

# 阿里云**ApsaraDB**诊断报告

---

## 1 文档简介

---

### 1.1 文档目的

本文档描述了阿里云客户支持服务对当前问题状态的理解和认识，并给出相关建议和结论。

本文档既用于让客户对当前问题有全面的了解，也用于协调各相关人员对问题状态达成统一的认识。

---

### 1.2 问题类别定义

编号	类别	说明
1.	应用问题	应用问题包含开发、设计等问题。
2.	操作使用问题	操作使用问题包含操作不规范,操作失误等问题。
3.	产品问题	产品问题包含软件产品、硬件产品及第三方产品问题。
4.	未知问题	未知问题指当前无法判断的问题类别。

---

### 1.3 问题严重级别定义

以下为我们对系统环境中问题的严重级别定义：

编号	类别	说明
S1	系统失败，任何处理都不能继续。	
S2	不能够继续选定的任务处理。	
S3	受限制的功能能力，但处理可以继续。	
S4	目前可以忽略的小问题。	

## 2 规格配置

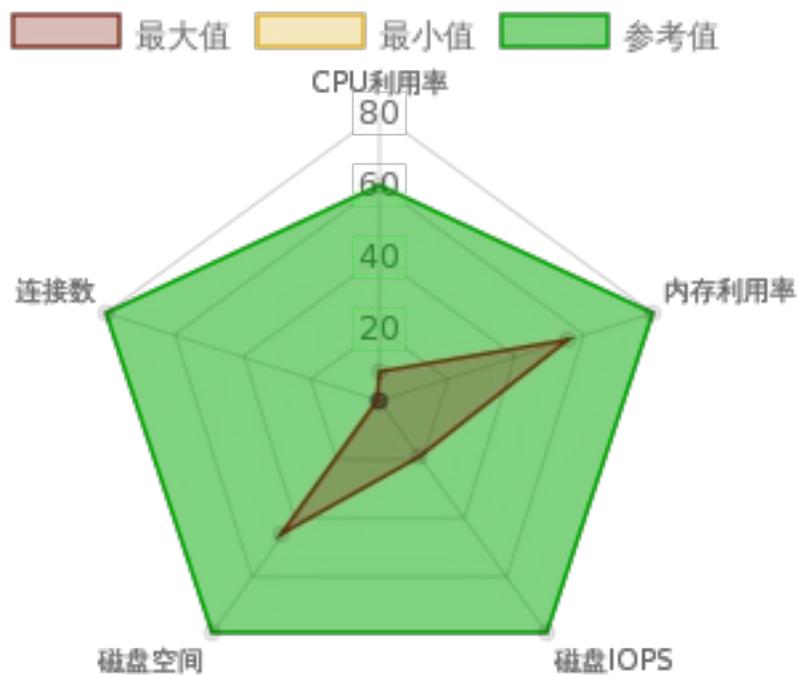
配置项	规格名称
实例名	rr-2ze83o0z0lc4tq346
数据库类型	mysql
数据库版本	5.6
规格名称	MySQL新规格8核16G
存储空间(M)	204800
最大连接数	4000
最大IOPS	8000
QPS参考值	999999

综合评分：75分



### 3 系统状态

时间范围 2018-12-16 23:30 至 2018-12-17 00:30



资源名称	资源状态	最大值	最小值	平均值	参考值
CPU利用率	空闲	8.20%	0.00%	1.06%	40%~60%
内存利用率	空闲	55.30%	55.30%	55.30%	60%~80%
磁盘IOPS	空闲	19.40%	0.10%	0.82%	40%~80%
磁盘空间	空闲	46.20%	46.20%	46.20%	60%~80%
连接数	空闲	0.80%	0.40%	0.52%	40%~80%

