GraphBLAS 在边缘的应用：网络流量的匿名化高性能流处理

Michael Jones[1], Jeremy Kepner[1], Daniel Andersen[2], Aydın Buluç[3], Chansup Byun[1], K Claffy[2], Timothy Davis[4], William Arcand[1], Jonathan Bernays[1], David Bestor[1], William Bergeron[1], Vijay Gadepally[1], Micheal Houle[1], Matthew Hubbell[1], Hayden Jananthan[1], Anna Klein[1], Chad Meiners[1], Lauren Milechin[1], Julie Mullen[1], Sandeep Pisharody[1], Andrew Prout[1], Albert Reuther[1], Antonio Rosa[1], Siddharth Samsi[1], Jon Sreekanth[5], Doug Stetson[1], Charles Yee[1], Peter Michaleas[1]
[1]MIT, [2]CAIDA, [3]LBNL, [4]Texas A&M, [5]Accolade Technology

摘要—远程检测是许多操作领域（陆地、海洋、水下、空中、太空等）防御的基石。在网络空间，远程检测需要分析来自各种观测站和前哨站的重要网络流量。在边缘网络设备上构建匿名超稀疏流量矩阵可以成为关键推动因素，通过提供显著的数据压缩格式来保护隐私并实现快速分析。GraphBLAS 非常适合于构造和分析匿名超稀疏流量矩阵。在一个接近最坏情况的交通场景中，使用 CAIDA 望远镜暗网数据包的连续流，在 Accolade Technologies 边缘网络设备上展示了 GraphBLAS 的性能。探讨了不同数量的流量缓冲区、线程和处理器核心的性能。匿名超稀疏流量矩阵可以以超过每秒 50,000,000 个数据包的速度构建；超过了典型的 400 千兆位网络链路。这一性能表明，可以在边缘网络设备上使用最少的计算资源轻松计算出匿名超稀疏流量矩阵，并且可以成为此类设备的有效数据产品。

***Index Terms*—互联网防御，数据包捕获，流图，超稀疏矩阵**

I. 介绍

互联网已成为商业、教育、健康和娱乐等活动必不可少的组成部分，与陆地、海洋、空中和太空一样重要 [1], [2]。自古代以来，远程探测一直是许多操作领域防御的核心 [3]–[10]。在网络空间中，已经建立了观测站和前哨来收集互联网流量数据，并提供探索远程检测的起点 [11]–[17]（见图 1）。最大的公共互联网观测站是应用互联网数据分析中心（CAIDA 望远镜），它运行各种传感器，包括来自未请求暗空间的持续数据包流，这大约占互联网的 1/256 部分 [18]–[21]。一般来说，远程探测需要分析来自多个观测站和前哨的重要网络流量 [22], [23]。分析互联网显著部分的数据量、处理需求以及隐私问题一直是难以克服的问题。北美互联网每秒生成数十亿个非视频互联网数据包 [1], [2]。GraphBLAS 标准提供了显著的性能和压缩能力，提高了分析这些数据量的可能性 [24]–[38]。特别是，GraphBLAS 非常适合构建和分析匿名的超稀疏流量矩阵。之前使用 GraphBLAS 的工作已经展示了每秒 750 亿个数据包 (pps) [39] 的速率，同时实现了每个数据包 1 比特 [17] 的压缩，并能够分析最大的公共历史存档，超过 40 万亿个数据包 [40]。来自各种来源的匿名超稀疏流量矩阵的分析揭示了幂律分布 [41], [42]，新颖的比例关系 [17], [40]，并激发了新的网络流量模型 [43]。

GraphBLAS 匿名超稀疏流量矩阵代表了一组用于分析网络流量的设计选择。具体而言，这种应用场景需要所有数据包的一些数据（不进行降采样）、高性能、高压缩率、基于矩阵的分析、匿名化和开放标准。存在广泛的替代图/网络分析技术，并且许多优秀的实现都能达到底层计算硬件性能极限 [44]–[54]。同样，有许多网络分析工具专注于为在网络流量中发现的数据的丰

This material is based upon work supported by the Under Secretary of Defense for Research and Engineering under Air Force Contract No. FA8702-15-D-0001, National Science Foundation CCF-1533644, and United States Air Force Research Laboratory and Artificial Intelligence Accelerator Cooperative Agreement Number FA8750-19-2-1000. Any opinions, findings, conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the Under Secretary of Defense for Research and Engineering, the National Science Foundation, or the United States Air Force. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

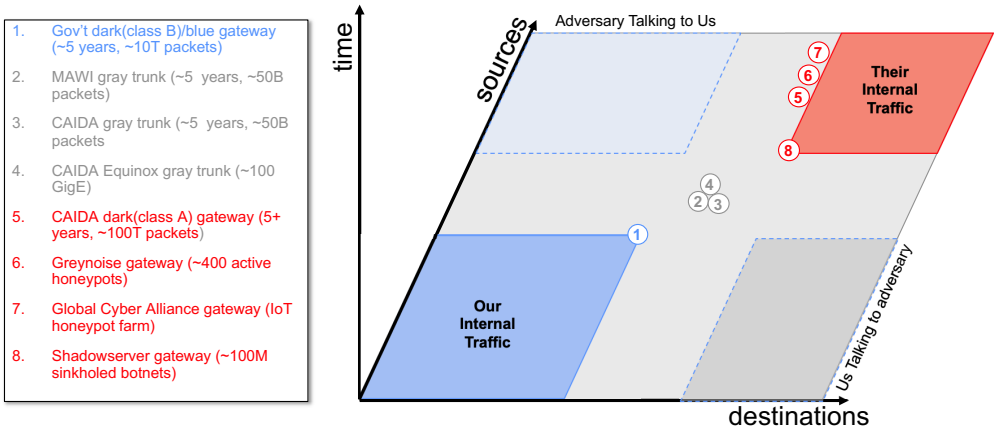

图 1. **互联网观测站和前哨**。互联网中的流量矩阵视图，展示选定的观测站和前哨及其与各种类型网络流量的假设接近度 [11]–[17]。

富多样性提供丰富的接口 [55]–[57]。这些技术中的每一种在互联网流量分析这一广泛领域内都有适当的应用场景。

将大量原始互联网流量发送到一个中央位置以构建匿名的超稀疏流量矩阵是不切实际的。为了实现提供所有数据包的一些信息而不进行降采样的目标，需要在网络本身中构建匿名的超稀疏流量矩阵，以便实现全面的数据压缩效益。本文的目标是通过测量边缘网络设备上的 GraphBLAS 性能来探讨这种方法的可行性。性能是在接近最坏情况的流量场景下使用 CAIDA 望远镜暗网数据包（主要是僵尸网络和扫描器）进行测量的，这些数据包具有不规则分布且几乎没有有效载荷数据（即全是头部）。

