

# Oracle 的监控系统

---分享自 2018 年 9 月 1 日\_日月新\_时悦 DBA

首先说下开发这套系统的初衷这个系统的初衷

## 1. 开发这套系统的初衷

### 1.1 快速了解一套数据库

不论甲方还是乙方，拿到一套数据库我们很难快速的知道他的配置，虽然我们手里有很多运维的脚本，但是无法有效的统一起来

### 1.2 提前定位性能瓶颈

如果你对一套系统不了解，在运维过程中我们往往是迷茫的，心里没有底的特别是性能问题

### 1.3 多套数据库统一管理

虽然 Oracle 有他的统一管理工作，但是我想大多数还是不用的吧

### 1.4 练手 Python

正好对于 Python 使用也有一段时间了，而工作上也有这种需求，所以才萌发了这个想法，根据自己实际运维中的需求来开发一套系统。

在开始今天的正式想说的是这套系统只是辅助我们日常的运维，对于 Oracle 本身的一些工具，如 `awr` 和 `statpack,ash` 等工具我们还是需要熟练掌握的

## 2. 接下来简单说下开发环境

操作系统:CentOS 7.4

Python 版本 :3.6

Django 版本: 1.10.5

linux/unix 模块:paramiko

Oracle 模块:cx\_Oracle

SQL Server 模块:pymssql

数据分析:pandas

前台展示:highcharts

数据存储:MySQL,redis

这里获取到的数据是存放在 MySQL 和 redis 中的，具体对应后面会说。前端图表显示用的 highcharts。系统采用了 Python+Django 的框架。Django 这里就不说 等会会发文档出来。下面说下我们需要获取哪些数据。

## 2. 监控数据的获取：通过 paramiko 模块获取

### 3.1 Linux/hp-unix:

CPU 内存--5 分钟

磁盘使用率 --5 分钟

### 3.2 Oracle 数据库数据：通过 cx\_Oracle 模块获取

TOP SQL 语句(每小时)

系统状态数据 如物理读等 (每小时)

等待事件(每小时)

命中率信息(每小时)

表空间使用情况(每天)

Job 执行情况(每小时)

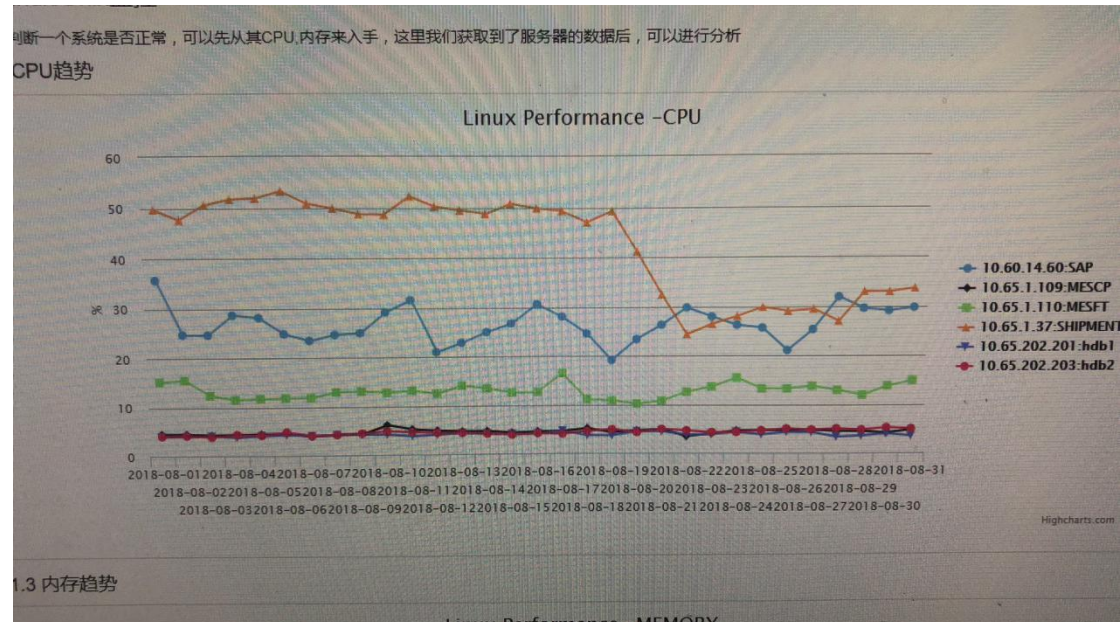
主要通过如下视图:1. `v$sqlarea` 2. `v$sysstat` 3. `v$system_event` 4. `v$librarycache`。好了 数据有了记下来就是如何处理他，达到我们想要的效果。这里使用 pandas 做数据的处理。

