钻井数据库集群监测系统的设计与实现

邱 爽,任树华
(大连工业大学 信息科学与工程学院,辽宁 大连 116034)

摘 要:为了及时了解制约整个钻井数据库集群系统在海量数据的OLTP商业应用中执行效率的主要性能瓶颈,有必要实施对集群的管理维护,保证系统性能的优化。因此,论文以钻井数据库集群系统为背景,分析集群系统的体系结构和性能优化,设计并实现了一种基于C#的轮询监测数据库集群系统的体系结构模型,还详细介绍了数据采集、管理以及数据可视化各个模块的具体实现。通过对钻井数据库集群系统进行轮询监测,周期的从节点中采样CPU利用率、网络流量以及各磁盘的状态情况等性能指标,进一步验证了该软件的可用性和有效性。

关键词:真正应用集群;性能优化;轮询监测
中图分类号:TP311 **文献标识码**:A **文章编号**:1673-629X(2011)02-0238-04

Design and Implementation of Drilling Database Cluster Monitoring System

QIU Shuang, REN Shu-hua
(Dept. of Information Science & Engineering, Dalian Polytechnic University, Dalian 116034, China)

Abstract: In order to keep abreast of the major performance bottlenecks in the drilling database cluster system which restrict the system to enhance the efficiency in the OLTP business applications of mass data, it is necessary to implement the management and maintenance of cluster, ensure optimal system performance. Therefore, the paper is on the background of drilling database cluster system, analyzing the cluster system architecture and performance optimization, and designing the architecture model and the implementation strategy of a C sharp-based polling monitoring system. Then the method of data collection, information management and the visualization of node state are introduced in detail. Finally, further verified the availability and effectiveness of the software through that system monitor the drilling database cluster system which sample periodically the performance indicators such as CPU utilization, network traffic and the state of the disk case from each node.

Key words: real application cluster; performance optimizing; polling mointor

0 引 言

某油田数据中心服务器系统运行着生产调度、地质设计等多个数据库系统,随着数据库中数据量的迅速增长,数据操作越来越复杂。因此,如何高效地管理和监控集群系统、及时了解各个节点的运行状况越来越受到广大数据库管理员(DBA)的关注和重视。正如IDC的研究报告中指出的那样,集群系统建设的"最大挑战来自集群系统的管理的软件的复杂性"。可见对集群系统各个节点的监视和控制在集群服务器中显得尤为重要。

目前,国内外许多研究机构都设计和开发了集群监控系统,如Ganglia、Negios、Hawkeye、DCMM等,来帮助DBA对集群系统的性能进行调整和优化,但精通这些工具并能通过它们来合理地分析集群性能状态,进而有效分配资源以优化集群性能也十分困难。针对这种情况,结合业界先进的数据库集群管理经验,设计一种基于C#的简单、直观的集群轮询检测平台,周期轮询地从节点中采样性能数据,实时地得到各节点的CPU利用率、网络流量情况、节点负载情况和各磁盘的详细信息等,从而帮助DBA及时发现和解决服务器的性能瓶颈,保证整个集群的高性能运行。

1 总体设计

1.1 集群系统的性能优化

油田数据中心服务器系统的数据库管理着生产调度、地质设计等各个方面的关键数据,因此保证数据库安全、稳定和高效的运行极为重要。为此,钻井数据库集群系统采用Oracle真正应用集群体系架构,自动适

收稿日期:2010-06-05;**修回日期**:2010-09-16
基金项目:大连市政府IT优秀教师专项资金(大信发[2008]39、40号)
作者简介:邱 爽(1986-),女,硕士,研究方向为数据库集群系统;任树华,硕士生导师,研究方向为企业信息化、web挖掘等。

应快速变化的业务需求和发生的负载变化[1]。Oracle RAC 在负载均衡、高可用性、自动存储管理和安全性能等方面不仅能够全面满足油田数据中心服务器系统的需求,同时提供了灵活的、可以充分保护已有投资的可扩展性,大大提高了系统的性能和用户访问效率。