本文其余部分的提纲如下。首先，定义了 CAIDA 望远镜测试数据以及一些以流量矩阵形式表示的基本网络量。接下来，描述了匿名超稀疏流量矩阵管道，并随后介绍了实验设置和实现。最后，展示了结果、结论及未来工作的方向。

II. 测试数据和流量矩阵

测试数据来自 CAIDA 望远镜暗网数据包（主要是僵尸网络和扫描器），这几乎是最糟糕的情况，具有高度不规则的分布且几乎没有有效载荷数据（即全是头部）。CAIDA 望远镜监控一组网络地址的进出流量，提供了一个观察网络流量的自然观测点。这些数据可以视为一个流量矩阵，每一行是一个源，每一列是一个目的地。CAIDA 望远镜的流量矩阵可以划分为四个象限（见图 2）。这四个象限代表了被监控地址集内外节点之间的不同流向。因为 CAIDA 望远镜网络地址是暗空间，只有左上角（外部 → 内部）象限将有数据。

互联网数据必须谨慎处理，CAIDA 率先提出了可信的数据共享最佳实践，这些实践结合了使用 CryptoPAN [58] 对源和目的地进行匿名化与数据共享协议。这些数据共享的最佳实践是此处提出的架构的基础，并包括以下原则 [22]

- 数据在整理过的仓库中提供访问
- 使用需要的标准匿名化方法：哈希、采样和/或模拟
- 注册到存储库并展示合法的研究需求
- 接收者依法同意既不重新发布语料库也不去匿名化数据
- 接收者可以发布审查研究所需的分析和数据示例
- 接受者同意引用该仓库并回传发表物到该仓库
- 存储库可以整理研究人员开发的丰富产品

构建匿名超稀疏流量矩阵的主要好处是通过矩阵数学高效计算各种网络量。图 3描述了所有流动态网络中发现的基本量。这些量都可以从源和目标创建的匿名流量矩阵中计算得出，这些源和目标来自互联网数据包头。

图 3中显示的网络量可从广泛用于表示网络流量 [60]–[63] 的匿名源-目的交通矩阵中计算得出。通常会过滤数据包以得到任何特定分析的有效集合。这样的过滤器可能会限制特定的来源、目的地、协议和时间窗口。为了减少统计波动，应将流数据分割为选定的时间窗口内所有数据集具有相同数量的有效数据包 [64]。在给定时间 t，N_V 个连续的有效数据包被聚合到流量中的一个超稀疏矩阵 $\mathbf{A}_t$ 中，其中 $\mathbf{A}_t(i,j)$ 是源 i 和目的地 j 之

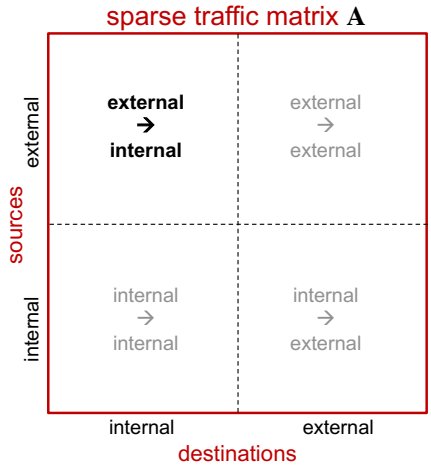

图 2. **网络流量矩阵**。交通矩阵可以分为四个象限，分离内部和外部流量。CAIDA 望远镜监控一片黑暗空间，因此只有左上（外部 → 内部）象限将会有数据。

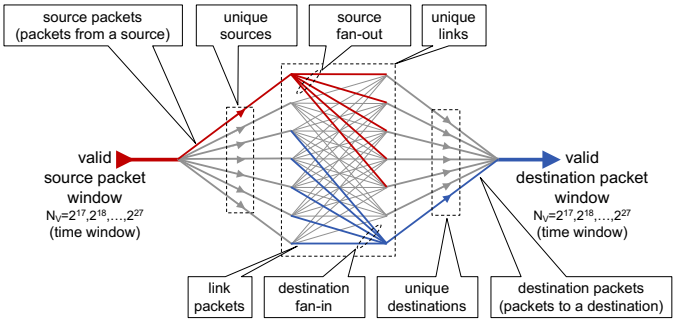

图 3. **流式网络流量数量**。互联网流量流中的 N_V 有效数据包被分为多种数量进行分析：源数据包、源扇出、唯一源目的地对数据包（或链接）、目的扇入和目的数据包。

间的有效数据包数量。$\mathbf{A}_t$ 中所有条目的总和等于 N_V

$$\sum_{i,j} \mathbf{A}_t(i,j) = N_V$$

恒定的数据包，可变的时间样本简化了对网络流量数量中常见的重尾分布的统计分析 [41], [42], [65]。图 3中描绘的所有网络数量都可以使用表 I列出的公式从 $\mathbf{A}_t$ 计算得出。由于矩阵运算通常对置换（行和列的重新排序）不变，因此可以从匿名数据中轻松计算这些量。此外，可以通过简单的矩阵乘法分析 IP 的子范围中的匿名数据。对于由匿名超稀疏对角矩阵 $\mathbf{A}_r$ 表示的子范围，其中 $\mathbf{A}_r(i,i) = 1$ 暗指源/目的地 i 在该范围内，子范围内的流量可以通过以下方式计算：$\mathbf{A}_r\mathbf{A}_t\mathbf{A}_r$。同样，为了在边缘实现额外的隐私保护，可以使用相同的方法从

表 I
从流量矩阵中获取的网络数量

从时间 t 的流量矩阵 $\mathbf{A}_t$ 计算网络数量的公式，包括求和符号和矩阵表示。$\mathbf{1}$ 是一个全为 1 的列向量，$^\mathsf{T}$ 是转置操作，而 $|\,|_0$ 是零范数，它将其参数中的每个非零值设置为 1 [59]。这些公式不受矩阵置换的影响，并且可以在匿名数据上使用。

