Red Hat Research Days presents

AIDA
A holistic AI-driven networking and processing framework for industrial IoT applications

September 21st | 3:00 - 4:30 PM CEST

Speaker
Anna Brunstrom
Karlstad University

Speaker
Muhammad Usman
Karlstad University

Speaker
Bestoun Ahmed
Karlstad University

Conversation Leader
Toke Høiland-Jørgensen
Red Hat

Conversation Leader
Simone Ferlin-Reiter
Red Hat
Contents

• Brief introduction to Karlstad University
• Background and overview of AIDA
• Distributed observability framework
• ML pipeline
Quick Facts about Karlstad University

• 19,000 students
• 260 doctoral students
• 1,500 staff
• Established 1999
• Teacher education since 1843
• Excellent research groups
  – Computer Science (CS)
  – Service Research Centre (CTF)
Computer Science

• 800 students
• ~60 staff
  – 20+ doctoral students
• Research profiles
  – Distributed systems and communications (DISCO)
  – Privacy and security (PriSec)
  – Software quality and digital modernization (SQuaD)

Our employees come from eighteen countries around the world and represent four continents.
AIDA
A HOLISTIC AI-DRIVEN NETWORKING AND PROCESSING FRAMEWORK FOR INDUSTRIAL IOT APPLICATIONS

RED HAT RESEARCH DAYS 2023-09-21
CONNECTED CYBER-PHYSICAL SYSTEMS ➔ NOW

- Needs/Trends:
  - Collecting and Making use of billions of sensor data ➔ IoT
  - Analyzing data and acting upon it in Real-time ➔ Analytics
  - Autonomous Decisions guided by algorithms ➔ ML
• Characteristics and Benefits
  • In software, virtualized, programmable, upgradable, commodity infrastructure, open, interoperable, customizable
  • Increase flexibility, reduce deployment time and cost
3 MAIN PILLARS FOR TRUSTWORTHY I-IOT APPLICATIONS

Data-Driven, Trustworthy Industrial IoT applications

- Getting The Data Fast, Under Guarantees
- Processing the Data Fast, under guarantees
- Making Sure, Data and Decisions are Correct
3 MAIN PILLARS FOR TRUSTWORTHY I-IOT APPLICATIONS

- **Data-Driven, Trustworthy Industrial IoT applications**

- **Real-time Networks** → WP1
- **Real-time Edge Processing** → WP2
- **Real-time ML-Testing And Validation** → WP3
3 MAIN PILLARS FOR TRUSTWORTHY I-IOT APPLICATIONS

Data-Driven, Trustworthy Industrial IoT applications

How to Configure Networks To provide Required Guarantees?

How to Monitor Big data Processing Edge Infrastructure?

How to Verify That ML Processing Is correct?
AIDA Architecture

Control Plane

Edge Node

Real-time Network

Edge Node

Sensors & Actuators
AIDA Architecture
AIDA Architecture
AIDA Architecture
Highlights – TSN Control Plane

• SDN based Control Plane for Time Sensitive Networks (TSN):
  – Microservice based Centralized Network Controller → OpenCNC → Open Source
    • Northbound: 802.1 Qdj, Southbound: NetConf/Yang for pushing configuration, verified through plugfest
    • Kafka-based Monitoring Backend for Telemetry
  – Endhost support for configuration of i.225/i.226 TSN cards through detd (intel)
  – Joint orchestration of TSN/Talker placement and Network Configuration

• Robust Network (Re-) Configuration
  – Synthesizing TSN configurations using external optimizer
    • Deep reinforcement Learning algorithm design ongoing
    • Digital-Twin based validation approach using simulator in the loop
    • Genetic Algorithm for finding tradeoff between optimality and cost for reconfiguration
Highlights – Real-time Performance Monitoring

• Design of AIDA Distributed Observability Framework (DESK)
  – Based on literature review on observability of distributed edge and containerized microservices
  – Complete implementation based on selected open source tools and metrics

• Experimentation and Analysis of DESK
  – Initial DESK overhead and usability analysis
  – Fault detection and recovery using monitored data at edge nodes

• Latency Monitoring with eBPF
  – Design of ePPing tool for passive RTT measurements
  – Filtering and aggregation for increased efficiency
  – Validation and performance evaluation (PAM 2023)
  – Integration in LibreQoS
  – Measurement study at an ISP in the US is ongoing
Highlights – ML pipeline

• Trustworthy ML in Production:
  – New method for using data augmentation for ML testing
  – New methods for ML Testing in production

• Concept Drift and ML Model Degradation:
  – Improving scalability of industrial processes using drift handling techniques
  – Proposing an adaptive drift detection mechanism

• ML pipeline and QA
  – DQ within MLOps
  – Model versioning and performance degradation
  – Formalizing a holistic robust MLOps framework

• Data-Centric ML Approach
  – Data quality scoring approach
  – Evaluation in real-time industrial use cases
  – Improve the overall ML performance is on going

• System anomaly detection using historical data.
  – Performing literature study on algorithms and challenged in anomaly detection.
  – Anomaly detection of customers energy consumption using historical consumption data.
Further information – selected pointers


- S. Sundberg, et. al., ”Efficient Continuous Latency Monitoring with eBPF”. Passive and Active Measurement (PAM), March 2023.


