

地震业务数据库自动监控与管理系统设计实现

丁睿¹ 王小明¹

1 上海市地震局,上海市兰溪路 87 号,200062

摘要: 基于 MVC 的 ThinkPHP 框架,集成 Smarty 模板,采用 PHP+MySQL,PHP+Oracle 双数据库链接原理,研制 B/S 架构的地震业务数据库管理软件。可提供数据库运行信息展示与查询、事故处理分类记录,生成巡检报表,为业务用户提供数据使用状态查询。同时,在 Nagios 开源网管平台中配置业务数据库实时监控模块,将故障报警通过手机短信及时发送,形成完整的地震业务数据库自动监控与管理系统的。

关键词: 地震业务数据库;ThinkPHP;MVC;Nagios;数据库监控

中图分类号: P315.7

文献标识码: A

地震行业 Oracle 数据库系统支持信息、前兆、应急、日常办公等多种应用,已成为地震业务、政务系统运行的基础保障^[1]。本文论述了 B/S 架构的地震业务数据库管理软件和监控报警模块的设计思路与实现方法。

1 系统需求分析

1) B/S 架构的管理方式。采用 C/S 架构的软件,如 PLSQL、TOAD、Lab128 等虽然功能较全、可视化好,却必须安装客户端,对权限要求高,不适合一般业务用户;Oracle 自带的 SQLplus 使用复杂,而 OEM 占用数据库服务器资源,且应用于各自主机,不方便数据库管理员(DBA)在同一平台使用。因此,需要一种零客户端维护、界面友好、信息集中、成本低廉、B/S 架构的数据库管理软件,以满足地震行业数据库运维需求。

2) 差异化的用户服务。DBA 采用 SQLplus 方式管理多个业务数据库时需要依次登录,繁琐耗时;而业务用户查询数据使用情况也采用 SQL 命令方式,其技术要求高且不安全。需要为 DBA 和业务用户提供差异化的用户查询服务。

3) 运维记录与管理。Oracle 数据库管理需要规范运维管理,电子化故障处理记录,并统一保存和浏览,以形成地震业务数据库运维知识库。

4) 实时监控与故障告警。实时监控数据库运行状态,及时获取故障告警。

2 管理软件的设计与实现

地震业务数据库管理软件基于 CentOS 开源

操作系统,应用 ThinkPHP 框架和 Smarty 模板引擎,采用双数据库链接原理(PHP+Oracle 和 PHP+MySQL)设计,实现 B/S 架构的多业务数据库的信息查询、运维与故障处理。

2.1 功能模块设计

地震业务数据库管理软件主要为 DBA 提供数据库运行状态、性能展示与查询、故障处理和巡检报告等功能,同时为业务用户提供数据使用状况查询,如表 1 所示。

2.2 后台管理数据库设计

管理软件后台数据库采用 MySQL,建立 12 张表,包括管理员表(zgxyl_admin_user)、权限表(zgxyl_access)、用户与组表(zgxyl_role_user, zgxyl_group)、故障处理(zgxyl_fault_hand)、巡检报告表(zgxyl_article)、系统设置(zgxyl_sys-set)等。后台数据库表间关联模型如图 1 所示。

后台 MySQL 数据库进行表的创建、更新、删除、字段修改等操作,可以通过部署在软件服务器上的 PHPMyAdmin 数据库管理工具来操作。

2.3 关键技术与实现

2.3.1 双数据库链接应用

PHP(hypertext preprocessor)是一种通用开源脚本语言,适用于 Web 开发。管理软件采用双数据库链接原理,即 PHP+Oracle 和 PHP+MySQL。PHP+Oracle 控制管理软件和业务 Oracle 数据库的对接查询。PHP 支持 Oracle,在 PHP 的扩展功能中包含大量有关链接数据库的函数^[2]。建立一个只具有查询权限的用户,通过

此用户就可获取业务数据库的运行信息,并保护业务数据库不被恶意更改。PHP+MySQL 控制业务用户数据查询模块、运维故障处理模块和用户账户模块与后台管理 MySQL 数据库之间的信息交互。