聚合属性	求和符号约定	矩阵记号
Valid packets N_V	$\sum_i \sum_j \mathbf{A}_t(i,j)$	$\mathbf{1}^\mathsf{T}\mathbf{A}_t\mathbf{1}$
Unique links	$\sum_i \sum_j \lvert\mathbf{A}_t(i,j)\rvert_0$	$\mathbf{1}^\mathsf{T}\lvert\mathbf{A}_t\rvert_0\mathbf{1}$
Link packets from i to j	$\mathbf{A}_t(i,j)$	$\mathbf{A}_t$
Max link packets ($d_{\max}$)	$\max_{ij} \mathbf{A}_t(i,j)$	$\max(\mathbf{A}_t)$
Unique sources	$\sum_i \lvert\sum_j \mathbf{A}_t(i,j)\rvert_0$	$\mathbf{1}^\mathsf{T}\lvert\mathbf{A}_t\mathbf{1}\rvert_0$
Packets from source i	$\sum_j \mathbf{A}_t(i,j)$	$\mathbf{A}_t\mathbf{1}$
Max source packets ($d_{\max}$)	$\max_i \sum_j \mathbf{A}_t(i,j)$	$\max(\mathbf{A}_t\mathbf{1})$
Source fan-out from i	$\sum_j \lvert\mathbf{A}_t(i,j)\rvert_0$	$\lvert\mathbf{A}_t\rvert_0\mathbf{1}$
Max source fan-out ($d_{\max}$)	$\max_i \sum_j \lvert\mathbf{A}_t(i,j)\rvert_0$	$\max(\lvert\mathbf{A}_t\rvert_0\mathbf{1})$
Unique destinations	$\sum_j \lvert\sum_i \mathbf{A}_t(i,j)\rvert_0$	$\lvert\mathbf{1}^\mathsf{T}\mathbf{A}_t\rvert_0\mathbf{1}$
Destination packets to j	$\sum_i \mathbf{A}_t(i,j)$	$\mathbf{1}^\mathsf{T}\lvert\mathbf{A}_t\rvert_0$
Max destination packets ($d_{\max}$)	$\max_j \sum_i \mathbf{A}_t(i,j)$	$\max(\mathbf{1}^\mathsf{T}\lvert\mathbf{A}_t\rvert_0)$
Destination fan-in to j	$\sum_i \lvert\mathbf{A}_t(i,j)\rvert_0$	$\mathbf{1}^\mathsf{T}\,\mathbf{A}_t$
Max destination fan-in ($d_{\max}$)	$\max_j \sum_i \lvert\mathbf{A}_t(i,j)\rvert_0$	$\max(\mathbf{1}^\mathsf{T}\,\mathbf{A}_t)$

流量矩阵

$$\mathbf{A}_t - \mathbf{A}_r\mathbf{A}_t\mathbf{A}_r$$

中排除一段数据范围。

这些数据的连续性使得可以探索从 $N_V = 2^{17}$（亚秒级）到 $N_V = 2^{27}$（分钟级）的一系列数据包窗口，提供了关于网络量如何依赖于时间的独特视角。这些观察结果为正常网络背景流量提供了新的见解，可用于异常检测、人工智能特征工程、多存储系统索引学习以及测试流式网络的理论模型 [66]–[68]。

网络流量是动态的，并在广泛的时间尺度上表现出不同的行为。给定的数据包窗口大小 N_V 将对相应时间尺度上的现象敏感。确定网络量如何与 N_V 扩展可以提供有关网络流量时间行为的见解。通过分层聚合不同时间窗口的数据，可以在多个时间尺度上高效地计算网络量 [64]。图 4说明了不同类型流式传输流量矩阵的二进制聚合。在每个层次级别计算每种数量可以消除如果分别计算每个数据包窗口时将执行的冗余计算。分层还确保大多数计算是在位于更快内存中的较小矩阵上进行的。矩阵之间的相关性意味着，两个各自包含 N_V 个条目的矩阵相加结果是一个包含少于 $2N_V$ 个条目的矩阵，随着矩阵的增长减少了相对的操作数量。

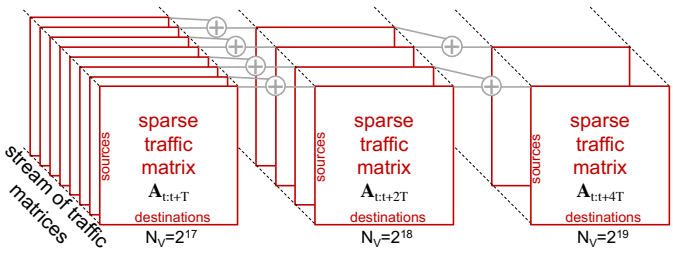

图 4. **多时态流式交通矩阵**。通过在不同时间窗口中分层聚合数据，可以实现多时间尺度上网络量的有效计算。

SuiteSparse GraphBLAS 库的一个重要功能是直接支持超稀疏矩阵，其中非零项的数量远小于矩阵的任一维度。如果数据包源和目的地标识符来自一个大的数字范围，例如在互联网协议中使用的那些，则 $\mathbf{A}_t$ 的超稀疏表示消除了跟踪额外索引的需求，并且可以显著加速计算 [39]。

III. 超稀疏矩阵流水线

前述分析目标设定了对图 BLAS 超稀疏流量矩阵管道的要求。具体来说，如果在尽可能靠近网络流量的位置在网络中构建图 BLAS 超稀疏流量矩阵，则压缩收益最大化，因为这会将需要通过网络发送的数据量降至最低。此外，在网络源头进行收集允许数据所有者构建并拥有匿名化方案，并仅在与负责分析数据的各方 [69] 签订可信的数据共享协议后分享匿名化数据。

图 GraphBLAS 超稀疏流量矩阵管道的第一步是捕获一个数据包，丢弃数据负载，并提取源和目标互联网协议（IP）地址（图 5）。为了当前的性能测试目的，仅使用存储为 32 位无符号整数的 IPv4 数据包。然后对每组 $N_V = 2^{17}$ 连续的数据包分别使用由 cryptoPAN 生成的匿名化表进行匿名处理。得到的匿名化的源和目标 IP 地址随后用于构建一个 $2^{32}\times 2^{32}$ 超稀疏 GraphBLAS 矩阵。64 个连续的超稀疏 GraphBLAS 矩阵各自以压缩稀疏行（CSR）格式用 LZ4 压缩并保存到 UNIX TAR 文件中（图 6）。TAR 文件可以使用其他压缩方法进一步压缩（如需要），然后传输给负责分析的相关方。例如，标准 gzip 压缩可将文件大小减少 40%，但也使性能降低 80%。

