Detecting Causal Structure on Cloud Application Microservices Using Granger Causality Models

Qing Wang[1], Laura Shwartz[1], Genady Ya. Grabarnik[2], Vijay Arya[3], Karthikeyan Shanmugam[4]

1. IBM Research, IBM T.J. Watson Research Center, Yorktown Heights, NY, US
2. Dept. Math & Computer Science, St. John's University, Queens, NY, US
3. IBM Research India, Bangalore, KA, IN
4. IBM Research AI, IBM T.J. Watson Research Center, Yorktown Heights, NY, US

Agenda

- Introduction
- Problem Statement & Challenges
- Related Work
- Problem Modeling & AI Methodologies
- Experiments
- Conclusion & Future Work

Introduction

- Since digital transformation have accelerated across the globe in nearly every industry, **AI for IT Operations (AIOps)** has become a critical capability for any enterprise that aims to use rapidly growing IT data to assist its IT operations in providing reliability for its applications.

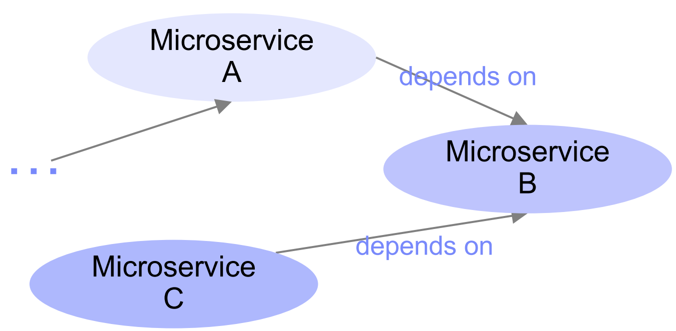
- What is **AIOps**: use IT data with AI models to resolve/predict IT incidents (i.e., problems) from applications.
  – e.g., IT data including logs, metrics, alerts, topologies and tickets.
  – e.g., IT incidents including various outages, anomalies, and unplanned downtime.
  – e.g., AI models including event correlation models, predictive models, anomaly detection models, etc.

- Why **AIOps:**
  – It can greatly help **Site Reliability Engineers (SREs)** 1) detect incidents early, 2) determine the root cause of incidents, and 3) recommend timely actions for solving them.
  – This, in turn, saves millions of dollars for enterprises and keep their customers satisfied. [1]

[1] https://www.ibm.com/cloud/blog/watson-aiops-bringing-ai-to-it-operations-management

Problem Statement & Challenges

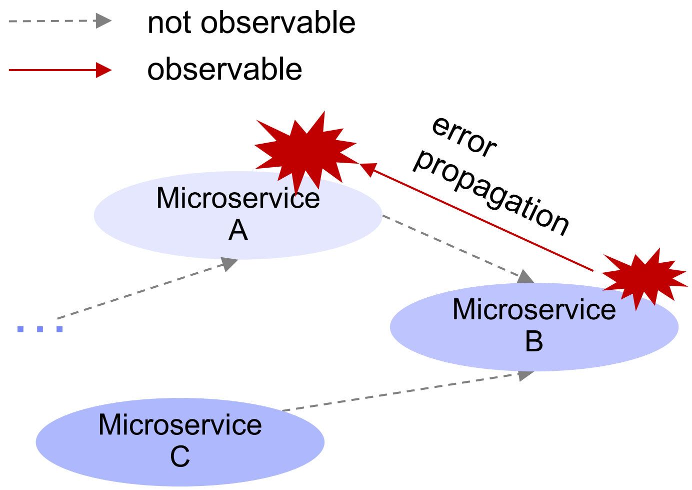
- In practice
  - The loosely-coupled microservices architecture has become popular.
  - Industrial microservice-based applications always have hundreds to thousands of microservices and complex **dependency relationships** among them. [2]
  - Many microservice-based applications suffer from limited observability, i.e., the complex topology of microservices is often **unknown** but fixed.
  - Localizing faults is extremely **challenging** in these microservice-based applications but **desirable** as it allows SREs to quickly find the faulty microservices.



