基于 Red5 服务器集群负载均衡调度算法研究

张亚波，苏 艺

（四川大学 计算机学院，四川 成都 610065）

摘 要： 基于动态负反馈的机制，参考现有的加权最小连接数调度和轮转调度算法，设计出了改进后的负载均衡调度算法。通过负反馈机制计算每台服务器的综合负载权重值，而综合负载权重值直接体现着服务器的当前处理能力。调度服务器根据综合负载权重值分配工作负载，实现负载的均衡分布。

关键词： Red5 流媒体服务器；集群技术；负载均衡

中图分类号： TP37 **文献标识码：** A **文章编号：** 1674-7720(2013)21-0069-03

Research on load balancing scheduling algorithm based on Red5 server cluster

Zhang Yabo, Su Yi

(Computer Science of Sichuan University, Chengdu 610065, China)

Abstract: This article designed an improved load balancing scheduling algorithm based on the dynamic feedback mechanism, and reference to the existing scheduling and rotation scheduling algorithm weighted least number of connections. The current processing capacity of the server is reflected on calculation of the negative feedback mechanism comprehensive load weight value for each server and the comprehensive load weight value directly. According to the comprehensive load weight value distribution workload, the scheduling server realizes the balanced distribution of the load.

Key words: Red5 streaming media serve; cluster technology; load balancing

1 常用负载均衡算法

常见的负载均衡调度算法有：轮转调度算法、加权轮转调度算法、最小连接数调度算法、加权最小连接数调度算法、目标地址哈希散列调度算法、源地址哈希散列调度算法等[1]。

下面主要介绍轮转调度算法和加权最小连接数调度算法的思想。

（1）轮转调度算法[2]

轮转调度算法平等地对待集群中各个服务器，认为所有服务器具有相同的处理性能。此算法将所有的 n 台服务器当作一个任务队列，当一个新的用户连接请求到来时，执行 $i=(i+1)\bmod i$，然后分配任务给第 i 台服务器。

（2）加权最小连接数调度算法[3]

加权轮转调度算法是轮转调度算法的改进，加权最小连接数调度算法是最小连接数调度算法的进一步改进，它很好地改进了最小连接数调度算法无法很好地处理集群中服务器间处理性能差异较大的缺点。即此时满足条件的服务器就是集群中正在处理的请求连接数与自身权值的比值最小的服务器，就认为它是当前负载最小的服务器，负载均衡器就会把新的请求连接交给这台服务器来处理。

2 动态反馈机制

负载均衡调度服务器收集负载信息的本质，就是每台服务器节点将自身的负载情况反馈给调度服务器的过程。这就是动态反馈机制[4]，如图 1 所示。

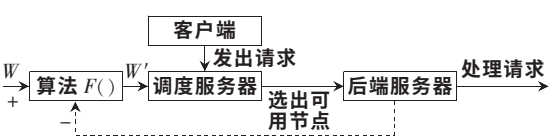

图 1 动态反馈机制模型图

图 1 中，W 为服务器更新之前状态时的权值，集群中每台服务器节点都会定期与调度服务器交互，将自身的负载信息情况反馈给调度服务器。在负载均衡调度服务器上，根据设计好的算法 $F()$ 将这些负载信息进行处

理，重新调整当前情况下服务器的新权值 W'，然后更新负载信息表，这样能够更好地反映出当前服务器的负载情况。

3 动态负反馈机制负载均衡调度算法

3.1 流媒体服务器综合权值的计算

借鉴加权最小连接数调度算法的思想，应该为每台流媒体服务器设定一个综合权值。计算综合权值所需要的指标如下[5]：

SCPUi：代表集群中第 i 台流媒体服务器的 CPU 使用率；

SMem$i$：代表集群中第 i 台流媒体服务器的内存使用率；

SNet$i$：代表集群中第 i 台流媒体服务器的网络带宽占用率；

SDisk$i$：代表集群中第 i 台流媒体服务器的硬盘 I/O 占用率；

SRoom$i$：代表集群中第 i 台流媒体服务器的聊天室数量占用率。

针对这 5 项指标，分别设定了 5 个权重值 w_1、w_2、w_3、w_4、w_5，并且有：

$$w_1+w_2+w_3+w_4+w_5=1 \tag{1}$$

每个负载指标特征参数反映了每个指标的相对重要性。而对于当前流媒体服务器的实时负载情况，有以下方式计算：

$$\text{Load}i=w_1\times\text{SCPU}i+w_2\times\text{SMem}i+w_3\times\text{SNet}i+w_4\times\text{SDisk}i+w_5\times\text{SRoom}i \tag{2}$$

