2023 5th International Conference on Electronics and Communication, Network and Computer Technology (ECNCT) | 979-8-3503-2866-0/23/$31.00 ©2023 IEEE | DOI: 10.1109/ECNCT59757.2023.10281148

# Design and Implement of AIOps System Based on Knowledge Graph

Yibing Zhao*, Yongkun Zheng, Haoran Luo, Dengrong Wei, Chun Liu, Kang Chen
China Telecom Guangdong Research Institute
Guangzhou, Guangdong, China
zhaoyb3@chinatelecom.cn

***Abstract*—AIOps (Artificial Intelligence for IT Operations) has emerged as a powerful solution to tackle the challenges involved in operating and maintaining complex microservice systems. Inspired by AIOps, this study proposes a novel approach that leverages a knowledge graph to construct an intelligent operation and maintenance (O&M) system. We first construct a knowledge graph called OpsKG, and then realize a series of functions based on OpsKG in the O&M system, including alarm query, root-cause analysis and location, and alarm prediction. With practical application, the proposed O&M system demonstrates a satisfactory performance, significantly reducing the burden of O&M and improving the efficiency of O&M personnel.**

***Keywords-knowledge graph, AIOps, IT operations, root-cause analysis***

## I. INTRODUCTION

The State R&D Cloud Platform is a research and development (R&D) information infrastructure developed by China Telecom to achieve the digital transformation of enterprise R&D. As a one-stop cloud-based R&D collaborative platform, the platform empowers R&D breakthroughs and supports both internal and external cloud services, making it an important component of China's national cloud strategy.

The State R&D Cloud Platform's capability is built upon an extensive network of business centers, comprising hundreds of microservices. The system architecture's complexity and the sheer volume of business applications present significant challenges for the O&M efforts of the platform. Unfortunately, traditional O&M methods are not suitable for microservice systems. First, there are complex call relationships between microservice applications and devices, which makes it difficult to locate the problem. Second, traditional O&M requires manual handling of a large number of repetitive faults from different sources, resulting in low efficiency. Third, traditional O&M is highly dependent on expert experience, while it is difficult to accumulate and apply operational experience. Moreover, traditional O&M is unable to extract predictive alarms.

To address the limitation of traditional O&M methods, Gartner first proposed the concept of AIOps [1]. AIOps integrates artificial intelligence methods into O&M tasks and leverages the extensive data provided by monitoring components to enhance the quality and dependability of IT services [2, 3]. The application areas of AIOps include anomaly detection [4, 5], failure prediction [6, 7], and root cause analysis [8, 9].

In recent years, many studies have introduced knowledge graphs into AIOps. A knowledge graph is a structured semantic knowledge base composed of entities, relationships, and semantic descriptions. Utilizing its distinct advantages in intelligent analysis and knowledge inference domains [10], the integration of knowledge graphs enables AIOps to access more precise and comprehensive data support, thus improving its overall performance. For example, some studies extract custom relationships and entities from logs to aggregate log events into a knowledge graph [11, 12]. Brandón et al. construct a knowledge graph based on KPI indicators and log data, and then match an anomalous graph representation with a previously happened one to establish its root cause [13]. Saha et al. utilize domain expert experience to construct a knowledge graph to optimize root cause analysis [8]. However, these works do not consider the complex relationships between microservices.

Based on the recent development of AIOps, this study proposes research on an intelligent O&M system based on a knowledge graph. The main objectives of this research are to explore the construction of an O&M knowledge graph and the design and implementation of the system. Our contribution can be summarized as follows:

- Based on the topology relationships in microservice system, we design the ontology elements and construct an O&M knowledge graph, called OpsKG.
- We design the architecture of the intelligent O&M system, taking into consideration practical business requirements.
- We employ root cause analysis algorithms and failure prediction algorithms in the intelligent O&M system to enhance the capabilities of the system.

## II. OPSKG CONSTRUCTION

Compared to other domain knowledge graphs, the construction of OpsKG has natural advantages. The State R&D Cloud Platform employs a microservice architecture, which contains rich software and hardware information, along with the inherent topology structure between devices, and the logical connections of applications. Hence, it is straightforward to construct entities and relationships in the OpsKG. In this study, we adopted the bottom-up method [14] to construct OpsKG. Fig. 1 illustrates the process of constructing the OpsKG.