---

## 4 问题与建议

---

### 4.1 死锁

没有发现死锁。

---

### 4.2 锁等待

没有发现锁等待。

---

### 4.3 实例安全

没有发现相关问题。

---

### 4.4 连接数

没有发现相关问题。

---

### 4.5 内存

没有发现相关问题。

---

### 4.6 实例空间

没有发现相关问题。

## 5 SQL分析

### 5.1 慢SQL分析

#### 5.1.1 SQL-1

SQL语句	<pre>SELECT task_id, garage_id, garage_code, company_code, org_id , task_type, car_type, user_account, user_phone, user_org_code , user_type, survey_number, survey_name, survey_mobile, survey_org , regist_no, regist_date, loss_id, license_no, vehicle_model , vehicle_brand, brand_code, car_owner, mobile, policy_no , address, province_name, city_name, district_name, detail_address , longitude, latitude, remark, batch_status, status , start_time, expected_time, confirm_time, inform_time, finish_time , task_source, ev_time, ev_id, fail_count, fx_garage_id , giveup_reason_code, giveup_reason_name, province_code, city_code, town_code , note, end_msg, repair_type, repair_name, repair_org , car_application, create_by, create_time, update_by, update_time FROM et_repair_task WHERE status = ? AND update_time &lt;= ? AND task_type = ? AND garage_code IS NULL AND regist_no IS NOT NULL AND batch_status = ? AND fail_count &lt; ? ORDER BY update_time</pre>
建议	<ol style="list-style-type: none"> <li>1. 行扫描 行扫描总数:28160614 , 执行次数:28。</li> <li>2. <b>order by</b> 该语句使用了"order by"。注意检查是否有效利用到了索引。</li> <li>3. 索引 该语句没有使用聚合函数和模糊查询。行扫描很高且没有记录返回。创建合理索引通常会带来很大的性能提升。</li> </ol>

#### 5.1.2 SQL-2

SQL语句	<pre>SELECT DISTINCT ui.case_no FROM prpm_upload_image ui LEFT JOIN prpm_loss_item l ON ui.case_no = l.case_no WHERE ui.stay_upload != ? AND l.loss_status = ? UNION SELECT DISTINCT ui.case_no FROM prpm_upload_video ui LEFT JOIN prpm_loss_item l ON ui.case_no = l.case_no WHERE ui.stay_upload NOT IN (?) AND l.loss_status = ? AND ui.video_type = ? UNION SELECT DISTINCT l.case_no FROM prpm_upload_loss_item_img ui LEFT JOIN prpm_loss_item_cyp l ON ui.loss_item_cyp_id = l.loss_item_cyp_id WHERE ui.stay_upload != ? AND (l.handle_status IN (?) OR ui.loss_item_cyp_id IS NULL OR ui.loss_item_cyp_id = ?)</pre>
建议	<ol style="list-style-type: none"> <li>行扫描 行扫描总数:5970934，执行次数:3。</li> <li><b>union</b> 该语句使用了"union"。建议检查语义是否可以替换成"union all"。</li> <li>索引 该语句没有使用聚合函数和模糊查询。行扫描很高且没有记录返回。创建合理索引通常会带来很大的性能提升。</li> </ol>

### 5.1.3 SQL-3

SQL语句	<pre>SELECT l.case_info_cyp_id, l.case_no, ui.* FROM prpm_upload_loss_item_img ui LEFT JOIN prpm_loss_item_cyp l ON ui.loss_item_cyp_id = l.loss_item_cyp_id WHERE ui.stay_upload != ? AND (l.handle_status IN (?) OR ui.loss_item_cyp_id IS NULL OR ui.loss_item_cyp_id = ?) ORDER BY ui.create_time DESC LIMIT ?, ?</pre>
建议	<ol style="list-style-type: none"> <li>行扫描 行扫描总数:4757319，执行次数:3。</li> <li><b>order by</b> 该语句使用了"order by"。注意检查是否有效利用到了索引。</li> <li>索引 该语句没有使用聚合函数和模糊查询。行扫描很高且没有记录返回。创建合理索引通常会带来很大的性能提升。</li> </ol>

### 5.1.4 SQL-4

SQL语句	<pre>SELECT loss_item_id, case_info_id, case_no, loss_id, loss_name , loss_type, loss_status, lossitem_status, surveyor_code, surveyor_name , mobile, contact, contact_phone, end_case_time, finish_case_time , remark, user_type, check_loss_usercode, check_loss_username, org_code , org_name, loss_deal_type, vip_level, create_by, create_time , modify_by, modify_time, fill_status FROM prpm_loss_item WHERE 1 = 1 AND surveyor_code = ? AND create_time &gt;= ? AND create_time &lt;= ? AND loss_id != ? AND loss_status NOT IN (?)</pre>
建议	<ol style="list-style-type: none"> <li>行扫描 行扫描与行返回之比为314556。</li> <li>索引 该语句没有使用聚合函数和模糊查询，行扫描与行返回之比很高。创建合理索引会带来很大的优化空间。</li> </ol>