[2] An Open-Source Benchmark Suite for Microservices and Their Hardware-Software Implications for Cloud & Edge Systems

Problem Statement & Challenges

- We propose a framework:
  - Learn the hidden causal graph among microservices utilizing fault injections to observe the **error propagations** from only logs collected by LogDNA.



Observed Logs

_time	_line	…	_level	_app	_container
t	http 500	…	error	B	B
		…	info		
(t+1)	http 500	…	error	A	A

Table 1. An example of logs data from 1) At time t, inject a fault to B; 2) After 5s, we observe an error from A.

Related Work

- AIOps (AI for IT Operations) has been extensively studied for many years, in order to help IT operations teams **determine**, **diagnosis** incidents (including root causa analysis), and finally **recommend** the next best actions to resolve the incidents. [3, 29, 33, 34, 36, 37]

- Mining temporal dependency structure among microservices is one of the critical tasks for incident diagnosis in AIOps.
  - [12] uses causal inference techniques to build a dependency graph for anomaly detection.
  - [19] models causal dependencies on metrics data to facilitate fault localization in the cloud system.

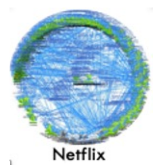
- It is still in its infancy using only log data to uncover the hidden causal relationships between microservices of a microservice-based application.

- Our work focus on the performance analysis of linear and nonlinear Granger causality models for detecting causal relationships.

System Overview

Log Modeling

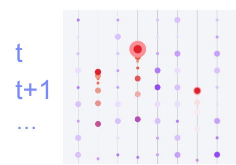
Fault Injection



Logs

logdna

Binning



Causal Modeling

1. Linear Granger Causality models
2. Nonlinear Granger Causality models

B ------>$_g$ A
B ------>$_g$ C
...

Problem Modeling & AI Methodologies

- **Log Modeling**
  - Log Data Collection: Inject a fault into one microservice of a subset of microservices of TrainTicket and collect the normal or erroneous logs from affected microservices by the faulty microservice.
  - Log Labeling:
    - Error patterns, e.g., "HTTP 500 Internal Server Error"
  - Binning logs as time series:
    - Use different time bin sizes (10ms, 100ms, 1s) and count the number of error logs from each microservice in each bin.

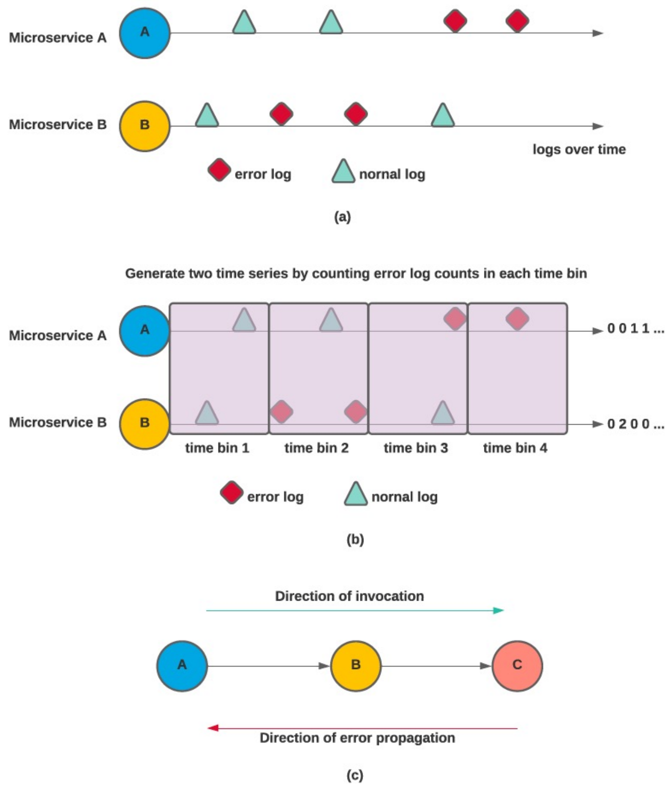


Fig. 2: (a): Log data after, (b) Modeling log data as multiple time series, and (c) A sample causal graph that we tend to infer which implies that errors in microservice A are caused by errors in microservice B are caused by errors in microservice C (i.e., root cause of errors is C).

