

自适应大规模服务器集群监控系统的构建

薛正华¹, 董小社¹, 伍卫国¹, 胡雷钧²

(1. 西安交通大学电子与信息工程学院, 710049, 西安; 2. 浪潮电子信息产业股份有限公司, 100085, 北京)

摘要: 针对大规模服务器集群监控系统域大小固定而不能动态适应集群规模变化的缺点, 提出了自适应动态域算法. 通过计算监控引入的负载, 根据负载容忍度要求, 动态改变域的个数和大小, 以适应集群规模变化. 同时, 提出了浮动域监控节点的方法, 以应对域监控节点负载过重的情况, 即在域监控节点负载较重时, 自适应地选择新的域监控节点, 并完成节点角色的自动更新. 采用自适应的线程池和数据库连接池可降低监控任务的开销, 加快执行任务的响应时间. 监控 CPU 利用率的测试结果表明, 系统执行单任务的响应时间为 128 ms, 20 个任务的响应时间为 242 ms; 数据库连接池的性能测试结果表明, 在 300 s 内, 采用了自适应池、不采用自适应池方案所耗处理器资源分别为 1.80% 和 7.53%, 处理任务数分别为 12 364 和 2 769.

关键词: 服务器集群; 监控系统; 自适应池

中图分类号: TP391.9 **文献标识码:** A **文章编号:** 0253-987X(2008)04-0399-05

Design and Implementation of an Adaptive Monitoring System for Large-Scale Server Clusters

XUE Zhenghua¹, DONG Xiaoshe¹, WU Weiguo¹, HU Leijun²

(1. School of Electronics and Information Engineering, Xi'an Jiaotong University, Xi'an 710049, China;

2. Langchao Electronic Information Industry Co., Ltd., Beijing 100085, China)

Abstract: An adaptive dynamic domain algorithm has been proposed to solve the issue that the fixed domain can not adapt to the variation of the cluster scale. By computing the intrusive load obtained by the monitoring system, the algorithm can control the numbers and the sizes of monitoring domains in an acceptable level according to the intrusion tolerance degree. The fixed domain monitoring node is not efficient as its load is high. A floating domain monitoring node method has been proposed to re-select the domain monitoring node and automatically update the roles of the nodes when the load of the node is high. To reduce the overhead incurred by monitoring system and improve the response time of executing monitoring tasks, an adaptive thread pool and a database connection pool have been designed. The measurements on monitoring CPU usage show that the response time of monitoring one node is 128ms, and that of monitoring 20 nodes is 242ms. A comparative performance measurement has been performed between the solutions with an adaptive database connection pool and that without a pool. The results show that the average intrusion on CPU usage in 300 seconds is 1.80% with a pool and 7.53% without a pool, and the numbers of completed tasks are 12 364 and 2 769, respectively.

Keywords: server cluster; monitoring system; adaptive pool

高性能服务器集群由于其良好的可扩展性、容错性和高性价比而成为目前最为广泛使用的计算基

基础设施.随着计算需求的不断增加,集群规模从过去的数百个节点扩充到了现在的数千、数万个节点^[1],如何快捷、有效地监控大规模集群中各节点的健康状况、性能指标和配置信息,已成为一项挑战性的工作.当集群规模较大时,传统的集中式集群监控系统^[2-3]的中央监控节点将成为系统的瓶颈.为了解决这一问题,一些系统^[1, 4-5]采用基于多域的监控架构,即将系统分为多个域,并在域内设定了一个域监控节点,监控本域内各节点的信息,分摊中央监控节点的负载.这种系统,其域的个数、大小(即每个域内的节点数)以及域内的监控节点都是固定的,而且依经验设定,所以不适应于集群规模和负载的动态变化.基于此,本文研究并实现了具有自适应特性的大规模服务器集群监控系统,它可根据集群规模的变化动态地改变域的个数和大小,根据域监控节点的负载,自适应浮动域监控节点.此外,设计并实现了自适应的线程池和数据库连接池,以降低监控系统引入的开销,减小系统的响应时间.

