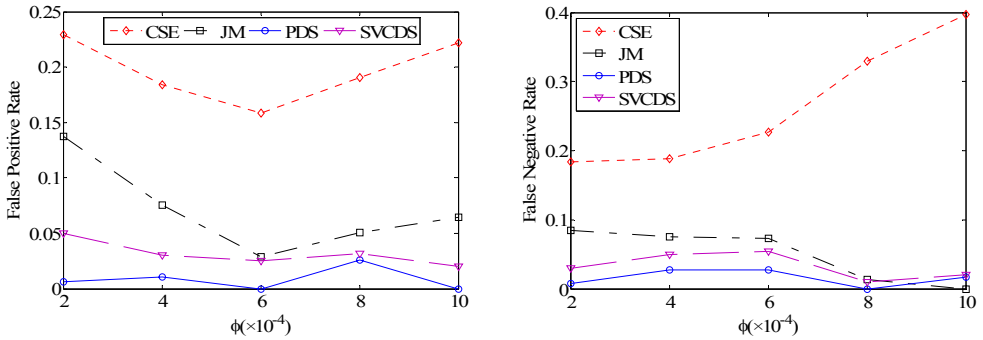
从而可以降低漏报;

- 另一方面,通过另一个 Sketch 数据结构验证因超列的错误组合产生的误报,从而能够有效地降低误报.

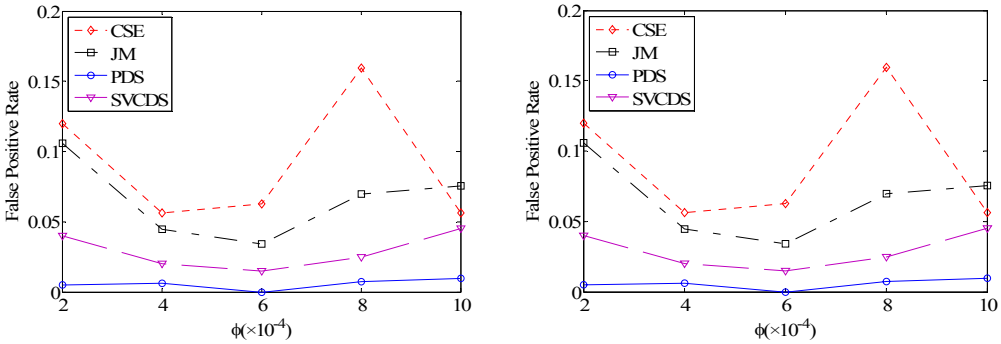
因此,实验结果表明,本文方法具有较高的精度、较低的漏报率和误报率.



(a) 误报率 (b) 漏报率

Fig.11 Comparison of accuracy of CSE, JM, SVCDS, PDS(SPort||SIP,DIP)

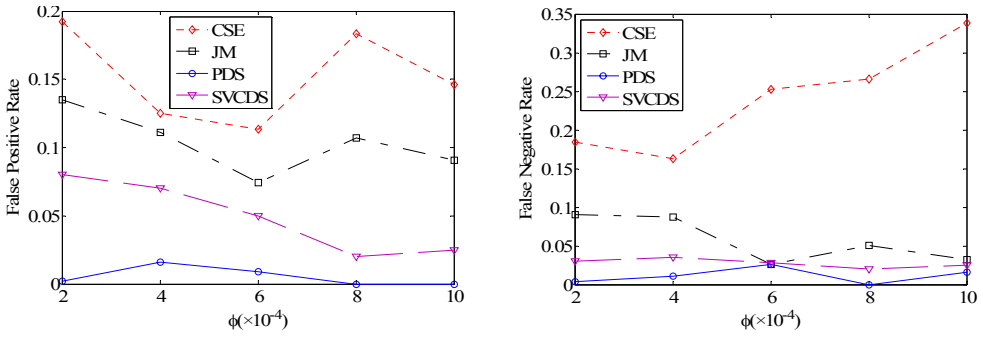
图 11 CSE,JM,SVCDS,PDS(SPort||SIP,DIP)的精度比较