钻井数据库集群系统优化的目标是提高 CPU 的利用率,缩短系统响应时间,达到性能的线性扩展。对 Oracle RAC 来说,优化时进行合理的资源配置,可达到节点间的负载均衡,改善性能,增加吞吐量和提高响应时间。而对 Oracle RAC 的性能调优,首先应从调整单个节点入手,然后扩到整个 RAC,整个过程采用 Down-Top 策略,先从单个节点考虑可能出现问题的环节,如 CPU 利用率、系统负载、各磁盘的详细信息等[2]。在完成了对节点的调整优化后,再查看影响 Oracle RAC 的全局参数,I/O 布局、网络传输信息以及 RAC 特有的相关调节等[3]。因此,文中针对单节点可能出现的问题,设计一个基于 C#的轮询监测系统,周期轮询的从各节点中采样性能数据,将影响性能的关键因素实时的反应给用户,以达到提高钻井数据库集群系统性能优化的目的。

1.2　结构模型

基于 C#的轮询监测系统结构模型如图 1 所示。

基本算法如下:

1)IP=输入值;

2)向指定 IP 地址的节点发送连接请求;

3)生成一个 Socket;

4)节点接受,建立连接,ServerDaemon 驻留节点,通过/proc 采集性能指标;

5)向 Oracle 数据库服务器发送采集结果,终止 ServerDaemon;

6)Oracle 服务器收到数据,存储数据;

7)监测终端通过定时轮询机制从 Oracle 数据库服务器提取性能指标信息;

8)加工(统计分析)和重组(排序)数据,调用业务逻辑层处理;

9)用户界面层加载数据显示给用户,借以帮助用户理解影响集群系统的性能指标。

轮询监测系统共可划分为如下四个模块:命令控制模块,供用户发送监视命令、控制监视行为;数据采集模块,负责获取集群节点性能数据;数据存储模块,通过封装简化对性能数据库的操作;数据展示模块,提供分析结果。

如图 1 可以看出,客户端发起查询请求,数据库服务器解析后进行数据采集,而每个节点上都有一个监测进程负责从/proc 文件系统中收集性能数据和故障信息,再由数据采集模块定时采集这些信息,然后将采集的数据按照协议解析和安全性检查与完备化后存入 Oracle 数据库。最后从 Oracle 数据库服务器中读取信息,并反馈到客户端,以图形界面的形式显示出来。为了减少通信资源的占用,客户端采用 socket 技术实现远程通信功能[4],提高轮询监测系统的实时性。

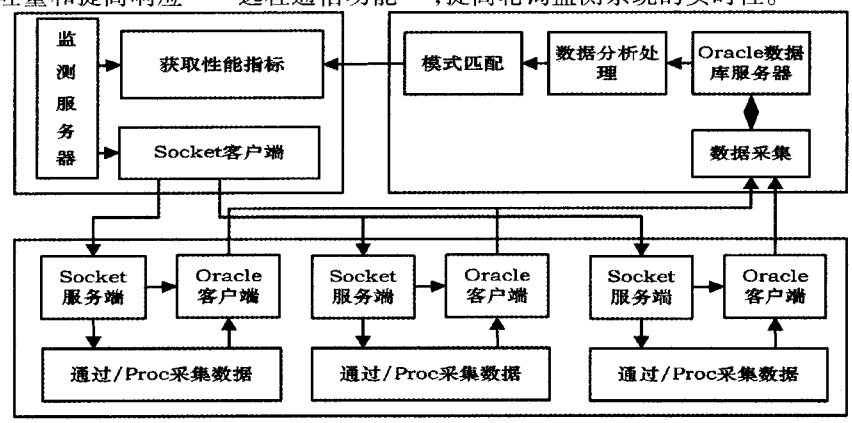

图 1　结构模型图

1.2.1　数据采集

采集数据的实现方法主要通过/proc 虚拟文件系统来实现[5]。考虑到 C 语言同 shell 之间的方便通信,信息和控制数据可以作为命令行上的参数传递给 C 语言。因此,该模块采用 C 语言实现,编译器 gcc 版本为 4.2.1。该模块的实现主要是对/proc 文件的解析,代码实现比较简单,大量的工作是对/proc 文件的解析上。

