AIOps for Cloud Reliability
Research and Development

AIOPS 2023
Academic Saloon
TU Berlin, May 23-25, 2023

https://aiops2023.github.io/aiops2023/

Prof. Jorge Cardoso
E-mail: jorge.cardoso@huawei.com
Chief Engineer for AIOps
Munich Research Center

23.05.2023

Overview
Talk & Conference

AIOps for Cloud Reliability: Research and Development

Abstract.

We started applying machine learning and predictive analytics (aka AIOps) to anticipate and respond to failures in real-time since 2016. The objective of our work has been to reduce the human intervention needed to execute day-to-day operations in HUAWEI CLOUD and datacenters, and to improve infrastructure reliability and availability.

This presentation provides: 1) an overview of emerging technologies in the field of automation, monitoring, observability and cloud operations; 2) a timeline of our past work on distributed trace analysis, log analysis, time series analysis, secure operations, hardware failure prediction, network verification, and AI-based offloading; 3) a list of future research topics in our pipeline; and 4) a brief description of our work on the use of LSTM, BERT, Attention Networks to solve cloud reliability problems.

This talk also discusses concrete problems we have addressed with a sketch of the solutions developed.

AIOPS 2023

Academic Saloon

Berlin, May 23-25, 2023

Organized by **Huawei - TU Berlin Innovation Lab**, **DOS TU Berlin**

Welcome to AIOPS 2023

We are very happy to announce that we are organizing a workshop on artificial intelligence for software development and IT operations on our beautiful university campus at Technical University Berlin. We aim at gathering researchers from academia and industry to present their experiences, results, and work in progress in this field. Auto-instrumentation, open telemetry, deep learning techniques for software coding, testing on the fly and many other trends impact the process of software development, verification, and operation. Our goal is to spend three days discussing the challenges in our field and create a community roadmap with topics to look for, which can help us and our PhD students to find orientation and collaboration opportunities. To enable a direct and fruitful discussion, we aim for a selected number of participants. We envision five rounds of discussions, three hours each, on topics determined beforehand via voting. For each topic, we will invite 2-3 short introductory presentations to set the scene for the follow-up discussion. The last session will be devoted to further open questions and the next steps.

Following the great sucess of last two years AIOPS 2020 (AIOPS 2020, videos, proceedings) and AIOPS 2021 (AIOPS 2021, videos, proceedings) this workshop will be held as a standalone event in Berlin from 23-25 of May 2023. The event will take place at the Einsten Digital Center with the address Wilhelmstraße 67, 10117 Berlin.

One of the goals of the event is to encourage a discussion on the important questions in the area. Therefore we intend to organize two panel discussions. The topics for the panel are to be decided via voting prior to the event. You can cast your vote for any of them, and for as many as you would like. The voting ends on 24.05.2025. The three topics with the most votes are to be discussed during the event. You can also suggest topics of your interest. The access to the topics for the panel discussion can be found on the following link: Panel Discussion Topic Selection.

Topics of Interest

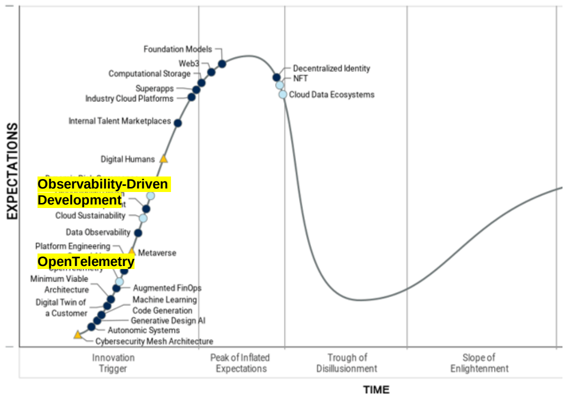
The focus of the workshop involves, but it is not limmited to the following topics:

- Autonomous instrumentation
- Safe and relible intelligent software coding
- Log analysis
- Anomaly detection
- Failure mode analysis
- Self-healing, self-correction and auto-remediation
- Benchmarking in AIOps
- Hardware and software failure prediction
- Root cause analysis
- Performance management
- Predictive and prescriptive maintenance
- Resiliency, reliability, and quality assurance
- IT system dependability
- Energy-efficient cloud operation
- Resource management
- Autonomous service provisioning
- Visual analytics and interactive machine learning
- Fault injection, verification testing and chaos engineering
- Use-cases, testbeds, evaluation scenarios

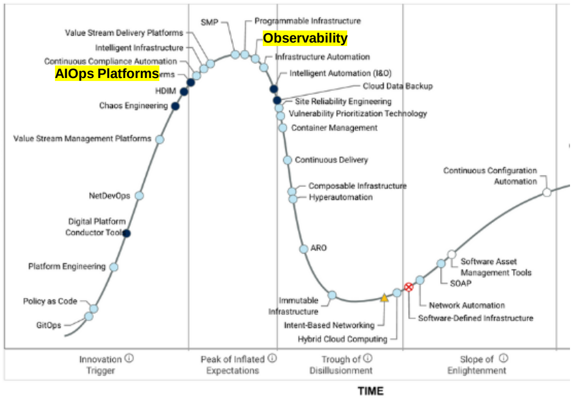
https://aiops2023.github.io/aiops2023/

AI for Cloud Operations
Trends and Hypes

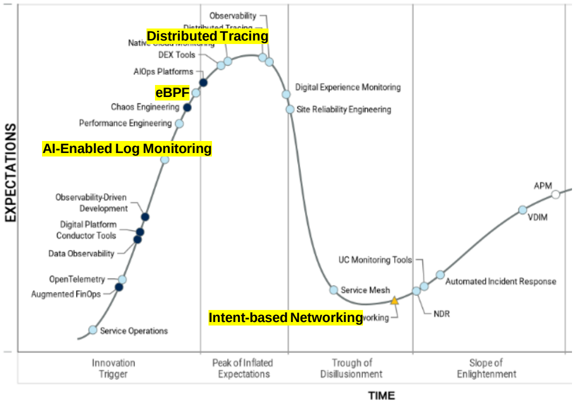
Hype Cycle for Emerging Technologies, 2022



Hype Cycle for I&O Automation, 2022



Hype Cycle for Monitoring, Observability and Cloud Operations, 2022



Observability-driven development (ODD) is an engineering practice that provides visibility and context into system state and behavior by designing systems to be observable. It relies on instrumenting code to expose system's internal state, to make it easier to detect, diagnose and resolve system anomalies

OpenTelemetry is a collection of specifications, tools, APIs and SDKs to support open-source instrumentation and observability for software.

Observability is the characteristic of software that enables them to be understood from their behavior. Tools enable to explore high-cardinality telemetry to explain faulty behavior

AIOps platforms analyze monitoring data, events and operational information to automate IT operations. Five characteristics: cross-domain data; topology; correlation between events; pattern recognition to detect incidents and root cause; and remediation.

Extended Berkeley Packet Filter (eBPF) is an enhancement to the Linux kernel that allows specific instruction sets to run inside the kernel.