(a) 误报率 (b) 漏报率

Fig.12 Comparison of accuracy of CSE, JM, SVCDS, PDS(SIP,DPort||DIP)

图 12 CSE,JM,SVCDS,PDS(SIP,DPort||DIP)的精度比较



(a) 误报率 (b) 漏报率

Fig.13 Comparison of accuracy of CSE, JM, SVCDS, PDS(DPort||DIP,SIP)

图 13 CSE,JM,SVCDS,PDS(DPort||DIP,SIP)的精度比较

4.4.2 存储开销

为了减少内存开销,CSE,JM,SVCDS 方法均使用位向量标识每个节点的链接信息.因此,节点链接度的准确估计依赖于位向量的大小.虽然 CSE 方法为每个节点建立了一个虚拟位向量,但其占用的内存空间主要由位向量的大小决定.JM 方法利用 hash 表存储每个节点的链接度信息,其占用的内存空间主要由 hash 表的大小决定.SVCDS 方法为每个节点建立了多个虚拟位向量,但其占用的内存空间主要由位向量的大小决定.假设测量的时间区间内总链接度为 N,对于每个不同的链接,PDS 方法更新(H+H^*)个位,则其所需的内存空间为 $N\times(H+H^*)$;SVCDS 方法更新 H 个位,则其所需的内存空间为 $N\times H$;JM 方法查询位向量与更新 hash 表,则其所需的内存空间为 N+64×N;CSE 方法更新 1 位,则其所需的内存空间为 N.假设总链接度为 1M,分析在不同的阈值下,4 种方法的内存开销.为 PDS 方法分配 256KB 内存空间,每个位数组的大小为 128KB(32bits×32K),从而使得该方法能够在不同的阈值下进行有效的检测.已知位向量的大小为 m,仅能准确估计链接度小于 $m\ln m$ 的节点.其原因在于:如果节点链接度的真实值大于 $m\ln m$,位向量中的所有 1 以高概率出现,从而获得的唯一信息就是节点链接度不小于 $m\ln m$.只要阈值小于 $m\ln m$,就认为该节点为超点.当 m=32 时,$m\ln m$=111>70(ϕ=2×10^{-4}).因此,对于阈值为 70(ϕ=2× 10^{-4}),140(ϕ=4×10^{-4}),210(ϕ=6×10^{-4}),280(ϕ=8×10^{-4}),350(ϕ=10×10^{-4}),m 为 32,64,64,128,128,CSE 方法中位向量的大小为 128KB,256KB,256KB,512KB,512KB.假设 JM 方法的抽样率为 1,为其分配 64bits×32K 内存空间,m=64,$m\ln m$=266>210.因此,对于阈值为 70(ϕ=2×10^{-4}),140(ϕ=4×10^{-4}),210(ϕ=6×10^{-4}),280(ϕ=8×10^{-4}),350(ϕ=10×10^{-4}),m 为 64,64,64,128,128,JM 方法中位向量的大小为 256KB,256KB,256KB,512KB,512KB.JM 方法的抽样率越小,其所需的内存空间也较小.假设 SVCDS 方法的抽样率为 1,对于阈值为 70(ϕ=2×10^{-4}),140(ϕ=4×10^{-4}),210(ϕ=6×10^{-4}),280(ϕ=8×10^{-4}),350(ϕ=10×10^{-4}),m 为 32,64,64,128,128,SVCDS 方法中位向量的大小为 128KB,256KB,256KB,512KB,512KB.

4 种方法所需的内存开销见表 4.

Table 4 Comparison of memory overhead of CSE, JM, SVCDS, PDS based super point detection (KB)

表 4 CSE,JM,SVCDS,PDS 的超点检测的内存开销比较 (KB)

超点检测方法	阈值 ϕ(×10^{-4})				
	2	4	6	8	10
CSE	128	256	256	512	512
JM	256	256	256	512	512
SVCDS	128	256	256	512	512
PDS	256	256	256	256	256

由表 4 可知:随着阈值的增加,4 种方法所需的内存空间均呈上升趋势.

