

一种基于预测与动态调整负载因子的SDN流表优化算法

史少平 庄雷 杨思锦

(郑州大学信息工程学院 郑州 450001)

摘要 通过对SDN流表更新的研究,发现了网络流量高峰期流表更新不及时的问题。提出了一种基于预测与动态调整负载因子的SDN流表优化算法。算法首先收集每个单位时间内的新增流条目,然后用二次移动平均算法对收集的历史数据进行分析,并估计下一个单位时间内新增的流条目,最终根据负载因子动态调整交换机流表中流条目的停滞超时时间。实验结果表明,该算法提高了流表匹配率和数据成功转发率,增加了活动流表项的数量。

关键词 软件定义网络,二次移动平均,负载因子,停滞超时时间

中图分类号 TP393 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2017.01.024

SDN Optimization Algorithm Based on Prediction and Dynamic Load Factor

SHI Shao-ping ZHUANG Lei YANG Si-jin

(School of Information Engineering, Zhengzhou University, Zhengzhou 450001, China)

Abstract This paper studied the adjustment of flow table in the software defined networking (SDN). A SDN optimization of the flow table algorithm (SOOTFTA) based on prediction and dynamic load factor was proposed to address the problem that switch do not have enough space in the flow table for newly arrived flows. Firstly, the SOOTFTA collects the various newly arrived flows during per unit of time. Based on the information, the SOOTFTA estimates the number of newly arrived flows in the next time unit by using the second moving average method (SMA). Finally, the SOOTFTA dynamically adjust the idle timeout of the flows in the table based on the dynamic load factor. Experimental result show that the flow table matching rate and data forwarding rate are improved by using the proposed SOOTFTA, increasing the number of activities in the flow table.

Keywords Software defined networking, Second moving average(SMA), Load factor, Idle timeout

1 引言

在传统网络体系结构中,核心设备与操作系统结合紧密,不对外兼容,且一般采用分布式管理的方式,各设备独立运行;另外,由于没有统一的标准,其可编程性较差。随着应用类型的不断丰富和网络规模的急剧膨胀,传统网络体系结构的不足更加明显。为解决现有网络体系结构中的诸多难题,世界各国对此都展开了大规模的研究。软件定义网络(Software Defined Networking, SDN)是其中的一种解决方案。它实现了网络控制平面和数据平面的分离^[1],广泛地应用在学术研究和工程领域中。与传统的网络体系结构比,软件定义网络在硬件通用性、管理控制方式、可编程性等方面都具有明显的优势^[2]。

软件定义网络的核心设备控制器通过支持 OpenFlow 协议的控制信道,配置其域内的若干个交换机。交换机将配置信息保存到各自的流表中。数据转发时,交换机根据本地流表中的流条目依次匹配数据包,选择匹配成功的流条目中优

先级最高的流条目作为匹配结果,最后根据流条目中的规则对数据包进行转发处理。而未找到流条目或按照流条目规则处理失败的数据包被交换机转发给控制器,控制器则对转发来的数据包生成新的转发规则,并把新规则传递给交换机,同时通知交换机更新流表。在交换机传递匹配失败或处理失败的数据包给控制器的过程中,交换机将占用控制信道,若控制器忙或控制信道拥塞,交换机转发给控制器的数据包将被丢弃;在控制器传递新的转发规则给交换机的过程中,控制器将占用控制信道,若交换机的流表已满,控制器传递给交换机的数据包转发规则将被丢弃。上述情况降低了交换机的可用性,严重影响了 SDN 的系统性能。在具有大规模数据流的数据中心网络中,此问题表现得尤为突出。

近年来,对提高 SDN 系统性能的相关研究工作可分为两个方向:优化 SDN 控制器性能和增强 OpenFlow 交换机性能。现有的优化 SDN 控制器性能的研究已经处于成熟阶段,其思路有 3 种:提升控制软件性能^[3-6],优化控制器管控能力;控制面纵向扩展^[7],使 OpenFlow 交换机具有部分控制处理

到稿日期:2015-10-17 返修日期:2016-02-09 本文受国家重点基础研究发展计划(“973”计划)基金资助项目(2012CB315901),国家自然科学基金资助项目(61379079),河南省国际合作项目(152102410021)资助。

