THE CASE OF CLOUD DESIGNING AND DEVELOPMENT

N.Y. Samokhin[1,a], P.V. Fedchenkov[1], O.I. Lazo[1], A.Y. Shevel[1,2], S.E. Khoruzhnikov[1], A.V. Shvetsov[2], O.L. Sadov[1], A.A. Oreshkin[2], A.V. Naikov[1]

[1] *ITMO University, 49 Kronverkskiy pr., St.Petersburg, 197101, Russia*

[2] *National Research Centre "Kurchatov Institute" Petersburg Nuclear Physics Institute, 1 mkr. Orlova roshcha, Gatchina, 188300, Russia*

E-mail: [a] samon@itmo.ru

The designing and development of a computing clouds is complex process where numerous factors have to be taken into account. For example, size of planned cloud and potential growth, hardware/software platforms, flexible architecture, security, ease of maintenance. Computing cloud is quite often consisted of several data centers (DC). The DC is considered to be a group of hardware and/or virtual servers which is dedicated to run the user virtual machines (VM) and/or storage servers. Each pair of DCs may be interconnected by one or more virtual data transfer links. To manage such cloud to form "Infrastructure as a Service" (IaaS) a distributed operating management system (DOMS) is needed. The proposed architecture for DOMS is a set of software agents. Important advantages of such approach are flexibility, horizontal scalability, independent development/maintenance of any agent. The specially developed protocol to send and receive requests between agents is also discussed. Due to geographical distribution, the requirements for system stability in terms of hardware and software malfunctions are high. Proposed DOMS architecture is addressed operating stability as well. Observation of prototype consisting of several DCs in ~100Km distance from each other and practical results were presented. Potential application fields where this development might be used is also discussed.

Keywords: SDN, NFV, data center, software agent, quantum key distribution system, SDS, OpenStack, Ceph

Nikita Samokhin, Petr Fedchenkov, Oleg Lazo, Andrey Shevel, Sergey Khoruzhnikov, Alexey Shvetsov, Oleg Sadov, Anatoly Oreshkin, Alexey Naikov

1. Introduction

The specific demands for computing power are often exceeds abilities of one data centre (DC), which leads to various types of DC clustering implementation. Currently, in order to ensure efficient use of the computing resources of individual DCs, approaches are being implemented based on DCs combination, such as Grid computing and Cloud computing. It eventually fulfilles the requirement of the redistributing available capacities flexibility. One example is the largest grid-type system - WorldLHC Computing Grid [1], which is used to process experimental data in basic research on the properties of matter. Other examples are the well-known cloud systems Amazon, Google, Azure, which serve a large number of consumers. For example, Azure has about five hundred million registered users.

Of great importance is the geographical distribution of data centres [2], which avoids stopping services or data loss during technological accidents, natural disasters (floods, earthquakes, hurricanes, lightning strikes, tsunamis, etc.) and social incidents (terrorist attacks, civil unrest, etc). It is stressed out that the number of DCs is growing each year due to growing of data volume, so it is important to consider using the geographically distribution approach as it makes computing power more flexible by resources aggregation and sharing.

Cloud computing offers tremendous potential benefits in agility, resiliency, and economy. Moreover, when using IaaS cloud model, it is possible to provide various type of infrastructures, including Grid, therefore it is often preferable. Thus, the basic requirements for geographically distributed IaaS could be these:

- the relative simplicity of the redistribution of resources between local data centres (computing power, data storages, transmission channels), geographically remote from each other;
- security/reliability of data transmission and long-term data storage;
- gradual (staged) degradation in case of failure of hardware and/or software components as failures of individual components should not lead to a shutdown of the whole system.

There are important questions about reasons of implementing of specific cloud system, e.g. how clients might avoid being locked in specific cloud system? Besides, there are always several issues about data storage and transmission safety guarantee, i.e. how can one store his data in data storage for many years and how to provide high-level security for data transmission between geographically distributed DCs? All these questions are being considered in the project presented in this paper.

The main goal of the project is the designing and development computing infrastructure, which permits to integrate the resources of Geographically Distributed DCs to form cloud Infrastructure as a Service (IaaS) and attempt to resolve the issues mentioned above.

