ITM Web of Conferences **10**, 04003 (2017) DOI: 10.1051/itmconf/20171004003
*2017 Seminar on Systems Analysis*

# The Hierarchical Distributed Agent Based Approach to a Modern Data Center Management

*Andrey* Gavrilov[1, *] and *Yury* Leokhin[1]

[1]National Research University Higher School of Economics (HSE), Moscow, Russian Federation

**Abstract.** This paper overviews and analyzes progressive trends in modern data center, and existing solutions to build the distributed cloud data center. Authors present the hierarchical distributed agent-based control plane architecture to build a web-scale control layer based on software-defined domains. The goal of this approach is to design the simple extensible agent that could be used for any management purposes, just by adding some specific code. Using of this approach makes easier the scalability and increases the efficiency of the management of a multi-site environment. There are five main use-cases of using this approach: distributed cloud, hybrid cloud, hyperscale data center, IoT, Continuous Integration.

## 1 Introduction

There is currently the growing concern among many companies for the digital transformation of business – the next step for companies to start leveraging digital and IT technologies in the prioritized way to improve the performance quantifiably [1,2]. This trend requires the business to actively implement information systems that support data analytics, the big data processing, the machine learning. At the same time to remain a competitive business in a rapidly changing world it is necessary to provide the operational flexibility and the adaptability of these IT systems.

The growing number of smart devices, the size and the variety of the network traffic [3], and as a result, the necessity to increase the storage capacity, the network bandwidth and the number of computing nodes forces companies to scale its IT infrastructure or, even, to build a new site. At the same time, the IT infrastructure in data centers typically is heterogeneous, resulting problems with the integration of individual devices or units.

Large enterprises and service providers with the geographically distributed infrastructure mainly focus on the service quality and resiliency [4], and the expense reduction. It is necessary not only to guarantee a secure and reliable data exchange between data centers, but also to provide the efficient management of entire infrastructure and services.

The number of hyperscale [5] data centers will continuously grow and will be doubled by 2020 [6]. These data centers will represent 47% of all installed servers by the same period. Additionally, the number of workloads that are processing by cloud data centers will reach 92% by 2020. Furthermore, the modular data centers and related equipment market will rapidly grow to $ 35,11 billion by 2020 [7].

The software-defined approach and related technologies make easier to implement “Infrastructure-as-a-Code” [8], providing the flexibility and dynamics of the IT infrastructure through the software control. Currently, most of the new data center designed as a software-defined [9].

The key issue for the analysis of data centers is the new paradigm, called the software-defined data center (SDDC) [10]. The core idea of this approach is that all possible elements of the infrastructure should be virtualized and can be configured via management applications. The SDDC involves five major components: the software-defined networking, the software defined storage, the virtualization technology, the automatization and the orchestration. The result is an extremely dynamic, manageable, cost-effective, and adaptable architecture that gives administrators improved programmability, automation, and control.

This approach makes a data center structure quite more complex and complicated, but at the same time, it gives more flexibility, agility, scalability, availability. However, the problem of the SDDC design has not yet received all the attention it deserves and should be continuously and carefully investigated. Nowadays the SDDC concept is young and immature, hence current solutions in this area still have many drawbacks and software vulnerabilities.

Taking all earlier mentioned into account, building the distributed cloud infrastructure become the subject matter to enter into the evolving “web-scale IT” world [11,12].

Tricircle [13] provides a cascading approach that has API gateway and many OpenStack instances. However, it only unites sites and unify the management. A microsegmentation of instances would require the

[*] Corresponding author: agavrilov@hse.ru

additional management overhead. Moreover, it is desirable to process some raw data and critical analytics locally on the site.

In this paper, we present the concept of the hierarchical distributed agent based control plane (HD-ABCP) approach that could be used to design modern cloud distributed data centers. HD-ABCP has many different agents that are centrally managed by the proxy and the management cluster. This approach makes a lot easier the scalability, gives all advantages of the fog computing (agents) and has the easy-to-understand architecture.

## 2 HD-ABCP

The core idea of HD-ABCP is to split the entire data center infrastructure into software-defined domains (SDD) that are under control of agents. The agent is the local management platform, the lightweight SDN-controller and an orchestration tool that could have some subordinate agents, which could have specific functions. The whole set of agents are controlled by “Local view” of the management cluster that implements the domain management and the local analytics. Furthermore, it is possible to use some additional management applications for custom policies or domain-specific operations. On the top level of the hierarchy, the proxy is placed, the central point that unites all domains and sites. It is controlled by “Global view” of the management cluster that processes summary data from agents and perform the global analysis. Figure 1 shows the illustration of this approach.