AI-enabled log-monitoring applies ML/AI to traditional log-monitoring to reduce operator's cognitive load via context and correlation of large volumes of log data from multiple data sources

Intent-based networking helps design, provision and operate a network based on business policies. Four characteristics: (1) translating higher-level policies to configurations; (2) automating network activities; (3) awareness of network state/health; and (4) continuous assurance and dynamic optimization

Among the emerging technologies, O&M-related technologies emerge in large numbers, focusing on observability and AI-driven analysis

HUAWEI

AI for Cloud Reliability
Fields of R&D

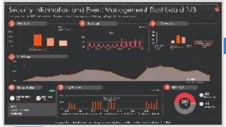
Trace Analysis

Anomaly detection *(traces, call chains)*
- Auto-Encoding Variational Bayes (AVB)
- Sequence/pairwise alignment
- Needleman-Wunsch algorithm
- **[CCGRID 2019, Cloud 2019]**

Log Analysis

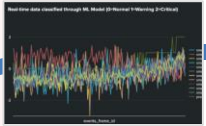
Root-cause analysis *(logs)*
- Template Mining, Temporal Correlation, Statistical Noise Removal
- Deep Learning
- Attention networks
- **[ECML 2020, ICPC 2022]**

Secure Operations



Command Analysis
- Pattern-matching
- Deep Learning, BERT, Attention networks
- **[to be published]**

Hardware Failure Prediction



RAM, HDD, Optical
- Machine Learning
- Random Forests, Grid Search
- **[CCGrid 2023, DSN 2023]**

Network Chaos Engineering
- Gray failure models
- Synthetic/fuzz traffic
- Network Observability
- High cardinality metric analysis

Flow Analysis
(Traces, Call Chains)
- Flow modeling
- Attention networks
- Decision trees

Trouble Ticket Analysis
- Document Question Answering
- Summarization
- BERT

Continuous Verification
- Template mining
- Log analysis
- Chi-Square, Kullback-Liebler Divergance

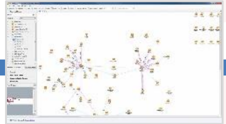
AI for Cloud Reliability

Cloud Analytics

AI for Network

AI for Cloud-Edge

Network Verification



VPC/P4 Verification
- SAT solver, BDD, eBPF
- **[NSDI 2024 ?]**

BGP Verification
- A*, Trie, MTBDD
- SMT solvers

Full-stack Network Verification
- eBFP
- ML, network state inference
- Cross-layer correlation
- Multi time-series analysis

Data Quality Analysis
(MELT)
- Statistics, sequence analysis
- LLM, NLP,
- Unit tests for data

End2End Network Verification
- Distributed Tracing, eBFP
- ML, cross-service correlation
- Multi time-series analysis

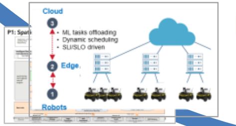
Time Series

Anomaly detection *(metrics)*
Failure prediction
- AutoML, outlier detection
- Symbolic Aggregate approXimation (SAX)
- Uni/multi time-series analysis
- **[AIOPS 2020a, AIOPS 2020b]**

Auto Change

Continuous Verification
- Template Mining
- Pearson's Correlation
- Machine Learning, Clustering
- **[ICDM 2020]**

AI-based Offloading



Edge/Cloud offloading
- Multi-criteria decision-making (MCDM)
- Adaptive Multi-Object Tracking (MOT)
- **[MIPS 2024 ?]**

Efficient AI Training
- Distance/density-based novelty detection
- Data pruning
- Model generalization

Efficient Logging
(Logs)
- Structured Logging
- Template Mining
- Local, decentralized analysis

Blast Radius Analysis
- eBPF and distributed traces
- Dynamic service topology
- Causality
- Process mining

Nov 2020 **Huawei-TUB Innovation Lab** (AIOps, AI for Networks, Intelligent CDN, Data Analytics)

2016 | 2018 | 2020-2021 | 2023 | 2024-2025 | 2026

QuLog: Data-Driven Approach for Log Instruction Quality Assessment. ICPC 2022
Self-Supervised Log Parsing, ECML PKDD 2020
Anomaly Detection and Classification using Distributed Tracing and Deep Learning. CCGRID 2019
Anomaly Detection from System Tracing Data using Multimodal Deep, IEEE Cloud 2019
Online Memory Leak Detection in the Cloud-based Infrastructures,)AIOPS 2020)
An Optical Transceiver Reliability Study based on SFP Monitoring and OS-level Metric Data, CCGrid 2023
HiMFP: Hierarchical Intelligent Memory Failure Prediction for Cloud Service Reliability, DSN 2023

Self-Attentive Classification-Based Anomaly Detection in Unstructured Logs (ICDM 2020)
IAD: Indirect Anomalous VMMs Detection in the Cloud-based Environment (AIOPS 2020)
Cost-Aware Reachability Policy Management in Public Cloud (NSDI, 2024)

HUAWEI

AI for Cloud Reliability

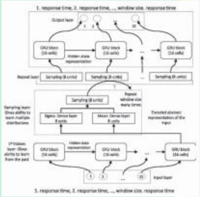
LSTM, Attention, BERT

LSTM

TRACE ANOMALY DETECTION

DISTRIBUTED TRACES/METRIC (2019)

Uses LSTM Networks to build a sequence model for the distributed traces and their response time. The model is used to detect anomalous traces



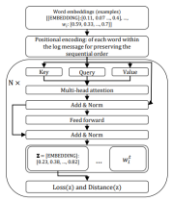
Anomaly Detection and Classification using Distributed Tracing and Deep Learning. Nedelkoski, S.; Cardoso, J. and Kao, O. In 19th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID), 2019.

ATTENTION NETWORKS

LOG ANOMALY DETECTION

LogSy (2020)

Uses Attention Networks to build a model which recognized normal logs and logs generated when a system is faulty

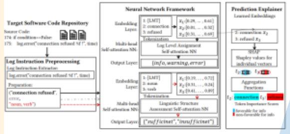


Self-Attentive Classification-Based Anomaly Detection in Unstructured Logs. Nedelkoski, S.; Bogatinovski, J.; Acker, A.; Cardoso, J. and Kao, O. In 20th IEEE International Conference on Data Mining (ICDM), Italy, 2020.

LOG QUALITY ANALYSIS

QULOG (2022)

Uses Attention Networks to build a language model from GitHub top rated open source projects. The model is used to evaluate the quality of log statements

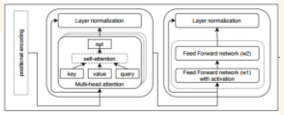


QuLog: Data-Driven Approach for Log Instruction Quality Assessment. Bogatinovski, J.; Nedelkoski, S.; Acker, A.; Cardoso, J. and Kao, O. In 30th IEEE/ACM International Conference on Program Comprehension, 2022.

LOG TEMPLATE MINING

NULOG (2020)

Uses Attention Networks to build a language model from the logs generated from an application. The model is used to identify the variables of logging statements

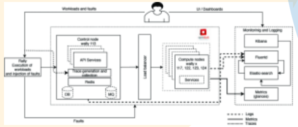


Self-Supervised Log Parsing. Nedelkoski, S.; Bogatinovski, J.; Acker, A.; Cardoso, J. and Kao, O. ECML-PKDD, 2020.