2. The constituent elements and control system architecture

An analysis of publications devoted to architectural and technical solutions that are used to create distributed computer infrastructure management systems, for example [3], shows that the most promising ones for creating an integrated management system (IMS) of scalable geographically distributed data centres (SGDD) could be these:

- OS Linux
- Software Defined Network (SDN) and Network Functions Virtualization (NFV) [4, 5]
- Software Defined Storage (SDS) [6, 7]
- Infrastructure as Code (IC) [8]
- Agent-based approach (microservices) [9]
- Quantum Key Distribution (QKD) for data coding keys [10]

The choice of the following technical solutions was determined by their modernity and rich component developers experience:

- OS NauLinux (RHEL clone)
- Ryu SDN controller
- CEPH as storage backend
- Openstack for virtual resources control [11]
- SaltStack for automation, remote task execution and configuration management
- Zabbix as central monitoring unit
- Grafana as data visualization tool

The components listed above are considered as plain tools only that can help to implement certain approaches to the design of the entire system. Among other things, it is important to avoid any dependence on specific software for IMS. The following approaches were used to create the prerequisites for such independence in the design of the SGDD control system:

- the architectural solution must be implemented on the basis of software-configurable infrastructure technologies;
- software components of the system should be implemented using opensource software only, which is configured for specific functionality;
- system components must be created as software agents that run in an operational-isolated environment, such as virtual machines or containers;
- to protect the communication lines between the data centres, quantum communication technology at the side frequencies must be applied.

Based on the mentioned approaches, the IMS of the SGDD was created, the architecture of which is shown in Fig. 1.

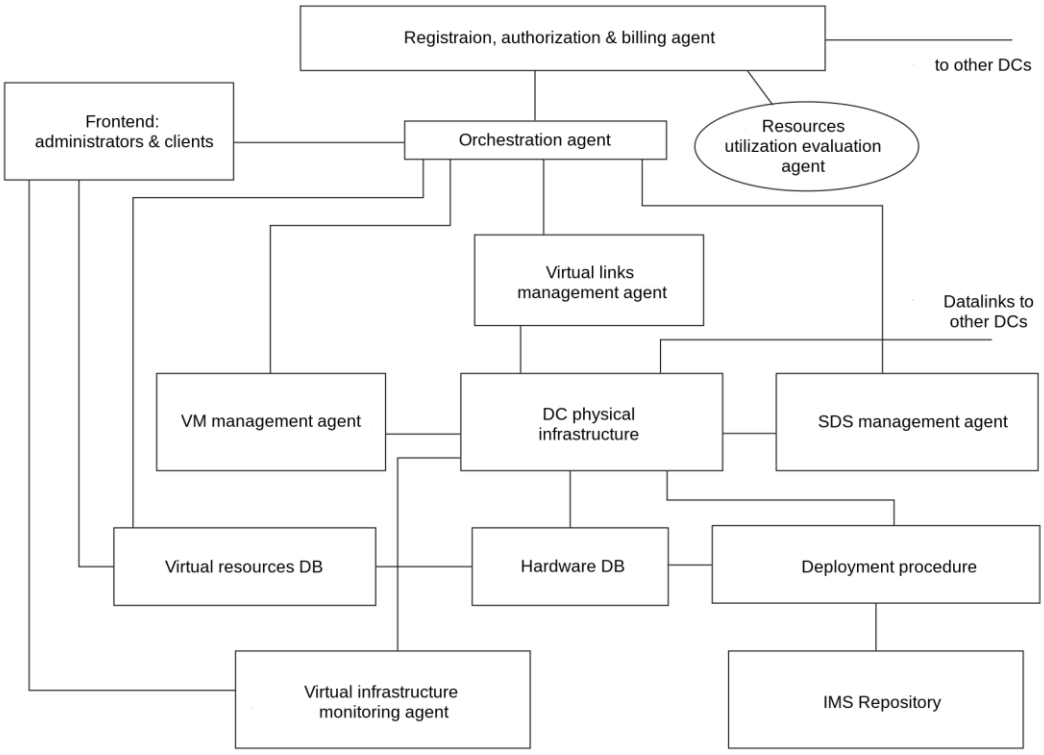

Figure 1. Integrated management software architecture for a scalable, geographically distributed data centre for one local data centre

The interaction between the software agents is implemented through a specially developed protocol, which includes organizing request queues using the distributed RabbitMQ message queuing

system between agents and storing all requests in a database, which came up for the reasons mentioned above and was examined in detail in [12].

