What I Learned This Month: Specialty Engine Savings Challenges

Scott Chapman
American Electric Power

Mainframe customers are always looking to save money on their software costs. Many (most) vendors use a model that is based on an arbitrary capacity rating (MIPS or MSUs) for the machine model that the software is installed on. Because the numbers involved are so large (hundreds of thousands or even millions of dollars *per month*), significant effort is spent trying to manage those costs. Since CPU utilization drives installed capacity, which drives software costs, mainframe capacity planning and performance tuning are important!

One of the techniques to help control the software cost is to move some processing to the zIIP or zAAP specialty engines. These CPUs are the same as the general purpose CPUs (aka, GCP, CP, or just CPU), but the zIIPs and zAAPs don't change the capacity rating used for determining software costs. If you can move work from the GCPs to the specialty engines, you may be able to lower your software costs.

Originally, zAAPs were only allowed to run Java code. The zIIPs were introduced to do more general system related (not application) work. The critical restriction is that for code to run on a zIIP, it must invoke a special license-restricted API. IBM doesn't license that API to customers, only to other software vendors (ISVs). The API also imposes limitations, such as code running on a zIIP can't do I/O[1]. Java applications just run on the zAAPs without any recoding work needed. As I understand it, reworking a product to run on zIIPs can be a significant undertaking.

It might be good to note here that IBM is phasing out zAAPs as a separate specialty engine and allowing zAAP-eligible workload to run on zIIPs. The next generation of mainframe will reportedly not support zAAPs, but I talk about them separately here because many shops (including mine) still have separate zAAPs and zIIPs.

Vendors are adding support for running at least a portion of their product on the zIIPs. In some cases they are coming out with specific applications that will do a certain type of processing on zIIPs. They tout this as having great potential for saving money. And well they should—shifting workload from GCPs to specialty engines could save the customer significant money. But there are a number of details that can make achieving any savings challenging. Before you jump into a proof of concept you should step back and remember some of the key facts about how software costs typically work on the mainframe.

---

[1] Or so we were told when the zIIPs were introduced. I don't know the actual technical details because I'm not a software vendor and hence haven't licensed the API from IBM. And if I had, I probably couldn't tell you about it.

First, most ISVs will not lower your software bill even if you reduce the capacity you have installed. IBM OTC[2] software seems to be handled similarly. However, many sites are using one of the variable workload licenses for their IBM MLC[3] software. These licenses (VWLC, AWLC, etc.) are based on the monthly utilization of the installed capacity. In theory, if you can lower your utilization, you can save money on your IBM MLC software. Unfortunately, it is not quite that simple.

The first issue is that the monthly metric used for determining your utilization is the peak rolling four-hour average utilization. We often abbreviate this simply "R4H". Depending on your utilization profile, you may hit that peak R4H value every day at a certain time of day or maybe you only hit it on a certain few days of the month. Or maybe you hit it on just a single spike per month when you're running some monthly process. In any case, reducing the utilization outside of the R4H peak(s) will not reduce your MLC charges.

So the very first thing you need to do is limit the timeframe(s) that you're looking at to just your peak R4H periods. Remember that if you have multiple periods that reach the same R4H value, you have to reduce all of them to reduce your MLC charges because a single peak sets the cost. Also, because it's a rolling 4 hour average, you need to include the hours immediately prior to the peak as well.

Once you have found the periods of interest, you need to determine how much of the utilization in those periods is from the workload that is proposed to be offloaded. Convert that percentage to MSUs. For example, if your peak R4H is 400 MSUs, and during those peak times the workload that's proposed to be offloaded is 5% of *all* the GCP CPU time during those intervals, then you have a potential savings of 20 MSUs.

Once you have a potential MSU savings, you need to figure out how much money you might save by eliminating those MSUs. When you are looking at this you cannot just apply the percentage to your overall MSU bill, because the cost decreases per MSU as you license more. So reducing your MSUs 5% may only save you 2.5% off your bill. This calculation gets much more complicated if you have multiple machines and multiple LPARs, especially if the licensed MSU software is not identical across all LPARs and machines. In short, you need to actually price out your configuration at the different utilization levels to figure out what the costs are so you can figure out the potential savings.

If you are investigating purchasing a product to offload work with the intent of saving money, hopefully the cost of the product works out to be less than the MLC savings.

---

[2] One Time Charge: typically products you pay once to license, then pay an annual support cost. These products tend to compete fairly directly with ISV offerings.

[3] Monthly License Charge: these products don't have an upfront license cost, but instead you pay a monthly license fee.

If it looks like the product can save you money, you finally need to consider whether you are within some sort of broader agreement with IBM, such as an Enterprise License Agreement (ELA). These agreements, while generally similar to each other, are individually negotiated; you need to look at the specific terms of your agreement. It is common for these agreements to specify how much you will spend on MLC each month. Reducing your MLC below that agreed-upon amount likely will have limited benefits. The full ELA planning and management story is much more complicated than this simple explanation. But the important note here is that if you're within an ELA, any reduction in your MLC costs may not yield real cost reductions until after the ELA expires.

To summarize, if a vendor approaches you with a product that will move some utility function off to your zIIPs to save money, take a close look at all the above items to make sure it really will save you money.

However, I would be remiss if I didn't mention that moving workloads to specialty engines can have performance benefits as well as cost benefits. In particular, if you are running a machine where the GCPs don't run full speed, then the fact that the specialty engines run full speed may provide a nice performance boost to workloads moved there. If you are running out of capacity in your peak times and you move some workload to the specialty engines, something else will likely consume the freed capacity. While that may result in no cost savings, it might be an important performance improvement for the workload that gets access to the freed capacity.

I was recently asked what I thought about a particular product that would move a certain utility function to the zIIPs. A cursory look indicated the function in question was approximately one-half of one percent of our total CPU, so there is not a lot of savings potential. We would almost certainly never notice that savings: something else would consume it. Even optimistically assuming we did reduce our peak 0.5%, that savings would be on the order of hundreds, not thousands, of dollars per month. And we are in an ELA so we likely would not realize any real savings for a couple of years.

Most people would agree that this is much more complicated than it should be. But real reform in mainframe software costs probably won't happen unless all the vendors (IBM and the ISVs) sit down and agree to throw out the antiquated concept of basing software costs on some arbitrary[4] machine capacity measurement. Unfortunately, I don't think that is likely anytime soon.

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

---

[4] "Arbitrary" because how much work a customer can get out of a given machine model is highly dependent on many things other than just the number and speed of the CPUs. That's why IBM says "don't use MIPS charts for capacity planning". This variability is only going to get worse as the CPU technology gets more sophisticated and a single capacity number (either MIPS or MSUs) for a machine will be rendered increasingly meaningless for capacity planning. Just to be clear, I don't believe I have a perfect answer for how software should be charged. But there must be a better solution than our current situation.