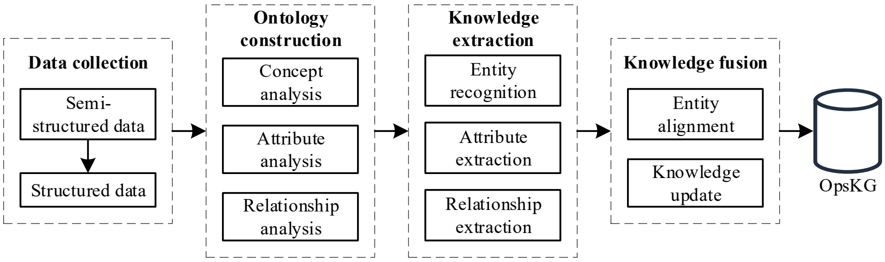

Figure 1. Construction Process of OpsKG

### A. Data Collection

The data source of the OpsKG comprises software data and hardware data extracted from the microservice system. The data mainly includes device data gathered by Prometheus, K8S Service data, as well as Service and topology data in SkyWalking. We collect these semi-structured data through the Prometheus API, K8S scripts, and SkyWalking API, respectively, and convert them into a structured form for storage purposes.

### B. Ontology Construction

Based on the domain experience of O&M experts, we conduct the O&M domain ontology analysis and determined the concepts, attributes, and relationships between concepts.

TABLE I. ONTOLOGY COMPONENTS OF OPSKG

| Ontology components | Type | Description |
|---|---|---|
| Concepts | Center | The business center of deployed application. |
| | Unit | Including application, cluster and domain. |
| | K8S-Pod | Kubernetes Pods. |
| | Device | Kubernetes Nodes. |
| Attributes | Business center attributes | Including center name, namespace, etc. |
| | Application unit attributes | Including unit name, belonging center, SkyWalking trace id, etc. |
| | Pod attributes | Including Pod name, belonging center, IP, etc. |
| | Device attributes | Including device name, belonging center, IP, etc. |
| Relationships | call | The calling relationship between Unit and between Unit and Device. |
| | belong | The subordinate relationship between Unit and Center. |
| | has | The containment relationship between Unit and Device. |
| | deployed_in | The deployment relationship between Unit, K8S-Pod and Device. |
| | provide | The supply relationship between K8S-Pod and Unit. |

Specifically, the concepts comprise various entity types in software and hardware data, including the Center ontology (business center) at the central level, the Unit ontology (application and service unit) at the unit level, and the K8S-Pod and Device ontologies (devices) at the device level. The attributes are the description of each entity, such as name, associated center and IP address. The relationships represent the collections between entities, including *call* relation, *belong* relation (subordinate), *has* relation (containment), *deployed_in* relation (deployment), and *provide* relation (supply). The ontology components of the OpsKG are described in Table 1.

The ontology model of OpsKG is constructed as shown in Figure 2. Units belong to a certain center, and there are call relationships between units as well as between units and devices. Units can be deployed either on a single device or incorporated as part of a cluster unit comprising multiple devices. K8S-Pod is deployed on devices to provide a runtime environment and resource management for units.

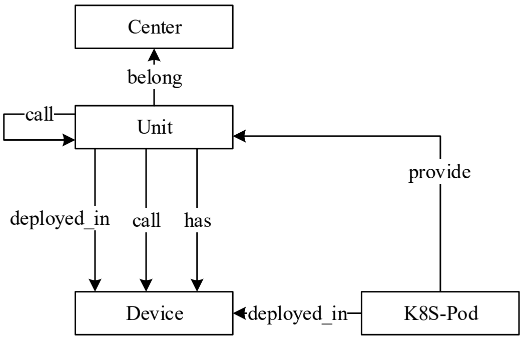

Figure 2. Ontology Model of OpsKG

### C. Knowledge Extraction

In this study, we employ a joint entity-relation extraction approach to simultaneously extract both entities and relationships. After the data collection process, semi-structured software and hardware dada have been converted into structured data, stored in a relational database. Hence, according to the semantic relationships between tables, it is straightforward to convert relational data into "entity-relation-entity" triplets and extract attributes directly based on table fields. For example, K8S endpoints record the access addresses of Pods or virtual machines associated with a Service. Therefore, based on the endpoints in the K8S Service information table, the IP addresses contained in endpoints can be parsed and linked to their respective K8S-Pod or Device entities. This process facilitates the establishment of both a deployed_in and a provide relationship.

