计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.191200072

基于 M/M/1 排队模型的网络服务尾延迟分析

郭子亭[1,2]　张文力[1]　陈明宇[1,2,3]
1 中国科学院计算技术研究所计算机体系结构国家重点实验室　北京 100190
2 中国科学院大学　北京 100049
3 鹏城实验室　广东 深圳 518055
(guoziting@ict.ac.cn)

摘　要　研究表明,在大规模的网络服务系统中,网络延迟往往表现出长尾效应,即存在一定比例的延迟远大于网络的平均延迟。延迟的长尾引起了广泛的关注,长尾效应会严重影响用户体验和内容供应商收益,尤其在对延迟敏感的大型交互式网络应用中。因此,网络服务系统研究的重点经历了从关注吞吐量和平均延迟到关注系统的尾延迟的变化。然而,现有的理论模型大多关注平均延迟,难以用来分析网络服务延迟的尾部特征,复杂网络服务中的尾延迟时间计算缺乏形式化的建模和计算方法。文中提出了一种将复杂的网络抽象为以 M/M/1 排队模型为基础的排队网络模型的方法,在该模型的基础上给出了串联、并行场景下逗留时间尾延迟分布的表达式,同时分析了当模型中个别子部件发生变化时对系统整体尾延迟的影响,并将模型预测结果和仿真网络的结果进行对比,误差不超过 2%。

关键词:尾延迟;建模;M/M/1;串联;并行
中图法分类号　TP391

Network Service Tail Latency Analysis Based on M/M/1 Queuing Model

GUO Zi-ting[1,2],ZHANG Wen-li[1] and CHEN Ming-yu[1,2,3]
1 State Key Laboratory of Computer Architecture,Institute of Computing Technology,Chinese Academy of Sciences,Beijing 100190,China
2 University of Chinese Academy of Sciences,Beijing 100049,China
3 Peng Cheng Laboratory,Shenzhen,Guangdong 518055,China

Abstract　Studies have shown that in large-scale network service systems,network latency often exhibits a long tail effect,i.e.,a certain percentage of latency are much larger than the average latency of the network service systems. The long tail of latency has caused widespread concern,and can seriously affect user experience and content provider revenue,especially in large interactive network applications that are sensitive to latency. Therefore,the focus of network service system research has experienced a change from focusing on throughput and average latency to the tail latency of the system of interest. However,most of the existing theoretical models focus on average latency,and it is difficult to analyze the tail characteristics of the network service latency. Due to the complexity of existing research,stay time calculations in complex network services lack formal modeling and computational methods. This paper proposes a method of abstracting a complex network into a queuing model. Based on the model,the expressions of stay time distribution in series and parallel scenes are given,and at the same time,the influence of the tail latency on the change of individual sub-components in the model is analyzed. The predicted results of the model analysis are compared with the results of the simulation network,the error does not exceed 2%.

Keywords　Tail latency,Modeling,M/M/1,Series,Parallel

1　简介

1.1　问题背景

快速响应用户需求是网络服务系统设计的目标之一。特别是交互式网络服务,如 Web 搜索、金融交易、游戏和社交网络等,需要快速平稳地响应来吸引和留住用户。因此,服务提供商为 95 百分位或更高百分位的响应时间定义了严格的目标,通常称为尾部延迟[1-5]。一些研究[1,6-7]表明,在互联网规模的网络服务系统中,网络延迟会表现出长尾行为,即存在一定比例的逗留时间远大于所有逗留时间的平均值,即具有较长的尾部延迟,特别是在大规模、并行和交互式应用场景下,会出现平均响应延迟无法反映网络延迟的极端边界情况。低

到稿日期:2019-12-12　返修日期:2020-04-05　本文已加入开放科学计划(OSID),请扫描上方二维码获取补充信息。
基金项目:国家重点研发计划项目(十三五)(2017YFB1001602)
This work was supported by the National Key Research and Development Program of China (13th Five-Year Plan) (2017YFB1001602).
通信作者:张文力(zhangwl@ict.ac.cn)

延迟对于交互式网络服务应用程序非常关键，对于网络应用的使用者尤其重要。实现持续性的低延迟是具有挑战性的。现代应用程序是高度分布式的，应用程序级操作往往需要完成几十个甚至几百个任务，例如许多对象压缩成一个 Web 页面或者聚合许多后台请求产生一个前端结果[8]。这意味着单个任务可能只有几毫秒或者几十毫秒的延迟预算，延迟分布的尾部非常关键。实际系统中通常会出现一些异常的尾部延迟值，这些异常值很难消除，因为它们在复杂网络中的来源很多。虽然在一个配置良好的系统中，个别操作通常能够正常工作，但对于整个系统来说，一定程度的不确定性仍然普遍存在。因此对于大型网络服务系统来说，降低尾部延迟是一项巨大的挑战。

本文研究主要关注尾部延迟，因为它对用户体验的影响是不成比例的。在高度并行的分布式系统中，较长的尾部延迟将影响大多数请求。例如考虑这样一个系统，该系统中的每个服务器通常在 10 ms 内响应，但它的第 99 分位延迟为 1 s，即有 1% 的用户的延迟将超过 1 s。如果仅在这样一个服务器上处理用户请求，那么 100 个用户请求中将有一个请求经历至少 1s 的延迟。如果用户请求必须收集来自 100 个这样的服务器的响应后才能返回，那么会有 63% 的用户请求将花费至少 1 s[1]。高延迟将会严重影响用户体验，使用户感受到网络的卡顿、不流畅，同时也会影响内容提供商的用户数和效益。例如，Google 有数据指出 400 ms 的网络延迟会使用户/搜索降低 0.59%[9]。Bing 搜索中每 2 s 的延迟会使效益下降 2%[9]。