IV. 实现

GraphBLAS 管道的有效实现要求匿名化、创建和保存结果文件的速度能够跟上典型高带宽链路的数据速率。为了测量这一性能，在两台 HP Proliant DL360 G9 服务器的 PCIe 插槽中安装了两个 Accolade Technology ANIC-200Kq 双端口 100 吉比特流量分类和分流适配器（图 7）。这两台服务器通过 Mellanox 40 吉比特网络连接相连。ANIC-200Kq 卡具有广泛的分析技术能力，本实验仅使用了它们的数据传输、分流和缓冲功能。

C 实现的 GraphBLAS 超稀疏流量矩阵管道如图 5 和 6所示，在接收服务器上的双 Intel Xeon E5-2683 处理器上运行。接收服务器上的 Accolade 卡在环形缓冲区中收集数据包，这些缓冲区的数量可以在初始化时设置。使用 C 线程 [70]，为每个 Accolade 硬件环分配一个 Accolade 工作线程。在每个 Accolade 工作线程内，会收集一组 2^{23}IPv4 数据包，并生成一个 GraphBLAS 工作线程来处理这些数据包的子块，每个子块包含 2^{17} 个数据包。每个子块使用 cryptoPAN 生成的匿名化向量进行匿名化处理，然后将得到的匿名源和目的地用于构建 GraphBLAS 矩阵。矩阵被序列化并追加到 TAR 缓冲区中，在所有 64 个子块处理完毕后保存至文件。代码更详细的概述如下：

主线程
- 设置 Accolade 硬件环的数量
- 加载 2^{32} 入站 IPv4 匿名化表
- 每个 Accolade 硬件环
  - 启动 荣誉称号工作者 线程

荣誉工作者
- 创建与 Accolade 设备的 libpcap 句柄
- 分配 64MB 缓冲区用于数据包处理
- 每个数据包
  - 从 Accolade 设备缓冲区中检索数据包
  - 将源和目的 IP 地址添加到缓冲区中
  - 如果缓冲区有 2^{23} 个数据包
    - 启动 图布拉斯工作者 线程并指向数据包缓冲区
    - 分配新的 64MB 缓冲区用于数据包处理

图布拉斯工作进程
- 初始化图布拉斯
- 每个由 2^{17} 个数据包组成的子块
  - 创建新的 GraphBLAS 矩阵
  - 创建行、列和值向量
  - 每个子块中的数据包
    - 查找匿名化表
    - 插入到行、列和值向量中
  - 从行、列和值向量构建 GraphBLAS 矩阵；对重复条目求和

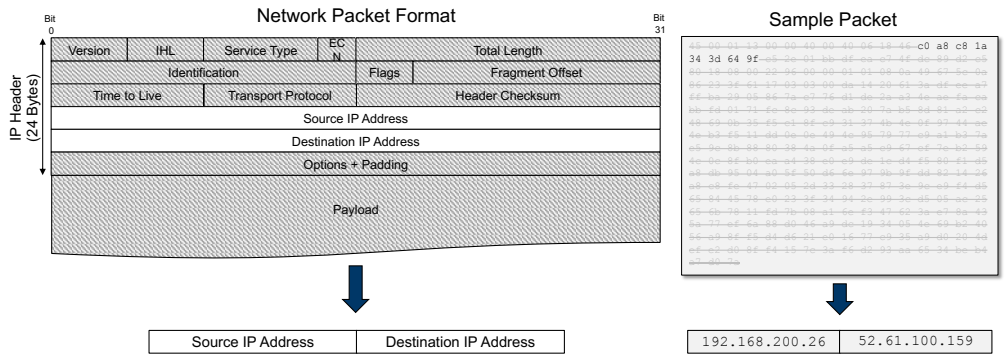

图 5. **网络数据包描述**。一个网络数据包由头部和负载组成。为了避免降采样并尽量减少隐私问题，只选择源地址和目标地址。

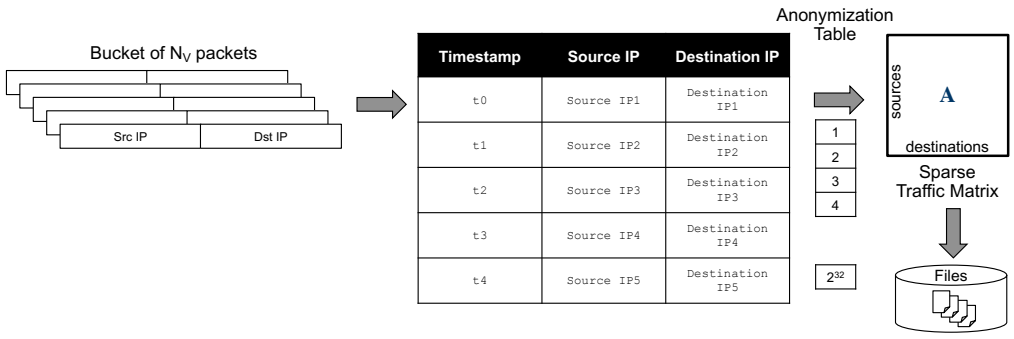

图 6. **匿名超稀疏矩阵流水线**。连续的 $N_V = 2^{17}$ 数据包序列从数据包头中提取，进行匿名处理，形成一个 GraphBLAS 超稀疏矩阵，然后被序列化，并以每 64 个 GraphBLAS 矩阵为一组保存到一个 UNIX TAR 文件中。

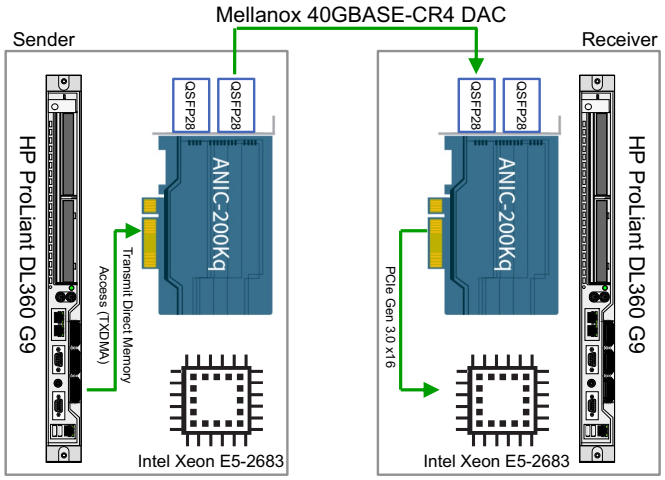

图 7. **实验设置**。测试系统由两个计算节点组成，每个节点通过网络连接一个 Accolade 卡。发送节点将 CAIDA Telescope 数据包读入内存，然后 Accolode 卡直接从内存通过网络将数据发送到接收节点。接收端的 Accolade 卡从网络上获取数据，将数据放入硬件环形缓冲区，并使数据可供接收处理器处理。

