

# ASAS 集群的模糊控制策略

夏明波<sup>1</sup>, 王晓川<sup>1</sup>, 金士尧<sup>1</sup>, 李祖秀<sup>2</sup>, 胡光强<sup>3</sup>

- (1. 国防科学技术大学 计算机学院 并行与分布式国家重点实验室, 湖南 长沙 410073;  
2. 华南师范大学 信息技术学院, 广东 广州 510631;  
3. 苏州国华科技有限公司, 江苏 苏州 215021)

**摘要:**传统的集群服务器依靠前端分发器基于负载均衡来分发客户请求,而ASAS集群采用了基于后端服务器负载分析后的主动调度策略。控制ASAS集群后端服务器的CPU利用客户请求的响应时间,或者连接容量,却很难建立一个精确的数学模型,因此传统的控制器很难控制。这里设计了一个模糊控制算法,控制算法的输入是CPU利用率和客户请求的响应时间,输出是连接数,试验表明这种模糊控制策略取得了很好的性能表现。

**关键词:**集群;主动自调度服务器;模糊控制

中图分类号:TP273<sup>1</sup>.4

文献标识码:A

文章编号:1673-629X(2006)12-0039-04

## Fuzzy Control Strategy in ASAS Cluster

XIA Ming-bo<sup>1</sup>, WANG Xiao-chuan<sup>1</sup>, JIN Shi-yao<sup>1</sup>, LI Zu-xiu<sup>2</sup>, HU Guang-qiang<sup>3</sup>

- (1. National Key Lab. for Parallel and Distributed Processing, School of Computer,  
National University of Defense Technology, Changsha 410073, China;  
2. School of Information and Technology, South China Normal University, Guangzhou 510631, China;  
3. Suzhou Gopha Technology Co., Ltd., Suzhou 215021, China)

**Abstract:** In traditional cluster server, the front-dispatcher distributes clients' requests by load balancing, but ASAS(Automatic Self-Allocating Server) cluster depends on its back-server's Automatic Self-Allocating based on load status. Because of the difficulty to build a precise mathematical model in the back-server of ASAS cluster to control CPU usage and response time of clients' request or link capacity, traditional controller is hard to manage, so we designed a fuzzy control algorithms. Its inputs are CPU usage and response time of clients' request, its output is link number, tests show that this fuzzy control method achieves good performance.

**Key words:** cluster; automatic self-allocating server; fuzzy control

## 0 引言

在因特网的发展中,网络带宽增长的速度远高于网络服务器处理性能的增长,千兆以太网、ATM、千兆以太网已相继投入使用,容量达数百个Gbit的宽带主干网已在中国不少地区建成,随之而来的是宽带用户的不断增加,使得今后网络应用的性能瓶颈越来越多地出现在服务器端。

高性能的网络服务器系统研究已经成为一项急迫的任务,ASAS(Automatic Self-Allocating Server)服务器应运而生。ASAS集群系统将以以往专用负载均衡器的负载均衡工作分解、融入到后端各个执行服务器,集中器对服务器的调度使用的唯一指标是执行服务器发送的容量信息,服务器调度的主要工作在事实上已转移到各执行服

器。

执行服务器中容量的分析是一项技术难度很大的任务,容量控制机制的好坏直接决定了整个集群系统的性能高低,因此一个高效的容量控制机制至关重要。

## 1 相关背景理论介绍和研究现状

### 1.1 集群服务器负载均衡技术

传统集群服务器方案中,都由专用的负载均衡器来接受用户请求,负载均衡器根据所有服务节点的处理能力和现状来决定将这个请求分发给某个服务节点,以NAT,IP隧道及直接路由<sup>[1]</sup>等方式将请求分发到后端各执行服务器。

现有的集群网络服务器方案,无论是基于域名调度的集群服务器还是基于分发器调度的集群服务器,均以统一的负载评估节点为基本特征,带来的问题主要是执行服务器负载评估不尽准确,在系统高负载下容易出现执行服务器“活锁”现象(即前端集中器很忙,而后端执行服务器空闲),在访问量骤增的情况下会导致整体性能明显下降,甚