系统中出现长尾效应的原因是多方面的。文献[1]进行了比较详细的研究，例如共享资源导致的资源争用，后台进程可能会导致延迟产生周期性的峰值；中间服务器和网络交换机中的多层排队放大了出现长尾现象的可能性；一些硬件上的限制也可能导致长尾现象的出现，如功率限制、垃圾收集和能源管理等。可想而知，提供可预测的低延迟服务是非常困难的，并且随着系统规模的扩大和复杂性的提高以及用户数量的增加，系统的尾延迟增加会更加显著且难以预测。

因此，尾延迟逐渐成为网络服务系统设计中的关注热点之一。文献[10]就通过一个 M/M/k 排队模型补充了 Amdahl 定律对并行、多核系统的分析，详细研究了尾延迟是如何影响系统设计的，尤其是在硬件方面。该文还讨论了在尾延迟的限制下系统设计如何在 CPU 性能、并行性、缓存与硬件资源之间进行权衡。但是对于更加复杂的网络服务系统，如多级服务，就需要更加复杂的模型来分析尾延迟。本文主要解决如何在复杂的网络服务系统中定量分析尾延迟。

1.2 现有方法的局限性和挑战

排队论是定量分析网络延迟的重要数学工具之一，是通过对服务对象到达过程以及服务时间的统计研究，得出系统的一些数量指标（等待时间、排队长度、忙期长短等）的统计规律，是优化系统的有效工具。排队论中提供了一个描述服务对象到达率、服务率和端到端的逗留时间之间关系的框架，如最简单的 M/M/1 排队模型就描述了按泊松分布到达、服务时间服从指数分布的排队过程。Jackson 排队网络[11]则描述了具有不同拓扑的开环网络。在排队论中，排队模型的经典分析方法一般是通过建立马尔可夫链，对排队模型的稳态情况进行求解。但是当马尔可夫链的状态数较多或者排队模型比较复杂时，求解稳态方程就变得比较困难，甚至无法求解。而且，排队论和排队网络中通常研究系统性能的平均指标，如平均队长、平均等待时间、平均排队时间等，无法满足描述经过系统之后逗留时间的尾部特征的需求。

当然，我们也可以使用仿真的方式来获取网络系统中关于尾延迟的信息，例如使用 Matlab 进行多次仿真，但是仿真的方式只能针对一组参数运行一次得到一个结果，效率较低，不容易得到通用的结论。而从网络服务系统中抽象出数学模型，利用数学手段可以更好地对尾延迟进行定量和趋势分析。由于现在网络的复杂性，复杂网络服务中的延迟时间分布缺乏形式化的建模和计算方法。建立合适的网络模型可用于辅助延迟敏感的系统分析、预测尾延迟，进而达到优化系统尾部延迟的目标，并且可以通过 Matlab 仿真来验证模型的准确性。

1.3 本文工作

本文工作是通过对网络服务系统抽象、建模，推导出系统逗留时间的分布，进而求出尾延迟的表达式，在此基础之上探究网络服务系统中部分子部件的变化对系统整体尾延迟的影响。

具体来说，本文首先将网络服务系统抽象为一个基于 M/M/1 队列串联、并行组合的排队网络模型，然后在 M/M/1 排队模型的基础上分别给出了多级 M/M/1 串联、并行场景下逗留时间分布的计算方法和逗留时间分布的表达式。由于逗留时间分布的表达式比较复杂，我们无法从中直接推导出尾延迟的精确解析表达式，因此我们借助数值工具求解出尾延迟的近似解，并分析了几种特殊情况：各级参数都相同时的多级串联、并行逗留时间的分布以及它们分别和单级 M/M/1 的对比情况；各级参数都相同时的多级串联、并行系统中有任意一级尾延迟发生变化后系统尾延迟的变化情况以及影响因素。同时，利用 Matlab 工具仿真了网络服务系统处理请求的过程，验证了表达式的准确性，并将上述结论应用到小型网络中来预测网络的尾延迟，所提计算方法的预测结果与 Matlab 仿真结果相比，误差不超过 2%。

2 模型建立

如何刻画网络服务系统中逗留时间和尾延迟的分布？

网络服务系统通常由许多服务器连接组成，如图 1 所示。请求到达系统接受多个服务器的服务后离开系统。假设网络服务系统中有 n 个服务器，每个请求会在服务器的输入队列排队等待被处理，然后离开该服务器到下一个服务器或离开系统，因此服务器可以被抽象为排队模型。本文将每个服务器抽象为最简单的 M/M/1 排队模型[10]，即到达服务器 i 的请求服从参数 λ_i 的泊松分布，服务器处理请求的时间服从参数为 μ_i 的指数分布，这是因为泊松分布是网络中最常见的数据包的到达分布，M/M/1 排队模型也是排队论中最简单的模型。请求在系统中的逗留时间为 t，目标尾延迟的百分位为 s。本文模型的目标是分析请求在系统中百分位 s 处的延迟 $t(s)$ 与上述变量的关系。表 1 列出了本文描述模型的相关符号。

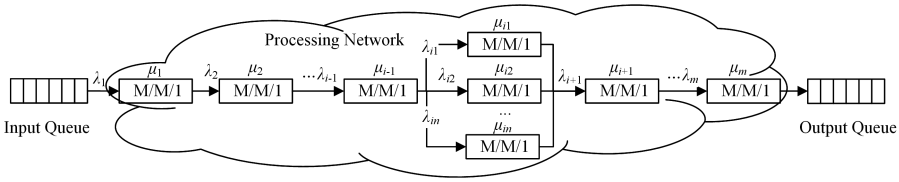

图 1 尾延迟分析模型

Fig. 1 Tail latency analysis model

表 1 模型相关的符号描述

Table 1 Symbol description related to model

符号	含义
λ	平均到达率
μ	平均服务速率
ρ	服务强度/利用率
W	平均逗留时间
s	百分位数
t	逗留时间

该模型中暂时不考虑反馈、环等设计。

定义 1(尾延迟) 逗留时间(在系统中的排队时间加上处理时间)分布的尾部,通常指大于 90% 的延迟。本文主要讨论在第 99 百分位处的延迟时间(即 $s=99$)。