4.4.3 时间开销

PDS 方法涉及 hash 运算、内存访问等,这些操作均需消耗一定的时间.为了能够直观地呈现 PDS 方法的时间效率优势.

- 一方面,将 4 种方法的执行时间进行比较.图 14 显示了 CSE,JM,SVCDS,PDS(SIP,DPort||DIP)的超点检测的处理时间.从图 14 可知:随着 ϕ 的增加,PDS(SIP,DPort||DIP)方法的超点检测的执行时间逐渐缩短,而 CSE,JM,SVCDS 方法的超点检测的执行时间只发生微小的变化,当 ϕ=8×10^{-4} 时,PDS(SIP,DPort||DIP)方法的超点检测的执行时间开始低于 CSE,JM,SVCDS 方法,表明 ϕ 的变化影响了超点检测的时间效率;
- 另一方面,通过实验将 PDS 方法与利用单线程实现的超点检测方法进行比较,用 DS 表示单线程实现的超点检测方法.两种方法的时间开销如图 15 所示.由图 15 可知:随着 ϕ 的增加,PDS 方法、DS 方法的时间开销先是显著下降然后再缓慢下降,而 PDS 方法的时间开销始终小于 DS 方法的时间开销,从而表明了 PDS 方法的实时性优势.

本文只给出了 Trace1 的超点检测的性能分析,Trace2 具有相似的结论.

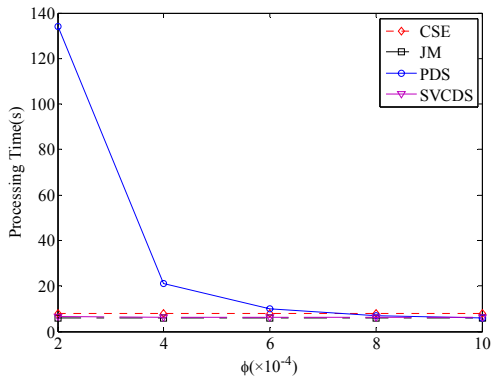


Fig.14 Comparison of processing time of CSE, JM, SVCDS, PDS

图 14 CSE,JM,SVCDS,PDS 的执行时间比较

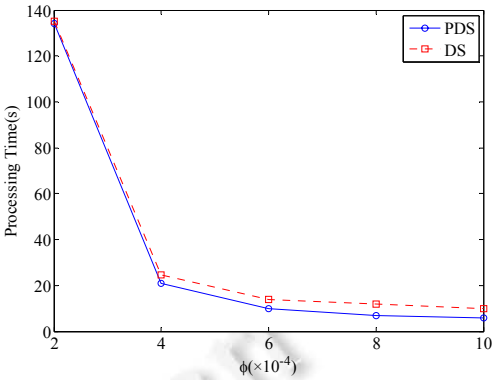


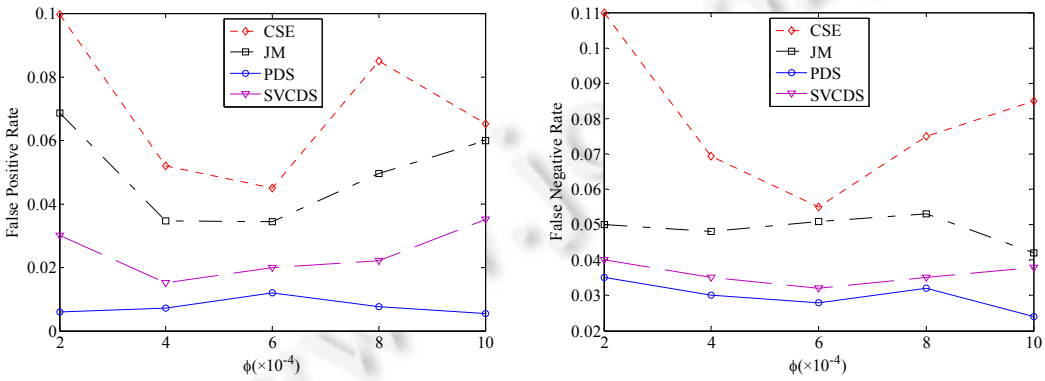
Fig.15 Comparison of processing time of DS, PDS