收稿日期:2006-03-23

作者简介:夏明波(1980-),男,湖南武冈人,硕士研究生,研究方向为信息安全与集群技术;金士尧,教授,博导,研究方向为系统仿真与信息安全。

至面临着宕机、数据丢失、系统崩溃等潜在危险。

为了提高服务器的性能,针对传统集群服务器的缺点,一种主动自调度方法被提出<sup>[2]</sup>。该方法已在苏州国华科技有限公司 ASAS-1000、ASAS-2000 等产品中成功运用,产品在苏州日报网站、北京金山软件网站运行良好。

ASAS 集群的负载均衡策略<sup>[3]</sup>与其他集群的负载均衡策略的本质区别在于 ASAS 是基于后端执行服务器根据自身负载的主动申请,调度的主体在后端,真正地实现了动态反馈调度;而一般集群是前端承担主要的调度任务,前端对后端资源情况的把握不如后端对后端本身资源的把握准确,也很难实现动态反馈调度。

## 1.2 模糊控制技术

1965 年美国的控制论专家 L. A. Zadeh 教授创立了模糊集合论<sup>[4]</sup>,从而为描述、研究和处理模糊现象提供了一种新的工具。1973 年他又给出了模糊逻辑控制的定义和相关定理,从而奠定了模糊控制的理论基础。1974 年英国的 Mamdani 设计了一个模糊控制器并用于锅炉和蒸汽机的控制,从而开创了模糊控制的新纪元。

将模糊集合理论运用于自动控制而形成的模糊控制理论,在近年来得到了迅速的发展,其原因在于对那些时变的非线性的复杂系统,当无法获得精确的数学模型的时候,利用具有智能的模糊控制器能给出有效的控制。例如,在炼钢、化工、人文系统、经济系统以及医学心理系统中,要得到正确而且精密的数学模型是相当困难的。对于这些系统却具有大量的以定性的形式表示的极其重要的先验信息,以及仅仅用语言规定的性能指标。同时,要求过程的操作人员是系统的基本组成部分等。所有这些都是一种不精确性,应用一般的控制理论是很难实现控制的,但是,这类系统由人来控制却往往容易做到。这是因为过程操作人员的控制方法是建立在直观的和经验的基础上,他们凭借实践积累的经验,采取适当的对策完成控制任务,于是,人们把操作人员的控制经验归纳成定性描述的一组条件语句,然后运用模糊集合理论将其定量化,使控制器得以接受人的经验,模仿人的操作策略,这样就产生了以模糊集合理论为基础的模糊控制器。

模糊控制作为智能领域中最具有实际意义的一种控制方法,已经在工业控制领域、家用电器自动化领域和其他很多行业中解决了传统控制方法无法或者难以解决的问题,取得了令人瞩目的成效。目前,模糊控制技术与神经网络、遗传算法、粗糙集等其他先进技术的集成在理论和工程上都取得了可喜的成绩。

## 2 模糊控制在 ASAS 中的应用

### 2.1 为什么采取模糊控制

打开 Windows 任务管理器,查看性能,将看到 CPU 利用率很不稳定,波动性很大,而且看不出它怎么变化。现在再打开几个应用程序,CPU 利用率骤升,但也只能肯定在程序启动时的几秒钟内平均 CPU 利用率较高,而在

这几秒内也可能有几个非常短的时间片内 CPU 利用率很低。也就是说,CPU 利用率突变性强,很难给出一个精确的数学模型。从对服务器 CPU 使用情况的监控来看,CPU 利用率的变化也如此。CPU 利用率是一个模糊的概念,没有严格的数学模型,传统控制器很难控制或者说控制效果不好,而模糊控制则有先天性优势,可以采用模糊控制技术。