3 尾延迟模型分析

3.1 M/M/1 排队模型的总结回顾

我们将复杂网络抽象为以 M/M/1 排队模型为基础的模型,排队论中已有关于请求的逗留时间分布的结论:

$$W(t)=1-e^{-(\mu-\lambda)t},t\geqslant 0 \tag{1}$$

根据式(1)可以得出在第 s 百分位下对应的逗留时间 t 为:

$$t=-\frac{\ln(1-s/100)}{\mu-\lambda} \tag{2}$$

即第 s 百分位的请求在系统中的逗留时间小于 t。

从式(2)可以看出,当到达率 λ 不变时,服务率 μ 越高,M/M/1 排队系统的利用率($\rho=\lambda/\mu$)越低,即服务器的处理能力越强,第 s 百分位的延迟越小。当服务率 μ 不变时,到达率 λ 越大(利用率 $\rho=\lambda/\mu$ 越高),第 s 百分位的延迟越大。因此以低利用率运行服务器可以改善尾延迟。

提高服务器的利用率意味着处理突发状况的能力下降,一个小的突发流有可能在服务器内构建出较长的队列。服务器的服务率越高表明服务器的处理能力越强,因此尾延迟也越低。

3.2 多级串联的排队模型的尾延迟分布

网络服务系统通常由许多服务器相互连接构成,如果将服务器看成 M/M/1 排队模型,那么网络服务系统则可以看成多个 M/M/1 排队模型串并联组成。但是排队论中很少有关于串联或者并行多个 M/M/1 排队模型之后的尾延迟的研究,这使得 M/M/1 排队模型无法直接应用于复杂的网络服务系统的尾延迟分析。

互联网中有多种级联网络的应用场景,例如电子商务中,当用户下单一件商品,后台往往要经过多个服务器处理,如查询是否有剩余库存,查询该用户是否有优惠券可以使用,选择地址,发送支付请求等步骤之后才能购买成功,整个过程可以看作一个多级串联的排队系统。

3.2.1 串联场景分析

当多个 M/M/1 排队模型串联时,其本质上是一个单队列多服务台的串联排队模型,如图 2 所示,下面将讨论在该排队模型下的逗留时间的分布。

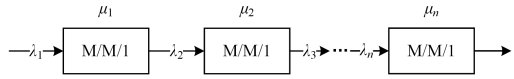

图 2 多级 M/M/1 串联模型

Fig. 2 Multi-level M/M/1 series model

在该模型下,第 i 级的请求按照参数为 λ_i 的泊松分布的规律到达,第 i 级的服务时间遵循参数为 μ_i 的指数分布。那么第 i 级在系统中的逗留时间的概率密度函数为:

$$f_i(t)=(\mu_i-\lambda_i)e^{-(\mu_i-\lambda_i)t},t\geqslant 0 \tag{3}$$

为了简化分析,本文令 $a_i=\mu_i-\lambda_i>0$。

对于 n 级串联系统,请求在系统中的逗留时间的累积分布函数 $P_{1,n}(t)$ 为:

$$P_{1,n}(t)=1-\sum_{i=1}^{n}\prod_{j\in[1,i]}^{j\neq i}\frac{a_i}{a_i-a_j}\int_0^t a_i\,e^{-a_1 x}dx \tag{4}$$

从式(4)的对称性可以看出,改变串联服务的顺序不影响尾延迟的分布。

3.2.2 多级相同 M/M/1 串联排队模型的尾延迟分布

本节讨论当每一级的参数都相同时的串联系统,即 $a_i=a_j\equiv a,i\neq j$,此时该系统退化成 n 阶 Erlang 分布系统,它是 n 个参数为 a 的指数分布的卷积。这时,请求在系统中的逗留时间的分布为:

$$P_n(t)=1-e^{-at}\sum_{i=1}^{n}\frac{(at)^{i-1}}{(i-1)!},t\geqslant 0,a>0 \tag{5}$$

当每一级模型的参数均为 a 时,可以将 at 看作一个整体,那么式(5)可以改写为:

$$P_n(t)=1-\sum_{i=1}^{n}\frac{(t)^{i-1}}{(i-1)!}e^{-t},t\geqslant 0 \tag{6}$$

现在分析 s 分位下对应的逗留时间 t,即系统中逗留时间小于 t 的请求所占的比例为 $s/100$。令 $P_n(t)=s/100$,求解 t 的值。由于根据该表达式无法用 s 百分位和系统级数 n 来表示相应的逗留时间 t,我们利用 Matlab 求解出给定 s 和 n 之后的近似解,并以单级 M/M/1 排队模型在对应百分位数的逗留时间作为基准值进行对比,结果如图 3 所示。从图 3 可以看出,不同百分位数的延迟和单级的比值与系统级数不总是相同的,即分布在参考线 Ref 两边。其中,在第 70 百分位处,多级相同的 M/M/1 模型串联之后的逗留时间与单级 M/M/1 模型逗留时间的比值基本与参考线 Ref(n 级系统比值为 n)相近,尤其在系统级数 n 较小时,其原因是第 70 百分位的值近似等于系统的平均值,并且随着分位值的减小,n 级系

统与基准值(1 级 M/M/1 排队模型)的比值越来越大(大于 n)；反之，分位值 s 越大，n 级系统与基准值的比值越来越小(小于 n)，并且由于在式(5)中用 t 代替了 at，那么两个时间 t 的比值与 a 无关，即与系统的到达率 λ 和服务率 μ 无关。如果 1 级 M/M/1 排队模型在第 99 百分位的延迟为 m，那么经过 n 级串联的系统所得到的第 99 百分位的延迟将小于 nm。文中后续主要以 5 级串联系统为例，从图 3 中可以看出，5 级相同的 M/M/1 模型串联的第 99 百分位的延迟是 1 级第 99 百分位延迟的 2.52 倍，10 级串联之后的第 99 百分位延迟则是单级的 4.079 倍。

