计算机研究与发展
Journal of Computer Research and Development
DOI:10.7544/issn1000-1239.2018.20170793
55(3):512-523,2018

边缘计算环境下应用驱动的网络延迟测量与优化技术

符永铨　　李东升
(国防科技大学计算机学院　长沙　410073)
(国防科技大学并行与分布处理重点实验室　长沙　410073)
(yongquanf@nudt.edu.cn)

Application Driven Network Latency Measurement Analysis and Optimization Techniques Edge Computing Environment: A Survey

Fu Yongquan and Li Dongsheng
(*College of Computer, National University of Defense Technology, Changsha* 410073)
(*Science and Technology on Parallel and Distributed Laboratory, National University of Defense Technology, Changsha* 410073)

Abstract　The technical advancements of Internet, mobile computing and Internet of things (IoT) have been pushing the deep integration of human, machine and things, which fostered a lot of end-users oriented network search, online social networks, economical business, video surveillance and intelligent assistant tools, which are typically referred to as online data-intensive applications. These new applications are of large scale and sensitive to the service quality, requiring stringent latency performance. However, end-user requests traverse heterogeneous environments including edge network, wide-area network and the data center, which naturally incurs a long-tail latency issue that significantly degrades users' experience quality. This paper surveys architectural characteristics of edge-computing applications, analyzes causes of the long-tail latency issue, categorizes key theories and methods of the network latency measurement, summarizes long-tail latency optimization techniques, and finally proposes the idea of constructing an online optimization runtime environment and discusses some open challenges.

Key words　edge network; edge-computing applications; long-tail latency; measurement; optimization

摘　要　互联网、移动计算、物联网技术的进步推动了人-机-物环境的深度融合,催生了一大批面向边缘用户的网络搜索、在线社会网络、电子商务、视频监控、智能助理等类型的边缘计算应用.边缘计算应用具有规模巨大、服务质量敏感等特性,对延迟性能提出迫切需求,然而,由于用户访问请求跨边缘网络、广域网、数据中心异构环境,"长尾延迟"问题导致边缘用户的体验质量严重下降.首先综述边缘计算应用的系统架构特征,然后分析长尾延迟的产生原因,分类介绍网络延迟测量的主要理论和方法,并归纳对长尾延迟的优化技术,最后提出在线优化运行环境的思想以及面临的挑战.

关键词　边缘网络;边缘计算应用;长尾延迟;测量;优化

中图法分类号　TP391

收稿日期:2017-10-23;**修回日期**:2017-12-11

基金项目:国家"九七三"重点基础研究发展计划基金项目(2014CB340303);国家自然科学基金项目(61402509)
This work was supported by the National Basic Research Program of China (973 Program) (2014CB340303) and the National Natural Science Foundation of China (61402509).

通信作者:李东升(dsli@nudt.edu.cn)

互联网、移动计算、物联网技术的进步构成了深度融合的人-机-物生态环境,带动产生了一大批面向边缘用户的商业应用[1],形成了网络搜索、在线社会网络、电子商务、视频监控、智能助理等在内的亿级规模的边缘用户市场.思科[2]预计至2021年全球无线和移动智能设备流量占比将超过60%.然而,在对大规模数据集进行在线实时计算的数据密集型应用场景中,云计算遇到严重的"长尾"(long tail)延迟问题:即存在一定比例的处理响应时间远大于所有处理时间的平均值,导致平均响应延迟无法反映网络延迟的极端边界情况[3-4].Google[3]发现其99.9百分位数的延迟是中位数延迟的多个数量级,这说明:1 000个用户中至少有一个经历过高的延迟,显著影响用户体验[5].据统计[6],Google网络延迟每增加0.5 s,其网络流量将下降20%;Amazon服务延迟若增加0.1 s其利润将下降1%.同时,响应时间增长已经影响了在线网络服务在Google搜索的排名顺序.

为解决云计算的不足,边缘计算[7-8]得到了学术界和工业界的广泛关注.与云计算集中式资源部署模式不同,边缘计算[7]在靠近用户的边缘位置部署设备资源以及利用用户的相互协作执行部分或者全部的数据过滤与压缩、计算、存储、通信和管理等任务,从而提高实时处理响应速度,降低网络传输带宽开销,增强数据隐私保护能力.预计到2020年将有超过50%的数据需要在边缘网络分析、处理与存储[8].

用户感受到的延迟取决于请求-响应全过程的延迟.根据计算机领域的Amdahl定律可知,边缘计算应用任何一个位置的延迟上升必然增大端到端延迟,因此,需要结合云计算和边缘计算技术对边缘计算应用全流程进行控制和优化.

在线实时大数据长尾延迟优化已经引起了学术和工业界的极大兴趣.例如,当部分节点和资源出现严重的长尾延迟现象时,如何调度关键节点和资源迅速承担信息处理任务,确保在线计算的延迟有界保障?当边缘计算应用对运行环境的需求发生改变时,如何通过灵活地增加或调整节点资源及其配置满足计算规模需求?已有的研究工作通常采取启发式方法对特定的应用进行延迟优化,其可扩展性、灵活性和适用性严重不足,对应用进行延迟优化的经济与计算成本居高不下.为了优化广域网延迟,应用提供商需要花费巨额成本将应用部署到多个数据中心或内容分发网络[9-10],以便用户访问请求可被定向到相对邻近的数据中心(边缘服务器).在数据中心内部,为了降低TCP数据传输延迟,研究者[11-15]提出改善TCP拥塞控制机制和交换机(路由器)优先级调度机制,在期限内发送尽可能多的应用数据,但是需要显著改动数据中心的软硬件环境.为了避免资源复用造成应用在缓存、内存、硬件预取等资源受到的干扰[16-17],研究者[18-22]提出将低干扰的应用调度到相同的服务器,并利用优先级区分延迟敏感型任务与其他任务,但是由于应用普遍采取多副本技术提高服务的可用性[3,23],相互竞争资源的服务比例急剧上升,降低了调度算法的有效性.

1 边缘计算体系结构

边缘计算应用典型的三大特征包括[1,24]:

1) 大规模.边缘设备具有亿级规模,数据量级达到PB甚至更高,为了保持负载均衡,应用提供商通常利用公有云和私有数据中心以高可扩展方式进行在线计算;

2) 软实时.用户请求需要在一定的期限内返回,否则用户的体验质量(quality of experience, QoE)将严重受损;

3) 关联性.用户历史请求之间以及不同用户请求之间存在相关性.大规模和软实时对应用提出了严峻挑战,而关联性则为应用优化提供了参考.

边缘计算应用的用户负载多数呈现周期性变化,并具有突发性,如图1所示,典型的边缘计算应用大多采取中心、边缘、端多尺度架构[7,24-25]来分散

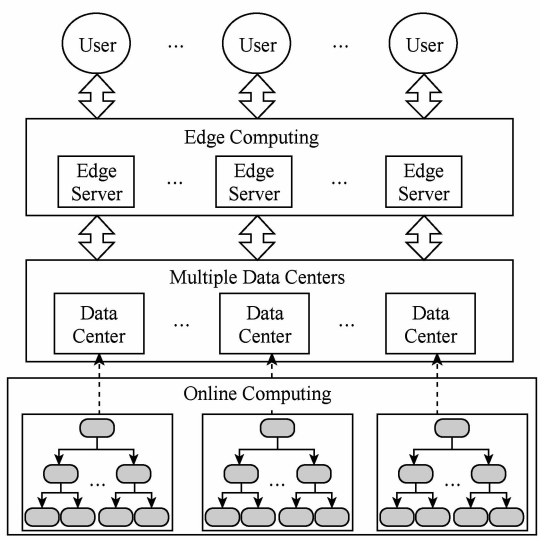

Fig. 1 Illustrative diagram of deployment architecture

图1 部署架构示意图

用户处理请求,并采取多副本技术避免单个网络位置产生性能瓶颈以获取较高的并发处理速度.