在 ASAS 集群中,客户的请求缓存在前端集中器,真正的服务在后端的执行服务器,后端的执行服务器通过采集本机的资源状态,然后量力去前端集中器上索取服务请求。执行服务器中的服务程序接到请求后就回复应答信息,这就是服务程序唯一的本职工作,给它的请求再多,它也会逐一应答,它不会因为请求过多而拒绝服务。这样,当缓存在集中器上的客户请求很多时,执行服务器的 CPU 资源的利用率就会非常高,网络流速也会超限,响应时间也会特别长,这不是我们所期望的。为此,需要在执行服务器中设计控制器,结合前一段的分析,决定采取模糊控制器。

### 2.2 模糊控制与 ASAS 的结合

#### 2.2.1 设计模糊控制器结构

模糊控制系统通常只有一个输入变量,就是  $y(k)$ ,即在  $k$  时刻的采样数据。利用该数据和所期望的一个预定值,还可以得出更多的值。在简单的控制器中,通常取采样数据与给定值的偏移,即  $e(k) = y - y(k)$ ,作为整个控制器的输入就够了。有些情况下单纯知道这个偏移还不够,于是进一步求取  $\Delta e(k) = e(k) - e(k-1)$ 。目前工业过程中大都采取这种二维结构,比如加热炉的加风量控制、燃料电池的温度控制、温室的湿度控制、溶液 pH 值控制及焦炉煤气管压力控制等等,都采用这种方法。

在 ASAS 集群的执行服务器中,可以采集的数据或者说要控制的数据有很多,比如 CPU 利用率、内存利用率、磁盘的读写、响应时间、网络流量等等,如果把这些量都放到模糊规则表中,规则数将达到  $7^5 = 16\ 807$  条(这里模糊变量取 7 个模糊子集),如果再求  $\Delta e$ ,规则库就不可控制了,这就是典型的维数爆炸问题。

事实上,要考虑的上述 5 个量,并不是同等重要的。国外有些学者提出“结构化模糊控制”方法(hierarchical fuzzy control<sup>[5]</sup>),把最重要的控制量放到第一层,次要的放到第二层,依次类推,第一层的输出基本上决定了输出结果,第二层可以对第一层的输出结果作一个修改,依次类推。这种方法的缺点是没有解决好各控制量之间的耦合问题。这里要考虑的这 5 个量之间是有一定的关联的,比如 CPU 忙的适合响应时间往往比较长,内存利用率过高的时候可能需要虚存,这和磁盘的读写也有直接关系。笔者采取的办法就是在保证服务质量的前提下,尽可能少地考虑控制量。这样只选择 CPU 利用率和响应时间这两个最重要的指标作为系统的输入变量。

选择  $e_{(cpu)}$  和  $e_{(response)}$  这两个量作为控制器的输

人,解决了维数爆炸问题,也不需要复杂的结构化控制器,同时也符合 ASAS 集群这一具体应用环境。图 1 就是笔者设计的模糊控制器。

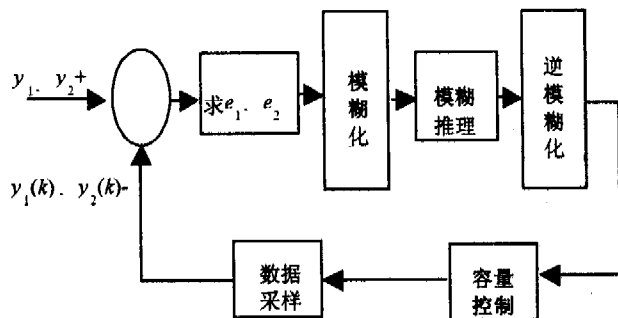


图 1 ASAS 中模糊控制器的结构

其中  $y_1, y_2$  分别表示 CPU 利用率和响应时间的预定值  $y_{(cpu)}$  及  $y_{(responsetime)}$ ,  $y_1(k), y_2(k)$  分别为 CPU 和响应时间在  $k$  时刻的采样值  $y_{(cpu)}(k)$  及  $y_{(responsetime)}(k)$ 。