结论 1　相同的 n 级串联系统中，70 百分位处的延迟与单级 M/M/1 排队模型逗留时间的比值基本与 n 相同。

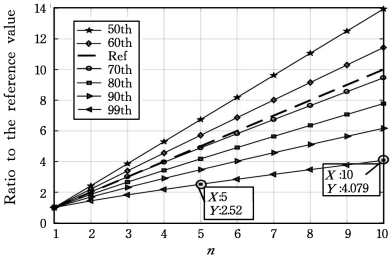

图 3　多级相同 M/M/1 串联模型分析

Fig. 3　Multi-level identical M/M/1 series model analysis

3.2.3　单级服务器尾延迟发生变化时对系统尾延迟的影响

在多级串联 M/M/1 系统中，各个阶段的请求数是相互独立的，各个阶段的处理也是相互独立的。当每一级的服务器的服务率都相同时，尾延迟是随着级数的增长而线性增长的。从图 3 中可以看出，以第 99 百分位的延迟为例，经过多级之后尾延迟是线性增加的；而当其中任意一个服务器的服务率变化时都不会影响其他阶段的逗留时间，但是会影响整体系统的逗留时间。那么，当任意一级的尾延迟发生变化时其是如何影响整体的？

我们从多组参数、多组实验中观察到，当任意一级的尾延迟变成原来的 k 倍且 k 大于某一值 K 时，该级的尾延迟和整体的尾延迟的差值几乎固定不变(如图 4 中 delta 曲线所示)，因此该级的尾延迟占整体尾延迟的比例将越来越大。那么当串联系统中任意一级服务器的尾延迟变化较大时，就可以用该级的尾延迟来近似预测整体的尾延迟。

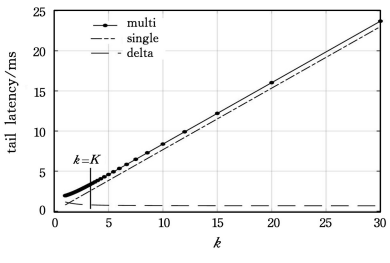

图 4　串联系统中单级变化倍数 k 和尾延迟的关系

Fig. 4　Relationship between single-level changes k and tail latency in series system

从图 4 可以观察到，存在某一个 K，当 $k<K$ 时，整体尾延迟并不随着 k 的线性增长而呈比例变化；只有当 $k\geqslant K$ 时整体尾延迟才会随着 k 成比例变化。即当单级尾延迟变化超过 K 倍时，我们才可以用该级的尾延迟来近似预测整体的尾延迟。

我们经过多组参数的多次计算和实验得出，当 n 固定时，K 值和 $\mu-\lambda$ 有关。从图 5 可以看出，在 5 级串联系统中，服务率 μ 固定，到达率 λ 依次减小，即 $\mu-\lambda$ 在变大，此时对应的 K 在变小($K_1>K_2>K_3$)。

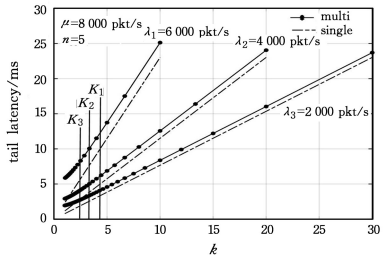

图 5　串联系统中尾延迟和单级变化倍数 k、到达率的关系

Fig. 5　Relationship between tail latency and single-level changes k and arriving rate in series system

结论 2　当服务率 μ 固定，服务率和到达率的差值 $\mu-\lambda$ 越大时，多级和单级的曲线越早开始平行变化，即 K 值越小。当 $\mu-\lambda$ 固定不变时，n 越大，单级变化越不容易影响整体延迟，即 K 越大。

具体是否可以用表达式来解释 n，$(\mu-\lambda)$ 和 K 的关系将是我们后续研究的重点。

3.3　并行排队模型的尾延迟分布

并行排队模型在网络服务系统中非常常见。在大规模分布式系统中存在着大量扇出的场景，根服务器节点通过中间服务器将请求分发到下层的叶子服务器节点，当所有结果都返回时才会将结果整合返回给用户。例如 Facebook 的 Web 请求可能访问数千个 Memcached 服务器[7]，Bing 搜索可以访问 10000 个索引服务器[2]，系统需要等待所有的服务器都处理完成之后才能将结果返回给用户。在这样的集群规模下，即使有极少数的服务器请求响应时间较长，也有很大概率导致用户请求的响应时间变长。

当有多个服务器并行运行时，一个请求的完成时间取决于最慢的服务器[1,12]。本文将讨论这种场景下尾延迟的分布。

3.3.1　并行场景分析

下面计算多个服务器并行的逗留时间的分布。每个服务器(都看作一个 M/M/1 排队模型)的累积分布函数为：

$$\begin{aligned}&F_{X_1}(t)=1-\mathrm{e}^{-(\mu_1-\lambda)t}\\&F_{X_2}(t)=1-\mathrm{e}^{-(\mu_2-\lambda)t}\\&\quad\vdots\\&F_{X_n}(t)=1-\mathrm{e}^{-(\mu_n-\lambda)t}\end{aligned}\tag{7}$$

