

# 基于组合定权法的负载评价模型研究

张 军,李泽平,王苍灵

(贵州大学 计算机科学与技术学院,贵州 贵阳 550025)

**摘 要:**随着互联网技术的快速发展以及海量数据的几何级增长,分布式存储等相关技术得到广泛应用。其中分布式文件系统的应用为海量文件存储和管理等提供了解决方案。当前主流的分布式文件系统,如 Ceph、GlusterFS、MogileFS 等因为各自的特点而得到广泛应用。针对评价开源分布式文件系统 FastDFS 负载参数指标单一,无法实时准确反映系统中各服务器实际工作负载状态这一问题,在研究现有负载均衡技术的基础上,提出了一种基于多指标的负载评价模型。该模型既考虑了主观因素,又考虑了客观因素,运用组合定权法确定各指标权重,使得模型的权重系数能够准确反映 FastDFS 在不同应用场景下各服务器的真实负载。仿真实验结果表明,所提出的评价模型是有效可行的,既实现了负载的合理分布,也提高了系统的资源利用率。

**关键词:**FastDFS;多指标;组合定权法;负载评价

**中图分类号:**TP393

**文献标识码:**A

**文章编号:**1673-629X(2017)07-0010-06

doi:10.3969/j.issn.1673-629X.2017.07.003

## Investigation on Load Evaluation Model with Combination Weighting Method

ZHANG Jun, LI Ze-ping, WANG Cang-ling

(College of Computer Science and Technology, Guizhou University, Guiyang 550025, China)

**Abstract:** With the rapid development of Internet technology and the geometric growth of massive data, the distributed storage and other related technologies are rapidly developed. The application of distributed file system has been a solutions for massive file storage and management. The current mainstream distributed file systems, such as Ceph, GlusterFS, MogileFS and so on, are widely used because of their own characteristics. Owing to the problem that the index of the load evaluation for open-source distributed file system FastDFS is single and impossible to reflect the real workload status of each server in real time, according to the research on existing load balancing technology, a multi-index based load evaluation model is proposed. It combines the subjective and objective factors and determines the weighting of each index by combination weighting method, so that the weight coefficient of the model can accurately reflect the real load of FastDFS under different application scenarios. The simulation results show that the proposed model is effective and feasible, which not only realizes the reasonable distribution of the load, but also improves the resource utilization rate of the system.

**Key words:** FastDFS; multiple indicators; combination weighting method; load evaluation

## 0 引 言

互联网技术的迅猛发展以及各种个人终端和互联网应用的普及,使得网络中的数据总量呈现几何级增长。为了更加合理有效地整合这些网络资源并加以利用,云计算与大数据两大概念应运而生。底层设施的构建是所有工作的基础,其主要目标是实现低成本、高性能、可扩展、易用的分布式存储系统。分布式文件系

统的主要功能是存储互联网应用所需的大量图片、音频、视频等非结构化的数据对象。其存储的数据可分为三种类型: BLOB ( Binary Large Object, 二进制大对象)、定长块以及大文件。需要解决的关键问题包括数据分布、一致性、容错、负载均衡、事务与并发控制等<sup>[1]</sup>。

负载均衡问题是一个优化组合问题,也是一个 NP

收稿日期:2016-09-13

修回日期:2016-12-28

网络出版时间:2017-06-05

**基金项目:**国家自然科学基金资助项目(61462014);贵州省优秀科技教育人才省长专项资金资助项目(黔省专合字[2011]34);贵州省科学技术基金资助项目(黔科合 J 字[2011]2201);贵州大学引进人才科研项目(贵大人基合字[2010]034)

