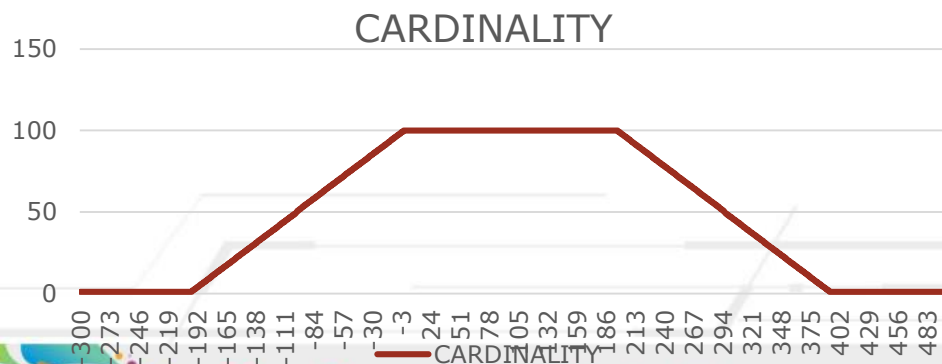
➤ Height Balanced直方图, 为非热值的密度:

$$\frac{(\text{BUCKETS_COUNT} - \text{POP_BUCKET_COUNT})}{\text{BUCKETS_COUNT} \times (\text{NUM_DISTINCT} - \text{POP_VALUE_COUNT})}$$

➤ Frequency直方图, 为最小频率值的密度的一半:

$$\frac{\text{MIN}(\text{FREQUENCY})}{\text{NUM_ROWS} \times 2}$$

➤ 基准密度也是介于最高最低之间且没有命中热值的密度



直方图的矫正公式

以基准选择度等比下降，最远程为（最高值-最低值），超出则为0.

➤ 小于最低值：

$\text{BASE_DENSITY} * \text{GREATEST}(0, 1 - (\text{MIN_VALUE} - \text{PRED_VALUE}) / (\text{MAX_VALUE} - \text{MIN_VALUE}))$

➤ 大于最高值

$\text{BASE_DENSITY} * \text{GREATEST}(0, 1 - (\text{PRED_VALUE} - \text{MAX_VALUE}) / (\text{MAX_VALUE} - \text{MIN_VALUE}))$

5/18的统计信息,最大-最小=90天，8/17正好到了极低值附近



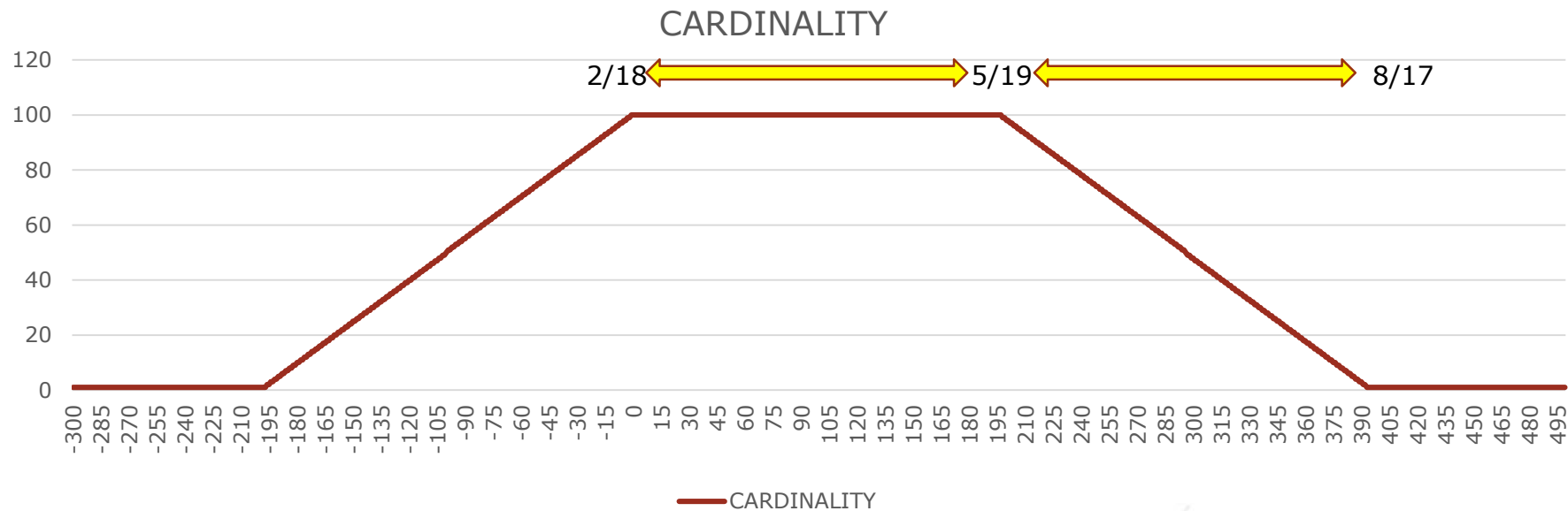
第七届



数据技术嘉年华
Data Technology Carnival



直方图的矫正公式

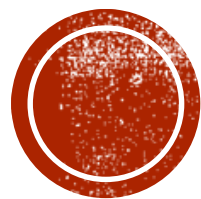


第七届



数据技术嘉年华
Data Technology Carnival





分区表统计信息收集策略



第七届



数据技术嘉年华
Data Technology Carnival



■ 另一个问题

统计信息都在收集，统计信息为何停留在 5/20？

COLUMN_NAME	LAST_ANALYZED
REPORT_DATE ...	2017/5/20 9:18:29 ▼



第七屆



数据技术嘉年华
Data Technology Carnival



另一个鲜为人知的技术

分区表收集策略：

- 全表收集- 太慢
- 单分区收集 -全表信息不更新？

Oracle 10.2.0.5 之前，默认情况下收集单分区统计信息不更新到全表。



第七屆



数据技术嘉年华
Data Technology Carnival



■ 另一个鲜为人知的技术

10.2.0.5及11g以后的新选项

granularity => 'APPROX_GLOBAL AND PARTITION'
把分区统计信息通过估算的形式更新到全表统计信息上。

```
dbms_stats.gather_table_stats  
(ownname      => i.owner,  
tabname        => i.table_name,  
partname       => i.partition_name,  
granularity    => 'APPROX_GLOBAL AND  
PARTITION',  
);
```



Q&A



第七届



数据技术嘉年华
Data Technology Carnival





云和恩墨
ENMOTECH

MAKE YOUR DATA DANCE.

一个分享交流的地方



微信号: eyygle



Long Press QR Code To
Identify The Concern

长按二维码识别关注



扫一扫，加入我们，分享更多知识



第七届



数据技术嘉年华
Data Technology Carnival





THANKS