这里按照我日常分析数据库的一个步骤来介绍这套系统的功能。

### 3. 监控系统的功能

#### 4.1 Linux/Unix 监控

我们要判断一个系统是否正常，可以先从其 CPU、内存来入手。这里我们可以看到 CPU/内存使用的情况，支持按天和按小时查看。



同样我们也可以实时获取捞到的数据来进行报警

#### 4.2 Oracle 监控

4.2.1 数据库巡检程序：这个功能是最近加上的，巡检的项目可能不全，但是理论上只要有数据都可以分析的~主要检查

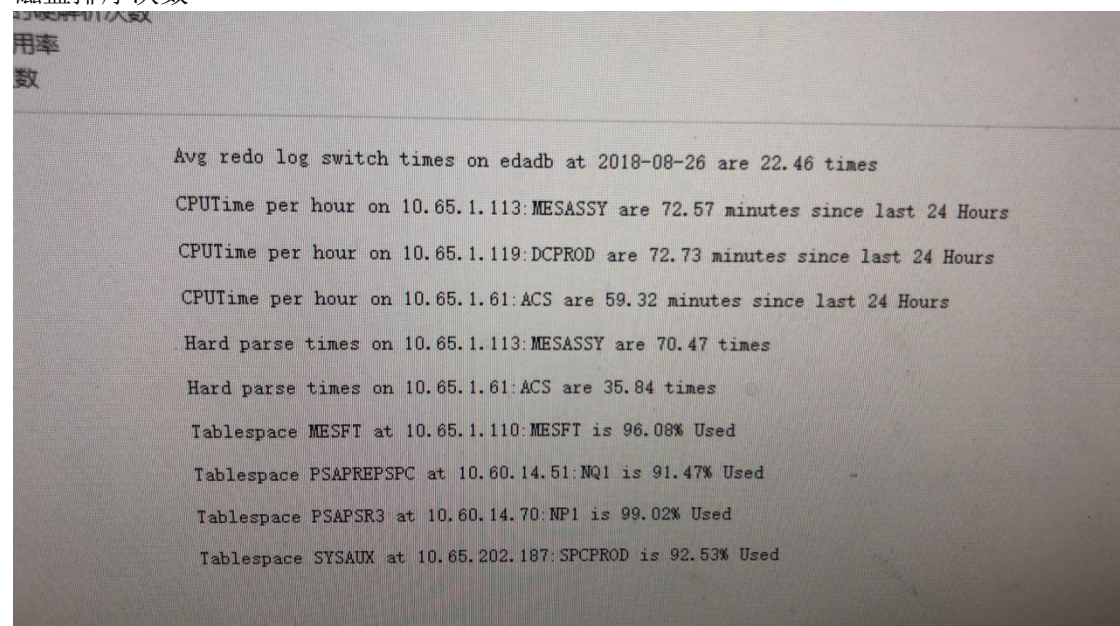
每小时 redo log 的产生量

每小时数据库的 CPUTime

数据库每秒的硬解析次数

表空间的使用率

磁盘排序次数

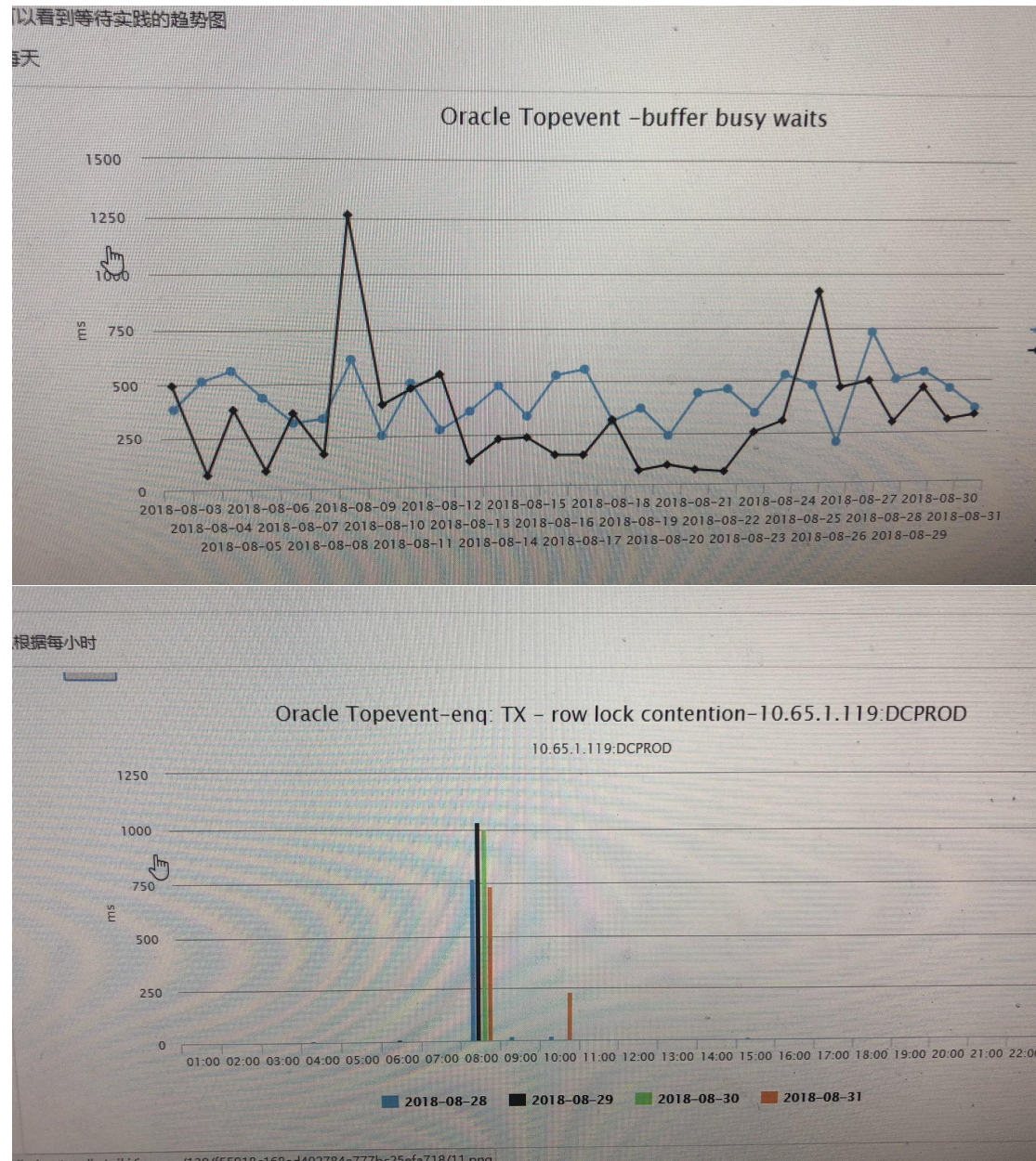




4.2.2 数据库的等待事件：接下来是一些等待事件的状态，捞取的是非空闲的等待事件，这里设置的是平均等待时间超过 30ms。通过上面两个程序并结合 CPU,内存使用率我们大体可以知道一套数据库的性能情况