由于一个请求的逗留时间取决于最慢的服务器，因此令 $Z=\max(X_1,X_2,\cdots,X_n)$，由此可以得到并行 n 个 M/M/1 排队模型逗留时间的分布为：

$$\begin{aligned}P\{Z\leqslant t\}&=P\{\max(X_1,X_2,\cdots,X_n)\leqslant t\}\\&=P\{X_1\leqslant t,X_2\leqslant t,\cdots,X_n\leqslant t\}\end{aligned}$$

$$=F_{X_1}(t)\times F_{X_2}(t)\times\cdots\times F_{X_n}(t)$$

$$=\prod_{i=1}^{n}(1-e^{-(\mu_i-\lambda)t}),t\geqslant 0 \tag{8}$$

3.3.2 相同参数的并行排队模型的尾延迟分布

如果有 n 个相同服务器并行处理请求，即 $\mu_1=\mu_2=\cdots=\mu_n=\mu$，那么此时逗留时间的累积分布函数为：

$$P\{Z\leqslant t\}=(1-e^{-(\mu-\lambda)t})^n,t\geqslant 0 \tag{9}$$

即 n 个相同的 M/M/1 排队模型并行之后的逗留时间的累积分布是单个队列的累积分布的 n 次方。

由此，可以得到 n 个相同的服务器并行处理请求时第 s 百分位的延迟为：

$$t_n=-\frac{\ln(1-\sqrt[n]{s/100})}{\mu-\lambda} \tag{10}$$

同样可以得出，n 个服务器并行在第 s 百分位下的逗留时间 t_n 与单个服务器对应的逗留时间 t_1 的比值，为：

$$\frac{t_n}{t_1}=\frac{\ln(1-\sqrt[n]{s/100})}{\ln(1-s/100)} \tag{11}$$

可以看出，当多个相同的服务器并行时，第 s 百分位的延迟与单个服务器第 s 百分位延迟的比值与到达率 λ 和服务率 μ 无关，只与 s 百分位和并行数 n 有关。例如，当 $s=99$ 时，随着并行数增加，并行队列与单个队列的比值如图 6 所示。

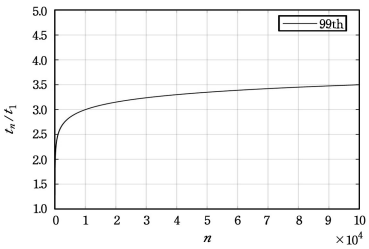

图 6 多级相同并行系统中整体尾延迟与单级的比值

Fig. 6 Ratio of overall tail latency to single-level in multi-level identical parallel system

由式(10)可知，并行的相同的服务器越多，系统的尾延迟会越长，但是从图 6 可以看出，尾延迟的增长速度和服务器并行数的增长速度并不成比例，尾延迟的增长速度远低于并行数的增长速度。当有 10 个服务器并行时，整体第 99 百分位的尾延迟是单个服务器的 1.499 倍；当有 100 个服务器并行时，整体第 99 百分位的尾延迟仅增长到单级的 1.999 倍；当有 10000 个服务器并行时，仅变成原来的 3 倍左右。这与人们的并行越多服务器，尾延迟增长得越剧烈的直觉相差甚远。

3.3.3 单级服务器尾延迟发生变化时对系统尾延迟的影响

正如前面提到，在大规模分布式系统中，即使有少数服务器请求响应时间较长，也有很大概率导致用户请求的响应时间变长，因此本文将量化这种变化对系统整体尾延迟的影响。我们考虑 n 个相同的 M/M/1 排队模型并行时，如果其中任意一个服务器的尾延迟变大，如变为原来的 k 倍，对系统的整体的尾延迟有什么影响？

利用 3.3.1 节得到的并行系统逗留时间的累积分布式(8)，计算多组情况来对比单级服务器变化 k 倍之后的尾延迟和整体尾延迟。我们发现，当单级服务器变化超过一定程度(变化倍数 k 超过某一值 K)时，该级服务器的尾延迟与整体的尾延迟近似相等。这是因为并行服务系统的尾延迟是所有服务器当中最长的尾延迟，当有任意一个服务器的尾延迟剧烈波动时，它将有很大可能性成为系统整体的尾延迟。图 7 给出一个具体的示例，可以发现当单级变化倍数超过 5 倍时，系统的整体尾延迟和单级尾延迟近似相等。

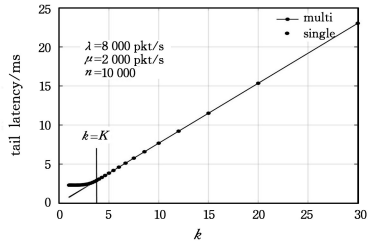

图 7 多级并行系统中尾延迟和单级变化倍数 k 的关系

Fig. 7 Relationship between tail latency and single-level changes k in multi-level parallel system

结论 3 当 n 个服务率均为 μ 的并行服务系统中，到达率为 λ，假如并行服务系统中任意一个服务器的尾延迟变为原来的 $k(k\geqslant K)$ 倍，整体的尾延迟变化倍数为：

$$k_{整体}=\frac{t_{现整体}}{t_{原整体}}=k\frac{\ln(1-s/100)}{\ln(1-\sqrt[n]{s/100})} \tag{12}$$

例如，在 10000 个到达率为 $\lambda=2000$ pkt/s，服务率为 $\mu=8000$ pkt/s 的服务器并行的初始情况下，根据前述结论，当其中任意一个服务器的延迟变为原来的 $k(k\geqslant K)$ 倍时，系统的整体的延迟将变为原来的 $k/3$ 倍。即当某一级尾延迟变为原来的 100 倍时，整体延迟变成原来的 33.3 倍；某一级尾延迟扩大 200 倍时，整体延迟则扩大为原来的 66.7 倍。

由此可以看出，当只有一个服务器在处理请求无法应对网络环境中的波动时，尾延迟可能有剧烈变化，这时并行多个服务器可以减少整体延迟。

从图 7 可以看出，存在一个值 K，当 $k<K$ 时，单级变化和整体变化并不成比例。只有当 $k\geqslant K$ 之后，单级和整体才会成比例变化。下面将讨论影响 K 大小的因素。

我们利用式(7)做了多组参数的计算和实验，发现 K 与 $\mu-\lambda$ 和 n 有关。

结论 4 当系统级数 n 固定，单级尾延迟变化倍数 k 超过一定阈值 K 后，该单级尾延迟接近系统整体尾延迟。该 K 值的大小和服务率与到达率的差值有关，即($\mu-\lambda$)越大，K 越小。