1 自适应监控系统结构

基于多域的集群监控系统将集群从逻辑上划分为若干个域,各域设定了一个域监控节点,其上驻留了一个域监控代理,负责监控本域内的节点,并执行来自集群监控节点的命令.在每个计算节点上均驻留着一个节点监控代理,它是一个被动实体,执行域监控节点的命令.

1.1 自适应动态域

当系统增加或减少节点时,域内的节点数目将发生变化.若节点增加得过多,域监控节点会因负载过重而成为瓶颈;若节点减少得过多,一些域就失去了存在的意义.为此,本文提出了一种自适应动态域方案.

设系统节点数目为 N (不包括集群管理节点和数据库服务器),域的个数为 P ,域的大小为 Q ,则

$$Q = \left\{ \left\lfloor \frac{N}{P} \right\rfloor, \left\lceil \frac{N}{P} \right\rceil \right\} \quad (1)$$

其中,一部分域的大小为 $\left\lfloor \frac{N}{P} \right\rfloor$,另一部分域的大小为 $\left\lceil \frac{N}{P} \right\rceil$.

监控系统引入的负载主要在节点处理器的利用率和网络流量方面.定义域监控节点在执行监控任务时引入的负载与域的大小满足如下关系

$$L_{\text{CPU}} = f_{\text{CPU}}(Q) \Leftrightarrow Q = f'_{\text{CPU}}(L_{\text{CPU}}) \quad (2)$$

$$L_{\text{Net}} = f_{\text{Net}}(Q) \Leftrightarrow Q = f'_{\text{Net}}(L_{\text{Net}}) \quad (3)$$

式(2)、式(3)中: L_{CPU} 、 L_{Net} 分别表示监控引入的CPU负载和网络流量; $f_{\text{CPU}}(Q)$ 和 $f_{\text{Net}}(Q)$ 与节点的硬件、操作系统类型以及监控信息收集方法等因素相关,可通过实际测试确定.

定义系统监控引入的负载容忍区间为

$$T_{\text{CPU}} \subset [L_{\text{minCPU}}, L_{\text{maxCPU}}]$$

$$T_{\text{Net}} \subset [L_{\text{minNet}}, L_{\text{maxNet}}]$$

该容忍区间规定了监控引入的处理器利用率和网络流量负载的上下限.设系统规模变化后的节点数为 N' ,域的个数为 P' ,域大小为 Q' ,则自适应动态域机制算法的步骤如下.

步骤1: 集群监控节点侦测集群系统规模,若系统新增节点,进入步骤2;若系统减少节点,进入步骤3.

步骤2: 将 N' 代入式(1)求得 Q' ,若 $Q' \leq \min(f'_{\text{CPU}}(L_{\text{maxCPU}}), f'_{\text{Net}}(L_{\text{maxNet}}))$,将新增节点均分给各个域,各域新增节点数为 $\left\lceil \frac{N-N'}{Q} \right\rceil$ 或者为 $\left\lfloor \frac{N-N'}{Q} \right\rfloor$;若 $Q' > \min(f'_{\text{CPU}}(L_{\text{maxCPU}}), f'_{\text{Net}}(L_{\text{maxNet}}))$,增加域,域的大小为 Q ,新增域的个数为 $\left\lceil \frac{N-N'}{Q} \right\rceil$.

步骤3: 若 $N' \leq \min(f'_{\text{CPU}}(L_{\text{minCPU}}), f'_{\text{Net}}(L_{\text{minNet}}))$,进入步骤4;若 $\min(f'_{\text{CPU}}(L_{\text{minCPU}}), f'_{\text{Net}}(L_{\text{minNet}})) < N' < N$,将 N' 代入式(1)求得 Q' ,如果 $Q' < \min(f'_{\text{CPU}}(L_{\text{minCPU}}), f'_{\text{Net}}(L_{\text{minNet}}))$ 合并域,各域大小为 Q ,域的个数为 $\left\lceil \frac{N'}{Q} \right\rceil$.

