分规则(Ruleset Partition) 并发检测模型

目录

- 现有IDS负载均衡技术介绍
- IDS检测性能影响参数
- 规则负载均衡技术

1. 现有IDS负载均衡技术介绍

- 基于流(flow based)的负载均衡[1]: 系统维持着一张流表(记录着活动流和并发运行着的线程之间的对应关系), 每个到达的报文都先和对应的流进行匹配，再查询上述的表，将报文分配到对应的线程进行相应的处理。但flow based负载均衡不能完全利用并行，因为存在着不均匀的流量 空间(space)、持续时间(duration)、大小(size)的分配。

- 自适应(adaptive)的负载均衡[2]: 能够通过对流量的预测来调整处理器之间的调度比，这可能能缓解flow based 负载均衡出现的问题，但是adaptive 负载均衡本质上依旧是flow based.而且，adaptive负载均衡由于自身的复杂性将不可避免的增加系统的开销。

1. 现有IDS负载均衡技术介绍

- 基于流量(flow based)的负载均衡:
  - 典型架构：Suricata: pipeline parallelization with static LB scheme
  - 架构图:

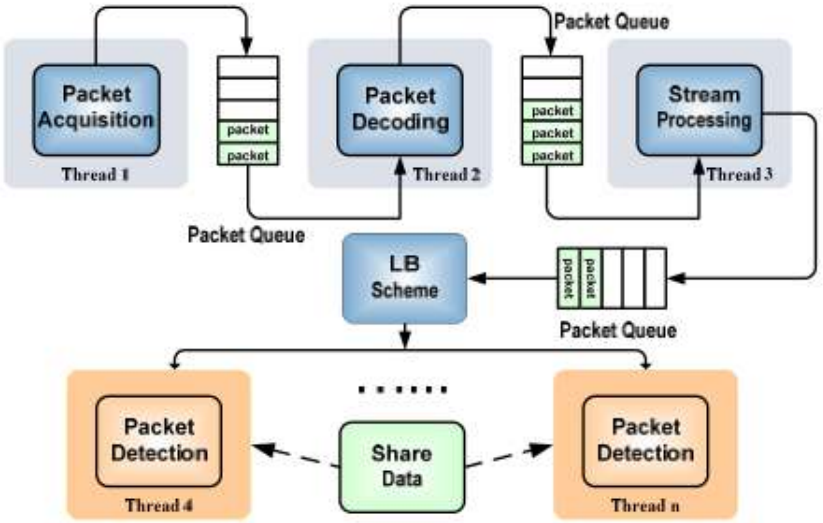

Fig.1 Parallel architecture of Suricata

  - 相关LB说明：最新的suricata默认使用static hash作为负载均衡（E.g. 使用tcp/ip报文头中的五元组进行hash)。当然由于suricata的高度模块化，所以我们可以很容易的对LB scheme模块进行替换·(这有利于我们针对不同LB方式进行测试和比较）。

1.现有IDS负载均衡技术介绍

- 基于自适应(adapative)的负载均衡:
  - 典型架构： Para-Snort[7]: Parallel Snort with adaptive LB scheme
  - 架构图:

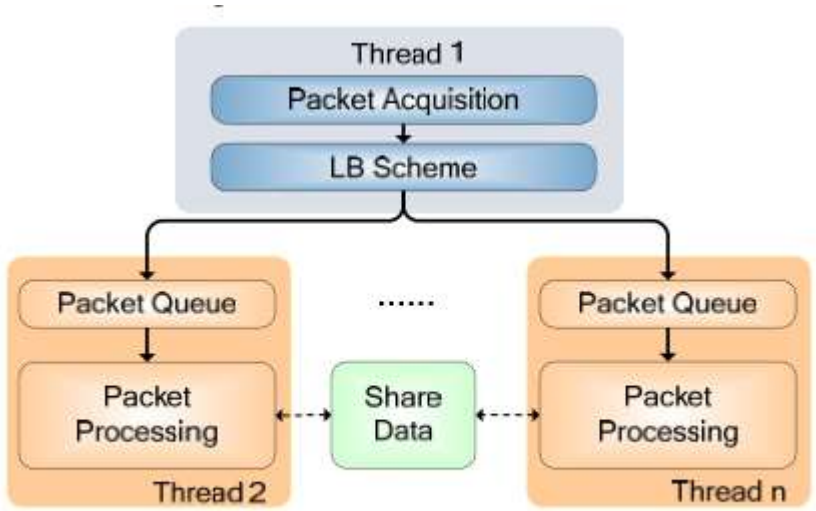

Fig.2 Parallel architecture of Para-Snort

  - 相关LB说明：para-snort 使用M- JSQ(modified Join-the-Shortest-Queue)进行自适应的负载均衡。每当属于一个流的报文到达时，若此流并未分配到一个线程上，那么(采用)JSQ(的进程)会查看所有的报文处理(packet processing)线程的缓存队列，并选出拥有最短缓存队列长度的线程来处理这个新的流。而M-JSQ是对JSQ进行了相应的扩展。

2. IDS检测性能影响参数

- 此文[3]将报文检测(PD)模块分为两个部分,进行分析:
  - 预过滤器(pre-filter): 使用多模式匹配(Multi-Pattern Matching)指纹识别(fingerprint)引擎过滤掉那些明显良性的报文,注: **fingerprint集合是从每条规则中选取的一小部分模式(pattern)组成的**。
  - 验证单元(Verification Units): 结合pre-filter中的指纹证据对可疑的报文进行深入(in-depth)的分析。

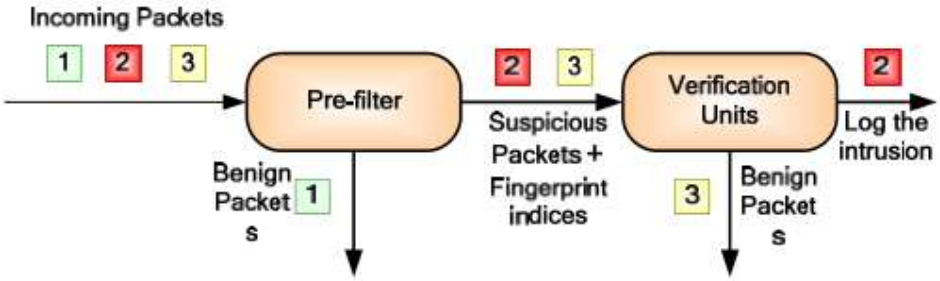

Fig.3 *packet detection* (PD) module in IDS

- 为了分析报文的处理时间，此文[3]作者建立了一个简单的模型，来计算PD模块处理每一个报文的平均时间。
- 公式： $T = \alpha T_{pre} + \overline{N} T_v$

其中：T_{pre} 为当pre-filter阶段只有一个fingerprint时检测一个报文的平均时间，α 为检测规则集增加时的影响系数（随着规则集的增大，由于多模式匹配引擎更容易发生TLB 和 cache 的不命中，所以此影响因子会相应的增加）。而对于VU阶段，Tv为只有一个规则时检测一个报文所需要的平均时间，N称作**验证计数**(即表示为每个报文在VU阶段需要验证的平均规则数量）。

[3] Jiang H, Xie G, Salamatian K. Load Balancing by Ruleset Partition for Parallel IDS on Multi-Core Processors[C]//International Conference on Computer Communications and Networks, ICCCN. 2013.

2. IDS检测性能影响参数

- 综合上文的分析，我们可以得到在系统配置一定的情况下，**单个**报文检测单元(PD)检测一个报文的时间和以下的参数有关：
  - 指纹(fingerprint)集合的选择: 影响所有的参数。
  - 规则集的大小(影响系数α以及验证计数N）

  如下图所示，在suricata单检测线程的情况下，使用DARPA intrusion detection evaluation traffic trace，我们可以观察到，随着规则集由1K 上升到9K， **报文通过PD单元的时间**，L2缓存的miss率，验证计数N都在显著的增加。