An important element of the system that combines all the data centres within the SGDD is the Registry directory (RD), inside which there are subdirectories describing each individual data centre, as well as data channels between the data centres. The RD is automatically replicated to all data centres within the SGDD. Access to the data in the RD is via https, the data itself in the directory is stored in encoded form. Thus, access to the data in the register directory is possible only for authorized software agents.

All software agents of SGDD are launched automatically in the process of loading hardware servers. Each software agent uses configuration parameters from its local RD which he inherited from the main RD via secured channel. Since any change of the SGDD system is reflected in RD, no changes in software agents are required, which is considered to be convenient. The list of IMS agents with their functions is presented in Table 1.

Table 1. The IMS components/agents and their functions

Agent	Main functions	Operation/usage
Registration, authorization & billing agent	Perfrom registration of clients, handling their credentials and all the billing operations. One instance in whole SGDD.	The only agent which decides on any configuration change. Directly connected with orchestration agent. All actions are performed with special web-portals – analytic portal, administrators and clients portals, configuration calculator portal.
Orchestration agent	Handling cloud orchestration requests from Registration, authorization & billing agent. At least one instance per DC and one top level instance per SGDD.	So-called «agent in the middle», receives requests (or transaction with several ones) and forwards them to the specific agent (VM, SDS, virtual links). Orchestration agent is a core agent of IMS.
VM management agent	Creation and maintaining of VMs via Openstack. At least one instance per DC.	Performing all operations on VMs together with orchestration agent and virtual objects database.
SDS management agent	Creation and maintaining of SDSs via CEPH. At least one instance per DC.	Performing all operations on SDSs together with orchestration agent and virtual objects database. Working with various types of storage (block, S3, etc) and SLA (encoding, replicas number, etc).
Virtual links management agent	Creation and maintaining of virtual links via Ryu and Open vSwitch. At least one instance per DC.	Providing virtual links inside single DC or between two DCs. SDN and NFV were used to provide the required SLA: compression, erasure coding, encryption using a quantum key distribution system, parallel communication channels.
Virtual infrastructure monitoring agent	Monitoring of virtual resources, providing statistical data for billing. At least one instance per DC.	Monitoring of virtual CPU and storage, collecting monitoring history for billing purposes, alert system. All alerts must be sent to orchestration agent in order to start damage evaluation procedure.
Physical infrastructure monitoring agent	Monitoring of power supply, climate control and fire extinguishing systems, DC invasion sensors, video cameras for video surveillance, etc. One instance per DC.	Monitoring, security issues, alert system. All alerts must be sent to orchestration agent in order to start damage evaluation procedure.

It is also necessary to provide special components for agent to operate properly. There are three main components for such issue:

- Database (DB). Since databases are one of the important components of IMSs, fault tolerance technology based on configuration of high availability architecture (master-master mode, where two masters are located on separate physical servers) is used to increase the reliability of operation [13]. The main components are hardware database and virtual objects database.

- Repository. It contains all software components (both specially designed and borrowed from external sources) that are used in the IMS. For any deployments or upgrades to the IMS, software components are used only from this repository. For the placement of all components in the repository, developers or the administrators of the support of the IMS are responsible at the initial stage. In other words, the repository should be the only source of software components of the IMS. There are several repositories: one production repository in each DC and one-two preproduction repositories. When any package has been added or upgraded and tested in preproduction repository all production repositories are synching with preproduction repository. Each repository is divided by subrepositories. Each subrepository can be developed independently from other subrepositories. Suggested repository architecture permits to build up virtual machines or/and containers to eliminate conflicts between different libraries, packages, etc. The evaluation of new features (technology preview) becomes much easier with such the approach.
- Semi-automated deployment procedure consists of two stages. The first one is installation planning where the administrator must allocate all IMS agents throughout hardware servers. Some services like DB needs to be allocated at least on two separate servers for creating an HA cluster. It is necessary to provide information about storage allocation for agents for storing logs, big databases and all configuration parameters for DC network. Afterwards, a special image of virtual machine is used for deployment process. It is used for the OS installation, building up the hardware database, preparing Registry directory and IMS components installation afterwards, one by one in according to the specification prepared in first stage. The deployment of the Registration, authorization & billing agent is implemented as completely separate step.