结论 5 当服务率与到达率的差值($\mu-\lambda$)固定时，系统级数 n 越大，单级尾延迟变化 k 倍后会对系统整体尾延迟产生成比例变化的阈值 K 越大。

针对结论 5 我们做了进一步的研究。改变多组 μ 和 λ 参数，并分别进行多组实验，观察到 n 和 K 有近似对数的关系。我们仍以图 7 的参数 $\lambda=2000$ pkt/s，$\mu=8000$ pkt/s 为例，改变并行数 n，取满足单级尾延迟和整体尾延迟相等的最小的 k 为 K，得到 n 和 K 的关系如图 8 所示。具体是否可以用表达式来解释 n 和 K 的关系将是我们后续研究的重点。

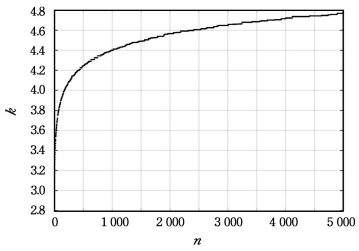

图 8　系统级数 n 和 K 的关系

Fig. 8　Relationship between system level n and K

由以上的计算和实验可知系统中某一个子部件变化对整体尾延迟变化的影响程度。因此，当探测到系统中的某个服务器受网络环境影响而导致尾延迟变高时，可以暂时将该服务器从系统中摘除，等它恢复之后再加入到系统中。

表 2 总结了上述串联、并行场景中得出的结论。通过式(6)可以得出，当多级相同参数模型串联时，70 百分位的尾延迟约等于系统平均延迟；当代入特定值求出尾延迟数值则可以看出，在多级串联和多级并联中，当其中有一级尾延迟变为原来的数倍，整体尾延迟变化都分为两个阶段：当该级变化较小时，对整体变化没有较大变化影响；当变化超过某个阈值 K 时，单级和整体的尾延迟将成比例变化。

表 2　串联、并行结论总结表

Table 2　Summary table of series and parallel conclusions

场景		参数	第 s 百分位的尾延迟
1. M/M/1 排队模型		到达率 λ，服务率 μ	$-\frac{\ln(1-s/100)}{\mu-\lambda}$
2. 多级串联	2.1 多级相同 M/M/1 串联模型	级数 n，到达率 $\lambda_1=\lambda_2=\cdots=\lambda_n=\lambda$，服务率 $\mu_1=\mu_2=\cdots=\mu_n=\mu$	$-\frac{\ln(1-s/100)}{\mu-\lambda}m$，$m$ 为由式(6)代入不同的百分位数 s 和级数 n 得出 t_n 和 t_1 的比值，当 $s>0.7$ 时，$m<n$
	2.2 多级相同串联模型中改变其中一级	场景 2.1 中某一级的尾延迟变为原来的 k 倍	$\approx-\frac{\ln(1-s/100)}{\mu-\lambda}k(k\geqslant K)$，存在一个值 K，当 $k\geqslant K$ 时单级和整体成比例变化；$k<K$ 时不成比例变化
3. 多级并联	3.1 多级相同 M/M/1 并行模型	并行数 n，到达率 λ，服务率 $\mu_1=\mu_2=\cdots=\mu_n=\mu$	$-\frac{\ln(1-\sqrt[n]{s/100})}{\mu-\lambda}$
	3.2 多级相同并行模型中改变其中一级	场景 3.1 中某一级的尾延迟变为原来的 k 倍	$-\frac{\ln(1-s/100)}{\mu-\lambda}k(k\geqslant K)$，存在一个值 K，当 $k\geqslant K$ 时单级尾延迟和整体尾延迟相等；$k<K$ 时不相等

3.4　小型网络举例

本文以图 9 的简单网络为例来验证上述模型的正确性。该网络由 10 条并行路径组成，其中一条路径由 10 个服务器串联组成，其他路径均为 1 个服务器，该小型网络的请求服从参数为 λ 的泊松分布，每一个服务器的服务时间都遵从参数为 μ 的指数分布。根据前面串联、并行中的结论，由图 3 可知，10 级串联的排队模型第 99 百分位的延迟是单级串联的 4.079 倍。再由并行的结论可以得出，10 级并行模型的第 99 百分位延迟是单级 M/M/1 第 99 百分位延迟的 1.499 ($1.499=\frac{\ln(1-\sqrt[10]{0.99})}{\ln(1-0.99)}$)倍。因此该网络整体的第 99 百分位的延迟是单级 M/M/1 模型第 99 百分位延迟的 4.079 倍，是 10 级并行模型(到达率为 λ，服务率为 μ)的 $\frac{4.079}{1.499}=2.72$ 倍。

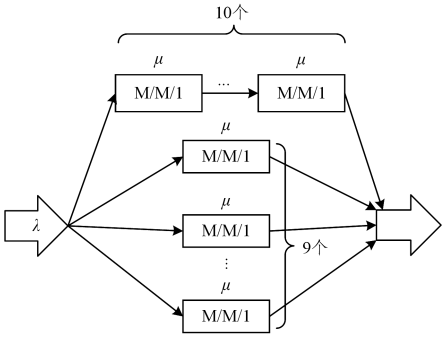

图 9　小型网络实例

Fig. 9　Small network instance

下面在服务率 $\mu=8000$ pkt/s 的情况下改变到达率 λ，利用上述结论计算第 99 百分位的理论延迟，并用 Matlab 模拟该小型网络，得出相同情况下实际的第 99 百分位的延迟，两者的对比情况如图 10 所示。

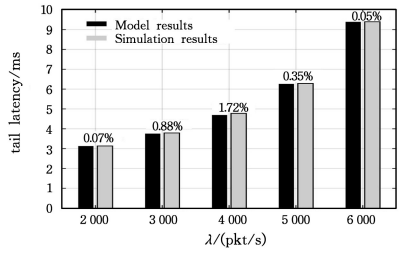

图 10　小型网络的模型预测结果和仿真结果对比(1)