边缘计算应用的分布并行系统架构对长尾延迟优化带来严峻挑战.首先,服务提供者难以获得在线实时大数据计算结构中各个位置的网络延迟分布,难以快速发现长尾延迟严重区域并对造成长尾延迟的关键位置进行在线优化.其次,在消息路由和响应推送阶段,由于缺乏网络感知手段,广域网数据路由受到高延迟路由路径和数据传输丢包的影响,导致在线实时大数据请求遇到严重的长尾网络传输延迟.在线实时大数据树型或者有向无环图型在线计算结构在汇聚节点附近可能产生通信热点及链路拥塞,加剧了网络传输延迟的长尾问题.在边缘计算和在线处理阶段,在线实时大数据处理任务需要多层的分布并行计算,服务器性能波动和数据中心网络流量调度放大了长尾延迟的影响范围.例如,若要求100个服务器并行地处理请求,每一个服务器以99%的比例在1 s内处理完毕一个请求,而在1%的比例花费较多的时间,那么超过63%的在线实时大数据任务的完成时间将超过1 s:$(1-(1-1\%)^{100})>0.63$.

2 长尾延迟溯源分析

根据延迟产生的位置,边缘计算应用长尾网络延迟的原因可归纳为工作流、资源复用、网络拥塞、路由变化、虚拟化等方面,如图2所示:

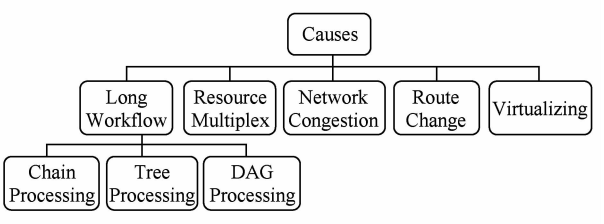

Fig. 2 Long-tail latency causes

图2 长尾延迟原因

1) 长工作流.边缘计算应用经常采用串行、树型拓扑或者有向无环图(directed acyclic graph, DAG)任务流来处理在线实时大数据请求,然而过长的工作流导致边缘计算应用面临长尾延迟的不足:

① 链式处理.在数据关联度高的环境下,大量数据访问不得不以顺序串行的形式实现.在线社交网络如Facebook和Twitter在生成用户页面时通常涉及100多个不同的用户数据查询.为了生成一个页面,Facebook平均需要在数据中心内部生成130个内部请求[26].这些内部请求因复杂的朋友链接关系是串行相关的,处理延迟则是内部请求延迟的累积.

② 树型处理.在数据之间关联性低的环境下,边缘计算应用通常以分解-会聚的方式处理在线请求.Google搜索引擎通过树型拓扑组织参与处理任务的组件服务器[25].Web服务器前端接收到查询请求后,将请求沿着由服务器组成的分布式树拓扑向下转发,最终分发到大量的叶节点.所有的叶节点以分布的方式共享搜索索引数据结构,每个叶节点负责在索引的一个分片上完成搜索任务.由于搜索索引中的大部分数据没有跨节点的副本,各个叶节点的结果都要返回,以最大化搜索结果的相关度.查询结果逐层会聚,最高评分的搜索结果返回给前端服务器.为了在期限内返回搜索结果,树型处理过程各个边均对应一定的期限,各个分支只有在都满足期限要求时,树型处理的完成时间才满足期限的要求.

③ DAG处理.通用的计算任务均可通过DAG表示[4].在有向无环图中,顶点对应一个功能单元,该功能单元可能对应一个或多个服务器、网络路由器、交换机、边代表功能单元之间的输入-输出依赖关系.Bing搜索引擎的DAG图具有15个顶点,其中10%的顶点在数千台服务器上并行地处理用户查询请求[4].DAG图拓扑能够显示出每个顶点的局部处理对端到端延迟的影响.DAG图上非关键路径的顶点对端到端延迟影响较小,优化这些顶点的处理延迟对降低端到端延迟帮助不大.其次,若顶点中任意服务器的处理延迟增大,那么端到端的查询响应延迟将显著上升.这是由于在线实时大数据关注尾部延迟,如99.99%的处理延迟.文献[4]指出,对于可并行运算的顶点数目为$n=100$或1 000的DAG图而言,各个服务器延迟的分布函数的99.99或99.999百分位数决定了一个具有n路并行顶点99百分位数的延迟值.

在树型拓扑和DAG中,相邻的顶点存在相关依赖关系,每个顶点通常包含多个服务器并发地处理任务.并行化工作流使得单个顶点的延迟波动加剧了长尾延迟效应.

2) 数据中心资源复用.为了充分利用数据中心闲置资源,对数据中心负载削峰填谷,在线实时大数据背景下的服务调度需要将服务部署到优化的服务器,达到既高效资源复用,同时又满足边缘计算应用

的低延迟需求.资源复用将多个任务部署到相同的服务器运行,通过服务整合有利于提高数据中心资源的利用率.资源复用要求隔离不同服务的运行环境,对用户程序的输入/输出(input/output, I/O)、内存、计算资源能够有效的额度控制.数据中心通常采取虚拟化管理、操作系统容器等技术来隔离内存、计算等资源,并利用分布式存储系统达到可扩展访存的目的.然而,不同应用可能出现相互干扰,导致无法获取期望的服务性能.例如缓存和 I/O 资源通常以共享方式为多个应用提供服务,操作系统调度策略可以产生严重的缓存和 I/O 请求干扰[27]:大多数的缓存(I/O)资源被少量的应用过度使用,导致其余的应用因无法得到所保障的资源份额,产生较大的处理延迟以及延迟波动.Mars 等人[18,28]发现由于异构环境对调度算法的影响,Google 工作负载延迟波动超过 40%,并且服务器从活跃到不活跃的节能状态切换导致服务器响应延迟显著上升.

3) 网络拥塞.不同的应用需要共享数据中心的网络带宽资源,然而数据中心网络仅限制不同应用的网络可达性,难以保障各个应用的网络性能,网络流在网络路径的不同位置竞争资源.短期的资源竞争可能产生网络拥塞,导致数据包丢失或重传,造成严重的长尾效应.同时,数据中心的层次拓扑结构链路利用率不均衡,容易在靠近核心部件的网络位置产生拥塞[29].数据包丢失将导致流传输超时,在数据中心内部,流的超时间隔通常设定为 10 ms.由于数据中心双向网络延迟通常在 250 μs 左右,短流的数据包超时将导致网络传输产生长尾,造成端到端网络延迟时间显著增大[12,30].设定短的超时间隔可以缓解短流的长尾问题,但是也会造成过多的网络重传,导致网络负载急剧增大.边缘计算应用的树型分割-会聚处理结构可能产生 TCP incast 问题[31-32]:一个分支的子女节点同时发送响应消息给父节点,造成父节点附近网络位置负载上升,产生相关的数据包丢失,造成流超时,引发长尾.

4) 路由变化.若边缘计算应用能够将用户请求定位到地理邻近的数据中心,那么不同用户请求经历的传播延迟大致相当.而服务依赖延迟、排队延迟、网络拥塞延迟、存储访问延迟具有较高的波动性.边缘计算应用的网络传输采取多阶段的 TCP 协议(用户-内容分发网络(content distribution network, CDN)阶段、内容分发网络-数据中心阶段、数据中心内部阶段)[33],通过持续性 TCP 会话避免建立 TCP 连接延迟,降低广域网网络拥塞对端到端数据传输速率的影响.广域网的网络延迟受到互联网路由动态性和负载波动性的影响不断地发生变化,导致在线实时大数据端到端延迟具有波动性[34].