史少平(1990—),女,硕士生,主要研究方向为软件定义网络、下一代互联网,E-mail:15538701005@163.com;庄雷(1963—),女,博士,教授,博士生导师,CCF 会员,主要研究方向为对等网络体系结构、分布式计算;杨思锦(1990—),女,硕士生,主要研究方向为软件定义网络、下一代互联网。

能力;控制面横向扩展^[8-14],建立多控制器分布协同工作模式。实际应用中需要针对特定的应用场景和需求,选取合适的控制器方案,也可将多种控制方案组合使用。增强 OpenFlow 交换机数据转发能力的相关方案大致可分为两类:基于输入流的特征分析的方案和基于流表资源复用的方案。文献[15]通过分析每次的输入流的到达数量,用 AR 算法^[16]预测下一个取样周期内到达的流数量,并用威布尔分布^[17]估算下一取样周期中仍然存在于流表中的流表项数量,提出了一种动态调整流表项超时时间的方案。但对于波动比较大的数据,AR 算法的预测结果的准确度较低,且 AR 算法要求历史数据在时间序列上存在依赖关系。另外,AR 算法实现复杂,加重了控制器负荷,降低了动态调整方案的性能。文献[18]通过分析流表的使用情况和流到达特征,估计流的吞吐量,预测活动流表项的数量及失效时间。该方案自适应调整流表项默认停滞超时时间,同时优化新入流的停滞超时时间,显著降低了输入流损失和控制信道吞吐量,保证了数据转发的可用性。但方案中的控制器需要实时地对输入流分布进行分析预测,增大了控制器负荷。文献[19]提出了一种 OpenFlow 交换机流表自动控制机制,用于解决大数据流情况下交换机的可用性问题。但是该方案对流表项停滞时间的优化处理仍然是静态优化,没有考虑到不同流的特征。文献[20]提出了一种智能超时控制机制,以更充分利用有限的流表资源;该方案在 SDN 控制器中增加了两个模块:流停滞超时登记模块和反馈控制模块。流停滞超时登记模块用以记录流的停滞超时时间特征,以便动态调整其停滞超时时间;反馈控制模块用于接收流表负载反馈,以便动态调整流表项停滞超时时间的最大值,减少流表溢出。然而该方案在控制器中增加的模块加重了 SDN 控制器的负载,降低了系统性能。文献[21]基于历史报文特征和处理方式,提出了一种对未匹配报文进行处理的解决方案。在该方案中,对于未匹配报文,交换机根据历史信息独立处理,不再与控制器进行通信。该方案虽然在一定程度上节省了控制器信道带宽,加快了报文的转发速度,但需在交换机上附加复杂控制功能,有悖于 SDN 控制平面和数据平面相分离的本质。

针对上述问题,本文提出了一种基于预测与动态调整负载因子的 SDN 流表优化算法。算法通过收集每个单位时间内新增的流表条目数量,预测下一时刻新增的流条目数量,并用预测值计算调整因子,最后根据调整因子动态调整交换机流表中流条目的停滞超时时间。算法在保证流条目活动流表项数量的同时,减少了与控制器的交互,有效降低了控制负载。

2 基于预测与动态调整负载因子的 SDN 流表优化算法

在网络流量高峰期,由于交换机流表资源有限,交换机丢弃了大量请求报文,严重影响了交换机性能。针对该问题,本文提出了一种流表更新算法以尽可能充分地利用流表资源。算法首先收集每个取样周期内新增的流条目数目;并根据收集的历史数据,用二次移动平均算法估计下一个取样周期内新增的流条目;最后通过计算负载因子动态调整交换机的流表中每个流条目的停滞超时时间,同时动态清除流表中时间超时的流条目。该算法使交换机的流表不仅有足够的空间容

纳新增流条目,而且保存了足够的活动流条目,保证了数据转发率。

算法框架如图 1 所示。

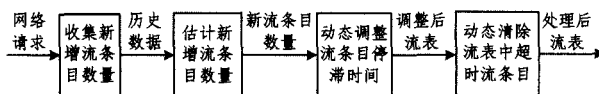


图 1 算法框架

2.1 基于 SMA 的数据分析

历史数据收集了每一个时间周期内新增加的流条目数量。本文通过使用二次移动平均算法对历史数据进行分析,估计下一时间周期内新增的流条目数量。移动平均算法是一种分析预测时间序列数据的算法,它通过计算一组观察值的均值,估计下一周期内可能出现的值。但在计算均值时,观察数据的数量必须明确规定,且当获取新的观察值时,需更新观察数据。移动平均法有两种极端情况:若观察数据的数量为 1,则最新观察值将为下一周期的估计值;若观察数据的数量未明确规定,则记录过去所有观察值,并将其均值作为估计值。