**作者简介:**张 军(1991-),男,硕士研究生,CCF 会员,研究方向为计算机网络与流媒体技术;李泽平,博士,教授,硕士研究生导师,研究方向为计算机网络与流媒体技术。

网络出版地址: <http://www.cnki.net/kcms/detail/61.1450.TP.20170605.1511.088.html>

完全问题。影响负载均衡的因素主要有负载均衡粒度、网络拓扑结构以及负载均衡算法。其中最核心的就是算法的选取。现有的负载均衡算法主要分为静态负载均衡和动态负载均衡两大类。静态负载均衡多考虑静态指标,按照预先设定好的分配策略进行任务分配,其有算法简单易行、系统开销小、效率高的特点,主要包括轮循调度(Round-Robin Scheduling)、加权轮循调度(Weighted Round-Robin Scheduling)、目标地址散列调度(Destination Hashing Scheduling)、源地址散列调度(Source Hashing Scheduling)等算法。动态负载均衡算法通过定期收集各节点负载情况实时调整任务的分配,在节点过载时能将负载迁移到轻载的节点上。该策略提升了系统的负载均衡度,加快了系统响应时间和任务的平均完成时间。其主要包括最小连接调度(Least-Connection Scheduling)、加权最小连接调度(Weighted Least-Connection Scheduling)、基于位置的最小连接(Locality-Based Least Connections Scheduling, LBLC)、带复制的基于位置的最小连接(Locality-Based Least Connections with Replication Scheduling, LBLCR)等算法<sup>[2-4]</sup>。合理的负载感知和评价,能为负载均衡提供可靠且准确的决策源,也直接影响着算法的效率和集群的整体性能。

### 1 问题提出

FastDFS 有着轻量级、高可靠性、高可扩展性、支持高并发访问等特点,适合以中小文件为载体的在线服务,其具有一个或多个跟踪服务器和存储服务器。由于分布式系统大都采用廉价低成本的硬件作为服务器,性能和可靠性往往不如专用服务器。所以根据每台服务器的性能和当前负载等指标合理地分配任务,最大化地利用系统资源成为提高集群总体性能的重要途径。

通过研究发现:

(1)存储服务器(Storage Server)分组(或称为分卷)组织的方式是 FastDFS 的一大特点。组中的存储服务器是完全冗余备份,随着集群负载加剧,通过增加组内服务器数量可以降低该组的平均负载。FastDFS 在组内采用静态轮询调度的方式选择服务器,而其改进算法多是根据当前服务器的性能比来分配任务。

(2)发现原有评价模型的存储服务器被选中的概率只与该组的存储空间正相关,而未能考虑到其他潜在因素,如 CPU、内存、连接数、网络带宽、操作系统等对服务器性能产生的影响。

另外,不同的应用服务对各种指标的依赖程度也有区别。例如,HTTP 服务对 CPU 的处理速度和内存有较高的要求。流媒体服务对音视频编转码和网络带

宽存在特殊需求,所以高性能 CPU、GPU 处理器以及足够高的网络带宽才能提升系统处理能力。FTP 服务侧重于服务器的磁盘 I/O、网络吞吐量等指标<sup>[5]</sup>。

针对以上问题,提出了服务器负载评价模型和组服务器负载评价模型,并实现了基于这两个模型的动态负载均衡策略。针对不同应用环境确定不同权值,增强了该模型的普适性。实验结果表明,改进算法使得分配给每个组的负载以及组内各服务器的负载与其实时性能相适应,提高了系统的性能和资源利用率。

## 2 负载评价算法

### 2.1 负载影响因素

分布式系统内的各个服务器在硬件配置上的差异使得它们能够承受的负载不尽相同。FastDFS 的原有算法因为仅考虑了存储空间单一指标,会出现某组服务器的存储空间相对较大而被频繁访问的情景。大量的用户请求都会集中在该组服务器上,所以该指标未能真实反映出结点实际所承受的负载量。此外,对于分布式文件系统而言,服务器的压力负载主要来自于客户端的读写请求。因此,性能的度量必然要涉及到磁盘 I/O<sup>[6]</sup>。例如:在 A、B 两个服务器的其他性能指标相同的前提下,若 A 的磁盘 I/O 大于 B 的磁盘 I/O,则在相同时间内 A 的吞吐量将大于 B 的吞吐量。如果在任务分配时未考虑磁盘 I/O 而选择了 B 服务器,客户端可能会出现较大延迟,同时性能更好的 A 服务器出现闲置,大大降低了整个系统的执行效率和资源利用率。

表 1 主要符号集

符号	含义
$R_{CPU}$	CPU 利用率,反映 CPU 的繁忙程度
$T_{total_n}$	从系统启动到时刻 $n$ , 总的使用时间
$T_{idle_n}$	从系统启动到时刻 $n$ , 除 I/O 等待时间以外的其他等待时间
$R_{Disk}$	存储空间的使用率
$D_H, D_T$	$D_H$ 为已使用的存储空间, $D_T$ 为节点总存储空间
$R_{Mem}$	内存使用率
$M_H, M_T$	$M_H$ 为已用内存大小, $M_T$ 为总内存大小
$R_{I/O}$	磁盘 I/O 访问率
$S_T, S_M$	$S_T$ 为磁盘实际 I/O 数据量, $S_M$ 为从磁盘系统总线上流过的最大数据量
$L_{ij}(R)$	第 $i$ 组的第 $j$ 个服务器的加权负载
$S_i$	第 $i$ 组服务器的总负载
$Max_i$	第 $i$ 组服务器的最大负载