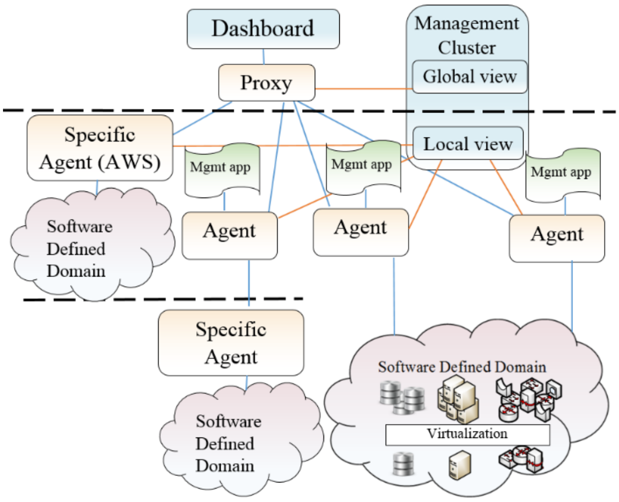

**Fig. 1.** HD-ABCP architecture.

### 2.1 SDD

SDD is composed of the physical and virtual infrastructure (including containers), hypervisors, management tools, that are managed by the agent.

Since all components are software-defined, mostly virtualized and hardware independent, we can use different platforms for specific needs. Such domains will form the distributed heterogeneous management level of the data center infrastructure that is easy-to-scale and easy-to-integrate. It allows us to use, for example, different hypervisors (MS Exchange and MS SQL Server are better to use with a Hyper-V virtualization, and Oracle databases - with the Oracle VM). Such zones will form the local software-level converged infrastructure. It is important to mention that SDD is not only the set of IT infrastructure, it is the set of components, including smart devices, applications, sensors and other elements, that should be monitored, managed and could be dynamically reconfigured via API.

### 2.2 Agent

The agent collects all the event and metric data from components of the domain via OpenStack API or different sources (directly, monitoring tools, 3rd party applications) to analyse the raw data, execute a set of commands, make retrospective reports for users and administrators and send a summary data to the proxy.

At the implementation level, agents are the set of microservices [14], like heartbeat service, self-healing service, agent-to-analytics service, agent-to-proxy service, etc.

Agents can be implemented in two ways: the thick agent and the thin agent that integrates with existing management platforms (OpenStack) The thick agent implements all needed management platforms to the proper control of distributed cloud data centers. The thin agent is placed on the top of local management platform hierarchy (integrates with existing local management stack) and control the domain via OpenStack REST API and other tools.

We can divide SDDs into smaller domains for specific purposes and operations. For example, the hypervisor with additional vendor management tools (VMware ESXi, MS Hyper-V, Oracle VM, KVM, Xen), the SDN-controller (OpenDaylight, Contrail, APIC), the computing type (bare-metal, virtual machine or container) or the public cloud (AWS, GCE, Rackspace). Furthermore, such domains could be used for specific type of services (SaaS, PaaS), workloads and applications.

### 2.3 Proxy

The main component in the control plane hierarchy that implements the centralized control of all agents and unites all sites to a single cloud/resource pool. It also process summary reports from agents for the global analytics (“Global View”) and the visualization. It is also responsible for make a decision about deploying workload based on the client region, SDD’s free space, SLA and other parameters. It collects all summary data from agents and perform analytics.

At the implementation level, the proxy is also the set of microservices, like heartbeat service, subscriber for data from agents service, proxy-to-analytics service, report service, proxy-to-dashboard service, etc.

### 2.4 Management Cluster

Management cluster implements IT operations analytics [15]. It is composed of two views: local and global. “Local View” implements capacity planning, dead VM, log analysis, uptime measuring, performance monitoring, health check. “Global View” implements WAN-optimization, workload balancing between DC/SDD, multi-SDD tenant’s workload placing analytics, adding and relocating components between SDDs.

At the implementation level, the management cluster is a set of virtual machines or containers that run applications. It is recommended to have the additional management tool for this cluster due to its importance.

The number of local view instances depends on the number of sites, because it needs to be placed on every site (it needs to be local due to self-healing principles and additional regional policies). The number of global views may vary, but should be more than one due to the high availability (instances may be in both active/active and active/standby modes).