设时间序列为 $x_1, x_2, \dots$, 一次移动平均计算公式为:

$$F_{t+1} = (x_t + x_{t-1} + x_{t-2} + \dots + x_{t-n+1}) / n \quad (1)$$

其中, x_t 是 t 时刻的观察值, F_{t+1} 是 $t+1$ 时刻的预测值, n 为选取的连续观察值的数量。式(1)表示历史数据值对预测值的贡献, n 越大则平滑效果越好。为避免移动平均算法产生系统误差,本文用二次移动平均算法(SMA)估计新增流条目。SMA 算法在对观察值进行一次移动平均的基础上,再进行一次移动平均。SMA 计算公式为:

$$S_t^1 = (x_t + x_{t-1} + x_{t-2} + \dots + x_{t-n+1}) / n \quad (2)$$

$$S_t^2 = (S_t^1 + S_{t-1}^1 + S_{t-2}^1 + \dots + S_{t-n+1}^1) / n \quad (3)$$

$$a_t = 2S_t^1 - S_t^2 \quad (4)$$

$$b_t = 2(S_t^1 - S_t^2) / (n-1) \quad (5)$$

$$F_{t+m} = a_t + b_t m \quad (6)$$

其中, S_t^1 是 t 时刻一次移动平均值; S_t^2 是二次移动平均值; a_t 是截距,用于对预测的初始点进行基本修正,使预测值与实际值之间不存在滞后现象; b_t 是斜率; m 为预测超前期数。

SMA 算法首先需计算第一次移动平均值,并在此基础上进行第二次移动平均计算,接下来根据公式计算截距和斜率,并用其计算最终预测值。

整个 SMA 算法的过程如下。

输入:历史数据 D , 时间 t , 连续观察值的数量 n , 预测超前期数 m

输出:预测值 N_p , 对历史数据 D 进行排序

```

while i <= n
    计算第一次移动平均值  $s_1$ ;
     $s_2 += s_1 / n$ ;
End while
计算截距  $a$ ;
计算斜率  $b$ ;
计算预测值  $N_p$ ;
  
```

SMA 算法的时间复杂度为 $O(n^2)$, 其中 n 为选取的连续观察值的数量。由算法可知,与 AR 算法^[15]相比,首先 SMA 算法实现简单、运行效率高、对历史数据没有依赖性,在交换机刚开始运行时就可以依据较少的历史数据进行预测工作;

而AR算法对历史数据具有依赖性,在进行首次预测之前需要收集至少上万条历史数据,交换机刚开始运行时由于收集的历史数据不足,不能输出预测结果,当收集的历史数据至少上万条以上时,可以输出预测结果,但因为其实现复杂,预测过程需要的时间较长,增大了控制器负载,降低了交换机的运行效率;其次,当数据波动比较大时,SMA算法比AR算法预测更准确,这是因为移动平均对原序列有修匀和平滑作用,削弱了原序列的上下波动,在观察数据值波动较大的情况下,移动平均的修匀作用能有效地减弱估计值中的随机波动;而对于具有明显趋势变动的数据,AR算法的预测值具有“滞后”现象。

2.2 流条目停滞超时时间分析

在数据转发过程中,控制器与交换机信道资源的大小、流表物理空间大小、控制器处理能力是制约数据转发能力的瓶颈,单纯增大流表不仅会增大硬件开销,而且会降低转发速度。因此在流表容量一定的情况下,流条目的处理是关键。如图2(a)所示,流表由存活的流条目和剩余空间构成。每个流条目包含转发规则、动作、流的停滞超时和流的硬超时等信息。当某一流条目的硬超时设定时间到达或没有在停滞超时设定的范围内出现时,该条目将从表中失效。为使系统发挥最佳性能,原则上空闲流表项必须保持一定且合适的数量。