使用如 CPU 频率、内存大小、磁盘大小、网络带宽等静态指标来表示服务器状态,它们并不能实时动态

地反映服务器真正的负载情况。文献[7]使用黑盒的方法,用平均响应时间、吞吐量等指标来量化系统性能,在负载相同的情况下,用远程存储设备的 I/O 性能来表示服务器性能。文中使用白盒的方法,针对指标的选取,提出了一些改进。表 1 为主要的符号集。

### (1) CPU 利用率。

CPU 利用率反映了当前 CPU 的繁忙程度,使用率越高则 CPU 负载就越大。在 Linux/Unix 下,CPU 利用率分为用户态、系统态和空闲态,分别表示 CPU 处于用户态执行的时间,系统内核执行的时间和空闲系统进程执行的时间。也可将 CPU 利用率表述为:CPU 执行非系统空闲进程的时间/CPU 总的执行时间<sup>[6]</sup>。

选取时间点  $t_1$  和  $t_2$ ,当两个时间点的间隔较短时,就可以把这个计算结果看作是 CPU 的即时利用率。

CPU 在  $t_1$  到  $t_2$  时间段总的使用时间为  $T_{total_2} - T_{total_1}$ ;

CPU 在  $t_1$  到  $t_2$  时间段的空闲时间为  $T_{idle_2} - T_{idle_1}$ 。

则 CPU 利用率为:

$$R_{CPU} = \frac{1 - (T_{idle_2} - T_{idle_1})}{T_{total_2} - T_{total_1}} \quad (1)$$

(2) 磁盘空间使用率:是指服务器已经使用的存储空间  $D_H$  与该节点总的存储空间  $D_T$  的比值。

$$R_{Disk} = D_H / D_T \quad (2)$$

(3) 内存使用率:指已用内存大小  $M_H$  与总内存大小  $M_T$  的比率。

$$R_{Mem} = M_H / M_T \quad (3)$$

(4) 磁盘 I/O 访问率:指磁盘在实际使用时的 I/O 数据量  $S_T$  与从磁盘系统总线上流过的最大数据量  $S_M$  的比例。

$$R_{I/O} = S_T / S_M \quad (4)$$

## 2.2 线性加权负载评价模型

### (1) 服务器负载评价模型。

线性加权求和法<sup>[8]</sup>是根据各指标对总体影响程度的差异,为其分配一定的权重系数,并对这些带权重系数的指标求和,最终得到反映当前服务器繁忙状态的负载指标。这样的负载模型充分考虑了多种因素的共同作用,同时也体现了各作用因素的重要程度,最终可以得出改进后的线性加权负载评价函数:

$$L_{ij}(R) = \sum_{s=1}^N k_s R_s \left( \sum k_s = 1 \right) \quad (5)$$

该函数表示第  $i$  组的第  $j$  个服务器的加权负载。其中,  $N$  为指标数量;  $k_s$  为权重系数,反映对应指标的影响力大小,已知权重系数,即可得出服务器负载评价函数。

### (2) 组全局负载评价模型

FastDFS 集群中 Tracker server 负责在请求到达时进行

组的选择。而 Storage server 主要用于存储和管理用户资源。每一组都由一台以上的 Storage server 组成,组与组之间、组内服务器之间在性能上都存在一些差异。为此做出如下假设:

共有  $T$  组服务器,第  $i$  组由  $c_i$  台服务器组成,其每台服务器的负载分别是  $L_{i1}, L_{i2}, \dots, L_{ic_i}$ ,则第  $i$  组的总负载为:

$$S_i = \sum_{j=1}^{c_i} L_{ij} \quad (6)$$

因为每组在服务器数量上可能存在差异,服务器多的分组其总负载往往较大,这显然不符合均衡负载的要求。故第  $i$  组的实际负载指标应取该组负载的均值  $\bar{S}_i$  更为合理,

$$\bar{S}_i = \sum_{j=1}^{c_i} (L_{ij} \times \frac{L_{ij}}{S_i}), 1 \leq i \leq T, 1 \leq j \leq c_i \quad (7)$$

