2015

Oracle技术嘉年华

Oracle Technology Carnival 2015

稳健•高效•云端 - 数据技术最佳实践

算法为王的 SQL优化

怀晓明

目 录

- Who am I？
- 应用困境
- 影响数据库性能的因素
- 从一个复杂的案例开始
- 一个看似简单的案例
- SQL优化的本质
- 以一个“简单”的案例结束
- 总结

Who am I?

- Lastwinner@itpub.net，资深数据库开发版主
- 《剑破冰山——Oracle开发艺术》、《DBA手记2》合著者
- 善于搜索的Troubleshooter

应用困境

- 现状
  - 随着社会对IT的需求不断扩展与扩张，IT应用呈爆发式增长
  - 大量的需求尤其是新需求，需要大量的开发者来完成
  - 于是，“码农”应运而生，而且是大量产生，这使得开发人员的总体水平越来越低
  - 然而，这是IT发展过程中的正常现象
  - 每个系统的数据量越来越大，随之而来的问题当然就是应用大量的性能问题

应用困境

- 大多数公司的解决办法
  - 升级硬件
  - 拓展架构

- 预想的实施结果
  - 解决性能问题

- 然而现实总是残酷的
  - 硬件升级太昂贵，架构拓展成本也不低
  - 即便钱不是问题，一年后呢？两年后呢？……
  - 老板不是傻瓜，无底洞的硬件投入，不可持续发展

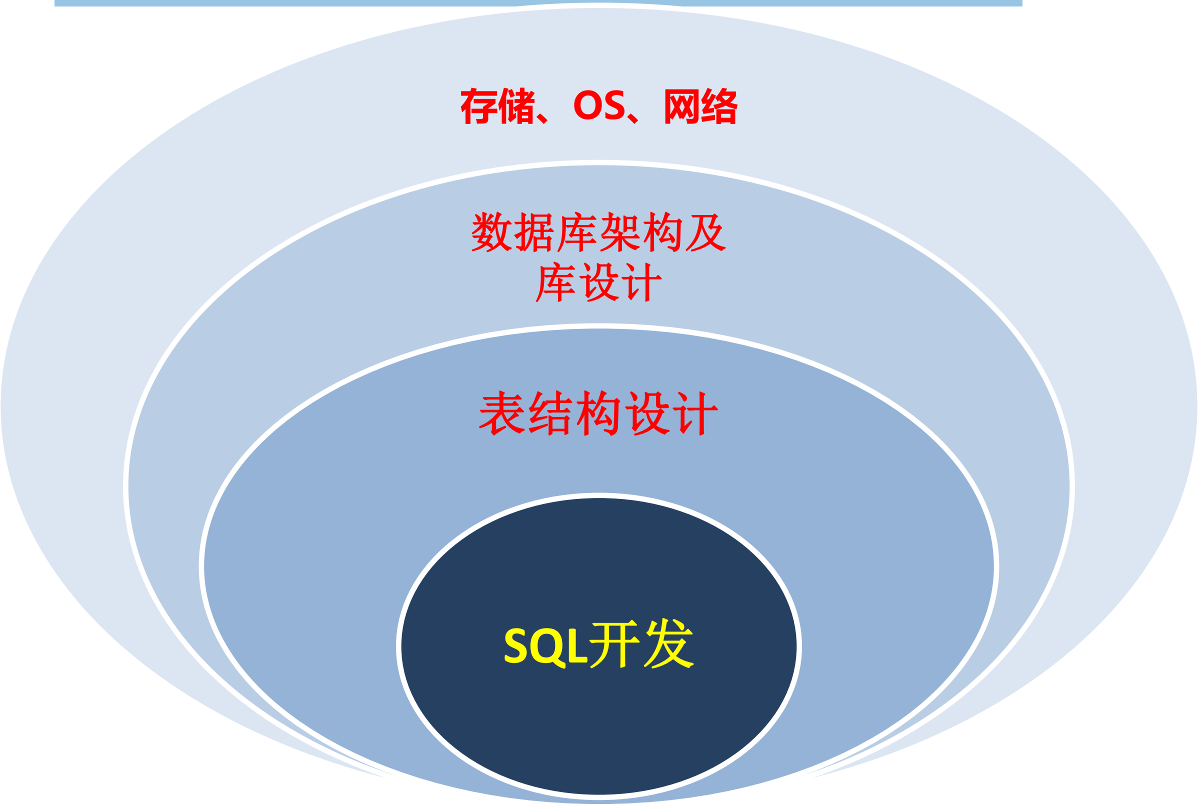
应用困境

- 如何真正的解决应用性能问题？
  - 战略：治病要除根
  - 战术：程序代码优化，尤其是SQL优化

影响数据库性能的因素

- 框架
- 表
- 索引
- 统计信息
- 执行计划
- SQL写法

影响数据库性能的因素—框架



影响数据库性能的因素—表

- 大表没分区
- 表没主键
- 外键无索引
- 主子表设计不合理
- ……

影响数据库性能的因素—索引

- 未建索引
- 索引无效
- 索引不可见
- 索引不合理
- 索引过于冗余
- ……

影响数据库性能的因素—统计信息

- 缺失
- 过期
- 不准
- ……

影响数据库性能的因素—执行计划

- 执行计划改变
- 执行计划不稳定（有多个）
- 执行计划非最佳
  - stats
  - CBO
  - hint
  - ……
- 执行计划错误
- ……

影响数据库性能的因素—SQL写法

- 未使用绑定变量
- 隐式转换
- SQL写得不够好
- SQL写得不太好
- SQL写得比较差
- SQL写得很差劲
- SQL写得很糟糕
- SQL写得……

影响数据库性能的因素—SQL写法

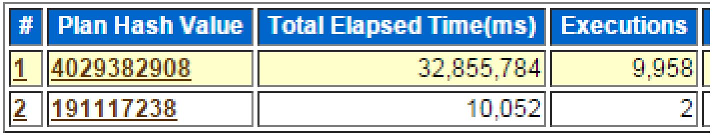
- 写得不好的SQL如何优化？
- 当然是靠改写！

从一个复杂的案例开始—SQL 文本

```
select sii.P_Name processName, sii.PID processId, sii.VID versionId, (nvl(max(t.ciid),
0)+nvl(max(queueMember.ciid), 0)+nvl(max(s.ciid), 0)+nvl(max(abs.ciid), 0)) totalTask from (select SI.*,
IP.P_Name as P_Name from (select Current_Process_Id as PID, Current_Version_Id as VID, Step_ID as SID
from step_info where step_status in (1, 6)) SI left join Installed_Process IP on SI.PID = IP.Process_ID and
SI.VID = IP.Version_ID ) sii left join (select Current_Process_Id, Current_Version_Id, count(v2.Incident_Id)
ciid from step_info v2 where getTaskPerformer(Belong_to, Assign_to, Escalation_to)=7785 and
Current_Process_Id>0 and Current_Version_Id>0 AND step_status in (1, 6) group by Current_Process_Id,
Current_Version_Id) t on sii.pid=t.Current_Process_Id and sii.vid=t.Current_Version_Id left join (select
Current_Process_Id, Current_Version_Id, count(v2.Current_Process_Id) ciid from step_info v2,
Queue_Member q where (v2.TaskFlag ='Q' and v2.urlid=q.urlid and q.User_Num= 7785) and QPICKUP <>
'Y' and Current_Process_Id>0 and Current_Version_Id>0 AND step_status in (1, 6) group by
Current_Process_Id, Current_Version_Id) queueMember on sii.pid=queueMember.Current_Process_Id
and sii.vid=queueMember.Current_Version_Id left join (select Current_Process_Id, Current_Version_Id,
count(v2.Incident_Id) ciid from step_info v2 where step_status=4 and getTaskPerformer(belong_to,
assign_to, escalation_to)=7785 and Current_Process_Id>0 and Current_Version_Id>0 group by Curre
nt_Process_Id, Current_Version_Id) s on sii.pid=s.Current_Process_Id and sii.vid=s.Current_Version_Id
left join (select Current_Process_Id, Current_Version_Id, count(v2.Incident_Id) ciid from step_info v2
where assign_to > 0 and belong_to =7785and belong_to= assign_by and escalation_to is null and
Current_Process_Id>0 and Current_Version_Id>0 AND step_status in (1, 6) group by Current_Process_Id,
Current_Version_Id) abs on sii.pid=abs.Current_Process_Id and sii.vid=abs.Current_Version_Id where
(t.ciid is not null or queueMember.ciid is not null or s.ciid is not null or abs.ciid is not null) group by sii.pid,
sii.vid, sii.P_Name order by nlssort(sii.P_Name, 'NLS_SORT=SCHINESE_PINYIN_M')
```