接下来我们可以根据上面的问题点进行分析

4.2.3 数据库等待事件检查：上面我们可能从报告中找到了某个等待事件似乎特别明显这时可以用如下图表观看



这里分 2 个，按天的话可以看出每天整体的量。按小时的话可以看到某天的各个时段相应的状态。例如上面可以看出 DC 每天 7~8 点的 row lock 较明显。这样就可以对数据库的等待事件有所了解。

4.2.4 数据库性能检查，这里我们针对常见的性能指标来画出趋势图，有：

Physical Reads

Logical Reads

DB Time

CPU Time



Hard Parse  
Total Parse  
User Commits  
User Rollbacks  
Logons  
Redo Size

这里也分每天的趋势图和每小时的趋势图



这里同样可以按照每天和每小时来查看。

#### 4.2.5 数据库 TOPSQL 检查

通过上面的检查我们可以定位到某小时某个指标较高，这时我们可以查询这个时间段的 TOP SQL，主要有：

diskreads  
buffergets  
elapsedtime

Cputime



这里可以看到具体的语句和模块。

4.2.6 数据库命中率查询：这个功能还在评估要不要加入

4.2.7 数据库基线设置：可以设置数据库某个指标的某天趋势为基线方便对比

4.2.8 数据库常用命令执行：这里还有个功能就是对于我们日常用到的一些运维脚本集成到网页上方便执行，主要有

检查数据文件创建时间

检查表的分析时间

查看数据库段的大小

查看进程对应的 SQL 语句

查看会话对应的进程号

查看 SQL 的执行计划

检查临时表空间使用率

检查执行次数等于一的语句

检查未绑定语句



请选择数据库				
ACS 10.65.1.61				
请选择命令				
检查数据文件创建时间				
确定				
数据文件创建时间-10.65.1.61-ACS				
数据文件名称	文件大小	表空间	自动扩展	创建
/u01/app/oracle/oradata/ACS/users8.dbf	1024	USERS	NO	2018
/u01/app/oracle/oradata/ACS/users7.dbf	1024	USERS	NO	2018
/u01/acsdata/datafile/acs_idx07.dbf	2048	ACS_IDX	NO	2018
/u01/acsdata/datafile/acs_idx06.dbf	2048	ACS_IDX	NO	2018
/u01/app/oracle/oradata/ACS/users6.dbf	1024	USERS	NO	2018
/u01/acsdata/datafile/acs05.dbf	4096	ACS01	NO	2017

