

Design and Analysis of OpenStack Cloud Smart Factory Platform for Manufacturing Big Data Applications

Dae Jun Ahn and Jongpil Jeong(✉)

Department of Smart Factory Convergence, Sunkyunkwan University, Suwon, Gyeonggi-do 16419, Republic of Korea
{djahn92,jpjeong}@skku.edu

Abstract. In building an open source cloud platform, you can configure the OpenStack cloud architecture to implement services with capabilities for smart factories. In addition, we study the limitations, speed, and security analysis of existing cloud. In addition, we confirm that the data that accumulates when smart factory data is generated becomes unstructured data, and solves the problem with the prototype and various types of open source cloud to overcome this problem. The Smart Factory is composed of users in various environments in connection with all tasks such as distribution, production, and research, as well as the manufacturing environment. We design a smart factory cloud that integrates these environments. When storing files of various types, it is possible to store secure data by encrypting and decrypting various types of files and the like. In addition, for the convenience of cloud users, all data can be stored, edited and modified on any local PC connected to the cloud. Finally, open source-based cloud computing optimized for big data environments for smart factories can be configured.

Keywords: Smart Factory · BigData · OpenSource · OpenStack Cloud · Local cloud disk

1 Introduction

The world is working on various forms of cloud-related research for new types of manufacturing. In addition, the idea of a smart factory that can communicate all the machines in forming a new type of manufacturing industry has emerged. The concept of smart factories is emerging as an extended method of operating factories technologically by integrating ICT (Information Communication Technology) technology into manufacturing industries. At this time, IoT (Internet of Things) is a key factor connecting all the machines. However, in a small-sized company, a smart factory is configured according to a manufacturing environment, and a general company that manages actual data needs to centrally manage all data. At this time, there is a need for a new type of cloud that can be used both by the Smart Factory and the general company.

© Springer Nature Switzerland AG 2019
S. Misra et al. (Eds.): ICCSA 2019, LNCS 11620, pp. 53–61, 2019.
https://doi.org/10.1007/978-3-030-24296-1_6

In order to apply the cloud that can overcome this point, we can build a new type of cloud through this study using open source. By building a cloud of data, which is a core data store for the smart factory, the real-time analysis can be done through the dashboard environment to predict. The objectives and goals of this paper are as follows. Data is exchanged and created through a connection between the actual manufacturing environment and general companies. Most of the documents used at this time are stored in files such as Doc, xlsx, and CSV. In this paper, we propose a method to apply the cloud environment to the smart factory and apply architecture that can be applied to the smart factory environment through researching a new type of cloud. This paper is a total of five sections. Section 1 describes the research background and purpose necessary for carrying out the research and describes the necessary explanations and specific goals for the paper. Section 2 introduces the concepts and broader insensibility that are necessary to construct the cloud of the key core of this paper. Section 3 introduces the data collection process and cloud construction method to implement the network architecture necessary for configuring the smart factory cloud, and to apply a new type of cloud by combining open source. Section 4 shows the Smart Factory environment test bed configuration with cloud component selection using open source. Finally, we discuss the conclusions of the ideas proposed in Sect. 5 and future research projects.

2 Related Work

OpenStack provides virtualization computing technology, data storage technology, and a network where various software can communicate. Using this technology, data used between different machines can be stored in the cloud storage using OpenStack technology. One of the biggest advantages of OpenStack is that it can be deployed as a reassembly to meet the needs of servers. Figure 1 shows the types of cloud platforms and cloud ecosystem used to apply the cloud. OpenStack solutions used in the form of Infrastructure as a Service IaaS (Infrastructure as a Service) have been researched in recent years and continue to be resolved through the open source community. OpenStack is also a reliable cloud networking platform developed by NASA [1]. This will allow you to build a cloud for smart factories that can accommodate these environments. At this time, as shown in Fig. 1, data can be stored using Swift (Object Storage) which can configure storage among the services of OpenStack. The proxy node stores data through object storage (swift), and is stored in a cluster form through the connected storage node, so that smart factory data can be stored and utilized [2]. As the connectivity of IoT comes up, we propose a cloud mechanism to collect sensor data and implement necessary functions in the cloud through Openstack services through architecture implementation. Devices such as various communication modules and open hardware, such as Arduino, enable all objects and equipment in everyday life to communicate at low cost.

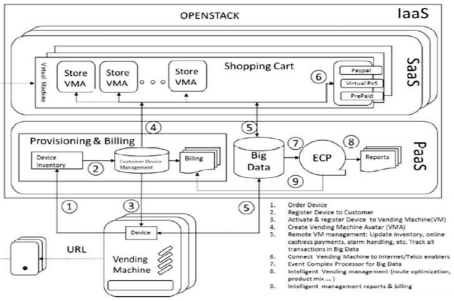

Fig. 1. High Level end to end system design applied to vending machines [3].