## 3 Proof of Concept

First of all, the interaction of agents with Openstack. It is necessary to use the cloud platform due to all advantages of the cloud computing and the growing number of workloads that are processed by cloud data centers [5,16]. OpenStack was chosen since it is the most popular cloud platform, it has the huge community, many commercial releases and, last but not least, it is opensource.

Second, the presented architecture. In our perspective, this pattern clearly illustrates all of control plane components and distinguish them with regard to their functions and purposes.

Mainly we focus on an agent like a favorable and convenient unit to build the big modern data center architecture. It is easy to customize due to the extensible architecture; we could add some additional code to improve the current functionality or develop own environment-specific agent. Furthermore, we can easily connect new sites to the existing distributed cloud, just by placing the new agent. If the thick agent will be used, building new site will become much easier since all needed platforms will be already integrated. Finally, we could always develop special agents for special purposes.

Additionally, agents are the realization of the fog computing. They process all raw data and send only summary to the proxy. Due to microservices realization, agents could self-heal and easily scale if it is necessary. Moreover, it is possible to build the distributed cloud with different version of OpenStack.

From the software-defined networking (SDN) [17,18] point of view, making the control plane distributed benefits to deploy less complicated management applications and services in data centers. This approach can make all important traffic isolated and simplify firewalls and internal devices configuration. Second, a hierarchy of controllers gives higher level of a scalability, an availability, a fault tolerance and less overhead; it makes easier disaster recover, because core configuration could be restored from the «parent» controller. In addition, it provides more efficient use of resources, since the pool of devices, which need to be controlled, greatly decreased. Finally, the customization of software-defined controllers by removing unnecessary components for a specific type of the cloud or a department reduces a management complexity and increases security, performance and availability of infrastructure and workloads.

However, this approach has some drawbacks: additional overhead in design planning and expenses on the security and the fault tolerance of agents. Notwithstanding, we believe that the advantages by far outweigh the disadvantages, and this concept may have potentially important influence on modern data center architecture design.

## 4 Use-cases

This part of the paper illustrates different types of the architecture pattern usage. There are five key use-cases: distributed cloud, hybrid cloud, hyperscale data center, IoT, Continuous Integration (CI).

The distributed DC (Cloud) is the typical realization of the HD-ABCP approach. The whole distributed infrastructure will be managed as a single resource pool that is split into domains. On every site, both agent(s) and management cluster’s “Local view” should be placed [19]. Due to self-healing agents, losing connection to proxy will not disrupt the agent operation. As soon as the connection will be recovered, the agent will send all summary and crucial data of this period to the proxy.

For the hybrid cloud will be used specific agent for public cloud provider. Thereby we can easily have infrastructure in several different public clouds. The whole infrastructure could be managed from one console and automatically balance workloads between clouds. As it was said earlier, it is easy to add new sites and such distributed cloud could simply scale [20].

Nowadays, DevOps [21] is spreading across the world, and CI is significant step. We can use presented concept with respect to blue/green deployment [22]. Since we develop applications using microservices and containers, it is easy to have the same environment on all stages of development (dev, test, QA, deployment on customer site). Agent could simply reconfigure the current environment if changes are occurred and it is permitted to change to the new version.

Since the IoT world moving towards the edge computing, it is possible to use HD-ABCP approach to build a highly distributed IoT-infrastructure. According to [23], we can use the specific agent – cloud agent, fog agent and dew agent – for every IoT level. On each level of this hierarchy an agent could manage not only smart devices, but also level-specific applications.

Hyperscale data centers has the huge amount of devices [24] that generate plenty of management traffic and causes the overhead on the central management point. Since we would use smaller infrastructure areas (SDD), it is possible to localize the management traffic and increase the number of event checks. Furthermore, using vendor-specific domains may increase the service catalog of the cloud provider.

## 5 Conclusion

Nowadays, the building of multi-site and hypersclae environment is the hot topic. The data center infrastructure should have not only huge amount of processors, terabytes and gigabits per second, but also could be automatically reconfigured based on demand or occurred events, should be high available and match client's needs and expectations.

HD-ABCP will make the data center infrastructure easier to reconfigure and highly scalable. The findings of this investigation bring a software-defined domain design to the foreground and reveal the necessity of using a heterogeneous distributed control plane in IT environment. The results anticipated are intended to establish patterns of a data center design and highlight the potential of the software-defined approach. The result is an adaptive, cost-effective, easy-to-reconfigure infrastructure-as-a-code.

