

Oracle中资源的共享与竞争及问题定位 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/143/2021_2022_Oracle_E4_B8_AD_E8_c102_143062.htm

前言 在单一的应用环境或业务相对简单的系统下,系统性能问题,瓶颈所在往往是不言自明,解决问题的前提 -- 定位问题。定位问题是比较容易解决的,但在一个复杂的应用环境下,各应用系统对系统资源往往是一种共享和竞争的关系,而且应用系统之间也可能存在着共生或制约的关系,资源利益的均衡往往是此消彼长,而这种环境下的应用系统一旦出现资源竞争,系统的瓶颈往往难以断定,甚至会发生不同应用设计人员之间互相推诿责任的扯皮现象。本文仅就此问题对 Linux 平台下各应用系统对 ORACLE 数据库的使用情况作一探讨,ORACLE 数据库的 TUNING 不是一个可以一言以蔽的主题,本文无意概全,内容仅涉及问题的定位及各应用对数据库资源的共享与竞争问题。 本文试验及问题取证的环境: RedHat6.1 Web server(Apache1.3.9 PHP4.0)

Client/Server(Pro*C) 之 Server 端 RedHat6.2 Oracle8.1.6.1.0

RedHat7.1 Web server(Apache1.3.20 PHP4.06) Oracle8.1.7.0.0 为方便问题的讨论,应用系统已做简化,竞争方仅包括一个 Pro*C 的 daemon 程序作为 C/S 模式的服务端,和由 Apache PHP 所支持的 WEB 网站业务。 1. 单个 SQL 语句的处理 首先,最简单的情况莫过于单个 SQL 语句的分析,SQL 语句的优化也是数据库优化的一个最直接最立竿见影的因素。SQL 语句的性能监控从监控工具来说大致可分为由高级语言提供和由 ORACLE 本身提供,高级语言以典型的应用 C 语言和 WEB 开发语言 PHP 为例, C 语言中可以用 gettimeofday 函数来在某一数据库

操作之前和之后分别获取一个时间值,将两个时间值之差做为衡量该数据库操作的效率,在 PHP中,也可以用 `gettimeofday`, 操作方法当然与 C 语言中有所不同。当然,PHP中也有其它一些函数可以达到同样的时间精度,关于时间精度的考虑,不能简单以大小衡量微秒级的时间数值。因为时钟中断的时间间隔从根本上决定了时间计算所能达到的精度。此外,操作系统本身对进程的时间片分配,及进程切换的开销等因素也在一定程度上影响时间数据的意义。所以,以下时间的计算最理想的情况是对同一操作在尽可能避免缓存的情况下进行多次的循环操作,取总的时间值加以平均,从而得到比较接近真实情况的时间值。C 语言的例子:

```
#define TV_START 0 #define TV_END 1
int how_long(int cmd, char *res). struct CMD_TIME{ int times. /*
times ocured within specified package number */ struct timeval
time. /* total time consumed by the cmd */ }. void foo() { int id.
how_long(TV_START, NULL). EXEC SQL WHENEVER
SQLERROR CONTINUE. EXEC SQL WHENEVER NOT
FOUND CONTINUE. EXEC SQL 0select user_id into :id from
users where name=slimzhao.2. how_long(TV_END,
time_consume). puts(time_consume). } int how_long(int cmd, char
*res) /* return value: -1 error, 0 sucess , res: 20 bytes is enough */ {
static struct timeval before, after. if(cmd == TV_START) {
gettimeofday(amp.after, NULL). if(res) { if(after.tv_usec >
before.tv_usec) { sprintf(res, "%ld %ld", after.tv_sec - before.tv_sec,
after.tv_usec - before.tv_usec). } else { sprintf(res, "%ld %ld",
after.tv_sec - before.tv_sec - 1, 1000000 after.tv_usec -
before.tv_usec). } } return 0. } else { return -1. } }
```

下面是一个 PHP

的例子(为简化起见,程序的错误检查被忽略) /how_long.inc".
how_long(TV_START, \$timestr). \$conn = OCILogon("username",
"password", "dblink"). \$stmt = OCIParse(\$conn, "0select ID from
users where name=slimzhao"). OCIDefineByName(\$stmt, ID, \$id).
OCIExecute(\$stmt). OCIFetch(\$stmt). OCIFreeStatement(\$stmt).
OCILogoff(\$conn). how_long(TV_END, \$timestr). echo " 用户
ID: \$id , 该操作消耗时间 :\$timestr". ?> 其中 how_long 函数的
PHP 版本如下 : \$after_SQL["usec"]) { \$str = (\$after_SQL["sec"] -
\$before_SQL["sec"] - 1)." 秒 ". (\$after_SQL["usec"] 1000*1000
-\$before_SQL["usec"])." 微秒 ". } else { \$str = (\$after_SQL["sec"] -
\$before_SQL["sec"])." 秒 ".
(\$after_SQL["usec"]-\$before_SQL["usec"])." 微秒 ". } } else { return
-1. } } ?> 100Test 下载频道开通 , 各类考试题目直接下载。详
细请访问 www.100test.com