TRACE ANOMALY DETECTION

DISTRIBUTED TARCES (2020)

Uses Attention Networks to build a sequence model for the distributed traces generated by OpenStack. The model is used to detect anomalous traces



Self-Supervised Anomaly Detection from Distributed Traces. Bogatinovski, J.; Nedelkoski, S.; Cardoso, J. and Kao, O. In IEEE/ACM 13th International Conference on Utility and Cloud Computing (UCC), 2020

BERT

SECURE OPERATIONS

OPERATORS' COMMANDS (2023)

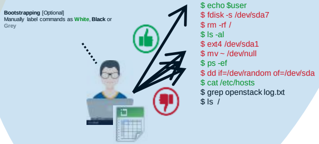
Uses LLM to build a sequence model to analyze commands, parameters and flags

Part of speech	Precision	Recall	F1 score
COMM	0.98	0.90	0.94
FLAG	1.00	1.00	1.00
FLAG_VALUE	0.88	0.94	0.91
OPERATOR	0.93	0.82	0.87
PARAM	0.82	0.86	0.84
SUBCOMM	0.84	0.81	0.83

SECURE OPERATIONS

COMMUNITY LABELLING (2023)

Uses LLM and Expectation–Maximization (EM) to assign risk labels to command based on their complexity and operators expertise



System Comprehension

- Generate human-like descriptions from technical architectures diagrams
- Answer questions with respect to a system behavior based on a given problem/context, knowledge base, documentation.
- Scan kubernetes clusters, diagnosing and triaging issues in English (e.g., k8sgpt.ai)

Chaos Engineering (Testing)

- Automatically generate call chains for end-to-end testing to improve cloud reliability (cf. tracetest.io)

Intent-based Networking (Change Mng)

- Translate natural language description of intents to domain specific language to configure data center networks (cf. Warp for Shell, autoregex.xyz, Cogram for SQL)

Standard Operation Procedures (SOP)

- Dynamically improve SOP procedures by building a sequence model with BERT to generalize/summarize operators' actions
- Based on objectives, suggest workflow steps (e.g., adept.ai)

Customer support & Trouble Tickets

- Understand the meaning of text for sentiment analysis, named entity recognition, and text classification (e.g., Viable, Enterpret, Cohere, and Anecdote).
- Automatically create tags/captions for trouble tickets, e.g., severity, importance, location, system.

Incident Response

- Summarizes the state of systems in response to questions during critical incidents (e.g., wildmoose.ai)

Development | Operations

RNN, LSTM — < 2014
RNN with attention — 2014
Self-attention, Transformers — 2017
BERT, GPT, ELMo — 2018
RoBERTa GPT-2 — 2019
GPT-3 DeBERTa — 2020
ChatGPT — 2022
PanGu-Σ — 2023
2025
2027

HUAWEI

Overview of AIOps Research

1990-2020

Results

- Majority of research (670 papers, 62.1%) are associated with failure management (FM)
  - Online failure prediction (26.4%)
  - Failure detection (33.7%)
  - Root cause analysis (26.7%)
- Most common problems in FM
  - Software defect prediction, system failure prediction, anomaly detection, fault localization and root cause diagnosis
- Failure detection has gained particular traction in recent years (71 publications for the 2018-2019 period)
- Root cause analysis (39) and online failure prediction (34)
- Failure prevention and remediation are the areas with least research

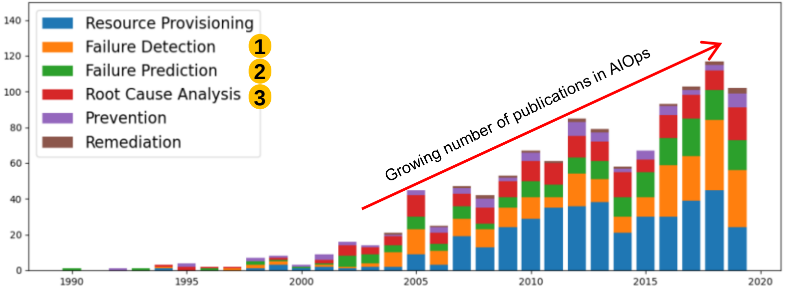


Fig. 4: Published papers in AIOps by year and categories from the described taxonomy.

Table 3: Selection of result papers grouped by data sources, targets and (sub)categories.

Ref.	Data Sources									Targets					Cat.
	Source Code	Testing Resources	System Metrics	KPIs/SLO data	Network Traffic	Topology	Incident Reports	Event Logs	Execution Traces	Source Code	Application	Hardware	Network	Datacenter	
[27]	•									•					1.1
[32]	•	•								•					1.2
[16]			•								•			•	1.3
[41]			•	•							•			•	1.3
[29]											•			•	1.4
[47]			•									•			2.1
[14]			•									•		•	2.1
[12]			•					•				•			2.1
[46]								•				•	•		2.1
[8]		•									•			•	2.2
[11]		•	•								•				2.2
[17]		•	•								•			•	2.2
[35]		•			•						•			•	2.2
[24]								•			•			•	2.2
[37]								•			•	•	•	•	2.2
[45]								•			•				2.2
[43]	•							•						•	3.1
[42]				•							•				3.1
[40]			•	•							•			•	3.1
[21]					•	•							•		3.1
[22]					•	•					•		•		3.1
[15]								•			•				3.1
[10]								•	•		•			•	3.1
[6]									•		•			•	3.1
[28]									•		•				3.1
[30]					•								•	•	3.2
[49]	•							•						•	3.3
[1]	•	•								•	•				4.1
[33]			•			•						•		•	4.1
[5]					•							•	•		4.1
[44]	•							•			•				4.2
[4]	•		•	•							•				4.2
[19]					•	•							•		4.2
[9]									•		•			•	4.2
[36]		•				•					•				4.3
[7]			•	•							•			•	4.3
[26]								•			•			•	4.3
[2]									•		•			•	4.3
[39]							•				•			•	5.1
[48]							•				•			•	5.2
[25]							•	•			•			•	5.2
[38]			•	•							•			•	5.3

(Sub)Category Legend					
1.1	Software Defect Prediction	2.2	System Failure Prediction	4.2	Root Cause Diagnosis
1.2	Fault Injection	3.1	Anomaly Detection	4.3	RCA - Others
1.3	Software Rejuvenation	3.2	Internet Traffic Classification	5.1	Incident Triage
1.4	Checkpointing	3.3	Log Enhancement	5.2	Solution Recommendation
2.1	Hardware Failure Prediction	4.1	Fault Localization	5.3	Recovery

A Systematic Mapping Study in AIOps. Notaro, P.; Cardoso, J. and Gerndt, M. In AIOPS 2020 International Workshop on Artificial Intelligence for IT Operations, Springer, 2020.