The significant part of the data transfer functionality is implemented as virtual network functions to meet SLA. Among SLAs for virtual data lines, there are requirements for encoding the transmitted data, including the requirement to use the quantum distribution of encoding keys, which is based on the technology of quantum communications at the side frequencies of modulated radiation in this paper [13, 14]. It was chosen due to high performance characteristics that are superior to analogues [15]. The physical principles of its functioning are described in [10], and the scheme of its implementation in the existing telecommunication network is presented in [15] with experimental confirmation of the reliability and long-term stability of the proposed solution.

When individual hardware component of the SGDD fails, the orchestration agent starts a damage assessment procedure, the results of which are forwarded to the system administrator through GLPI, a system for handling claims and incidents. Built-in inventory system with the FusionInventory plugin allows tracking the history of each software change and hardware state. Using the knowledge base and information received, administrator will be able to choose some options for minimizing damage.

3. Experiments

Experimental studies of IMS were carried out on a prototype of a scalable geographically distributed data centre built in conjunction with the SMARTS company (Russia, Samara). The layout diagram is shown in Fig. 2.

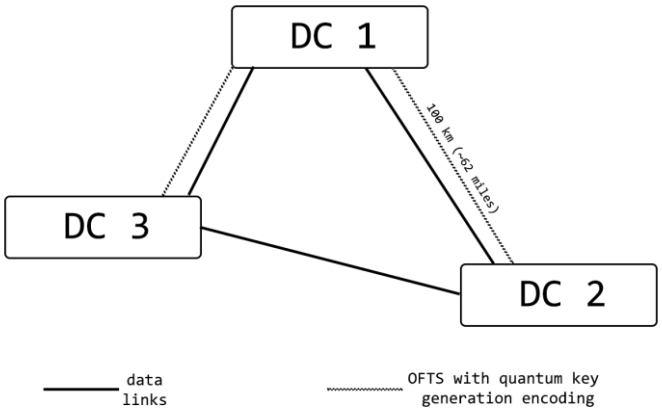

Figure 2. Layout diagram of the geographically distributed data processing and storage centre for experimental research (OFTS - optical fiber transmission system)

DCs (3 microDCs) are combined into a single system with a fiber-optic cable, one of the fibers is used to transmit quantum keys. Maximum distance between DCs is 100 km (~62 miles). Every DC is provided with the same hardware, which is:

- HPE DL380 Gen10 8LFF CTO servers with OS NauLinux for VMs and SDSs (2 x Intel Zeon Gold CPU, 128 GB RAM, 30 TB storage, 6 network interfaces);
- HP FF 5700-32XGT-8XG-2QSFP+ network switches with OpenFlow protocol support;
- Special quantum key distribution units.

There were dozens of tests carried out, such as virtual objects creation and deletion tests, read/write tests for VMs and SDSs, data links with OpenFlow rules tests, etc. All of those were successfully completed and were performed on the basis of the test patterns developed within the project. That showed the reliable performance of the adopted technical solutions and created software.

