B-MEG: Bottlenecked-Microservices Extraction Using Graph Neural Networks

Gagan Somashekar, Anurag Dutt, Rohith Vaddavalli, Sai Bhargav Varanasi, Anshul Gandhi

PACE Lab, Stony Brook University, Stony Brook, New York, USA

{gsomashekar,adutt,rvaddavalli,sbvaranasi,anshul}@cs.stonybrook.edu

PACE
Stony Brook University

Introduction

- Microservices architecture is replacing monolithic and multi-tier architecture.
- Performance is crucial as these are usually customer-facing applications.

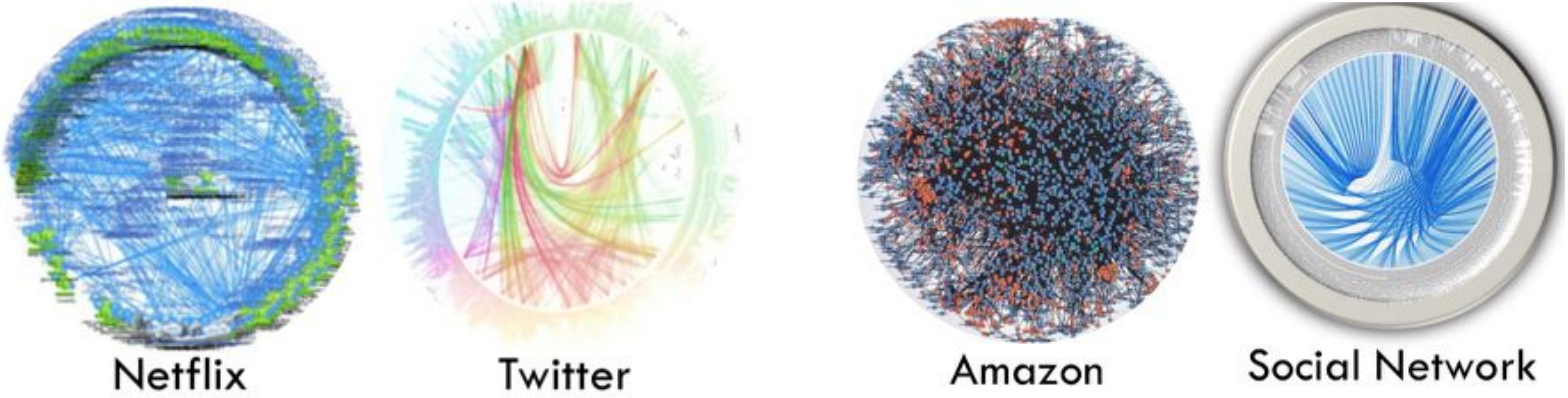


Image source: Gan et al., *ASPLOS'19*

Motivation

- Detecting and mitigating performance bottlenecks is paramount.
- With Microservices architecture, the problem of finding bottlenecks is exacerbated by the following:
  - Difficult to precisely pinpoint bottlenecks.
  - Scarcity of labeled data in production systems.
  - Dynamic nature.

How to precisely pinpoint bottlenecks using limited labeled data in a generalizable way?

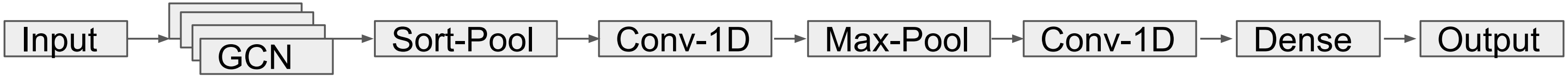
Graph Neural Networks (GNN)

Why GNN?

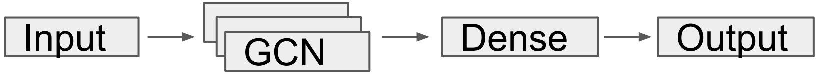
- Superior performance on graph data.
- Learns dependency among nodes.
- Handles multi-class imbalance problem well.
- Generalize well through transfer learning.

