

# Mainframes are dead, Right?



Don MacVittie, 2012-20-09

Funny thing about the hype cycle in high tech, things rarely turn out the way cheerleaders proclaim it will. Mainframes did not magically disappear in any of the waves that predicted their demise. The reason is simple – there is a lot of code running on mainframes that works, and has worked well for a long time, rewriting all of that code would be a monumental undertaking that, even today, twenty years after the first predictions of its demise, many organizations – particularly in financials - are not undertaking.

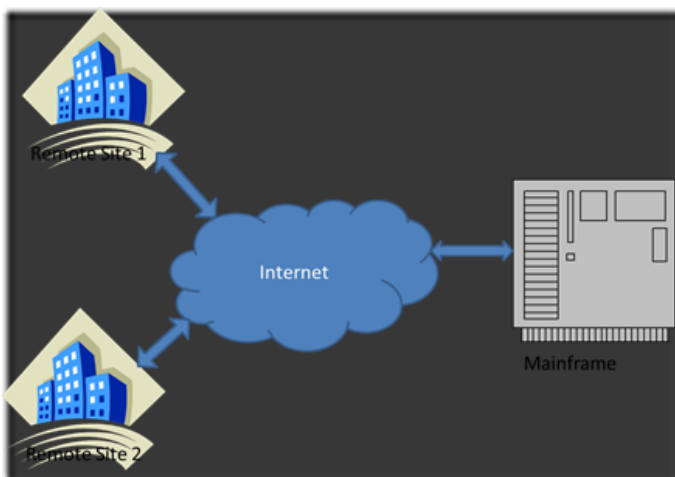
Don't get me wrong: There are a variety of reasons why mainframes in their current incarnation are doomed to a small vertical market at best in the very very long run, but the cost of recreating systems just to remove mainframes is going to continue to hold them in a lot of datacenters in the near future.

But they do need to be able to communicate with newer systems if they're going to hang around, and the last five years or so have seen a whole lot of projects to make them play more friendly with the distributed datacenter.

While many of these projects have come off without a hitch, I saw an interesting case in financial services the other day involving a mainframe and a slow communications channel. Thought it was a slick solution, and thought I'd write it up in case any of you all run into similar problems.

This company had, over time, become very distributed geographically, but still had some systems running on their mainframe back in corporate HQ. The systems needed to communicate back to the mainframe, but some of them were on horrible networks that would sometimes suffer latency and line quality issues, yet the requirement to run apps on the mainframe persisted. The mainframe has limited I/O capabilities without expensive upgrades that the organization would like to avoid, so they are considering alternate solutions to resolve the problem of one branch tying up resources with retransmits and long latency lags while the others back-filled the queue.

It's more complex than that, but I'm keeping it simple for this blog, in hopes that if it applies to you directly, you'll be better able to adapt the scenario to your situation. No highly distributed architecture with mainframe interconnects is simple, and they're rarely exactly the same as another installation that fits the same description, but this will (hopefully) give you ideas.



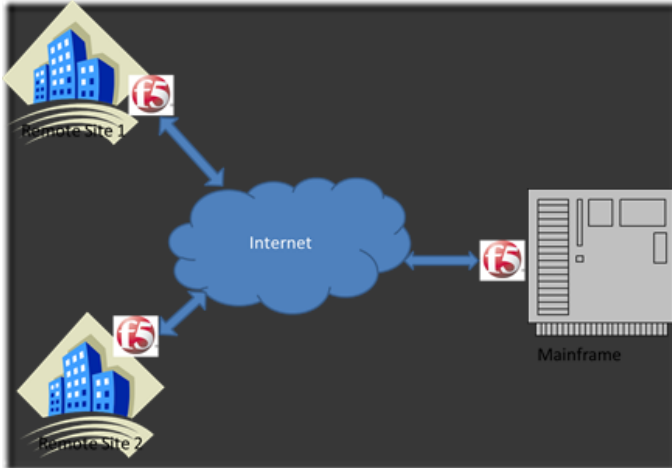
This is the source problem – when site 1 (or site N) had connection problems, it locks up some of the mainframe's I/O resources, slowing everything down. If multiple sites have problems with communications links, it could slow the entire "network" of sites communicating with the mainframe as things backed up and make even good connections come out slow.

In the following slide, of course I'm using F5 as an example – mainly because it is the solution I know to be tested. If you use a different ADC vendor than F5, call your sales or support reps and ask them if you could do this with their product. Of course you won't get all the excellent other features of the market-leading ADC, but you'll solve this problem, which is the point of this blog.

The financial services organization in question could simply place BIG-IP devices at the connecting points where the systems entered the Internet. This gives the ability to configure the F5 devices to terminate the connections, and buffer responses. The result is that the mainframe only holds open a connection long enough for it to transfer over the LAN, and eliminate the latency and retransmit problems posed by the poor incoming connections.

Again, it is never that simple, these are highly complex systems, but it should give you some ideas if you run into similar issues.

Here is what the above diagram would look like with the solution pieces in place.



Note that the F5 boxes in the branches would not be strictly necessary within the confines of the problem statement – you only care about alleviating problems on the mainframe side – but offer the ability to do some bi-directional optimizations that can improve communications between the sites. It also opens the possibility of an encrypted tunnel in the future if needed, which in the case of financial services, is highly attractive.

I thought it was cool that someone was thinking about it as a network problem with mainframe symptoms rather than the other way around, and it is a relatively simple fix to implement. Mainframes aren't going away, but we'll see more of this as they're pushed harder and harder and put behind more and more applications. Inventive solutions to this type of problem will become more and more common. Which is pretty cool.

## Related Articles and Blogs:

- [FNZ Teams with F5 to Deliver the Financial Services Industry's Most ...](#)
- [FishNet Security Ensures Application Performance and Availability ...](#)
- [Cloud Changes Cost of Attacks](#)
- [Network Optimization Won't Fix Application Performance in the Cloud](#)
- [Forget Performance IN the Cloud, What About Performance TO the ...](#)
- [Data Center Feng Shui: Architecting for Predictable Performance](#)
- [F5 Networks Expands Application Ready Network™ to Include ...](#)
- [F5 Networks Announces Application Ready Network™ for Oracle](#)
- [New F5 Application Ready Network \(ARN\) for Oracle Siebel](#)

---

F5 Networks, Inc. | 401 Elliot Avenue West, Seattle, WA 98119 | 888-882-4447 | [f5.com](#)

F5 Networks, Inc.  
Corporate Headquarters  
[info@f5.com](mailto:info@f5.com)

F5 Networks  
Asia-Pacific  
[apacinfo@f5.com](mailto:apacinfo@f5.com)

F5 Networks Ltd.  
Europe/Middle-East/Africa  
[emeainfo@f5.com](mailto:emeainfo@f5.com)

F5 Networks  
Japan K.K.  
[f5j-info@f5.com](mailto:f5j-info@f5.com)