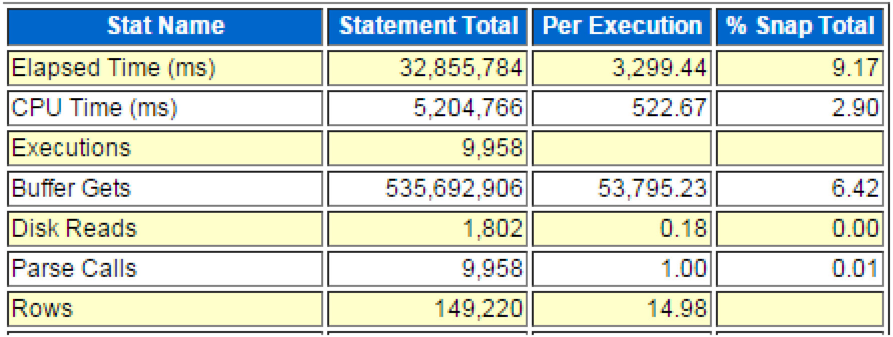
从一个复杂的案例开始—相关信息

- 多个执行计划

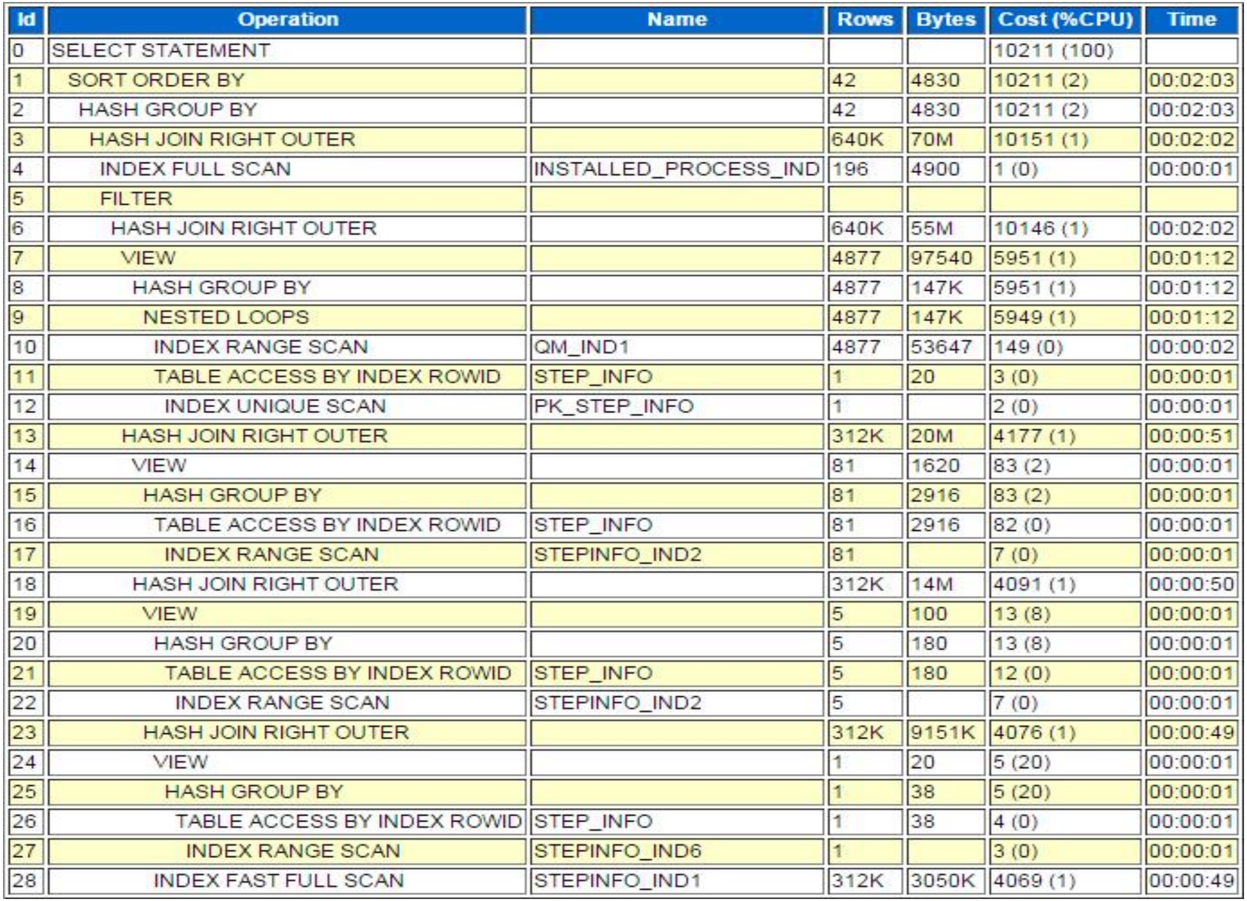


#	Plan Hash Value	Total Elapsed Time(ms)	Executions
1	4029382908	32,855,784	9,958
2	191117238	10,052	2

- 执行计划信息

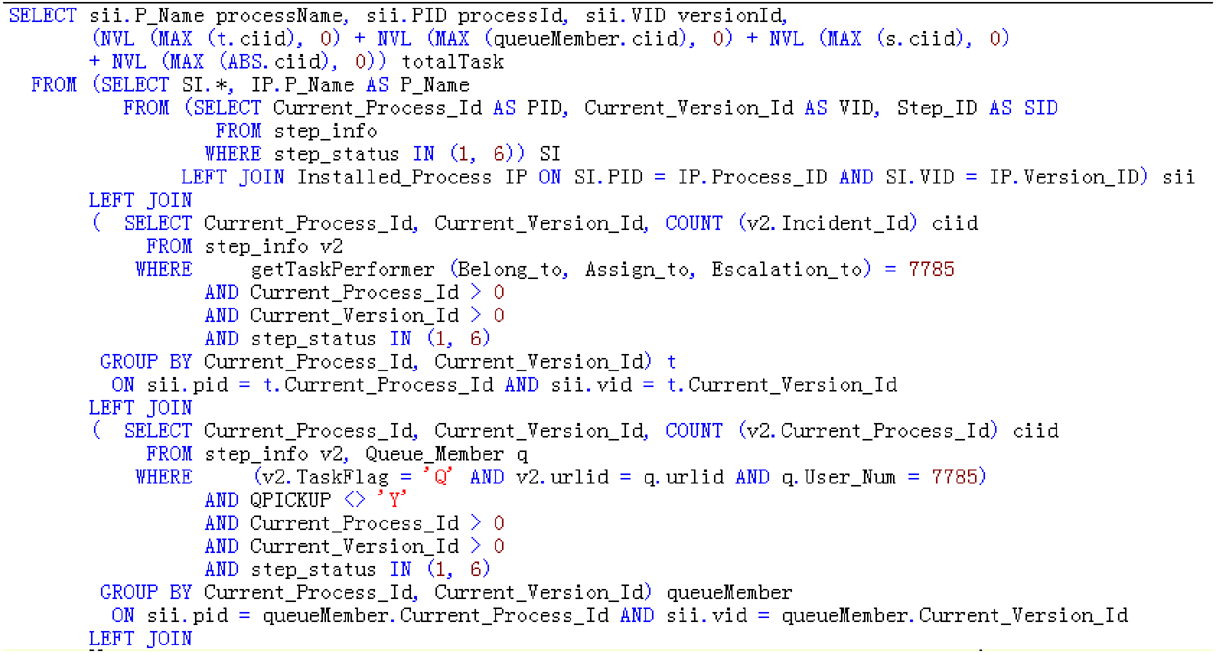


Stat Name	Statement Total	Per Execution	% Snap Total
Elapsed Time (ms)	32,855,784	3,299.44	9.17
CPU Time (ms)	5,204,766	522.67	2.90
Executions	9,958		
Buffer Gets	535,692,906	53,795.23	6.42
Disk Reads	1,802	0.18	0.00
Parse Calls	9,958	1.00	0.01
Rows	149,220	14.98	



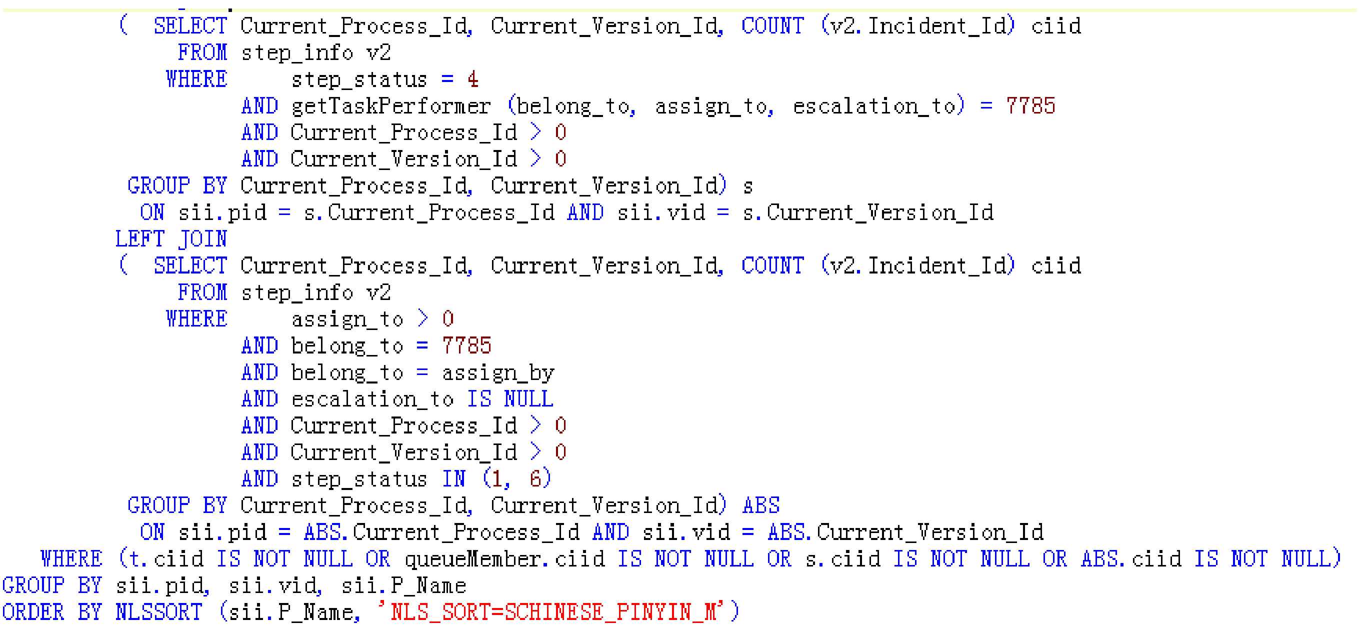
Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT				10211 (100)	
1	SORT ORDER BY		42	4830	10211 (2)	00:02:03
2	HASH GROUP BY		42	4830	10211 (2)	00:02:03
3	HASH JOIN RIGHT OUTER		640K	70M	10151 (1)	00:02:02
4	INDEX FULL SCAN	INSTALLED_PROCESS_IND	196	4900	1 (0)	00:00:01
5	FILTER					
6	HASH JOIN RIGHT OUTER		640K	55M	10146 (1)	00:02:02
7	VIEW		4877	97540	5951 (1)	00:01:12
8	HASH GROUP BY		4877	147K	5951 (1)	00:01:12