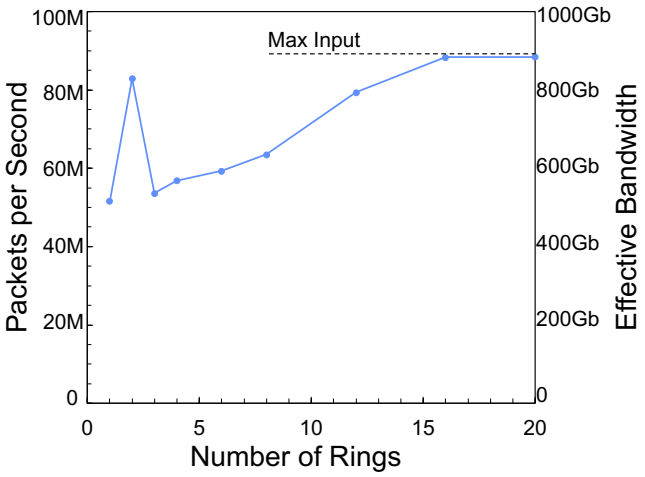

图 8. **性能结果**。每秒处理的包数与使用的硬件环数量的关系。可以通过使用每个包代表 10,000 位的数据包大小将数据包性能转换为估算的等效带宽（见右侧纵轴）

  - 序列化并压缩 GraphBLAS 矩阵
  - 加到 TAR 缓冲区
- 将 TAR 缓冲区写入文件

V. 结果

使用图 7中所示的实验设置，进行了若干性能实验。在这些实验中，发送服务器会将 100×2^{23} 个 CAIDA

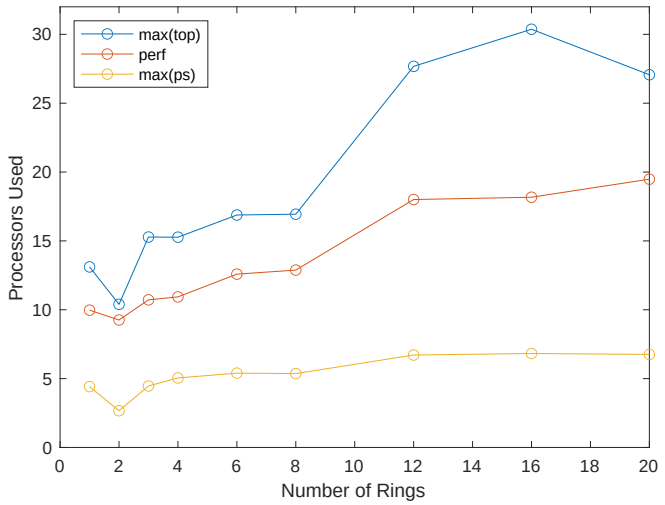

图 9. **使用的处理器**。使用的处理器数量与由 Linux 命令**顶部**的最大负载、Linux 命令**性能**的平均负载以及 Linux 命令**记法**的最大负载所确定的硬件环的数量。

Telescope 数据包加载到发送 Accolade 卡上，并以定义的速度通过网络将数据包发送至接收服务器，在那里接收 Accolade 卡会将数据包加载到其硬件环中。同样地，在接收服务器上，执行了前一节描述的 GraphBLAS 超稀疏流量矩阵流水线。cryptoPAN 匿名化表离线创建、存储，并在启动时加载。这样做的原因是单核 cryptoPAN 性能大约为每秒 700,000 个 IP 地址。包含 2^{32} 条目的 cryptoPAN 匿名化查找表极大地加快了性能。生成该表可以并行运行，可以在几秒钟内完成（如果需要）。发送数据包的速度被调整到不丢失数据包的最大速度，这表明 GraphBLAS 超稀疏流量矩阵流水线能够跟上流入的流量。硬件环的数量有所变化，从而增加了使用的线程/核心数量。

CAIDA 望远镜数据几乎是最糟糕的情况，因为几乎所有数据包都没有负载，并且使用相同源和目标连接的数据包非常少，因此生成的超稀疏流量矩阵中大于 1 的条目很少。CAIDA 望远镜数据中负载较少的一个优势是，它允许模拟代表当前实验设置能力之外的更高带宽网络的数据流。

图 8显示了每秒数据包性能与所使用硬件环数量的关系。由于需要进行底层索引排序，GraphBLAS 矩阵构建对缓存非常敏感。单个环使用少量处理线程/核心的性能超过 50M 数据包每秒。使用两个环时性能显著增加，而使用三个环时则因为缓存效应导致下降。使用 16 个环通过提高并行性弥补了缓存效应，并使性能达到发送服务器向接收方发送的最大数据包数量（约 88M 数据包每秒）。可以通过代表性的每个数据包 10,000 比特的数据包大小将数据包性能转换为等效带宽的估计值（参见图 8右侧垂直轴）。性能测量结果表明，一台标准服务器能够以高于典型 400 吉比特网络链路相应的速率构建匿名稀疏流量矩阵。

图 9显示了通过 Linux 命令**顶部**的最大负载、Linux 命令**性能**的平均负载以及 Linux 命令**示例输入**的最大负载计算出的处理器使用情况。**顶部**是在精确的轮询时刻测量的利用率，采样间隔为 1 秒。**性能**是在整个进程生命周期内对处理器负载的估算平均值。ps 报告了在整个进程生命周期中运行所花费的时间百分比，并且以 1 秒间隔进行采样。结合这些结果表明，使用 2 个环（见图 8）达到的峰值性能只需要少数几个核心。