### 5.1.5 SQL-5

SQL语句	<pre>SELECT * FROM prpm_loss_item</pre>
建议	<ol style="list-style-type: none"> <li>返回记录数 语句最大返回记录数为3145557。该语句每次返回记录数过高，需要从业务和设计的角度评估是否可以优化。</li> </ol>

### 5.1.6 SQL-6

SQL语句	SELECT loss_item_id, case_info_id, case_no, loss_id, loss_name , loss_type, loss_status, lossitem_status, surveyor_code, surveyor_name , mobile, contact, contact_phone, end_case_time, finish_case_time , remark, user_type, check_loss_usercode, check_loss_username, org_code , org_name, loss_deal_type, vip_level, create_by, create_time , modify_by, modify_time, fill_status FROM prpm_loss_item WHERE 1 = 1 AND case_no LIKE CONCAT(?, ?) AND surveyor_code = ? AND create_time <= ? AND loss_id != ? AND loss_status NOT IN (?)
建议	<ol style="list-style-type: none"> <li>行扫描 行扫描总数:3145552 , 执行次数:1。</li> <li>索引 该语句没有使用聚合函数和模糊查询。行扫描很高且没有记录返回。创建合理索引通常会带来很大的性能提升。</li> </ol>

### 5.1.7 SQL-7

SQL语句	SELECT * FROM et_repair_task_dispatch
建议	<ol style="list-style-type: none"> <li>返回记录数 语句最大返回记录数为2558280。该语句每次返回记录数过高，需要从业务和设计的角度评估是否可以优化。</li> </ol>

### 5.1.8 SQL-8

SQL语句	SELECT * FROM prpm_item_used
建议	<ol style="list-style-type: none"> <li>返回记录数 语句最大返回记录数为2413036。该语句每次返回记录数过高，需要从业务和设计的角度评估是否可以优化。</li> </ol>

### 5.1.9 SQL-9

SQL语句	SELECT * FROM prpm_loss_item_history
建议	<ol style="list-style-type: none"> <li>返回记录数 语句最大返回记录数为2312256。该语句每次返回记录数过高，需要从业务和设计的角度评估是否可以优化。</li> </ol>

### 5.1.10 SQL-10

SQL语句	SELECT DISTINCT ui.* FROM prpm_upload_image ui LEFT JOIN prpm_loss_item l ON ui.case_no = l.case_no WHERE ui.stay_upload != ? AND l.loss_status = ? ORDER BY ui.create_time DESC LIMIT ?, ?
建议	<ol style="list-style-type: none"> <li>行扫描 行扫描总数:1948500，执行次数:4。</li> <li><b>order by</b> 该语句使用了"order by"。注意检查是否有效利用到了索引。</li> <li>索引 该语句没有使用聚合函数和模糊查询。行扫描很高且没有记录返回。创建合理索引通常会带来很大的性能提升。</li> </ol>

### 5.1.11 SQL-11

SQL语句	SELECT * FROM prpm_case_info_cyp
建议	<ol style="list-style-type: none"> <li>返回记录数 语句最大返回记录数为1452553。该语句每次返回记录数过高，需要从业务和设计的角度评估是否可以优化。</li> </ol>

### 5.1.12 SQL-12

SQL语句	SELECT * FROM prpm_loss_item_cyp
建议	<ol style="list-style-type: none"> <li>返回记录数 语句最大返回记录数为1452509。该语句每次返回记录数过高，需要从业务和设计的角度评估是否可以优化。</li> </ol>

### 5.1.13 SQL-13

SQL语句	SELECT * FROM prpm_case_info
建议	<ol style="list-style-type: none"> <li>返回记录数 语句最大返回记录数为1346722。该语句每次返回记录数过高，需要从业务和设计的角度评估是否可以优化。</li> </ol>

### 5.1.14 SQL-14

SQL语句	SELECT * FROM et_repair_task
建议	<ol style="list-style-type: none"> <li>1. 返回记录数 语句最大返回记录数为1059475。该语句每次返回记录数过高，需要从业务和设计的角度评估是否可以优化。</li> </ol>