9	NESTED LOOPS		4877	147K	5949 (1)	00:01:12
10	INDEX RANGE SCAN	QM_IND1	4877	53647	149 (0)	00:00:02
11	TABLE ACCESS BY INDEX ROWID	STEP_INFO	1	20	3 (0)	00:00:01
12	INDEX UNIQUE SCAN	PK_STEP_INFO	1		2 (0)	00:00:01
13	HASH JOIN RIGHT OUTER		312K	20M	4177 (1)	00:00:51
14	VIEW		81	1620	83 (2)	00:00:01
15	HASH GROUP BY		81	2916	83 (2)	00:00:01
16	TABLE ACCESS BY INDEX ROWID	STEP_INFO	81	2916	82 (0)	00:00:01
17	INDEX RANGE SCAN	STEPINFO_IND2	81		7 (0)	00:00:01
18	HASH JOIN RIGHT OUTER		312K	14M	4091 (1)	00:00:50
19	VIEW		5	100	13 (8)	00:00:01
20	HASH GROUP BY		5	180	13 (8)	00:00:01
21	TABLE ACCESS BY INDEX ROWID	STEP_INFO	5	180	12 (0)	00:00:01
22	INDEX RANGE SCAN	STEPINFO_IND2	5		7 (0)	00:00:01
23	HASH JOIN RIGHT OUTER		312K	9151K	4076 (1)	00:00:49
24	VIEW		1	20	5 (20)	00:00:01
25	HASH GROUP BY		1	38	5 (20)	00:00:01
26	TABLE ACCESS BY INDEX ROWID	STEP_INFO	1	38	4 (0)	00:00:01
27	INDEX RANGE SCAN	STEPINFO_IND6	1		3 (0)	00:00:01
28	INDEX FAST FULL SCAN	STEPINFO_IND1	312K	3050K	4069 (1)	00:00:49