### D. Knowledge Fusion

Knowledge fusion refers to the process of integrating data and knowledge from various sources to achieve a more complete, accurate, and consistent understanding. In the case of OpsKG, which utilizes Prometheus metric data, K8S data, and SkyWalking topology data as its data sources, we must remove redundant data from different sources by aligning entities and relationships. Given the standardized naming convention of software and hardware in the State R&D Cloud Platform, we primarily employ rule-based strategies for entity alignment. For instance, the SkyWalking Service data follows the naming convention "deployment.namespace:port", whereas Services derived from Prometheus metrics are named simply as "deployment". Regular expressions can be utilized to identify their deployment and merge them into a single Unit entity.

### *E. Knowledge Storage*

OpsKG is stored in the Neo4j graph database. Neo4j provides a Java Driver to meet the front-end requirements for modifying and querying data by using Cypher statements.

## III. Design of the Intelligent O&M System

### *A. Architecture Design*

The intelligent O&M system mainly consists of three layers: the data storage layer, the knowledge graph construction layer, and the application layer, as shown in Figure 3.

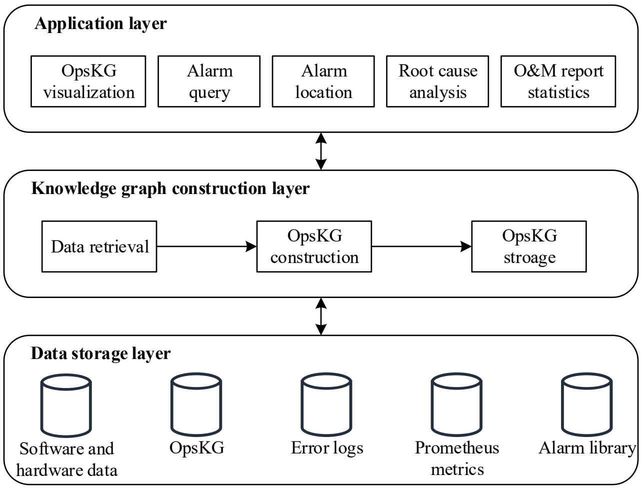

Figure 3. Architecture design of intelligent O&M system

Specific introductions of each layer are as follows.

*1) Data Storage Layer:* This layer stores the O&M data required by the system, including software and hardware data, the OpsKG, error logs, Prometheus metrics and alarm library.

*2) Knowledge Graph Construction Layer:* The specific construction process of OpsKG is described in Section 1. In practical applications, the OpsKG is first initialized based on offline data for the day. As the business evolves and grows, there will be upgrades and replacements of applications. Therefore, this layer will regularly update the OpsKG.

*3) Application Layer:* This layer offers a user-friendly interface, enabling users to easily perform operations like editing OpsKG and querying alarms through the functional modules (See Section 3.2).

### *B. Function Design*

The functional modules of our intelligent O&M system are mainly composed of a knowledge graph module, a real-time alarm module, and an O&M report statistics module.

*1) Knowledge Graph Module:* The Knowledge Graph module visualizes the OpsKG, facilitating user browsing and editing the graph. Users can search for nodes, select connected relationships to display, and interactively perform operations such as collapsing or expanding sub-nodes..

*2) Real-time Alarm Module:* The real-time alarm module presents both real-time alarms and predictive alarms. Real-time alarm data is obtained from error logs and Prometheus metris, whereas the predictive alarms are solely read from the monitoring metrics in Prometheus. This module is responsible for receiving and parsing alarms, which effectively displays the various alarms from multiple perspectives using a combination of charts and tables. Moreover, this module can also perform root cause analysis on the alarms to facilitate quick localization of possible failure reasons.