Hardware Failure Prediction

Memory Failure Prediction

PAIN POINT

Several incidents of in cloud computing infrastructures are caused by hardware failures

Fig. *Hardware failures (e.g., hard drives, memory, optical connectors) are the root cause of many cloud failures*

§	Root cause	#Sv	Cnt	%	Cnt '09-'15
	UNKNOWN	29	355	-	M.M.M.M.M.M.M
5.1	UPGRADE	18	54	16	7.4.M.5.M.4.7
5.2	NETWORK	21	52	15	4.4.6.8.M.8.5
5.3	BUGS	18	51	15	M.4.9.8.9.9.2
5.4	CONFIG	19	34	10	2.2.7.2.5.M.4
5.5	LOAD	18	31	9	2.5.5.5.4.8.2
5.6	CROSS	14	28	8	-.2.4.M.5.3.4
5.7	POWER	11	21	6	5.4.3.5.3.1.-
5.8	SECURITY	9	17	5	7.-.2.1.3.4.-
5.9	HUMAN	11	14	4	-.1.4.4.2.1.2
5.10	STORAGE	4	13	4	2.-.-.3.5.3.-
5.11	SERVER	6	11	3	-.3.-.2.2.4.-
5.12	NATDIS	5	9	3	1.1.3.2.1.1.-
5.11	HARDWARE	4	5	1	1.-.-.3.1.-.-

[1] Why Does the Cloud Stop Computing? Lessons from Hundreds of Service Outages

Problem

- In computing infrastructures, memory failure is the most important cause of system failure

TECHNOLOGIES

Combine hierarchical memory features and ML techniques for failure prediction

	Key technology
1	Static features (manufacturer, frequency, …), MCE Log (CE, UCE Error), Memory Events (CE storm, overflow, …)
2	Unique deeper level features (bit-level)
3	Combine in-band and out-band data
4	Hierarchical MFP framework
5	Combine expert rules and ML model

- **Insight**. Bit-level features and patterns are extremely important in predicting memory failure for Huawei V5 servers

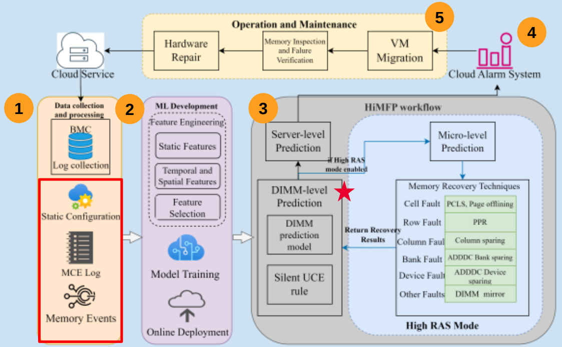
DESCRIPTION

MAIN ACHIEVEMENT

Feature Development and System Design

- Expert rules and Bit-level CE features
- Hierarchical framework to adapt multi-level failure recovery techniques
- Outperformed baseline algorithm Intel/ByteDance (2022) by 11% (F1)

HOW IT WORKS



Design of memory failure prediction pipeline. Only the "star" ★ is running in production.

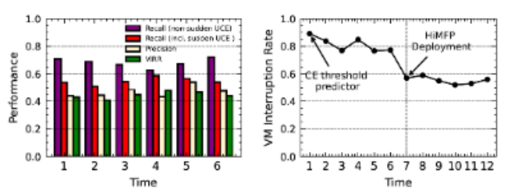
ASSUMPTIONS & LIMITATIONS

- Data quality and timeliness are key elements for a proper failure prediction

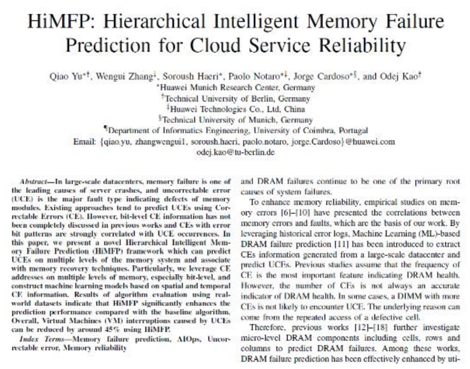
TRL 9: *Algorithm operates in production environment and reduces VM interruptions.*

IMPACT

Migrate customers VMs before failures happen



VM interruption rate dropped ~20% after memory failure prediction algorithm was deployed in production



HiMFP: Hierarchical Intelligent Memory Failure Prediction for Cloud Service Reliability

Qiao Yu*†, Wengui Zhang‡, Soroush Haeri*, Paolo Notaro*‡, Jorge Cardoso*§, and Odej Kao†
*Huawei Munich Research Center, Germany
†Technical University of Berlin, Germany
‡Huawei Technologies Co., Ltd, China
§Technical University of Munich, Germany
¶Department of Informatics Engineering, University of Coimbra, Portugal
Email: {qiao.yu, zhangwengui1, soroush.haeri, paolo.notaro, jorge.Cardoso}@huawei.com
odej.kao@tu-berlin.de

Abstract—In large-scale datacenters, memory failure is one of the leading causes of server crashes, and uncorrectable error (UCE) is the major fault type indicating defects of memory modules. Existing approaches tend to predict UCEs using Correctable Errors (CE). However, bit-level CE information has not been completely discussed in previous works and CEs with error bit patterns are strongly correlated with UCE occurrences. In this paper, we present a novel Hierarchical Intelligent Memory Failure Prediction (HiMFP) framework which can predict UCEs on multiple levels of the memory system and associate with memory recovery techniques. Particularly, we leverage CE addresses on multiple levels of memory, especially bit-level, and construct machine learning models based on spatial and temporal CE information. Results of algorithm evaluation using real-world datasets indicate that HiMFP significantly enhances the prediction performance compared with the baseline algorithm. Overall, Virtual Machines (VM) interruptions caused by UCEs can be reduced by around 45% using HiMFP.

Index Terms—Memory failure prediction, AIOps, Uncorrectable error, Memory reliability

and DRAM failures continue to be one of the primary root causes of system failures.

To enhance memory reliability, empirical studies on memory errors [6]–[10] have presented the correlations between memory errors and faults, which are the basis of our work. By leveraging historical error logs, Machine Learning (ML)-based DRAM failure prediction [11] has been introduced to extract CEs information generated from a large-scale datacenter and predict UCEs. Previous studies assume that the frequency of CE is the most important feature indicating DRAM health. However, the number of CEs is not always an accurate indicator of DRAM health. In some cases, a DIMM with more CEs is not likely to encounter UCE. The underlying reason can come from the repeated access of a defective cell.

Therefore, previous works [12]–[18] further investigate micro-level DRAM components including cells, rows and columns to predict DRAM failures. Among these works, DRAM failure prediction has been effectively enhanced by uti-

HiMFP: Hierarchical Intelligent Memory Failure Prediction for Cloud Service Reliability (DSN'23). Q. Yu, et al.. 2023.