Objective and System Design

- Detecting anomalous traces → graph classification.
- Detecting bottlenecks in potential anomalous traces → node classification.
- Hierarchical classifier (B-MEG):
    - Graph classifier (Deep Graph Convolutional Neural Network [Chen 2020, ICML.]).



    - Node classifier.



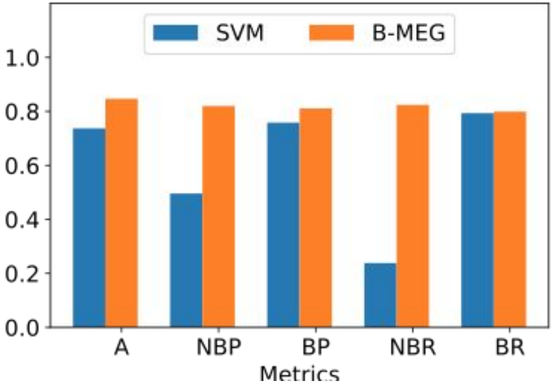
Dataset

- Original dataset (Qui et al., OSDI’20) - traces of the following benchmarking applications:
  - Social Networking (SN), Media Microservices (MM), and Hotel Reservation (HR) from DeathStarBench (Gan et al., ASPLOS’19) .
  - Train Ticket (TT) application (Zhou et al., ICSE 2018).
- Each trace consists of a single bottleneck induced via artificial interference.
- Randomly sampled imbalanced datasets A (30%), B (10%), and C (1%).

Evaluation and Preliminary Results

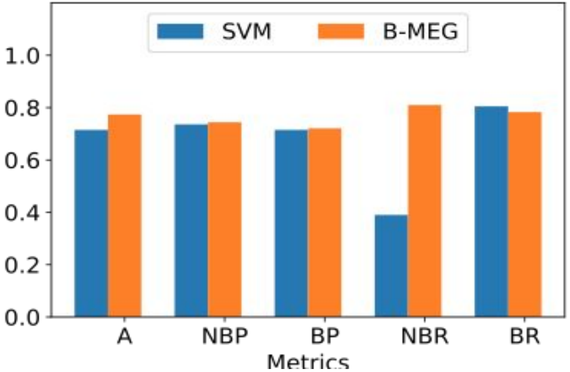
- B-MEG maintains good balance between precision and recall for complex call graphs (SN) even when the dataset is highly imbalanced.
- Simple call graphs (MM, HR, TT) aid SVM in learning thresholds.

Acc SVM: 0.74
B-MEG: 0.85



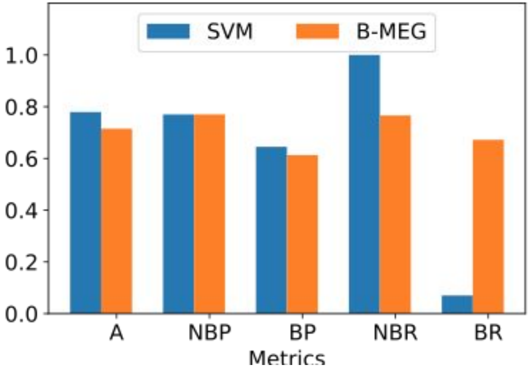
Dataset A (30%)

NBR SVM: 0.39
B-MEG: 0.81



Dataset B (10%)

BR SVM: 0.07
B-MEG: 0.67



Dataset C (1%)

Conclusion and Future Work

Conclusion:

- B-MEG shows superior performance in detecting bottlenecks on imbalanced datasets for large and complex call graphs compared to SVM.

Future Work:

- Generalization through transfer learning.
- Dataset with multiple bottlenecks.
- Analysis of the impact of dataset size on performance and training effort.
- Comparison against related work and Application Performance Management tools (Eg: AppDynamics).