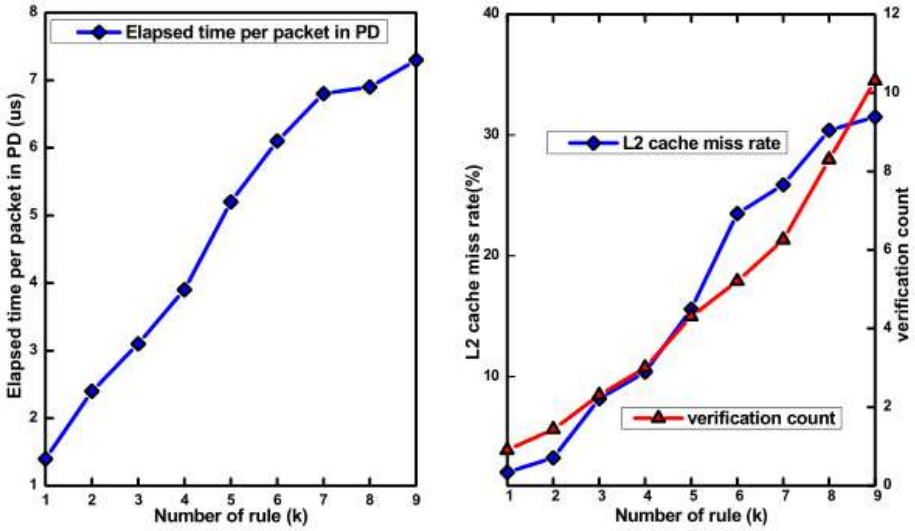

Fig.4 Performance of Suricata with 1 PD thread and increasing ruleset from 1000 to 9000 rules

3. 分规则负载均衡技术

- 3.1 背景：

- 通过上一步关于IDS检测性能影响参数的分析，可以得到IDS的检测单元在更少的规则下运行的更快，这也是规则负载均衡(ruleset partition balancing)技术出现的直接诱因。
- 分规则负载均衡技术 ：RPB 把规则集分成不同的子集，并将子集和检测线程进行一一对应。由于ruleset partition并不依赖于流量的具体特征，所以RPB比flow based 负载均衡更加健壮。再者，RPB的规则划分可以依据用户的配置信息在离线(off-line)情况下进行

- 3.2 Related work

- 关于ruleset partition的想法已经在一些现存的IDS系统[3-6]中出现，不过在这些系统所做的是pattern partition,而不是rule partition, 因为他们只考虑了pattern中的字符，而忽略了(模式中的)特征中定义的protocol信息。这些划分策略并不能很好的应用在IDS中，因为只有在规则集中有对应的特征，网络报文中的特征才能被检测出来。

3. 分规则负载均衡技术

- 3.2 Related work

- 在[4] 中，作者实现的PM(pattern matching)系统结构图如下所示，该系统的工作流分为两个步骤：离线预处理阶段和实时PM阶段。
- 在离线预处理阶段中，1) NPA(negative pattern analyzer)浏览规则集(pattern set)来获得NPs并将它们放入加速器中 2) 同时，ESP(Exclusive Set Partitioner)观测规则集，并将其划分成exclusive子集，而后将他们分别存放在相应的TCAM(三态内容寻址存储器)块中。
- 在实时PM阶段中，输入流会被分配到硬件加速器中进行内容的预过滤(pre-filering),例如对流进行分割，接着TCAM块会依据存储在其内的exclusive子集对网络流进行简单的模式检查。然后这些匹配信息（例如，匹配特定的simple pattern 的报文在流中的位置）信息会传送给general-purpose处理器进行进一步的后处理。
- 此文中，对输入流量的分配方式是**共享**，即多个TCAM块共用一份输入流。