从一个复杂的案例开始—格式化



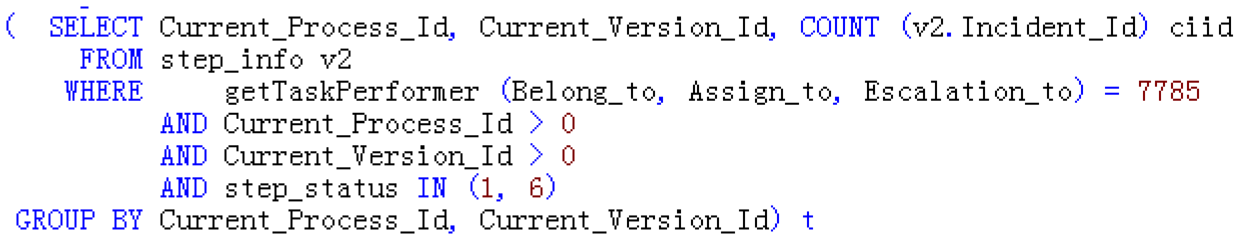
```
SELECT sii.P_Name processName, sii.PID processId, sii.VID versionId,
       (NVL (MAX (t.ciid), 0) + NVL (MAX (queueMember.ciid), 0) + NVL (MAX (s.ciid), 0)
       + NVL (MAX (ABS.ciid), 0)) totalTask
  FROM (SELECT SI.*, IP.P_Name AS P_Name
          FROM (SELECT Current_Process_Id AS PID, Current_Version_Id AS VID, Step_ID AS SID
                  FROM step_info
                 WHERE step_status IN (1, 6)) SI
               LEFT JOIN Installed_Process IP ON SI.PID = IP.Process_ID AND SI.VID = IP.Version_ID) sii
       LEFT JOIN
       (  SELECT Current_Process_Id, Current_Version_Id, COUNT (v2.Incident_Id) ciid
            FROM step_info v2
           WHERE     getTaskPerformer (Belong_to, Assign_to, Escalation_to) = 7785
                 AND Current_Process_Id > 0
                 AND Current_Version_Id > 0
                 AND step_status IN (1, 6)
        GROUP BY Current_Process_Id, Current_Version_Id) t
         ON sii.pid = t.Current_Process_Id AND sii.vid = t.Current_Version_Id
       LEFT JOIN
       (  SELECT Current_Process_Id, Current_Version_Id, COUNT (v2.Current_Process_Id) ciid
            FROM step_info v2, Queue_Member q
           WHERE     (v2.TaskFlag = ’Q’ AND v2.urlid = q.urlid AND q.User_Num = 7785)
                 AND QPICKUP <> ’Y’
                 AND Current_Process_Id > 0
                 AND Current_Version_Id > 0
                 AND step_status IN (1, 6)
        GROUP BY Current_Process_Id, Current_Version_Id) queueMember
         ON sii.pid = queueMember.Current_Process_Id AND sii.vid = queueMember.Current_Version_Id
       LEFT JOIN
```

从一个复杂的案例开始—格式化

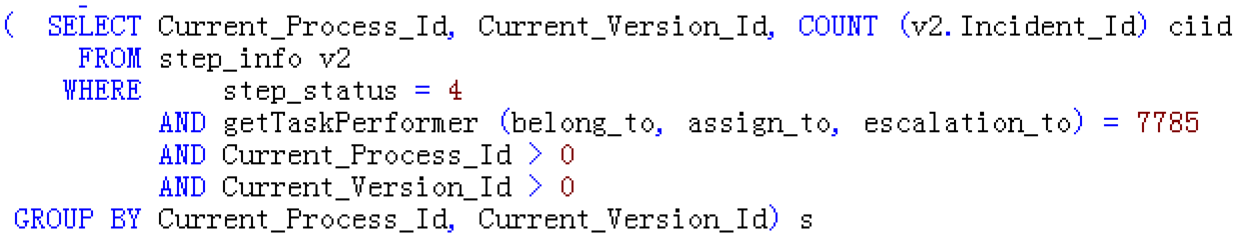


```
         (  SELECT Current_Process_Id, Current_Version_Id, COUNT (v2.Incident_Id) ciid
              FROM step_info v2
             WHERE     step_status = 4
                   AND getTaskPerformer (belong_to, assign_to, escalation_to) = 7785
                   AND Current_Process_Id > 0
                   AND Current_Version_Id > 0
          GROUP BY Current_Process_Id, Current_Version_Id) s
           ON sii.pid = s.Current_Process_Id AND sii.vid = s.Current_Version_Id
         LEFT JOIN
         (  SELECT Current_Process_Id, Current_Version_Id, COUNT (v2.Incident_Id) ciid
              FROM step_info v2
             WHERE     assign_to > 0
                   AND belong_to = 7785
                   AND belong_to = assign_by
                   AND escalation_to IS NULL
                   AND Current_Process_Id > 0
                   AND Current_Version_Id > 0
                   AND step_status IN (1, 6)
          GROUP BY Current_Process_Id, Current_Version_Id) ABS
           ON sii.pid = ABS.Current_Process_Id AND sii.vid = ABS.Current_Version_Id
   WHERE (t.ciid IS NOT NULL OR queueMember.ciid IS NOT NULL OR s.ciid IS NOT NULL OR ABS.ciid IS NOT NULL)
GROUP BY sii.pid, sii.vid, sii.P_Name
ORDER BY NLSSORT (sii.P_Name, ’NLS_SORT=SCHINESE_PINYIN_M’)
```

从一个复杂的案例开始—分析

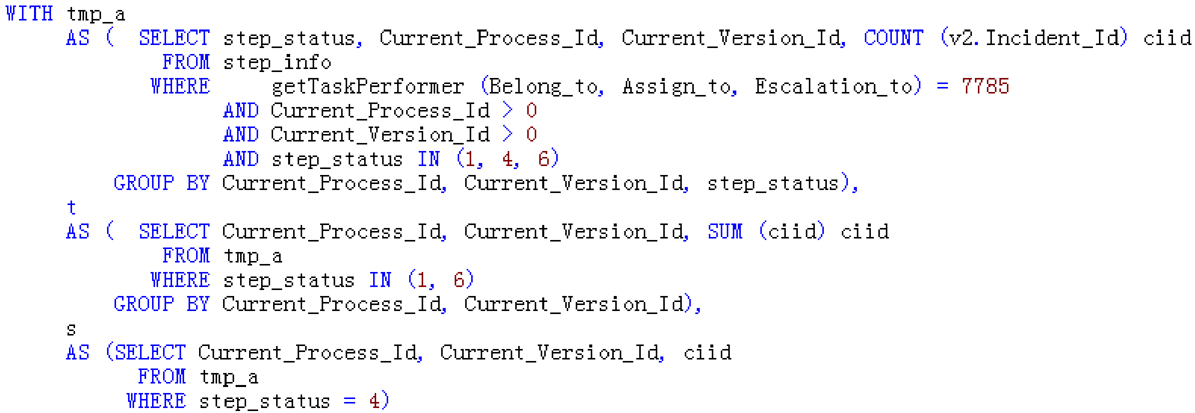


```
(  SELECT Current_Process_Id, Current_Version_Id, COUNT (v2.Incident_Id) ciid
     FROM step_info v2
    WHERE     getTaskPerformer (Belong_to, Assign_to, Escalation_to) = 7785
          AND Current_Process_Id > 0
          AND Current_Version_Id > 0
          AND step_status IN (1, 6)
 GROUP BY Current_Process_Id, Current_Version_Id) t
```