VI. 结论与未来工作

对于许多操作领域（陆地、海洋、海底、空中、太空等），远程检测是防御的基石。在网络安全域中，需要分析来自各种观测站和前哨点的大量网络流量来进行远程检测。在网络边缘设备上构建匿名化的高稀疏度交通矩阵可以通过提供快速可分析格式中的显著数据压缩来成为关键推动力，同时保护隐私。构造和分析匿名化高稀疏度交通矩阵的操作非常适合 GraphBLAS 高性能库的理想使用场景。通过使用 Accolade Technologies 的网络边缘设备，在一个接近最坏情况下的流量场景中演示了 GraphBLAS 的性能，该场景采用 CAIDA 望远镜暗网数据包的连续流。通过对流量环形缓冲区的数量进行变化来探索性能，这些数量与使用的线程和处理器核心数成正比。实现了超过每秒 50,000,000 个数据包的速度来构建匿名化高稀疏度交通矩阵，这超过了典型的 400 吉比特网络链路。

这一表现表明，匿名的超稀疏流量矩阵可以在边缘网络设备上使用最少的计算资源轻松计算，并且可以成为此类设备的一种可行的数据产品。这项工作提出了未来可能追求的各种方向：（1）探索额外的网络卡；（2）为多个观测站和前哨开发适当的关键管理架构；（3）分析匿名流量矩阵中的时空模式以识别敌对活动；（4）不同观测站和前哨数据之间的交叉相关性分析；（5）开发用于背景流量分类的人工智能算法；（6）创建交通的基础模型。

致谢

作者希望感谢以下人士的贡献和支持：Bob Bond、Stephen Buckley、Ronisha Carter、Cary Conrad、Alan Edelman、Tucker Hamilton、Jeff Gottschalk、Nathan Frey、Chris Hill、Mike Kanaan、Tim Kraska、Andrew Morris、Charles Leiserson、Dave Martinez、Mimi McClure、Joseph McDonald、Sandy Pentland、Christian Prothmann、John Radovan、Steve Rejto、Daniela Rus、Matthew Weiss、Marc Zissman。

参考文献

[1] "*Cisco Visual Networking Index: Forecast and Trends.*" https://newsroom.cisco.com/press-release-content?articleId=1955935.

[2] "*Cisco Visual Networking Index: Forecast and Trends, 2018 – 2023.*" https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html.

[3] W. P. Delaney, *Perspectives on Defense Systems Analysis.* MIT Press, 2015.

[4] S. Topouzi, A. Sarris, Y. Pikoulas, S. Mertikas, X. Frantzis, and A. Giourou, "Ancient mantineia's defence network reconsidered through a gis approach," *BAR INTERNATIONAL SERIES*, vol. 1016, pp. 559–566, 2002.

[5] Y. Shu and Y. He, "Research on the historical and cultural value of and protection strategy for rammed earth watchtower houses in chongqing, china," *Built Heritage*, vol. 5, no. 1, pp. 1–16, 2021.

[6] R. Cacciotti, "The 'guardian of the pontifical state': structural assessment of a damaged coastal watchtower in south lazio," Master's thesis, Universidade do Minho, 2010.

[7] R. A. Watson-Watt, *Three Steps to Victory: A Personal Account by Radar's Greatest Pioneer.* London: Odhams Press, 1957.

[8] W. P. Delaney, "Air defense of the united states: Strategic missions and modern technology," *International Security*, vol. 15, no. 1, pp. 181–211, 1990.

[9] J. Geul, E. Mooij, and R. Noomen, "Modelling and assessment of the current and future space surveillance network," *7th ECSD*, 2017.

[10] K. W. O' Haver, C. K. Barker, G. D. Dockery, and J. D. Huffaker, "Radar development for air and missile defense," *Johns Hopkins APL Tech. Digest*, vol. 34, no. 2, pp. 140–153, 2018.

[11] "*CAIDA Anonymized Internet Traces Dataset (April 2008 - January 2019).*" https://www.caida.org/catalog/datasets/passive_dataset/.

[12] "*UCSD Network Telescope.*" https://www.caida.org/projects/network_telescope/.

[13] "*Global Cyber Alliance.*" https://www.globalcyberalliance.org/.

[14] "*Greynoise.*" https://greynoise.io/.

[15] "*MAWI Working Group Traffic Archive.*" http://mawi.wide.ad.jp/mawi/.

[16] "*Shadowserver Foundation.*" https://www.shadowserver.org/.

[17] J. Kepner, C. Meiners, C. Byun, S. McGuire, T. Davis, W. Arcand, J. Bernays, D. Bestor, W. Bergeron, V. Gadepally, R. Harnasch, M. Hubbell, M. Houle, M. Jones, A. Kirby, A. Klein, L. Milechin, J. Mullen, A. Prout, A. Reuther, A. Rosa, S. Samsi, D. Stetson, A. Tse, C. Yee, and P. Michaleas, "Multi-temporal analysis and scaling relations of 100,000,000,000 network packets," in *2020 IEEE High Performance Extreme Computing Conference (HPEC)*, pp. 1–6, 2020.

[18] K. Claffy, "Measuring the internet," *IEEE Internet Computing*, vol. 4, no. 1, pp. 73–75, 2000.

[19] B. Li, J. Springer, G. Bebis, and M. H. Gunes, "A survey of network flow applications," *Journal of Network and Computer Applications*, vol. 36, no. 2, pp. 567–581, 2013.

[20] M. Rabinovich and M. Allman, "Measuring the internet," *IEEE Internet Computing*, vol. 20, no. 4, pp. 6–8, 2016.

[21] k. claffy and D. Clark, "Workshop on internet economics (wie 2019) report," *SIGCOMM Comput. Commun. Rev.*, vol. 50, p. 53 – 59, May 2020.

[22] J. Kepner, J. Bernays, S. Buckley, K. Cho, C. Conrad, L. Daigle, K. Erhardt, V. Gadepally, B. Greene, M. Jones, R. Knake, B. Maggs, P. Michaleas, C. Meiners, A. Morris, A. Pentland, S. Pisharody, S. Powazek, A. Prout, P. Reiner, K. Suzuki, K. Takhashi, T. Tauber, L. Walker, and D. Stetson, "Zero botnets: An observe-pursue-counter approach." Belfer Center Reports, 6 2021.

[23] S. Weed, "Beyond zero trust: Reclaiming blue cyberspace," Master's thesis, United States Army War College, 2022.

[24] J. Kepner and J. Gilbert, *Graph algorithms in the language of linear algebra.* SIAM, 2011.

[25] J. Kepner, D. Bader, A. Buluç, J. Gilbert, T. Mattson, and H. Meyerhenke, "Graphs, matrices, and the graphblas: Seven good reasons," *Procedia Computer Science*, vol. 51, pp. 2453–2462, 2015.

