What I Learned This Month: Pony Motors and Light Speed

Scott Chapman
American Electric Power

This month I learned what a pony motor is. I would have preferred to come to this knowledge outside the context of my professional life. I also contemplated the rather unfortunate universal speed limit of the speed of light.

First, large diesel engines sometimes need a pony motor, a small gasoline powered engine, to help them start. A pony motor may also be a small electrical motor used to start a large synchronous motor[1] or generator. Without a correctly functioning pony motor, the larger engine / motor / generator can't start.

A generator large enough to power a data center might be considered "large" in this context.

Shortly after learning what a pony motor is, I was asked to research what it would take to split our parallel sysplex[2] workloads between two local data centers in an active / active configuration. The goal would be that if one data center suffered an outage, the workload would continue in the other data center mostly or completely unaffected.

Today we have two mainframes sitting a few meters apart with sysplexes crossing both machines. During planned maintenance or upgrade we can take down one system and the applications continue to function on the other. Unexpected outages usually are temporarily disruptive to about half the workload (the work running on the system that had the outage).

Conceptually the idea of extending the distance between the machines seems simple enough. Pick one machine up, move it to a new data center. Place a second disk subsystem in the new data center and enable synchronous data replication so whatever is written to disk in one data center is also written to disk in the second data center. Connect the machines in the two data centers with the appropriate fiber connections.

My initial reaction was that it would be great fun to engineer that system, but there would be a significant performance penalty which would increase with distance. This is because of that pesky speed of light limitation. Roughly speaking, it takes about 5µs (microseconds, or millionths of a second) to get a signal through 1km of fiber. We generally need a response or confirmation from any signal we're sending between systems, so the total propagation delay added by 1km of fiber is about 10µs.

I wasn't too worried about the impact to disk I/O time. I/Os satisfied out of the disk subsystem cache may be a few hundred µs but cache misses will likely be

---

[1] See, for example: http://en.wikipedia.org/wiki/Synchronous_motor .

[2] A parallel sysplex is a mainframe configuration utilizing multiple machines to present a single system "image" to the applications. Properly configured applications running in a sysplex can run on any and all systems within the sysplex, all while sharing and updating the same set of data.

several milliseconds (ms, thousandths of a second). Averaged total time in the range of 1-3ms is not unusual, so the average impact is not a large percentage. Of course only some of the total I/Os are "important" in terms of noticeably affecting application and system responsiveness, so we'd need to look at those in particular to see how they would be impacted. But overall, the propagation delay for disk I/O would likely be an acceptable overhead.

What I was concerned about was coupling facility response times. Coupling facilities (CFs) are systems in a sysplex that act as shared memory for all the systems, which allow the applications to update the same set of data with integrity. Locks and updated data are written and read from the CFs. The number of requests per second to the CFs can be pretty high on a busy system: hundreds of thousands or even millions of requests per second.

Lock requests are some of the highest volume and most performance sensitive of requests. They also are processed synchronously, meaning the task that made the request spins on the CPU waiting for the response. This waste of CPU cycles makes sense because the performance of those requests is so critical. On our current zEC12 systems, I've seen CF lock request response times in the 5-10µs range. The amount of CPU spent waiting on the request can be calculated by multiplying the average response time by the number of requests. The amount of CPU time expended on synchronous CF requests can be significant during busy periods. If you're responsible for the performance and capacity of a parallel sysplex, you may want to occasionally calculate that overhead. It's an interesting and important metric to understand.

If you remember that 1 km of fiber adds about 10µs, and that our current CF lock response times are less than that, you can see why I was concerned. When we last looked into doing a second local data center 10 years ago, parallel sysplex distances were limited to 40km. I knew that the published distance was now 100km. I wasn't sure how it could be possible to do active data sharing across that distance.

We got into contact with the relevant IBM GDPS[3] experts, from whom I learned that many customers around the world are doing data sharing between data centers in an active / active configuration. But almost all of these are less than 10km apart, and my contact was unaware of any that were more than 20km apart. So you can extend a sysplex 100km, but as a practical matter you can't have active data sharing between two locations that far apart.

This made sense to me. Given that the proposed distance between the data centers was 50km "as the fiber runs", I had to report that extending data sharing across that distance wouldn't be technically feasible. On the off chance that we might decide to build the data centers closer together, we still had to go through the exercise of designing such a configuration and estimating all the costs and effort that it would take to implement such a configuration.

---

[3] Geographically Dispersed Parallel Sysplex – an IBM offering to manage and automate sysplex operations, often when the sysplex extends beyond one data center.

A significant part of the cost equation was the impact on software costs due to the extra overhead caused by the elongated synchronous response time. I had to assume a much smaller distance between the machines to get the propagation delay down to something palatable. Even still, the software cost impact would have been significant.

In the end, I recommended that we simply add GDPS to our existing DR solution to automate the process of restarting the systems in the DR environment. We already have all of that hardware in place, and have regularly demonstrated successful recovery and application restart in that environment using our manual procedures. Automating it would make it all the better.

Once again, I found myself wishing for coupling links that could utilize quantum entangled photons to break the speed-of-light barrier. Supposedly this is not possible. But pretty much everything about quantum mechanics seems impossible to me. So if such technology ever becomes reality, remember that I predicted it! I also predict that those links will be a premium priced option.

Speaking of pricing, there was another thing I learned this month. If you use On/Off Capacity on Demand (OOCoD) on your mainframe systems, IBM announcement 614-001[4] has effectively raised the price of that as you will now have to pay an additional maintenance for capacity invoked via OOCoD. Previously, on recent machines, if you had purchased more capacity than was delivered as permanent capacity, you could use OOCoD to increase the capacity to some level less than the pre-purchased level without incurring any additional hardware charges. Of course, additional software charges may apply. After April 8th, there will also be an additional maintenance cost. My guess is that in many situations it will not be significant, but it is something to be aware of if your capacity plans involve utilizing OOCoD.

So that's what I learned this month. And if you are interested in a little more information about why I learned about pony motors, see the first 2.5 minutes or so of this video: http://www.youtube.com/watch?v=umIfOpXKleo (it involves explosions, fire, and shattered manhole covers!)

As always, if you have questions or comments, you can reach me via email at sachapman@aep.com.

[4] See http://www-01.ibm.com/common/ssi/cgi-bin/ssialias?infotype=AN&subtype=CA&htmlfid=897/ENUS614-001&appname=USN