Anomaly Detection
Detecting Faulty Hypervisors

PAIN POINT

Virtualization failures affect VMs but cannot be observed directly

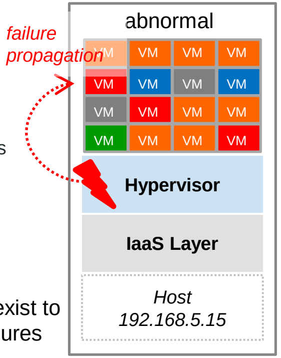


Fig. VMs exhibit problems when the hypervisor has technical issues

Problem

- No effective solution exist to detect hypervisors failures

TECHNOLOGIES

Indirect approach to detect hypervisor failures by monitoring VMs

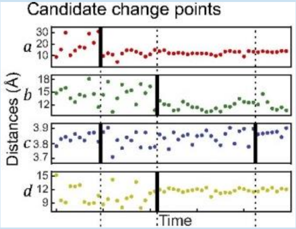


Fig. Several time-series generated by several VMs running in the same hypervisor

- **Insight**. When an hypervisor is malfunctioning, resource saturation of VMs suddenly changes, within a window w

DESCRIPTION

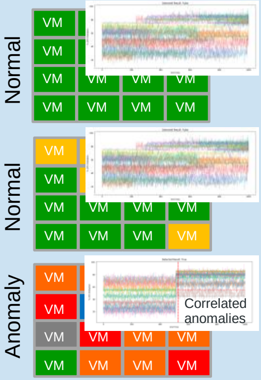
APPROACH

Quorum change-point detection

- Analyzes individual time-series, and uses change points and voting to decide whether there is an hypervisor malfunction
- Key results: **F1 72%** (2 VMs); **80+%** (3+ VMs)

HOW IT WORKS

Analyze VM resources to detect correlated anomalies



Method 1 (Change Points)
1. Treat time-series as univariate
2. Detect change points
3. Vote to decide global changes

Method 2 (Isolation Forest)
1. Treat time-series as features
2. Detect significant changes

Method 3 (ECP E.Divisive)
1. Treat time-series as multivariate
2. Detect multiple change points

ASSUMPTIONS & LIMITATIONS

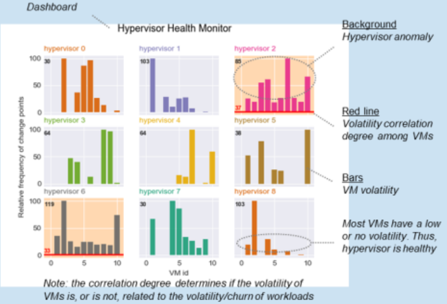
- Datasets used for evaluation were collected from simulation environment, synthetic data generator and public sources

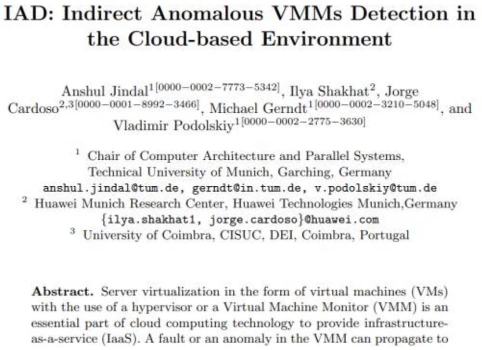
TRL 5. *Basic technological components are integrated with realistic supporting elements so it can be evaluated in testbed environment*

IMPACT

Predictive Maintenance

Migrate customers' VMs before hypervisors fail





IAD: Indirect Anomalous VMMs Detection in the Cloud-based Environment

Anshul Jindal[1][0000–0002–7773–5342], Ilya Shakhat[2], Jorge Cardoso[2,3][0000–0001–8992–3466], Michael Gerndt[1][0000–0002–3210–5048], and Vladimir Podolskiy[1][0000–0002–2775–3630]

[1] Chair of Computer Architecture and Parallel Systems, Technical University of Munich, Garching, Germany
anshul.jindal@tum.de, gerndt@in.tum.de, v.podolskiy@tum.de
[2] Huawei Munich Research Center, Huawei Technologies Munich,Germany
{ilya.shakhat1, jorge.cardoso}@huawei.com
[3] University of Coimbra, CISUC, DEI, Coimbra, Portugal

Abstract. Server virtualization in the form of virtual machines (VMs) with the use of a hypervisor or a Virtual Machine Monitor (VMM) is an essential part of cloud computing technology to provide infrastructure-as-a-service (IaaS). A fault or an anomaly in the VMM can propagate to

IAD: Indirect Anomalous VMMs Detection in the Cloud-based Environments, Jindal, A.; Shakhat, I.; Cardoso, J.; Gerndt, M. and Podolskiy, V. International Workshop on AIOPS 2021, Springer, 2021.

HUAWEI

Root Cause Analysis
Application Logs

PAIN POINT

Once an anomaly is detected, root cause analysis (RCA) is fundamental to resolve problems

Several forms of RCA exist

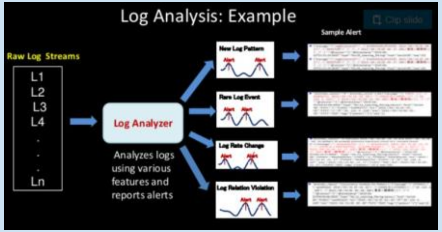
- App logs, metrics, traces, events, etc.

Problem

- Mainly log severity level has been used for AD & RCA
- High number of false positives

TECHNOLOGIES

Use a novel, fast algorithms for RCA using log analytics



- **Insight**. Recent research shows it is possible to model the underlying structure of application logs using machine learning [1, 2]

DESCRIPTION

MAIN ACHIEVEMENT

Performs RCA based on application logs

- Anomaly detection in large volume of semi-structured logs
- Correlation between metric anomalies and alarms and logs
- Log summarization that 100x reduces amount of data a human has to process

HOW IT WORKS

1. **Template mining**. Fast log template reconstruction using Drain algorithm
2. **Natural Language Processing**. Language-aware log parsing and keyword extraction using NLP approaches (www.spacy.io)
3. **Dynamic Grouping**. Time-series classification using Poisson model Grouping using Pearson correlation coefficient Distance-aware correlation

10000 error messages

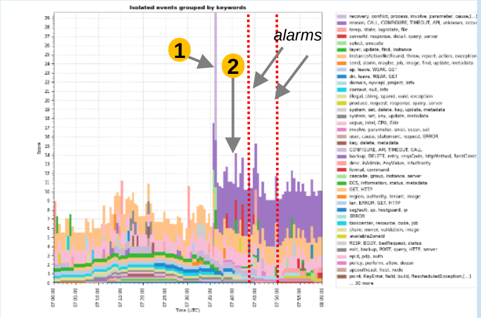
ASSUMPTIONS & LIMITATIONS

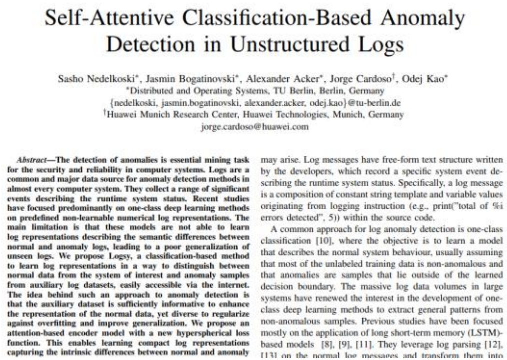
- On-demand processing requires a certain range of logs to learn normality
- Results depend on service logs quality

