文章编号: 1673-1549(2011)01-0082-04

遗传算法在负载均衡系统中的应用研究

曹　兰[1], 梁　梁[2], 全秀祥[3]
(1.漳州职业技术学院电子工程系, 福建　漳州　363000; 2.阳新一中, 湖北　黄石　435200;
3.漳州师范学院计算机科学与工程系, 福建　漳州　363000)

摘　要: 文章将并行分布式系统中广泛使用的遗传算法应用到增值业务计费系统即负载均衡系统的设计中, 并根据增值业务计费系统的具体特点, 对遗传算法作了适当的改进, 提高了后台服务器CPU的利用率, 从而改善系统性能。

关键词: 集群; 遗传算法; 优化; 负载均衡
中图分类号: TP391.9　　文献标识码: A

当前计费系统[1]是电信运营商进行市场竞争的核心支撑系统, 是运营商进行市场运作的神经中枢, 电信运营商提供电信服务所产生的服务质量问题, 越来越集中到计费领域, 所以计费系统始终处于不断变化与发展的状态, 要做到计帐准确、收帐准确和收帐及时等问题, 这就需要更高性能服务器作为计费系统, 而采用集群技术来代替高性能服务器, 具有更好的扩展性, 能根据系统负载情况进行调整。

在集群[2]负载均衡技术中, 负载均衡算法是核心, 它的好坏直接影响均衡系统的性能。由于增值业务话单的分配问题与作业调度问题原理一致, 通过均衡器可实现将多个话单一次性地分配到后台若干个服务器上进行处理, 以使总的处理时间"最小化", 进而"最优化"后台服务器端 CPU利用率。本文通过对遗传算法适当改进将其运用到均衡器分配中, 实现了一个性能优良的负载均衡系统。

1　系统的网络结构

系统的网络框架如图 1所示, 整个系统主要由三部分组成: 前端分配器、服务器节点和网关。

前端分配器: 前端分配器的功能是接受客户端的Web请求, 与客户端建立 TCP连接, 接受客户端任务请求并分析其中的内容, 然后根据一定的负载调度策略将连接和任务内容分配到相应的后端服务器节点中。

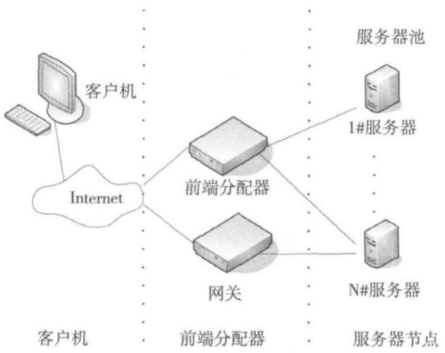

图 1　集群系统网络框架

服务器节点: 后端服务器节点根据前端分配器传递过来的连接信息重新建立与客户端的连接, 然后再接受前端分配器分配来的客户的任务请求并提供相应的服务。

网关: 后端服务器对客户端的响应数据包通过网关返回客户端, 数据包在通过网关时做必要的网络地址转换(NAT), 即将后端服务器节点的 IP地址变为前端调度器的 IP地址。

系统作为一个整体对外向客户端提供一个虚拟 VIP(VIP即是前端调度器的 IP地址), 而在系统内部, 后端服务器节点之间使用内部 IP进行通信。

2　遗传算法的设计

遗传算法(Genetic Algorithms, GA)[3-5]是一种借鉴

收稿日期: 2010-10-19
作者简介: 曹　兰(1982-), 女, 湖北天门人, 助教, 硕士, 主要从事自动化装置与嵌入式系统应用方面的研究。

生物自然选择和自然遗传学机理上的迭代自适应概率性搜索算法。从试图解释自然系统中生物的复杂适应过程入手, 模拟生物进化的机制来构造人工系统的模型。它将每个可能的问题解表示为“染色体”, 从而得到一个由染色体组成的“群体”, 这个群体被限制在问题特定的环境里。根据预定的目标函数对每个个体利用遗传算子对这些个体按“适者生存”的原则进行交叉组合产生后代。由于继承了父代的一些优良性状, 后代明显优于上一代。这样, “染色体”组成的“群体”将逐步朝着更优解的方向进化。主要流程如图 2所示:

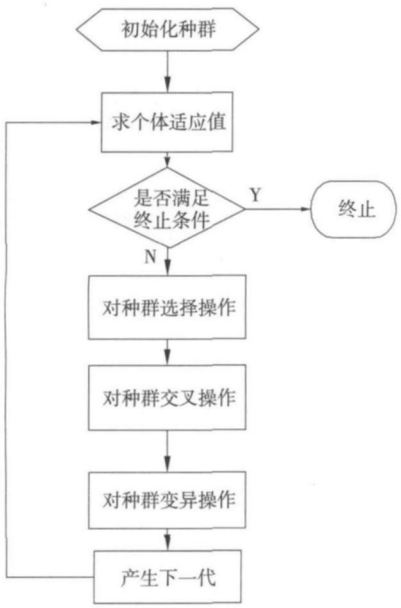

图 2　遗传算法流程图

2.1 染色体的定义

算法是根据从 CP网关收到的话单存入均衡器队列中, 然后取出一定数量的话单按“最优”分配方案分别发送到后台服务器组进行处理, 故本算法中, 用染色体表示一种分配方案。由于在均衡器的数据缓冲区中需要转发的数据包的个数不确定, 所以选择了动态数组作为染色体的结构。如图 3所示:

D0	D1	D0	D2	D1	D0	D2

图 3　染色体的表示

每个数组元素分别表示该序号的话单待分配的服务器编号, 如果按图 3给出的分配方案, 其结果是编号为 0的服务器将获 3个话单的分配, 1号服务器将获 2个话单的分配, 2号服务器将获 2个话单的分配。这样, 对每个染色体来说, 由于预先知道了每台服务器当前的 CPU 占用率, 通过计算可以得出对于当前请求分配方案, 集群中各 CPU的平均负载预算值。

$$LOAD(i) = (X(i) + 1)* RATE(i)$$

$$SUM(load) = \sum LOAD(i)$$

$$AVERAGE(load) = SUM(load)/N$$

式中, X(i)表示第 i台服务器分配的话单数; RATE(i)表示第 i台服务器 CPU的占用率; N表示集群中可用服务器的数目。

2.2 染色体适应度函数的设计[6]

当各 CPU的利用率越高, 各服务器间的负载偏差率越小, 说明系统负载均衡的性能就会越好。因负载偏差率反映的是各 CPU负载的总体分布规律, 所以本文的适应度函数的设计就采用该参数。

$$Variance(i) = LOAD(i) - AVERAGE(load)$$

$$R(variance) = \sqrt{(Variance(i))^2}/SUM(load)$$

这样, 负载偏差率 R(variance)总是在 0- 1之间取值, 值越小, 表明 CPU的负载分布越匀称, 系统整体的性能越好。

根据适应度函数的要求, 本算法设计的染色体适应度函数定义为:

$$Fitness = 1 - R(variance)$$

计算结果表明值越大, 该染色体的适应度越好, 对应的分配方案被选中的概率越高。

2.3 选择操作

选择操作采用典型的轮盘赌方法, 即将每个染色体的适应度相加, 得到一个总的适应度, 每个染色体占有一个槽, 第一个染色体的槽的范围为 0到它的适应值, 以后每个染色体的槽的范围为上一个染色体的槽的上界到这个值自已的适应值。例如, 有 3个染色体, 他们的适应值分别为 0.3、0.7和 0.5, 那么它们所占的槽分别为 (0- 0.3)、(0.3- 1.0)和 (1.0- 1.5)。在 0到总适应值中随机取一个数, 当这个数落到某个染色体所占的槽时, 这个染色体被选入下一代。如此依次选取染色体, 直到种群数目为止。

