# ICS性能优化开发方案

主要体现以一下几条SQL语句方面

1. SQL语句1

select count(user\_id) from sp.sp\_user\_login\_his where org\_no=:1

此语句是查询ICS系统历史登录人数，每个小时的执行次数大约为223次，每次执行的时间为10s。

在不更改业务逻辑的前提下，不太适合建立索引，可以考虑降低执行频率

查询数据量，

一次执行28次，每个月执行一次

优化方案：

一次将所有网省的数据都查询出来

优化SQL：

select org\_no,count(1) cn from sp.sp\_user\_login\_his where org\_no in ('') group by org\_no;

生产环境测试执行时间：30s

2、SQL语句2

DELETE FROM R\_SER\_BUSI\_STATI WHERE STATTYPE = :B5 AND ORG\_NO = :B4 AND STAT\_DATE BETWEEN TO\_CHAR(:B3 , :B1 ) AND TO\_CHAR(:B2 , :B1 )

此语句涉及的表R\_SER\_BUSI\_STATI没有主键，也没有索引，每次的执行时间为100S。可以考虑在字段STAT\_DATE上创建索引。

具体什么作用

1. SQL语句3

select count(\*) from ((select A.USER\_ID from sp.sp\_user\_login A where A.login\_time>= trunc(SYSDATE) and A.login\_time<= trunc(SYSDATE+1) and A.org\_no=:1 ) UNION (select B.USER\_ID from sp.sp\_user\_login\_his B where B.login\_time>= trunc(SYSDATE) and B.login\_time<= trunc(SYSDATE+1) and B.org\_no=:2 ))

此语句是查询ICS系统当天登录人数，每个小时的执行次数大约为220次，每次执行的时间为10s。

在不更改业务逻辑的前提下，不太适合建立索引，可以考虑降低执行频率,并建议把count（\*）修改为count(1)。

SQL语句优化

一次执行28次，每个月执行一次

优化方案：

一次将所有网省的数据都查询出来

优化SQL：

select nvl(a.org\_no,b.org\_no)org\_no, nvl(a.cn1,0)+nvl(b.cn2) cn

from (select org\_no, count(1) cn1

from sp.sp\_user\_login a

where a.login\_time >= trunc(SYSDATE)

and a.login\_time <= trunc(SYSDATE + 1)

and org\_no in ('')

group by org\_no) a full join

(select org\_no, count(1) cn2

from sp.sp\_user\_login\_his a

where a.login\_time >= trunc(SYSDATE)

and a.login\_time <= trunc(SYSDATE + 1)

and org\_no in ('')

group by org\_no) b

on a.org\_no = b.org\_no;

生产环境测试执行时间：10s

1. SQL语句4  
   select count(\*) from knwlg\_base t where t.know\_status = '01' and t.end\_time>=sysdate and (t.organ\_id = :1 OR (T.SHARE\_FLAG='1' and T.KNOW\_ID in (select t.know\_id from KNWLG\_ENABLE\_ORG t where t.org\_id =:2 ))) and t.cat\_id IN (SELECT r.INFO\_ID FROM SER\_CATALOG\_INFO r where r.info\_id in (select info\_id from SER\_DIRECTORG\_RELA where org\_no = :3 ) and length(r.info\_id) = '6' and length(r.pr\_info\_id) = '6' start with r.info\_id = '1' connect by prior r.info\_id = r.pr\_info\_id)

此语句是ICS系统首页“知识库列表”检索SQL，工作日系统高峰期3个小时的执行次数大约为33,458次，系统最低峰早上5点~6点期间，此语句的执行数据也有3714次，虽然每次执行的时间仅为0.07s，但对数据库的压力也不小。与开发人员沟通了解到此功能为每10秒钟系统自动刷新执行的SQL语句。而且此语句是对旧知识库的数据进行的统计，目前此功能ICS系统几乎无人使用。

可以考虑降低执行频率，并建议把count（\*）修改为count(1)。

首页JS定时器去掉，消息提醒功能优化下次专题

1. SQL5

select a.tablespace\_name, ROUND(a.bytes/1024/1024, 3) SUMSIZE, ROUND((a.bytes-b.bytes)/1024/1024, 3) USEDSIZE, ROUND(b.bytes/1024/1024, 3) fREESIZE, round(((a.bytes-b.bytes)/a.bytes)\*100, 2) PERUSED from (select tablespace\_name, sum(bytes) bytes from dba\_data\_files group by tablespace\_name) a, (select tablespace\_name, sum(bytes) bytes, max(bytes) largest from dba\_free\_space group by tablespace\_name) b where a.tablespace\_name=b.tablespace\_name and a.tablespace\_name='DATA\_ICS'

此语句是ICS系统KPI指标定时执行的查询表空间使用率的语句，每小时执行的次数为233次，每次执行的时间为1.44秒，由于此语句的结果是一样的，不会有太大变动，可以考虑降低执行频率。

