

## Data Structure Design for Cloud Integration

### Introduction

Cloud technology has enabled the provision and sharing of resources and services over the internetworks. *Cloud computing* is a paradigm that has taken the trouble of scaling and managing out of providing resources and services, with little/no down time [1]. According to the National Institute of Standards and Technology (NIST), “Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. This cloud model promotes availability” [2]. More recently, cloud computing technology has been applied to tackle arising issues with big *data*. This combined with the rapid growth of cloud technology has transformed the methods of pooling, configuring, storing, and managing data [3]. Data management systems have always implemented relational database technology, such as Structure Query Language (SQL), to interact with data. The emergence of cloud technology and big data introduced newer non-relational database technology, such as NoSQL [4]. Furthermore, there is a push for connecting devices and platforms to each other, and/or to shared resources and services. Generally referred to as the *Internet of Things (IoT)*, sharing resources and services with cloud computing technology has allowed for more ways of connecting devices and platforms. This paper reviews cloud computing technology, data structure, and their roles in the internetwork of things.

### Cloud Computing

There are primarily three categories of cloud computing. The first is the *Software as a Service (SaaS)*. This category faces the end user that manages the application resources, typically online, such as business applications, web services, multimedia, and other applications. Examples of these applications include Google Apps, Facebook, Microsoft Web Apps, Salesforce.com, among others. The second is the *Platform as a Service (PaaS)*. This category allows users to deploy their own applications using software frameworks like Java, Python, or .NET, and storage platforms. Examples include Microsoft Azure, Amazon S3, Google AppEngine, and many more. The third category is the *Infrastructure as a Service (IaaS)*. This final category manages infrastructure and hardware resources that provide processing power and storage. This category encapsulates two resources – infrastructure and hardware. Infrastructural resources include virtual

machines (VM) and storage blocks, such as Amazon EC2, GoGrid, and Flexiscale. Hardware resource maintains the CPU, memory, physical disks, and bandwidth, that will be the primary focus for data centers [1] [5].

### **Data Structure Design**

The design of data and its structure primarily depend on the application it is pertaining to. Traditionally, relational data offered the possibility of connecting data that were related, such that changes are reflected across all levels and views to maintain accuracy and consistency. Relational database management systems (RDBMS) handled the methods of storing, maintaining, and retrieving connected data, while maintaining integrity rules across all the data views [6]. SQL data methods require and ensure the structure of the data and its relations. This means that the structure of data must be complete in design to maintain consistency in creating, storing, and retrieving of data [7]. The newer approach removes the relational structure and simplifies the definition process of data. It makes databases useful for semi-structured data, and simplifies development processes by using key-value model to structure the data. The key-value model eases creating, storing, and retrieving data [4]. Hosting a database on a cloud service like *Amazon DyanmoDB* bills a price of \$0.25 per GB [8]. To host non-database structured files, a cloud service like *Amazon S3* storage offer a pricing tier of \$0.023 per GB [8].

### **Internetwork of Things**

The IoT paradigm integrates several technologies and communication solutions. This is an effective application that allows sharing resources and services. The paradigm allows multiple devices and platforms to subscribe to a data source, independently of other service or resource. The reverse of publishing to shared resources is an IoT paradigm that allows multiple devices and platforms to create and store data to shared resources and services. Some IoT inspired services and resource that offers sharing of hardware, memory, CPU, and bandwidths often are used to hold software applications, server applications, and similar types of non-volatile data as a shared service. Running an instance of cloud hardware services, like *Amazon EC2*, bills a range of \$0.0058 per hour to \$3.2 per hour [8].

## References

- [1] C. Höfer and G. Karagiannis, "Cloud computing services: taxonomy and comparison," *Journal of Internet Services and Applications*, vol. 2, no. 2, pp. 81-94, 2011.
- [2] P. M. Mell and T. Grance, "The NIST Definition of Cloud Computing," 2011. [Online]. Available: [www.nist.gov/publications/nist-definition-cloud-computing](http://www.nist.gov/publications/nist-definition-cloud-computing). [Accessed 25 Oct 2017].
- [3] C. Ji, Y. Li, W. Qiu, U. Awada and K. Li, "Big Data Processing in Cloud Computing Environments," 2012. [Online]. Available: [http://cs.wichita.edu/~sinha/teaching/spring15/cs898aj/project/bd\\_cloud.pdf](http://cs.wichita.edu/~sinha/teaching/spring15/cs898aj/project/bd_cloud.pdf). [Accessed 25 10 2017].
- [4] J. Bhogal and I. Choksi, "Handling Big Data Using NoSQL," 2015. [Online]. Available: <http://open-access.bcu.ac.uk/1712>. [Accessed 25 10 2017].
- [5] T. B. Adji, F. S. Nggilu and S. . Sumaryono, "Overhead Analysis as One Factor Scalibility of Private Cloud Computing for IAAS Service," , 2013. [Online]. Available: <http://ijser.org/researchpaper/one-factor-scalibility-of-private-cloud-computing.pdf>. [Accessed 25 10 2017].
- [6] "A Relational Database Overview," , . [Online]. Available: <https://docs.oracle.com/javase/tutorial/jdbc/overview/database.html>. [Accessed 25 10 2017].
- [7] D. Bartholomew, "SQL vs. NoSQL," *Linux Journal*, vol. 2010, no. 195, p. 4, 2010.
- [8] Amazon, "Amazon Web Service," Amazon, 2017. [Online]. Available: <https://aws.amazon.com/pricing/>. [Accessed 23 Oct 2017].