Relevance of *9 Fallacies*
In
Cloud based Microservices World

Rohan Wadiwala

Introduction

Microservices is a new buzzword in town. Companies are trying to get on to this bandwagon and trying to migrate their existing application to this SOA-like architecture. But Microservice is not an entirely new concepts and neither is the basic principles it is based on. Microservice can be vaguely considered as a fork of SOA or Service-oriented architecture and hence can be considered as a part of distributed computing framework.

Distributed computing has been around for ages and has proven to be one of the more robust and scalable paradigms on which to base your systems on. One of the interesting observational pieces released for distributed computing was written by *Peter Deutsch* of Sun Microsystems in 1994 titled "***Fallacies of Distributed Computing***" (later updated by James Gosling in 1997). This whitepaper focused on 7 assumptions that are made by developers while developing distributed systems. It has been observed time & again that these assumptions form the source of great pains to the developers & architects in the long run.

The following are the 7 fallacies noted in the whitepaper:

1. *The network is reliable.*
2. *Latency is zero.*
3. *Bandwidth is infinite.*
4. *The network is secure.*
5. *Topology doesn't change.*
6. *There is one administrator.*
7. *Transport cost is zero.*
8. *The network is homogeneous*

Let's try to see if these fallacies still hold true in the cloud based microservice world as well or does the combination of microservices and cloud computing can help ease paints in these areas. We will rate each fallacy according to its relevance with 1★ being the least relevant and 5★ being completely relevant.

The Network is Reliable ★★☆☆☆

With the advent of Cloud computing the infrastructure bits have surely become much more reliable (or at least fault tolerant). A well architected environment has fail-safes and back-ups in place to ensure uptime. But there are still many factors which cannot be controlled by developers alone (e.g. 3rd part service provider networks, etc.)

On the software side, Microservice enablers & platforms have baked-in features like real-time monitoring and service recoveries in place. Also, most deployments involve setting up secondary replicas on different geo-locations making the overall system more robust by ensuring local issues don't affect the uptime.

In conclusion we can safely say this is one fallacy which has been made redundant the most due to cloud based microservices as we can ensure networking issues don't affect the overall system by following sound architectural & design practice.

Latency is zero ★★★★☆

Latency basically implies the amount of time required to transfer data from one node to another. If you look at any application hosted on cloud you can clearly demark 3 regions with respect to network namely

- Client to cloud/service region (Internet based)
- Service to service region (Inter cloud)
- Service to On-premise (WAN / Internet).

Latency differs for each of these regions and aggregated latency amounts to summation of all these.

Since decent chunk of these networks are not under our control, trying to minimize latency comes down to 3rd party SLA management (ISP's SLAs, etc.) as well as analysing traffic to find sweet spot for location management of various deployments and inter-cloud connectivity.

A well-constructed microservice deployment will try to deploy services depending on the interaction volume and will be deployed in multiple locations distributed according to external traffic. The aim is to minimize server hopping as much as possible.

Latency can raise its head especially in hybrid scenarios where network objects are emulated as local. Till the time special considerations are made for these the final product will be latency prone. Fortunately cloud providers provide various services likes dedicated WANs and ISP's can be requested to configure specific connections to suite the On-premise - cloud connectivity.

To conclude, Latency still is a thorn in the architect's flesh. Microservices can reduce the impact a little but can add to the overall latency as well. Facets like traffic location, deployment locations and traffic monitoring need to be embedded in the system to make it work.

Bandwidth is infinite ★★☆☆☆

Bandwidth or rather bandwidth availability is one component in overall infrastructure domain which has consistently improved with time. With cloud vendors providing dedicated high bandwidth as well as improved bandwidth at customers end this fallacy is redundant. Most microservices now prefer non-descriptive languages likes JSON over XML. On the same lines most of the platforms do implement some form of compression. Both these factors have helped reduce the overall package sizes of the data thus making the fallacy redundant.