5) 虚拟化.王国辉等人[35]发现虚拟化技术增大了传输层协议的双向网络延迟,虚拟机管理器调度延迟可达数百毫秒,远大于用户发送请求的传播延迟,显著增大了数据中心的端到端网络延迟.Amazon EC2 平台上长尾延迟现象较无虚拟化的数据中心更为严重[21],这是由于 Amazon EC2 平台对 CPU 计算密集型与延迟敏感型的任务协同调度,导致 VM 的长尾延迟要超过无虚拟化环境下 2~4 倍,而且较差的节点总是具有较高的长尾延迟.相同硬件配置下虚拟机之间的延迟差异可超过 1 个数量级.

3 理论模型

针对大规模节点对间的网络延迟,研究者发现网络延迟具有低维度、三角不等性违例、分簇等特点,提出了多种数学模型来建模静态的网络延迟矩阵,如图 3 所示.然而,已有的理论模型难以分析网络延迟的尾部特征,只适用于平均延迟.

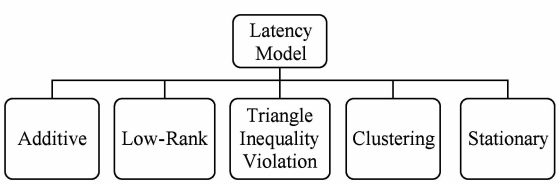

Fig. 3 Fundamental attributes of network latency

图 3 网络延迟的基本特点

1) 叠加性.从物理设施角度看,端到端网络延迟涉及操作系统内核、处理器、内存、存储相关的 I/O 延迟、网络相关的 I/O 延迟.从高层基础设施来看,端到端网络延迟包括 DNS 域名服务器、TCP 传输协议、Web 服务器、网络链路、路由器产生的延迟.由于延迟具有叠加性,任意节点位置的网络延迟增大都会加剧边缘计算应用的长尾延迟现象.

2) 低维度.网络延迟矩阵可以通过低秩(矩阵的秩指的是矩阵的线性无关的行向量的极大数目或者矩阵的线性无关的列向量的极大数目)的矩阵近似.研究者[36-40]证实网络延迟空间可以利用 2 至 9-维的矩阵近似.Abrahao 等人[41]根据度量空间中的分形维度(fractal measure of dimension)模型发现网络延迟空间的维度值不大于 2.Fraigniaud 等人[42]

根据低度量模型发现网络延迟空间的增量维度和倍增维度均比较低.符永铨等人扩展低度量模型至绕道路由环境[43],发现增量维度和倍增维度适中.

3) 三角不等性违例.即存在三元组(A,B,C)满足$d_{AB}>d_{BC}+d_{CA}$或者$d_{AB}<|d_{BC}-d_{CA}|$.研究证实[44]网络延迟空间包含一定比例的三角不等性违例.Savage 等人[45]测量了几十个广域网节点间的网络延迟,发现超过 20%的三元组包含三角不等性违例.Zheng 等人[34]指出网络延迟空间的三角不等性违例呈现持久性和广泛性.Lumezanu 等人[44]发现三角不等性违例随着时间动态的变化,并且与网络延迟计算方法有关,选择网络延迟测量结果的中间值或者平均值得到的三角不等性违例结果存在差异.

4) 分簇.节点集合存在由邻近节点组成的簇,使得簇内节点之间的网络距离低于簇间节点之间的网络距离.由于互联网呈现层次结构,网络延迟空间显示出显著的分簇特征[46].例如 Ramasubramanian 等人[47]发现广域网的网络延迟和网络带宽呈现层次的分簇结构.Zhang 等人[46]和 Lee 等人[40]通过 DNS 服务器间的网络延迟证实了网络延迟空间包含 3~4 个分簇,并且簇间节点间的平均距离大于或者等于簇内节点距离的 2.5 倍.网络延迟空间的分簇特征造成同一个簇内的节点到其他节点的网络延迟产生相关性,使得网络延迟空间可以通过低维度的网络坐标进行近似.此外,网络坐标方法通过感知分簇结构能够提高计算结果的精确性.例如 Zhu 等人[48]分别计算分簇内节点之间的网络延迟和分簇间节点之间的网络延迟,提高了网络坐标的精确度.

5) 稳态.学术界已经针对单条链路的延迟波动建立了统计模型.发现网络延迟在短期内呈现显著的稳态性.例如 Mukherjee[49]证实了网络延迟在短时间内呈现稳态性的特点,并且发现单个网络路径的双向网络延迟呈现偏移的伽马分布.Claffy 等人[50]发现互联网路由路径的单向延迟具有分层偏移特征,并且每一层稳定的时间通常以天为量级.Zhang 等人[51]则发现互联网路径延迟至少在分钟量级内是稳态的.

网络延迟空间的理论模型通过数学工具描述节点间的网络延迟状况,可以帮助研究者深入分析网络延迟空间的属性.已有的理论模型包括 2 种:基于拓扑空间的模型和基于几何空间的模型,二者不是相互独立的关系,而是存在重叠.

1) 基于拓扑空间的模型主要包括基于三角不等性假设的度量空间和允许三角不等性违例的非度量空间模型.早期的网络坐标方法[38,52]以及网络邻近度估计方法[53]主要通过度量空间分析网络延迟空间.而为了适应网络延迟空间的三角不等性违例的特点,研究者提出了低度量(inframetric model)模型[42,54].

2) 基于几何空间的模型主要包括欧氏空间、双曲空间、球面空间、向量空间等.欧氏空间、双曲空间、球面空间要求网络距离满足三角不等性,因此又属于度量空间范畴.而向量空间允许三角不等性违例和非对称性存在,因而不属于度量空间的范畴.网络坐标方法主要基于几何空间模型计算节点之间的网络延迟.向量空间和低度量空间对距离关系的约束条件最少;度量空间要求距离关系满足三角不等性条件,其表示能力弱于向量空间和低度量空间;欧氏空间、双曲空间和球面空间不仅假设三角不等性成立,而且限定了距离计算公式.

4 测量机制

网络延迟测量得到了研究者广泛的关注.已有的研究可以分为数据中心、内容分发网络、边缘网络等不同尺度范围的测量系统,如图 4 所示:

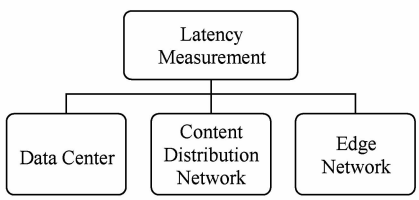

Fig. 4 Network latency measurement placements

图 4 网络延迟测部署位置

1) 数据中心.云平台测量通过收集底层设施的使用情况,为管理员管理提供决策依据,也为用户提供性能监测途径.Amazon CloudWatch 对 AWS 实例以及实例运行的进程进行监测,可以读取用户自定义监测指标,并允许用户设置警报.Amazon 中的 EC2 实例内嵌了 CloudWatch 基本监测功能,主要包括,CPU 利用率,数据传输,磁盘使用情况等.盛大监测宝支持 Linux/Unix 以及 Windows 服务器监测.它通过简单网络管理协议(simple network management protocol, SNMP)监测远程服务器的性能,而远程服务器需要配置 SNMP 监测代理,通过身份验证(支持 v2c 和 v3)向信任的节点提供 SNMP 查询信息.阿里云监测支持 Linux/Unix 和 Windows

平台的监测,能够对站点可用性和服务器性能进行监测,并提供短信报警支持.米海波等人[55]针对阿里云平台提出了细粒度的无监督性能诊断工具.针对网络应用通常包含多个网络流的特点,Lee 等人[56]提出了对属于相同应用会话的多个流进行优先级采样的方法,大幅提升网络流的分类精度.针对数据中心拓扑可控可知的特点,Peng 等人[57]提出了基于探测路径剪枝的数据中心网络监视方法,根据网络丢包模式对失效链路定位.Mace 等人[58]提出了针对分布式系统性能追踪的 pivot tracing 方法,通过对系统不同位置动态插桩收集并合并不同位置的统计数据,能够得到跨分布式系统组件的性能结构视图.