5.监控程序的调用：目前数据获取的程序是通过 **crontab** 调用的

6. 异常处理：所有程序都写了异常处理模块，并重定向了标准错误输出到文件以确保所以程序异常得到及时处理

7. 监控报警机制，最后总结下，这套 系统有哪些报警机制：支持通过邮箱和微信企业号进行通知。指标先紧后松，一步步修改，主要有以下方面。

连通性测试:每次的数据获取如有异常则报警

CPU/内存:连续 5 次大于设定阈值则报警

分区使用率:大于 90%则报警

表空间使用率:大于 90%则报警

Job 执行情况:执行失败或超过 2 小时报警

备份情况监控:备份失败则报警

DataGuard 监控:DataGuard 是否同步

Oracle alert 日志有错误报警

等待事件平均等待时间超过 30ms 报警

redo log 每小时超过 12 个报警

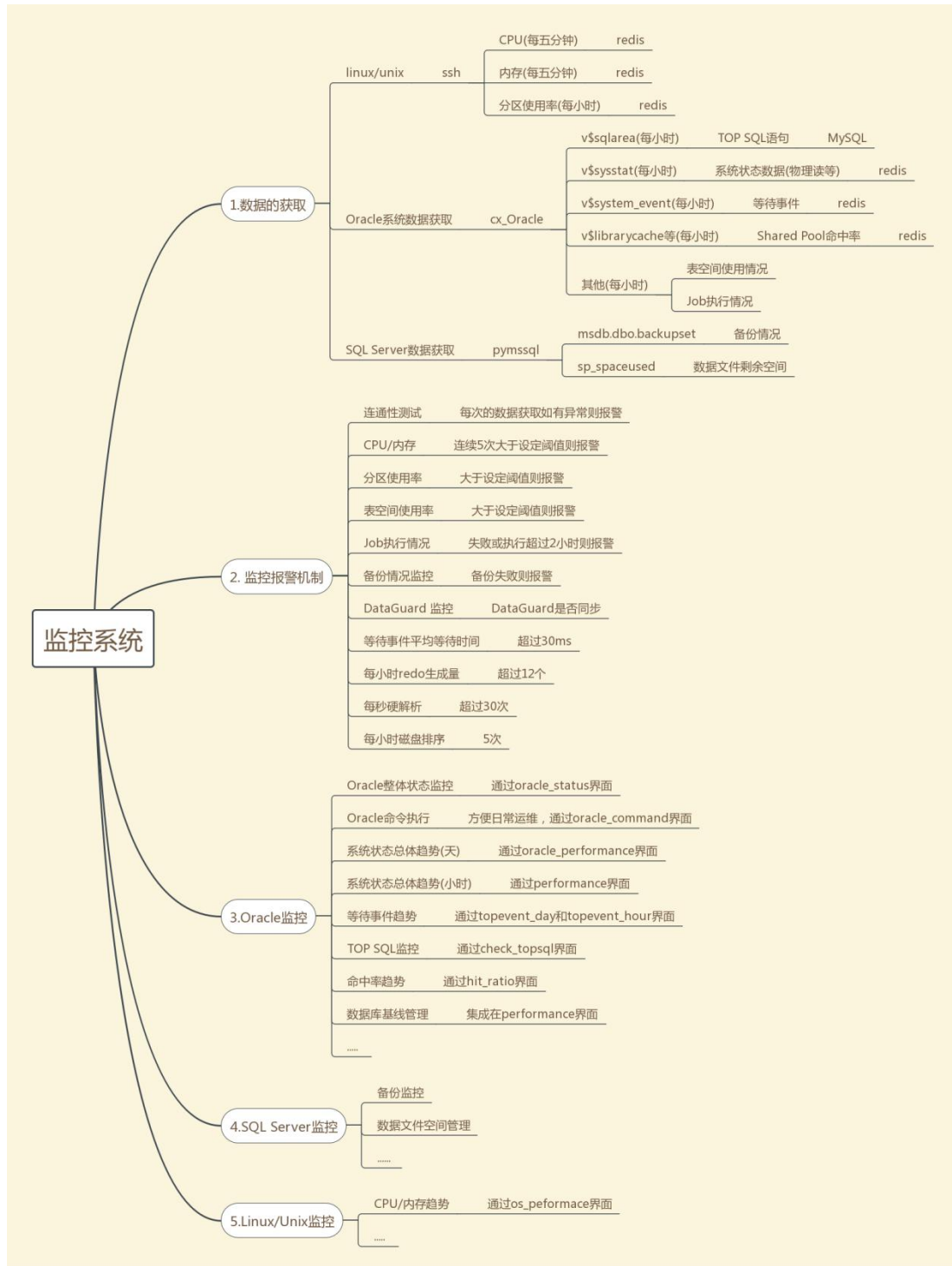
硬解析每秒超过 30 次报警

磁盘排序每小时超过 5 次报警

....

所有报警的阈值可以针对数据库或服务器单独设置

最后发个思维导图，方便大家查看



好了 今天的分享到这里就结束了 有疑问可以提出来 谢谢！