

可伸缩 TAGS

闵 帆 张君雁 杨国纬

(电子科技大学计算机学院 成都610054)

Scalable TAGS

MIN Fan ZHANG Jun-Yan YANG Guo-Wei

(Computer Science Institute, UESTC, Chengdu 610054)

Abstract In a distributed Web server system where tasks are unpreemptible, the most important issue for improving quality of service (QoS) is how to realize fairness and reduce average slow down. In this paper we present an algorithm named Scalable TAGS by integrating Central Queue algorithm and Task Assignment by Guessing Size (TAGS), together with its performance analysis, system parameter setting algorithm subject to fairness requirement, and optimal grouping method.

Keywords Distributed web server system, Task assignment, Fairness, Slow down, Bounded pareto

1 引言

近年来,分布式 Web 服务器系统由于其价廉、易于扩充等特点被广泛采用。该类系统的模型如图1所示。任务到达任务分配服务器后,被分配到某一个任务处理服务器(下文中均简称为服务器)执行。在该类系统中的任务分配策略对服务质量起着决定性作用。

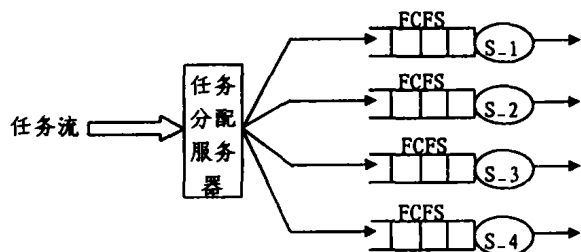


图1 任务分配模型

若把本文假设条件(见2.1节)中的任务长度改成指数分布,则集中队列算法能获得最佳性能^[1];但任务长度服从有界 Pareto 分布,且服务器数量较少时,TAGS 算法能获得比集中队列算法好得多的性能^[2]。

本文结合集中队列算法与 TAGS 算法,提出一种可伸缩的 TAGS 算法,并依次给出其性能计算式、公平性要求下的系统参数设置算法及服务器最佳分组方法。

2 可伸缩 TAGS 算法

2.1 假设条件

本文采用与文[2]相同的假设条件:1. 任务长度(按处理

所需时间计)不可预知;2. 任务不可剥夺,即任务在执行过程中被中断,则已做工作全部作废;3. 各服务器以先来先服务的方式处理任务;4. 任务到达服从 Poisson 分布;5. 任务长度服从有界 Pareto 分布;6. 各服务器处理能力完全相同。

2.2 性能评价标准

定义1(延迟, delay) 任务从进入系统到正式被处理的间隔时间。

定义2(延迟比, slow down) 任务延迟与其处理时间需求之比。

定义3(服务器平均延迟比) 在指定服务器完成任务的延迟比均值。

定义4(系统平均延迟比) 系统内所有任务的延迟比均值。

定义5(任务级公平) 系统中所有任务具有相同的延迟比。

定义6(服务器级公平) 系统中所有服务器具有相同的平均延迟比。

在多数情况下,定义5中所描述的任务级公平难以达到;我们采用定义6所描述的服务器级公平作为前提,并以定义4所描述的系统平均延迟比为性能评价标准。

2.3 算法描述

可伸缩 TAGS 算法如图2所示。

任务到达任务分配服务器时,若 S_{i-1} (图中 $i=1,2,3$) 中有一个或多个空闲,则将该任务分配给其中最早进入空闲状态者,否则将任务放入任务分配服务器队尾; S_{i-1} 完成当前任务时,若任务分配服务器中队列不为空,则取出其队首任务执行,否则进入空闲状态。

到达 S_{i-1} 的任务如果在 s_j (具体说明见表1)个单位时间

闵 帆 博士研究生,主要研究方向:分布式系统、主动网络。张君雁 博士研究生,主要研究方向:移动 Agent、主动网络。杨国纬 教授,博士生导师,主要研究方向:计算机网络、数据库。