日指标

1. 其它SQL语句

1）、UPDATE T\_TD\_FIRED\_TRIGGERS SET INSTANCE\_NAME = :1 , FIRED\_TIME = :2 , STATE = :3 , JOB\_NAME = :4 , JOB\_GROUP = :5 , IS\_NONCONCURRENT = :6 , REQUESTS\_RECOVERY = :7 WHERE SCHED\_NAME = 'scheduler' AND ENTRY\_ID = :8

2）、UPDATE T\_TD\_EXT\_FIRED\_TRIGGERS\_HIS SET JOB\_NAME = :B3 , JOB\_GROUP = :B2 , STATE = :B1 WHERE ENTRY\_ID = :B7 AND SCHED\_NAME = :B6 AND TRIGGER\_NAME = :B5 AND TRIGGER\_GROUP = :B4

3）、DELETE FROM T\_TD\_FIRED\_TRIGGERS WHERE SCHED\_NAME = 'scheduler' AND ENTRY\_ID = :1  
4）、UPDATE T\_TD\_EXT\_FIRED\_TRIGGERS\_HIS SET END\_TIME = (SYSDATE - 8 / 24 - TO\_DATE('1970-01-01', 'yyyy-mm-dd')) \* 86400000 , STATE = 'COMPLETED' , REQUESTS\_RECOVERY = :B1 WHERE ENTRY\_ID = :B5 AND SCHED\_NAME = :B4 AND TRIGGER\_NAME = :B3 AND TRIGGER\_GROUP = :B2

上述4条语句所涉及的表所属用户为SGUAP，每次执行时间均为30S左右，为KPI定时任务调用时执行的SQL语句，与开发沟通确认，这四条语句暂无优化建议。

执行次数

开发优化建议：

一、SQL1，SQL3，SQL5都是统计计算KPI指标项，SQL1和SQL3是统计ICS历史登录数和ICS当天登录数，SQL5统计的是ICS表空间使用率,为了优化ICS数据库执行效率，缓解数据库访问压力，现提出对KPI指标项统计访问的三张UAP底层表进行二次封装，及数据库占用空间大小统计进行封装，创建一张新表定时访问UAP底层表，定时统计数据库占用表空间大小，将统计结果存储到新表，应用层直接获取这张新表，从而达到缓解数据库底层的压力。

经过汇总KPI指标项统计计算数据的来源为：

业务数据占用表空间大小：dba\_data\_files dba\_free\_space

在线用户数： sp.sp\_user\_login

日登录人数：sp.sp\_user\_login sp.sp\_user\_login\_his

累计访问次数： sp.sp\_user\_login\_his

|  |  |  |
| --- | --- | --- |
| QOS\_KPI\_VALUE\_TEMP | | |
| 字段编码 | 数据类型 | 字段名 |
| TEMP\_ID | VARCHAR2(16) | ID |
| ORG\_NO | VARCHAR2(32) | 网省编码 |
| KPI\_DATE | DATE | 日期 |
| AVGSER\_RESP\_TIME | NUMBER(10,4) | 服务平均响应时长 |
| SESS\_NO | NUMBER(5) | 页面会话连接数 |
| ONLINE\_OP\_NO | NUMBER(5) | 在线用户数 |
| REGIST\_NO | NUMBER(5) | 注册用户数 |
| TOTAL\_VISTNO | NUMBER(15) | 累计访问人次 |
| BUS\_RATE | NUMBER(10,6) | 业务数据占用表空间大小 |
| D\_LOGIN\_NO | NUMBER(15) | 日登陆人数 |
| DB\_RESP\_TIME | NUMBER(12,4) | 数据库平均响应时长 |



Create product

补充优化

1. 查询在线用户数

原来是每个网省单独查询，修改为一次查询

select org\_no,count(distinct(USER\_ID)) cn from sp.sp\_user\_login where login\_time>= trunc(SYSDATE) and org\_no in('') group by org\_no;

1. 查询上次KPI指标

原来是每个网省单独查询，修改为一次查询（由于要查询最新的一条数据，使用了oracle的分析函数）

select \* from (select row\_number() over(partition by org\_no order by KPI\_DATE DESC NULLS LAST) rno,KPI\_ID kpiId,

KPI\_CLASS kpiClass,

ORG\_NO orgNo,

to\_char(KPI\_DATE, 'yyyy-MM-dd hh24:mi:ss') kpiDate,

AVGSER\_RESP\_TIME avgserRespTime,

SYSRUN\_TIME sysrunTime,

SESS\_NO sessNo,

ONLINE\_OP\_NO onlineOpNo,

REGIST\_NO registNo,

TOTAL\_VISTNO totalVistno,

BUS\_RATE busRate,

D\_LOGIN\_NO dLoginNo,

DB\_RESP\_TIME dbRespTime

from QOS\_IMS\_KPI WHERE ORG\_NO in('')) a where rno=1