the Availability Digest

www.availabilitydigest.com

It’s Official! Leap Day Caused the Windows Azure Outage

May 2012

In our March, 2012, Never Again article entitled “Windows Azure Cloud Succumbs to Leap Year,” we related how Microsoft’s Windows Azure Platform as a Service (PaaS) cloud went down for a day and a half as the result of what appeared to be a leap year software bug. At the time, the conjecture was that validity dates for SSL (Secure Sockets Layer) certificates were calculated erroneously. As it turns out, the conjecture was pretty close.

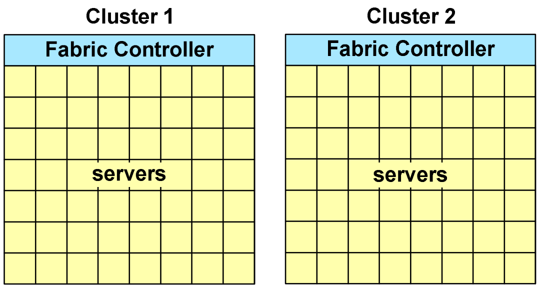
Following in the path of Google and Amazon, who have been very transparent in describing publicly what happened during major outages, Microsoft has released a detailed timeline of exactly what went wrong in this major outage and the sometimes frantic efforts to restore service to its customers. The rather lengthy description is found in a Microsoft blog[1] authored by none other than Bill Laing, Microsoft’s Vice President of Windows Servers and Solutions.

In this article, we summarize the events related in his blog. But first, let us look at some architectural aspects of the Azure cloud, as related by Mr. Laing, that are important to understand for purposes of this outage.

The Azure Cloud Architecture

As in any cloud, applications running in the Azure cloud consist of virtual machines (VMs) running on physical servers in Microsoft data centers. Microsoft manages six Azure centers around the world.

Azure Clusters



Each Azure center contains thousands of physical servers. Servers are grouped into clusters of about 1,000 servers each. Each cluster is independently managed by a redundant software facility called the Fabric Controller (FC).

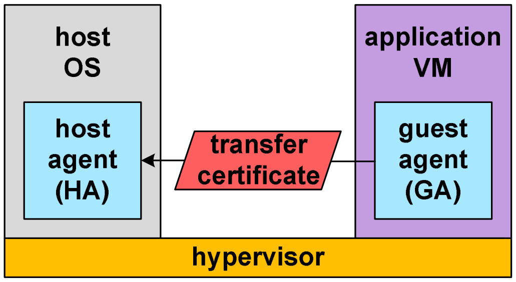
The FC manages the life cycle of applications running in the cluster, including provisioning the physical resources needed by the applications, deploying applications in VMs in the cluster, updating applications and guest operating systems, scaling out applications, and monitoring the health of the physical servers and other hardware in the cluster. If a server should fail, the

[1] Summary of Windows Azure Service Disruption on Feb 29th, 2012, *MSDN Blogs*; March 9, 2012. http://blogs.msdn.com/b/windowsazure/archive/2012/03/09/summary-of-windows-azure-service-disruption-on-feb-29th-2012.aspx

FC reincarnates the server's virtual machines on healthy servers within the cluster. This is called *service healing*.

Agents

A physical server is tightly integrated with the VMs running on it through the use of a *guest agent* (GA) bound into the operating system image of each VM. Each physical server has a *host agent* (HA) that the FC leverages to communicate with the GAs that the HA's server is hosting. For instance, the HA deploys application secrets such as SSL certificates that an application uses to secure HTTPS end points. The HA also sends *heartbeats* to each GA on its server to determine if the GA's VM is healthy.



Important to understanding the cause of the Azure outage is that when a VM is initialized, its GA creates a transfer certificate using public-key encryption to protect application secrets that will be transmitted over physical or logical networks. The first step that a GA takes when connecting with its HA is to pass to the HA its public key in a *transfer certificate*. The HA can then encrypt secrets readable only by the GA because the GA has the private key.

The GA generates a new transfer certificate whenever:

- a VM is created (that is, when an application is deployed).
- when an application is scaled out.
- when a deployment updates its operating system.
- when the FC reincarnates a VM that was running on a server that the FC has determined to be unhealthy (service healing).

Service Healing

The HA is responsible for ensuring that new VMs come into service properly. When a VM is first created, the HA waits up to 25 minutes to receive a transfer certificate from the VM. If a transfer certificate is not received in that time, the HA will reinitialize the VM and will wait again.

If after three such attempts the HA still has not heard from the VM, it assumes that the physical server may be down. At this point, it will inform the FC that the server is faulty. The FC will reincarnate the VM on another physical server, and it will place the suspected server into Human Interaction (HI) mode for operator intervention. The FC will service-heal the other VMs running on the faulty server by moving them to other healthy servers in the cluster.

The Leap-Day Bug

The Software Bug

We can now understand what happened to take down Azure. When the GA creates a transfer certificate, it gives it a one-year validity range. It uses midnight of the current day as the *valid-from* date and simply adds one to the year to get the *valid-to* date. Consequently, certificates created on February 29, 2012, had an expiration date of February 29, 2013.