博士研究生,主要研究方向:移动 Agent、主动网络。杨国纬 教授,博士

- 7 Cvijovic D, Klinowski J. Taboo search: an approach to the multiple minima problem. *Science*, 1995, 267: 664~666
- 8 Laguna M, Glover F. Bandwidth packing: a tabu search approach. *Management Science*, 1993, 39: 492~500
- 9 Lee C Y, Kang H G. Cell planning with capacity expansion in mo-

bile communications: a tabu search approach. *IEEE Trans. on Veh. Tech.*, 2000, 49: 1678~1691

- 10 Matsumura T, Nakamura M, Tamaki S, Onaga K. A parallel tabu search based on aspiration control and its cooperative execution. *IEICE Trans. on Fundamentals*, 2000, E83-A: 2196~2202

内完成,则退出本系统,否则被分配到 $S_{i,j+1}$ 重新执行。

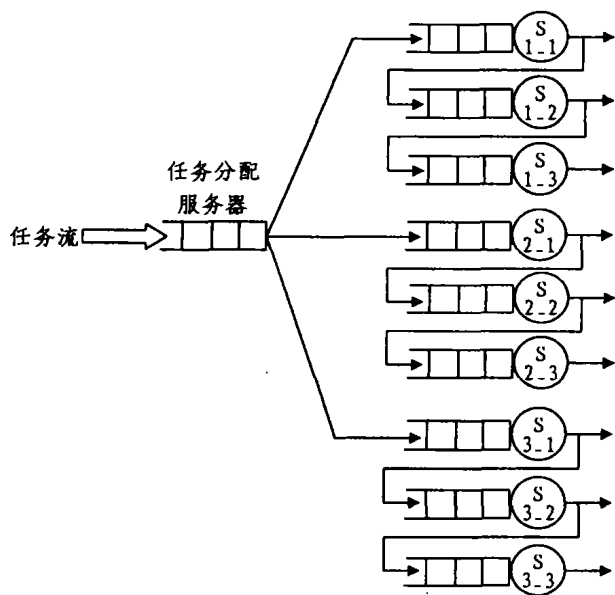


图2 可伸缩 TAGS 算法

各队列均为先进先出(FIFO),即:各服务器(包括任务分配服务器)均为先来先服务(FCFS)。

简而言之,该算法将服务器分成了若干组,组内采用 TAGS 算法^[2],组间采取集中队列算法^[1]。显然,组数为1时,该算法退化为 TAGS 算法;每组服务器数为1时,该算法退化为集中队列算法。

3 服务器平均延迟比的计算

3.1 符号及其含义

表1 符号及其含义

c	服务器组数
n	每组内服务器数
$B(k, p, a)$, 简记 X	任务长度分布,任务长度指单个服务器将其完成所需时间,下同
k	任务长度下界
p	任务长度上界
$f(x)$	$B(k, p, a)$ 的密度函数
a	$f(x)$ 的参数,本文中假设 $0 < a < 2$
s_j	$S_{i,j} (i=1, 2, \dots, n)$ 服务时间上界,规定 $k=s_0 < s_1 < s_2 < \dots < s_n=p$
λ	系统任务到达率
$\lambda_{i,j}$	$S_{i,j}$ 任务到达率
$\lambda_{i,j}$	$S_{i,j}$ 任务到达率
ρ	系统总体负载,不考虑服务器资源浪费
$\rho_{i,j}$	$S_{i,j}$ 负载
$E\{X\}$	任务长度均值
$E\{X_{i,j}\}$	$S_{i,j}$ 中所完成任务长度均值
$E\{X_{i,j}\}$	$S_{i,j}$ 中所执行任务在该服务器中实际执行时间均值
$E\{W_{i,j}\}$	$S_{i,j}$ 中所执行任务在该服务器中实际等待时间均值
$E\{W_{i,j}\}$	$S_{i,j}$ 中所完成任务实际等待时间(包括在前面各服务器中等待与执行时间)
$OVR_{i,j}$	$S_{i,j}$ 中所执行任务中未完成任务的比例
$E\{S_{i,j}\}$	$S_{i,j}$ 平均延迟比

