Why Not All Redundancies Are Created Equal

White Paper

2600Hz is a telecom solutions engineering firm building communication systems for businesses of all sizes. A leader in cloud communications design, 2600Hz offers the reliability of the legacy providers as well as the flexibility of cutting-edge features and scalability like never seen before.

140 Geary St, 3rd Floor,
San Francisco, CA 94108

+1 (855) 464-2600
contact@2600hz.com

www.2600hz.com
www.blog.2600hz.com

Intro

Redundancy doesn't necessarily mean resiliency. In turn, not all redundancy protocols will result in failover. In light of recent network outages that have brought systems down for extended periods and affected businesses worldwide, it's important to understand that while most UCaaS and CPaaS providers claim their solution is redundant, most are NOT truly redundant. At first, that might not seem like a big deal, but it can mean the difference between normal operations and being left in the dark when disaster strikes or even when routine maintenance is being performed. Let's explore the most important behind-the-scenes differences between the majority of solutions claiming redundancy and a solution that is truly redundant.

It All Starts Behind-the-Scenes

Behind the scenes, every communication system can be conceptualized in the OSI model. The OSI (Open Systems Interconnection) model is a breakdown of the interconnectedness within communication systems or, more simply put, a representation of how applications communicate over a network. As explained in TechTarget’s definition of the OSI model, “the main concept of OSI is that the process of communication between two endpoints in a network can be divided into seven distinct groups of related functions, or layers. In this architecture, each layer serves the layer above it and, in turn, is served by the layer below it.”

When a single communication is sent - whether it is a phone call, chat, presence, etc. - data flows through the layers, across the network, and through the layers in the receiving device. Because the Layers within a network are interconnected and work together, it is imperative to take a holistic approach to redundancy, looking all the way through to Layer 7 where data processing happens, to ensure there is no single point of failure. Only then can a UCaaS solution be truly redundant.

The Two Redundancy Strategies

When it comes to redundancy, distributed systems and hot standbys are the two strategies commonly discussed, but only one - a distributed system - can result in a truly redundant solution. However, when most companies claim they have a redundant solution, they actually have a hot standby because it is easier to implement. There are two key behind-the-scenes reasons a hot standby is not truly redundant. First, the hot standby is only a replica of the primary softswitch and second, a hot standby lacks geo-redundancy. To explain, let’s dive into those a little deeper.

Why Hot Standbys Aren't Truly Redundant

Hot standbys are set up as primary-replica systems with only one primary "write" instance of the data and a replica of "read-only" instance. The "write" instance is the only one that can save changes to data because the "read-only" instance is just a copy of the "write" instance, but it can only read the data - it cannot process changes if the "write" instance goes down. When all systems are working properly, the "write" instance saves changes, then sends a copy of the new information to the "read-only" instance. This is not ideal because if the primary instance goes down, data can't be added or updated since the replica databases can only read. In addition, the hot standby is sitting idle, waiting for a disaster to strike, which is a huge operational cost.

Most companies utilizing a hot standby have the replica or "read-only" instance in the same data center as the primary or "write" instance of data, therefore lacking geo-redundancy and creating a single point of failure. Putting all of one's data in a single geographic location is risky business as many learned during Hurricane Sandy. When disaster strikes and the single data center goes down, you and your customers are completely out of luck until it is back up and running.

What about companies who do utilize more than one data center as part of their hot standby setup? They're still not in the clear. Another downside of a hot standby is the inherent delay when sending data between data centers due to distance which could - and usually does - lead to data loss in the process due to time required for failover. This is a frequent cause of demise for these systems because they're unable to reliably detect if a server is down or just running slower. Major service issues, such as not being able to make or receive calls, are likely to occur despite having the standby if the primary instance of the data goes down.

The 2600Hz Difference

So, how does 2600Hz's KAZOO stack up? We have a distributed system, and KAZOO is a truly redundant platform that is designed from the lowest level component, the programming language Erlang, up to the highest operational level to expect and plan for something going awry. Our engineers design with the expectation that anything can and will go wrong, and we figure out how the cluster as a whole should function when a given subset of components fail for an intentional reason like during maintenance, or unintentional like a hurricane flooding the data center.

When it comes to Layer 7 and the processing of data, 2600Hz utilizes a multi-primary database setup instead of a primary-replica setup (hot standby). Our multi-primary database setup contains 3 primary, or "write", instances of data so any server can receive a request and satisfy it, even if it doesn't hold the data itself. By utilizing multiple primaries, any server can accept a change and that change will be replicated across the cluster of servers. This keeps things going even if one data center were to go down. It is also key when it comes to our geo-redundancy.

The geo-redundancy in KAZOO, which we call Zones, is the ability to segment the cluster along the data center boundaries. We always want to keep as much data as possible in the local data center to minimize the delay in processing a request. However, if one data center goes down, since the database is consistent across zones (aka data centers), every server will come to the same decision on how to process a request and all of our servers are in use and not sitting idle. That means if the entire east coast data center is down, calls and API requests can be shifted to midwest or west coast data centers and still work as expected and most importantly, without a delay like there would be with a hot standby.

Conclusion

While it sounds great to hear that a solution is redundant, it pays to dig deeper and find out exactly how the redundancy protocol is set up. Is it a hot standby set up? Is it geo-redundant? Is it truly redundant? Distributed systems may be harder to build, which is the reason so many companies tend to lean towards Hot Standbys. But in the long run, the savings you receive from redundancy, performance, and resiliency highly outweigh the initial costs. All it takes is one major outage for you to lose customers and your reputation.

If you would like to find out more about 2600Hz or set up a free trial of KAZOO, please contact our sales team at sales@2600hz.com.