## 2.3 改进后的负载均衡流程

改进后的负载均衡流程如图 1 所示。

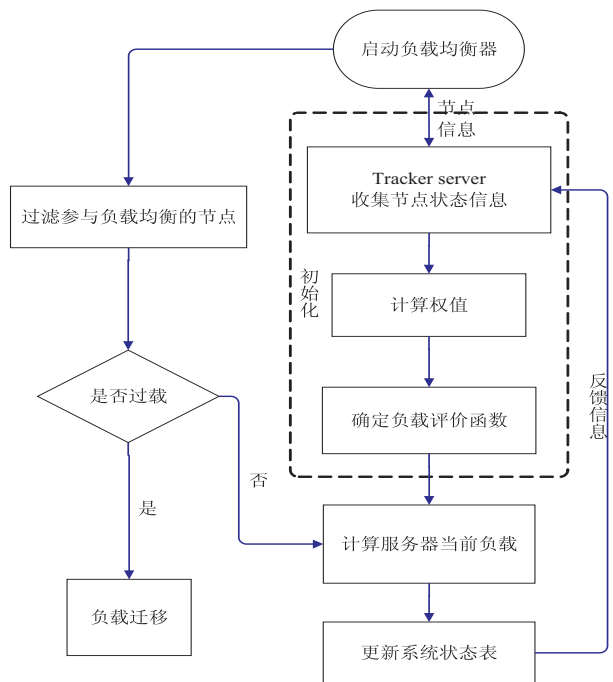


图 1 负载均衡流程图

与大部分分布式系统类似, FastDFS 集群中的 Storage server 会周期性地与 Tracker server 通过心跳连接交换状态信息,该数据包含了节点的文本同步情况、文件上传/下载次数、负载状态等基本信息。主节点将收到的心跳包中的数据转发给负载均衡器进行决策,负载均衡器通过组合定权法确定各指标权重,并根据权重向量求得负载评价函数,而后由负载评价函数计算出各服务器负载和组负载,最后 Tracker server 根据计算结果更新系统状态列表。该表记录了集群中所有服务器的序号和其当前负载情况。

以数据上传为例,为了减轻主节点在计算各服务

器负载上的开销,利用帕列托法则(也称为二八定律、巴莱特定律),给服务器组的负载设置两个阈值<sup>[9]</sup>:

(1)  $S_i \leq 20\% * \text{Max}_i$  时,当前服务器组为轻载状态;

(2)  $S_i > 80\% * \text{Max}_i$  时,当前服务器组为过载状态。

其中,  $\text{Max}_i$  表示第  $i$  组的最大负载。

进行任务分配和调度时,优先考虑轻载的服务器组,而过载的服务器组则不再参与到负载均衡中,Tracker 服务器也不再给过载节点分配新的任务,直到该组负载降到阈值以下。

### 3 确定权重

在多属性的综合决策问题上,如何合理又客观地确定指标权重显得十分重要。权重值反映了不同指标的重要性,直接影响综合评价结果。目前,根据计算数据的来源不同,可以将确定权重的主要方法分为三类:主观定权法、客观定权法、组合定权法<sup>[10]</sup>。

主观定权法是专家根据自己积累的经验和实际的应用环境来确定各个指标的影响程度,其操作简单易行,工作量较小,如德尔菲法(Delphi)和层次分析法(AHP)等。客观定权法的提出主要是为了规避主观方法的不确定和不合理性,提高决策结果的准确性和可靠性,如主成分分析法、熵权法等。组合定权法结合主客观定权法的优势,应用越来越广泛。文献[11]讨论的是仅使用熵权法进行人才的评估。文献[12]提出基于熵权法和 VIKOR 方法的传输网络方案的综合评价。文献[13]进行了基于熵权法的输电网络规划的综合模糊评价。文献[14]使用熵权法和灰色关联分析法来解决决策问题。文献[15]提出了一种熵权组合的方法用于能力评价体系的研究。采用组合定权法确定各指标权重,并将着重介绍熵权法的相关定义。

熵的概念来自热力学。熵用于描述系统的不确定性的程度。1948 年,香农首次将熵引入信息论中,从而提出了信息熵的概念。在信息论中,系统无序程度是使用熵描述的。根据熵的定义,熵值越小,信息的无序程度越低,信息的效用也就越大,那么对应的指标权重也就越高。相反,熵值越大,则指标的权重也就越低。熵权法根据数据的内在规律,客观确定了各指标的权重<sup>[14]</sup>。利用熵权法计算指标权重的过程如下:

Step1:建立评价矩阵。

如果系统有  $m$  种状态,每种状态出现的概率为  $p_i (i = 1, 2, \dots, m)$ ,则该系统的熵为:

$$E = -\sum_{i=1}^m p_i \ln p_i \quad (8)$$

其中,  $0 \leq p_i \leq 1, \sum_{i=1}^m p_i = 1$ 。

假设有  $n$  个评估对象,  $m$  个评价指标,初始化判定矩阵为:

$$X_{n \times m} = \begin{matrix} & C_1 & C_2 & \cdots & C_m \\ \begin{matrix} A_1 \\ A_2 \\ \vdots \\ A_n \end{matrix} & \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1m} \\ x_{21} & x_{22} & \cdots & x_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{nm} \end{bmatrix} \end{matrix} \quad (9)$$

$$\omega = [\omega_1 \quad \omega_2 \quad \cdots \quad \omega_m]^T$$

其中,  $A_1, A_2, \dots, A_n$  是可能的候选方案,  $C_1, C_2, \dots, C_m$  是相关指标;  $x_{ij}$  表示可选方案  $A_i$  对应  $C_j$  指标的数值;  $\omega_j$  是  $C_j$  的权重。

指标  $C_j (j = 1, 2, \dots, m)$  的信息熵为:

$$E_j = - \sum_{i=1}^n p_{ij} \ln p_{ij} \quad (10)$$

其中,  $p_{ij} = x_{ij} / \sum_{i=1}^n x_{ij}$ 。

Step2:指标类型一致化。

评价指标通常分为三种类型<sup>[11]</sup>:效率型、成本型、区间型。指标值越大越好,那么它就属于效率型。指标值越小越好,它就属于成本型。如果指标的最优解是一个区间值,那么它就属于区间型。对于区间型指标,往往需要使用点值来代替理想区间。

$$x'_{ij} = \begin{cases} \frac{x_{ij} - \min_i \{x_{ij}\}}{\max_i \{x_{ij}\} - \min_i \{x_{ij}\}}, & x_{ij} \text{ 是效益型指标} \\ \frac{\max_i \{x_{ij}\} - x_{ij}}{\max_i \{x_{ij}\} - \min_i \{x_{ij}\}}, & x_{ij} \text{ 是成本型指标} \end{cases} \quad (11)$$

进而得到一个标准化的矩阵  $X' = (x'_{ij})_{n \times m}$ 。显然,经过标准化处理后,指标都转换成了 Larger-the-Better 类型。

Step3:指标标准化。

$$y_{ij} = x'_{ij} / \sum_{i=1}^n x'_{ij} \quad (12)$$

则评价矩阵转换成:

$$Y_{n \times m} = \begin{bmatrix} y_{11} & y_{12} & \cdots & y_{1m} \\ y_{21} & y_{22} & \cdots & y_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ y_{n1} & y_{n2} & \cdots & y_{nm} \end{bmatrix} \quad (13)$$

Step4:计算熵值。

第  $j$  个指标的熵为:

$$e_j = -k \sum_{i=1}^n y_{ij} \ln y_{ij} \quad (14)$$

其中,  $k$  是玻尔兹曼常数,  $k > 0$  且  $k = 1/\ln(n)$ , 这也就保证了  $e_j \in [0, 1]$ 。并且当  $y_{ij} = 0$  时,  $y_{ij} \ln y_{ij} = 0$ 。



Step5:计算指标差异系数。

由熵的性质可知,熵值越大,指标评价值的差异性就越小,则指标在综合评价中造成的影响就越小。定义差异系数为:

$$\bar{e}_j = 1 - e_j \quad (15)$$

则当  $\bar{e}_j$  越大时,指标越重要。

Step6:计算指标熵权。

指标  $j$  的权重可定义为:

$$\omega_j = \bar{e}_j / \sum_{j=1}^m \bar{e}_j (0 \leq \omega_j \leq 1, \sum_{j=1}^m \omega_j = 1) \quad (16)$$