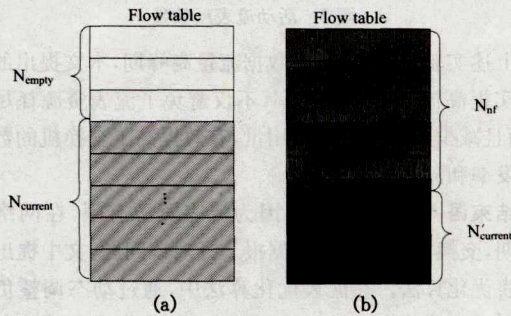


图2 流表结构模型

本文通过更改流条目的停滞超时时间调整活动项的失效速度,达到数据转发率和流表匹配率动态平衡的目的。假定取样周期 T 时的流表结构如图2(a)所示,由此图可知:

$$N_{max} = N_{current} + N_{empty} \quad (7)$$

其中, N_{max} 是流表的总的物理空间, $N_{current}$ 是流表中存在流条目的数量, N_{empty} 是流表空闲空间。通过对最近一段时间的历史数据进行分析,估计下一时间周期可能产生的新增流条目数量,根据估计值调整流表中原有条目的停滞时间。若估计值较大,表明流表在下一时间周期可能需应对产生大量流条目的情况,流表应保留较大的空闲空间,因此应加快原流条目的失效速度。若估计值较小,表明在下一时间周期产生的流条目数量较少,因此应减慢原流条目的失效速度,提高流表匹配率。假定 $N'_{current}$ 为取样周期 $T+1$ 时的流表中的流条目, N_{nf} 为流表中新增流条目,如图2(b)所示。

$$N_{max} = N'_{current} + N_{nf} \quad (8)$$

式(8)表示 $T+1$ 时刻的流表状态。由式(7)和式(8)可知,若 $N_{nf} > N_{empty}$ 则 N_{old} 中部分流条目应被清除,为新增流条目预留空间;若 $N_{nf} < N_{empty}$, 则 N_{old} 空余空间足以容纳新增流条目。根据上述分析,本文提出了调整因子 α 的概念,用以调整流表项的失效速度。

$$\alpha = N_e / (1 + N_p) \quad (9)$$

其中, N_e 是流表剩余空间的大小, N_p 是通过分析历史数据并利用SMA算法预测获得的估计值。其中分母加1是为了防止新流规则估计值为0。根据调整因子,本文对流条目的停滞时间提出了如下反馈机制:

$$\begin{cases} T_{idle} = \alpha * t_{def}, & \text{if } \alpha \leq 1 \\ T_{idle} = t_{def}, & \text{else} \end{cases} \quad (10)$$

其中, T_{idle} 是调整后流表项的停滞超时时间, t_{def} 是默认停滞超时时间。由反馈机制可知,当流表剩余空间的大小和新流表项估计值之间的比值不大于1,即流表剩余空间不足以容纳新流规则时,反馈机制规定流条目默认停滞超时时间乘以调整因子,以缩小流表项的停滞超时时间;当流表剩余空间的大小和新流表项估计值之间的比值大于1,即流表剩余空间足够容纳新流表项时,反馈机制规定流表项停滞超时时间仍然是默认值。该反馈机制不仅增强了流表应对网络流高峰期的能力,而且使流条目的写入速度和原流条目的失效速度达到了一个动态平衡的过程,减少了数据包被丢弃的情况,提高了数据的转发率和流表的匹配率。

2.3 流表优化算法

本算法通过调用SMA算法对收集的历史数据进行分析,估计下一个时间单位内可能新增的流条目数量,并根据估计值获得调整因子;最后用调整因子动态调整流表中每一个流条目的停滞时间,使新流条目的写入速度和原流条目的失效速度达到一个动态平衡。算法过程如下。

输入: 历史数据集 D , 流表空间大小 s_{all} , 当前流表已用空间大小 s_{use} , 流表集合 T , 默认停滞时间 t

输出: 调整后的流表 T'

步骤1 历史数据分析算法

调用SMA算法对历史数据集 D 进行分析, 得到估计值;

输出估计值 n_p

步骤2 计算调整后的停滞时间

$$\alpha = (s_{all} - s_{use}) / (1 + n_p)$$

If $\alpha > 1$ do

$$t_{adjust} = t$$

Else

$$t_{adjust} = t * \alpha$$

End If

输出调整后的停滞时间 t_{adjust}

步骤3 调整流表条目停滞时间

For every 流条目 Item $\in T$ do

若 Item 的停滞时间已到时, 则清除流条目;

否则将 Item 的停滞时间更改为 t_{adjust} ;