在 Linux 系统中,/proc 目录下用文件的形式保存了系统的负载情况。守护进程根据客户端提供的集群节点 IP,定期访问指定节点/proc 目录中的相应文件,经过计算,得到该节点的系统资源利用情况,包括 CPU 利用率、内存页和内存交换信息、磁盘可用率、系统负载以及各磁盘的详细信息等[6]。下面以获取系统负载为例做一简要说明。

```
[root@ rac1 proc]# cat loadavg
1.00 1.00 1.00 1/216 4829
```

从上面的输出可以看出,文件内容共 5 个数据,前三个是通过计算运行队列中的平均任务数得到的过去 1 分钟、5 分钟和 15 分钟的系统平均负载,如果除以 CPU 的数目高于 5,则表明系统在超负荷运转。第四个是正在运行的任务数和总任务数,最后是上次使用的进程号。根据节点系统负载平衡可以大体上了解系统的负载,由此来比较各节点从而判断整个系统的负载情况。其算法如下[7]:

文件 include/linux/sched.h:

```
#define FSHIFT 11 / * nr of bits of precision * /
#define FIXED_1 (1<#define LOAD_FREQ (5 * HZ) /
* 5 sec intervals * /
#define EXP_1 1884 / * 1/exp(5sec/1min) as fixed-
point, 2048/pow(exp(1), 5.0/60) * /
#define EXP_5 2014 / * 1/exp(5sec/5min), 2048/pow
(exp(1), 5.0/300) * /
#define EXP_15 2037 / * 1/exp(5sec/15min), 2048/
pow(exp(1), 5.0/900) * /
#define CALC_LOAD(load,exp,n) \
load * = exp; \
load += n * (FIXED_1-exp); \
load >>= FSHIFT;
```

读取/proc/loadavg 文件的代码如下：

```
void SearchLoadavg(float * one,float * five,float *
fifteen)
{
char lonemin[12],lfivemin[12],lfifteenmin[12],
queue[12],str[12], * p;
int i=0;
FILE * fp=fopen("/proc/loadavg","r")
fscanf(fp,"%s %s %s %s",lonemin,lfivemin,lfif-
teenmin);
fclose(fp);
double * one=atof(lonemin);
double * five=atof(lfivemin);
double * fifteen=atof(lfifteen);
do
{
str[i]=queue[i];/ * 存放负载信息 * /
i++;
}
while(queue[i]=='/');
}
```