Problem Modeling & AI Methodologies

- **Causal Modeling**
  - In practice, Granger causality is more straightforward and robust for learning causal relationships among multivariate time series.
  - Linear Granger Causality Modeling: Vector Auto-Regression (VAR) models.
    - BLR and BLasso

$$\mathbf{y}_{\bullet,t} = \sum_{l=1}^{L} (\mathbf{W}^l)^\intercal \mathbf{y}_{\bullet,t-l} + \epsilon,$$

  - Nonlinear Granger Causality Modeling: use the nonlinear autoregressive function $f(\cdot)$ for Granger causality problem in time series analysis.
    - cMLP and cLSTM

$$\mathbf{y}_{\bullet,t} = f(\mathbf{x}_t) + \epsilon,$$

TABLE I: Important Notations

Notation	Description
$\mathbf{Y}$	A set of time series.
K	The number of time series in $\mathbf{Y}$.
T	The length of time series.
L	The maximum time lag for VAR model.
s	The sparsity of the temporal dependency, denoted as the ratio of coefficients with zero value to K.
$\mathbf{y}_i$	The i^{th} time series.
$\mathbf{y}_{j,t}$	The value of j^{th} time series at time t.
$\mathbf{y}_{\bullet,t}$	A column vector containing the values of all time series at time t.
$\mathbf{x}_t$	A column vector built from all time series with time lag L at time t.
$\mathbf{W}^l$	The coefficient matrix for time lag l in VAR model.
$\mathbf{w}_j$	The coefficient vector used to predict j^{th} time series value in Bayesian Lasso model.
$\mathbf{w}_{j,t}$	The coefficient vector used to predict j^{th} time series value at time t in time-varying Bayesian Lasso model.
λ	The penalty parameters for $\mathbf{w}_j$.

Experiments

- **Data:**
  - – Synthetic data
    - Linear VAR data: the time series data are generated with linear VAR model.
    - Lorenz-96 data: the time series data are generated with nonlinear Lorenz-96 model [13].
    - Evaluation Metric: **AUROC** score. [16]
  - – Benchmark data
    - TrainTicket log data: collect log data by deleting the microservice ‘ts-basic-service’ from the system.
    - Evaluation Metrics: Precision, Recall, F1.

- **Models:**
  - – Linear Granger Causality models
    - BLR(q_0): q_0 is for the regularization term.
    - Blasso(λ): λ is for the regularization term.
    - MMPC(ParCorr): MMPC with partial correlation Conditional Independence testing
  - – Nonlinear Granger Causality models
    - cMLP: with a single multilayer perceptron layer.
    - cLSTM: with LSTM architecture.
    - MMPC(RCoT): MMPC with RCoT Conditional Independence testing

IBM

Results on Synthetic Data

TABLE II: AUROC comparisons of simulated VAR data with $s = 0.2$ on linear and neural Granger causality methods.

K	5	5	10	10
T	1000	10000	1000	10000
BLR(1.0)	1.0	1.0	1.0	1.0
BLasso(1.0)	1.0	1.0	1.0	1.0
cMLP	0.92	**0.95**	0.62	0.64
cLSTM	0.52	0.78	0.63	0.73

TABLE III: AUROC comparisons of Lorenz-96 data with $K = 5$ on linear and neural Granger causality methods.

F	10	10	40	40
T	500	1000	500	1000
BLR(1.0)	0.75	0.75	0.48	0.48
BLasso(1.0)	0.78	0.75	$0.54_{(10.0)}$	$0.47_{(0.1)}$
cMLP	0.96	**0.98**	0.52	0.52
cLSTM	0.71	0.81	0.57	**0.57**

K is the number of time series.
T is the number of samples.
F is used to determine the nonlinear level and chaos in the time series.

Results on TrainTicket Data

1) We delete the microservice ‘ts-basic-service’ from the system which results in the other four microservice emitting error logs.
2) 226 error logs are emitted.