步骤4: 取消域,集群监控节点直接监控所有的节点.

当集群规模发生变化时,集群监控节点根据动态域算法自动构建新的监控拓扑域,拓扑域将指定节点角色(是否是域监控节点)以及每个域监控节点所辖的所有计算节点ID,拓扑域信息将保存到数据库中的系统拓扑数据表中.集群监控系统根据数据表信息,更新相关节点的角色属性,并启/停驻留在相关节点的域监控代理.启动的域监控代理,从系统拓扑数据表中读取其所辖域内的节点ID列表,并开始监控,这样即可快速完成域的自动更新.

1.2 浮动域监控节点

在基于域的集群监控系统中,各域监控节点本身也是计算节点,承担一定的计算任务.当域监控节点负载较重时,监控任务会因得不到系统资源而受

阻,该节点所负责的域内所有节点将处于失控状态.本文采用域监控节点的浮动机制解决此问题,其工作流程如图 1 所示,描述如下.

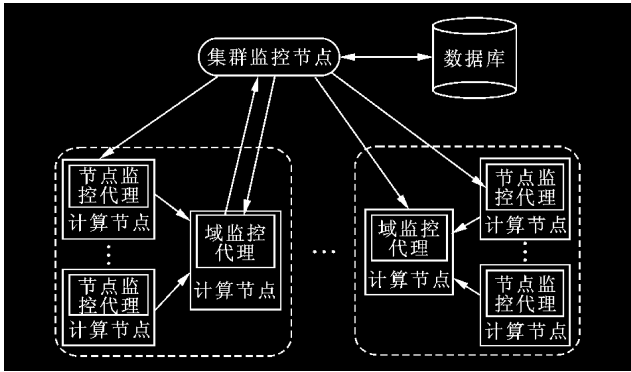


图 1 浮动域监控节点工作流程

- (1)域监控节点若感知到计算任务较重,将自动产生具有高优先级的请求进程,并向集群监控节点发送迁移域监控代理请求,此时该域为请求域.
- (2)集群监控节点在数据库中查询,若发现在最近的一段时间内请求域存在轻载节点,则将该轻载节点选定为新的域监控节点,并将拓扑数据表中该节点的角色更改为域监控节点,同时设置该域内的所有节点为其辖内节点,而将请求域监控节点的角色更改为计算节点;否则,查找邻域内的轻载节点.若邻域内有轻载节点,将拓扑数据表中该轻载节点的角色更改为域监控节点,同时设置请求域内所有节点为其辖内节点,而将请求域监控节点的角色更改为计算节点并设置为邻域的辖内节点(维持域大小均衡);否则,拒绝域监控节点请求,结束流程.
- (3)集群监控节点关闭请求域内的域监控节点代理.若在请求域内选定了轻载节点,启动轻载节点的域监控代理,否则启动邻域内轻载节点的域监控代理并作为请求域的新的域监控节点.
- (4)集群监控节点重启邻域中的域监控节点,启动时从更新后的系统拓扑数据表中读取辖内节点 ID,并监控这些节点.此步骤使得邻域的域监控节点不再管辖浮动到请求域中的节点,同时接管了请求域的监控节点.

2 自适应池机制

在基于域的集群监控系统中,域监控节点需同时对多个节点的信息进行周期性监控.在大规模集群环境中,被监控的节点数较多,监控任务具有周期性,因此在域监控节点端,监控任务呈现出数量多、

生命期短、频率高的特点. 据此,本文采用自适应线程池机制解决此问题,核心思想是线程可复用并按需构建.