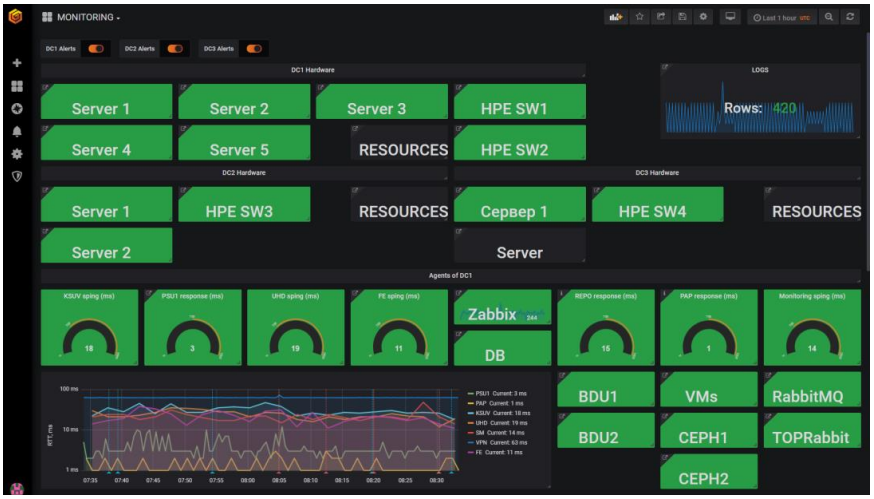

Figure 3. Grafana panel for virtual infrastructure monitoring agent

In the process of passing the test, the state of the agents of the system and equipment was monitored by displaying the data entering the Zabbix on the Grafana panel (Fig. 3). Monitoring agents of the system was carried out by sending a special message according to the developed protocol in order to determine the response time of the agent to the request. In this case, the events of both disconnecting one of the agents and disconnecting the data center were investigated.

4. Conclusion

A description is given of a control system for a geographically distributed DC created using modern architectural approaches and technical means. Particular attention is paid to the flexibility of rebuilding the system to new tasks, efficient use of resources, reliable operation in the event of failure of system components and protecting information from unauthorized access. The use of IMS when conducting experimental research on the SGDD layout confirmed the effectiveness of the main decisions made during the development. Implemented according to the Infrastructure as a Service model, the hardware-software complex demonstrates wide possibilities for creating cloud infrastructures and using it for business purposes. Furthermore, such DC clustering model could provide a solution for several issues concerning specific demands for computing power for a single DC. Thus, such model does make computing power more flexible in terms of aggregation of DC resources.

The main advantages of developed solution are:

- Opportunities for the rapid reallocation of resources of a geographically distributed DC in terms of computing power, data flows, and engineering infrastructure with IMS.
- IMS might use not only bare metal servers but virtual resources from existing clouds. It gives a lot of flexibility because someone can use virtual resources from several existing clouds.
- With minor corrections IMS start use different underlying components, e.g. vmware instead Openstack.
- Horizontal scalability by starting several instances of same type agents in DC.
- A high degree of autonomy of the functioning of the components of a geographically distributed DC due to the architecture based on software agents running in operationally independent environments.
- Own software repository containing the Linux distribution and all necessary software packages (including source codes), which ensures the autonomy of the system and preservation of its operability even in the case of network isolation of DC.
- Secure medium between nodes with the quantum technology of sending encoding keys in combination with the technology of SDN.
- Platform for long-term storage and transmission of big data, which is formed by a combination of software components used in the system.
- Unique semi-automated deployment procedure of IMS for a single DC. Such procedure was designed for this project, so it has no resemblance with other well-known auto deployment procedures. Such procedure is considered to be convenient when one has to add some extra hardware into the existing and operating DC.

5. Acknowledgement

The research has been carried out with the financial support of the Ministry of Science and Higher Education of the Russian Federation under grant agreement No.03.G25.31.0229.

References

[1] Smirnova O. Current Grid operation and future role of the Grid. Journal of Physics: Conference Series, vol. 396, no. 4, pp. 042055. doi:10.1088/1742-6596/396/4/042055

[2] Siqi J., Baochun L. Wide area analytics for geographically distributed datacentres. Available at: https://ieeexplore.ieee.org/abstract/document/7442496 (accessed: 06.10.2019). doi: 10.1109/TST.2016.7442496

[3] Bleikertz S., Kurmus A., Nagy Z.A., Schunter M. Secure Cloud Maintenance: Protecting workloads against insider attacks. ASIACCS '12 Proc. of the 7th ACM Symposium on Information, Computer and Communications Security, 2012, pp. 83–84. doi:10.1145/2414456.2414505