### 5.1.15 SQL-15

SQL语句	SELECT * FROM et_repair_record
建议	<ol style="list-style-type: none"> <li>1. 返回记录数 语句最大返回记录数为279413。该语句每次返回记录数过高，需要从业务和设计的角度评估是否可以优化。</li> </ol>

### 5.1.16 SQL-16

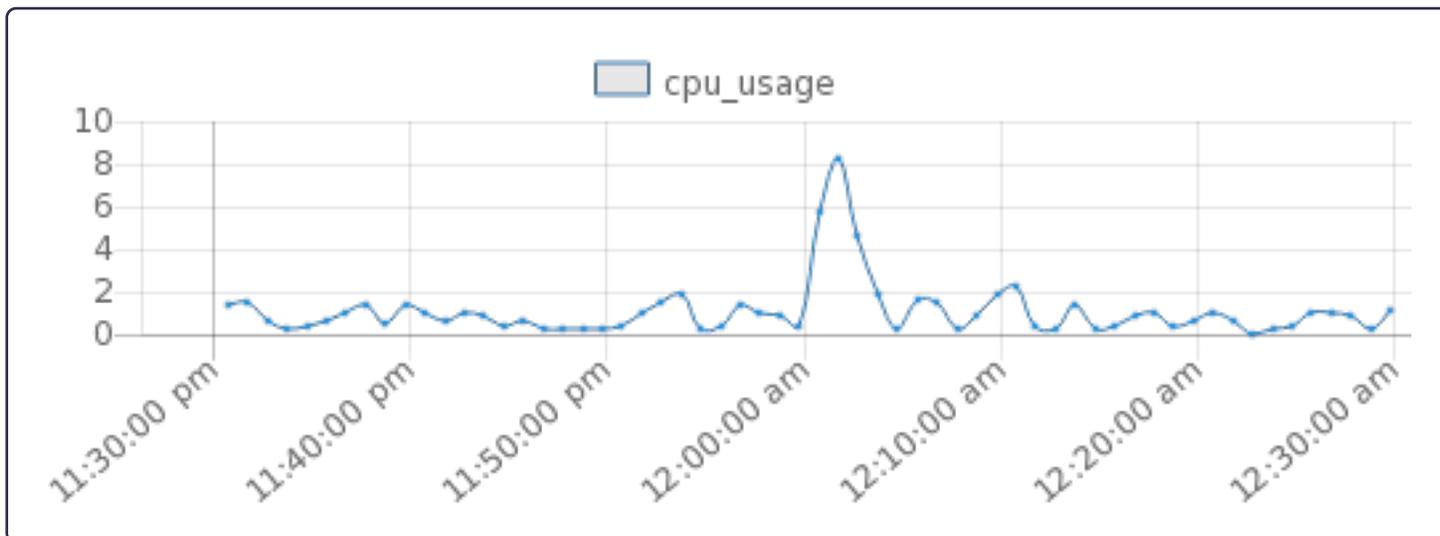
SQL语句	SELECT ui.* FROM prpm_upload_image ui WHERE ui.stay_upload = ? AND (ui.oss_key LIKE ? OR ui.img_type IN (??)) ORDER BY ui.modify_time DESC LIMIT ?, ?
建议	<ol style="list-style-type: none"> <li>1. 行扫描 行扫描总数:144421，执行次数:1。</li> <li>2. <b>order by</b> 该语句使用了"order by"。注意检查是否有效利用到了索引。</li> <li>3. 索引 该语句没有使用聚合函数和模糊查询。行扫描很高且没有记录返回。创建合理索引通常会带来很大的性能提升。</li> </ol>

