

Intelligent orchestration of database resources in power system

Junjun Hu¹, Bo Yang², Mingjie Yang², Qiong Wang², Runzhen Yan² and Kaiyun Lin^{1,*}

¹State Grid Electric Power Research Institute, Nanjing, China

²State Grid Gansu Electric Power Company Information & Telecommunication, Lanzhou, China

Abstract. With the continuous development of cloud computing technology, the size of database clusters is getting larger and larger. The database resources in existing power systems need to be manually deployed, which puts great pressure on operation and maintenance personnel. Therefore, resource orchestration and automatic deployment are proposed and realized. This method implements a visual orchestration interface through the GoJS framework, establishes a deployment template, uses Spring template to encapsulate task execution scripts, orchestrates database instances, high-availability standby machines, and logical replication at one time, simplifying the deployment process. Practice shows that this method can be effective Reduce deployment time and improve work efficiency.

1 Introduction

U As a pillar industry of the society, the electric power industry plays a huge supporting role in the development of all sectors of the society. Due to the importance and particularity of the industry, the industry has a high demand for technology in terms of stability, so the technology update is relatively slow. In recent years, with the rapid development of information technology, especially cloud computing, cloud computing is widely used in all walks of life due to its high reliability, virtualization, high scalability and other characteristics. With the introduction of cloud computing, the resource scale of power industry grows rapidly every year.

With the expansion of the scale of database resources in the power system, the early cloud resource management platform can hardly meet the needs of daily operation, especially when it comes to large-scale deployment of database clusters. Currently, cloud platform resource deployment and maintenance mainly face three problems:

1. Low delivery efficiency

Traditional network resource pool in allocating resources, not only need to be configured on the cloud platform, also need to configure the physical environment, such as power system resource pool: the main database, the standby database, the server, all the different equipment manufacturers, network requirements are different, different deployment strategy, so mad configuration, serious drag on the overall business process.

2. Operation and maintenance are difficult

Since the operation and maintenance personnel need unified management after resource allocation, but the current cloud management platform of the power system lacks a unified management interface, the operation and maintenance personnel need to consult relevant

personnel to find the maintenance equipment they are responsible for during daily maintenance, and the resource deployment is stiff. Operation and maintenance personnel cannot have a visual understanding of the overall deployment, equipment performance, network load needs manual supervision time-consuming and laborious.

3. Lack of flexibility

Every time you deploy a device, you need to replan the configuration of the device, and you can't make quick changes to devices that have been deployed successfully.

Therefore, in order to facilitate database resource management, shorten the time of database resource deployment, and improve the work efficiency of operation and maintenance personnel, the resource orchestration module based on GoJS and StringTemplate framework on the cloud platform is designed and implemented. The module provides a visual interface for interactive operations, encapsulates scheduling scripts, and generates reusable templates. The separation of real interface and business data improves the efficiency of deployment and reduces the difficulty of operation and maintenance.

2 Research trends

For rapid deployment problem of the resource pool equipment, for Oracle cluster structures, Feng Zhipeng et al. [1] verify the possibility of automated deployment, and execution of the script to design a key; Li Chao et al. [2] proposed a cloudy conditions of fine-grained CADep elastic application automatic deployment framework, using lightweight kernel to realize the core of the deployment mechanism, through hierarchical description and application execution environment, realize the fine-grained management. Zhang Rongqi et al. [3] designed

* Corresponding author: 18017566925@163.com

an automatic deployment mechanism based on Open Stack to support automatic deployment of services at the operating system level and application level, thus reducing the management difficulty. Lai Peiyuan et al. [4] designed an automated deployment architecture based on overlay SDN technology, providing a workable scheme for network automation deployment of telecom operators. Li Liyao et al. [5] proposed an automatic deployment scheme on the OpenStack cloud platform by copying the mirror templates established according to different nodes, and then writing scripts to modify the configuration to achieve deployment. Marica Antonacci et al. [6] used techniques such as Docker and Apache Mesos and standard rules describing application architecture to simplify the process of creating repositories of digital assets using cloud resources.