End For

输出调整后的流表 T'

基于二次移动平均的动态调整负载因子调整算法的时间复杂度为 $O(n^2)$, 其中 n 为选取的连续观察值的数量。

3 实验结果与分析

为验证算法的性能, 首先建立了一个基本模型来仿真数据转发过程。考虑到输入流符合泊松分布, 以符合泊松分布的数据流作为输入流, 用于评估在网络流高峰期流表的可用性。另外, 实验假定没有匹配成功的数据包将被无延迟地转发给控制器。

3.1 实验环境与参数

实验环境:CPU 为 Intel(R)Core(TM)i3-2130 3.4GHz,内存为 4GB,操作系统为 Microsoft Windows 7;算法编写环境为 MATLAB2012;数据集来源于文献[21],数据集数量为 8000 条且有波动趋势;算法所需参数的设定如表 1 所列。

表 1 实验参数

流表物理空间	流类别	平均速度	控制器性能
400	10000	350/s	20 个/s,可缓存 10 组数据

3.2 性能指标

本文通过 3 个指标即数据成功转发率、流表匹配率以及活动流表项数量,将所提方案与固定停滞超时方案进行了比较。

数据成功转发率的计算公式如式(11)所示。

$$DSFR=D_{fs}/D_{tr} \tag{11}$$

其中,DSFR 是数据成功转发率, D_{fs} 是每一个时间周期内转发成功的流的数量, D_{tr} 是每一个时间周期中数据流请求的总数量。

流表匹配率的计算公式如式(12)所示。

$$FTMR=D_{ms}/D_{tr} \tag{12}$$

其中,FTMR 是流表匹配率, D_{ms} 是每一个时间周期内在流表中能直接匹配成功的请求数据流的数量。

活动流表项数量即每一个时间周期内在流表中能和任意一条或多条请求数据流匹配成功的流条目的数量。

3.2.1 数据成功转发率

图 3 表示在一段时间内数据成功转发率的变化曲线。由图可知,本文提出的方案的数据成功转发率明显高于固定停滞超时方案的数据成功转发率。主要原因是在固定停滞超时的方案中,在数据高峰期,流表没有合适的空闲空间容纳新流,导致数据包被丢弃,严重影响了数据的成功转发率。在本文的方案中,通过 SMA 算法预测出了新流表项的数量,并以此预测值计算调整因子,这种随周期变化的调整因子的大小使得流条目的停滞时间通过反馈机制得到了动态调整,在数据高峰期加快了原流条目的失效速度,调整了流表空闲空间的大小以适应大的数据流量,减少了数据包被丢弃的情况,提高了数据转发率。

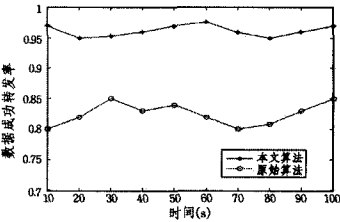


图 3 数据成功转发率

3.2.2 流表匹配率

图 4 表示在一段时间内流表匹配率的变化曲线。从图中可知,本文提出的方案并不会降低流表的匹配率,因为本文的算法是对停滞超时时间的动态调整,根据调整因子加快流条目的失效速度;但算法对匹配率较高的流条目的影响较小,因为匹配成功的流条目会重置停滞时间,随时间推移反而会不断优化流表条目,使流表中保存了匹配率较高的流条目,提高了整个流表的匹配率。

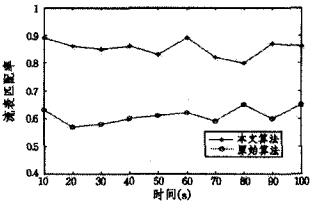


图 4 流表匹配率

3.2.3 活动流表项数量

图 5 示出了一段时间内活动流条目的变化情况。由图 5 可知,在固定停滞超时方案中,即使在大数据流量下仍有约一半的空余流表项,严重浪费了流表空间,降低了交换机性能。在本文提出的方案中,活动流表项的比例始终能占据整个流表空间的 $\frac{3}{4}$ 以上,使得流表的资源得到了更充分的利用,交换机性能得到了更充分的发挥,同时减少了信道拥堵情况的发生。