### 5.1.17 SQL-17

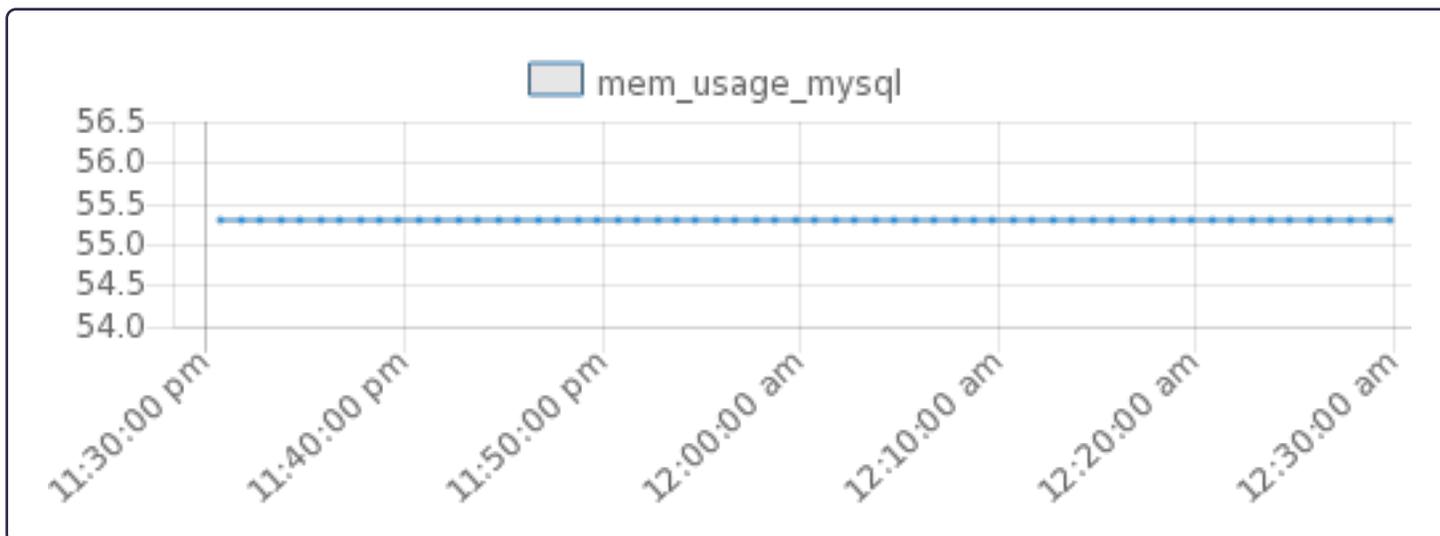
SQL语句	SELECT * FROM et_garage_core
建议	<ol style="list-style-type: none"> <li>1. 返回记录数 语句最大返回记录数为93724。该语句每次返回记录数过高，需要从业务和设计的角度评估是否可以优化。</li> </ol>