*a) Real-time Alarm Graph:* Based on the OpsKG, the real-time alarm graph displays real-time and predictive alarms for each node within a 5-minutes timeframe, while also counts the number of alarms of different levels. By clicking on the alarm icon, users can navigate to the alarm tables to examine detailed alarm information for the current node.

*b) Alarm Event Table:* The alarm event table provides a comprehensive representation of alarms within a specific timeframe. It takes the form of a structured list, containing essential details such as node information, alarm name, alarm category, alarm level, and alarm source. Additionally, it includes informative statistics regarding the initial and final occurrence times of alarms, as well as their frequency within the time period. The alarm event table supports user queries by time, business center, unit, and device, as well as queries based on alarm name, alarm category, and alarm level. By selecting a specific alarm entry, users can navigate to the real-time alarm graph and locate the alarm. Besides, users can input the alarm handling method, thus contributing to the accumulation of operational experience within the alarm library.

*3) Predictive Alarm Table:* Similar to the alarm event table, the predictive alarm table presents warning information in a tabular format. By selecting a specific alarm, users can navigate to the alarm graph display for further visualization.

*4) Root Cause Analysis:* In the alarm event table and predictive alarm table, root cause analysis can be conducted on alarms. Our system aggregated the MicroRCA [9] algorithm to realize root cause analysis. We select the top 5 root cause nodes with the highest probabilities, and the subgraph is partitioned and displayed from the OpsKG.

### *C. O&M Report Statistics Module*

O&M system regularly collect and analyze alarm data to generate annual and monthly reports. The reports are structured around four key criteria: business center, device, unit, and alarm library. The analysis of alarm data aims to ascertain the number of nodes experiencing alarms, the percentage of affected nodes, and the frequency of alarms at each level.

## IV. Implementation and Result

The intelligent O&M system has been implemented online. As of June 30, 2023, there are a total of 1781 nodes and 4343 relationships in the OpsKG, which intuitively reveals the complexity of the State R&D Cloud Platform. The specific data is shown in Table 2.

TABLE II. The Number of Entity and Relationship in OpsKG

| Components | Type | Counts |
|---|---|---|
| Entity | Center | 64 |
| | Unit | 765 |
| | K8S-Pod | 633 |
| | Device | 319 |
| Relationships | call | 410 |
| | belong | 765 |
| | has | 1671 |
| | deployed_in | 703 |
| | provide | 794 |

In June 2023, the intelligent O&M system has recorded a total of 3,150,076 alarms, including 6,994 high-level alarms (accounting for 0.22%), 4,068 medium-level alarms (accounting for 0.13%), and 3,139,014 low-level alarms (accounting for 99.65%). It shows that in the production environment of the platform system, high-level alarms are rare and most of the alarms are non-critical. Benefit from the alarm classification data in the alarm library, we help users to identify which faults require immediate attention.

TABLE III. The Spent Time of Several Core Functions

| Module | Function | Spent time |
|---|---|---|
| Knowledge graph module | OpsKG construction | 35.76s |
| Real-time alarm module | Getting real-time alarms in real-time alarm graph | 2.76s |
| | Getting predictive alarms in real-time alarm graph | 0.29s |
| Root cause analysis module | Root cause analysis | 0.75s |

Moreover, we can make some observations regarding the efficiency of several core functions through their spent time, as shown in Table 3. The OpsKG construction takes 35.76s in average, but we run the process during idle hours when users rarely use the system, which has little impact on user experience. In the real-time alarm graph, the average response time for real-time alarms is 2.76s, and for predictive alarms is 0.29s. This is because there are often more real-time alarms than predictive ones, which takes more time to parse. As for root-cause analysis, the average response time is 0.75s, indicating that our system can quickly deduce the root cause from the complex graph structure.