[26] J. Kepner, P. Aaltonen, D. Bader, A. Buluç, F. Franchetti, J. Gilbert, D. Hutchison, M. Kumar, A. Lumsdaine, H. Meyerhenke, S. McMillan, C. Yang, J. D. Owens, M. Zalewski, T. Mattson, and J. Moreira, "Mathematical foundations of the graphblas," in *2016 IEEE High Performance Extreme Computing Conference (HPEC)*, pp. 1–9, 2016.

[27] A. Buluç, T. Mattson, S. McMillan, J. Moreira, and C. Yang, "Design of the graphblas api for c," in *2017 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*, pp. 643–652, 2017.

[28] J. Kepner, M. Kumar, J. Moreira, P. Pattnaik, M. Serrano, and H. Tufo, "Enabling massive deep neural networks with the graphblas," in *2017 IEEE High Performance Extreme Computing Conference (HPEC)*, pp. 1–10, IEEE, 2017.

[29] C. Yang, A. Buluç, and J. D. Owens, "Implementing push-pull efficiently in graphblas," in *Proceedings of the 47th International Conference on Parallel Processing*, pp. 1–11, 2018.

[30] T. A. Davis, "Algorithm 1000: Suitesparse:graphblas: Graph algorithms in the language of sparse linear algebra," *ACM Trans. Math. Softw.*, vol. 45, Dec. 2019.

[31] J. Kepner and H. Jananthan, *Mathematics of big data: Spreadsheets, databases, matrices, and graphs.* MIT Press, 2018.

[32] T. A. Davis, "Algorithm 1000: Suitesparse: Graphblas: Graph algorithms in the language of sparse linear algebra," *ACM Transactions on Mathematical Software (TOMS)*, vol. 45, no. 4, pp. 1–25, 2019.

[33] T. Mattson, T. A. Davis, M. Kumar, A. Buluc, S. McMillan, J. Moreira, and C. Yang, "Lagraph: A community effort to collect graph algorithms built on top of the graphblas," in *2019 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*, pp. 276–284, IEEE, 2019.

[34] P. Cailliau, T. Davis, V. Gadepally, J. Kepner, R. Lipman, J. Lovitz, and K. Ouaknine, "Redisgraph graphblas enabled graph database," in *2019 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*, pp. 285–286, IEEE, 2019.

[35] T. A. Davis, M. Aznaveh, and S. Kolodziej, "Write quick, run fast: Sparse deep neural network in 20 minutes of development time via suitesparse: Graphblas," in *2019 IEEE High Performance Extreme Computing Conference (HPEC)*, pp. 1–6, IEEE, 2019.

[36] M. Aznaveh, J. Chen, T. A. Davis, B. Hegyi, S. P. Kolodziej, T. G. Mattson, and G. Szárnyas, "Parallel graphblas with openmp," in *2020 Proceedings of the SIAM Workshop on Combinatorial Scientific Computing*, pp. 138–148, SIAM, 2020.

[37] B. Brock, A. Buluç, T. G. Mattson, S. McMillan, and J. E. Moreira, "Introduction to graphblas 2.0," in *2021 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*, pp. 253–262, IEEE, 2021.

[38] M. Pelletier, W. Kimmerer, T. A. Davis, and T. G. Mattson, "The graphblas in julia and python: the pagerank and triangle centralities," in *2021 IEEE High Performance Extreme Computing Conference (HPEC)*, pp. 1–7, 2021.

[39] J. Kepner, T. Davis, C. Byun, W. Arcand, D. Bestor, W. Bergeron, V. Gadepally, M. Hubbell, M. Houle, M. Jones, A. Klein, P. Michaleas, L. Milechin, J. Mullen, A. Prout, A. Rosa, S. Samsi, C. Yee, and A. Reuther, "75,000,000,000 streaming inserts/second using hierarchical hypersparse graphblas matrices," in *2020 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*, pp. 207–210, 2020.

[40] J. Kepner, M. Jones, D. Andersen, A. Buluç, C. Byun, K. Claffy, T. Davis, W. Arcand, J. Bernays, D. Bestor, W. Bergeron, V. Gadepally, M. Houle, M. Hubbell, A. Klein, C. Meiners, L. Milechin, J. Mullen, S. Pisharody, A. Prout, A. Reuther, A. Rosa, S. Samsi, D. Stetson, A. Tse, C. Yee, and P. Michaleas, "Spatial temporal analysis of 40,000,000,000,000 internet darkspace packets," in *2021 IEEE High Performance Extreme Computing Conference (HPEC)*, pp. 1–8, 2021.

[41] J. Kepner, K. Cho, K. Claffy, V. Gadepally, P. Michaleas, and L. Milechin, "Hypersparse neural network analysis of large-scale internet traffic," in *2019 IEEE High Performance Extreme Computing Conference (HPEC)*, pp. 1–11, 2019.

[42] J. Kepner, K. Cho, K. Claffy, V. Gadepally, S. McGuire, L. Milechin, W. Arcand, D. Bestor, W. Bergeron, C. Byun, M. Hubbell, M. Houle, M. Jones, A. Prout, A. Reuther, A. Rosa, S. Samsi, C. Yee, and P. Michaleas, "New phenomena in large-scale internet traffic," in *Massive Graph Analytics* (D. Bader, ed.), pp. 1–53, Chapman and Hall/CRC, 2022.

[43] P. Devlin, J. Kepner, A. Luo, and E. Meger, "Hybrid power-law models of network traffic," *arXiv preprint arXiv:2103.15928*, 2021.

[44] A. Tumeo, O. Villa, and D. Sciuto, "Efficient pattern matching on gpus for intrusion detection systems," in *Proceedings of the 7th ACM International Conference on Computing Frontiers*, CF '10, (New York, NY, USA), p. 87 – 88, Association for Computing Machinery, 2010.

[45] M. Kumar, W. P. Horn, J. Kepner, J. E. Moreira, and P. Pattnaik, "Ibm power9 and cognitive computing," *IBM Journal of Research and Development*, vol. 62, no. 4/5, pp. 10–1, 2018.

[46] J. Ezick, T. Henretty, M. Baskaran, R. Lethin, J. Feo, T.-C. Tuan, C. Coley, L. Leonard, R. Agrawal, B. Parsons, and W. Glodek, "Combining tensor decompositions and graph analytics to provide cyber situational awareness at hpc scale," in *2019 IEEE High Performance Extreme Computing Conference (HPEC)*, pp. 1–7, 2019.

