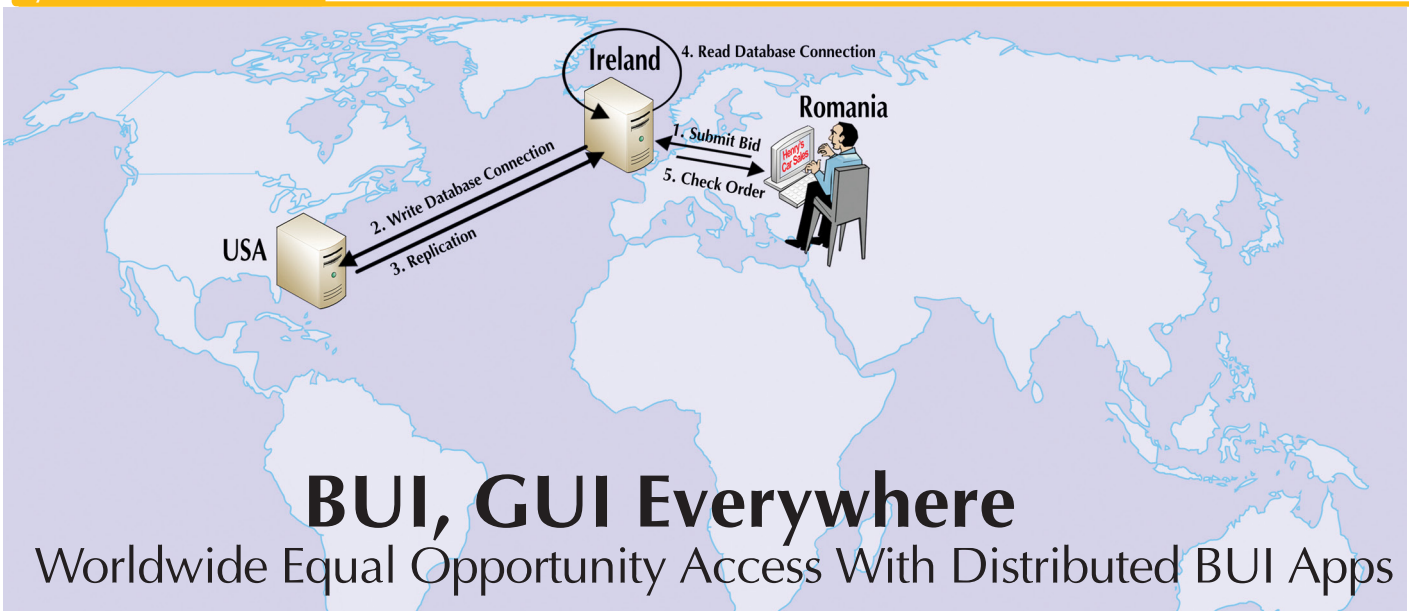

BUI, GUI Everywhere

Worldwide Equal Opportunity Access With Distributed BUI Apps

Henry, a developer on the East Coast of the United States, soon got his GUI program running on the cloud server in BUI. Anyone in the world could now navigate to the company page and browse the online database of new cars! His program offered 360° exterior, and interior views of each automobile, allowing the user to choose colors, upholstery, different spoilers, wheels, and other customizations. Customers could build the auto of their dreams, submit it to the dealer, negotiate the price, and have it delivered without ever setting foot in the showroom, all online for any of the dealer's worldwide locations.

But, there were already problems. Henry got reports that the program took two seconds to load for customers in North America, ten seconds to load in Europe, and fifteen seconds to load in Asia. Fifteen seconds is a long time to wait for a page to load. But wasn't this to be expected? After all, his server was in the US; the further away a customer is from the server, the longer the program takes to load.

Henry thought, "Surely, there's a way around this," and actually, there is!

The solution to providing consistently quick access to a BUI app anywhere in the world is to set it up as a distributed application using readily available tools from BASIS and Amazon. While Amazon offers tools for reaching multiple machines with the same address, BBj® offers tools for replicating programs and data across several machines. In this article, we will discuss Amazon geo-aware DNS (Domain Name System) addresses, and BBj File/Directory-Database Replication, and how they can be combined to make Henry's auto sales application run equally well and transparently from anywhere in the world.

The Primary Tools

The two major tools for creating a distributed BUI application are Amazon's geo-aware DNS addresses and BBj File/Directory-Database Replication. Geo-aware DNS addresses allow one DNS to correlate to several IP addresses, where each address is located in a different region. BBj Replication maintains up-to-the-minute copies of programs and data between machines in all of the regions. Let's take a more detailed look at these tools.

By Shaun Haney
Quality Assurance Engineer

Geo-aware DNS Addresses

One major feature of cloud computing is being able to provide content to users from a server close to them. In the past, you might provide North American users a .com address, British customers a .co.uk address, and German customers a .de address, etc. It would be great if your customers did not need to know what address was near them and you could just use the same web site address all over the world. Geo-aware addresses provide exactly this. Using Amazon's Route 53 configuration tool, you can specify a single DNS name, assign it an IP address, and then specify a region of the world where the DNS belongs. To add more locations, simply add more DNS records with IP addresses and specify a different region for each address. For example, North American customers will go to the IP address of a server on the East Coast or West Coast, while European customers will go to an address in Ireland, South American customers will go to a server in Sao Paulo, and Asian customers will go to an address in Japan. All of those customers will type the same website name in the address bar of their web browser and Amazon will reroute them to the correct machine behind the scenes, without the customer having to do anything.