图 15 DS,PDS 的执行时间比较

4.5 PDS在分布式超点监控中的应用

目前,大部分的超点监控任务是在单个监测点上实施的.实际上,许多网络安全事件(如 DDoS 攻击、蠕虫传播、端口扫描)需要在多个监测点上协作完成检测任务.例如,要在多个网络接入点检测 DDoS 攻击,根据归并后的链接度大小判断遭受攻击的牺牲者.当 DDoS 攻击发生时,由于大量的攻击者可能分布于互联网的不同位置,攻击者所产生的流量通过不同的路径到达目标服务器,此时,依据单个监测点上检测目标服务器的链接度大小可能不足以识别超点,只有归并多个监测点上的链接度才能发现牺牲者.因此,分布式超点监控有助于检测网络安全事件.

在分布式超点监控应用中,每个监测点执行超点检测算法产生计算和存储开销.同时,由于监测点分布在互联网的不同位置,监测点之间需要通信协作来完成检测任务,此过程产生一定的通信开销.为了减少通信开销,只有在测量时间周期结束后才归并各个监控点的链接度信息,将链接度超过总链接度的一定百分比的节点检测为超点. DDoS Attack 2007 数据集[30]是 CAIDA 在 2007 年 8 月 4 日 DDoS 攻击发生过程中采集的流量,流量 trace 的持续时间为 1 小时,大小为 5.3GB(压缩后),只包含向牺牲者的攻击流量和向攻击者的响应流量.利用上述部分 DDoS 攻击流量来测试超点检测方法的性能.通过误报率和漏报率评价超点检测的精度,利用执行时间来评价超点检测的时间效率.如图 16、图 17 所示:对多种超点检测方法进行对比,本文方法在检测精度和时间方面具有较好的性能优势,满足分布式超点监控的应用需求.



(a) 误报率 (b) 漏报率

Fig.16 Comparison of accuracy of super point detection

图 16 超点检测的精度比较

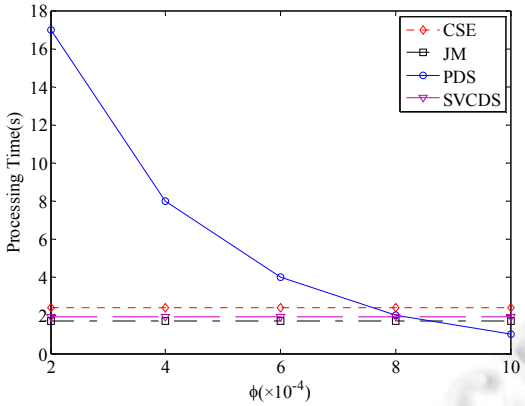


Fig.17 Comparison of processing time of super point detection

图 17 超点检测的执行时间比较

5 结束语

由于基于流抽样的超点检测方法存在计算负荷重、检测精度低、实时性差等问题,在多核处理器计算平台上,本文提出了一种并行数据流方法.本文从理论上分析了 PDS 方法的存储开销、准确性及计算开销,讨论了可逆 Sketch 数据结构中参数对检测精度的影响.通过实验验证了 PDS 方法的有效性,并和相关方法进行了比较.实验结果表明:PDS 方法的链接度估计精度、超点检测精度及处理时间均优于 CSE,JM 方法.同时,PDS 方法占用的存储空间也较小.因此,PDS 方法能够满足高速网络流量监测的应用需求.将 PDS 方法部署到实际网络中的多个测量点,实施在线的分布式的流量监控,以进一步验证 PDS 方法的有效性,这将是下一步的研究方向.