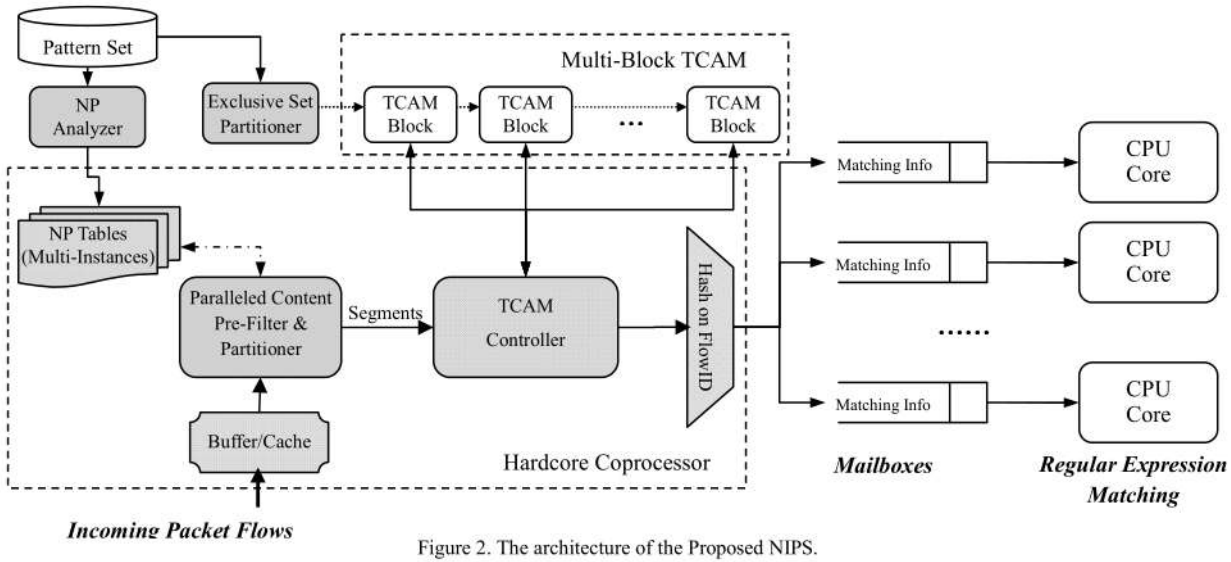

Figure 2. The architecture of the Proposed NIPS.

3. 分规则负载均衡技术

3.2 Related work

- 在[6]中，作者设计的HPSMM(hybrid parallel signature matching model)如Fig.2 所示，在此模型中，SIMD GPU作为多核CPU的一个协同处理器(co-processor). 系统中采用了pattern set partition（不同的multiprocessor间），以及Input Text Partition(单个multiprocessor内不同的Stream Processor, 参见Fig. 3)，藉此来达到最大化并行的目的。
- 该系统中，pattern set partitioning 和 input text partitioning 按如下方式协同工作：
- 对于不同的multiprocessor，每一份Input text 都被发送到所有的multiprocessor, 然后系统用partition set partition对规则集进行划分，并分配给不同的multiprocessor。简言之就是，不同的multiprocessor工作在相同的输入数据上，但是拥有不同的规则子集。
- 对于单个multiprocessor内的不同流处理器(stream processor)而言，它们操作在相同的规则子集上，但是不同的流处理器处理Input Text不同的部分(输入数据的划分是在多核CPU上进行的，放入不同的缓冲区中，然后与GPU进行批量数据交换）。

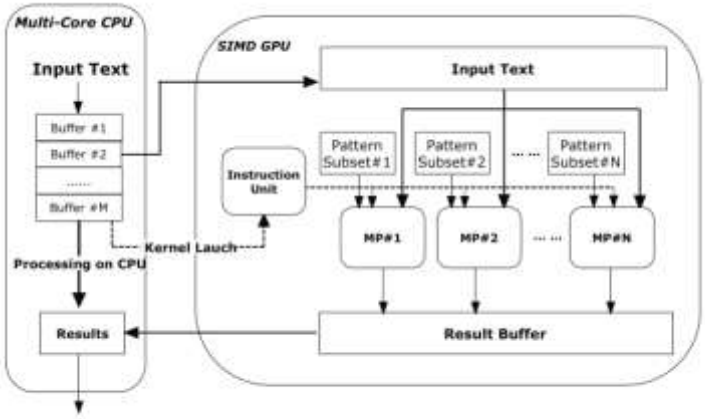

Fig. 2. The hybrid parallel signature matching model - HPSMM

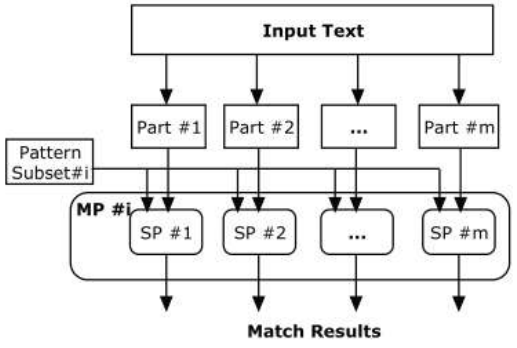

Fig. 3. Input text partition inside each multiprocessor

3. 分规则负载均衡技术

3.3 Methodology and algorithm

RPB中的关键问题是如何进行规则集的划分。

论文[3]提出了两种方法：

1. Simple partition： 给每个规则一个单调递增的ID号，并使用hash函数将其均匀分配给特定的检测单元。
2. Partition with Leaf Pruning: 先把规则集组织成Linked Rule Array(LRA)数据结构，然后应用Leaf Pruning(LP)策略进行划分。