3.2 总体分析

由表1中定义,有^[3]:

$$f(x) = \frac{\alpha p^* k^*}{p^* - k^*} x^{-\alpha-1} \quad (1)$$

$$\rho = \frac{\lambda \cdot E\{X\}}{c \cdot n} \quad (2)$$

各组服务器具有相同的服务器数和服务时间上界设置,因此它们具有对称性;为计算整个系统的性能,只需计算任何一组服务器的性能。

$E\{S_{i,j}\}$ 的计算分成三个步骤:

1. 准备工作,见3.3节;

2. 计算 $E\{W_{i,j}\}$,分成如下两种情况:

$a, j=1$. 只考虑任务分配服务器与 $S_{i,1}$,显然 $S_{i,1}$ 中不存在排队问题,而任务分配服务器的任务到达率服从 Poisson 分布,所以可将其看成 $M/G/c$ 系统。见3.4节;

$b, j>1$. 只考虑 $S_{i,j}$,为便于分析,我们假设其任务到达服从 Poisson 分布^[2,6],当然这只是一种近似,因为实际的任务到达间隔时间一定不小于 s_{j-1} 。这时可将其看作 $M/G/1$ 系统。见3.5节;

3. 计算 $E\{S_{i,j}\}$,见3.6节。

3.3 准备工作

$$E\{X_{i,j}\} = \int_{s_{j-1}}^{s_j} f(x) \cdot x dx = \begin{cases} \frac{S_j S_{j-1}}{S_j - S_{j-1}} (\ln S_j - \ln S_{j-1}) (\alpha = 1) \\ \frac{\alpha}{1-\alpha} \cdot \frac{S_j^{1-\alpha} - S_{j-1}^{1-\alpha}}{S_j^{-\alpha} - S_{j-1}^{-\alpha}} (\alpha \in (0, 1) \cup (1, 2)) \end{cases} \quad (3)$$

$$E\{X_{i,j}^2\} = \int_{s_{j-1}}^{s_j} f(x) x^2 dx = \frac{\alpha}{2-\alpha} \cdot \frac{S_j^{2-\alpha} - S_{j-1}^{2-\alpha}}{S_j^{-\alpha} - S_{j-1}^{-\alpha}} \quad (4)$$

$$E\{X_{i,j}^{-1}\} = \int_{s_{j-1}}^{s_j} \frac{f(x)}{x} dx = \frac{\alpha}{\alpha+1} \cdot \frac{S_{j-1}^{-\alpha-1} - S_j^{-\alpha-1}}{S_{j-1}^{-\alpha} - S_j^{-\alpha}} \quad (5)$$

$$OVR_{i,j} = \frac{\int_{s_{j-1}}^{s_j} f(x) dx}{\int_{s_{j-1}}^{s_j} f(x) dx} = \frac{S_j^{-\alpha} - p^{-\alpha}}{S_{j-1}^{-\alpha} - p^{-\alpha}} \quad (6)$$

$$E\{X_{i,j}\} = OVR_{i,j} \cdot S_j + (1 - OVR_{i,j}) \cdot E\{X_{i,j}\} \quad (7)$$

$$E\{X_{i,j}^2\} = OVR_{i,j} \cdot S_j^2 + (1 - OVR_{i,j}) \cdot E\{X_{i,j}^2\} \quad (8)$$