2.4 交叉操作

本文选用多点交叉操作, 由于多点交叉具有破坏性, 这种性质可以促进解空间的搜索, 使搜索更加健壮。对于交叉率来说, 交叉操作的频率越高, 可以越快地收敛到最有希望的“最优”解区域[7], 所以选取较大的交叉率, 但太高的频率也可能导致过早收敛, 一般取 0.4- 0.9, 这里根据经验取 0.6。

首先, 通过随机函数产生 0- 1间的随机数, 如小于 0.6就发生交叉操作, 否则就不进行交叉操作。如果发生交叉操作, 则通过随机函数产生 0- n之间整数确定参加本轮交叉操作的染色体上的基因位个数 m, 即数组元素个数, n为本次遗传操作中动态数组的数组元素总

数。接着,随机产生 m 个不同的随机数字,用以表示交叉位置。然后将参加交叉操作的两个父母个体的 m 个位置上的服务器编号进行交换。

2.5 变异操作

对于每个染色体来说,变异就是其上的基因发生改变。一个染色体是否发生变异是由变异率决定,变异率通常选取 0.001– 0.1 间,故在这里取变异率为 0.09。采取的变异操作,就是随机产生 0– 3之间的数据赋于染色体的某个基因。

最后,采取固定遗传代数作为终止条件,算法结束时,适应值最高的个体对应的请求调度方案将被选中。

3 遗传算法分析

对遗传算法进行测试,种群大小为 40,每个个体含有 15个元素,数值为[0, 3],表示该数组序号对应的缓冲区将分配给服务器的编号,后台服务器为 4台,其编号及 CPU利用率分别为:(0, 0.3)、(1, 0.5)、(2, 0.4)和(3, 0.2)。然后利用遗传算法进行测试。

遗传算法种群在 30代后的最优个体和适应值见表 1,按轮询算法测试(测试条件不变,其中适应值按上面遗传算法的方法计算)其适应值见表 2。

表 1 种群在 30代后的最优个体

次数	一	二	三	四	五
最优个体情况	2	3	3	3	0
	1	3	2	1	3
	0	2	2	3	3
	1	0	0	0	2
	0	0	0	3	1
	3	0	0	3	0
	3	0	3	2	3
	3	3	1	0	1
	3	3	3	3	3
	0	1	0	0	3
	0	2	3	0	2
	3	3	3	2	3
	3	0	0	0	3
	2	2	1	1	3
	2	1	3	2	0
最优适应值	0.976	0.929	0.927	0.929	0.913

由表 1,表 2分析可知,遗传算法的分配方案明显优于传统轮询算法,而且采用遗传算法进行调度分配起到的负载均衡效果更好,更稳定。

4 系统性能分析

以图 1所示搭建一个测试平台,它们的配置分别为:前端均衡器是 P4 1.4G, 256M 内存, Intel82540 10/100M 以太网卡,操作系统为 RedHat Linux9.0,以官方的 2.4.20版本内核;后端均为 P3 800Hz, 256M 内存, Intel 22530 10/100M 以太网卡,服务器软件为 Apache,使用

表 2 其最优个体的适应值

次数	一	二	三	四	五
轮询分配情况	0	3	2	1	0
	1	0	3	2	1
	2	1	0	3	2
	3	2	1	0	3
	0	3	2	1	0
	1	0	3	2	1
	2	1	0	3	2
	3	2	1	0	3
	0	3	2	1	0
	1	0	3	2	1
	2	1	0	3	2
	3	2	1	0	3
	0	3	2	1	0
	1	0	3	2	1
	2	1	0	3	2
适应值	0.815	0.836	0.872	0.819	0.815

的服务器分别提供页面、图片以及视频文件的服务;网关为 P3 550Hz, 128M 内存, Intel22530 10/100M 以太网卡。用四台 PC作为客户端产生网络流量。