## 2.2.2 模糊集和隶属函数的确定

$e_{(cpu)} = y_{(cpu)} - y_{(cpu)}(k)$ ,  $e_{(responsetime)} = y_{(responsetime)} - y_{(responsetime)}(k)$ , 各参数意义同上。 $e_{(cpu)}$  和  $e_{(responsetime)}$  都分成如下的 7 个模糊语言集:

- (1) Big positive (BP)
- (2) Medium positive (MP)
- (3) Small positive (SP)
- (4) Zero (ZE)
- (5) Small negative (SN)
- (6) Medium negative (MN)
- (7) Big negative (BN)

对应的隶属函数如图 2 和图 3 所示。

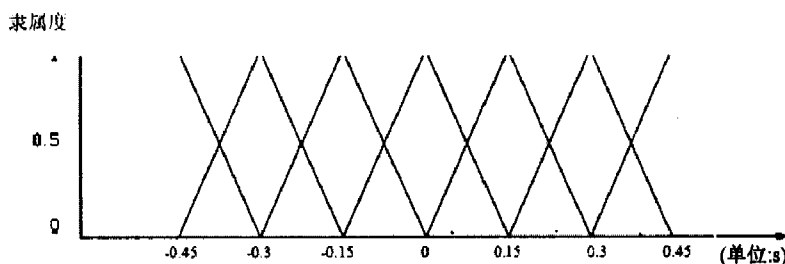


图 2  $e_{(cpu)}$  隶属函数图

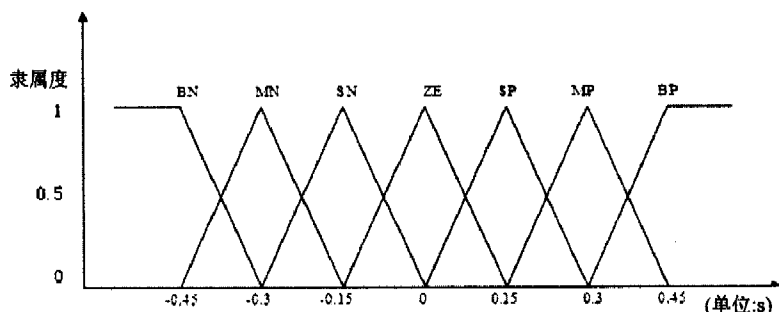


图 3  $e'_{(responsetime)}$  隶属函数图

图 2 中 CPU 的预定值为 0.5, 即 CPU 利用率为 50%。图 3 中响应时间的预定值为 0.7s。这些值可以在

具体使用环境中稍作调整。

输出变量  $u(k)$  为执行服务器向前端集中器索取的连接数, 分成 9 个模糊子集, 即再增加 HP(Huge positive) 和 HN(Huge negative), 其隶属函数如图 4 所示。

## 2.2.3 建立规则表

根据经验设计的模糊规则表如表 1 所示。

表 1 模糊控制规则表

$u$		$e_{(cpu)}$						
		BN	MN	SN	ZE	SP	MP	BP
$e_{(responsetime)}$	BN	HN	HN	BN	MN	SN	ZE	ZE
	MN	HN	BN	MN	SN	ZE	ZE	SP
	SN	BN	BN	MN	SN	ZE	SP	SP
	ZE	MN	SN	SN	ZE	SP	SP	MP
	SP	SN	ZE	ZE	ZE	SP	MP	BP
	MP	SN	ZE	ZE	SP	MP	BP	HP
	BP	ZE	SP	SP	MP	BP	HP	HP*

第 \* 条规则为: if  $e_{(cpu)}$  为 BP and  $e_{(responsetime)}$  为 BP then  $u$  为 HP, 即当 CPU 利用率很低且响应时间很快时就大量增加连接数。