TRL 5. *Basic technological components are integrated with realistic supporting elements so it can be evaluated in testbed environment*

IMPACT

Lower troubleshooting time in 80%





Self-Attentive Classification-Based Anomaly Detection in Unstructured Logs

Sasho Nedelkoski*, Jasmin Bogatinovski*, Alexander Acker*, Jorge Cardoso†, Odej Kao*
*Distributed and Operating Systems, TU Berlin, Berlin, Germany
{nedelkoski, jasmin.bogatinovski, alexander.acker, odej.kao}@tu-berlin.de
†Huawei Munich Research Center, Huawei Technologies, Munich, Germany
jorge.cardoso@huawei.com

***Abstract*—The detection of anomalies is essential mining task for the security and reliability in computer systems. Logs are a common and major data source for anomaly detection methods in almost every computer system. They collect a range of significant events describing the runtime system status. Recent studies have focused predominantly on one-class deep learning methods on predefined non-learnable numerical log representations. The main limitation is that these models are not able to learn log representations describing the semantic differences between normal and anomaly logs, leading to a poor generalization of unseen logs. We propose Logsy, a classification-based method to learn log representations in a way to distinguish between normal data from the system of interest and anomaly samples from auxiliary log datasets, easily accessible via the internet. The idea behind such an approach to anomaly detection is that the auxiliary dataset is sufficiently informative to enhance the representation of the normal data, yet diverse to regularize against overfitting and improve generalization. We propose an attention-based encoder model with a new hyperspherical loss function. This enables learning compact log representations capturing the intrinsic differences between normal and anomaly**

may arise. Log messages have free-form text structure written by the developers, which record a specific system event describing the runtime system status. Specifically, a log message is a composition of constant string template and variable values originating from logging instruction (e.g., print("total of %i errors detected", 5)) within the source code.

A common approach for log anomaly detection is one-class classification [10], where the objective is to learn a model that describes the normal system behaviour, usually assuming that most of the unlabeled training data is non-anomalous and that anomalies are samples that lie outside of the learned decision boundary. The massive log data volumes in large systems have renewed the interest in the development of one-class deep learning methods to extract general patterns from non-anomalous samples. Previous studies have been focused mostly on the application of long short-term memory (LSTM)-based models [8], [9], [11]. They leverage log parsing [12], [13] on the normal log messages and transform them into

Self-Attentive Classification-Based Anomaly Detection in Unstructured Logs. Nedelkoski, S.; Bogatinovski, J.; Acker, A.; Cardoso, J. and Kao, O. In 20th IEEE International Conference on Data Mining (ICDM), 17-20 November, 2020, Italy, 2020.

HUAWEI

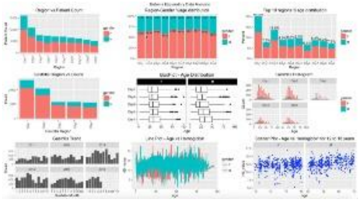
Anomaly Detection

Multi-modal Anomaly Detection

PAIN POINT

Move from single source, single dimension to multi-source & dimensions

Fig. Metrics, logs, and traces are monitored by separated systems



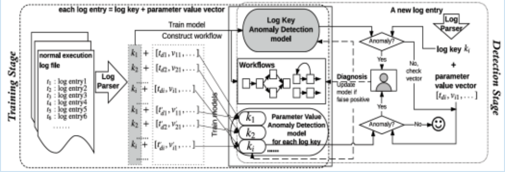
Problem

- High percentage of false positive alarms. Noisy signals requires new AD & RCA robust techniques

TECHNOLOGIES

Apply recent Sequence Learning approaches to AIOps

- State of the art results in many applications: image, video, translation and speech recognition to extract long-term dependencies



- e.g., unsupervised anomaly detection in log files (DeepLog)

DESCRIPTION

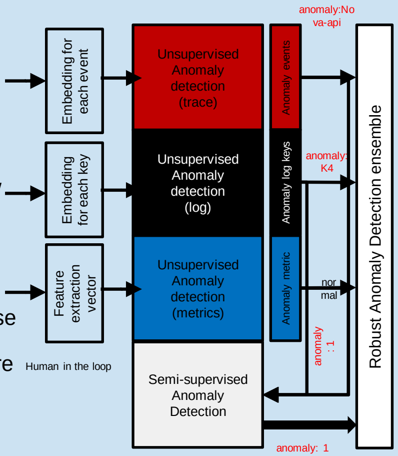
MAIN ACHIEVEMENT

New ensemble AI Algorithms to Detect Anomalies in Multi-source, Multi-dimension data

- Robust anomaly detection ensemble
- Extend approaches such as SkyWalking

HOW IT WORKS

1) Requests generate log events, traces, and metrics
2) Access and Data Transformation to provide an uniform view
3) Robust Anomaly Detection using an ensemble (multi-view)
4) Root Cause Analysis use the neural network and backward anomaly score propagation to identify the root of the problem



ASSUMPTIONS & LIMITATIONS

- Requires a special (not trivial) software development of recurrent neural networks, like LSTM
- Requires access to Topology Services

TRL 3: *Active research and development is initiated. Analytical studies and laboratory studies to validate analytical feasibility of the approach*

IMPACT

Lower false positive alarm rate

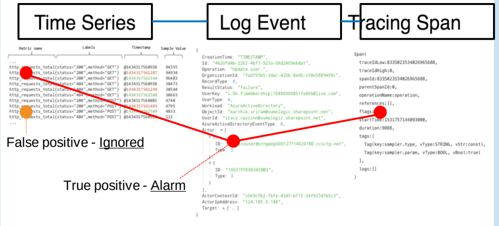
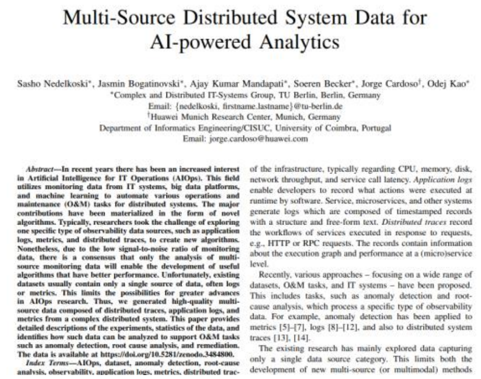


Fig. Multi-source analysis
Correlate single anomalies as a way to improve precision



Multi-Source Distributed System Data for AI-powered Analytics

Sasho Nedelkoski*, Jasmin Bogatinovski*, Ajay Kumar Mandapati*, Soeren Becker*, Jorge Cardoso†, Odej Kao*
*Complex and Distributed IT-Systems Group, TU Berlin, Berlin, Germany
Email: {nedelkoski, firstname.lastname}@tu-berlin.de
†Huawei Munich Research Center, Munich, Germany
Department of Informatics Engineering/CISUC, University of Coimbra, Portugal
Email: jorge.cardoso@huawei.com