由于各组是对称的,有:

$$\lambda = \frac{\lambda}{c} \quad (9)$$

$$\lambda_{i,j} = \lambda \cdot \int_{s_{j-1}}^{s_j} f(x) dx = \frac{\lambda}{c} \cdot \frac{p^* k^*}{p^* - k^*} (S_{j-1}^{-\alpha} - p^{-\alpha}) \quad (10)$$

$$\rho_{i,j} = \lambda_{i,j} \cdot E\{X_{i,j}\} \quad (11)$$

3.4 $E\{W_{i,j}\}$ 的计算

由 $M/G/c$ 系统与 $M/M/c$ 系统的关系^[6],有:

$$E\{W_{i,j}\} = E\{W_{M/M/c}\} \cdot \frac{E\{X_{i,j}^2\}}{2E\{X_{i,j}\}^2} \quad (12)$$

其中^[8]:

$$E\{W_{M/M/c}\} = \frac{\frac{\rho_{1,1}}{c}}{\lambda(1 - \frac{\rho_{1,1}}{c})^2} \cdot p_c \quad (13)$$

$$\rho_{1,1} = \frac{\lambda \cdot E\{X_{1,1}\}}{c} \quad (14)$$

$$p_c = \frac{\rho_{1,1} \cdot p_0}{c!} \quad (15)$$

$$p_0 = \left[\sum_{j=0}^{c-1} \frac{\rho_{1,j}^j}{j!} + \frac{c \rho_{1,c}^c}{c!(c-\rho_{1,1})} \right]^{-1} \quad (16)$$

3.5 $E\{W_{i,j}\} (2 \leq j \leq n)$ 的计算

由 Pollaczek-Khinchin 公式^[9], 有:

$$E\{W_{i,j}\} = \frac{\lambda_{i,j} E\{X_{i,j}^2\}}{2(1-\lambda_{i,j} E\{X_{i,j}\})} \quad (17)$$

3.6 $E\{S_{i,j}\}$ 的计算

$$E\{W'_{i,j}\} = E\{W_{i,j}\} + \sum_{m=1}^{j-1} (E\{W_{i,m}\} + s_{i,m}) \quad (18)$$

由于任务长度与其等待时间独立, 有:

$$E\{S_{i,j}\} = E\{W'_{i,j}/X_{i,j}\} = E\{W'_{i,j}\} \cdot E\{X_{i,j}^{-1}\} \quad (19)$$

综上所述, 如果已知 $f(x)$, n, c, λ (或 ρ) 和 s_i , 那么根据以上各式就可以求得 $E\{S_{i,j}\}$ 。

另外, 从各式中可看出, $E\{S_{i,j}\}$ 的值与 i 无关, 即它与服务器处于哪一个组无关。这也进一步证实了 3.2 节中所述对称性。

4 服务器级公平要求下系统平均延迟比与 s_i 的计算

服务器级公平可表述为:

$$E\{S_{i,1}\} = E\{S_{i,2}\}$$

其中, $i=1, j=1, 2, j \in \{1, 2, \dots, n\}$ (20)

显然, 在该约束下, 系统平均延迟比与各服务器平均延迟比相等。

根据第 3 节中各式可看出, $E\{S_{i,j}\}$ 与 i 的取值无关, 因此 (20) 式实质上表示由 $n-1$ 个互不相同的方程组成的方程组。

给定 $f(x)$, n, c, λ (或 ρ), 而将 s_i 作为变量, 则该方程组共有 $n-1$ 个变量。该方程组最多只有一个有意义的解, 但无法用数学方法直接解出, 因此使用下面的试探性算法计算:

```
初始化  $s_1, \text{delta}$ ;
while (true){
    计算  $E(S_{i,1})$ ;
    for (int j=2; j<=n; j++){
        使用二分法调整  $s_j$ , 使得
         $\text{abs}(E(S_{i,j}) - E(S_{i,1})) / E(S_{i,1}) < \text{delta}$ ;
    }
    计算  $E(S_{i,n})$ ;
    if ( $\text{abs}(E(S_{i,n}) - E(S_{i,1})) / E(S_{i,1}) < \text{delta}$ )
        break; // (20) 式满足, 算法结束
    if ( $(E(S_{i,n}) < 0) \parallel (E(S_{i,n}) > E(S_{i,1}))$ )
        // 最后一个服务器负荷过重
        增加  $s_1$ ; // 使用二分法
    else
        减小  $s_1$ ; // 使用二分法
}
```