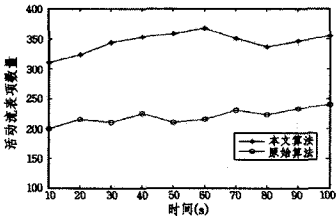


图 5 活动流表项数量

上述实验结果表明,在数据流量高峰期,本文提出的方案能切实提高流表资源利用率,不仅避免了流表资源耗尽的情况,而且减少了控制信道占用量,有效提高了交换机的数据成功转发率和流表匹配率。

结束语 SDN 网络中,因为流表资源有限,在网络流量高峰期,交换机经常丢弃数据报。针对该问题,文中提出了一种流表优化算法。在流表优化算法中,通过动态调整负载因子来调整流表中每一个流条目的停滞超时时间,使流条目的写入速度和原流条目的失效速度达到一个动态平衡,减少了数据包被丢弃的情况。该算法不仅提高了交换机的性能,而且减少了控制信道资源的占用量,提高了数据的成功转发率。但算法在调整流条目的停滞超时时间时未考虑每一个流条目自身的特点,对使用频率多的流条目不能单独处理,因此对流条目的自身特点还需进一步研究。

参 考 文 献

[1] MCKEOWN N. Software-defined networking [OL]. http://klamath.stanford.edu/~nickm/talks/infocom_brazil_2009_v1-1.pdf.
[2] MCKEOWN N, Anderson T, Balakrishnan H, et al. OpenFlow: enabling innovation in campus networks[J]. ACM SIGCOMM Computer Communication Review, 2008, 38(2): 69-74.
[3] CAI Z, COX A L, EUGENE T S. Maestro: a system for scalable OpenFlow control; Technical Report TR10-08[R]. Rice University, 2010.
[4] GUDE N, KOPONEN T, PETTIT J, et al. Nox: Towards an operating system for networks[J]. ACM SIGCOMM Computer Communication Review, 2008, 38(3): 105-110.
[5] ERICKSON D. The beacon openflow controller [C]// Proceedings of the 2nd ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking (HotSDN 13). New York, NY,