But this fallacy is not eliminated. Introduction of services like IPTVs can easily hog even the highest of bandwidths. This fallacy is especially highlighted if we have multiple chatty services and we don't consider this aspect during the deployment and configuration. This can lead to not only hogging of resources but also can introduce latency. Hence to summarise this assumption is still a *somewhat relevant* fallacy.

The network is secure ★★★★★

This fallacy needs no introduction and no conclusion. It's still as relevant in 2019 as it was in 1997. Cloud platforms do provide plethora of features to counter various security vulnerabilities. But securing the network needs a multi-pronged approach from multiple directions. We not only have to add security considerations while architecting the system, but also need to consider them during drawing out management and other strategies.

Microservices does expose a larger surface for attack hence precautions need to be taken during configuration and construction. As these services are built upon other platforms and not just the operation system, we must ensure to include platform patching along with OS patching. Fortunately, PaaS & SaaS services are managed by cloud vendors who generally are on top of their game when it comes to patching.

Threat modelling and evaluation should be a norm rather than a luxury. Risks should be identified and mitigated either by process or technical updates.

Topology doesn't change ★★★☆☆

This is again one of the fallacies which has started to show its age. These days everyone is aware that network topology is a fluid system rather than a static mesh of inter-connections.

On one hand, with advent of deployment & management platforms likes Kubernetes we can never be sure where our application has been deployed, this is because Kubernetes and other associated platforms keep moving the deployed services from one node to another to maintain the recommended state and fight off against failures. But most of these system work using naming services rather than IP addresses along with abstraction provided by these platforms. This ensure change in topology does not affect the overall system output.

On the other hand, from the client-side heterogeneous mix of devices gets attached and detached. Add to this there is constant introduction of new devices by manufacturers; this results in us having a plethora of non-standard (sometimes nom-conforming) devices exchanging information with the application. This pretty much ensures your topology is in a constant state of flux.

Thus, we can conclude that this fallacy is still relevant up to an extent, but due to inherent understanding of the issue and many automated inbuild features of modern cloud platforms we can negate the effects.

There is one administrator ★★★★☆

This is more of personnel related fallacy rather than technical. In most enterprise scenarios this fallacy is magnified many folds as multiple parts / services are managed by various vendors many having their own administrators. Each of these administrator's primary concern will be their own piece of application rather than complete package. This leads to requiring high level of coordination and increase in overall development time. In worst scenarios this can lead to conflicts and finger-pointing scenarios.

Hence to sum-up this fallacy holds true and proactive steps needs to be taken right from the get-go to ensure seamless communication and interaction between the admins.

Transport cost is zero ★★★★★

This fallacy assumes that there are no side-effects for data transmission and accessing data/objects over the network. This is a fallacy since we do need to serialize & deserialize the data for transport and this costs resources (CPU / RAM /etc.) along with adding to latency.

Badly architected micro-service implementation can struggle greatly due to this as location of chatty services as well as deployment location of clusters can eat-up performance & resources. Neither micro-services nor cloud platform can make-up for this fallacy and hence this fallacy is quite relevant even today.

The network is homogeneous ★☆☆☆☆

This fallacy does not really hold true in current environments as any enterprise level implementation is a heterogeneous environment and everyone is aware of this fact. In microservices implementation the network implementation is abstracted and is generally not managed by developers. Hence managing variation in network is more-or-less automated.

From the above discussion it is safe to say that the 8 fallacies of distributed computing still hold true to this day. These fallacies still provide a sort of guiding principles if we must design & develop a robust, reliant & high-performance distributed system.

About Capgemini

A global leader in consulting, technology services and digital transformation, Capgemini is at the forefront of innovation to address the entire breadth of clients' opportunities in the evolving world of cloud, digital and platforms. Building on its strong 50-year heritage and deep industry-specific expertise, Capgemini enables organizations to realize their business ambitions through an array of services from strategy to operations. Capgemini is driven by the conviction that the business value of technology comes from and through people. It is a multicultural company of 200,000 team members in over 40 countries. The Group reported 2016 global revenues of EUR 12.5 billion.

Learn more about us at www.capgemini.com

People matter, results count.

This document contains information that may be privileged or confidential and is the property of the Capgemini Group.