In addition, as shown in Fig. 1, it is possible to provide an OpenCart type service which is an open hardware type. It is implemented as an Openstack based integrated platform for this configuration. Once the service is built in the OpenCart format, the VM can be built in the integrated OpenStack and the service can be implemented in conjunction with the device. In other words, the user can purchase the integrated environment in which the user can connect the device and store and serve the big data as a vending machine, receive the service after connection, easily integrate the data between the users or the manufacturing, You can make it possible. In addition, research on the manufacturing cloud for the Smart Factory has created a new concept called CM (Cloud Manufacturing) system through research on data interoperability (Morgan and O'Donnell 2017) as shown in Fig. 2.

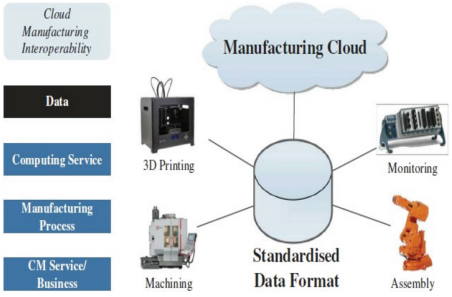

Fig. 2. Data Interoperability [4].

The CM must be able to communicate with the standardized data format through communication, all hardware and software must be able to communicate within the system, and the resources of the cloud should be described and managed and presented to the user. In addition, a data model is designed based on an ontology, and a cloud that can be used as a central data format is constructed. [4]. Each layer is divided into a physical layer, control layer, application layer, etc.

[5]. At this time, the proxy performs both roles of encryption/decryption and can perform this role when a security key is requested to a user receiving data. [6]. Once the role of the cloud server is defined in the Smart Factory, a database of IIoT data generated between the fats is constructed [7] and load balancing for resource allocation through networking as well as proxy nodes is additionally needed as follows. [8] In addition, an integrated architectural design of these elements is needed. In order to apply Smart Factory, we need to implement a unified architecture like [9–11], and it is necessary to study how to store big data from the network type design [12,13].

3 OpenStack Cloud Smart Factory Platform Design Study

The architecture of Keystone is shown in Fig. Token Backed manages Tokens for each user, and Catalog Backed manages URLs accessible to all services of OpenStack. In this paper, we create and manage Token URL that can manage Swift that manages Smart Factory DB and Storage Data in Swift. Policy backed integrates and manages accounts of newly created users and existing registered users, and integrates and manages user authentication through Identity Backed. In addition, the user information is confirmed through the API information, the policy and the authority used in the cloud are given, and the function of storing, deleting and modifying the ID is performed. The algorithms used to implement Swift and Unified Architecture can be represented as follows, and the Open Source type data can be defined in accordance with the operation flow chart. As shown in Fig. 3, there are 7 types of data authentication procedure through Keystone in Smart Factory environment applied OpenStack Cloud. The procedure is as follows. (1) Perform a request to transfer smart factory data. (2) Generation of authentication key through keystone. (3) API call request to Proxy Server via Token. (4) Account data and first data check through Token key. (5) Swift Data Check to provide data to the Web. (6) The Dashboard enables data transmission and confirmation, and data download and upload functions are provided. (7) It becomes possible to manufacture a Smart Factory cloud by

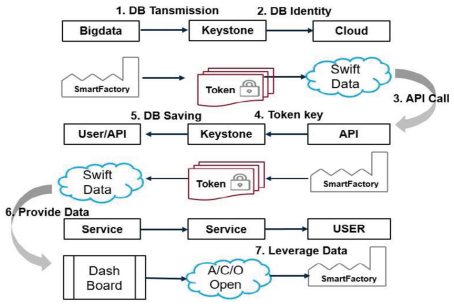

Fig. 3. OpenStack Cloud Keystone authentication procedure diagram

applying A/C/O data. Through the above 7 types, Swift Data can be configured to be able to transmit and store data through Keystone authentication key and various servers. You can also design and demonstrate the algorithm through the source code for these procedures as shown below. As it is based on open source, we can change the source according to the Smart Factory environment through the following source code algorithm, then apply it after encoding, and explain the Python source which is the core of each source.

