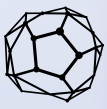
Three Important Considerations for Cloud-Native Application Design:

The Value of Designing for Resilience



Nebula.com

Table of Contents

Introduction

Cloud computing has shaken up the world of IT. Companies of all shapes and sizes have turned to the cloud to take advantage of the higher scalability, flexibility, and cost-effectiveness that an elastic cloud infrastructure can offer. But at the same time, there's a lot of wariness, as public cloud services offered by third parties come with a significant risk in the areas of security and regulatory compliance.

Overall, cloud computing presents huge advantages in cost and efficiency when compared to legacy data center models. That said, many enterprises have already made significant investments of time, manpower, and capital in their existing IT infrastructure, resulting in a lot of hesitation around embracing the cloud model as a paradigm shift. the midst of this debate, a third way is increasing in popularity: the private cloud, enabling organizations to maintain data locality and leverage existing data center infrastructure while still taking advantage of the benefits in efficiency and scalability afforded by the public cloud.

But taking advantage of this model requires a new way of thinking about how applications are designed, systems are architected, and capacity is planned . This paper outlines the challenges inherent in designing applications that take advantage of the strengths of cloud computing – and the value of doing so in terms of resilience, economics, and security.

Value 1: Resilience

No matter how much you try to avoid it, hardware fails. Many enterprise-grade magnetic hard drives have been found to have an individual failure rate between 2 and 5 percent, which is generally considered acceptable. But when you're running five server racks, each potentially holding 40 or more servers, the rate of failure will increase by an order of magnitude. In other words, administrators can always expect some part of their IT infrastructure to be in some state of faultiness at any given point. That translates to limited uptime and reduced overall performance for that data center, resulting in lowered productivity. The cost in man-hours is significant, and often, time that could be spent on innovation and delivering real business value is invested in merely keeping the lights on, so to speak.

Distributed Storage

The traditional approach when deploying applications on so-called "bare metal" servers is to work around the inevitable breakdowns with expensive, specialized storage hardware. This hardware is responsible for maintaining and storing the application's state, transferring workloads between server racks when the inevitable happens.

Ideally, the failover process is invisible to end-users and downtime is minimized, but it adds another point of failure and a significant dollar cost to scalability. After all, adding capacity to an application means adding racks of servers, which in turn means bolstering storage arrays.

The current crop of server virtualization technologies can bolster the overall reliability of an application, it's really only a stop-gap measure – virtualization hinges on "tricking" one piece of hardware into thinking that it's actually several systems, which reduces the need for computing infrastructure in a data center and improves overall availability, but still hinges on a legacy approach to application design.

That approach focuses on maintaining application state, which is where one of the major paradigm shifts presented by the cloud demonstrates its real value. If the traditional approach discussed here is a "black box," where state is maintained by a single outside source, then architecting for the cloud is more like a shotgun.

Since compute, storage and networking resources are abstracted in a cloud, applications don't have to tie into any specific piece of physical hardware. Modern distributed storage systems take advantage of this abstraction by disassociating storage from the application itself. Instead, storage is spread across cloud nodes, such that the failure of any single node has negligible impact on the whole.

If an instance in the cloud fails, an application that's designed for the cloud simply fails over to another without any kind of user intervention or going to an outside system. This model of architecture, where storage is separated out from the application, is referred to as "stateless," and it's integral to the value proposition of designing applications for the cloud.

Take MySQL – the open source relational database that provides the backbone for many web applications – as an example of how applications can benefit. MySQL only allows applications to write to one row of the database at a time, a distributed storage system allows for the throttle to be opened wide. Any row on any node can be written to simultaneously, meaning that while there is a theoretical maximum to write speed, it's certainly a lot faster than the traditional, one-row-at-a-time approach. The boost this method brings allows for a narrowing of the performance gap between

MySQL and expensive, vendor-locked enterprise database solutions.

A good example of how this can be achieved is found in the Apache Cassandra open source cache management project, which provides a distributed application cache with no single point of failure. Because it's designed to handle data across multiple cloud nodes, it shines when being run massively in parallel. This makes Cassandra easy to scale, which in turn makes it ideal for a cloud deployment.