Step7:定义组合权重。

为了更为全面地反映各个指标的重要程度,让评估的结果更加真实可信,选择组合定权法来定义每一个指标的组合权重。

$$\theta_j = \alpha_j \omega_j / \sum_{j=1}^m \alpha_j \omega_j, j = 1, 2, \dots, m \quad (17)$$

## 4 仿真实验及结果分析

通过 MATLAB 进行数值仿真,并使用 OPNET 进行网络部分的仿真。实验将对比原有 FastDFS 基于单一指标的负载均衡算法、基于连接数的负载均衡算法和改进负载评价模型下负载均衡算法的系统负载均衡度,以验证提出算法的有效性。OPNET 提供了三层建模机制,分别在进程域、节点域和网络域进行建模,同时在仿真过程中采用离散事件驱动的模拟机理,能够准确地分析复杂网络的性能和行为<sup>[16]</sup>。

为了检验算法在非对称节点集群系统下的均衡效果,集群由十台性能各异的服务器组成,提供 HTTP 和 FTP/E-mail 服务;客户端由 600 个节点组成,统一向负载均衡器发送请求;仿真时间为 30 min。

由熵权法确定指标的权向量  $\vec{\omega} = (0.199, 0.165, 0.258, 0.378)$ ;根据主观定权法(德尔菲法),从评估专家处得到的每个指标的主观权重  $\vec{\alpha} = (0.336, 0.205, 0.195, 0.264)$ 。最终可得组合权重  $\vec{\theta} = (0.266, 0.135, 0.201, 0.398)$ 。

与原算法相同,将使用轮询算法进行任务的调度和分配。因为指标类型已经一致化,所以评价价值越高的节点,其负载越低,性能也就越好。在磁盘空间较大的情况下,由于原算法仅考虑磁盘空间,故对于磁盘 I/O 占用率很大的情况,该节点仍然会接收到新的任务。最终将导致网络拥塞,网络延迟增加,这将大大降低服务器的整体性能和用户体验。