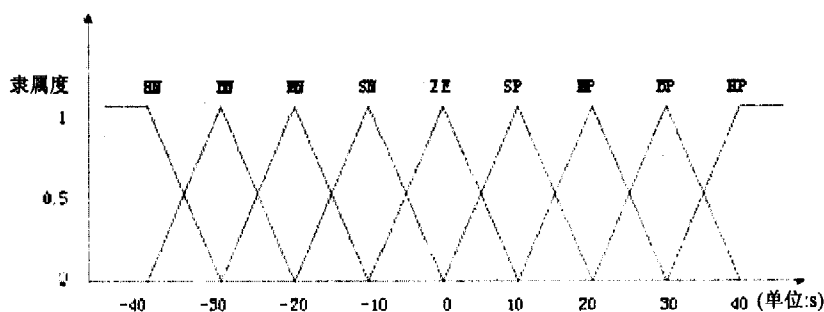


图 4 输出变量  $u(k)$  隶属函数图

## 2.2.4 模糊推理

下面以一个具体例子来说明推理过程。取 CPU 利用率的预定值为 0.5, 响应时间的预定值为 0.7s。假定测到 CPU 利用率为 0.275, 响应时间为 0.73s, 则  $e_{(cpu)} = 0.225$ ,  $e_{(responsetime)} = -0.03$ 。

### (1) 规则匹配。

从上面的隶属函数图, 可以计算出它们对应的隶属度:

$$u_{SP}(0.225) = 0.5, u_{MP}(0.225) = 0.5$$

$$u_{ZE}(-0.03) = 0.8, u_{SN}(-0.03) = 0.2$$

从而可以得到推理结果, 如表 2 所示。

从表 2 可知, 被触发的规则有四条:

R1: if  $e_{(cpu)}$  is SP and  $e_{(responsetime)}$  is SN then  $u$  is ZE

R2: if  $e_{(cpu)}$  is SP and  $e_{(responsetime)}$  is ZE then  $u$  is SP

R3: if  $e_{(cpu)}$  is MP and  $e_{(responsetime)}$  is SN then  $u$  is SP

R4: if  $e_{(cpu)}$  is MP and  $e_{(responsetime)}$  is ZE then  $u$  is SP

表 2 模糊推理结果

u		e(cpu)						
		BN	MN	SN	ZE	SP(0.5)	MP(0.5)	BP
e(response time)	BN	0	0	0	0	0	0	0
	MN	0	0	0	0	0	0	0
	SN(0.2)	0	0	0	0	$u_{ZE}$	$u_{SP}$	0
	ZE(0.8)	0	0	0	0	$u_{SP}$	$u_{SP}$	0
	SP	0	0	0	0	0	0	0
	MP	0	0	0	0	0	0	0
	BP	0	0	0	0	0	0	0

## (2) 计算可信度。

在同一条规则内,前提之间通过“与”的关系得到规则结论。前提的可信度之间通过取小运算,得到每条规则总前提的可信度为:

R1 前提的可信度为  $\min(0.5, 0.2) = 0.2$

R2 前提的可信度为  $\min(0.5, 0.8) = 0.5$

R3 前提的可信度为  $\min(0.5, 0.2) = 0.2$

R4 前提的可信度为  $\min(0.5, 0.8) = 0.5$

将前提的可信度和表 2 进行“与”运算,得到规则的可信度如表 3 所示。

表 3 规则可信度表

u		e(cpu)						
		BN	MN	SN	ZE	SP(0.5)	MP(0.5)	BP
e(response time)	BN	0	0	0	0	0	0	0
	MN	0	0	0	0	0	0	0
	SN(0.2)	0	0	0	0	$\min(u_{ZE}, 0.2)$	$\min(u_{SP}, 0.2)$	0
	ZE(0.8)	0	0	0	0	$\min(u_{SP}, 0.5)$	$\min(u_{SP}, 0.5)$	0
	SP	0	0	0	0	0	0	0
	MP	0	0	0	0	0	0	0
	BP	0	0	0	0	0	0	0

模糊系统总的可信度为各条规则可信度推理结果的并集,即

$$u_{\text{agg}} = \max\{\min(u_{ZE}, 0.2), \min(u_{SP}, 0.2), \min(u_{SP}, 0.5), \min(u_{SP}, 0.5)\} = \max\{\min(u_{ZE}, 0.2), \min(u_{SP}, 0.5)\}$$

