

MOVARTO: Server Migration across Networks using Route Triangulation and DNS

Problem: How do I move a server across the Internet?

Server migration across wide area networks is a fundamental issue faced by many enterprises in various contexts. We present two motivating examples. In today's world many applications are hosted on servers at remote data centers and accessed via the Internet. These applications are required to be up 24 x 7. But competition and turmoil in the market as well as maintenance requirements force the lights to go out occasionally even in the best of datacenters. This forces the migration of servers across the wide area network without disrupting existing connections. Another example is from the world of CDNs (Content Delivery Networks, e.g. Akamai). CDNs sometimes face malicious or otherwise harmful traffic that is specific to one of their customers but threatens to disrupt all their customers because they are all hosted on the same physical server. In such situations there is often a need to isolate the harmful traffic by moving the threatened virtual server to a different physical server across the wide area network without affecting other customers. Movarto addresses such problems by a layering of careful configuration on top of standard technologies (such as VMotion which are primarily designed for migration within the same network segment and not across the wide area network).

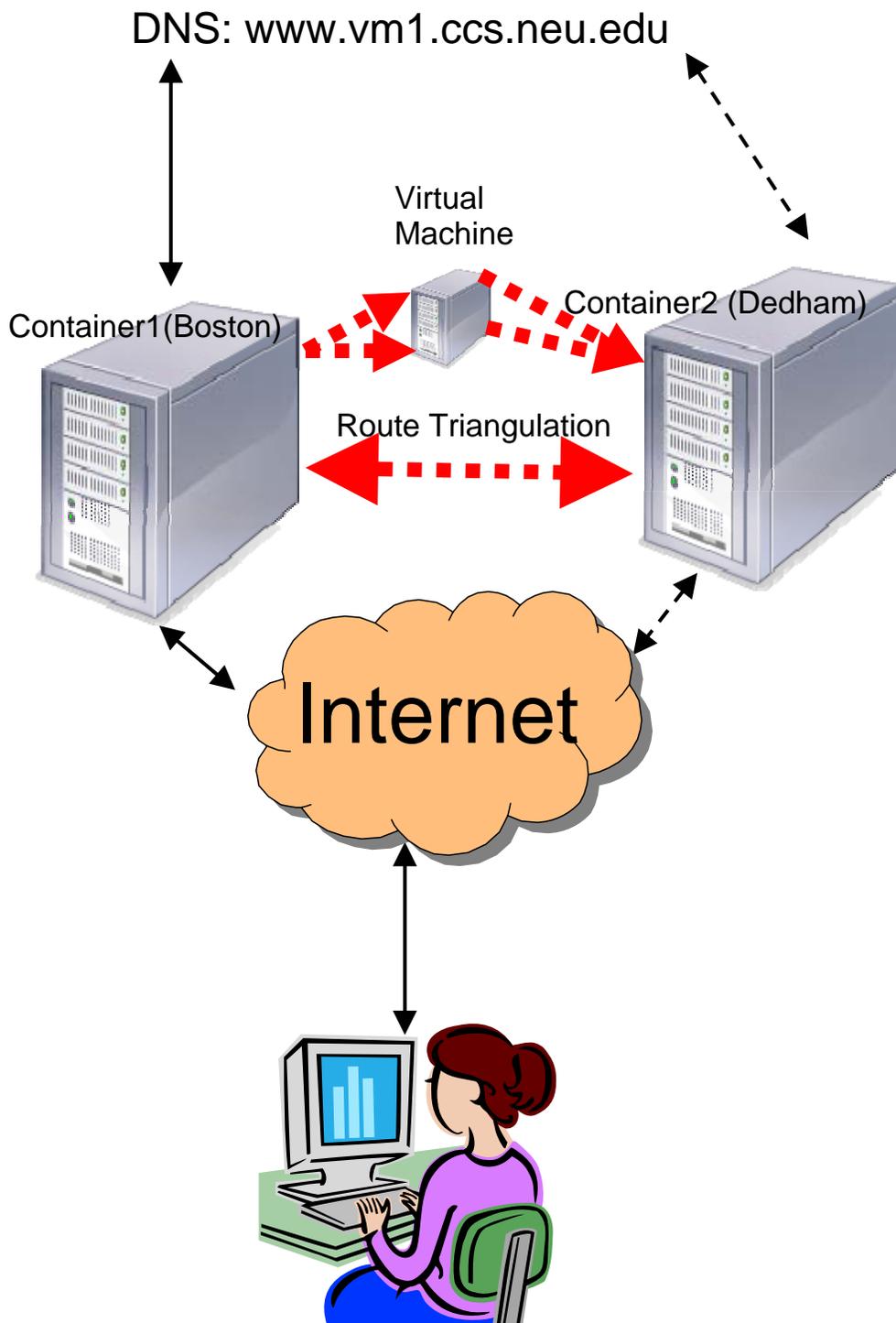
Solution Concept and Implementation

The basic idea is to extend the notion of virtualization from servers to networks or *network virtualization*. Standard migration of the virtual machine enables a process to continue its lifetime after only being momentarily paused, but this does not allow for the network to continue running without interruption. The key to Movarto is uninterrupted migration of a server along with its network. In order to do this we need to guarantee two things: first, new connections must be sent to the new location and second, existing connections must be seen to originate from the current location otherwise client-side firewalls would deny the incoming packets. We achieve the first by coupling DNS changes closely with the server migration and we achieve the second by route triangulation.

Given an existing web service running on a machine we use standard tools (e.g. VMWare's Converter) to transform the current physical server into a virtual web server. Next, we create an additional virtual *forwarding* server on the current physical server which forwards all packets to and from the virtual web server. We then migrate the virtual web server running at the current location to the new location by synchronizing the backend data store and memory. After the data is synchronized we lock the virtual web server momentarily and pause for one last synchronization. After this last synchronization, all of the information necessary to run the virtual machine on the new physical machine is available and we resume the virtual web server in its new location. At the new location we also have a virtual forwarding server. Simultaneous with the migration of the virtual web server we change the DNS configuration so that new connections to the web service will automatically be sent to the new location. And simultaneous with the migration too, the forwarding server at the old location forwards all packets to and from the new forwarding server which in turn reciprocates for the old connections that exist from before the migration. Movarto uses route triangulation to redirect the packets in this manner until the old connections are drained off. This guarantees that existing connections will be uninterrupted. Finally, Movarto constantly monitors the use of the route triangulation and brings it down once the last connection terminates, allowing for the original physical machine to be turned off entirely. We have built a working demo using a web server (supporting the game Tetris) over Linux/Qemu and OpenVPN for forwarding.

Conclusion

We identified a need for server migration across the wide area network (not just within a network segment) and have provided a comprehensive solution that achieves the goal of minimizing downtime without disrupting existing connections



————— Pre-move

..... Move

----- Post-move