- Github: https://github.com/AIDA-KAU

- Web page: https://sola.kau.se/aida/
DISTRIBUTED OBSERVABILITY FRAMEWORK

RED HAT RESEARCH DAYS 2023-09-21
Container-based Edge Computing Platform
Challenges in Monitoring of Distributed Systems

- Modular
- Distributed
- Dynamic

- IT Systems

- Several alternatives
- Interconnected components
- Lack of customization

- Platforms

- Several Microservices
- Multiple Interconnections
- Dynamic

- Use Case

- Diverse
- Varying requirements

- Applications
Observability in Distributed Systems

- Latency (Request service time)
- Traffic (User demand)
- Errors (Rate of failed requests)
- Saturation (Overall system capacity)
Real-time Performance Observability & Optimization Framework

AIDA Overall Architecture

DistributEd obServability frameworkK (DESK)

Server-side Components/Services

Measurement Agent(s)
DistributEd obServability frameworK (DESK)

**Storage service** provides short-term and long-term data storage capabilities.

**Delivery service** transfers and routes data among multiple services.

**Visualization service** accesses processed data and transforms it to graphical outputs. **Notification service** is the reactive component of the architecture that initiates actions when data values change.

**Fusion service** integrates the collected data and analyzes it, e.g., SLA validations.

**P&O service** takes care of deployment and re-(configuration) of deployed microservices in the edge cluster.
CNCF hosts around 103 projects for observability and analysis (44 projects are open source)

**Metrics**
- Telegraf | Promtail | OpenTelemetry SDKs

**Logging**
- Kafka | ZooKeeper
- Opentelemetry collector

**Tracing**
- Apache Spark
- Jaeger (Traces)

**Visualization & Notification Service**
- Grafana | Prometheus Alert Manager

**P&O Service**
- Ansible | Kubectl

**Fusion Service**
- Prometheus (Metrics)
- Loki (Logs)

**Storage Service**
- Prometheus (Metrics)
- Loki (Logs)
- Jaeger (Traces)

**Measurement Namespace**
- Telegraf | Promtail | OpenTelemetry SDKs

**Observability Namespace**
- Grafana | Prometheus Alert Manager
- Apache Spark
- Kafka | ZooKeeper
- Prometheus (Metrics)
- Loki (Logs)
- Jaeger (Traces)

[https://github.com/AIDA-KAU/Distributed-Observability-Framework.git](https://github.com/AIDA-KAU/Distributed-Observability-Framework.git)
Experimental Setup

- **Hardware:** Desktop-based
- **OS:** Ubuntu 22.04.2 LTS
- **Kernel:** 5.15.0-72-generic
- **Kubernetes version:** v1.26.0
- **Containerd version:** 1.6.21
- **ThingsBoard** edge IoT Platform
- Custom-developed **Simulated** Sensor Data Pipeline
Measurements Overhead Experimentation (Metrics)

- Agents are created as Daemonset
- Measurement interval: 1s, 5s, 10s
- Number of metrics: 90
Fault Detection based on Measured Data (1/2)

• Develop an end-to-end system:
  ▪ Generate a continuous stream of simulated sensor data

<table>
<thead>
<tr>
<th>Sensor</th>
<th>Data</th>
<th>Transmission</th>
</tr>
</thead>
<tbody>
<tr>
<td>Normal</td>
<td>Valid</td>
<td>Regular</td>
</tr>
<tr>
<td>Abnormal</td>
<td>Valid</td>
<td>Irregular</td>
</tr>
<tr>
<td>Mixed</td>
<td>Valid + Invalid</td>
<td>Irregular</td>
</tr>
</tbody>
</table>

Operations Technology (OT)

1. Generate Data
2. Publish
   - a. Any Action
3. Subscribe
4. Consume
5. Store
6. Publish

Information Technology (IT)

Realtime Message Queue

Storage

End

Event Processing Platform

Source

Destination

CPU Usage (ms)

Number of logs

Timestamp

Sensor Data

Transmission

Normal Pods

Abnormal Pod

Mixed Pod

Sensor-1

Sensor-2

Sensor-3

Sensor-4

Sensor-5
Fault Detection based on Monitored Data (2/2)

Error Logs

Connected traces
Fault Detection/Recovery Using Metrics at Edge Nodes

Detect crashed applications inside a container that is reported as operational by Kubernetes
ML PIPELINE

RED HAT RESEARCH DAYS 2023-09-21
Towards Robust ML Systems in Production

- **Robust performance** is essential for **trustworthy AI systems** (according to EU guidelines)
- **DataOps**: end-to-end data processing operations in production
- **ModelOps**: the set of operations that are performed on the learning task of the ML model
- **Automation**: the engine that drives and coordinates the overall operations
Modern ML Systems in Production

Modern ML Systems

- Continuous Integration (CI)
- Continuous Delivery (CD)
- Continuous Training (CT)
- Continuous Monitoring (CM)

Testing and validating code, components, data, and models

Not only deploy a single service, but automatically deploy another service.