In terms of visual operation and maintenance, Li Ming et al. [7] realized the one-button management of graphics and data on the graphics service based on the information and communication integration dispatching operation support platform (SG-I6000). Zhao Xiao et al. [8] summarized common visualization technologies and verified the possibility of applying visualization technology to power information operation and maintenance. Ma Yong et al. [9] introduced the graphical modeling of information devices, developed the graphical modeling and monitoring platform of information room based on Scalable Vector Graphics (SVG) technology, and realized the function of monitoring and alerting the running state of database and server system.

Can be found in the present study for rapid deployment and visualization operations combined with the application of less, unable to effectively solve the cloud platform to deploy a new generation of electric power system, the difficulty of low efficiency and operational, thus combines the technique of rapid deployment and operations of visualization technology, adopt GoJS design interactive page, use StringTemplate encapsulation power system resource scheduling script, achieve the function of cloud resource scheduling.

3 An introduction to the resource intelligence orchestration method

This module function is mainly implemented by GoJS and StringTemplate framework. GoJS is responsible for implementing the interactive page of resource marshalling, generating the resource marshalling template, and StringTemplate encapsulating the script of deploying database resources, so as to realize the separation of business logic and presentation logic and improve the deployment efficiency.

3.1. GoJS framework

GoJS is a JavaScript and TypeScript database for building interactive charts, including simple flowcharts and organization charts to highly specific industry charts. GoJS makes it easy to build JS charts that are

originally composed of complex nodes and links by using customizable templates and layouts. GoJS interaction for the user provides many advanced features, such as drag and drop, copy and paste, in text editing, tooltips, context menu, automatic layout, templates, data binding and model, state of affairs, and undo management, palette, an overview of the event handler, command, used for custom operating tool of extensible and customizable animation.

Through the GoJS framework, the visual choreography interface is realized, which enables the operation and maintenance staff to understand the performance of the deployed resources in real time and realize the unified management of resources. The operation and maintenance staff can realize the rapid deployment of resources by dragging the resource icon to the corresponding location of the schedule, without caring about the scheduling script.

Part of the code based on GoJS framework is as follows:

```
{
    "class": "go.GraphLinksModel",
    "linkFromPortIdProperty": "fromPort",
    "linkToPortIdProperty": "toPort",
    "modelData": {
        "position": "-235.50000000000003
        270.60219726562536"
    },
    "nodeDataArray": [
        {
            "key": "racArea",
            "groupType": 1024,
            "titleText": "RAC Cluster",
            "strokeWidth": 2,
            "dash": null,
            "category": "RacGroups",
            "isGroup": "true",
            "isAdd": true,
            "font": "bold 13px sans-serif",
            "loc": "2.977152501692112 -259.5110540699991"
        }
    ],
    "linkDataArray": [
        {
            "from": "racArea",
            "to": "slaveGroup",
            "fromPort": "B",
            "toPort": "T",
            "points": [
                158.494, -39.359962969596324,
                158.494, -29.359962969596324,
                158.494, -3.8692881254231786,
                158.5, -3.8692881254231786,
                158.5, 21.621386718749967,
                158.5, 31.621386718749967
            ],
            "isAdd": true,
            "strokeWidth": 1,
            "dash": [3,2]
        }
    ]
}
```

The generated resource layout page is shown in figure 1:

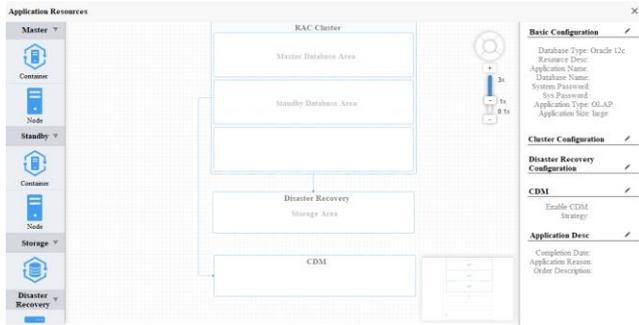


Fig. 1. Resource Layout Page.