2)内容分发网络.内容分发网络在大量的网络中部署了边缘服务器,具有覆盖范围广、稳定性好的优势,因此成为理想的网络延迟测量平台.典型的基于内容分发网络的网络延迟测量系统是 Akamai 公司的 EdgePlatform 平台[59]和 Google 测量实验室的网络诊断工具[60]等.然而,由于内容分发网络的主要功能是向用户提供内容分发服务,过高的主动测量将会干扰内容分发业务的服务质量,因此内容分发网络难以利用充裕的带宽资源收集大量节点之间的网络延迟状况,导致网络延迟测量范围比较窄.

3)边缘网络.边缘网络可以通过主动或被动测量工具收集网络延迟.用户可以利用 Traceroute 工具探测逐个跳步的路由链路,根据路由器网络地址将路由路径相互连接得到一个全局连通的拓扑图,然后利用互联网地址路由规则[61]计算节点间的路由路径以及对应的网络延迟.典型的拓扑测量方法包括 iPlane[61],iPlane nano[62]和 Path Stiching[63].由于同一个路由器的不同端口具有不同的网络地址,同时路由器的端口映射关系是网络提供商的隐私信息,因此该方法面临着如何将端口地址映射到真实路由器的挑战性问题.此外,为了提高拓扑结构的覆盖范围,该方法需要测量大量的路由路径信息.基于域名服务器测量方法利用公开递归域名服务器测量到其他域名服务器的网络延迟,作为与域名服务器邻近的节点之间的网络延迟.典型的基于域名服务器的网络延迟值测量方法包括 King[64],Turbo-King[65]和 Internet Sibilla[66].由于查询者不需要维护任何的基础设施,因此这种方法的经济成本极低.然而,过高的查询请求增大了域名服务器的负载,降低了网络延迟测量的精确度,同时干扰了域名服务器的正常业务功能.Levchenko 等人[67]提出了一个通用的测量端点(例如软件代理、专用服务器、嵌入式系统)接口,将测量逻辑从端点移动到独立的实验控制服务器,降低了端点开销.

为了降低边缘网络测量开销,网络坐标方法利用互联网网络延迟呈现近似低维度[46]的特点,赋予边缘节点虚拟几何空间中的坐标位置,并根据坐标距离表示节点之间的网络延迟值.基于地标的网络坐标方法预先设定静态的地标节点,首先计算地标节点的网络坐标,然后根据地标节点的坐标位置计算非地标节点的坐标.典型的集中式网络坐标方法包括 GNP[38],IDES[68],ICS[69]等.基于分布计算的网络坐标方法随机选择逻辑邻居,然后定期地根据到逻辑邻居的网络延迟调整坐标.典型的分布式网络坐标方法包括 PIC[52],Vivaldi[70],DMF[71],Htrae[72],Phoenix[73],RMF[74]等.这种方法避免了性能瓶颈,并且允许节点动态地加入或者退出系统,因此适应大规模、能力受限的互联网用户.

5 延迟优化技术

边缘计算应用延迟优化具有巨大的潜在经济效益,因此研究者们提出了大量的延迟优化技术,根据延迟优化位置可以分为数据中心网络优化、内存计算、调度优化、路由优化、边缘缓存等方式,如图 5 所示:

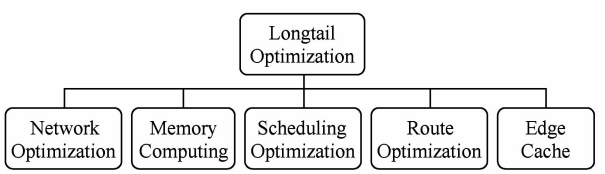

Fig. 5 Long-tail latency optimization mechanisms

图 5 长尾延迟优化方式

1)数据中心网络优化.随着 40Gbps+以太网、光网络、RDMA 传输技术发展,数据中心内部延迟逐渐逼近物理极限约 2 μs[75].网络延迟的降低提高了边缘计算应用的响应速度,在一定延迟界限下使用越来越多的服务器计算资源,发送更多的网络数据包.为了控制数据中心数据传输的延迟长尾问题,Wilson 等人[11]认为数据中心环境下的 TCP 协议需要具有期限感知能力,在端用户位置对每个延迟敏感的流根据流大小和延迟界限计算期望的速率,然后将速率信息发送给交换机,交换机根据会聚的速率值为每个流分配输出链路容量,不具有延迟界限的流以公平的方式共享空闲网络带宽.DCTCP[12]通过

TCP 协议的早期拥塞通知(early congestion notification, ECN)机制降低交换机队列长度,降低了短流的延迟,但是并不能够保证具有特定延迟约束的短流,也不提供对不同大小延迟要求的服务进行有区别的对待. PDQ[13]在数据中心交换机中部署近似最短任务优先、更早期限优先等流抢先策略. DeTail[14]以跨层的方式通过拥塞感知的数据包路由与链路层优先级排队机制,控制传输层端到端延迟. D^{2}TCP[15]扩展了 DCTCP 的窗口延迟函数,保证具有更小剩余延迟的流得到更多的传输速率. 李紫阳等人[76-77]针对商用服务器网络环境下的数据中心网络提出了上下文感知的流调度机制. Kalia 等人[78]针对传统单向 RDMA 事务处理系统灵活性和可扩展性受限的不足,提出基于双向不可靠数据报文的快速远程过程调用,显著提升了 RDMA 网络传输性能.

2) 内存计算. 为了支持数据中心的横向扩展能力,在线实时大数据所需的数据通常被划分到大量的数据中心服务器. 为了降低磁盘 I/O 访问的延迟,边缘计算应用体系结构的一个趋势是将磁盘中的数据迁移到内存,在数据中心服务器的内存存储并处理数据. 例如 Google 搜索算法已经基于内存放置搜索索引,Facebook 中大约 75%的非图片数据存储在内存中. 例如 Facebook,Google 等在数据中心内部广泛地部署了 memcached 分布式内存键值对(key-value, KV)存储系统. Facebook 在超过 800 台服务器规模部署了 memcached 系统,每秒支持超过 1500 万个访问请求[26]. 为了降低内存计算元数据开销,符永铨等人[79]提出了树型布鲁姆过滤器索引机制. 张一鸣等人[80-82]针对分布内存存储系统提出了快速的失效检测与恢复机制. Li 等人[83]利用商用 x86 架构服务器对分布式内存键值存储系统的硬件和软件进行一体化测量,发现采取预取和现代硬件特性(直接缓存访问 DCA、多队列网卡等)能够显著提升键值存储访问性能.