***Abstract*—In recent years there has been an increased interest in Artificial Intelligence for IT Operations (AIOps). This field utilizes monitoring data from IT systems, big data platforms, and machine learning to automate various operations and maintenance (O&M) tasks for distributed systems. The major contributions have been materialized in the form of novel algorithms. Typically, researchers took the challenge of exploring one specific type of observability data sources, such as application logs, metrics, and distributed traces, to create new algorithms. Nonetheless, due to the low signal-to-noise ratio of monitoring data, there is a consensus that only the analysis of multi-source monitoring data will enable the development of useful algorithms that have better performance. Unfortunately, existing datasets usually contain only a single source of data, often logs or metrics. This limits the possibilities for greater advances in AIOps research. Thus, we generated high-quality multi-source data composed of distributed traces, application logs, and metrics from a complex distributed system. This paper provides detailed descriptions of the experiments, statistics of the data, and identifies how such data can be analyzed to support O&M tasks such as anomaly detection, root cause analysis, and remediation. The data is available at https://doi.org/10.5281/zenodo.3484800.**

***Index Terms*—AIOps, dataset, anomaly detection, root-cause analysis, observability, application logs, metrics, distributed trac-**

of the infrastructure, typically regarding CPU, memory, disk, network throughput, and service call latency. *Application logs* enable developers to record what actions were executed at runtime by software. Service, microservices, and other systems generate logs which are composed of timestamped records with a structure and free-form text. *Distributed traces* record the workflows of services executed in response to requests, e.g., HTTP or RPC requests. The records contain information about the execution graph and performance at a (micro)service level.

Recently, various approaches – focusing on a wide range of datasets, O&M tasks, and IT systems – have been proposed. This includes tasks, such as anomaly detection and root-cause analysis, which process a specific type of observability data. For example, anomaly detection has been applied to metrics [5]–[7], logs [8]–[12], and also to distributed system traces [13], [14].

The existing research has mainly explored data capturing only a single data source category. This limits both the development of new multi-source (or multimodal) methods

Multi-source Distributed System Data for AI-Powered Analytics. Nedelkoski, S.; Bogatinovski, J.; Mandapati, A. K.; Becker, S.; Cardoso, J. and Kao, O. In Service-Oriented and Cloud Computing (ESOCC), 2020.

HUAWEI

Anomaly Detection & Root Cause Analysis
Distributed Traces

PAIN POINT

While popular, only visualization tools exist for trace management

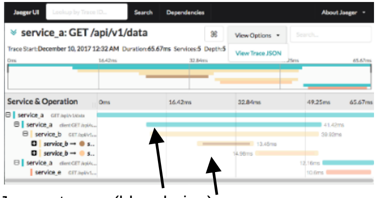


Fig. Jaeger traces (blue, beige)

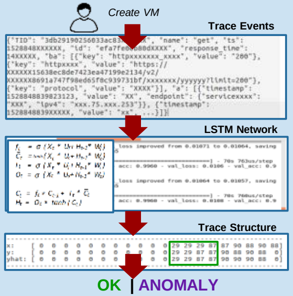
Current limitations

- Tracing tools only provides trace visualization
- Analyzing traces manually is error-prone and not scalable

TECHNOLOGIES

Apply recent ML and statistical approaches to process sequential data

- Explore the use of Deep Learning: Long Short Term Memory (LSTM)
- Explore the use of attention networks
- Explore the use of association rules



DESCRIPTION

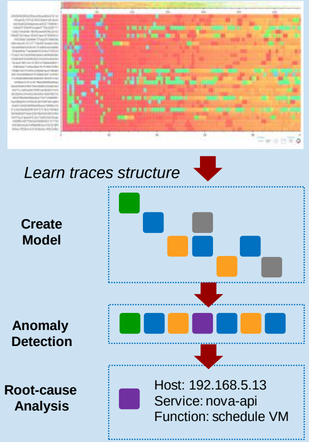
MAIN ACHIEVEMENT

Trace anomaly detection and root-cause analysis using trace structure

- Previous tentative using Deep Learning (LSTM, CNN), Machine Learning (Optiks), Sequence Analysis (LCS, Multiple sequence alignment, Needleman-Wunsch) algorithms did not enable a precise root cause analysis

HOW IT WORKS

1) *Learning*. For each service endpoint, learn the traces' structure it generates
2) *Modeling*. Aggregate all the traces into a behavior model
3) *Anomaly detection*. When a new trace is generate, compare its structure with the behavior model. If it was not seen before, an anomaly exists
4) *Root-cause analysis*. When an anomaly is detected, determine in which span it occurred and identify host, service, function



ASSUMPTIONS & LIMITATIONS

- Microservices are instrumented with tracing capabilities

TRL 4. *Small scale prototype. Basic technological components are integrated to establish that they will work together.*

IMPACT

Improve trace-based RCA in 90%

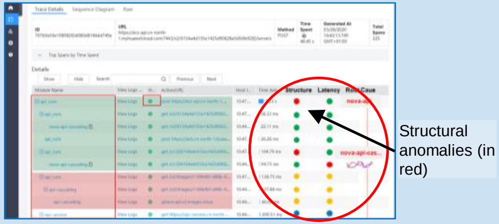
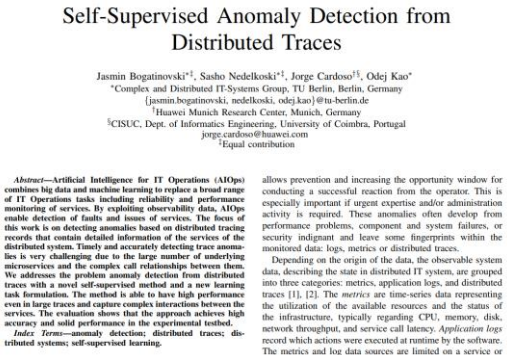


Fig. Trace management & trace analysis
Red circles show structural anomalies



Self-Supervised Anomaly Detection from Distributed Traces

Jasmin Bogatinovski*‡, Sasho Nedelkoski*‡, Jorge Cardoso†§, Odej Kao*
*Complex and Distributed IT-Systems Group, TU Berlin, Berlin, Germany
{jasmin.bogatinovski, nedelkoski, odej.kao}@tu-berlin.de
†Huawei Munich Research Center, Munich, Germany
§CISUC, Dept. of Informatics Engineering, University of Coimbra, Portugal
jorge.cardoso@huawei.com
‡Equal contribution

***Abstract*—Artificial Intelligence for IT Operations (AIOps) combines big data and machine learning to replace a broad range of IT Operations tasks including reliability and performance monitoring of services. By exploiting observability data, AIOps enable detection of faults and issues of services. The focus of this work is on detecting anomalies based on distributed tracing records that contain detailed information of the services of the distributed system. Timely and accurately detecting trace anomalies is very challenging due to the large number of underlying microservices and the complex call relationships between them. We addresses the problem anomaly detection from distributed traces with a novel self-supervised method and a new learning task formulation. The method is able to have high performance even in large traces and capture complex interactions between the services. The evaluation shows that the approach achieves high accuracy and solid performance in the experimental testbed.**

***Index Terms*—anomaly detection; distributed traces; distributed systems; self-supervised learning.**

allows prevention and increasing the opportunity window for conducting a successful reaction from the operator. This is especially important if urgent expertise and/or administration activity is required. These anomalies often develop from performance problems, component and system failures, or security indignant and leave some fingerprints within the monitored data: logs, metrics or distributed traces.