Data: ID/PW Login
Result: OpenStack Keystone Token;
while *Token* **do**
  Value;
  if *Account/Object/Container* **then**
    Policy-0;
    Timestamp : 1501164819, X-ID : TXD7VEXE607;
    Main = Proxy-server, A/C/O Server;
    Token Identity;
  else
    Hash Token with path Identity;
  end
  Identity Reload;
end

Algorithm 1. OpenStack Token ID/PW Identity

Algorithm 1 shows the source and contents of Swift and Keystone. Through the ID/PW, the keystone applies the token value to acquire and apply various login information to the user. Then check the association with the data in Swift's Account through the Hash value.

Data: OpenStack Server
Result: A/C/O Server, Hash$_p$ath;
while *Server.py* **do**
  Util.py,Ring.py;
  if *Transfer Storage Value* **then**
    Server.py;
    Util.py;
    ring.py;
    ring data with identity;
  else
    Hash Ring with path Identity;
  end
  Server Data Open;
end

Algorithm 2. OpenStack Server data wiht Identity

Algorithm 2 shows the source and contents of Server and Storage data. This procedure is transmitted through the Swift Proxy Server, and the information is

checked on the Main Servers to prepare to transmit the data. Through the API, the Proxy receives input to provide the data. After this procedure is performed, the data is retrieved according to the Token value in the next Fig. 4. When the token is confirmed, it checks the path through the hash value and provides data corresponding to the post-operation state of the source which confirms whether the cloud user is proper policy. At this point, Ring performs the role of transporting the list in A/C/O Swift Storage. Algorithms can be used to identify and modify various key sources and apply them to smart factories according to the situation.

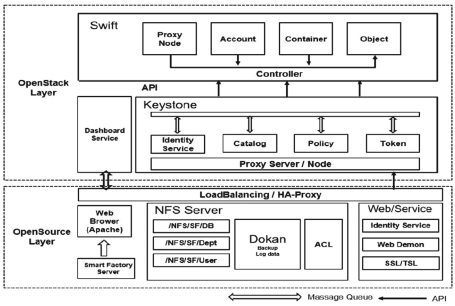

Fig. 4. Openstack base Open Cloud Architecture

In OpenStack Layer, Proxy Node and A/C/O, which are the components of Swift mentioned above, call API through the controller and provide data. OpenSource Layer should build an open source environment to build optimized services for OpenStack Layer resources and the web. The NFS server provides this, and when requesting data with the data path, it is possible to request a key for receiving data through the web service environment through Dokan. In addition, HA-Proxy, in LoadBalancing, should be used to block the load on the server and to configure the server for smooth interoperation between the client and the server. By selecting components according to the environment of the server, a fast and encrypted network environment can be constructed in accordance with the smart factory environment to utilize the services.

4 Designed Big Data Application Cloud TEST BED and ANALYSIS

The plant complex is composed of dozens of sensors for manufacturing automobile parts. Through the technology research institute, integrated management of manufacturing environment through CAD, ERP and MES systems is being carried out. In order to build a smart factory environment by integrating the data of the Smart Factory cloud environment, there is an example of data loss, which

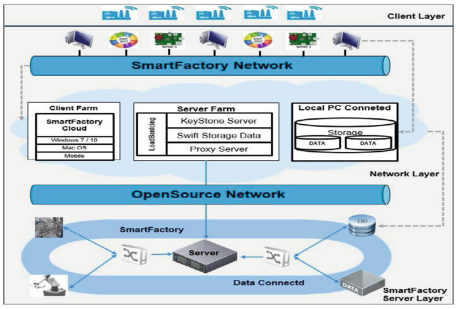

Fig. 5. Smart Factory cloud integration diagram

is not possible to centralize the integrated management data of the manufacturing environment. The Client Layer has completed the router connection for connecting each client user and the manufacturing factory to operate the Smart Factory and the firewall settings for allowing the server IP access shown in Fig. 5. In the Network Layer, we configured an environment that can connect with the local PC, ERP, and manufacturing environment used by the client in order to verify the data through keystone authentication. This paper evaluates and compares the performance of smart factory clouds using the following formulas and Tables 1.

Table 1. Smart Factory cloud speed and measurement related table.