根据式(2)，可以计算得出流媒体服务器的负载范围，由于各项负载评估指标都已经以百分比的形式量化，因此在最理想情况下，所有指标都为 0，即 SCPUi=SMem$i$=SNet$i$=SDisk$i$=SRoom$i$=0，所以：

$$\text{Load}i\ (\min)=w_1\times\text{SCPU}i+w_2\times\text{SMem}i+w_3\times\text{SNet}i+w_4\times\text{SDisk}i+w_5\times\text{SRoom}i=0 \tag{3}$$

此时实时负载值为 0，表明目前流媒体服务器处于空闲状态。而相对的，当流媒体服务器负载过重时，各项负载评估指标都为 1，即 SCPUi=SMem$i$=SNet$i$=SDisk$i$=SRoom$i$=1，所以：

$$\text{Load}i\ (\max)=w_1\times\text{SCPU}i+w_2\times\text{SMem}i+w_3\times\text{SNet}i+w_4\times\text{SDisk}i+w_5\times\text{SRoom}i=1 \tag{4}$$

此时实时负载值为 1，表明目前流媒体服务器已处于满载状态。

3.2 改进的负载均衡调度算法

每隔一段周期，负载均衡调度服务器都会采集服务器的实时负载情况，然后计算综合负载权重值。当综合负载权重值发生改变，这个变化量用 α 表示，那么借鉴动态反馈的机制，调整方式为：

$$W'_i=\begin{cases}W_i-\delta\times\alpha & |\alpha|>f\alpha\\ W_i & |\alpha|\leqslant f\alpha\end{cases} \tag{5}$$

式中，α 即为负载的改变量，δ 是负载对综合权重值的影响因子，$f\alpha$ 为设定的改变量的阈值。通过这样的反馈，在调度服务器上将动态调整服务器的综合负载权重值。在相关文献中提出了这样负反馈计算公式：

$$W'_i=W_i+A\times\sqrt{0.95-L_i} \tag{6}$$

该公式指定以 0.95 作为最佳综合负载值不一定能很好地体现出是否达到了最优的负载状态。而在本文中，采用的是动态的参考值作为参考标准：通过整合当前所有流媒体服务器的综合负载值，取其平均值作为参考标准 L_{avg}，然后对于单台的服务器，与其此时的综合负载值 L_i 与参考标准做比较，有以下计算：

$$L_{\text{avg}}=\sum_{i=0}^{n-1}L_i/n \tag{7}$$

但是为了达到更好的参考标准，还可以对其进行改进，采用加权负载均值的方式，能更好地体现出集群的平均负载标准，计算公式变为：

$$L_{\text{avg}}=\frac{\sum_{i=0}^{n-1}(W_i\times L_i)}{\sum_{i=0}^{n-1}W_i} \tag{8}$$

L_{avg} 将标识集群系统的平均使用率。当 L_{avg} 过高，则应该考虑增加集群的服务器节点数量来扩大集群规模；而 L_{avg} 过低，应考虑减少集群中服务器节点数量，缩小集群的规模，提高服务器节点的利用率。在集群服务器运行前，根据每台服务器的性能，初始化设定一个综合权重值 W_{start}，并且所有服务器的综合权重值都会存储在负载均衡调度服务器上。调度服务器采用动态负反馈机制，公式改进为：

$$W'_i=\begin{cases}W_i-A\times\sqrt{L_i-L_{\text{avg}}} & L_i>L_{\text{avg}}\ \text{And SOverLoad=0}\ \text{And SUnavailable=0}\\ W_i+A\times\sqrt{L_{\text{avg}}-L_i} & L_i<L_{\text{avg}}\ \text{And SOverLoad=0}\ \text{And SUnavailable=0}\end{cases} \tag{9}$$

式(9)中，W_i 为状态改变前的综合负载权重值，刚刚开始运行时则为 W_{start}，A 是调整系数，L_{avg} 为参考标准。如果当前负载 L_i 大于 L_{avg}，W'_i 将相应减小，相反 W'_i 将相应增大。而如果服务器超载(SOverload=1)或者不可用(SUnavailable=1)，W'_i 将被置为 0。改进后的调度算法步骤如下：

(1)Red5 流媒体服务器开始运行后，将加入到集群中，同时会在负载均衡调度服务器上进行注册，将自身相关信息(IP、提供服务的端口号等)发送给调度服务器；

(2)调度服务器周期性地与每台流媒体服务器交互(设定周期时间为 10 s)，采集负载信息。在流媒体服务器上，负载信息收集线程收集实时负载，通过式(2)计算实时负载 Load$i$，并将 Load$i$ 发送给调度服务器；

(3)调度服务器与流媒体服务器之间通过心跳检测进行检测,如果 3 min 内调度服务器未收到流媒体服务器的心跳,则该服务器不可用;

(4)如果当前流媒体服务器上某个负载评估指标超过设定的负载上限阈值,发送过载通知;

(5) 调度服务器将维护一张综合负载权重值表,在收集并计算每台流媒体服务器的综合负载之后,通过式(9)计算出综合负载权重值 W'_i,更新表中数据;