Depending on the origin of the data, the observable system data, describing the state in distributed IT system, are grouped into three categories: metrics, application logs, and distributed traces [1], [2]. The *metrics* are time-series data representing the utilization of the available resources and the status of the infrastructure, typically regarding CPU, memory, disk, network throughput, and service call latency. *Application logs* record which actions were executed at runtime by the software. The metrics and log data sources are limited on a service or

Self-Supervised Anomaly Detection from Distributed Traces. Bogatinovski, J.; Nedelkoski, S.; Cardoso, J. and Kao, O. In IEEE/ACM 13th International Conference on Utility and Cloud Computing (UCC), 2020

HUAWEI

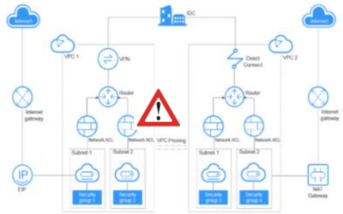
Anomaly Detection & Root Cause Analysis

Network Verification

PAIN POINT

Users deploy Virtual Private Networks (VPC) to organize their Virtual Machines (VM)

Fig. Complex VPC configuration errors are difficult to localize



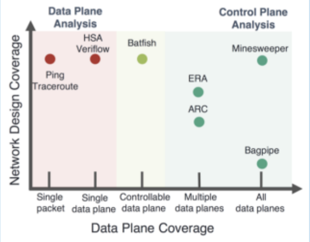
Problem

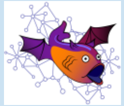
- Users are not able to verify the reachability of their VPCs, localize faults, carry recovery actions, …

TECHNOLOGIES

Optimize Binary Decision Diagrams (BDDs) for large-scale verification scenarios

Fig. Network analysis tools (left [1]) and the Bastfish network configuration analysis tool (below)





- **Insight**. The optimization of BDDs (e.g., via parallelization, pruning, compression) allows to speedup verification tasks while reducing the computing resources needed

DESCRIPTION

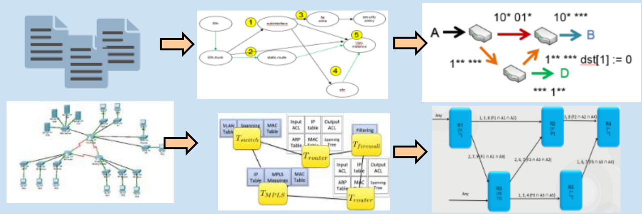
MAIN ACHIEVEMENT

VPC Reachability Verification

- Model networks with all of the elements, e.g., VPC, ECS, Subnets, ACLs, Security Groups, accurately using "extended" BDDs
- Compute all classes of packets that are expected to flow between a source-destination pair
- Highly efficient detection and localization mechanisms

HOW IT WORKS

1 VPC configuration **2** Network object model **3** Abstract data plane



Large-scale continuous verification of virtual networks with end-to-end support

ASSUMPTIONS & LIMITATIONS

- Users need to understand which rule of a given network element precludes the reachability and eventually manually correct it
- This task is complex. In the future, research on assisted correction is needed

TRL 5. *Basic technological components are integrated with realistic supporting elements so it can be evaluated in testbed environment*

IMPACT

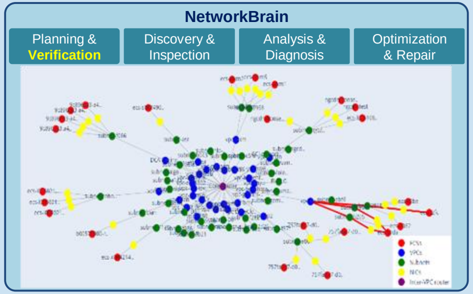
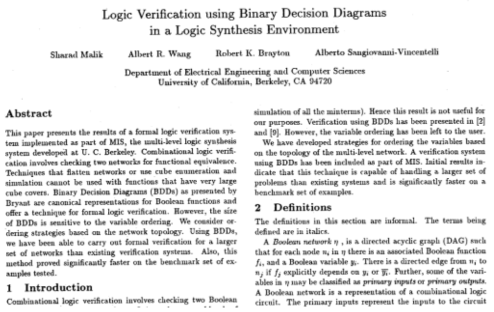


Fig. Network construction and verification. Colored nodes represent different types of equipment. Red lines show reachability problems between nodes.



Logic Verification using Binary Decision Diagrams in a Logic Synthesis Environment

Sharad Malik Albert R. Wang Robert K. Brayton Alberto Sangiovanni-Vincentelli

Department of Electrical Engineering and Computer Sciences
University of California, Berkeley, CA 94720

Abstract

This paper presents the results of a formal logic verification system implemented as part of MIS, the multi-level logic synthesis system developed at U. C. Berkeley. Combinational logic verification involves checking two networks for functional equivalence. Techniques that flatten networks or use cube enumeration and simulation cannot be used with functions that have very large cube covers. Binary Decision Diagrams (BDDs) as presented by Bryant are canonical representations for Boolean functions and offer a technique for formal logic verification. However, the size of BDDs is sensitive to the variable ordering. We consider ordering strategies based on the network topology. Using BDDs, we have been able to carry out formal verification for a larger set of networks than existing verification systems. Also, this method proved significantly faster on the benchmark set of examples tested.

1 Introduction

Combinational logic verification involves checking two Boolean

simulation of all the minterms). Hence this result is not useful for our purposes. Verification using BDDs has been presented in [2] and [9]. However, the variable ordering has been left to the user.

We have developed strategies for ordering the variables based on the topology of the multi-level network. A verification system using BDDs has been included as part of MIS. Initial results indicate that this technique is capable of handling a larger set of problems than existing systems and is significantly faster on a benchmark set of examples.

2 Definitions

The definitions in this section are informal. The terms being defined are in italics.

A *Boolean network* η, is a directed acyclic graph (DAG) such that for each node n_i in η there is an associated Boolean function f_i, and a Boolean variable y_i. There is a directed edge from n_i to n_j if f_j explicitly depends on y_i or $\overline{y_i}$. Further, some of the variables in η may be classified as *primary inputs* or *primary outputs*. A Boolean network is a representation of a combinational logic circuit. The primary inputs represent the inputs to the circuit

Logic verification using binary decision diagrams in a logic synthesis environment, Malik, Sharad, et al.,1988 IEEE International Conference on Computer-Aided Design. IEEE Computer Society, 1988.

[1] Beckett, A General Approach to Network Configuration Verification, SIGCOMM '17

HUAWEI

Thank you.

Bring digital to every person, home and organization for a fully connected, intelligent world.

HUAWEI