3. 分规则负载均衡技术

- Partition with Leaf Pruning

1. 首先，我们把整个规则集组织成Linked Rule Array(LRA)结构。LRA的结构如下图所示，每一层以一个特征进行区分，从根到叶节点组成一个具有相同特征集的规则子集（E.g.图中是以tcp源宿端口作为划分的特征）。

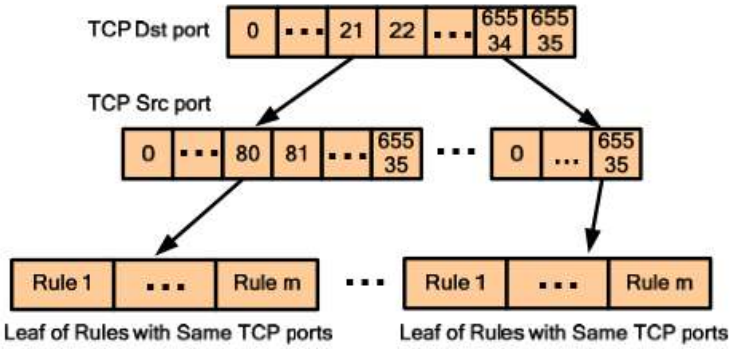

Fig.6 LRA structure

2. 然后对每一个叶子节点中的规则子集采取simple partition均匀划分到每一个检测单元中。

3. 分规则负载均衡技术

- 3.4 Validation and Comparision

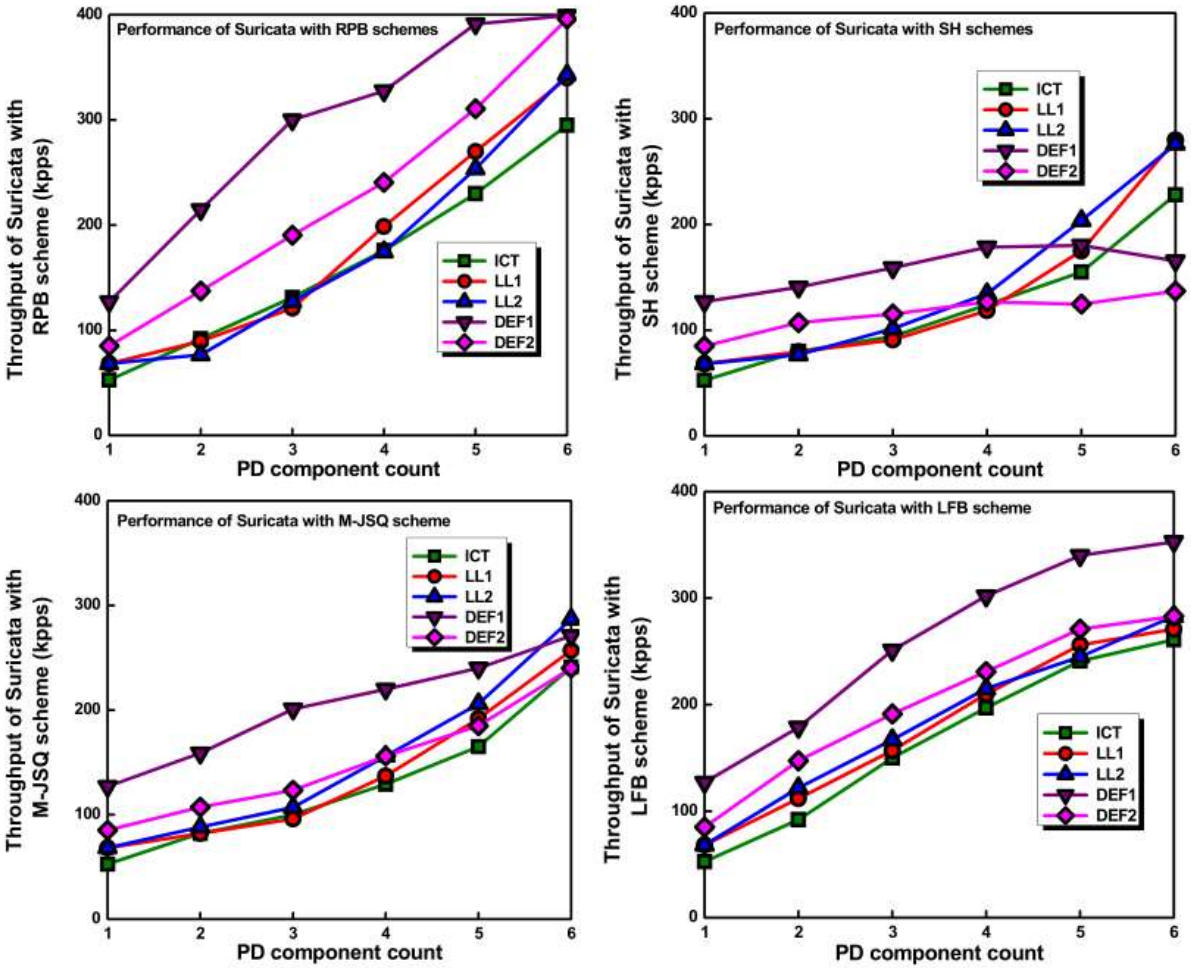

Fig.8 Performance of Suricata with different LB schemes (RPB, SH, M-JSQ and LFB)

Trace LL1 and LL2： come from MIT Lincoln Lab,
DEF1 & DEF2： the DefCon traces
trace ICT： was captured in 2012 from the export of a Giga-Ethernet link of the operating network in the Institute of Computing Technology of the Chinese Academy of Science.

SH: static hash
LFB [7]: Last Flow Bundle
M-JSQ [8]:
Modified Join-the-Shortest-Queue

这三种模式都是**flow based**负载均衡模式。

LFB[8]

- 基于自适应(adapative)的负载均衡
- Cache affinity LB scheme

- Last Flow Bundle (LFB)被设计用来最大化多核系统的cache亲和度。LFB通过确保单个线程只处理同一流的所有报文,而不是对来自不同流的报文进行交叉处理，借此来提高cache的命中率，进而提高系统的检测效率。

参考文献

[1] Guo, D., et al. 2008. A scalable multithreaded L7-filter design for multi-core servers. In Proceedings of the 3th ACM/IEEE Symposium on Architectures For Networking and Communications Systems (San Jose,California, November 06 - 07, 2008).