3) 调度优化. 数据中心资源分配已经得到了广泛关注,例如 Google[3]提出了在队列中区分服务等级,优先调度在线等待的处理任务,尽量缩短每个队列的长度,以提高调度过程的响应速度,CloudScale 针对应用行为难以预先评估的问题利用在线需求预测和预测错误处理来判断应用资源需求[84]. 为了测量资源复用下不同应用的相互干扰现象,Govindan 等人[16]根据服务器的物理内存地址信息提出了量化不同负载环境下缓存干扰程度的方法,Zhang 等人[17]利用指令周期数(cycles-per-instruction, CPI)量化不同负载的干扰程度,并通过关闭造成干扰的负载来保持各负载正常运行. 为了控制资源干扰,优化边缘计算应用的完成时间,Mars 等人[18]针对异构的数据中心环境提出了描述工作负载对内存压力敏感性的两阶段方法,通过组合优化方法为 Google 负载选择优化的服务器配置[28]. Nathuji 等人[19]提出了基于控制反馈的资源分配机制,消除资源复用下各个负载在缓存、内存、硬件预取等方面的干扰. Paragon 方法[20]根据协同过滤方法对跨资源的异构与干扰下的应用运行状态分类,将低干扰的应用调度到相同的服务器,达到低延迟响应与高效利用资源的目的. Bobtail 方法[21]根据 Amazon EC2 中较差的虚拟机总是具有较高的长尾延迟的现象,设计了无长尾虚拟机选择启发式方法. sparrow[22]利用批量采样技术分布式地调度延迟敏感型数据分析处理任务,采用虚拟预留技术让服务器分布式地确定任务启动时间,避免了任务耗时预计误差. 张钊宁等人[85-86]针对数据中心虚拟机集群提出了快速镜像部署调度机制. 针对数据中心虚拟机监视器易于部署的特点,Katta 等人[87]提出基于虚拟机监视器的流负载均衡方法,通过 traceroute 机制发现路径并持续监视路径拥塞状态,然后利用虚拟机监视器控制数据报文发送路径.

4) 路由优化. Amazon 公司的 AWS Route53 商业服务基于延迟测量将用户请求重定向到延迟较小的数据中心副本. DR2 方法[10]通过集中式矩阵分解预测用户与数据中心前端服务器的延迟,并选择具有最短延迟的数据中心作为用户请求的入口,从而降低了用户请求跨越公网的延迟. 为控制数据中心内部的请求处理时间,有必要降低在线实时大数据处理结构的关键路径上端到端延迟. Google[3,23]通过向 BigTable 系统多个副本服务器发送相同的请求,并将最快的响应返送给用户,能够将查询 1 000 个值的 99.9 百分位数延迟从 1 800 ms 降至 74 ms. 另一方面,副本技术也说明了长尾延迟容易受到随机干扰. Vulimiri 等人[88]根据服务组件副本集合以及 DAG 处理拓扑结构计算最短的处理拓扑路径,降低了用户请求的处理延迟. 然而,增加副本数量显著增大了数据中心的处理负载,可能加剧数据中心网络拥塞,增大队列的排队延迟. 针对广域网低延迟通信需求,李小勇等人[89]提出了带宽延迟协同优化的覆盖网路由机制,符永铨等人[43]提出了基于扩展低度量模型的近似最优分布式绕道路由机制.

5) 边缘缓存.复制和容错编码技术是提高容错的基本手段,也是降低在线实时大数据端到端延迟的常用技术.大型的边缘计算应用提供商通常部署(或租用)了内容分发网络,地理分布的数据中心来加速响应用户请求.为了降低用户请求到达数据中心的时间,在线实时大数据将用户请求定向到邻近的数据中心,边缘计算应用在一个数据中心内部完整的处理用户请求.在数据中心内部,处理过程触发在线实时大数据计算任务,同时需要读或者写存储层节点,用户的数据以异步复制的方式存放到其他数据中心,提供松弛的副本一致性保证[9].针对应急环境下低延迟数据分发的需求,马行空等人[90-93]提出了云数据中心辅助的分布式发布订阅机制.由于每个数据中心存放数据的完整副本增大了存储开销,为了在容错能力和存储开销之间取得优化的折中,裴晓强等人[94-95]针对分布式系统提出了基于纠删码的精简存储和快速修复机制.为了降低纠删码系统延迟,李慧霸等人[96]提出了基于推测的纠删码快速读写机制.

6 展 望

综合上述分析,目前国内外在低延迟在线优化技术方面已经初步成果.但总体说来,未能很好满足多样性边缘计算应用有界可控延迟的需求.

针对边缘计算应用多样性的特点,从提高在线实时大数据长尾延迟优化能力出发,支持多种边缘计算应用的正常运行成为未来的发展趋势.通过构建一个在线实时大数据在线优化运行环境,对边缘处理、消息路由、在线处理、响应推送阶段组成的端到端信息分发闭环进行一体化的测量与路由优化,确保用户访问时间有界可控:

首先,在边缘网络,运行环境将地理分布的缓存服务器组织为一个统一的边缘计算网络,缓存用户的在线计算任务与结果,若用户请求在缓存服务器命中,则缓存服务器直接将响应结果发送给用户,降低重复计算任务的冗余开销.

其次,在广域网范围,运行环境将多个地理区域的数据中心组织为统一透明的海量信息处理在线计算服务,各个数据中心在功能上互为备份,都提供相同的边缘计算应用服务,避免单个数据中心造成单点瓶颈,导致服务失效.运行环境将用户请求定向到邻近的可用数据中心,降低广域网传输延迟.

最后,在数据中心内部,运行环境根据计算需求,构建长尾延迟优化的计算架构,提高请求处理的并行程度,支撑快速完成计算任务.

为了支持运行环境的高效运行未来需要对长尾延迟测量、分析和优化问题进行深入研究:

1) 测量问题.在线实时大数据用户规模大,信息处理结构复杂,如何在不干扰在线实时大数据正常运行的前提下,收集海量用户与在线实时大数据之间的请求-响应延迟?为此,测量长尾延迟需要以允许的误差范围通过尽量低的测量开销收集广域网、数据中心网的延迟分布.同时测量过程因机器异常、网络拥塞、网络屏蔽等原因产生测量噪音或缺失,导致测量过程缺乏健壮性.因此,测量过程要容忍测量噪音,修复缺失的测量结果.

2) 建模问题.长尾延迟不仅是用户访问延迟的端到端特征,更是在线实时大数据计算结构各个分段的局部性特征.而且正是局部的长尾造成了更为严重的端到端的长尾延迟现象.如何设计可扩展的长尾延迟分布模型,建模带有复杂拓扑结构的多尺度长尾延迟特征?研究者已经提出单条路径的网络延迟分布函数模型,但是模型参数难以预先确定,无法对带有复杂拓扑结构的在线实时大数据做整体特征分析.现有的网络延迟模型具有确定的模型参数,适用于静态的平均延迟或最小延迟,但对长尾分布延迟不再有效.因此,需要研究新的长尾延迟理论模型,分析具有不变性的模型参数,指导长尾延迟的测量与优化研究.

3) 优化问题.用户访问在线实时大数据需要跨越边缘网络、广域网、数据中心网络等复杂异构网络环境.由于延迟具有叠加性,在线实时大数据处理任意位置的延迟增大势必导致端到端的长尾延迟,因此如何设计一体化的优化机制来控制端到端的在线实时大数据网络延迟成为必然选择.同时,广域网不在在线优化运行环境的控制范围内,需要探讨是否可以绕过高延迟的路由区域转发用户请求,而且控制网络传输过程中的网络拥塞对请求-响应延迟造成的影响.

参 考 文 献

[1] Ranganathan P. From microprocessors to nanostores: Rethinking data-centric systems [J]. IEEE Computer, 2011, 44(1): 39-48

[2] CiscoSystems Inc. Cisco visual networking index: Forecast and methodology, 2016—2021 [EB/OL]. [2017-10-08]. https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni

[3] Dean J, Barroso L A. The Tail at Scale [J]. Communications of the ACM, 2013, 56(2): 74-80

[4] Jalaparti V, Bodik P, Kandula S, et al. Speeding up distributed request-response workflows [C] //Proc of ACM SIGCOMM'13. New York: ACM, 2013: 219-230

[5] Chen Yingying, Mahajan R, Sridharan B, et al. A provider-side view of Web search response time [C] //Proc of ACM SIGCOMM'13. New York: ACM, 2013: 243-254

[6] Greenberg A G, Hamilton J R, Maltz D A, et al. The cost of a cloud: Research problems data center networks [J]. Computer Communication Review, 2009, 39(1): 68-73

[7] Shi Weisong, Sun Hui, Cao Jie, et al. Edge computing—An emerging computing model for the Internet of everything era [J]. Journal of Computer Research and Development, 2017, 54(5): 907-924 (in Chinese)
(施巍松, 孙辉, 曹杰, 等. 边缘计算: 万物互联时代新型计算模型[J]. 计算机研究与发展, 2017, 54(5): 907-924.)