[47] P. Gera, H. Kim, P. Sao, H. Kim, and D. Bader, "Traversing large graphs on gpus with unified memory," *Proceedings of the VLDB Endowment*, vol. 13, no. 7, pp. 1119–1133, 2020.

[48] A. Azad, M. M. Aznaveh, S. Beamer, M. Blanco, J. Chen, L. D'Alessandro, R. Dathathri, T. Davis, K. Deweese, J. Firoz, H. A. Gabb, G. Gill, B. Hegyi, S. Kolodziej, T. M. Low, A. Lumsdaine, T. Manlaibaatar, T. G. Mattson, S. McMillan, R. Peri, K. Pingali, U. Sridhar, G. Szarnyas, Y. Zhang, and Y. Zhang, "Evaluation of graph analytics frameworks using the gap benchmark suite," in *2020 IEEE International Symposium on Workload Characterization (IISWC)*, pp. 216–227, 2020.

[49] Z. Du, O. A. Rodriguez, J. Patchett, and D. A. Bader, "Interactive graph stream analytics in arkouda," *Algorithms*, vol. 14, no. 8, p. 221, 2021.

[50] S. Acer, A. Azad, E. G. Boman, A. Buluç, K. D. Devine, S. Ferdous, N. Gawande, S. Ghosh, M. Halappanavar, A. Kalyanaraman, A. Khan, M. Minutoli, A. Pothen, S. Rajamanickam, O. Selvitopi, N. R. Tallent, and A. Tumeo, "Exagraph: Graph and combinatorial methods for enabling exascale applications," *The International Journal of High Performance Computing Applications*, vol. 35, no. 6, pp. 553–571, 2021.

[51] M. P. Blanco, S. McMillan, and T. M. Low, "Delayed asynchronous iterative graph algorithms," in *2021 IEEE High Performance Extreme Computing Conference (HPEC)*, pp. 1–7, IEEE, 2021.

[52] N. K. Ahmed, N. Duffield, and R. A. Rossi, "Online sampling of temporal networks," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 15, no. 4, pp. 1–27, 2021.

[53] A. Azad, O. Selvitopi, M. T. Hussain, J. R. Gilbert, and A. Buluç, "Combinatorial blas 2.0: Scaling combinatorial algorithms on distributed-memory systems," *IEEE Transactions on Parallel and Distributed Systems*, vol. 33, no. 4, pp. 989–1001, 2021.

[54] D. Koutra, "The power of summarization in graph mining and learning: smaller data, faster methods, more interpretability," *Proceedings of the VLDB Endowment*, vol. 14, no. 13, pp. 3416–3416, 2021.

[55] R. Hofstede, P. Čeleda, B. Trammell, I. Drago, R. Sadre, A. Sperotto, and A. Pras, "Flow monitoring explained: From packet capture to data analysis with netflow and ipfix," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 4, pp. 2037–2064, 2014.

[56] R. Sommer, "Bro: An open source network intrusion detection system," *Security, E-learning, E-Services, 17. DFN-Arbeitstagung über Kommunikationsnetze*, 2003.

[57] P. Lucente, "pmacct: steps forward interface counters," *Tech. Rep.*, 2008.

[58] J. Fan, J. Xu, M. H. Ammar, and S. B. Moon, "Prefix-preserving ip address anonymization: measurement-based security evaluation and a new cryptography-based scheme," *Computer Networks*, vol. 46, no. 2, pp. 253–272, 2004.

[59] J. Karvanen and A. Cichocki, "Measuring sparseness of noisy signals," in *4th International Symposium on Independent Component Analysis and Blind Signal Separation*, pp. 125–130, 2003.

[60] A. Soule, A. Nucci, R. Cruz, E. Leonardi, and N. Taft, "How to identify and estimate the largest traffic matrix elements in a dynamic environment," in *ACM SIGMETRICS Performance Evaluation Review*, vol. 32, pp. 73–84, ACM, 2004.

[61] Y. Zhang, M. Roughan, C. Lund, and D. L. Donoho, "Estimating point-to-point and point-to-multipoint traffic matrices: an information-theoretic approach," *IEEE/ACM Transactions on Networking (TON)*, vol. 13, no. 5, pp. 947–960, 2005.

[62] P. J. Mucha, T. Richardson, K. Macon, M. A. Porter, and J.-P. Onnela, "Community structure in time-dependent, multiscale, and multiplex networks," *science*, vol. 328, no. 5980, pp. 876–878, 2010.

[63] P. Tune, M. Roughan, H. Haddadi, and O. Bonaventure, "Internet traffic matrices: A primer," *Recent Advances in Networking*, vol. 1, pp. 1–56, 2013.

[64] J. Kepner, V. Gadepally, L. Milechin, S. Samsi, W. Arcand, D. Bestor, W. Bergeron, C. Byun, M. Hubbell, M. Houle, M. Jones, A. Klein, P. Michaleas, J. Mullen, A. Prout, A. Rosa, C. Yee, and A. Reuther, "Streaming 1.9 billion hypersparse network updates per second with d4m," in *2019 IEEE High Performance Extreme Computing Conference (HPEC)*, pp. 1–6, 2019.

[65] J. Nair, A. Wierman, and B. Zwart, "The fundamentals of heavy tails: Properties, emergence, and estimation," *Preprint, California Institute of Technology*, 2020.

[66] A. J. Elmore, J. Duggan, M. Stonebraker, M. Balazinska, U. Cetintemel, V. Gadepally, J. Heer, B. Howe, J. Kepner, T. Kraska, *et al.*, "A demonstration of the bigdawg polystore system," *Proceedings of the VLDB Endowment*, vol. 8, no. 12, p. 1908, 2015.

[67] T. Kraska, A. Beutel, E. H. Chi, J. Dean, and N. Polyzotis, "The case for learned index structures," in *Proceedings of the 2018 International Conference on Management of Data*, SIGMOD 18, (New York, NY, USA), pp. 489–504, Association for Computing Machinery, 2018.

[68] E. H. Do and V. N. Gadepally, "Classifying anomalies for network security," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2907–2911, 2020.

[69] S. Pisharody, J. Bernays, V. Gadepally, M. Jones, J. Kepner, C. Meiners, P. Michaleas, A. Tse, and D. Stetson, "Realizing forward defense in the cyber domain," in *2021 IEEE High Performance Extreme Computing Conference (HPEC)*, pp. 1–7, IEEE, 2021.

[70] B. Nichols, D. Buttlar, and J. P. Farrell, *Pthreads programming*. O'Reilly & Associates, Inc., 1996.