2.1 主从式线程池

主从式自适应线程池包括了一个主线程和若干个执行线程. 执行线程是一个被动实体,它执行主线程分配的监控任务,而主线程充当管理者,其工作包括:①按一定的策略将监控任务分派给空闲的执行线程;②定期检查各线程的状态(工作、空闲或死亡),捕获线程抛出的异常,记录空闲线程的空闲时间,根据相应事件设置线程状态标记(如给线程分配任务,并将状态标记为工作;线程完成任务,将状态标记为空闲;线程异常退出,删掉线程的相关记录);③设置线程优先级,创建和销毁线程.

2.2 自适应池机制

线程池的参数描述如表 1 所示.

表 1 描述线程池的参数及其含义

参数名	含义
poolSize	池中的线程数目
idleSize	池中空闲线程数目
queueSize	阻塞队列中的任务数
maxSize	池容纳的最大线程数
minIdleSize	池维护的最小空闲线程数
maxIdleTime	空闲线程的最大空闲时间
maxQueue	无可用线程时等待任务的最大队列长度

表中,maxSize 参数限制了线程池创建线程的数量,以控制域监控节点的开销. 池中至少需维护 minIdleSize 个线程,确保任务快速响应. 在完成任任务之后,线程不能立刻释放,而是存于池中等待接受新的任务,以此避免频繁创建和销毁线程带来开销. 若空闲线程的空闲时间超出了 maxIdleTime,该空闲线程应销毁以节省系统资源. 基于自适应池机制的算法步骤描述如下.

- 步骤 1: 主线程侦听各种事件,若新任务到达,下转步骤 2;若任务结束,下转步骤 3;若空闲线程超过 maxIdleTime,下转步骤 4.
- 步骤 2: 若池内有空闲线程,选用空闲时间最短的线程执行新任务,并将该线程的状态记为 busy,同时记录其起始工作时间;若池内无空闲线程且 poolSize 参数值小于 maxSize,创建新线程,将任务交给新线程,并置新线程状态记为 busy,同时记录

其起始工作时间;若池内无空闲线程且 poolSize 参数值大于或等于 maxSize、queueSize 参数值小于 maxQueue,将任务置入等待队列;若 queueSize 参数值大于或等于 maxQueue,丢弃任务。

步骤3: 将任务结束的线程记为 idle,并记录该线程开始空闲的时间。

步骤4: 若 idleSize 参数值大于或等于 minIdleSize,销毁该空闲线程。

域监控节点在收集到监控信息之后,要将这些信息写入数据库,为历史查询和性能分析所用。该操作需频繁创建和释放数据库连接,同样会产生性能开销和响应延迟。系统采用与线程池相同的自适应池算法来构建数据库连接池,可缓解性能开销和响应延迟问题。

3 测试与分析

图2为本文开发的集群管理监控系统。



图2 集群管理监控系统

3.1 测试环境

在西安交通大学国家高性能计算中心的浪潮天梭集群上对本文集群监控系统进行了测试。节点配置:处理器为 2-way Intel Xeon 3.0 GHz,内存为 2 GB,硬盘为 73 GB(SCSI),网络为 100 Mb/s 以太网,节点操作系统为 RedHat AS3.0 (2.4.20-8),数据库为 MySQL-Server-4.1.12。

3.2 测试结果与分析

对节点的 CPU 利用率、内存利用率、网络流量、平均负载和页面调入调出次数进行了监控和信息采集,当信息采集周期为 5 s 并对不同数目的节点进行监控时,监控引入负载的测试结果如图3、图4所示。