## 6 性能曲线

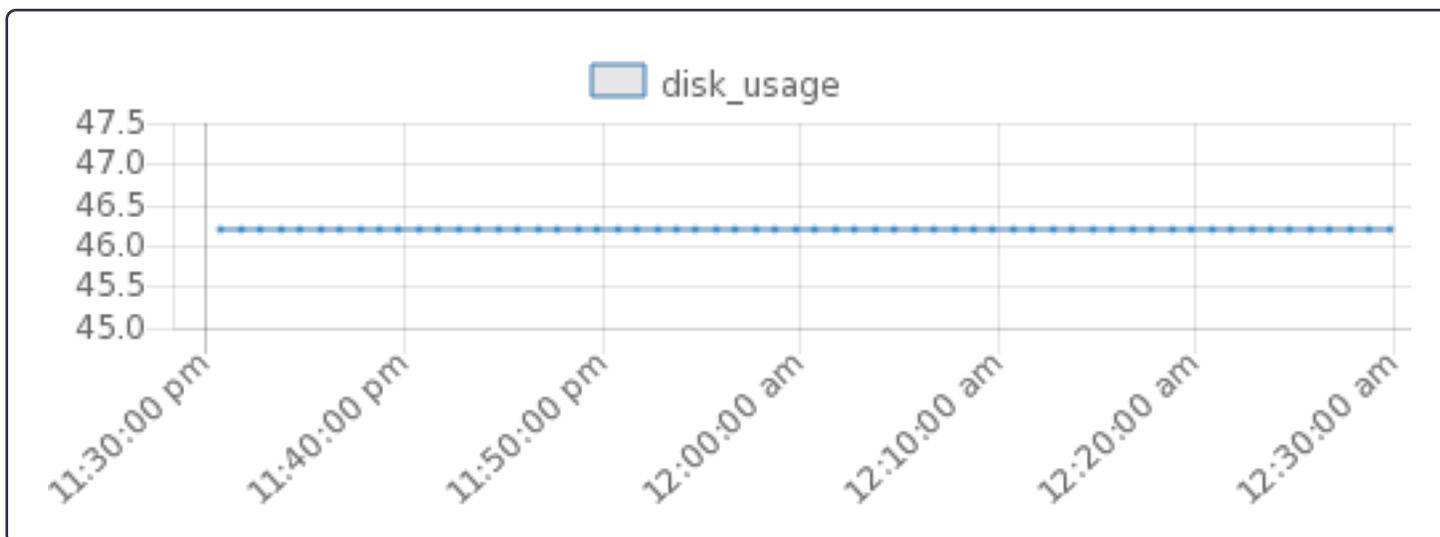
### 6.1 CPU利用率



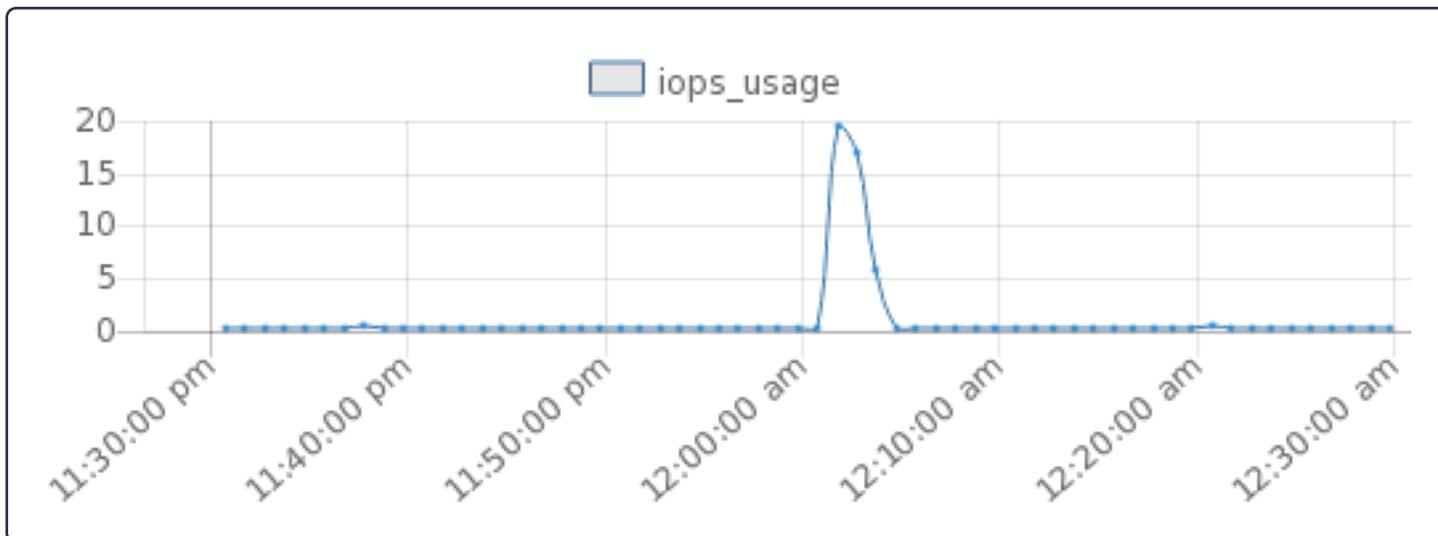
### 6.2 内存利用率



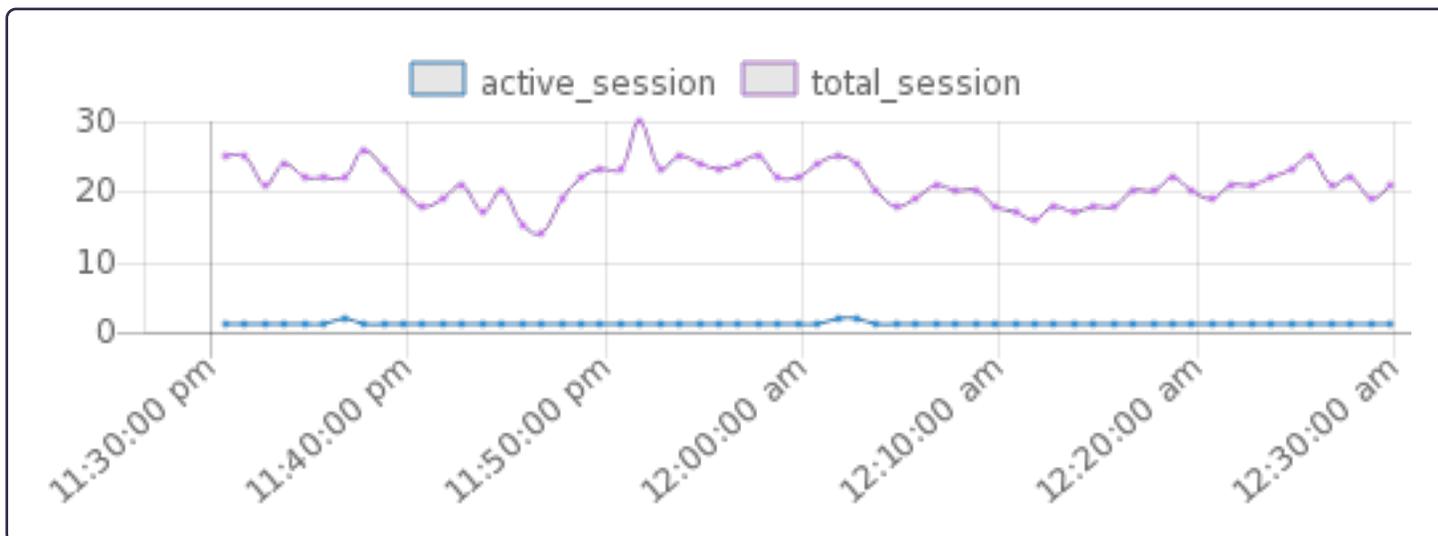
