Cloud Computing Reliability—Failure is an Option

Mazin Yousif
T-Systems, International

Cloud reliability is a measure of the probability that the cloud delivers the services it is designed for. This implies that the service is available, and performs in the way intended. When we use cloud services, it is easy to assume that they will deliver what they are designed and marketed to deliver. However, the cloud, just like everything else we develop, will fail at some time, due to hardware failures from natural disasters or human issues, unexpected software failures, or massive-scale cyberattacks such as Distributed Denial of Service attacks. It is not a matter of *if*, but rather *when*. It is also not a matter of a specific provider; they have all had outages.

When a Cloud Service Provider (CSP) delivers cloud services to millions of people and businesses, including businesses that rely on the cloud to deliver services to millions of customers, any failure could have major ramifications on the future of that CSP or business. There are many famous examples over the past few years, such as an AWS Elastic Load Balancer issue that impacted many customers, including Netflix, leading to the loss of service for tens of millions of Netflix customers on a Christmas Eve. There are many more examples of failures, some of which were documented in 2011 (www.infoworld.com/article/2622201/cloud-computing/the-10-worst-cloud-outages--and-what-we-can-learn-from-them-.html) and others that were documented in 2017 (www.computerworlduk.com/galleries/infrastructure/ten-datacentre-disasters-that-brought-firms-offline-3593580), including the worldwide British Airways failure. A major lesson is that most major failures are caused by humans. That is why CSPs augment their cloud infrastructure with technologies, architectures, resources, organization designs, processes, and workflows to make sure the infrastructure delivers its cloud services, is secure from attack, and to help protect against human error. Another potential source of failure is the interaction between applications and the cloud infrastructure. In a PaaS, for example, legacy software with its own optimizations for accessing local storage may conflict strongly with the CSP optimizations leading to performance degradation and possibly failure.

Examples of technologies and architectures include redundancies or tooling to monitor and provide visibility or Artificial Intelligence and analytics to predict failures or bottlenecks. Examples of resources and organization designs can be in the form of having teams responsible for manag-

Copublished by the IEEE CS and IEEE ComSoc
2325-6095/18/$33.00 USD ©2018 IEEE

ing backup and disaster recovery instead of single person, or conversely, only having a single person or role that has to approve changes. Examples of processes can be in the form of replication management, recovery management and fault masking. Examples of workflows can be in the form of triggered automated live migration when a certain load on a Virtual Machine (VM) or server or rack is reached. Companies have been trying innovative techniques beyond the traditional approaches we have used for years. For example, Netflix introduced the Chaos Monkey tool, which is designed solely to test how reliable operations are in the presence of failures. It kills random VMs or application modules to make sure that component outages will not disrupt the system as a whole. This approach has proved to be very effective, evolving to become a comprehensive suite of tools, referred to as the Simian Army, which is a collection of open source tools to test the resiliency of cloud operations covering reliability, security and recoverability. Examples of what is in this army are Latency Monkey, Doctor Monkey, Janitor Monkey, Security Monkey and Chaos Gorilla. I want to highlight two things about the Chaos Monkey approach and why it has proved effective: first, it creates disruption with an element of surprise and second, it proves that *the best way to avoid failure is to persistently fail.*

But no matter how good cloud reliability gets, application developers and other cloud customers have to design software architectures to tolerate cloud failures; including code design itself. We have been able to do because we have been good at identifying and understanding the majority of failures in our basic units of architectures. By basic units of architecture, I refer to units of deployments such as servers, switches, routers, firewalls, and storage appliances. Types of failures are numerous and can range from network or port failures to storage, to execution environments and many others. In other words, the key is to anticipate failures like these while the software architecture is designed.

Calculating the reliability of cloud services in a given hyper-scale datacenter with its many diverse heterogeneous efforts and features to boost reliability is a monumental task. But I am sure it can be done. I urge the community to investigate how best to determine reliability indices for such datacenters either formally or through simulations such as Agent-Based Modeling (ABM), where agents could represent the basic units of deployment mentioned earlier. In addition to the scale of the datacenter, the main crux of the challenge is the interaction among every hardware, software and mechanical component in the datacenter and the timescale of its operations. For example, we should be able to simulate how micro-failures, when they happen in mass and in any variety, propagate inside a datacenter and their overall impact on the datacenter as a whole.

Artificial Intelligence (AI) can also be very effective in boosting the reliability of cloud datacenters. We can collect data from every device and corner in the datacenter and if we place all that data in a datacenter datalake, we can analyze that data for all types of objectives including improving reliability, availability, efficiency, and security as well as reducing and optimizing energy consumption. Other emerging technologies, such as virtual, augmented and mixed reality, may also be of use. I also urge the community to investigate how best to leverage AI and other emerging technologies for the benefits of cloud datacenters and services. In fact, I am looking for authors to help here as I am editing a book on Intelligence in the Cloud and I am in the process of inviting authors to contribute to it. So if you think you can contribute, reach out to me.

This special issue on Cloud Reliability brings together an interesting collection of columns and papers. I, especially, urge our readers to check the Guest Editor Introduction because it provides a great overview of cloud reliability and also details the contents of the special issue.

About the Author

Mazin Yousif is the Editor-in-Chief of *IEEE Cloud Computing*. He's the chief technology officer and vice president of Digital Transformation at T-Systems International. He has a PhD in computer engineering from Pennsylvania State University. Contact him at mazin@computer.org.