1.2.2 数据管理

数据管理部分是整个系统的关键，完成集群系统中性能数据和故障信息的采集工作，同时把采集到的数据通过网络保存到 Oracle 数据库服务器。其主要解决以下问题：

(1)完成数据的接收工作；

(2)解析上述接收到的监控信息，这些信息均采用 Hash 表数据结构进行存储。因此，该模块只解析此结构中的信息即可得到单个节点的监控信息；

(3)将解析所得的静态监控信息存放到 Oracle 数据库服务器；

(4)动态监控信息与从数据库中取得的预先设定的阈值进行比较，将超过阈值的信息生成报警/预警信息。

该模块主要使用. NET Framwork 异步编程模型处理网络服务请求，首先定义 Receiver( )开始清理资源、创建 Oracle 数据库的连接、初始化性能参数[8]，然后创建服务器端侦听 Socket 对象，调用静态方法 ThreadPool. QueueUserWorkItem( )在线程池中创建核心处理线程。服务器端循环等待客户端连接请求，一旦有请求，检测该客户端 IP 地址，一切正常后建立 TSession 对象，并调用异步方法接收客户端 Socket 数据包[9]。代码中，Socket 读到数据时的回调 AsyncCallback 委托方法 EndReceiveData( )即完成数据的接收工作[10]。进而将接收的监控信息采用 Hash 表数据结构进行存储，此 Hash 表数据结构是由现成的 APR 软件包提供的，省去重复开发的时间。Hash 表数据结构因为是常驻内存空间中，从其读取数据或向其写数据时效率更高，减少系统资源，而且占有较少的内存。最后，由终端运行的守护程序与指定的集群节点建立 TCP 连接，周期轮询的从 Hash 表中获取监控数据，将能表征节点性能状态的相关指标通过网络保存在 Oracle 数据库服务器中[11]。如果动态监控数据超过从数据库服务器中取得的预先设定的阈值，则生成报警信息提示给数据库管理员。

1.2.3 数据显示

该模块主要负责从 Oracle 数据库中读出数据，并根据专家库进行分析，判断系统中存在的潜在故障和错误，通过 Socket 网络通信，将数据传输到监测终端，经过监测终端的相应计算与处理[12]，调用绘图类将这些性能参数以图形化的方式显示在终端界面上，出错则给出报警信息。图形界面按节点分别显示 CPU 利用率、网卡流量、负载状况、各磁盘的使用情况等信息。

显示界面主要有主界面和异常退出两部分，其中主界面是数据显示的核心界面。

异常退出对话框是在监视终端在退出时由于网络连接异常导致退出失败时弹出的对话框。

主界面是对节点运行情况的一个直观的显示界面，它显示了该客户端所连接监控服务器的性能指标。其中流量信息显示了用于心跳和对外提供服务的两块网卡流量信息；CPU 监视信息提供了用户模式(user)下 CPU 的消耗；负载信息则是通过计算运行队列中的平均任务数得到的过去 1 分钟、5 分钟和 15 分钟的系统平均负载；磁盘信息则显示了本地及共享磁盘的当前已用磁盘空间。

2 测试与分析

实验环境为由三台 PC 服务器组成的 Oracle RAC

集群系统,QNAP NAS 做共享存储,在钻井数据库集群系统上对设计的轮询监测系统进行测试,集群节点配置如下:CPU 为双 Intel(R)Quad 2.50Hz,内存为 4GB,硬盘容量为 500GB,网络为 100Mb/s 以太网,心跳网络为 1000Mb/s,节点操作系统为 RHEL5.4,数据库为 Oracle Database 11gR2。通过在被监控节点上运行 ServerDaemon 守护进程后,即可在监控服务器上查看相应的监控信息,监测结果按节点显示各种资源的信息。表 1 为输入指定 IP 地址的节点性能参数,包括 CPU 利用率,网络流量以及各磁盘的状态情况。

表 1　节点性能参数

	当前值	最大值	平均值
CPU 利用率(%)	25.02	25.03	25.00
eth0 流量(流入)	2.97	3.90	2.85
eth1 流量(流入)	9.15	9.21	9.01
disk1 状态(G)	94.47	94.48	94.47
disk2 状态(G)	38.69	38.72	38.70

3　结束语

研究了轮询监测钻井数据库集群系统的开发方法,搭建了一种简单而有效的监测平台,并验证了该软件的可用性和有效性。

文中提出的监测系统实现了两级的异步通信。一是节点数据的获取,通过数据采集模块的定时信息采集,只需从 Oracle 数据库中读取监视信息;二是客户端采用.NET Framwork 异步编程模型处理网络服务请求,隐藏了大数据量的传输时间。两级的异步操作使得对于钻井数据库集群系统的实时监测成为可能。考虑到安全性,该系统没有对集群进行控制的功能,所以有交互性差的缺点。

参考文献:

[1] 谷长勇,王　彬,陈　杰. Oracle 11g 权威指南[M]. 北京:电子工业出版社,2008.

[2] 孙凤栋,闫海珍. Oracle 10g 数据库系统性能优化与调整[J]. 计算机技术与发展,2009,19(2):84-86.