[8] Edge Computing Consortium. White paper of edge computing consortium [EB/OL]. (2016-11-30) [2017-10-08]. http://ecconsortium.net (in Chinese)
(边缘计算产业联盟. 边缘计算产业联盟白皮书[EB/OL]. (2016-11-30) [2017-10-08]. http://ecconsortium.net)

[9] Lloyd W, Freedman M J, Kaminsky M, et al. Stronger semantics for low-latency geo-replicated storage [C] //Proc of USENIX NSDI'13. Berkeley, CA: USENIX Association, 2013: 313-328

[10] Zhu Jieming, Zheng Zibin, Lyu M R. DR2: Dynamic request routing for tolerating latency variability online cloud applications [C] //Proc of IEEE CLOUD'13. Piscataway, NJ: IEEE, 2013: 589-596

[11] Wilson C, Ballani H, Karagiannis T, et al. Better never than late: Meeting deadlines datacenter networks [C] //Proc of ACM SIGCOMM'11. New York: ACM, 2011: 50-61

[12] Alizadeh M, Javanmard A, Prabhakar B. Analysis of DCTCP: Stability, convergence, and fairness [C] //Proc of ACM SIGMETRICS'11. New York: ACM, 2011: 73-84

[13] Hong C Y, Caesar M, Godfrey P B. Finishing flows quickly with preemptive scheduling [C] //Proc of ACM SIGCOMM'12. New York: ACM, 2012: 127-138

[14] Zats D, Das T, Mohan P, et al. DeTail: Reducing the flow completion time tail datacenter networks [C] //Proc of ACM SIGCOMM'12. New York: ACM, 2012: 139-150

[15] Vamanan B, Hasan J, Vijaykumar T. Deadline-aware datacenter TCP (D2TCP) [C] //Proc of ACM SIGCOMM'12. New York: ACM, 2012: 115-126

[16] Govindan S, Liu Jie, Kansal A, et al. Cuanta: Quantifying effects of shared on-chip resource interference for consolidated virtual machines [C] //Proc of ACM SOCC'11. New York: ACM, 2011: Article No. 22

[17] Zhang Xiao, Tune E, Hagmann R, et al. CPI^2: CPU performance isolation for shared compute clusters [C] //Proc of EuroSys'13. New York: ACM, 2013: 379-391

[18] Mars J, Tang L, Hundt R. Heterogeneity in "Homogeneous" warehouse-scale computers [J]. IEEE Computer Architecture Letters, 2011, 10(2): 29-32

[19] Nathuji R, Isci C, Gorbatov E. Exploiting platform heterogeneity for power efficient data centers [C] //Proc of IEEE ICAC'07. Piscataway, NJ: IEEE, 2007: Article No. 5

[20] Delimitrou C, Kozyrakis C. Paragon: QoS-aware scheduling for heterogeneous datacenters [C] //Proc of ACM ASPLOS'13. New York: ACM, 2013: 77-88

[21] Xu Yunjing, Musgrave Z, Noble B, et al. Bobtail: Avoiding long tails the cloud [C] //Proc of USENIX NSDI'13. Berkeley, CA: USENIX Association, 2013: 329-341

[22] Ousterhout K, Wendell P, Zaharia M, et al. Sparrow: Distributed, low latency scheduling [C] //Proc of ACMSOSP'13. New York: ACM, 2013: 69-84

[23] Barroso L A, Clidaras J, Hölzle U. The datacenter as a computer: An introduction to the design of warehouse-scale machines [M]. 2nd ed. Williston, VT, USA: Morgan & Claypool Publishers, 2013

[24] Lu Xicheng, Wang Huaimin, Wang Ji, et al. Internet-based virtual computing environment: Beyond the data center as a computer [J]. Future Generation Computer Systems, 2013, 29(1): 309-322

[25] Barroso L A, Dean J, Hölzle U. Web search for a planet: The google cluster architecture [J]. IEEE Micro, 2003, 23(2): 22-28

[26] Nishtala R, Fugal H, Grimm S, et al. Scaling memcache at Facebook [C] //Proc of USENIX NSDI'13. Berkeley, CA: USENIX Association, 2013: 385-398

[27] Yang Ziye, Fang Haifeng, Wu Yingjun, et al. Understanding the effects of hypervisor I/O scheduling for virtual machine performance interference [C] //Proc of IEEE CloudCom'12. Piscataway, NJ: IEEE, 2012: 34-41

[28] Mars J, Tang Lingjia. Whare-map: Heterogeneity "Homogeneous" warehouse-scale computers [C] //Proc of ACM ISCA'13. New York: ACM, 2013: 619-630

[29] Chowdhury M, Kandula S, Stoica I. Leveraging endpoint flexibility data-intensive clusters [C] //Proc of ACM SIGCOMM'13. New York: ACM, 2013: 231-242

[30] Winstein K, Balakrishnan H. TCP ex machina: Computer-generated congestion control [C] //Proc of ACM SIGCOMM'13. New York: ACM, 2013: 123-134

[31] Alizadeh M, Edsall T. On the data path performance of Leaf-Spine datacenter fabrics [C] //Proc of IEEE HOTI'13. Piscataway, NJ: IEEE, 2013: 71-74

[32] Jeyakumar V, Alizadeh M, Kim C, et al. Tiny packet programs for low-latency network control and monitoring [C] //Proc of ACM HotNets-XII. New York: ACM, 2013: 8:1-8:7

[33] Flach T, Dukkipati N, Terzis A, et al. Reducing Web latency: The virtue of gentle aggression [C] //Proc of ACM SIGCOMM'13. New York: ACM, 2013: 159-170

[34] Zheng H, Luo E, Pias M, et al. Internet routing policies and round-trip time [G] //LNCS 3431: Proc of PAM'05. Berlin: Springer, 2005: 236-250

[35] Wang Guohui, Ng T S E. The impact of virtualization on network performance of Amazon EC2 data center [C] //Proc of IEEE INFOCOM'10. Piscataway, NJ: IEEE, 2010: 1163-1171

[36] Fu Yongquan, Wang Yijie, Biersack E. A general scalable and accurate decentralized level monitoring method for large-scale dynamic service provision hybrid clouds [J]. Future Generation Computer Systems, 2013, 29(5): 1235-1253

[37] Fu Yongquan, Wang Yijie. HyperSpring: Accurate and stable latency estimation the hyperbolic space [C] //Proc of IEEE ICPADS'09. Piscataway, NJ: IEEE, 2009: 864-869

[38] Ng T S E, Zhang Hui. Predicting internet network distance with coordinates-based approaches [C] //Proc of IEEE INFOCOM'02. Piscataway, NJ: IEEE, 2002: 170-179

[39] Tang Liying, Crovella M. Virtual landmarks for the Internet [C] //Proc of ACM IMC'03. New York: ACM, 2003: 143-152

[40] Lee S, Zhang Z L, Sahu S, et al. On suitability of euclidean embedding for host-based network coordinate systems [J]. IEEE/ACM Trans on Networking, 2010, 18(1): 27-40

[41] Abrahao B, Kleinberg R. On the Internet delay space dimensionality [C] //Proc of ACM IMC'08. New York: ACM, 2008: 157-168

