Running GraphBLAS on the FABRIC testbed

Vaneshi Ramdhony
Department of Computer Science
Illinois Institute of Technology
Chicago, USA

Hyunsuk Bang
Department of Computer Science
Illinois Institute of Technology
Chicago, USA

Nik Sultana
Department of Computer Science
Illinois Institute of Technology
Chicago, USA

***Abstract*—This paper describes the first FABRIC deployment of GraphBLAS—a powerful linear algebra-based framework for network traffic analysis. FABRIC is an international network testbed that is used for teaching and research in networking.**

This work has two goals to provide a foundation for further research on network security: (1) deploying an advanced network monitoring technique on FABRIC, and (2) making this deployment shareable with other FABRIC users.

These goals are achieved by creating FABRIC experiments that build and deploy GraphBLAS to process network traffic on FABRIC. These experiments are described as executable Jupyter notebooks that are shareable with other researchers.

These experiments also include the replaying of network workloads (from pcap files), which we use to test and evaluate the deployment.

***Index Terms*—GraphBLAS, Network Monitoring, FABRIC testbed**

I. Introduction

The FABRIC testbed [1] provides a platform for network research that includes programmable network hardware, and that can be flexibly configured to suit its researchers needs.

This paper describes a project that uses FABRIC to reach two goals: (1) deploying an advanced network monitoring technique on FABRIC, and (2) making this deployment shareable with other FABRIC users.

Achieving these goals will help provide a foundation for further research on securing cyberinfrastructure. This research benefits from FABRIC in two ways:

1) FABRIC provides an experimentation environment in which to carry out empirical research. In this paper, we describe the deployment of advanced network monitoring techniques.
2) FABRIC is *itself* an interest example of a powerful computing instrument that needs to be secure. By monitoring FABRIC's network we can detect whether any communication patterns need further investigation. Examples of such patterns include those of worms, botnets, and their command-and-control traffic. Of course, on a system like FABRIC, that traffic might be from an experiment that involves simulating an attack; but in any case we would want to ensure that such experiments remain contained.

GraphBLAS [2] is a graph analysis tool that can analyze communication patterns in a network. It is able to scale well and it has been used on large, busy computer networks. GraphBLAS is a tool that could help advance security research on FABRIC, and of FABRIC-like, shared, integrated research infrastructure, as described above. Further background is provided in Section II, and related work is described in Section III.

In this paper, we describe the deployment of GraphBLAS on FABRIC. This deployment consists of an examplar Jupyter notebook that sets up GraphBLAS on a VM instance running on FABRIC. That instance is then provided with network traffic to analyze. Section IV describes our methodology, and Section V describes our evaluation. This Jupyter notebook is made available for other researchers to use and extend.[1]

In this deployment, GraphBLAS running on FABRIC analyzes pcap files, but in the future this can be extended to analyze FABRIC network traffic—related to other experiments that are running on FABRIC.

Section VI describes a path to follow-up research by building on the Patchwork [3] dataplane observability system. Patchwork can programmatically filter traffic across FABRIC's dataplane, and direct that traffic to GraphBLAS for analysis. This analysis could serve both the users of testbeds and its operators, to monitor its network activity at a configurable granularity using state-of-the-art techniques.

II. Background

This section provides a brief background on FABRIC and GraphBLAS to support the sections that follow.

FABRIC [1] is a federation of 33 sites that are distributed across north America, Asia, and Europe. Each site provides hardware resources—including servers and various types of network equipment—that can be virtualized to simultaneously serve different researchers' experiment needs. Experiments can span several sites, and can be flexibly managed using FABRIC's portal or by using FABRIC's Python-based API.

GraphBLAS [2] implements linear algebra techniques for network traffic analysis. Network traffic is used to build a matrix in which rows denote source addresses, columns denote destination addresses, and the value at each cell denotes the number of packets exchanged between a given source and destination.

III. Related Work

Research testbeds usually have monitoring infrastructure—such as CoMon [4] on PlanetLab or MFlib [5] on FABRIC—but their scope and role differ from this GraphBLAS deployment. This GraphBLAS deployment is scoped in a single VM,

[1] https://gitlab.com/d-r-r/release/gbf

while testbed monitoring infrastructure tends to be distributed across the infrastructure. The role of this deployment is to clarify the process of setting up GraphBLAS within FABRIC. This serves as a foundation for future work—described in Section VI—in which GraphBLAS (and research extensions of GraphBLAS) are used to monitor experiment-specific and testbed-wide network traffic.

IV. Methodology and Testing

The first steps of this work involved compiling GraphBLAS in a fresh VM on FABRIC. These steps involved finding out implicit software dependencies and automating their installation by scripting them in a Jupyter notebook. GraphBLAS was obtained from GitHub—the precise repo and revision are documented in our code repo.