References:

[1] Cheng G, Gong J, Ding W, Wu H, Qiang SQ. Adaptive sampling algorithm for detection of superpoints. Science in China Series (E: Information Sciences), 2008,38(10):1679−1696 (in Chinese with English abstract).

[2] Zhao Q, Xu J, Kumar A. Detection of super sources and destinations in high-speed networks: Algorithms, analysis and evaluation. IEEE Journal on Selected Areas in Communications, 2006,24(10):1840−1852. [doi: 10.1109/JSAC.2006.877139]

[3] Roesch M. Snort: Lightweight intrusion detection for networks. In: Proc. of the LISA. Washington: USENIX Association, 1999. 229−238. http://www.usenix.org

[4] Plonka D. FlowScan: A network traffic flow reporting and visualization tool. In: Proc. of the LISA. USENIX Association, 2000. 305−317. http://www.usenix.org

[5] Yoon MK, Li T, Chen S, Peir JK. Fit a compact spread estimator in small high-speed memory. IEEE/ACM Trans. on Networking, 2011,19(5):1253−1264. [doi: 10.1109/TNET.2010.2080285]

[6] Wu KD. Microprocessor: Multicore has become main stream. 2009 (in Chinese). http://tech.sina.com.cn/h/2009-04-09/1337298708 2.shtml

[7] Sekar V, Reiter MK, Willinger W, Zhang H, Kompella RR, Andersen DG. cSamp: A system for network-wide flow monitoring. In: Proc. of the NSDI. San Francisco: USENIX Association, 2008. 233−246. https://www.usenix.org

[8] Henke C, Schmoll C, Zseby T. Empirical evaluation of hash functions for multipoint measurements. ACM SIGCOMM Computer Communication Review, 2008,38(3):39−50. [doi: 10.1145/1384609.1384614]

[9] Wang P, Guan X, Qin T, Huang Q. A data streaming method for monitoring host connection degrees of high-speed links. IEEE Trans. on Information Forensics and Security, 2011,6(3):1086−1098. [doi: 10.1109/TIFS.2011.2123094]

[10] Whang KY, Vander-Zanden BT, Taylor HM. A linear-time probabilistic counting algorithm for database applications. ACM Trans. on Database Systems, 1990,15(2):208−229. [doi: 10.1145/78922.78925]

[11] Wang HB, Cheng SD, Lin Y. On flow sampling for identifying super-connection hosts in high speed networks. Acta Electronica Sinica, 2008,36(4):809−818 (in Chinese with English abstract).

[12] Venkataraman S, Song D, Gibbons PB, Blum A. New streaming algorithms for fast detection of superspreaders. In: Proc. of the NDSS. San Diego: ISOC, 2005. 1−18. http://repository.cmu.edu/

[13] Estan C, Varghese G, Fisk M. Bitmap algorithms for counting active flows on high speed links. In: Proc. of the 3rd ACM SIGCOMM Conf. on Internet Measurement. New York: ACM Press, 2003. 153−166. [doi: 10.1145/948205.948225]

[14] Kamiyama N, Mori T, Kawahara R. Simple and adaptive identification of superspreaders by flow sampling. In: Proc. of the INFOCOM. Anchorage: IEEE, 2007. 2481−2485. [doi: 10.1109/INFCOM.2007.305]

[15] Cao J, Jin Y, Chen A, Bu T, Zhang Z. Identifying high cardinality internet hosts. In: Proc. of the INFOCOM. Rio de Janeiro: IEEE, 2009. 810−818. [doi: 10.1109/INFCOM.2009.5061990]

[16] Guan X, Wang P, Qin T. A new data streaming method for locating hosts with large connection degree. In: Proc. of the GLOBECOM. Honolulu: IEEE, 2009. 1−6. [doi: 10.1109/GLOCOM.2009.5426280]

[17] Li T, Chen S, Luo W, Zhang M. Scan detection in high-speed networks based on optimal dynamic bit sharing. In: Proc. of the INFOCOM. Shanghai: IEEE, 2011. 3200−3208. [doi: 10.1109/INFCOM.2011.5935169]