### 6.3 磁盘空间



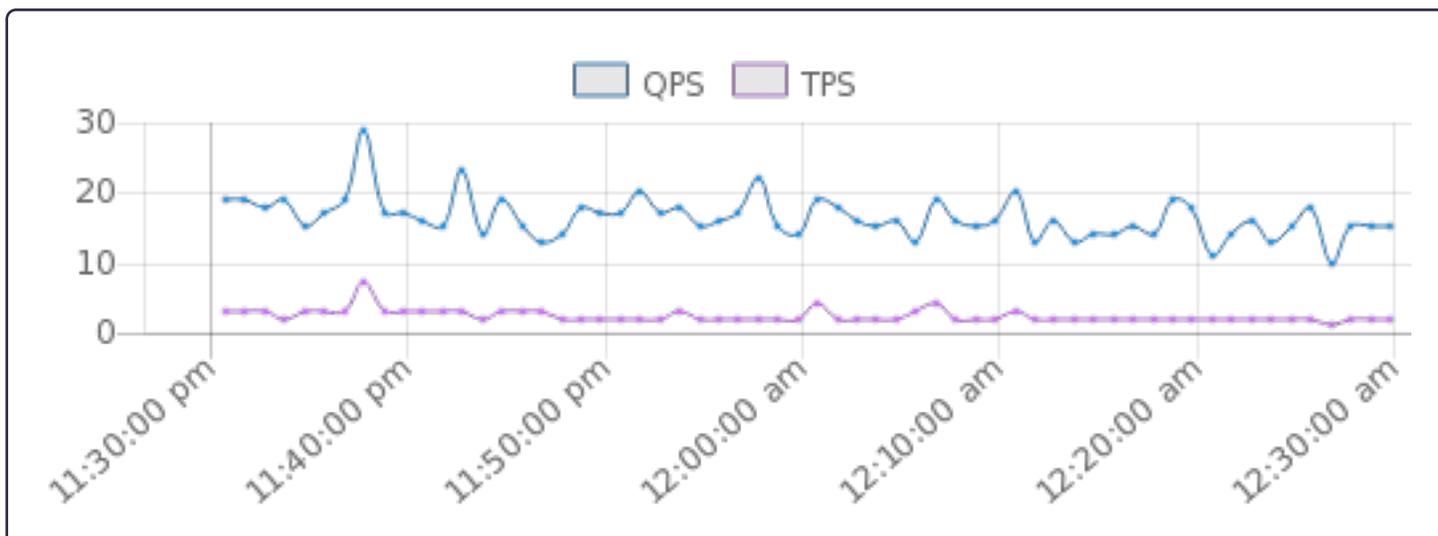
### 6.4 IOPS利用率



### 6.5 会话



### 6.6 QPSTPS



### 6.7 InnoDB

### ApsaraDB 诊断报告