Once the dependencies were clarified, the Jupyter notebook was structured into three stages. First, FABRIC resources are allocated. These include host (VM) and network resources on FABRIC. Second, GraphBLAS is compiled in a VM that receives traffic for analysis. Finally, network traffic is captured and made available to GraphBLAS. In the current deployment, this involves providing GraphBLAS with packet captures ("pcap" files), and using the `pcap2grb` tool to produce a GraphBLAS traffic matrix.

As a more complex deployment example, we prepared an extended version of the notebook that implements a small 5-node topology on FABRIC. On this virtual network, several instances of iperf are used to send traffic to other nodes, and the communication between nodes was stored as a matrix by GraphBLAS.

V. Evaluation

To test the setup, we wrote a script that applied the following operations to the set of pcap files described below: (1) running the `pcap2grb` tool on each pcap file produces tar files, each containing 64 `.grb` files. These files encode the serialized GraphBLAS matrix. (2) extracting the tar files and running `gbdump` on each of them. `gbdump` generates ASCII output showing pairs of communicating IPv4 addresses, which we use to validate the setup.

The pcap files that were used for testing were obtained online. The source of each pcap file is documented in our code repository.

Traffic capture	Size (# packets)
fuzz-2006-08-27-677	637
fuzz-2006-06-26-2594	691
gmail.pcapng	793
fuzz-2010-06-29-8087	86401
fuzz-2009-06-27-1565	282703
bigFlows	791615

VI. Future Work

The initial deployment of GraphBLAS on FABRIC supports future work on research and teaching. Future research work includes integrating with Patchwork [3]. Patchwork is a programmable network profiler for FABRIC. It gathers traffic samples that GraphBLAS could then analyze. In future work, this integration could form part of a long-running traffic analysis experiment on FABRIC. GraphBLAS could also be applied to traffic generated by load generation tools [6] to characterize different communication patterns—such as command-and-control behavior for malware—and assessing their reproducibility for research and teaching [7]. Other future work include packaging this notebook for easier uptake by other researchers, using systems like Trovi [8].

VII. Conclusion

This paper described an initial adaptation of GraphBLAS to run on the FABRIC testbed. The focus of this work involved automating this setup after discovering implicit software dependencies. This work produced a Jupyter notebook that encapsulates our deployment process, and which we hope that others can find useful. Future work includes more extensive testing and scaling, and integration with other testbed-related services and research.

Acknowledgment

We thank Michael Jones and Jeremy Kepner for feedback and for help with getting started using GraphBLAS. We thank Komal Thareja and Nishanth Shyamkumar for help with FABRIC-related questions. This work was supported by the National Science Foundation (NSF) under award 2346499. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of funders.

References

[1] I. Baldin, A. Nikolich, J. Griffioen, S. Inder, and P. Shenoy, "FABRIC: A national-scale programmable experimental network infrastructure," *ACM SIGCOMM Computer Communication Review*, vol. 46, no. 4, pp. 59–66, 2016.

[2] M. Jones, J. Kepner, D. Andersen, A. Buluç, C. Byun, K. Claffy, T. Davis, W. Arcand, J. Bernays, D. Bestor, W. Bergeron, V. Gadepally, M. Houle, M. Hubbell, H. Jananthan, A. Klein, C. Meiners, L. Milechin, J. Mullen, S. Pisharody, A. Prout, A. Reuther, A. Rosa, S. Samsi, J. Sreekanth, D. Stetson, C. Yee, and P. Michaleas, "GraphBLAS on the Edge: Anonymized High Performance Streaming of Network Traffic," in *2022 IEEE High Performance Extreme Computing Conference (HPEC)*, 2022, pp. 1–8.

[3] N. Shyamkumar, S. Cummings, H. Bang, and N. Sultana, "Towards Testbed-Wide Traffic Profiling for FABRIC," in press: 11th International Workshop on Computer and Networking Experimental Research using Testbeds (CNERT 2024).

[4] K. Park and V. S. Pai, "CoMon: a mostly-scalable monitoring system for PlanetLab," *ACM SIGOPS Oper. Syst. Rev.*, vol. 40, no. 1, pp. 65–74, 2006. [Online]. Available: https://doi.org/10.1145/1113361.1113374

[5] "FABRIC Experiment Measurement APIs (MFlib)," https://learn.fabric-testbed.net/article-categories/mflib-api/.

[6] J. Brassil, "Network Traffic as a Federated Testbed Service," in *2022 IEEE Future Networks World Forum, FNWF 2022, Montreal, QC, Canada, October 10-14, 2022*. IEEE, 2022, pp. 450–455. [Online]. Available: https://doi.org/10.1109/FNWF55208.2022.00086

[7] F. Fund, "We Need More Reproducibility Content Across the Computer Science Curriculum," in *Proceedings of the 2023 ACM Conference on Reproducibility and Replicability*, ser. ACM REP '23. New York, NY, USA: Association for Computing Machinery, 2023, p. 97–101. [Online]. Available: https://doi.org/10.1145/3589806.3600033

[8] M. Powers, "Sharing Experiments with Trovi," https://www.chameleoncloud.org/blog/2022/06/01/sharing-experiments-with-trovi/, May 2022, accessed: 2024-06-10.