Resilience carries other rewards. An application that's designed to tolerate failure, with the storage abstracted away, is easier to scale linearly: Adding computing horsepower to a cloud-native application is a matter of spinning up more instances; adding more storage is a matter of racking in more physical media. And when the theoretical resource utilization window becomes a more practical ceiling, adding capacity is a matter of adding more nodes, with no re-architecting or redesigning required.

Finally, managing these applications and the resources they consume via a cloud interface prevents oversubscription, with utilization clearly measured and delineated, making capacity planning a much simpler affair.

Value 2: Economics

The economic benefit of cloud computing is that it's immensely more feasible to run an application at larger scales. Cheap, scalable computing power is drawing many startups and smaller businesses are turning to large public cloud providers, as they can offer a formidable chunk of computing resources inexpensively.for pocket change. This enables even the smallest companies to scale a service or product as far as they can afford.

But the public cloud isn't for every organization. While the ability to achieve massive scale with no in-house infrastructure required is a major point of value, many enterprises hit a point where they simply have too much data for it to be cost-effective. And even when cost isn't a consideration, there are matters of security or data location to consider. We'll address the former in a bit, but the latter carries with it an economic burden.

Big Data

When processing large amounts of information – for instance, genomic research, financial transactions, or anything else in the realm of so-called "big data" – the bandwidth costs can be prohibitive, and the latency between your data source and the cloud provider carries a huge penalty to performance and agility.

Adopting a cloud – particularly a private cloud – makes it simple to add storage and compute resources. With the aforementioned highly scalable application architecture, you don't need expensive commodity storage hardware . A cheap, commodity server is just as capable of keeping an application's state persisted as an expensive storage array.

The caveat, of course, is that resources are limited – but the advantage is that oversubscription isn't a danger when the orchestration layer is ensuring resources are being used in an optimal manner.

The bottom line is that elasticity is a must when designing for any cloud environment, but one should always keep in mind resource utilization and other factors.

Value 3: Security

For governmental organizations, financial institutions, or any others who must be in compliance with mandated data regulations (or simply believe it's good business practice), it is crucial that they maintain control over their own data. Advanced security certifications consider factors like physical access to data centers, meaning that if nothing else, it's important to know in which facility sensitive data resides.

Privacy concerns for public and private cloud users are typically diametrically opposed. The data generated and stored in private clouds is owned and controlled by the operator of the cloud, who is able to deploy technologies such as data loss prevention (DLP) protection, file inspection, deep packet inspection and prescriptive firewalls. In contrast, privacy and the fear of interception is one of the primary barriers to adoption for the public cloud, so many of these controls cannot be used in that setting.

"Ownership" is an important cornerstone of cloud system and application design. For those organizations where the risk of the public cloud outweighs the reward, the private cloud offers comparable levels of performance and efficiency, without sacrificing control over their data. If security is a priority, the private cloud presents a much smaller risk of intrusion and breach.

In terms of security, a major advantage of designing applications for the cloud is in access control: While maintaining security at cloud scale can be difficult, it's also comparatively easy to audit a cloud from a central point of management. From that same perspective, access control and permissions delegation are easily managed from a single touch point.

Advantages of Cloud-native Apps

- **Scalable** – Adding resources is a matter of adding more commodity hardware, making them ideal for building Big Data analytics tools.
- **Resilient** – Cloud-native applications are built to keep running after a node fails.
- **Secure** – A key aspect of cloud-native application design is access control, making it easy to manage permissions even across massive infrastructures.

About Nebula

Nebula is dedicated to enabling all businesses to easily, securely, and inexpensively deploy large private cloud computing infrastructures. The company has developed a hardware appliance that allows any business to easily build a massive private computing cloud from hundreds or thousands of inexpensive computers.

Nebula's goal is to ignite a new era of global innovation by making big data and large scale computing accessible to every business in the world. We believe that the proliferation of data will fuel an "information revolution" across all industries, and will be enabled by democratizing web-scale cloud technology.

Nebula is privately held and venture-funded by Kleiner Perkins Caufield & Byers and Highland Capital Partners. Other investors include Google's first investors, Andy Bechtolsheim, David Cheriton, and Ram Shriram.

Nebula, Inc.
215 Castro St, 3rd Floor
Mountain View, CA 94041
(650) 539-9900
Nebula.com