The whole resource layout page is divided into three parts, the left is the element menu, the middle is the resource layout, the right is the basic information.

(1)Element menu bar

The element menu bar is equivalent to the color palette, which is composed of containers, nodes and other elements. When generating the orchestration scheme, the effect of the color palette can be presented through the custom grouping template, node template and data model object of GoJS.

(2)Resource layout column

The intermediate resource orchestration mainly implements the functions of initializing the basic diagram, dragging the master database container, providing the overall preview, and saving the data.

The basic chart initializes the JSON to data model object method provided by GoJS, and converts the JSON string to data model object to present the chart effect.

Dragging the master database container triggers a drag-and-drop event of the database container, and adds the master database container to the host database area using the add member method of the master database container.

The overall preview is based on the Overview component provided by GOJS, by specifying the viewer as an icon object and setting the preview to scale with the horizontal position of the content.

The data model object of the graph presented in the canvas area is converted into a JSON string and then saved by using the method of data model object conversion to JSON. The added master database container and node and master database storage container and flashback area, data file node are saved to JSON data.

(3) Basic information bar

The basic information on the right side can display various operations such as node increase and decrease, undo and restore, etc., and read its attribute information through the specified node data object.

3.2 String template framework

StringTemplate is a java-based template engine for generating source code, web pages, E-mail, or other formatted text output, especially for multi-objective code

generators, multi-site look-and-feel, and internationalization and localization. Compared to other engines, it is more flexible and maintainable due to its strict implementation of model-view separation.

When using StringTemplate to encapsulate scheduling scripts, users write scripts to schedule resources according to their own requirements and use the StringTemplate framework to encapsulate scripts to generate templates. This template is easy to extend and modify, and can be reused.

Part of the code based on the StringTemplate framework is as follows:

```
{
  "author": "cloud",
  "description": "createOracleDB",
  "fsJobs": [{
    "author": "cloud",
    "childJobKey": "2",
    "dependJobResult": "0",
    "executorAppName": "commonExecutor",
    "executorBlockStrategy": "DISCARD_LATER",
    "executorFailStrategy": "FAIL_ALARM",
    "executorHandler": "shellJobHandler1",
    "executorRouteStrategy": "ROUND",
    "fsParam": {
      "obs": [{
        "fileFolderName": "oracleRac",
        "cmd": "./step1/createOracleDB.sh
/u01/app/oracle#test12ab#test12ab#dbcloud1,dbcloud2#
u01/app/oracle/product/12.2.0/db#abcd_123abcd#abcd_
123abcd#abcd_123abcd",
        "destMachine": [{"hostName": "dbcloud2", "password": "1
23456", "port": "22", "IP": "192.168.41.188", "user": "root"}]
      }],
      "isBulk": "false",
      "type": "depend"
    },
    "glueType": "BEAN",
    "id": 0,
    "jobDesc": "create oracle master",
    "jobName": "job-create-oracle-master",
    "jobOrderInTask": 1,
    "runStatus": "ready",
    "taskID": 0
  }],
  "id": 0,
  "listenerName": "CreateOracleDBCallback",
  "taskCron": "",
  "taskName": "task-createOracleDB",
  "taskUuid": "",
  "userIdentity": "cloud",
  "filePath": "D:/shell/oracleScripts.zip",
  "dirName": "oracleRac"
}
```

By encapsulating the scheduling script with StringTemplate, the entire automatic deployment process is roughly as follows:

(1) manage the security policy of the operating system of the two nodes respectively.

(2) upload the automatic deployment script to the server

(3) upload the installation package corresponding to the database to the installation directory corresponding to the automatic deployment script of the server

(4) configure the script and database on the server.

(5) authorize the execution of the script by entering the corresponding script command and configuration file address and other properties into the template edited by StringTemplate.