算法中 delta 表示近似精度要求, 通常取 0.001 已足够。

5 服务器最佳分组方法

本节讨论给定 $f(x)$, $c * n, \lambda$ (或 ρ) 时, 如何对服务器进行分组, 以在公平性要求下使系统具有最好性能 (最低平均延迟比)。

由于 c 和 n 均为整数, 我们可以采取穷举法列出所有可能的分组方法, 分别计算系统平均延迟比, 并选取使其最低的。下面用一个例子予以说明。

例 当 $k=1, \rho=10^7, \alpha=1, \rho=0.5$ 时, $c * n=4$ 或 6 时, 求系统的最佳分组。

求得不同的设置下的平均延迟比, 如表 2 所示:

表 2 不同 c 和 n 的设置下的延迟比

$c \backslash n$	1	2	3	4	5	6	7	8
1	155105	148	18.5	7.76	5.19	4.35	4.12	4.2
2	5170	27.3	6.17	3.48				
3	313	7.87						
4	20	2.66						
5	1.21							
6	0.0675							
7	0.00345							
8	0.00016							

从表 2 易知: 若有 4 个结点, c 取 1、2、4 时系统平均延迟比依次为 7.76、27.3、20, 因此应取 $c=1$; 同理, 若有 6 个结点, 则应取 $c=6$, 即 $n=1$ 。

从表 2 中还可看出如下趋势: 结点数量较少时 (本例中为 4 个之内), 最佳分组使算法退化为 TAGS; 而随着结点总数的增加 (本例中为 5 个以上), 最佳分组使算法迅速退化为集中队列法。对于不同的参数设置, 该趋势均成立, 不同之处在于其结点个数的分界线。

显然, 采取最佳分组时, 算法的性能不低于集中队列法与 TAGS 中的任何一个。

小结 本文提出了一种可伸缩 TAGS 算法, 并解决了其参数设置的一系列问题, 只要给定 $f(x)$, $c * n, \lambda$ (或 ρ), 根据本文的方法就可计算出满足定义 6 中所述公平性的最佳设置。

在实际系统运行过程中, 任务分配服务器可根据统计得出 $f(x)$ 的参数和 λ (或 ρ), 以进行系统的最佳分组与服务时间上界 s_i 的设置。统计与设置可以按时间段 (如每小时一段) 进行, 以使系统总处于最优状态。

参考文献

- Wolff R W. Stochastic Modeling and the Theory of Queues. Prentice Hall, 1989
- Harchol-Balter M. Task Assignment with Unknown Duration. In: Distributed Computing Systems, 20th Intl. Conf. on, 2000. 214~224
- Johnson N L, Kotz S. Continuous Univariate Distributions-1, Houghton Mifflin Co., Boston, 1970
- Harchol-Balter M, Downey A. Exploiting process lifetime distributions for dynamic load balancing. ACM Transactions on Computer Systems, 1997, 5(3): 253~285
- Milojicic D S, et al. Process Migration. ACM Computing Surveys, 2000, 32(3): 241~299
- Harchol-Balter M, Crovella M, Murta C. On choosing a task assignment policy for a distributed server system. IEEE Journal of Parallel and Distributed Computing, 1999, 59: 204~228
- Winston W. Optimality of the shortest line discipline. Journal of Applied Probability, 1977, 14: 181~189
- 唐应辉, 唐小我. 排队论——基础与应用[M]. 成都: 电子科技大学出版社, 2000
- Hillier F S, Lieberman G J. Introduction to Operations Research. McGraw-Hill, 5th edition, 1990