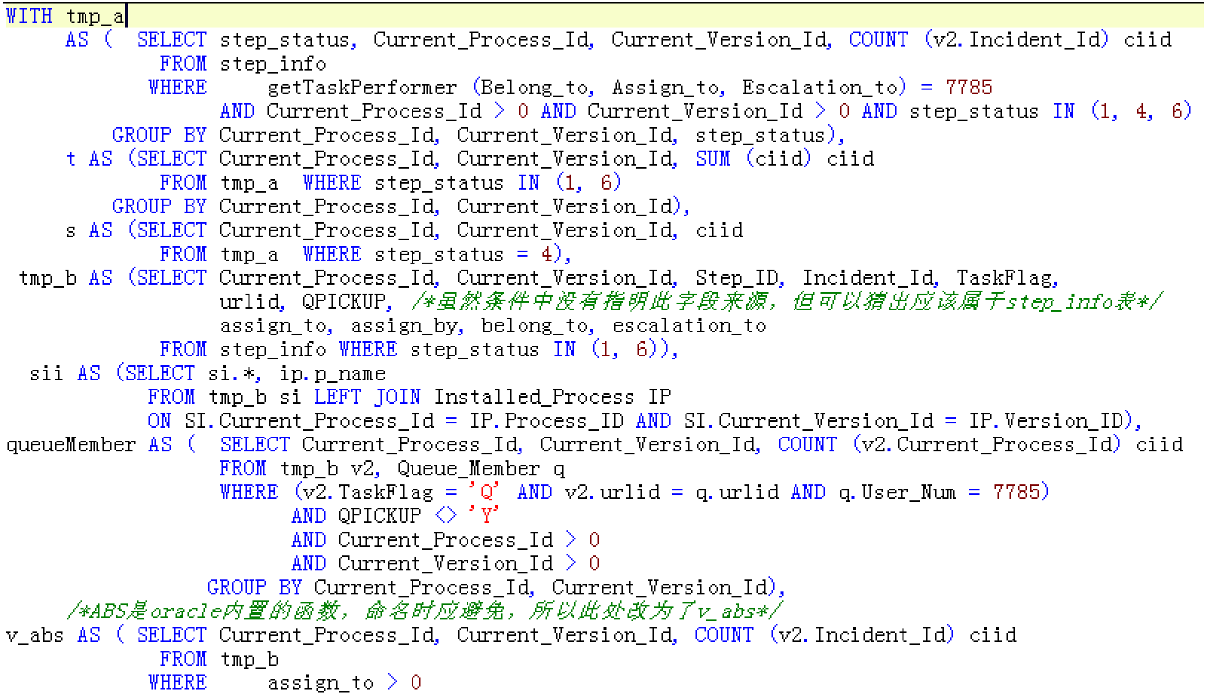
```
(  SELECT Current_Process_Id, Current_Version_Id, COUNT (v2.Incident_Id) ciid
     FROM step_info v2
    WHERE     step_status = 4
          AND getTaskPerformer (belong_to, assign_to, escalation_to) = 7785
          AND Current_Process_Id > 0
          AND Current_Version_Id > 0
 GROUP BY Current_Process_Id, Current_Version_Id) s
```

从一个复杂的案例开始—查询合并



```
WITH tmp_a
     AS (  SELECT step_status, Current_Process_Id, Current_Version_Id, COUNT (v2.Incident_Id) ciid
             FROM step_info
            WHERE     getTaskPerformer (Belong_to, Assign_to, Escalation_to) = 7785
                  AND Current_Process_Id > 0
                  AND Current_Version_Id > 0
                  AND step_status IN (1, 4, 6)
         GROUP BY Current_Process_Id, Current_Version_Id, step_status),
     t
     AS (  SELECT Current_Process_Id, Current_Version_Id, SUM (ciid) ciid
             FROM tmp_a
            WHERE step_status IN (1, 6)
         GROUP BY Current_Process_Id, Current_Version_Id),
     s
     AS (SELECT Current_Process_Id, Current_Version_Id, ciid
           FROM tmp_a
          WHERE step_status = 4)
```

从一个复杂的案例开始—查询合并

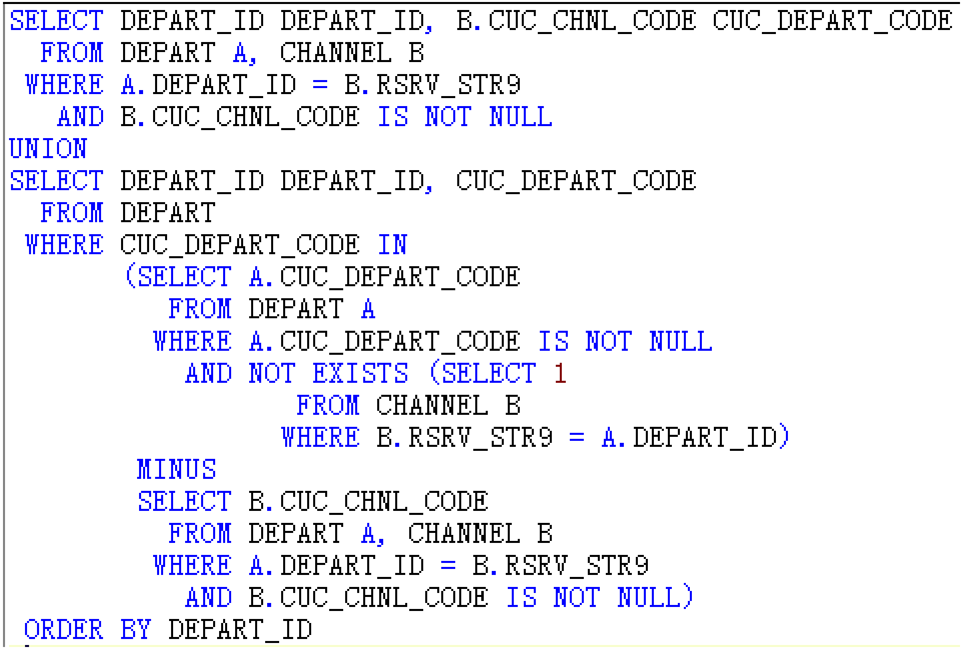


```
WITH tmp_a
     AS (  SELECT step_status, Current_Process_Id, Current_Version_Id, COUNT (v2.Incident_Id) ciid
             FROM step_info
            WHERE     getTaskPerformer (Belong_to, Assign_to, Escalation_to) = 7785
                  AND Current_Process_Id > 0 AND Current_Version_Id > 0 AND step_status IN (1, 4, 6)
         GROUP BY Current_Process_Id, Current_Version_Id, step_status),
     t AS (SELECT Current_Process_Id, Current_Version_Id, SUM (ciid) ciid
             FROM tmp_a  WHERE step_status IN (1, 6)
         GROUP BY Current_Process_Id, Current_Version_Id),
     s AS (SELECT Current_Process_Id, Current_Version_Id, ciid
             FROM tmp_a  WHERE step_status = 4),
 tmp_b AS (SELECT Current_Process_Id, Current_Version_Id, Step_ID, Incident_Id, TaskFlag,
                  urlid, QPICKUP, /*虽然条件中没有指明此字段来源，但可以猜出应该属于step_info表*/
                  assign_to, assign_by, belong_to, escalation_to
             FROM step_info WHERE step_status IN (1, 6)),
  sii AS (SELECT si.*, ip.p_name
            FROM tmp_b si LEFT JOIN Installed_Process IP
            ON SI.Current_Process_Id = IP.Process_ID AND SI.Current_Version_Id = IP.Version_ID),
queueMember AS (  SELECT Current_Process_Id, Current_Version_Id, COUNT (v2.Current_Process_Id) ciid
                  FROM tmp_b v2, Queue_Member q
                  WHERE (v2.TaskFlag = 'Q' AND v2.urlid = q.urlid AND q.User_Num = 7785)
                        AND QPICKUP <> 'Y'
                        AND Current_Process_Id > 0
                        AND Current_Version_Id > 0
                 GROUP BY Current_Process_Id, Current_Version_Id),
     /*ABS是oracle内置的函数，命名时应避免，所以此处改为了v_abs*/
v_abs AS ( SELECT Current_Process_Id, Current_Version_Id, COUNT (v2.Incident_Id) ciid
             FROM tmp_b
            WHERE     assign_to > 0
```

从一个复杂的案例开始—小结

- 其实，单纯从SQL文本上看，就能做出如上改写
- 本次改写涉及到：
  - 查询合并
  - 查询结果分发（行列转换）

一个看似简单的案例—SQL文本



```
SELECT DEPART_ID DEPART_ID, B.CUC_CHNL_CODE CUC_DEPART_CODE
  FROM DEPART A, CHANNEL B
 WHERE A.DEPART_ID = B.RSRV_STR9
   AND B.CUC_CHNL_CODE IS NOT NULL
UNION
SELECT DEPART_ID DEPART_ID, CUC_DEPART_CODE
  FROM DEPART
 WHERE CUC_DEPART_CODE IN
       (SELECT A.CUC_DEPART_CODE
          FROM DEPART A
         WHERE A.CUC_DEPART_CODE IS NOT NULL
           AND NOT EXISTS (SELECT 1
                  FROM CHANNEL B
                 WHERE B.RSRV_STR9 = A.DEPART_ID)
        MINUS
        SELECT B.CUC_CHNL_CODE
          FROM DEPART A, CHANNEL B
         WHERE A.DEPART_ID = B.RSRV_STR9
           AND B.CUC_CHNL_CODE IS NOT NULL)
 ORDER BY DEPART_ID
```