Fig. 10　Comparison of model predicted results and simulation results for small networks(1)

下面在到达率 $\lambda=2000$ pkt/s、服务率 $\mu=8000$pkt/s 时，改变串联路径中一个服务器的服务率，再利用上述结论计算第 99 百分位的理论延迟，并与 Matlab 模拟结果作对比，结果如图 11 所示。

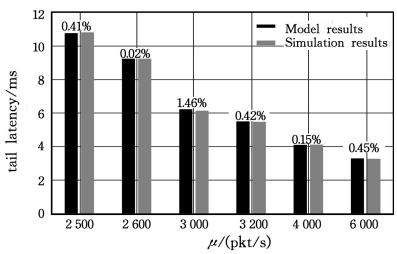

图 11　小型网络的模型预测结果和仿真结果对比(2)

Fig. 11　Comparison of model predicted results and simulation results for small networks (2)

从图 10 和图 11 可以看出，用 Matlab 仿真该小型网络得到的仿真结果和由模型结论预测得到的结果基本一致，两者误差不超过 2%。

4 相关工作

4.1 尾延迟的影响因素

数据中心服务现在通常根据其第 99 或第 99.9 的延迟进行评估。为了提供快速响应，网络服务组件中响应的中位数和尾部都必须很低。然而，网络服务系统中的各种因素导致了尾延迟的变化。Dean 等[1]详细讨论了延迟发生抖动的原因，并在文中描述了他们在谷歌交互式应用程序中为了控制尾延迟所做的努力，其目标是让并行系统能够消除单个组件的延迟变化，用不太可能预测的部分创建一个可预测的响应整体。他们也介绍了减少尾部延迟的基本方法——冗余请求。文献[13-15]描述了如何利用冗余请求的技术降低尾延迟以及在此基础上的优化。

文献[16]从硬件、操作系统和应用程序 3 个层次探讨了延迟变化的原因，并将服务器建模为单队列模型，分别研究了服务器的利用率、并行性以及恰当的排队规则对延迟的影响。本文在 3.1 节也从理论上说明了利用率对单个服务器尾延迟的影响。

4.2 对尾延迟的建模分析

排队论中提供了一个描述服务对象到达率、服务率和端到端的逗留时间之间关系的框架，如最简单的 M/M/1 排队模型就描述了按泊松分布到达、服务时间服从指数分布的排队过程。Jackson 排队网络[17]则是描述了具有 m 个相连队列，任意节点的外部到达过程为泊松过程，服务时间为指数分布，队列 i 处的顾客服务结束之后，将以概率 P_{ij} 转移到新的队列 j 或以概率 $1-\sum_{j=1}^{m}P_{ij}$ 离开队列的开环网络。Jackson 排队网络的均衡分布计算形式简单且网络具有积形式解。但是排队论中对于系统的性能度量通常得到的是平均值的结论，而对尾延迟的理论研究少之又少。

虽然缺乏针对尾延迟指标的理论研究，但是随着网络服务系统中尾延迟逐渐成为一个重要的评价指标，针对尾延迟建模预测的模型也越来越多。

在高可扩展的互联网服务场景中，一个请求通常涉及任务的划分和合并，从而难以预测请求的尾部延迟来满足严格的服务目标，文献[18]通过仅使用任务响应时间的均值和方差作为输入的预测模型，来预测高负载区域请求的尾延迟。一些文献则考虑使用 Fork-Join 结构来模拟数据中心中应用程序的工作流的基础结构。Fork-Join 排队网络模型在排队论[19]中是出了名的难解，以往的研究大多是试图找到作业平均响应时间[20-21]或边界[22-23]的近似值。随着一些延迟敏感应用的出现，尾部延迟越来越受到人们的关注。最近一些研究[24-27]试图为 Fork-Join 排队网络的尾部延迟提供解析解。

文献[10]则研究了尾延迟如何影响硬件设计，它通过一个基本排队理论的 M/M/k 模型来补充 Amdahl 定律对并行、多核系统的分析。

文献[10,16,18]都利用了排队理论对服务器进行建模来预测响应时间，但是本文在把单个服务器建模为 M/M/1 排队模型的基础上，将整个网络服务系统抽象成排队模型，考虑不同服务器的串联或并行，在已知请求的到达率和服务器的服务率后，利用数学工具给出明确的逗留时间的分布函数，可以方便地求出特定百分位的延迟，更加方便分析复杂的网络服务系统的尾延迟特征。同时，本文还着重分析了当任意子部件发生变化时对网络服务系统的尾延迟的影响。

结束语 尾延迟逐渐在网络服务系统设计中成为一个关注点，较高的尾延迟不但会影响用户体验，还会减少内容供应商的收益和用户数量。而随着网络规模的扩大和系统结构的复杂性提高，想要提供可预测的低延迟服务变得越来越困难。并且由于现有研究的复杂性，复杂网络中的延迟时间计算缺乏形式化的建模和计算方法。本文首先将网络服务系统抽象为一个基于 M/M/1 队列串联、并行组合的排队网络模型；然后回顾了 M/M/1 排队模型中逗留时间的分布函数，并据此得出尾延迟的表达式；在此基础之上分别给出了在多级 M/M/1 串联、并行场景下逗留时间分布的计算方法和逗留时间分布的表达式，并分析了尾延迟的特征。此外，我们着重分析了当任意一级服务器的尾延迟发生变化时对整体尾延迟的影响。本文利用 Matlab 工具仿真了处理过程，理论分析和实验结果都充分说明了模型是合理、有效的。同时本文利用 Matlab 工具来模拟不同场景下逗留时间的分布，分析了一些典型情况下的尾延迟，并将上述结论应用到小型网络中求解比较复杂网络的尾延迟，本文的计算方法和模拟过程的误差不超过 2%。

随着互联网的发展，网络服务系统的规模扩大且复杂性提高，处理网络不仅包含多个处理器串联或者并行，可能会有反馈或者环等结构，或者队列中请求带有优先级等更加复杂的因素，是多种不同类型的网络相互作用、相互连接的结果。未来可以将本文计算方法继续应用到更加复杂的网络服务系统中，例如考虑到达系统的请求的优先级或者系统中存在的反馈等信息，建立更加精确的模型来分析网络延迟的尾部特征，为系统设计提供一些支持和建议。

参 考 文 献

[1] DEAN J, BARROSO L A. The tail at scale[J]. Communications of the Acm, 2013, 56(2): 74-80.