(6)根据新的权值,调度服务器将所有正常工作的服务器按照权值从高到低分为三组;

(7)新的请求到来时,根据步骤(6)中所分出的权重值最大的一组,依照轮询调度算法思想分配任务;

(8)判断是否是新的采样周期,如果不是则继续步骤(7),否则转向步骤(2);

(9)如果当前没有合适的服务器,则暂时停止响应用户请求,通知用户等待。

4 算法测试与分析

评价集群系统整体性能有一个重要的参数指标:平均响应时间。测试所用环境如表 1、表 2 所示。

表 1 硬件配置表

设备	CPU	内存/GB	硬盘
调度服务器	Intel Xeon E5606 2.13 GHz 四核	8	500 G 7200 rpm
数据库服务器	Intel Xeon E5606 2.13 GHz 四核	8	500 G 7200 rpm
Red5 流媒体服务器	Intel Xeon E5606 2.13 GHz 四核	8	500 G 7200 rpm
客户端	Intel core i3 3220T 2.8 GHz 双核	4	500 G 7200 rpm

表 2 软件配置表

设备	软件配置
调度服务器	Ubuntu Linux
数据库服务器	Windows Server 2008 + SqlServer 2005 + MySQL
Red5 流媒体服务器	Ubuntu Linux
客户端	Windows Server 2008

本文的设定服务器的性能阈值为 0.9,在提供常规服务时,同时向负载均衡调度服务器发送 100、200、300、400 和 500 个并发请求,比较加权最小连接数调度算法和改进动态反馈负载均衡调度算法的平均用户响应时间。测试结果如表 3 所示。

表 3 平均用户响应时间结果

请求数量	加权最小连接数调度算法响应时间/ms	改进动态反馈负载均衡调度算法响应时间/ms
100	29.8	32.5
200	95.6	101.4
300	204.0	200.3
400	484.4	452.7
500	989.1	885.4

根据表 3 的数据,在并发请求数量不是很多的情况下,两种算法的平均响应时间差不多。当并发量逐渐增大后,这时改进算法的效率明显优于加权最小连接数调度算法。

集群系统的稳定性也是必须需要考虑的问题。当采用加权最小连接数调度算法时,每台 Red5 流媒体服务器的负载情况变化如图 2 所示(采样周期为 10 s)。

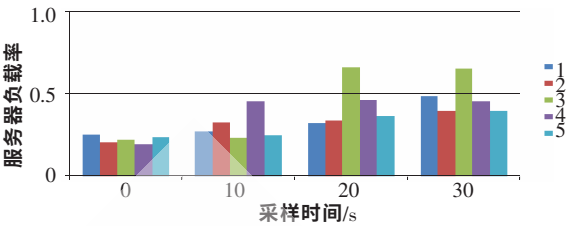

图 2 加权最小连接数调度算法负载变化图

同样条件下,集群负载均衡调度采用本文的改进算法后,每台 Red5 流媒体服务器的负载情况变化如图 3 所示。

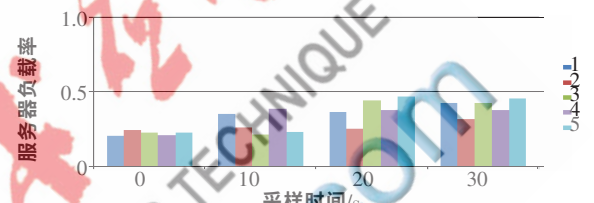

图 3 改进调度算法负载变化图

从图 2 和图 3 可知,改进的负载均衡调度算法很好地保证了整个集群的负载均衡,同时也能很好地处理当大量并发请求到来时的情况。

本文通过在轮转调度算法和动态加权最小连接数调度算法的基础上,设计了一种基于动态负反馈机制的负载均衡调度算法,能够更好地考虑到集群中每台流媒体服务器的处理性能,达到更好地负载均衡效率。而通过实验对比,可知改进的负载均衡调度算法具有更短的平均用户响应时间,也能保证集群长时间的稳定运行,能够达到负载均衡的目的。

参考文献

[1] 买京京.Web 服务器集群负载均衡技术研究[D].太原:中北大学,2008.

[2] 童瑞霞. 基于动态反馈机制的集群负载均衡算法研究[D].武汉:武汉理工大学,2011.

[3] 秦晓晨.基于动态负载均衡技术的培训系统的设计与实现[D].西安:西安电子科技大学,2012.

[4] 陈广东. 流媒体服务器集群负载均衡算法研究 [D].武汉:华中师范大学,2007.

[5] 张洪武.服务器集群与均衡技术研究[D].重庆:重庆大学,2005.

(收稿日期:2013-08-19)

作者简介:

张亚波,男,1989 年生,硕士研究生,主要研究方向:计算机网络与信息系统。

苏艺,男,1988 年生,硕士研究生,主要研究方向:计算机网络与信息系统。