图 2~4 是最小连接数(Least-Connection Scheduling)算法、FastDFS 原算法和改进算法的服务器负载结果。 万方数据

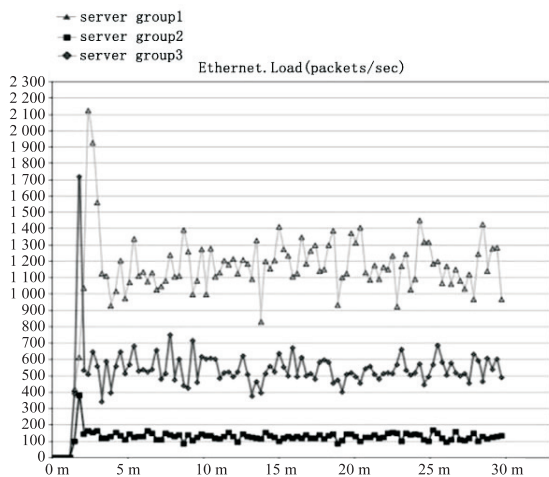


图 2 改进算法

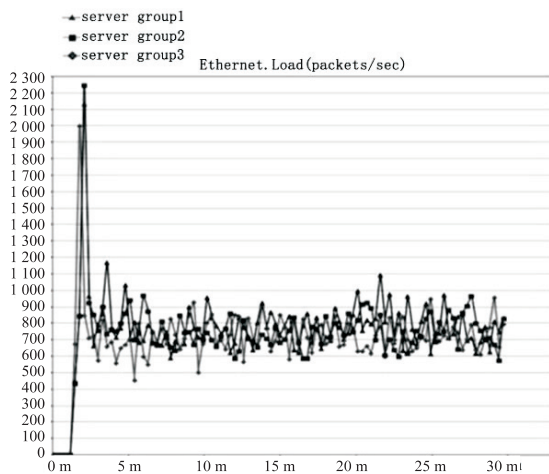


图 3 原算法

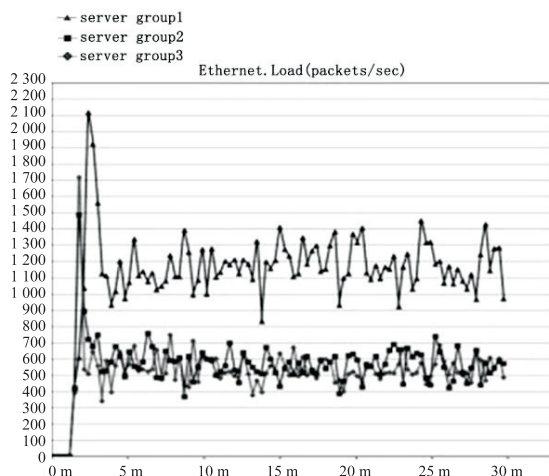


图 4 最小连接数算法

负载均衡算法最终的目的是充分利用节点状态信息,将任务进行合理调配,使得各节点分配到的任务数与其当前负载相适应。实验结果表明,最小连接数算法与 FastDFS 原负载均衡算法都无法对服务器真实负载做出合理评价,所以各服务器得到的任务数没有太大区别。而改进后的评价方法为负载均衡提供了合理

决策源,服务器组得到的任务数与其负载比例相似,达到了预期的结果。

## 5 结束语

通过研究 FastDFS 分布式文件系统,分析找出其在负载评价方法的不足,在深入研究相关负载均衡算法和确定权值方法的基础上,提出了组合定权法的方法,使得负载评价结果更加真实、可信、全面。根据得到的权值和建立的两个评价模型,可得服务器的负载评价价值,主节点得以进行负载均衡,实现了任务的合理分配和调度。

仿真结果表明,该算法在反映的服务器负载状态方面优于原算法,且在降低集群整体负载的情况下,提高了资源利用率和系统响应时间。

### 参考文献:

- [1] 杨传辉. 大规模分布式存储系统与案例解析与架构实践[M]. 北京:机械工业出版社,2015.
- [2] 韩秋英,任国恒,张少辉. 云计算中负载优化模型及算法研究[J]. 计算机与数字工程,2014,42(9):1683-1686.
- [3] 张玉芳,魏钦磊,赵 膺. 基于负载权值的负载均衡算法[J]. 计算机应用研究,2012,29(12):4711-4713.
- [4] Chi X, Liu B, Niu Q, et al. Web load balance and cache optimization design based Nginx under high-concurrency environment[C]//Third international conference on digital manufacturing and automation. [s. l.]:IEEE,2012:1029-1032.
- [5] 王 鹏,张 磊,任 超,等. 云计算系统相空间分析模型及仿真研究[J]. 计算机学报,2013,36(2):286-296.
- [6] 康承昆,刘晓洁. 一种基于多衡量指标的 HDFS 负载均衡算法[J]. 四川大学学报:自然科学版,2014,51(6):1163-1169.
- [7] Zhou L, Wang Y C, Zhang J L, et al. Optimize block-level cloud storage system with load-balance strategy[C]//26th international parallel and distributed processing symposium. [s. l.]:IEEE,2012:2162-2167.
- [8] Peng X, Zhang B, Rong L. Ship motion prediction of combination forecasting model based on adaptive variable weight[C]//Control conference. [s. l.]:[s. n.],2015:4015-4019.
- [9] 张焕青,张学平,王海涛,等. 基于负载均衡蚁群优化算法的云计算任务调度[J]. 微电子学与计算机,2015(5):31-35.
- [10] 杨学强,李文俊,岳 勇. 综合评价指标权重确定方法[J]. 装甲兵工程学院学报,2015(1):101-105.
- [11] Jing N, Wang J. Evaluate the international talents in professional social networks using the entropy weight method[C]//2015 international conference on computational science and computational intelligence. [s. l.]:IEEE,2015:220-225.
- [12] Nie H, Yu T. Comprehensive evaluation for transmission network planning scheme based on entropy weight method and VIKOR method[C]//International conference on mechatronic science, electric engineering and computer. [s. l.]:[s. n.],2011:278-281.
- [13] Ke D, Pan L, Zhang H. Comprehensive fuzzy evaluation for power transmission network planning based on entropy weight method[C]//Second international conference on intelligent computation technology and automation. [s. l.]:IEEE,2009:676-680.
- [14] Wang Q, Wu C, Sun Y. Evaluating corporate social responsibility of airlines using entropy weight and grey relation analysis[J]. Journal of Air Transport Management,2015,42:55-62.
- [15] Oh K H, Kang H K, Park J C, et al. WAGE: weighting with AHP, grey numbers, and entropy for multiple-criteria group decision making problem[C]//16th international conference on computational science and engineering. [s. l.]:IEEE,2013:360-367.
- [16] 耿 强,黄雪琴. 一种基于自适应权值的负载均衡算法[J]. 科学技术与工程,2013,13(14):4079-4082.