2.3.2 MVC 模式的软件实现

管理软件研发采用 MVC 模式的 ThinkPHP 框架,并集成 Smarty 模板。ThinkPHP 严格采用 MVC 设计模式,即模型(M)-视图(V)-控制器(C)。MVC 把用户界面设计、流程控制和事务逻辑进行分离,降低模块之间的耦合度^[4]。采用 MVC 模式进行研发,使用者首先在视图的界面上发请求,视图将请求转发给控制器,控制器调用相应的模型进行业务逻辑处理,并返回数据,最后控制器调用相应的视图来显示模型返回的数据。Smarty 是一个 PHP 模板引擎,它分离了逻辑程序和外在的内容^[5],使代码编写与界面设计相互独立。

软件实现过程中,模型(M)的定义由 Model 类来完成,该类在能实现相关数据表的 ORM 和 CURD 操作,需要单独业务逻辑封装时才被定义。管理软件通过定义和实例化,生成了 7 个模型,分别是 AdminUser Model、Common Model、Form Model、GroupModel、NodeModel、RoleModel 和 UserModel,位于项目的\Lib\Model。

控制器(C)分为应用控制器(APP 类)和 Action 控制器。前者负责调度控制,后者完成业务过程。控制器每个模块都是一个 Action 文件,位于项目的 Lib/Action 目录下面,由几十个不同的 Action 文件组成,定义和描述在用户的操作下,管理软件读取、展示业务数据库的运行状态信息、文档记录等一系列过程。例如,查看数据库实例状态的 Action 代码如下:

```
class StatusAction extends CommonAction {
    function index(){
        import("ORG.Util.Oracle");
        global $OracleSys;
        $sql='select * from v$instance';
        $arr = $OracleSys->Oracle_fetch_row($sql);
        //var_dump($arr);
        //import("@.ORG.Util.Page");
        // $this->assign('page', $page);
        $this->assign('list', $arr);
        $this->display();
    }
}
```

视图(V)包括 View 类和模板文件。Action

控制器把要输出的数据通过模板变量赋值的方式传递到视图类,由 View 类来展现。同时 View 类还和模板引擎进行接口,完成布局渲染、输出替换、页面 Trace 等功能。管理软件的视图应用 Smarty 模板引擎,模板的实现和框架无关,可以独立预览和制作。

3 监报告警模块的配置

1) 确定监控数据库。

将上海地震集成观测数据库、前兆数据库和应急数据库纳入监控范围,涵盖测震、强震、前兆和应急业务用户,数据库节点即为独立监控对象。被监控的数据库需要安装如 GCC、Nagios-Plugin 等插件。

2) 性能监控项选择和报警阈值设定。

Nagios Plugins 有很多脚本,其中用于实现对 Oracle 数据库监控的脚本是 check_Oracle_health,可以监控 cache、tns、tablespace 等信息。配置此脚本,确定监控对象如下:

① 地震业务数据库的主机状态(up/down)。

② 数据库实例运行状态(up/down)。

③ 数据库监听(up/down)。

④ 缓存命中率。命中率 = $1 - (\text{physical reads} / (\text{db block gets} + \text{consistent gets}))$,其中,db block gets 为数据请求总数,consistent gets 存取所有满足要求的请求,physical reads 为磁盘上数据文件存取请求的总数。SGA 中共享缓存命中率应该不小于 1%。

⑤ 系统负载 load average,其值应小于“CPU 个数 × 核数 × 0.7”,否则就启动报警。

⑥ 系统远程进程 system process, show parameter process 可以得到数据库允许的进程数,超过则启动报警。

⑦ 表空间使用,包括业务数据用户表空间和系统表空间。使用量超过 80%则启动黄色警告,超过 90%则启动红色警报。

以上监控项目 90 s 轮询一次。图 2 所示为核心观测集成数据库节点 1 中监控对象。

3) 告警短信发送^[6]。

在 Nagios 网管平台上安装飞信客户端和 LIB 库。根据 Naigos 平台数据库模块中设置的监控项告警阈值,一旦达到,即触发短信告警。故障处理完毕,监控项的值恢复正常,再触发短信通知数据库管理员。图 3 所示为一次前兆数据库处理后手机上收到的实例宕机及恢复告警短信。