[18] Wang P, Guan X, Towsley D, Tao J. Virtual indexing based methods for estimating node connection degrees. Computer Networks, 2012,56(12):2773−2787. [doi: 10.1016/j.comnet.2012.03.025]

[19] Shi X, Chiu DM, Lui J. An online framework for catching top spreaders and scanners. Computer Networks, 2010,54(9):1375−1388. [doi: 10.1016/j.comnet.2009.12.003]

[20] Shin S, Im E, Yoon M. A grand spread estimator using a graphics processing unit. Journal of Parallel and Distributed Computing, 2014,74(2):2039−2047. [doi: 10.1016/j.jpdc.2013.10.007]

[21] Das S, Antony S, Agrawal D, Abbadi AE. CoTS: A scalable framework for parallelizing frequency counting over data streams. In: Proc. of the ICDE. Shanghai: IEEE, 2009. 1323−1326. [doi: 10.1109/ICDE.2009.231]

[22] Das S, Antony S, Agrawal D, Abbadi AE. Thread cooperation in multicore architectures for frequency counting over multiple data streams. In: Proc. of the VLDB. Lyon: ACM Press, 2009. 217−228. [doi: 10.14778/1687627.1687653]

[23] Cafaro M, Tempesta P, Pulimeno M. A paralleld space saving algorithm for frequent items and the Riemann-Hurwitz zeta distribution. Technical Report, CRM-3322, 2012. http://www.crm.umontreal.ca/pub/Rapports/3300-3399/3322.pdf

[24] Cafaro M, Tempesta P. Finding frequent items in parallel. Concurrency and Computation: Practice and Experience, 2011,23(15): 1774−1788. [doi: 10.1002/cpe.1761]

[25] Zhang Y, Fang B, Zhang Y. Parallelizing weighted frequency counting in high-speed network monitoring. Computer Communications, 2011,34(4):536−547. [doi: 10.1016/j.comcom.2010.04.026]

[26] Zhang Y, Sun Y, Zhang J, Xu J, Wu Y. An efficient framework for parallel and continuous frequency item monitoring. Concurrency and Computation: Practice and Experience, 2014,26(18):2856−2879. [doi: 10.1002/cpe.3182]

[27] Cormen TH, Leiserson CE, Rivest RL, Stein C. Introduction to Algorithms. 3rd ed., Cambridge: The MIT Press, 2009. 926−954.

[28] Zheng X. IP Trace distribution system. 2011. http://iptas.edu.cn/src/system.php

[29] Claffy KC. The CAIDA UCSD anonymized Internet traces 2012. 2012. http://www.caida.org/passive passive_2012_dataset.xml

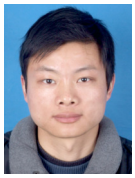
[30] Claffy KC. The CAIDA DDoS attack 2007 dataset. 2007. http://www.caida.org/data/passive/ddos-20070804_dataset.xml

附中文参考文献:

[1] 程光,龚俭,丁伟,吴桦,强士卿.基于自适应抽样的超点检测算法.中国科学(E 辑:信息科学),2008,38(10):1679−1696.

[6] 吴康迪.微处理器:多核已成为主流.2009. http://tech.sina.com.cn/h/2009-04-09/13372987082.shtml

[11] 王洪波,程时端,林宇.高速网络超连接主机检测中的流抽样算法研究.电子学报,2008,36(4):809−818.



周爱平(1982－),男,江苏泰州人,博士生,CCF 学生会员,主要研究领域为网络测量,网络安全.

郭晓军(1983－),男,博士生,讲师,主要研究领域为网络测量,网络安全.

程光(1973－),男,博士,教授,博士生导师,CCF 高级会员,主要研究领域为网络测量,网络安全,网络管理.

梁一鑫(1980－2016),男,博士生,讲师,主要研究领域为流量管理,传输控制.