[42] Fraigniaud P, Lebhar E, Viennot L. The inframetric model for the Internet [C] //Proc of IEEE INFOCOM'08. Piscataway, NJ: IEEE. 2008: 1085-1093

[43] Fu Yongquan, Wang Yijie, Pei Xiaoqiang. Towards latency-optimal distributed relay selection [C] //Proc of IEEE/ACM CCGrid'15. Piscataway, NJ: IEEE, 2015: 433-442

[44] Lumezanu C, Baden R, Spring N, et al. Triangle inequality and routing policy violations the Internet [C] //LNCS 5448: Proc of PAM'09. Berlin: Springer, 2009: 45-54

[45] Savage S, Anderson T, Aggarwal A, et al. Detour: Informed Internet routing and transport [J]. IEEE Micro, 1999, 19(1): 50-59

[46] Zhang Bo, Ng T S E, Nandi A, et al. Measurement-based analysis, modeling, and synthesis of the Internet delay space [J]. IEEE/ACM Trans on Networking, 2010, 18(1): 229-242

[47] Ramasubramanian V, Malkhi D, Kuhn F, et al. On the treeness of Internet latency and bandwidth [C] //Proc of ACM SIGMETRICS'09. New York: ACM, 2009: 61-72

[48] Zhu Yibo, Chen Yang, Zhang Zengbin, et al. Taming the triangle inequality violations with network coordinate system on real Internet [C] //Proc of the Re-Architecting the Internet Workshop. New York: ACM, 2010: 7:1-7:6

[49] Mukherjee A. On the dynamics and significance of low frequency components of Internet load [J]. Internetworking: Research and Experience, 1992, 5: 163-205

[50] Claffy K, Braun H, Polyzos G. Measurement considerations for assessing unidirectional latencies [J]. Internetworking: Research and Experience, 1993, 4(3): 121-132

[51] Zhang Yin, Duffield N G. On the constancy of Internet path properties [C] //Proc of the ACM SIGCOMM Workshop on Internet Measurement. New York: ACM, 2001: 197-211

[52] Costa M, Castro M, Rowstron A, et al. PIC: Practical Internet coordinates for distance estimation [C] //Proc of IEEE ICDCS'04. Piscataway, NJ: IEEE, 2004: 178-187

[53] Wong B, Slivkins A, Sirer E G. Meridian: A lightweight network location service without virtual coordinates [C] //Proc of ACM SIGCOMM'05. New York: ACM, 2005: 85-96

[54] Fu Yongquan, Wang Yijie. DKNNS: Scalable and accurate distributed K nearest neighbor search for latency-sensitive applications [J]. Science China Information Sciences, 2013, 56(3): 1-17

[55] Mi Haibo, Wang Huaimin, Zhou Yangfan, et al. Toward fine-grained, unsupervised, scalable performance diagnosis for production cloud computing systems [J]. IEEE Trans on Parallel Distributed System, 2013, 24(6): 1245-1255

[56] Lee M, Hajjat M Y, Kompella R R, et al. A flow measurement architecture to preserve application structure [J]. Computer Networks, 2015, 77: 181-195

[57] Peng Yanghua, Yang Ji, Wu Chuan, et al. deTector: A topology-aware monitoring system for data center networks [C] //Proc of USENIX ATC'17. Berkeley, CA: USENIX Association, 2017: 55-68

[58] Mace J, Roelke R, Fonseca R. Pivot tracing: Dynamic causal monitoring for distributed systems [C] //Proc of ACM SOSP'15. New York: ACM, 2015: 378-393

[59] Nygren E, Sitaraman R K, Sun J. The akamai network: A platform for high-performance Internet applications [J]. SIGOPS Operation System Review, 2010, 44(3): 2-19

[60] Measurement Laboratory. What is Measurement Lab? [EB/OL]. (2013-02-04) [2011-01-12]. http://www.measurementlab.net/

[61] Madhyastha H V, Isdal T, Piatek M, et al. iPlane: An information plane for distributed services [C] //Proc of USENIX OSDI'06. Berkeley, CA: USENIX Association, 2006: 367-380

[62] Madhyastha H V, Katz-Bassett E, Anderson T E, et al. iPlane nano: Path prediction for peer-to-peer applications [C] //Proc of USENIX NSDI'09. Berkeley, CA: USENIX Association, 2009: 137-152

[63] Lee D K, Jang K, Lee C, et al. Scalable and systematic Internet-wide path and delay estimation from existing measurements [J]. Computer Networks, 2011, 55(3): 838-855

[64] Gummadi K P, Saroiu S, Gribble S D. King: Estimating latency between arbitrary Internet end hosts [C] //Proc of ACM IMW'02. New York: ACM, 2002: 5-18

[65] Leonard D, Loguinov D. Turbo king: Framework for large-scale Internet delay measurements [C] //Proc of IEEE INFOCOM'08. Piscataway, NJ: IEEE, 2008: 31-35

[66] Jang K, Lee D K, Moon S B, et al. Internet sibilla: Utilizing DNS for delay estimation service [C] //Proc of ACM CoNEXT'08. New York: ACM, 2008: Article No. 54

[67] Levchenko K, Dhamdhere A, Huffaker B, et al. Packetlab: A universal measurement endpoint interface [C] //Proc of ACM IMC'17. New York: ACM, 2017

[68] Mao Y, Saul L K, Smith J M. IDES: An Internet distance estimation service for large networks [J]. IEEE Journal on Selected Areas Communications, 2006, 24(12): 2273-2284

[69] Lim H, Hou J C, Choi C-H. Constructing an Internet coordinate system based on delay measurement [J]. IEEE/ACM Trans on Networking, 2005, 13: 513-525

[70] Dabek F, Cox R, Kaashoek F, et al. Vivaldi: A decentralized network coordinate system [C] //Proc of ACM SIGCOMM'04. New York: ACM, 2004: 15-26

[71] Liao Y, Geurts P, Leduc G. Network distance prediction based on decentralized matrix factorization [C] //Proc of IFIP NETWORKING'10. Piscataway, NJ: IEEE, 2010: 15-26

[72] Agarwal S, Lorch J R. Matchmaking for online games and other latency-sensitive P2P systems [C] //Proc of ACM SIGCOMM'09. New York: ACM, 2009: 315-326

[73] Chen Yang, Wang Xiao, Shi Cong, et al. Phoenix: A weight-based network coordinate system using matrix factorization [J]. IEEE Trans on Network and Service Management, 2011, 8(4): 334-347

[74] Fu Yongquan, Xu Xiaoping. Self-stabilized distributed network distance prediction [J]. IEEE/ACM Trans on Networking, 2017, 25(1): 451-464

[75] Ousterhout J, Agrawal P, Erickson D, et al. The case for RAMClouds: Scalable high-performance storage entirely DRAM [J]. SIGOPS Operation System Review, 2010, 43(4): 92-105

[76] Li Ziyang, Bai Wei, Chen Kai, et al. Rate-aware flow scheduling for commodity data center networks [C] //Proc of IEEE INFOCOM'17. Piscataway, NJ: IEEE, 2017: 1-9

[77] Li Ziyang, Zhang Yiming, Li Dongsheng, et al. OPTAS: Decentralized flow monitoring and scheduling for tiny tasks [C] //Proc of IEEE INFOCOM'16. Piscataway, NJ: IEEE, 2016: 1-9

[78] Kalia A, Kaminsky M, Andersen D G. FaSST: Fast, scalable and simple distributed transactions with two-sided (RDMA) datagram RPCs [C] //Proc of USENIX OSDI'16. Berkeley, CA: USENIX Association, 2016: 185-201

[79] Fu Yongquan, Biersack E. False-positive probability and compression optimization for tree-structured bloom filters [J]. ACM Trans on TOMPECS, 2016, 1(4): 19:1-19:39