The verification of the proposed approach will be based on the data, obtained in experiments. The methodology obtained might dramatically change the way data centers will be designed.

## References

1. K. O'Connell, K. Delaney, R. Moriarty. *Digital Business Transformation: Disrupt To Win* [Data file]. (2015) http://www.cisco.com/c/dam/en_us/about/business-insights/docs/digital-business-transformation.pdf.
2. K. Korpela, K. Mikkonen; J. Hallikas, M. Pynnönen, 2016 49th Hawaii International Conference on System Sciences (HICSS), 3959–3968 (2016).
3. *The Zettabyte Era — Trends and Analysis – Cisco.* (2016). http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/vni-hyperconnectivity-wp.html
4. M. Alhamad, T. Dillon, C. Wu, E. Chang, iiWAS '10: Proceedings of the 12th International Conference on Information Integration and Web-based Applications & Services, 603-606 (2010)
5. L. Fang, F. Chiussi, D. Bansal, V. Gill, T. Lin, J. Cox, and G. Ratterree, Proceedings of the 1st ACM SIGCOMM Symposium on Software Defined Networking Research, SOSR'15, 1-13, (2015).
6. Cisco Global Cloud Index: Forecast and Methodology, 2015–2020 [Data file] (2016) http://www.cisco.com/c/dam/en/us/solutions/collateral/service-provider/global-cloud-index-gci/white-paper-c11-738085.pdf.
7. M. Smolkas, Modular data center market to triple its size by 2020 (2015) http://www.datacenterdynamics.com/content-tracks/power-cooling/modular-data-center-market-to-triple-its-size-by-2020/94114.fullarticle/
8. K. Morris, *Infrastructure as Code* (O'Reilly Media, 2016).
9. D. Adami, B. Martini, C. Callegari, et al. Spects '15: Proceedings of the International Symposium on Performance Evaluation of Computer and Telecommunication Systems, 1-6 (2015)
10. R. Buyya, R. N. Calheiros, J. Son, A. V. Dastjerdi, and Y. Yoon, Proceedings of the 3rd International Conference on Advances in Computing, Communications and Informatics (ICACCI 2014), IEEE. IEEE Press, 2014.
11. C. Haight, *Enter Web-scale IT* (2013) http://blogs.gartner.com/cameron_haight/2013/05/16/enter-web-scale-it
12. A.V. Gavrilov, Y.L. Leokhin, System Administrator, **11**, 76-80 (2016)
13. Tricircle. https://github.com/openstack/tricircle
14. A. Balalaie, A. Heydarnoori, P. Jamshidi, IEEE Software, **33**(3), 42-52 (2016)
15. IT Operations Analytics. https://en.wikipedia.org/wiki/IT_Operations_Analytics
16. E. Nikulchev, E. Pluzhnik, O. Lukyanchikov et al. International Journal of Advanced Computer Science and Applications, **6**(6), 22–26 (2015)
17. M. Casado , M. J. Freedman , J. Pettit et al. ACM SIGCOMM Computer Communication Review, **37**(4), 1-12 (2007)
18. N. McKeown , T. Anderson , H. Balakrishnan et al. ACM SIGCOMM Computer Communication Review, **38**(2), 69-74 (2008)
19. E. Nikulchev, E. Pluzhnik, O. Lukyanchikov et al. International Journal of Advanced Computer Science and Applications, 7(11), 165 (2016)
20. E. Nikulchev, E. Pluzhnik, O. Lukyanchikov et al. International Journal of Advanced Computer Science and Applications, **7**(1), 30-36 (2016)
21. M. Rajkumar, Anil Kumar Pole, Vittalraya Shenoy Adige, Prabal Mahanta, 2016 International Conference on Advances in Computing, Communication, & Automation (ICACCA) (Spring), 1-6 (2016)
22. M. Fowler, BlueGreenDeployment (2010) https://martinfowler.com/bliki/BlueGreenDeployment.html [Accessed 16 Jan. 2016].
23. K. Skala, D. Davidovic, E. Afgan, I. Sovic, Z. Sojat, Open Journal of Cloud Computing, **2**(1), 16 (2015)
24. AWS re:Invent 2014: AWS Innovation at Scale with James Hamilton (SPOT301) [Video file]. (2014). https://www.youtube.com/watch?v=JIQETrFC_SQ