由测试结果知,域监控节点引入的负载随着域内节点数目的增多而不断增长,因此根据集群规模的变化,动态改变域的大小是必要的。依据测试数

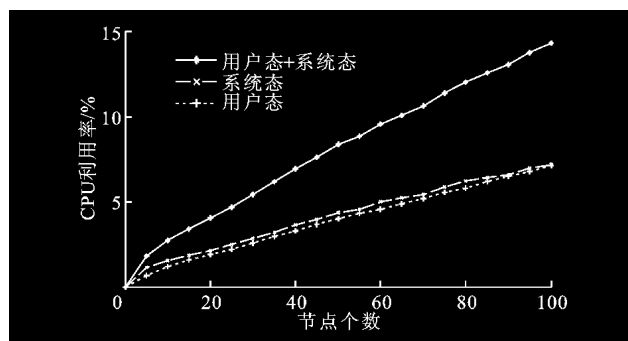


图3 监控引入的CPU负载

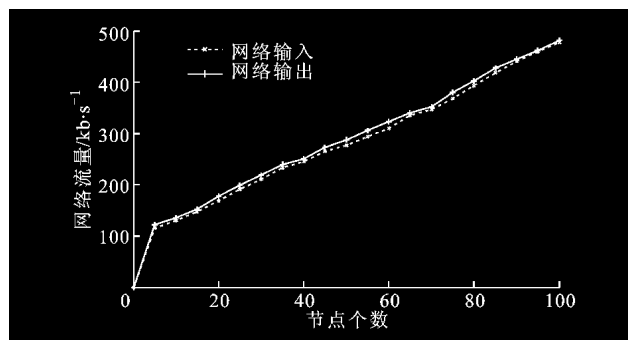


图4 监控引入的网络负载

据,采用曲线拟合技术可得到资源利用率与节点个数的函数关系式。本文采用 Matlab 数据分析工具包提供的 polyval 函数对所测数据进行拟合,并用 polyfit 函数对拟合所得函数进行评测。对图3和图4所测数据的拟合式为

$$y_{\text{CPU}} = -0.0017x^2 + 0.29x + 0.1697 \quad (4)$$

$$y_{\text{Net}} = -1.1x^3 - 182.4x^2 + 16211.7x + 85565.69 \quad (5)$$

式(4)、(5)中: x 表示域内节点数目; y_{CPU} 和 y_{Net} 分别表示监控引入的处理器负载和网络流量负载。

假设集群系统对监控引入的 CPU 开销的容忍区间为 $[4\%, 10\%]$,依据式(4)可得域的大小区间为 $[20, 60]$;假设引入的网络流量负载的容忍区间为 $[512, 1024]$ kb/s,依据式(5)可得域的大小(个数)区间为 $[40, 104]$ 。根据容忍区间的要求,当集群系统规模变化时,监控系统将根据自适应动态域算法自动调整域大小,以确保域监控节点的负载在可控范围内高效执行监控任务。

对监控系统的响应时间进行了测试,该响应时间为域监控代理开始执行监控任务到监控信息写入数据库的时间。以监控各节点的 CPU 利用率为例,测试了并发执行 1、5、10、15 和 20 个节点的响应时间,结果如图5所示。

测试结果表明,系统执行单个任务的响应时间

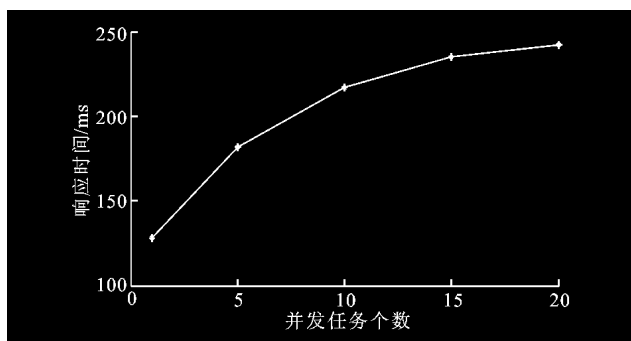


图5 系统响应时间

为 128 ms,若采用基于自适应池的并发机制,当任务数增多时,响应时间大大缩短,如并发执行 20 个任务的响应时间为 242 ms,系统的响应时间随节点数目的增加未呈线性增长态势。

以数据库连接池为例,测试分析了自适应池的系统开销和吞吐能力.测试场景为:客户端不断生成从数据库查询节点 CPU 利用率请求,系统接受请求并将结果返回给客户端.在同一时间段内,采用及未采用自适应池的 2 种方案的系统开销和吞吐能力如图 6、图 7 所示.从中看出,在同一时间段内,使用同一客户端的自适应池方案所引入的系统负载明显低于不采用池方案的结果,且前者处理任务的个数也明显高于后者.如,在 300 s 内,2 种方案的处理器资源分别为 1.80%、7.53%,处理任务的个数分别为 12 364 和 2 769.