[3] Antognini C. Troubleshooting Oracle Performance[M]. [s. l.]:Apress,2008:29-35.

[4] 于　涛,王　健. 基于 Socket 通讯技术的上层监控软件的实现[J]. 计算机技术与发展,2009,19(3):244 - 245.

[5] 李东亮,王海花. 基于/proc 文件系统及对内核信息的获取[J]. 河北工程大学学报:自然科学版,2007,24(2):74-77.

[6] 廖家建. 集群监控中的数据采集技术研究[D]. 武汉:华中科技大学,2008:31-38.

[7] 牛　峰,胡昌振. 内核信息获取的通信方法[J]. 计算机工程,2003,29(8):114-115.

[8] Manning P, Gennick J. Pro ODP . NET for Oracle Database 11g RAC[M]. [s. l.]:Apress,2010:54-62.

[9] Liu Ling, Özsu M T. Encyclopedia of Database Systems[M]. US:Springer, 2009:2093 - 2094.

[10] 黄光芳. 基于.NET 的数据访问基类的构建[J]. 计算机技术与发展,2008,18(3): 149-150.

[11] 贺致智,王　晖. 瘦客户端的 Oracle 数据库连接技术研究[J]. 计算机技术与发展,2006,16(7):8-9.

[12] 郭东升,田秀华. Linux 环境下基于 Socket 的网络通信[J]. 软件导刊,2009, 8(1): 117- 118.

(上接第 237 页)

参考文献:

[1] 张凌晓,刘克成. 基于 UML 的全程办税系统的建模与实现[J]. 计算机技术与发展,2008,18(10):211-213.

[2] 孙志杰,卢　雷. 基于数据库中间件技术的 WEB 应用数据访问的实现[J]. 计算机系统应用,2008(1):87-90.

[3] 戴文娟,王晓峰. 基于 XML 和 BizTalk 数据集成平台的设计与构建[J]. 计算机技术与发展,2008,18(10):162-165.

[4] 洪　玫,刘　佳. 基于服务数据对象的异构系统数据集成方案研究[J]. 计算机应用,2007(6):21-23.

[5] 王艳华,薛胜军,蒲秋梅. 公积金监管系统中的多级数据集成研究[J]. 武汉理工大学学报:交通科学与工程版,2007(3):544-547.

[6] 舒清录. 基于.NET 的异构数据源数据迁移技术[J]. 计算机技术与发展,2010,20(3):109-112.

[7] Bao Tie, Liu Shufen. A Method for Network Data Collection and Processing in the Pervasive Computing Environment [C]//Proc. of International Symposium on Pervasive Computing and Applications. [s. l.]: IEEE Press, 2006: 599-603.

[8] Kim Do-Hyeon, Kang Kyung-Woo. Design and Implementation of Integrated Information System for Monitoring Resources in Grid Computing[C]//Proc. of the 10th International Conference on Computer Supported Cooperative Work in Design. Nanjing, China: IEEE Press, 2006.

[9] Pan Zhangsheng, Chen Xiaowu, Ji Xiangyu. Research on database access and integration in UDMGrid[C]//Parallel and Distributed Processing and Applications - ISPA2005 Workshops. Berlin:Springer-Verlag,2005:496-505.

[10] Saleh K, Probert R, Khanafer H. The distributed object computing paradigm:concepts and applications[J]. Journal of Systems and Software,1999,47(2):125-131.

[11] 姜坚华,饶若楠. 用基于 JDBC 的中间件实现铁道部 DMIS 系统的异构分布式数据库访问[J]. 计算机应用与软件,2004,21(12):25-27.

[12] 江凤莲,邓书显. 基于工厂模式的跨平台数据库访问中间件[J]. 商丘师范学院学报,2008,24(3):85-87.

[13] 林伟伟,齐德昱,李拥军. 基于网格的分布式异构数据集成模型[J]. 计算机工程,2006,32(24):48-49.