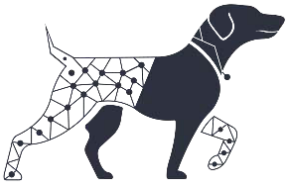

KAIOPS

AIOPS FOR KUBERNETES

White Paper

Introduction

In the fields of information technology and systems management, ITOps describes the methods used to retrieve, analyze, and report data for IT operations. Modern ITOps is closely tied to container orchestration, which provides the automation of much of the operational effort required to run containerized workloads and services on a distributed network. Container orchestration and cluster management involves a wide scope of tasks. These tasks include capacity management, event monitoring, alerting, and remediation.

AIOps is the application of artificial intelligence for IT operations. With AIOps, operations teams can leverage data that would otherwise be intractable. AIOps frameworks use machine learning to optimize IT tasks over one or more objectives. These optimization objectives are expressed as metrics, and can include measures of cost, resource usage, response times, etc.

Use Cases

AIOps is used to tackle many use cases, including capacity management, event monitoring, and incident remediation. Each use case has the end goal of optimizing an IT network on one or more objectives, such as service response time, resource usage, or cost.

Capacity Management

Capacity management involves allocating and delegating resources within an IT network as a continual response to service demand. This is achieved through various tasks like autoscaling and scheduling. Autoscaling is a method used in cloud computing that dynamically adjusts the size of computational resources in a network.

Event Monitoring

Event monitoring encompasses data monitoring and synthesis for the identification and prediction of network events and activities. Specific event monitoring tasks include anomaly detection and event correlation. Anomaly detection is the classification of events, incidents, and traffic patterns to predict anomalies, distinguishing normal (routine/cyclical) from abnormal (threat potential) instances to ensure mission-critical measures are taken preemptively. Event correlation is the process of identifying and classifying relationships between disparate data as an individual or related events.

Incident Remediation

Incident remediation entails alerting and remediation operational tasks, like triggering incident escalation workflows, provisioning alerts and response recommendations for relevant teams and subject matter experts, automating remediation procedures by extracting and inferring incident root causes from alerts and event logs, or launching appropriate ITSM processes.

Security

Security tasks include asset inventory auditing, configuration risk assessment, compliance monitoring, network security scans, identity and access management, data security, and vulnerability/threat detection. AIOps can be used to detect advanced threats, such as cryptojacking, malware-infected instances, lateral movement, and other types of advanced persistent threats (APTs) to protect workloads (virtual machines, containers and serverless deployments) against application-level attacks during runtime.

Implementation

AI as a Service

KAIOps is implemented by deploying an AIaaS (AI as a Service) for each use case. Through the AIaaS architecture, Kgents (KAIOps agents) serve as external consultants to the IT network, by providing action recommendations or forecast predictions that trigger automation workflows. Relevant network data is sourced through open-source services such as cAdvisor or Kubelet, and is ingested by the Kgents for model optimization and inference (e.g. scale resources, predict security threats). The AIaaS framework enables KAIOps to serve multi-cloud environments, where applications or IT networks may be distributed across multiple clusters and/or hosted by multiple cloud-service providers.

Machine Learning

Kgents on machine learning models (e.g.Deep Neural Networks) to optimally perform tasks. For a machine learning model to efficiently learn a task, it is pivotal that the input data provided to the model is pre-processed in a way which preserves and organizes the encoded information. A Kubernetes cluster presents itself as a collection of various types of resources that relate to each other in particular ways. The relationship between two resources depends upon their type. For instance, Deployments and Pods are two types of resources. A Deployment can be configured to manage a particular Pods. This defines a relationship between a Pod and a Deployment.