[4] Fedchenkov P.V., Khoruzhnikov S.E., Samokhin N.Y., Shevel A.Y. The designing of cloud infrastructure consisting of geographically distributed data centres. Proc. of the VIII International Conference «Distributed Computing and Grid-technologies in Science and Education» (GRID 2018), Dubna, Moscow region, Russia, 2018, pp. 32–36. Available at: http://ceur-ws.org/Vol-2267/32-36-paper-5.pdf (accessed: 06.10.2019)

[5] Cox J.H., JR., Chung J., Donovan S., Ivey J., Clark R.J., Riley G., Owen H.L. Advancing Software-Defined Networks: A Survey. IEEE Access, 2017, vol. 5, pp. 25487–25526. doi: 10.1109/ACCESS.2017.2762291

[6] Carlson M., Yoder A., Schoeb L., Deel D., Pratt C., Lionetti C., Voigt D. Software Defined Storage. Available at: https://www.snia.org/sites/default/files/

SNIA_Software_Defined_Storage_%20White_Paper_v1.pdf (accessed: 02.10.2019)

[7] The State of Software-Defined Storage, Hyperconverged and Cloud Storage. Sixth annual market survey. Available at: https://www.datacore.com/document/state-of-sds-hci-cloud-storage-sixth-annual (accessed: 05.10.2019)

[8] Morris K. Infrastructure as Code: Managing Servers in the Cloud. O'Reilly Media, 2016, 362 p.

[9] Mohamed Galal Hafez, Mohamed Shaheen Elgamel. Agent-Based Cloud Computing: A Survey. IEEE 4th International Conference on Future Internet of Things and Cloud (FiCloud), 2016. doi: 10.1109/FiCloud.2016.48

[10] Gleim A.V., Egorov V.I., Nazarov Y.V., Smirnov S.V., Chistyakov V.V., Bannik O.I., Anisimov A.A., Kynev S.M., Ivanova A.E., Collins R.J., Kozlov S.A., Buller G.S. Secure polarization-independent subcarrier quantum key distribution in optical fiber channel using BB84 protocol with a strong reference. Optics express, 2016, vol. 24, no. 3, pp. 2619–2633. doi: 10.1364/OE.24.002619

[11] Barkat, Amine; Diniz dos Santos, Alysson; Ikken, Sonia. Open Source Solutions for Building IaaS Clouds. SCALABLE COMPUTING. PRACTICE AND EXPERIENCE. - ISSN 1895-1767. - 16:2(2015), pp. 187-204

[12] Samokhin N.Yu., Oreshkin A.A., Suprun A.S. Implementation of agent interaction protocol within cloud infrastructure in geographically distributed data centres. Scientific and Technical Journal of Information Technologies, Mechanics and Optics, 2019, vol. 19, no. 6, pp. 1086–1093 (in Russian). doi: 10.17586/2226-1494-2019-19-6-1086-1093

[13] Amir Taherkordi, Feroz Zahid, Yiannis Yerginadis, Geir Horn. Future Cloud Systems Design: Challenges and Research Directions. IEEE Access, vol. 6, 2018. doi: 10.1109/ACCESS.2018.2883149

[14] M. H. Ghahramani, MengChu Zhou, Chi Tin Hon. Toward Cloud Computing QoS Architecture: Analysis of Cloud Systems and Cloud Services. IEEE/CAA JOURNAL OF AUTOMATICA SINICA, VOL. 4, NO. 1, JANUARY 2017. doi: 10.1109/JAS.2017.7510313

[15] Gleim, A.V., Chistyakov V.V., Bannik O.I., Egorov V.I., Buldakov N.V., Vasilev A.B., Gaidash A.A., Kozubov A.V., Smirnov S.V., Kynev S.M., Khoruzhnikov S.E., Kozlov S.A., Vasilyev V.N. Sideband quantum communication at 1 Mbit/s on a metropolitan area network. Journal of Optical Technology, 2017, vol. 84, no. 6, pp. 362–367. doi: 10.1364/JOT.84.000362