一个看似简单的案例—相关信息

- 表所占空间

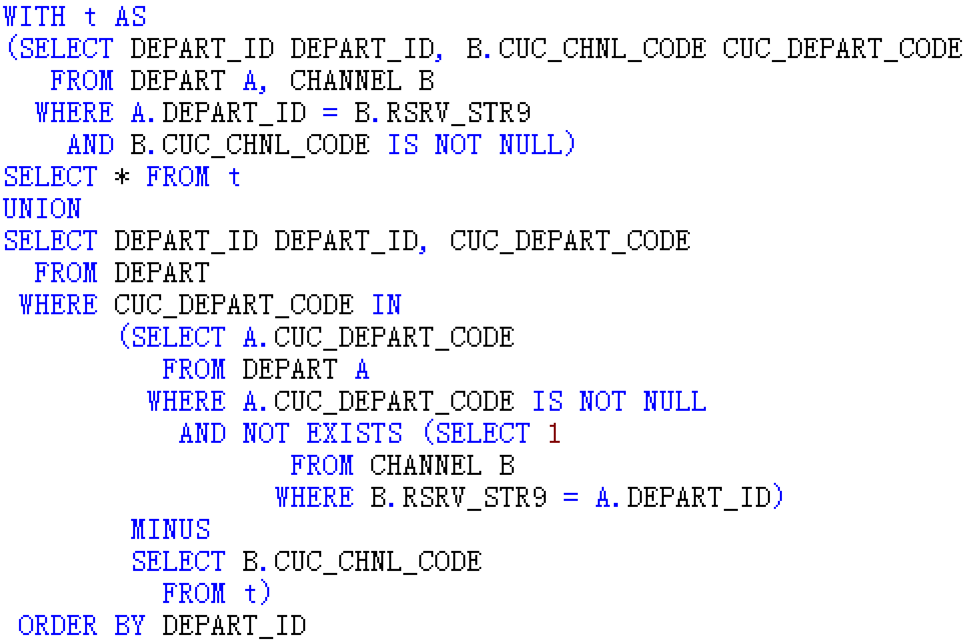
SEGMENT_NAME	MB
DEPART	20
CHANNEL	8

- 记录数以及据此计算出表所占block总大小

TABLE_NAME	NUM_ROWS	AVG_ROW_LEN	MB
DEPART	28810	154	4.2
CHANNEL	28390	237	6.4

- SQL问题
  - SQL在指定运行时间窗口内没执行完，而作为某业务关键路径上的SQL，其执行时间平常都没超过10秒
  - 如何优化？

一个看似简单的案例—改写方案

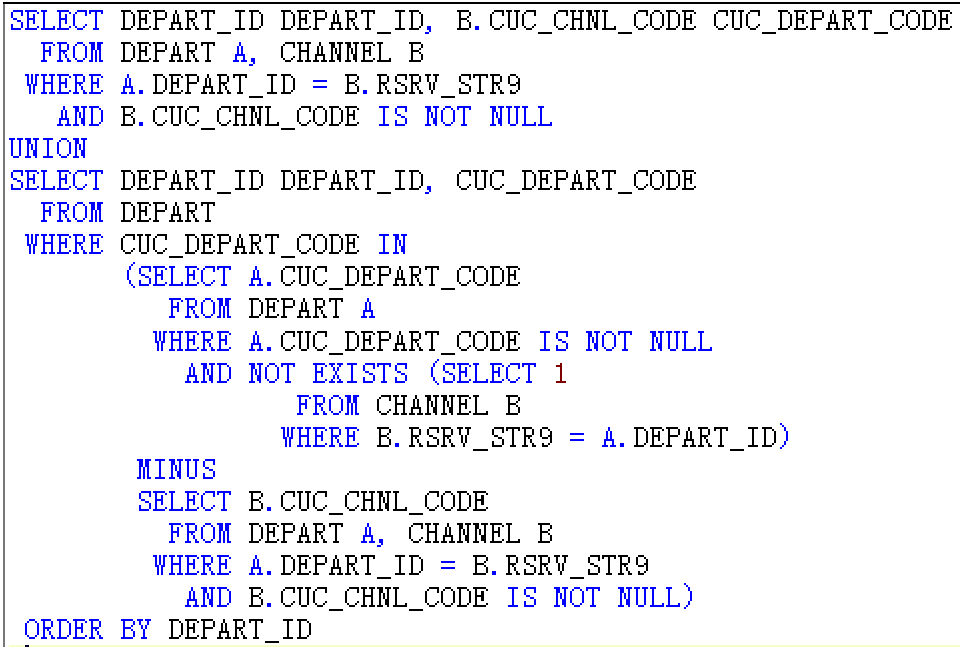


```
WITH t AS
(SELECT DEPART_ID DEPART_ID, B.CUC_CHNL_CODE CUC_DEPART_CODE
   FROM DEPART A, CHANNEL B
  WHERE A.DEPART_ID = B.RSRV_STR9
    AND B.CUC_CHNL_CODE IS NOT NULL)
SELECT * FROM t
UNION
SELECT DEPART_ID DEPART_ID, CUC_DEPART_CODE
  FROM DEPART
 WHERE CUC_DEPART_CODE IN
       (SELECT A.CUC_DEPART_CODE
          FROM DEPART A
         WHERE A.CUC_DEPART_CODE IS NOT NULL
           AND NOT EXISTS (SELECT 1
                  FROM CHANNEL B
                 WHERE B.RSRV_STR9 = A.DEPART_ID)
        MINUS
        SELECT B.CUC_CHNL_CODE
          FROM t)
 ORDER BY DEPART_ID
```