## V. Conclusions

In response to the challenges posed by complex application structures and demanding nature of O&M tasks in the State R&D Platform, this study presents a design for an intelligent O&M system utilizing a knowledge graph. Firstly, an O&M knowledge graph, OpsKG, was constructed based on the microservice system's structure. Then an intelligent O&M system was developed leveraging the OpsKG. The system has been successfully deployed in practical use and has demonstrated its efficacy in assisting O&M personnel with troubleshooting tasks, leading to notable improvements in operational efficiency.

Future optimization directions for the intelligent O&M system may include (1) improve the speed of the OpsKG construction process, and (2) optimize machine learning and artificial intelligence methods to offer more efficient failure prediction and diagnosis.

## References

[1] Andrew Lerner. AIOps Platforms. 2017[2023-04-03]. https://blogs.gartner.com/andrew-lerner/2017/08/09/aiops-platforms/ .

[2] Y. Dang, Q. Lin, and P. Huang, "Aiops: real-world challenges and research innovations," in 2019 IEEE/ACM 41st International Conference on Software Engineering: Companion Proceedings (ICSE-Companion). IEEE, 2019, pp. 4–5.

[3] P. Notaro, J. Cardoso, and M. Gerndt, "A survey of aiops methods for failure management," ACM Transactions on Intelligent Systems and Technology (TIST), vol. 12, no. 6, pp. 1–45, 2021.

[4] H. Ren, B. Xu, Y. Wang, C. Yi, C. Huang, X. Kou, T. Xing, M. Yang, J. Tong, and Q. Zhang, "Time-series anomaly detection service at microsoft," in Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining, 2019, pp. 3009–3017.

[5] S. Nedelkoski, J. Cardoso, and O. Kao, "Anomaly detection from system tracing data using multimodal deep learning," in 2019 IEEE 12th International Conference on Cloud Computing (CLOUD). IEEE, 2019, pp. 179–186.

[6] Y. Li, Z. M. Jiang, H. Li, A. E. Hassan, C. He, R. Huang, Z. Zeng, M. Wang, and P. Chen, "Predicting node failures in an ultra-large-scale cloud computing platform: an aiops solution," ACM Transactions on Software Engineering and Methodology (TOSEM), vol. 29, no. 2, pp. 1–24, 2020.

[7] F. Yu, H. Xu, S. Jian, C. Huang, Y. Wang, and Z. Wu, "Dram failure prediction in large-scale data centers," in 2021 IEEE International Conference on Joint Cloud Computing (JCC). IEEE, 2021, pp. 1–8.

[8] A. Saha and S. C. Hoi, "Mining root cause knowledge from cloud service incident investigations for aiops," in Proceedings of the 44th International Conference on Software Engineering: Software Engineering in Practice, 2022, pp. 197–206.

[9] L. Wu, J. Tordsson, E. Elmroth, and O. Kao, "Microrca: Root cause localization of performance issues in microservices," in NOMS 2020-2020 IEEE/IFIP Network Operations and Management Symposium. IEEE, 2020, pp. 1–9.

[10] S. Ji, S. Pan, E. Cambria, P. Marttinen, and S. Y. Philip, "A survey on knowledge graphs: Representation, acquisition, and applications," IEEE transactions on neural networks and learning systems, vol. 33, no. 2, pp. 494–514, 2021.

[11] F. Wang, A. Bundy, X. Li, R. Zhu, K. Nuamah, L. Xu, S. Mauceri, and J. Z. Pan, "Lekg: a system for constructing knowledge graphs from log extraction," in Proceedings of the 10th International Joint Conference on Knowledge Graphs, 2021, pp. 181–185.

[12] A. Ekelhart, F. J. Ekaputra, and E. Kiesling, "The slogert framework for automated log knowledge graph construction," in European Semantic Web Conference. Springer, 2021, pp. 631–646.

[13] A. Brand´on, M. Sol´e, A. Hu´elamo, D. Solans, M. S. P´erez, and V. Munt´es-Mulero, "Graph-based root cause analysis for service-oriented and microservice architectures," Journal of Systems and Software, vol. 159, p. 110432, 2020.

[14] Z. Zhao, S.-K. Han, and I.-M. So, "Architecture of knowledge graph construction techniques," International Journal of Pure and Applied Mathematics, vol. 118, no. 19, pp. 1869–1883, 2018.