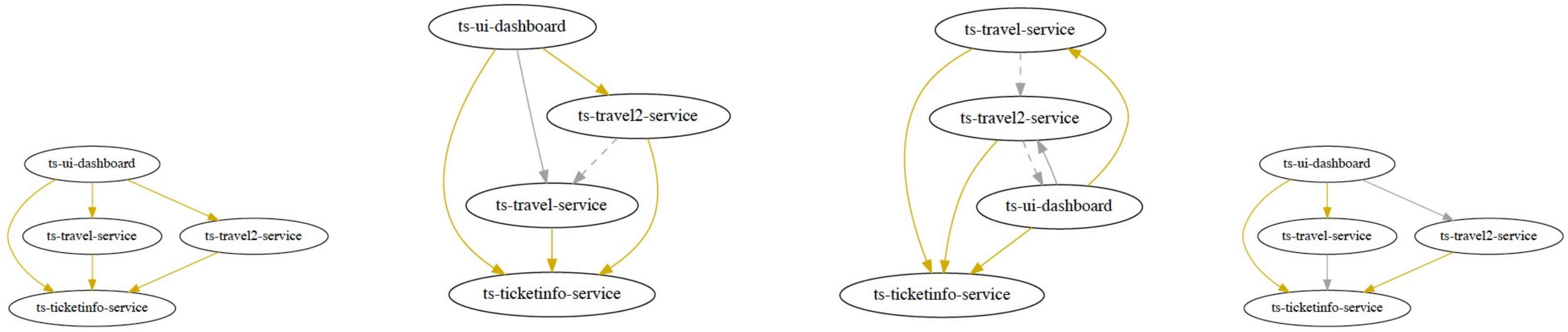


Fig. 4: Causal graphs inferred using different linear and nonlinear Granger methods. (left to right): Ground truth, MMPC(ParCorr, bin=100ms), BLR(1.0, bin=100ms), and cMLP(bin=10ms) model. False +ves (superfluous causal relationships) are marked by dashed gray edges and false -ves (missing causal relationships) with gray edges.

Results on TrainTicket Data

- In this experiment,
  - All the models' results shown in the table with their optimal parameter setting.
  - Linear Granger models performs better on this dataset from the results.

- Findings:
  - If K (the number of time series) is small, cMLP is the best candidate since it performs well on both linear and nonlinear Granger causality models even with a small T.
  - cLSTM model works better to capture more complicated nonlinear dependencies.
  - BLR and Blasso have good performance on linear time series data when T is large enough.

Methods	bin(ms)	Precision	Recall	F_1
MMPC (ParCorr)	10	0.45	1.0	0.62
MMPC (ParCorr)	10^2	0.8	0.8	**0.8**
MMPC (ParCorr)	10^3	1.0	0.4	0.57
MMPC (RCoT)	10	0.62	1.0	0.76
MMPC (RCoT)	10^2	0.33	0.2	0.25
MMPC (RCoT)	10^3	0.66	0.4	0.5
BLR(1.0)	10-10^2	1.0	0.6	**0.75**
BLR(1.0)	10^3	0.75	0.6	0.66
BLasso(1.0)	10	0.66	0.8	0.72
BLasso(1.0)	10^2-10^3	0.75	0.6	0.66
cMLP	10	1.0	0.60	**0.75**
cMLP	10^2	0.50	0.80	0.67
cMLP	10^3	0.56	1.0	0.71
cLSTM	10	0.45	1.0	0.62
cLSTM	10^2	0.41	1.0	0.59
cLSTM	10^3	0.38	1.0	0.56
After tuning parameters using ground truth information				
BLasso(10.0)	10	1.0	1.0	**1.0**
BLasso(10.0)	10^2	1.0	0.6	0.75
BLasso(10.0)	10^3	1.0	0.4	0.57

TABLE IV: Performance results of different causal inference methods. PC models: MMPC with partial correlation and RCoT. Linear Granger causality models: BLR, BLasso. Neural Granger causality models: cMLP, cLSTM.

Conclusion & Future Work

- Conclusion
  - Hundreds to thousands of microservice and complex dependency relationships among them makes root casual analysis extremely challenging.
  - These dependency information is often unknown.
  - We develop a system using only log data to learn the dependency graph among microservices and carefully study the performance of linear and nonlinear Granger causality models on learning causal relationships.

- Future Work
  - We will extend our analysis on a large dataset involving multiple faults and microservices.
  - We also want to develop new deep causal models for causal learning, which can directly take the raw log data as inputs.

Thanks