BBj Replication

BBj replication is invaluable anywhere redundancy is needed. Replicated data is not only a quickly available up-to-the minute backup, but is also a read-only copy of the original data accessible in real time. BBj replication is not just for databases, but for any file that needs a copy maintained on one or more remote machine(s). Replicated BBj programs can run on the replication target machine, but any changes to the programs should be made only on the source machine.

Supplementary Tools

Other important tools include AWS EC2 (Amazon Web Services Elastic Compute Cloud) and BBj databases and files. Amazon's EC2 framework is what allows Cloud developers and administrators to create, configure, upgrade, start, and stop cloud machines. A complete discussion of EC2 is beyond the scope of this article, but the reader should be aware that EC2 is the underlying framework for running cloud servers.

BBj databases and files are also critical to any non-trivial BUI application. Rather than discussing data structures in any detail, it is more important to know that an application's data will replicate along with its source files, and that the "genuine" data will reside on the source machine while a read-only copy will reside on each target machine. The BUI application design should include two connections for each set of data: a "read" connection that accesses the local data and a "write" connection for the "genuine" source data.

The Architecture

Now that Henry knows a distributed BUI application is possible and what tools he needs for the process, what does a distributed BUI application's configuration look like?

A distributed BUI application consists of one source server and any number of target servers such as shown in **Figure 1**. The source server and target servers all have an install of BBj, the BUI application source code, and the application's data. Each server has a permanent IP address. All of the IP addresses are associated with a single geo-aware DNS Name. On each machine, the BUI application is identical and has two connections for each dataset used: a read

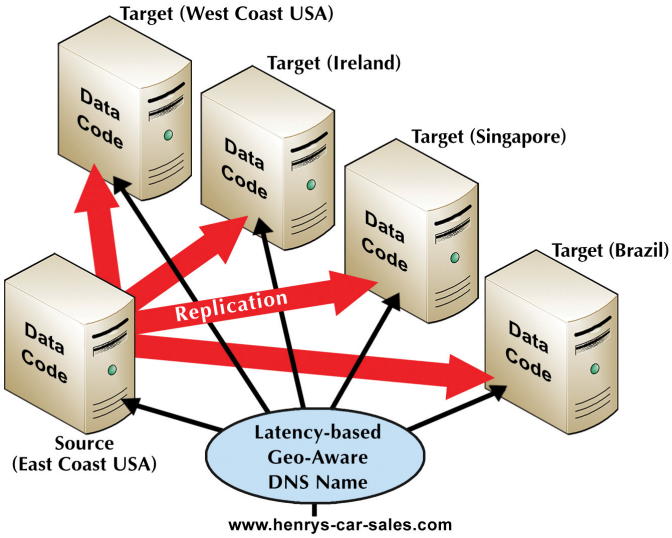

Figure 1. Sample configuration of a distributed BUI application

connection and a write connection. The read connection goes to localhost so that reads occur on the local copy of the data. The write dataset connection goes to an unpublished name for the source server so that all writes are on the source's copy of the data. Meanwhile, replication maintains up-to-the-minute copies of the data, BUI source files, and any relevant configuration files on all of the target machines. In a distributed BUI application configuration, if one makes changes to data, source code, or the application configuration on the source server, replication automatically propagates those changes to the target machines.

After planning the architecture, Henry put his nose to the grindstone. In the course of a week, he sets up servers in California, Virginia, Ireland, Brazil, and Singapore. He chose Virginia for his source server because, even though the source server could be anywhere, it made the most sense to have it in the same region as those in his company who will maintain it.

Now with servers in so many regions, Henry knows that customers will have faster load times for his BUI car sales application and as a huge bonus, he now has several redundant copies of his application and data in case of a failure.

How it Works

With so many servers relatively close to almost any customer in the world and application load times much faster, how exactly does using the BUI application work? Recall that Henry's main server is in Virginia.

While Henry is fast asleep in the U.S., Ivan, a customer in Romania, decides it's time to buy a new car. Ivan simply types `www.henrys-car-sales.com` *into his favorite browser's address bar – the exact same address any other customer in the world would type – to connect him to the closest server, which is in Ireland. Ivan browses for cars and then chooses the color, seat fabric, and window tinting, all from the server in Ireland. After perfecting his future vehicle, Ivan saves his customization and offers a price to start the negotiation process.*

Ivan's car customizations and initial bid transmit to the interpreter that is running the BUI application in Ireland. The BUI application's write connection in Ireland then redirects Ivan's data to the primary database in Virginia. As the database in Virginia updates, the updated records replicate back to the local copy of the database on the machine in Ireland. Because the process only takes a couple of minutes, Ivan will be able to review his order with all the details displayed directly from Ireland's local database.

The More BUIs the Merrier

Amazon's geo-aware DNS addresses and BASIS' replication feature come together to allow BUI applications to perform quickly, no matter where in the world the user runs them. The best part is that with replication, most of the application maintenance occurs on the source and propagates to the target machines.

After Henry setup the other servers and the replication jobs, he still only has to maintain and update a single server. All the other servers are maintained automatically. So, while this adds some time to the initial deployment, after it's up and running Henry doesn't have to do any more work to get updates to all the servers than he does to a single server.

The result is a distribution of a BUI application that is completely transparent to customers around the world. For all they know, it is running on a single server in their own backyard! ■

Read *Are You Prepared for Cloud Failure?* at links.basis.com/12cloud