一个看似简单的案例—思路

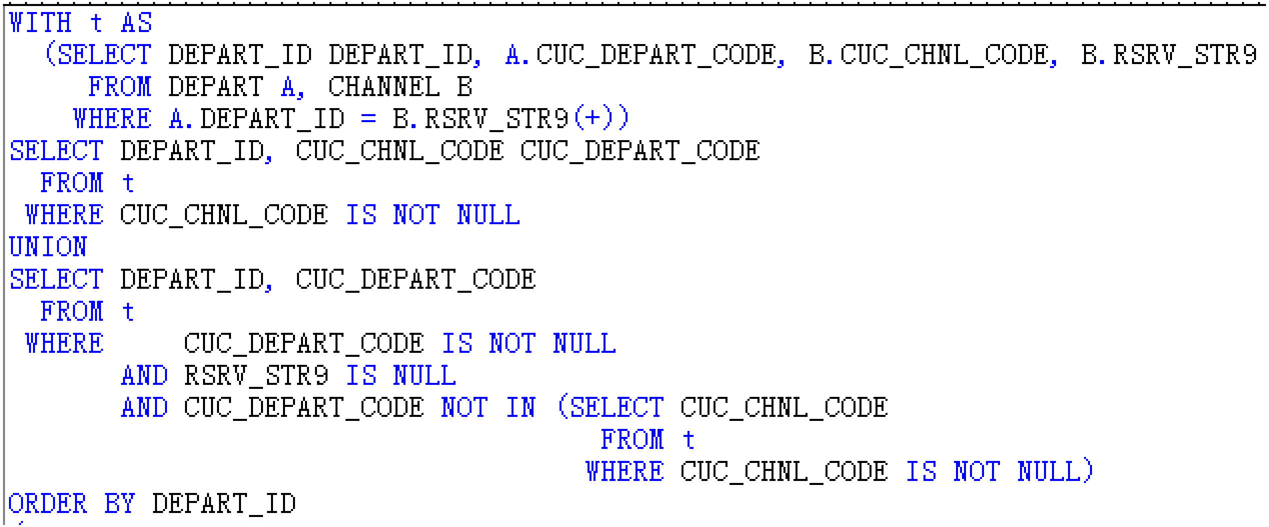
- 如何做出更好的改写?
- **读懂业务！算法为王！！**

一个看似简单的案例—SQL文本副本



```
SELECT DEPART_ID DEPART_ID,  B.CUC_CHNL_CODE CUC_DEPART_CODE
  FROM DEPART A,  CHANNEL B
 WHERE A.DEPART_ID = B.RSRV_STR9
   AND B.CUC_CHNL_CODE IS NOT NULL
UNION
SELECT DEPART_ID DEPART_ID,  CUC_DEPART_CODE
  FROM DEPART
 WHERE CUC_DEPART_CODE IN
       (SELECT A.CUC_DEPART_CODE
          FROM DEPART A
         WHERE A.CUC_DEPART_CODE IS NOT NULL
           AND NOT EXISTS (SELECT 1
                  FROM CHANNEL B
                 WHERE B.RSRV_STR9 = A.DEPART_ID)
        MINUS
        SELECT B.CUC_CHNL_CODE
          FROM DEPART A,  CHANNEL B
         WHERE A.DEPART_ID = B.RSRV_STR9
           AND B.CUC_CHNL_CODE IS NOT NULL)
 ORDER BY DEPART_ID
```

一个看似简单的案例—更好的改写方案



```sql
WITH t AS
  (SELECT DEPART_ID DEPART_ID, A.CUC_DEPART_CODE, B.CUC_CHNL_CODE, B.RSRV_STR9
     FROM DEPART A, CHANNEL B
    WHERE A.DEPART_ID = B.RSRV_STR9(+))
SELECT DEPART_ID, CUC_CHNL_CODE CUC_DEPART_CODE
  FROM t
 WHERE CUC_CHNL_CODE IS NOT NULL
UNION
SELECT DEPART_ID, CUC_DEPART_CODE
  FROM t
 WHERE     CUC_DEPART_CODE IS NOT NULL
       AND RSRV_STR9 IS NULL
       AND CUC_DEPART_CODE NOT IN (SELECT CUC_CHNL_CODE
                                     FROM t
                                    WHERE CUC_CHNL_CODE IS NOT NULL)
ORDER BY DEPART_ID
```

一个看似简单的案例—改写方案

- 优化结果
  - 如果你只能学一个SQL的优化技能，那么以下案例，你想学哪一个（均非缺索引）？
    - a. 一分钟到0.25秒
    - b. 9分钟到4.5分钟
    - c. 1小时18分到3秒
    - d. 400秒到40毫秒
    - e. 3.5秒到1秒以内（未有实际环境验证效率）

一个看似简单的案例—小结

- 查询合并
- 熟悉各种join的执行结果会是什么样
- 熟悉集合操作的结果会是什么样
- **读懂业务，算法为王！**

SQL优化的本质

- SQL优化，是数据库性能优化的重要一环
- 一般的，会将数据库性能优化当作就是SQL优化
- SQL优化要做哪些工作？
  - 包括SQL改写在内的SQL层面的优化，60%以上
  - 表和索引等对象的优化，25%左右
  - 模型优化（如主子表），7%左右
  - 其他，8%左右

- SQL优化更偏向业务层面，只有对业务理解透彻，才能做好SQL优化

SQL优化的本质

- 按目标定义，什么是SQL优化的本质？
  - **片面**：SQL运行得越快越好
  - **深刻**：平衡系统内的资源使用，充分发挥硬件资源，让大多数SQL运行得越快越好
  - **全面**：平衡SQL优化服务成本与SQL优化效果期望，做到广义上的资源平衡，达到最优的性能目标

SQL优化的本质

- SQL优化对企业的意义？
  - 从根源上提升系统效率，降低硬件设备损耗，延长硬件寿命
  - 延缓企业为解决性能问题而采购新设备的周期，等价于为企业省钱
  - 同样的硬件资源可以支撑更多的应用系统，等价于为企业省钱

- 成功案例：**某客户系统，经过我方优化，虚拟化环境下的两节点的RAC，CPU从每节点12颗直接降低到每节点8颗，大大节约了宝贵的硬件资源。**
  而这部分节省出的资源，客户又将其划分给其他系统使用，无异于节省了昂贵的硬件成本。

SQL优化的本质

- 如何找到能提供优秀的SQL优化的服务团队？

云和恩墨

ENMOTECH

数据驱动 成就未来

Make Your Data Dance

Who am I？

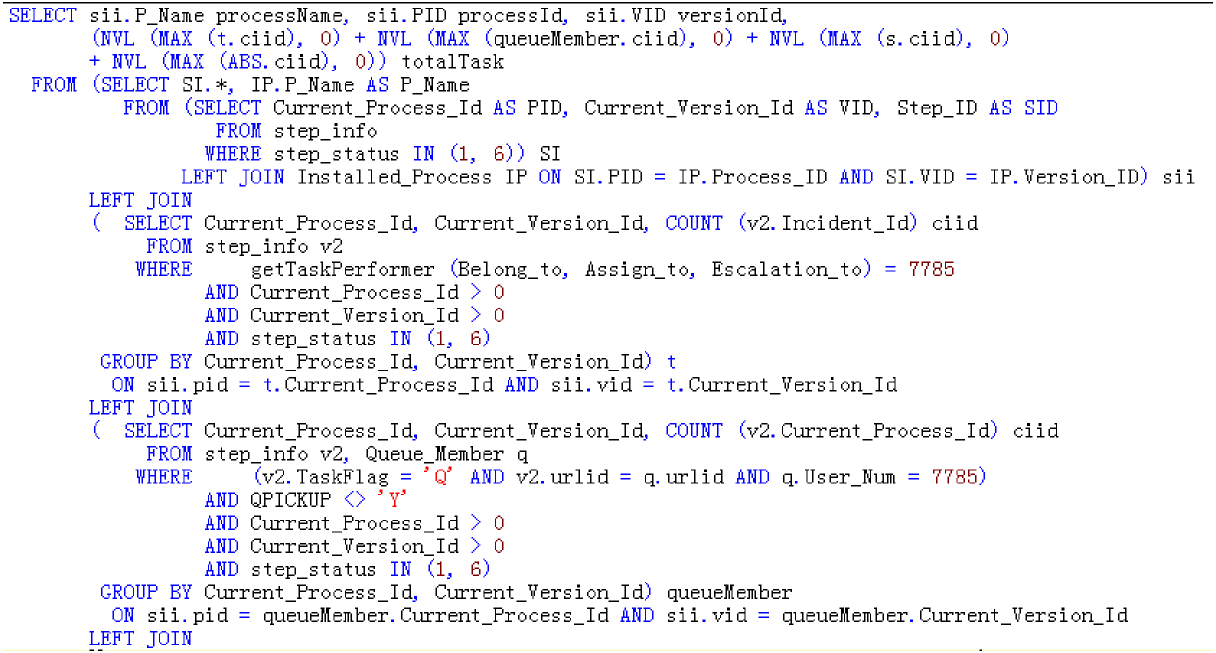
- 2012 年全国SQL大赛评委
- 2013 年全国SQL大赛评委

SQL优化的本质

- 如何加入如此优秀的SQL优化服务团队？
  - 熟悉SQL开发及数据库基本知识
  - 逻辑严密，思路清晰
  - 工作认真仔细，能吃苦
  - 工作年限不限，优秀的毕业生也可
  - HR@enmotech.com