Devision	Explanation
W	Workfow Speed
C	Server Load
Idle	Authentication speed of authentication server
D	Check the amount of data capacity
N	Network speed
A	API Call Check Speed
P	Proxy Node Count (EA)

Equation (1) is a formula for calculating the work speed of Workflow, and C is a formula for checking the load on the server.

$$C = C_w + C_n \tag{1}$$

Equation (2) is a formula for calculating the operation speed of the section that is authenticated in KeyStone Sever. It can check the network speed amount by checking the token value generation interval through Idle.

$$Idle = A_n + P_w \tag{2}$$

By calculating W, it is possible to check the time taken in the interval to transmit and store the final data after the whole process of the cloud is finished.

$$W = N \times Idle_n + C_n + P_w \tag{3}$$

In Eq. (3), the network speed is related to the speed of authenticating the Keystone Token. Since the load of the server is allocated according to the generation of the proxy node, the entire Workflow can be calculated by the formula as shown in the formula. The Cloud Disk environment for the Smart Factory is constructed to enhance user's convenience and real-time data from the factory is converted to excel data and Text file from parts such as quality management and production management, And additional features allow the stored data to be transferred to the analytics environment and dashboard for the Smart Factory.

5 Conclusion

Smart Factory A new type of open source-based cloud computing architecture for big data applications will be used to collect, analyze and utilize the data used in smart factory operations. Unlike conventional cloud computing, a computing device can be located at the edge of a device close to a terminal device rather than being located in a distant center. With the emergence of a new data storage method, the amount of information is increasing exponentially. Today, it is a computing method that attempts to jump to a new level by abandoning existing data processing methods. Research on cloud systems has been increasing among various processing methods for processing data generated in smart factory operations. In future research, we will build a tested environment by constructing an architecture that integrates various types of cloud proposed in this paper. In addition, through the study of the open source base which all users can develop, it is aimed to construct various network platforms that can be used in the Smart Factory and to configure the network to connect various types of Smart Factory.

Acknowledgment. This research was supported by the MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2019-2018-0-01417) supervised by the IITP (Institute for Information & communications Technology Promotion). This work has supported by the Gyeonggi Techno Park grant funded by the Gyeonggi-Do government (No. Y181802).

References

1. Sefraoui, O., Aissaoui, M., Eleuldj, M.: OpenStack: toward an open-source solution for cloud computing. Int. J. Comput. Appl. **55**, 38–42 (2012)
2. Ruan, M., et al.: On the synchronization bottleneck of OpenStack Swift-like cloud storage systems. IEEE Trans. Parallel Distrib. Syst. **29**, 2059–2074 (2018)
3. Solano, A., Dormido, R., Duro, N., Sanchez, J.M.: A self-provisioning mechanism in OpenStack for IoT devices. Sensors (Basel) **16**, 1306 (2016)

4. Wang, X.V., Wang, L., Gördes, R.: Interoperability in cloud manufacturing: a case study on private cloud structure for SMEs. Int. J. Comput. Integr. Manuf., 1–12 (2017)
5. Wan, J., Tang, S., Shu, Z., Li, D., Wang, S.: Software-defined industrial internet of things in the context of industry 4.0. IEEE Sens. J. **16**, 7373–7380 (2016)
6. Chekired, D.A., Khoukhi, L., Mouftah, H.T.: Industrial IoT data scheduling based on hierarchical fog computing: a key for enabling smart factory. IEEE Trans. Ind. Inform. **14**, 4590–4602 (2018)
7. Sookhak, M., Yu, F.R., Khan, M.K., Xiang, Y., Buyya, R.: Attribute-based data access control in mobile cloud computing: taxonomy and open issues. Future Gener. Comput. Syst. **72**, 273–287 (2017)
8. Fu, J.-S., Liu, Y., Zhang, Z.-J., Chao, H.-C., Bhargava, B.K.: Secure data storage and searching for industrial IoT by integrating fog computing and cloud computing. IEEE Trans. Ind. Inform. **14**, 4519–4528 (2018)
9. Pereira, R.I.S., Dupont, I.M., Carvalho, P.C.M., Jucab, S.C.S.: IoT embedded Linux system based on Raspberry Pi applied to real-time cloud monitoring of a decentralized photovoltaic plant. Measurement **114**, 286–297 (2018)
10. Li, D., Tang, H., Wang, S., Liu, C.: A big data enabled load-balancing control for smart manufacturing of Industry 4.0. Cluster Comput. **20**, 1855–1864 (2017)
11. Ahn, D.J., Jeong, J.: A PMIPv6-based user mobility pattern scheme for SDN-defined smart factory networking. Procedia Comput. Sci. **134**, 235–242 (2018)
12. Ahn, D.J., Jeong, J., Lee, S.: A novel cloud-fog computing network architecture for big-data applications in smart factory environments. In: Gervasi, O., et al. (eds.) ICCSA 2018. LNCS, vol. 10964, pp. 520–530. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-95174-4_41
13. Chi, Y., Li, G., Chen, Y., Fan, X.: Design and implementation of OpenStack cloud platform identity management scheme. In: 2018 International Conference on Computer, pp. 1–5(2018)