从图 4,图 5中可以明显看出,遗传算法与传统的算法相比在任务请求数较少时系统的响应延时和吞吐量的差异不是很明显,但在任务数慢慢增大的情况下它们之间的差异越来越明显,遗传算法明显地在网络延迟、吞吐量方面都要优于传统算法。因此,基于请求内容的遗传负载均衡算法能够在任务请求数较大的情况下更加合理的分配任务请求,提高整个 Web服务器负载均衡集群系统的可用性。

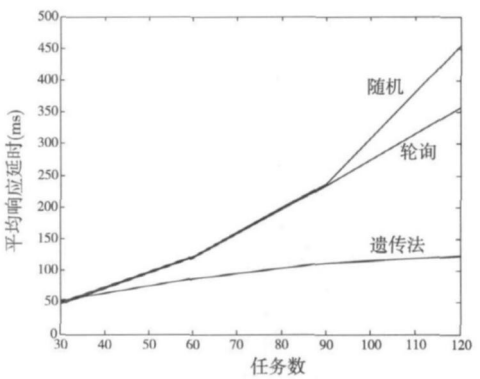

图 4 平均响应延时比较

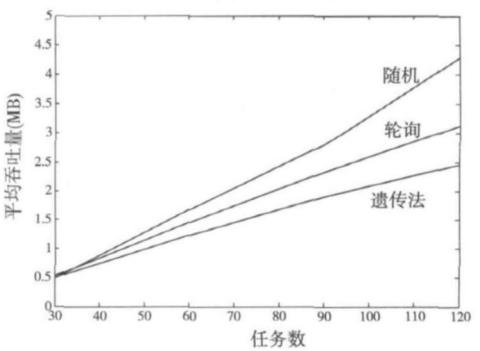

图 5 吞吐量比较

5 结束语

通过遗传算法在负载均衡系统中的应用,能较好地实现系统动态调度。在系统吞吐量、平均响应延时和负载均衡,遗传算法比传统的调度算法有更好的效果。

参考文献:

[1] 王建芬,陈 曦,曾凡锋.基于 Linux的网关计费系统的
实现[J].北方工业大学学报,2006,18(3):16-19.

[2] 王晋鹏,潘龙法,李降龙. LVS集群中的动态反馈调度算法[J].计算机工程,2005,31(19):40-58.

[3] 王小平,曹立明.遗传算法-理论、应用与软件实现[M].西安:西安交通大学出版社,2002.

[4] Goldberg D E. Genetic Algorithms in Search, Optimization and Learning[M]. New York: Addison-Wesley, 1989.

[5] Mitsuo C, Runwei C. Genetic Algorithms and Engineering Design[M]. New York: Wiley, 2002.

[6] 程海根,盛双福,张显昆.基于遗传算法和 BP神经网络的结构损伤识别[J].四川理工学院学报:自然科学版,2009,22(5):82-85.

[7] 许小丽. 一种新的交叉粒子群算法[J].四川理工学院学报:自然科学版,2010,23(1):19-21.

Load-balancing System's Study of Value-added Business Billing System Based on Genetic Algorithm

CAO Lan[1], *LIANG Liang*[2], *QUAN Xiu-xiang*[3]

(1. Department of Electronic Engineering, Zhangzhou Institute of Technology, Zhangzhou 363000, China; 2. Yangxin No. 1 Middle School, Huangshi 435200, China; 3. Computer Science & Engineering Department, Zhangzhou Normal University, Zhangzhou 363000, China)

Abstract A billing system of value-added business load-balancing based on genetic algorithm, which is applied to parallel and distributed systems popularly, is designed. According to particular of value-added business billing system, through some improvements on genetic algorithm, the utility of servers CPU is increased, and performance of the system of load-balancing is enhanced.

Key words cluster; genetic algorithm; optimization; load balancing