可见实际触发了 2 条规则。

## (3) 逆模糊化。

在计算精确的输出值时,通常有三种方法:最大隶属度法、取中位法和加权平均法。最大隶属度法利用的信息比较少,会引起一定的不精确性;而取中位法充分利用模糊子集提供的信息,但是计算太复杂。这里采用普通加权平均法。

从上面的计算可知,模糊系统的输出  $u_{\text{agg}}$  实际上是两

个规则 R1 和 R2 推理结果的并集。对应 R1 规则,输出变量隶属度 0.2,对照图 4(ZE 部分)可以计算出两个输出值为 -8 和 8。对应 R2 规则,输出变量隶属度 0.5,对照图 4(SP 部分)可以计算出两个输出值为 5 和 15。则输出量

$$u = \frac{0.2 * (-8) + 0.2 * 8 + 0.5 * 5 + 0.5 * 15}{0.2 + 0.2 + 0.5 + 0.5} = 50/7$$

则在原来链接数的基础上,再增加 7 个。

## 2.3 模糊控制合理性分析

当 CPU 利用率比较轻松 ( $50\% - 27.5\% = 22.5\%$ ), 响应时间刚好满足客户需求 ( $0.7 - 0.73 = -0.03$ ) 的时候,下一时刻增加 7 个连接数,可以提高 CPU 利用率,但是增加的并不是很多,这样就不会影响服务质量。通过模糊控制器计算出的这个值是很符合现实环境的。

## 3 结论及展望

至此,模糊控制器已设计完成。其工作原理是:从后端的执行服务器获取 CPU 利用率,从前端集中期得到服务响应时间,为了使这两个指标保持在一定的水平,通过决定执行服务器与前端服务器的连接数来控制。比如,当 CPU 利用率较低、响应速度很快时,说明执行服务器比较空闲,于是增加它的连接数,即加重下一时间片内的负载,从而提高 CPU 资源的利用率;而当 CPU 利用率较高、响应时间较长时,就减少连接数,于是减轻了下一时间片的任务,从而可以加快响应,保证服务质量。

文中设计的模糊控制器,取得了很好的性能表现。下一步研究中,将考虑引进神经网络以增加学习能力,同时可以吸取遗传算法的思想。

## 参考文献:

- [1] 章文嵩. 可伸缩网络服务的研究与实现[D]. 长沙:国防科技大学计算机学院, 2000.
- [2] 金士尧. 主动式集群服务器总体设计. 中国, 02114011.1 [P]. 2003.
- [3] 王晓川. 主动式集群网络服务器调度机制的研究[D]. 长沙:国防科技大学计算机学院, 2001.
- [4] Zadeh L A. Fuzzy Sets[J]. Information and Control, 1965, 8: 33-35.
- [5] Raju G V S, Zhou Jun. Hierarchical fuzzy control[R]. Athens, Ohio: ECE Department, Ohio University, 1999.

(上接第 38 页)

而新的串行接口标准——光纤通道使用简单的点到点互连,提供了双口传输能力,使数据在两个独立的数据通路上传输,增加了故障冗余,提高了可靠性,同时降低了电缆连接的复杂程度。

## 参考文献:

- [1] ANSI. X3T9.3 Task Group. Fibre Channel Physical and Sig-

naling Interface (FC-PH), Rev. 4.2[M]. US:ANSI, 1993.

- [2] Fiber Channel Association. Fibre Channel: Connection to the Future[M]. [s.l.]: Fiber Channel Association, 1994.
- [3] 颜浩南, 华晓红. 光纤通道浅述[J]. 高性能计算技术, 2004, 167(2): 27-30.
- [4] Kessler G. Changing channels[J]. LAN Magazine, 1993(12): 69-78.
- [5] Meggyesi Z. Fibre Channel Overview[J]. High Speed Interconnect, 1994(8): 15-18.