- USA; ACM, 2013; 13-18.
- [6] GREENBERG A, HJALMIYSSON G, MALTZ D A, et al. A clean slate 4D approach to network control and management [J]. ACM SIGCOMM Computer Communication Review, 2005, 35(5): 41-54.
 - [7] YU M, REXFORD, FREEDMAN M J, et al. Scalable Flow-based Networking with DIFANE[J]. ACM SIGCOMM Computer Communication Review, 2010, 40(4): 351-362.
 - [8] TOOTOONCHINA A, GANJALI Y. HyperFlow: A distributed control plane for OpenFlow[C]//Proceedings of the 2010 Internet Network Management Conference on Enterprise Networking. USENIX Association, 2010; 3.
 - [9] LIN Ping-ping, BI Jun, HU Hong-yu, et al. A Mechanism for Scalable Intra-domain Control Plane in SDN[J]. Journal of Chinese Computer Systems, 2013, 34(9): 1790-1794. (in Chinese)
林萍萍, 毕军, 胡虹雨, 等. 一种面向 SDN 域内控制平面可扩展性的机制[J]. 小型微型计算机系统, 2013, 34(9): 1790-1794.
 - [10] LIN P, BI J, HU H. Asic: an architecture for scalable intra-domain control in openflow[C]//Proceedings of the 7th International Conference on Future Internet Technologies. ACM, 2012: 21-26.
 - [11] KOPONEN T, CASADO M, GUDE N, et al. Onix: a distributed control platform for large-scale production networks[C]//OSDI. 2010; 101-106.
 - [12] VOLKAN Y, ALI O. Controlling a Software-Defined Network via Distributed Controllers[C]//Proceedings of the 2012 NEM Summit, 2014. Istanbul; arXiv preprint arXiv, 2014; 19-27.
 - [13] HASSAS Y S, GANJALI Y. Kandoo: a framework for efficient and scalable offloading of control applications[C]//Proceedings of the First Workshop on Hot Topics in Software Defined Networks. ACM, 2012; 19-24.
 - [14] TAM A S W, XI K, CHAO H J. Use of devolved controllers in data center networks[M]//Computer Communications Workshops(INFOCOM WKSHPS). IEEE, 2011; 596-601.
 - [15] KIM T, LEE K, LEE J, et al. A Dynamic timeout control algorithm in software defined networks[J]. International Journal of Future Computer and Communication, 2014, 3(5): 331-336.
 - [16] AKAIKE H. Fitting auto regressive models for prediction[J]. Annals of the institute of Statistical Mathematics, 1969, 21(1): 243-247.
 - [17] TSU T C, MUGELE R A, MCCLINTOCK F A. A statistical distribution function of wide applicability[J]. Journal of Applied Mechanics-Transactions of the Asme, 1952, 19(2): 233-234.
 - [18] ZHOU B, GAO W, WU C, et al. AdaFlow: Adaptive control to improve availability of OpenFlow forwarding for burst quantity of flows[M]//Testbeds and Research Infrastructure: Development of Networks and Communities. Springer International Publishing, 2014; 406-415.
 - [19] XIE L, ZHAO Z, ZHOU Y, et al. An adaptive scheme for data forwarding in software defined network[C]//2014 Sixth International Conference on Wireless Communications and Signal Processing (WCSP). IEEE, 2014; 1-5.
 - [20] ZHU H, FAN H, LUO X, et al. Intelligent timeout master: Dynamic timeout for SDN-based data centers[C]//IFIP/IEEE International Symposium on Integrated Network Management (IM), 2015. IEEE, 2015; 734-737.
 - [21] BENSON T, AKELLA A, MALTZ D A. Network traffic characteristics of data centers in the wild[C]//Proceedings of the 10th ACM SIGCOMM Conference on Internet Measurement. ACM, 2010; 267-280.
-
- (上接第 99 页)
- [6] CHEN C. Character-sense Association and Compounding Template Similarity: Automatic Semantic Classification of Chinese Compounds[C]//Proceedings of the 3rd SIGHAN Workshop on Chinese Language Processing. Barcelona. 2004; 33-40.
 - [7] LU Xiao-fei. Hybrid Model for Chinese Unknown Word Resolution[D]. The Ohio State University, 2006.
 - [8] LU Xiao-fei. Hybrid Model for Semantic Classification of Chinese Unknown Words[C]//Proceedings of North American Chapter of the Association for Computational Linguistics-Human Language Technologies 07, 2007. New York, 2007; 188-195.
 - [9] ZHANG Rui-xia, XIAO Han. The construction of Lattice based on HowNet [J]. Journal of North China Institute of Water Conservancy and Hydro Electric Power, 2008, 29(3): 53-56. (in Chinese)
张瑞霞, 肖汉. 基于《知网》的词图构造[J]. 华北水利水电学院学报, 2008, 29(3): 53-56.
 - [10] LU Xiao-fei. Hybrid Model for Chinese Unknown Word Resolution[D]. The Ohio State University, 2006.
 - [11] LU Xiao-fei. Hybrid Models for Semantic Classification of Chinese Unknown Words[C]//Proceedings of Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics, 2007. USA, 2007; 188-195.
 - [12] BORDES A, GABRILOVICH E. Constructing and Mining Web Scale Knowledge Graphs: WWW 2015 Tutorial[C]//Proceedings of International Conference on World Wide Web, 2015. Italy, 2015; 1523.
 - [13] MASS Y, SAGIV Y. Knowledge Management for Keyword Search over Data Graphs[C]//Proceedings of the 23rd ACM International Conference on Information and Knowledge Management, 2014. China, 2014; 2051-2053.
 - [14] WANG Z, ZHANG J, FENG J L, et al. Knowledge Graph and Text Jointly Embedding[C]//Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, 2014. Qatar, 2014; 1591-1601.
 - [15] ROKACH L, MAIMON O. Data mining and knowledge discovery handbook(2nd ed)[M]. US; Springer, 2005; 321-352.
 - [16] ALFRED R, FUN T S, TAHIR A, et al. Concepts Labeling of Document Clusters Using a Hierarchical Agglomerative Clustering (HAC) Technique[C]//The 8th International Conference on Knowledge Management in Organizations. Springer Netherlands, 2013; 263-272.
 - [17] TONG H, FALOUTSOS C, PAN J Y. Fast Random Walk with Restart and Its Applications[C]//Proceedings of IEEE International Conference on Data Mining, 2006. China, IEEE Computer Society, 2006; 613-622.
 - [18] XIA J, CARAGEA D, HSU W H. Bi-relational Network Analysis Using a Fast Random Walk with Restart[C]//Proceedings of IEEE International Conference on Data Mining, 2009. USA, IEEE Computer Society, 2009; 1052-1057.