- 技术方面你能获得什么提升？
  - 一年之后 能处理大部分初中级的SQL性能问题
  - 两年之后 能处理所有的初中级问题和部分高级的SQL性能问题
  - 三年之后 独立处理各种系统的SQL优化事宜

以一个“简单”的案例结束—SQL文本



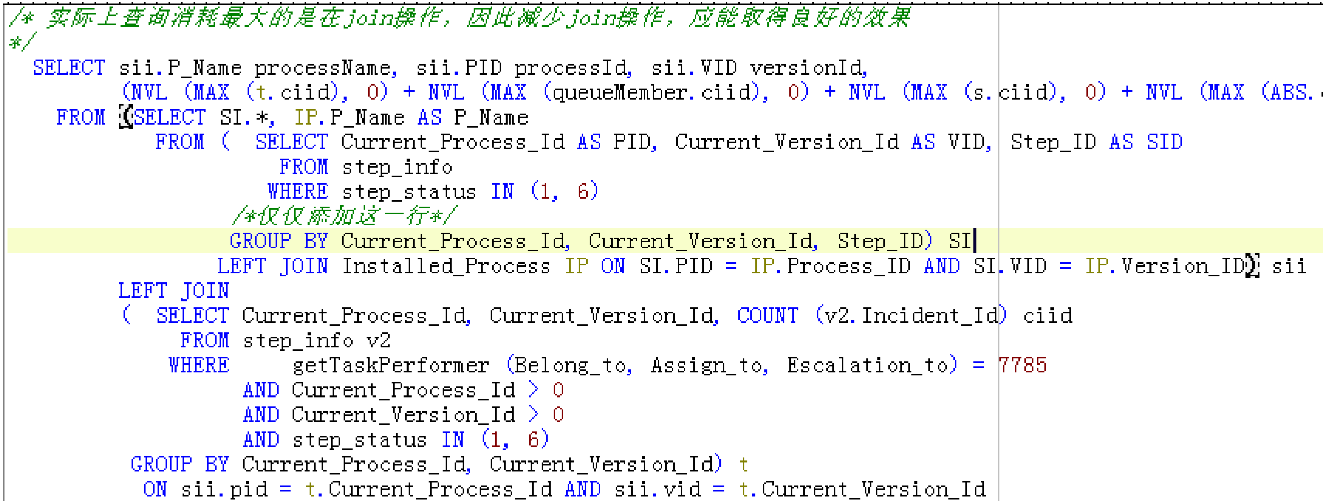
```
SELECT sii.P_Name processName, sii.PID processId, sii.VID versionId,
       (NVL (MAX (t.ciid), 0) + NVL (MAX (queueMember.ciid), 0) + NVL (MAX (s.ciid), 0)
       + NVL (MAX (ABS.ciid), 0)) totalTask
  FROM (SELECT SI.*, IP.P_Name AS P_Name
          FROM (SELECT Current_Process_Id AS PID, Current_Version_Id AS VID, Step_ID AS SID
                  FROM step_info
                 WHERE step_status IN (1, 6)) SI
               LEFT JOIN Installed_Process IP ON SI.PID = IP.Process_ID AND SI.VID = IP.Version_ID) sii
       LEFT JOIN
       (  SELECT Current_Process_Id, Current_Version_Id, COUNT (v2.Incident_Id) ciid
            FROM step_info v2
           WHERE     getTaskPerformer (Belong_to, Assign_to, Escalation_to) = 7785
                 AND Current_Process_Id > 0
                 AND Current_Version_Id > 0
                 AND step_status IN (1, 6)
        GROUP BY Current_Process_Id, Current_Version_Id) t
         ON sii.pid = t.Current_Process_Id AND sii.vid = t.Current_Version_Id
       LEFT JOIN
       (  SELECT Current_Process_Id, Current_Version_Id, COUNT (v2.Current_Process_Id) ciid
            FROM step_info v2, Queue_Member q
           WHERE     (v2.TaskFlag = 'Q' AND v2.urlid = q.urlid AND q.User_Num = 7785)
                 AND QPICKUP <> 'Y'
                 AND Current_Process_Id > 0
                 AND Current_Version_Id > 0
                 AND step_status IN (1, 6)
        GROUP BY Current_Process_Id, Current_Version_Id) queueMember
         ON sii.pid = queueMember.Current_Process_Id AND sii.vid = queueMember.Current_Version_Id
       LEFT JOIN
```

以一个“简单”的案例结束—优化思路

读懂业务，

算法为王！

以一个“简单”的案例结束—优化结果



```
/* 实际上查询消耗最大的是在join操作，因此减少join操作，应能取得良好的效果
*/
  SELECT sii.P_Name processName, sii.PID processId, sii.VID versionId,
         (NVL (MAX (t.ciid), 0) + NVL (MAX (queueMember.ciid), 0) + NVL (MAX (s.ciid), 0) + NVL (MAX (ABS.
    FROM (SELECT SI.*, IP.P_Name AS P_Name
            FROM (  SELECT Current_Process_Id AS PID, Current_Version_Id AS VID, Step_ID AS SID
                      FROM step_info
                     WHERE step_status IN (1, 6)
                  /*仅仅添加这一行*/
                  GROUP BY Current_Process_Id, Current_Version_Id, Step_ID) SI
                 LEFT JOIN Installed_Process IP ON SI.PID = IP.Process_ID AND SI.VID = IP.Version_ID) sii
         LEFT JOIN
         (  SELECT Current_Process_Id, Current_Version_Id, COUNT (v2.Incident_Id) ciid
              FROM step_info v2
             WHERE     getTaskPerformer (Belong_to, Assign_to, Escalation_to) = 7785
                   AND Current_Process_Id > 0
                   AND Current_Version_Id > 0
                   AND step_status IN (1, 6)
          GROUP BY Current_Process_Id, Current_Version_Id) t
            ON sii.pid = t.Current_Process_Id AND sii.vid = t.Current_Version_Id
```

以一个“简单”的案例结束—小结

- 四两拨千斤，有木有？

总 结

- 程序=数据+算法
- 读懂业务，算法为王，写出极致SQL，达到SQL优化的极高境界！

- 对于开篇那个复杂的案例，如果你能独立完成全面的SQL优化工作（即不仅仅是我讲解过的方法的组合），不要犹豫，即刻发送简历，您有可能直接入职云和恩墨！

关注微信公众号 获取文档和更新

云和恩墨

（支持ORA错误自动查询）

恩墨学院

Oracle新闻

Z3 – SQL审核工具提升SQL质量

- 独特的 SQL 视角
  - SQL生命周期管理 – 捕获，分析，归档，形成SQL全周期管理平台；

z3 – SQL审核

zData – 高性能弹性分布式存储解决方案

- 大数据整合与集中面临的平台压力
  - 去"IE"架构，通过Virtual SAN替代FC SAN
  - 同样的成本获得 20倍+ 的IO性能
  - 动态扩展，高性能的存储解决方案

zData – 分布式存储

Announcing：BayMax 自动化巡检 即将开放云服务

- 自动化巡检 – 让DBA去完成那20%最有价值的工作

BayMax自动化巡检

特别感谢　合作伙伴

云和恩墨
ENMOTECH

Shannon Systems
宝存科技

ACOUG
All China Oracle User Group
中国 Oracle 用户组

恩墨学院
ENMOEDU

Data Driven World
EN·CORE
恩核(北京)信息技术有限公司

OCMU
OCM联盟

SDOUG
Shandong Oracle User Group
山东Oracle用户组

ITPUB

ChinaUnix

IT168.com
www.it168.com

31会议

Mellanox
TECHNOLOGIES
Connect. Accelerate. Outperform.™

乐导科技
alidao.com

慕课网
imooc.com

清華大學出版社

Broadview®
www.broadview.com.cn

TURING
图灵教育

华章科技

THANKS