Taking the deployment Oracle RAC[2] script encapsulated by StringTemplate as an example, the implementation of automatic deployment consists of the following 8 scripts:

(1) parmlist.init is the script that initializes the parameters.

(2) the script for start.sh to install the database.

(3) setupssh.sh configuration root node mutual trust mechanism script.

(4) sshusersetup.sh main program file of mutual trust mechanism.

(5) setenv.sh configured the environment variables of the database and created Oracle and grid users and user groups.

(6) installgrid.sh installs the grid script and creates the disk group.

(7) install oracle.sh to install the database script.

(8) Upgrade. Sh to update the patch package script.

The execution process of the script is shown in figure 2:

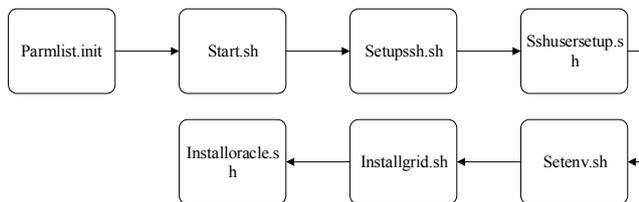


Fig. 2. Execution Process of the Script.

When formally deployed, operations personnel only need to operate through the interface. When you need to deploy other types of databases and other resources, you only need to load the corresponding deployment script on the basis of the original template, thus greatly reducing the difficulty of deployment.

4 Conclusion

Based on the existing mature GoJS framework and StringTemplate framework, this paper combines the database automatic deployment technology to realize the intelligent resource orchestration function. Compared with the existing rapid resource deployment technology, this resource choreography technology allows users to complete resource choreography without worrying about the low-level details of database deployment, and only need to drag and draw a picture on the page. The existing resource orchestration model can be used as a fixed template for later modification when similar requirements are encountered. The application on the new generation of cloud management platform shows that this function module can effectively improve the

deployment efficiency and reduce the operation and maintenance cost.

This work was financially supported by the science and technology projects of headquarters of State Grid Corporation (Research on Key Technologies of High Availability Self-Service Database Cloud Based on Physical Resources).

References

1. Feng Zhipeng, Yu Cong, Li Yan et al. Research and application of rapid deployment method of large enterprise cloud resource pool database [C]. Proceedings of power industry informatization annual conference, 2016
2. Li Chao, Hua Lei, Song Yunkui. Automatic deployment framework of distributed application for cloud computing [J]. Computer technology and development, 2018 (06 2018) : 12-16
3. ZHANG R Q, SHANG Y L, ZHANG S. An automatic deployment mechanism on cloud computing platform [C]//2014 IEEE 6th International Conference on Cloud Computing on Cloud Computing Technology and Science (CloudCom), Dec 15-18, 2014, Singapore. New Jersey: IEEE Press, 2014:511-518
4. Lai Peiyuan, Ma Weimin, Liu Yi, etc. The cloud resource research and implementation of network deployment automation [J]. Journal of telecommunications, science, 2015, 31 (7) : 96-103
5. LI Liyao, ZHAO Shaoka, WANG Ye, et al. Dandelion: Rapid deployment mechanism of cloud platform based on OpenStack [J]. Journal of Computer Applications, 2015, 35(11): 3070-3074
6. Antonacci M, Brigandi A, Caballer M, et al. Digital repository as a service: automatic deployment of an Invenio-based repository using TOSCA orchestration and Apache Mesos [C]//EPJ Web of Conferences. EDP Sciences, 2019, 214: 07023.
7. Li Ming, Xia Yun, Kang Rui, Zhang Wenpeng. The Application of Graphic Service in Electric Power Enterprise Information Monitoring. Electric Power Information and Communication Technology, 2016, 14 (10) : 103-108
8. Zhao Xiao, Li Liang, Yan Wentao, et al. Application of visualization technology in the operation and maintenance of power information system [C]//2013 power industry information conference, 2013
9. Ma Yong, Li Min, Yu Shi, Shen Hongjie. Research on visualization modeling and monitoring platform technology of IT equipment in data center [J]