Automatically retrain models, when automatically detect changes and performance degradation.

Catching errors in production systems, and monitoring production data.

Experimentation/Development
- Data Scientist/ML Engineering

Continuous Training
- ML Engineering/Data Scientist

Model Deployment
- ML Engineering/DevOps Engineer

Continuous Monitoring
- ML Engineering/DevOps Engineer
Deploying AI at Scale

• Continuity and automation .. Towards continuous everything
• Several Challenges:
  • **Data Quality and Quantity:** Large-scale deployment requires a huge amount of high-quality, labeled data
  • **Model Performance:** AI model degradation
  • **Integration with existing systems:** compatibility and technical difficulties.
  • **Maintenance and Updating:** AI models need to be maintained and updated regularly.

---

**MLOps Levels**

- MLOps level 0
  - Manual process
- MLOps level 1
  - ML pipeline automation
- MLOps level 2
  - Automation CI/CD pipeline
WP3 Architecture

Not many details with the current architectures in the literature, e.g., Google reference Architecture.

Several Software Engineering concepts are missing in the current architectures
MLOps Lifecycle

- **Design**: Project conceptualization and goal-setting.
- **Development**: Model building, training, and evaluation.
- **Deployment**: Model goes live in production.

**ML Lifecycle in MLOps:**

- **Added value**
- **Business Requirements**
- **Key metrics**
- **Data processing**
- **Feature engineering**
- **Experiment tracking**
- **Model training & evaluation**
- **Runtime environments**
- **Microservices architecture**
- **CI/CD pipeline**
- **Monitoring & retraining**
Development to Deployment

• Development Environment: Used for model development and testing.
• Production Environment: Where the ML model operates in real-time, making predictions based on incoming data.
• Once the ML model is developed, we need to move the ML model into the production environment.
CI/CD

- The use of **continuous integration and continuous deployment**
- **Automating Deployment**: of code, including ML models.
- **Series of Steps**: developing, testing, and deploying code, enabling incremental changes and efficient production deployment.
Data-centric problems in Production

- Data verification and training data evolvability
- ML decision making correctness and algorithm testing
- Testing for ML model degradation.
- **Training Data Evolvability** - test the training data against the used model
- **Quality of the data**: Insufficient data, irrelevant features, non-representative training data, overfitting, underfitting, outliers.
Data Drift and Model Degradation

- **Problem Context:** reduce operational costs
- **Task:** prevent *costly* breakdowns
- **Initial Predictive Models:** trained on historical data
- **Machinery degradation -> Drift -> ML model degradation**
Dealing With the Problems in MLOps and DataOps

• Building customisable micro services to deal with the custom data and ML problems
End to End Approach
Self-Adaptive Drift Handling

• ML software to **predict the minimum pressure value** of a pumping event
• The minimum pressure value is predicted every **30 seconds** for up to **3 minutes**
• Evaluate: **Predicted value < pressure threshold**
• Benefit: **Early identification of invalid pumping events**
• Industrial process scalability: Introducing a **new furnace** to the industry
• **Fast integration** in the predictive system
Drift Handling for Self-Adaptive ML in Scalable Industrial Processes

- **Collect —> Adapt —> Deploy —> Monitor —> Decision**
- Shift adaptation: **importance weighting, Kernel Mean Matching (KMM)**
- ML model: **Random Forests(RF) and XGBoost**
- Evaluation metric: **mean absolute percentage error (MAPE)**
Data Quality Scoring

- ML approach
- Score n data points using the pipeline approach
- Train ML regression on the training datasets of size n
- Predict the score of the testing dataset of size l

**DQSOps: Continuous Data Quality Scoring Framework for Data-Driven Applications**
Data Augmentation for Limited Data

[Diagram showing the process of data augmentation with ESR Pressure Data, Initial Pressure and Pump-down Time, and Augmented Samples.]
Detecting and Predicting Faults in Electricity Grid Using Customer Data and Topology

• Simulating the Overall Topology
Tracking Changes

SUBSTATION

T435 Load = 67.39 Status

Consumption 12.73 0 44 10.66

CUSTOMER

C1 C2 C3 C4 Status

Timestamp = t

T435 Load = 67.39

C1 C2 C3 C4

12.73 0 44 10.66

T435 Load = 59.39

C1 C2 C3 C4

10.73 Missing 40 8.66

Timestamp = t1 Timestamp = t2
What is Next?

- Implementing End-to-End Services in Cloud.
- Continuous testing for model rollback
- Seamless deployment in production and robustness
- Continuous mutation testing for data augmentation
- Continuous anomaly detection and QA.
Committed to excellence in distributed systems and communication, security and privacy, and software quality