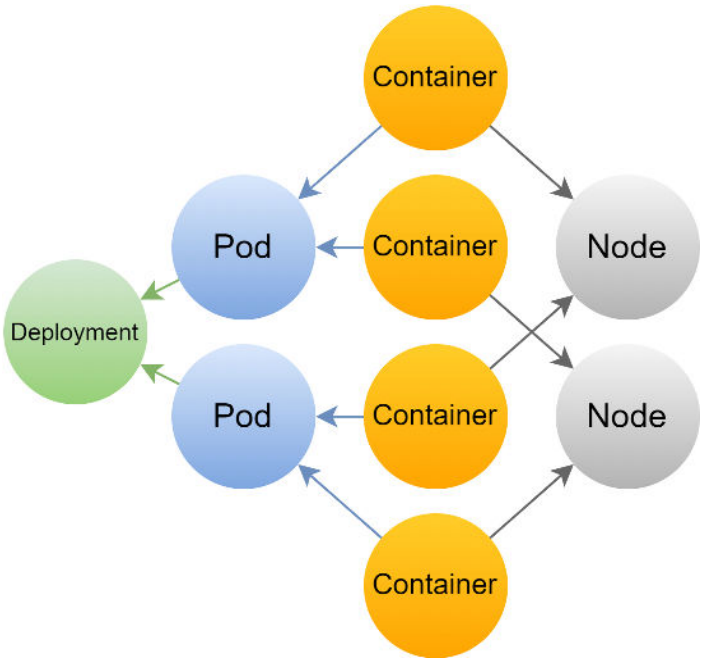

Figure 1: Example Graph Structure for Kubernetes Resources

Because of this structure, a Kubernetes cluster is best represented as an undirected graph of nodes connected by edges, where each node represents a cluster resource, and each edge represents a relationship between two resources, as illustrated in Figure 1. A Graph Neural Network (GNN) model operates on graph data, where nodes and edges are represented as n-dimensional arrays of numbers that encode metadata of the resource and associated telemetry, such as metrics, logs, and traces. A cluster graph is represented as a time-series, with each timestep representing the state of the cluster at that point in time. After the data has been pre-processed, GNN models can be utilized for training and inference.

There are two main types of model outputs, depending on the specific task the Kgent is performing. The first output type is a signal that indicates a particular action needs to be taken via an API operation. In this case, one or more API operations are defined as model responses. Each response encodes an optional signal to execute the corresponding API operation, as well as any related parameters taken by the API. The second type of output is the production of a resource configuration. A GNN model can receive and/or produce outputs that encode serialized data, such as JSON or YAML. Fully connected layers encode ordered collections like lists and tuples, while attention layers encode unordered collections like sets or mappings. Through this style of encoding, models can produce complete resource configurations.

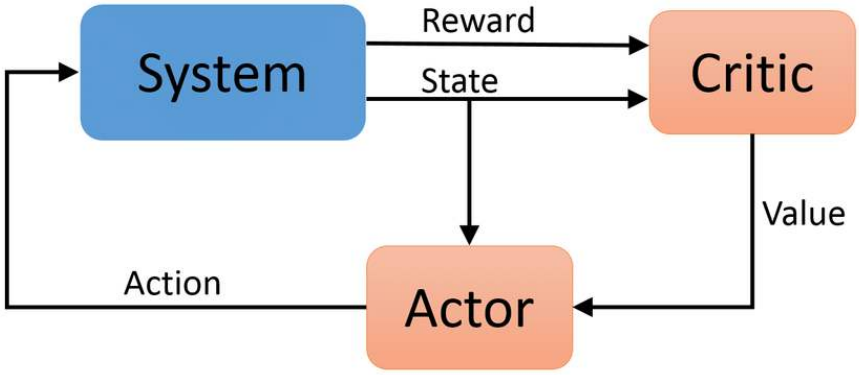

Figure 2: Actor-Critic Reinforcement Learning

Reinforcement Learning

The Deep Reinforcement Learning framework involves using neural network models as universal function approximators to learn value functions that map state-action pairs to their expected future reward given a particular reward function. Actor-Critic methods are a class of Reinforcement Learning frameworks that use an “actor” model to infer optimal actions based upon the current environment state. A separate “critic” model predicts the future reward based upon the actor’s recommendation. The critic learns by observing reward signals from the environment, while the actor learns by optimizing to produce outputs that satisfy the critic, as depicted in Figure 2.

For KAIOps, the reward function is defined as a scalar value representing the combination of all optimization objectives, such as cost, resource usage, etc. To account for non-uniform objective prioritizations, each objective metric is scaled by a weight before they are summed to produce the reward value. A larger weight corresponds to greater importance given to the corresponding objective metric during the model optimization process.

Balancing Optimization Objectives

A key innovation is that the appropriate objective metric weights are learned based on feedback provided by the user. The user is periodically prompted to complete a form, which requires each optimization objective to be assigned a value that represents a slice of a total budget. The provided values represent the user’s relative satisfaction for each optimization objective. An even distribution of the budget over each optimization objective indicates that the user is perfectly satisfied by the current balance of objectives. The goal is to learn the objective metric weights that produce a model optimization which fully satisfies the user.

The Feedback Predictor is a model which learns to predict future human feedback based on given objective weights and environment state. The Objective Modifier is a model which learns to produce objective weights given the current environment states. Gradients from the Feedback Predictor are used to advise the learning of the objective modifier. These auxiliary tasks are learned in parallel with the Reinforcement Learning framework and Graph Neural Network models, resulting in a reward function which optimizes to the users goals based on past feedback and predicted future feedback.

Research and Product Goals

Research and development efforts continue to build upon existing Machine Learning methods to create novel model architectures, data structures, and workflows that are conducive for the container orchestration space. KIOps will soon leverage additional cutting edge AI/ML methods, including the latest in-house and open-source research in the fields of generative modeling, adversarial machine learning, Multimodal Learning, and energy-based models. KAIOps' product goals are to expand the scope of Kgent automation to expand its use cases of capacity management, event monitoring, alerting and remediation, and security, within a fully integrated architecture.