This is an invalid date, and the certificate creation failed. On leap day, no GA could create a transfer certificate. Therefore, no newly initiated GA was able to connect to its HA, and its VM was terminated.

Cascading Outages

This caused the HA to declare the physical server faulty. In response, the FC not only tried to reinitialize the VM on another server; but it also moved all other VMs on the presumably faulty server to other servers in the cluster. Now all these VMs tried to generate transfer certificates, all failed to do so, and all were moved to still other servers. The server outages rapidly cascaded throughout the entire Azure cloud.

This disastrous effect was partly mitigated by an HI threshold in the FC designed to prevent a software bug from propagating throughout the cluster. If a certain number of servers in a cluster transit to the HI state, the FC moves the entire cluster to the HI state. It stops service healing so that VMs are no longer moved between physical servers. It suspends other service management functions such as creating new VMs, applying updates, and scaling out so that the operators have an opportunity to take control of the cluster and to repair the problem before it progresses further. During this time, VMs in the cluster that were working before the cluster was moved to the HI state continue working; they just can't be modified by service management.

The Time Line

As one would now expect, the Leap Day bug started precisely at 00:00 UST (Universal Standard Time) on February 29, 2012, which was 4 PM PST (Pacific Standard Time) on February 28th. GAs in newly launched VMs were unable to create transfer certificates. Precisely 75 minutes later (after three retries 25 minutes apart), at 5:15 PM PST, the HI threshold in some clusters was exceeded and these clusters went into HI state. This problem was compounded by the fact that operations staff was just in the midst of a rollout of new versions of the FC, the HA, and the GA. This ensured that the clusters involved in the rollout would hit the Leap Day bug immediately. The bug worked its way more slowly through other clusters.

From here on, staff efforts to identify and rectify the outage tracked the following time line (all times are PST):

> 6:38 PM, February 28: The cause of the bug was identified.
>
> 6:55 PM: The operators disabled service management to stop the cascading. At this time, no users could deploy new applications; nor could they stop, update, or scale existing applications. However, already deployed applications continued to run properly.
>
> 10:00PM: A test plan for the upcoming corrected GA was prepared.
>
> 11:20 PM: The correction to the GA code was completed.
>
> 1:50 AM, February 29: The corrected GA code was successfully tested using a test cluster.
>
> 2:11 AM: The fix was then rolled out to some of Microsoft's own clusters and ran properly.
>
> 5:23 AM: The fix was rolled out to most clusters, and these clusters were restored to service. As each cluster was updated, service manageability in that cluster was restored.

The Secondary Outage

"Most clusters" are the operable words. An additional problem occurred with seven clusters that had just started their rollouts of the updated FC, HA, and GA components. Some servers in these clusters had the old GA and others had the updated GA, all with the Leap Day bug. Servers in these clusters were all rolled back to the original HA code but with the new and corrected GA code. Unfortunately, no one was aware that the new networking plugin that was installed along with the new GA code was incompatible with the old HA. The networking plug-in configures the network for the VM. Consequently, no VMs in these clusters had networking capabilities.

The updating of these seven clusters was completed with the incompatible combination by 2:47 AM on February 29th, only to find that all of the VMs in these clusters, even those that had been healthy, had become disconnected from the network because of the network plugin incompatibility. This situation was fixed and the correct update of these clusters was completed by 8:00 AM

However, a number of servers were left in a corrupted state because of all these transitions. The developers and operations staff worked feverishly to manually restore and validate these servers, but it was not until 2:15 AM, March 1, a day and a half after the outage, that all clusters were finally restored to service.

Improvements for the Future

Microsoft is undertaking several initiatives to preclude such outages in the future. They are improving the four phases of the incident life cycle – prevention, detection, response, and recovery.

Prevention: Code analysis and testing will be improved, especially with respect to time-related bugs. The FC will be improved to recognize the difference between hardware and software bugs so that the cluster HI state will not be entered due to a software bug. Service management will be reconstructed to make it more finely granular so that only those portions of the services that need to be disabled in an emergency will be.

Detection: Taking 75 minutes to recognize a problem is too long. There will be improvements to allow fast-fail capabilities.

Response: The dashboard capacity will be increased to handle anticipated outage loads, and better summary information will be provided. Customer support staffing will be improved to eliminate long hold times in emergencies; and better use will be made of other channels such as blogs, Facebook, and Twitter.

Recovery: Internal tools will be improved. Better control of dependency priorities will be implemented so that necessary services are brought up before other services that depend upon them. Better customer visibility into the recovery progress will be provided.

In addition to its heartfelt apology for the inconvenience felt by Azure customers, Microsoft has issued a 33% credit to every customer for its affected billing months (February for all, March for some).

Summary

What a catastrophe a little software bug can make. It is not much of a step to add to the statement “increment year by one” the statement “if Feb. 29, decrement day by one.” This error should have been caught in code reviews and testing, an improvement that Microsoft is now making. Shortcutting these important steps in the development phase can potentially cost orders of magnitude more than what is saved.