[2] JALAPARTI V, BODIK P, KANDULA S, et al. Speeding up distributed request-response workflows[J]. ACM SIGCOMM Computer Communication Review, 2013, 43(4): 219-230.

[3] DECANDIA G, HASTORUN D, JAMPANI M, et al. Dynamo: amazon's highly available key-value store[J]. ACM SIGOPS Operating Systems Review, 2007, 41(6): 205-220.

[4] HE Y, ELNIKETY S, LARUS J, et al. Zeta: Scheduling interactive services with partial execution[C]// Proceedings of the Third ACM Symposium on Cloud Computing. 2012: 1-14.

[5] YI J, MAGHOUL F, PEDERSEN J. Deciphering mobile search patterns: a study of yahoo! mobile search queries[C]// Proceedings of the 17th International Conference on World Wide Web. 2008: 257-266.

[6] XU Y, MUSGRAVE Z, NOBLE B, et al. Bobtail: Avoiding Long

Tails in the Cloud[C]// Proceedings of the 10th USENIX Conference on Networked Systems Design and Implementation. USENIX Association,2013:329-342.

[7] NISHTALA R,FUGAL H,GRIMM S,et al. Scaling Memcache at Facebook[C]// Usenix Conference on Networked Systems Design and Implementation. USENIX Association, 2013: 385-398.

[8] BARROSO L A,DEAN J,HOLZLE U. Web search for a planet:The Google cluster architecture[J]. IEEE Micro, 2003, 23(2):22-28.

[9] SURESH L,CANINI M,SCHMID S,et al. C3:Cutting tail latency in cloud data stores via adaptive replica selection[C]// 12th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 15). 2015:513-527.

[10] DELIMITROU C,KOZYRAKIS C. Amdahl's law for tail latency[J]. Communications of the ACM,2018,61(8):65-72.

[11] ALESAWI S,NGUYEN M,CHE H,et al. Tail Latency Prediction for Datacenter Applications in Consolidated Environments [C]// 2019 International Conference on Computing,Networking and Communications (ICNC). IEEE,2019:265-269.

[12] MIRHOSSEINI A,WENISCH T F. The queuing-first approach for tail management of interactive services[J]. IEEE Micro, 2019,39(4):55-64.

[13] LUMB C R,GANGER G R,GOLDING R. D-SPTF:Decentralized Request Distribution in Brick-based Storage (CMU-CS-03-202)[J]. Acm Sigops Operating Systems Review,2004,39(11): 37-47.

[14] VULIMIRI A,GODFREY P B,MITTAL R,et al. Low latency via redundancy[C]// Acm Conference on Emerging Networking Experiments & Technologies. ACM,2013.

[15] WU Z,YU C,MADHYASTHA H V. CosTLO:Cost-Effective Redundancy for Lower Latency Variance on Cloud Storage Services[C]// Usenix Conference on Networked Systems Design & Implementation. USENIX Association,2015.

[16] LI J,SHARMA N K,PORTS D R K,et al. Tales of the tail: Hardware,os,and application-level sources of tail latency[C]// Proceedings of the ACM Symposium on Cloud Computing. ACM,2014:1-14.

[17] JACKSON J R. Networks of waiting lines[J]. OPER. RES., 1957,5(4):518-521.

[18] NGUYEN M,LI Z,DUAN F,et al. The tail at scale:how to predict it? [C]// Usenix Conference on Hot Topics in Cloud Computing. USENIX Association,2016.

[19] THOMASIAN A. Analysis of fork/join and related queueing systems[J]. ACM Computing Surveys (CSUR),2014,47(2):1-71.

[20] NELSON R,TANTAWI A N. Approximate analysis of fork/join synchronization in parallel queues[J]. IEEE transactions on computers,1988,37(6):739-743.

[21] CHEN R J. A hybrid solution of fork / join synchronization in parallel queues[J]. IEEE Transactions on Parallel and Distributed Systems,2001,12(8):829-845.

[22] BALSAMO S,DONATIELLO L,VAN DIJK N M. Bound performance models of heterogeneous parallel processing systems [J]. IEEE transactions on parallel and distributed systems, 1998,9(10):1041-1056.

[23] CHEN R J. An upper bound solution for homogeneous fork/join queuing systems[J]. IEEE Transactions on Parallel and Distributed Systems,2010,22(5):874-878.

[24] RIZK A,POLOCZEK F,CIUCU F. Computable bounds in fork-join queueing systems[C]// Proceedings of the 2015 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems. 2015:335-346.

[25] NGUYEN M,ALESAWI S,LI N,et al. Forktail:a black-box fork-join tail latency prediction model for user-facing datacenter workloads[C]// Proceedings of the 27th International Symposium on High-Performance Parallel and Distributed Computing. 2018:206-217.

[26] QIU Z,PÉREZ J F,HARRISON P G. Beyond the mean in fork-join queues:Efficient approximation for response-time tails[J]. Performance Evaluation,2015,91:99-116.

GUO Zi-ting, born in 1995, postgraduate. Her main research interests include data center networks and so on.

ZHANG Wen-li, born in 1980, Ph.D, is a senior member of China Computer Federation. Her main research interests include architecture and algorithm optimization for high end computers and datacenter networks.