[2] Martin, R., et al. 2006. Accuracy and dynamics of hash-based load balancing algorithms for multipath Internet routing. Broadband Communications, Networks and Systems, 2006. BROADNETS 2006. 3rd International Conference, vol., no., pp.1-10, 1-5 Oct. 2006.

[3] Jiang H, Xie G, Salamatian K. Load Balancing by Ruleset Partition for Parallel IDS on Multi-Core Processors[C]//International Conference on Computer Communications and Networks, ICCCN. 2013.

[4] Kai Zheng, Xin Zhang, Zhiping Cai, Zhijun Wang, Baohua Yang:Scalable NIDS via Negative Pattern Matching and Exclusive Pattern Matching. INFOCOM 2010: 731-739

[5] Hyunjin Kim, Hyejeong Hong, Dongmyong Baek, Sungho Kang: A Pattern Partitioning Algorithm for Memory-Efficient Parallel String Matching in Deep Packet Inspection. IEICE Transactions 93-B(6): 1612-1613 (2010)

[6] Wu, C., Yin, J., Cai, Z., Zhu, E., Chen, J.: A Hybrid Parallel Signature Matching Model for Network Security Applications Using SIMD GPU. In: Dou, Y., Gruber, R., Joller, J.M. (eds.) APPT 2009. LNCS, vol. 5737, pp. 191–203. Springer, Heidelberg (2009)

参考文献(cont.)

[7] Xinming Chen, Yiyao Wu, Lianghong Xu, Yibo Xue and Jun Li. Para-Snort: A Multi-thread Snort on Multi-Core IA Platform. Proc. of the 21th IASTED International Conference on Parallel and Distributed Computing and Systems (PDCS), 2009.

[8] T. Nelms and M. Ahamad, "Packet scheduling for deep packet inspection on multi-core architectures", in Proc. ANCS, 2010, pp.21-21.