[80] Zhang Yiming, Guo Chuanxiong, Li Dongsheng, et al. CubicRing: Enabling one-hop failure detection and recovery for distributed in-memory storage systems [C] //Proc of USENIX NSDI 15. Berkeley, CA: USENIX Association, 2015: 529-542

[81] Zhang Yiming, Li Dongsheng, Guo Chuanxiong, et al. CubicRing: Exploiting network proximity for distributed in-memory key-value store [J]. IEEE/ACM Trans on Networking, 2017, 25(4): 2040-2053

[82] Zhang Yiming, Li Dongsheng, Tian Tian, et al. CubeX: Leveraging glocality of cube-based networks for RAM-based key-value store [C] //Proc of IEEE INFOCOM'17. Piscataway, NJ: IEEE, 2017: 1-9

[83] Li Sheng, Lim H, Lee V W, et al. Full-stack architecting to achieve a billion-requests-per-second throughput on a single key-value store server platform [J]. ACM Trans on Computer System, 2016, 34(2): 5:1-5:30

[84] Shen Zhiming, Subbiah S, Gu Xiaohui, et al. CloudScale: Elastic resource scaling for multi-tenant cloud systems [C] //Proc of ACM SOCC'11. New York: ACM, 2011: Article No. 5

[85] Zhang Zhaoning, Li Dongsheng, Wu Kui, et al. VMThunder: Fast provisioning of large-scale virtual machine clusters [J]. IEEE Trans on Parallel Distributed System, 2014, 25(12): 3328-3338

[86] Zhang Zhaoning, Li Dongsheng, Wu Kui. Large-scale virtual machines provisioning clouds: Challenges and approaches [J]. Frontiers of Computer Science, 2016, 10(1): 2-18

[87] Katta N P, Ghag A, Hira M, et al. Clove: Congestion-aware load balancing at the virtual edge [C] //Proc of ACM CoNEXT 2017. New York: ACM, 2017: 323-335

[88] Vulimiri A, Godfrey P B, Mittal R, et al. Low latency via redundancy [C] //Proc of ACM CoNEXT'13. New York: ACM, 2013: 283-294

[89] Li Xiaoyong, Wang Yijie, Fu Yongquan, et al. BLOR: An efficient bandwidth and latency sensitive overlay routing approach for flash data dissemination [J]. Concurrency and Computation: Practice and Experience, 2015, 27(14): 3614-3632

[90] Ma Xingkong, Wang Yijie, Pei Xiaoqiang, et al. Scalable and elastic total order content-based publish/subscribe systems [J]. Computer Networks, 2015, 83: 297-314

[91] Ma Xingkong, Wang Yijie, Pei Xiaoqiang. A scalable and reliable matching service for content-based publish/subscribe systems [J]. IEEE Trans on Cloud Computing, 2015, 3(1): 1-13

[92] Wang Yijie, Ma Xingkong. A general scalable and elastic content-based publish/subscribe service [J]. IEEE Trans on Parallel Distributed System, 2015, 26(8): 2100-2113

[93] Ma Xingkong, Wang Yijie, Pei Xiaoqiang, et al. A cloud-assisted publish/subscribe service for time-critical dissemination of bulk content [OL]. [2017-10-01]. http://onlinelibrary.wiley.com/doi/10.1002/cpe.4047/full

[94] Pei Xiaoqiang, Wang Yijie, Ma Xingkong, et al. Cooperative repair based on tree structure for multiple failures distributed storage systems with regenerating codes [C] //Proc of ACM CF'15. New York: ACM, 2015: 14:1-14:8

[95] Pei Xiaoqiang, Wang Yijie, Ma Xingkong, et al. Efficient in-place update with grouped and pipelined data transmission erasure-coded storage systems [J]. Future Generation Computer Systems, 2017, 69: 24-40

[96] Li Huiba, Zhang Yiming, Zhang Zhiming, et al. PARIX: Speculative partial writes erasure-coded systems [C] //Proc of USENIX ATC'17. Berkeley, CA: USENIX Association, 2017: 581-587

Fu Yongquan, born 1983. PhD, lecturer. Member of CCF. His main research interests include network measurement, social networks, and distributed systems.

Li Dongsheng, born 1978. PhD, professor, PhD supervisor. Member of CCF. His main research interests include distributed computing, cloud computing, computer network, and large-scale data management.

2018 年《计算机研究与发展》专题(正刊)征文通知

——新型存储系统前沿技术

近年来,随着国家和社会信息化发展的不断加速,对信息存储的提出了越来越高的要求.一方面,大数据时代,数据存储的规模和处理需求越来越高,亟需新型存储系统和技术以提供更高的性能和更好的可扩展性.另一方面,由于各种人工智能系统及相关技术的出现,现有的存储技术和系统难以满足上层系统和技术的需求.因此,存储系统结构技术研究面临诸多新的机遇和挑战.

《计算机研究与发展》开辟了"计算机体系结构前沿技术"系列,并计划于 2018 年出版"新型存储系统结构前沿技术"专题.本专题将集中讨论面向非易失存储器件的存储体系结构技术、海量信息存储系统研究与技术、新型工艺支撑的计算机存储体系结构技术、存储前沿技术等,相关研究可以涉及不同应用领域、环境、场景的存储系统,也可以涉及存储系统的不同层次和部件,还可以包括基于新型工艺和器件的存储结构设计等.

本次专题受到中国计算机学会信息存储专委会的支持.专题录用的文章作者将被邀请参加 2018 年 9 月全国信息存储学术会议,邀请作者到会演讲并参与讨论.欢迎相关领域的专家学者和科研人员踊跃投稿.现将专题论文征集的有关事项通知如下.

征文范围(但不限于)

1) 面向非易失存储器件的存储体系结构技术,如:面向非易失存储器件的持久性内存(memory)和存储(storage)结构与技术;面向持久性内存的编程模型、空间管理、文件系统和数据库等;固态 SSD 存储与技术;新型存储系统及其应用技术.

2) 海量信息存储系统技术,如:高性能计算机存储系统、分布式文件系统;网络存储、数据存储网格及相关技术;海量信息存储技术;网络存储系统可用性、安全性、可靠性及容灾研究;重复数据删冗、归档、分级等存储技.

3) 新型工艺支撑的计算机存储体系结构技术,如:非易失存储器结构技术;PIM (Processing in Memory)技术;3-维堆叠结构技术.

4) 存储前沿技术,如:软件定义存储与存储超融合;类脑存储;量子存储.

稿件内容也可以为上述所列内容的交叉.本刊也鼓励作者讨论与上述领域相关但此处未予提及的重要问题、创造性的研究思路和探索视角.

投稿要求

1) 论文应属于作者的科研成果,未在国内外公开发行的刊物或会议上发表,不存在一稿多投问题.作者在投稿时,需向编辑部提交版权转让协议.

2) 论文一律用 Word 格式排版,格式体例参考近期出版的《计算机研究与发展》的要求(http://crad.ict.ac.cn/下载区域).

3) 论文通过期刊网站(http://crad.ict.ac.cn)投稿,投稿时需提供作者的联系方式.作者投稿时请务必在留言中注明"存储系统前沿 2018 专题"(否则按自由来稿处理).

重要日期

征文截止日期:2018 年 4 月 5 日　　录用通知日期:2018 年 5 月 31 日

修改稿提交日期:2018 年 6 月 15 日　　出版日期:2018 年 9 月

特邀编委

刘志勇　研究员　中国科学院计算技术研究所　zyliu@ict.ac.cn

舒继武　教　授　清华大学　shujw@tsinghua.edu.cn

联系方式

编 辑 部:crad@ict.ac.cn,010-62620696,010-62600350

通信地址:北京 2704 信箱《计算机研究与发展》编辑部　　邮政编码:100190