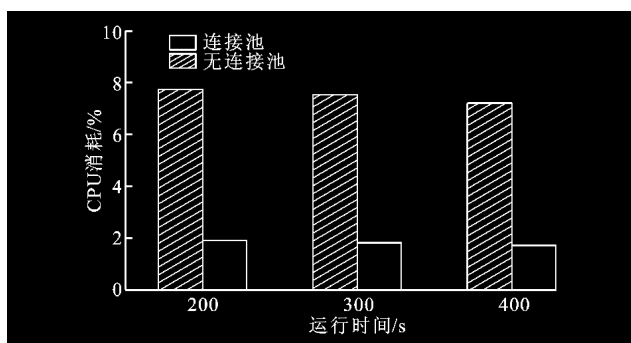


图6 系统开销

4 结 论

本文设计并实现了一种具有自适应特性的大规模服务器集群监控系统,系统具有弹性体系结构,可根据集群规模的变化动态改变域的个数和大小,同时可根据域监控节点负载,自适应地浮动域监控节

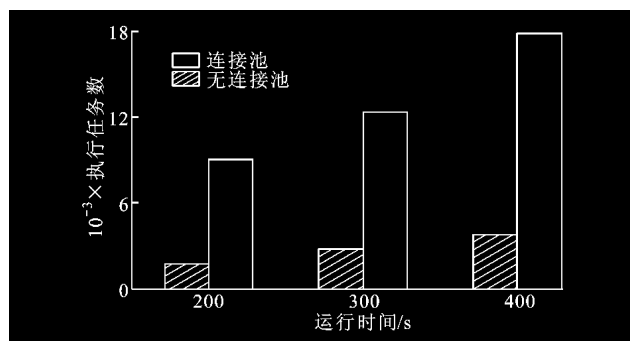


图7 吞吐能力

点.此外,设计了自适应线程池和数据库连接池,使得线程与数据库连接按需构建,从而降低了执行监控任务的开销,加快了执行任务的响应时间.对本文系统的测试结果表明,自适应池机制能够减少系统执行并发任务的响应时间,降低监控系统引入的负载,提高系统的吞吐能力.本文的部分研究成果已集成到浪潮天梭 30000 高性能服务器集群管理监控系统中,该系统已应用于某些大规模集群的管理监控。

参考文献:

- [1] ALMASI G, BACHEGA L, BELLOFATTO R, et al. System management in the bluegene/L supercomputer [C]//Parallel and Distributed Processing Symposium. Los Alamitos, USA: IEEE Computer Society, 2003: 1-8.
- [2] BUYYA R. PARMON: a portable and scalable monitoring system for cluster [J]. Software Practice and Experience, 2000, 30(7): 723-739.
- [3] SOTTILE M J, MINNICH R G. Supermon: a high-speed cluster monitoring system [C]//2002 IEEE International Conference on Cluster Computing. Piscataway, USA: IEEE, 2002: 39-46.
- [4] MASSIE M L, CHUN B N, CULLER D E. The ganglia distributed monitoring system: design, implementation, and experience [J]. Parallel Computing, 2004, 30(7): 817-840.
- [5] TURCK F D, VANHASTEL S, VOLCKAERT B, et al. A generic middleware-based platform for scalable cluster computing [J]. Future Generation Computer Systems, 2002, 18(4): 549-560.

(编辑 苗凌)