PVS vs. MCS Revisited

**By** [Nick Rintalan](http://blogs.citrix.com/author/nicholasr/) · **Published August 12, 2013**·

It’s been a few months since my last article, but rest assured, I’ve been keeping busy and I have a ton of stuff in my head that I’m committed to getting down on paper in the near future. Why so busy? Well, our Mobility products are keeping me busy for sure. But I also spent the last month or so preparing for 2 different sessions at [BriForum Chicago](http://briforum.com/). My colleague, [Dan Allen](http://blogs.citrix.com/author/danal/), and I co-presented on the topics of IOPS and Folder Redirection. Once Brian makes the videos and decks available online, I’ll be sure to point people to them.

So what stuff do I want to get down on paper and turn into a future article? To name a few…MCS vs. PVS (revisited), NUMA and XA VM Sizing, XenMobile Lessons Learned “2.0″, and Virtualizing PVS Part 3. But let’s talk about that first topic of PVS vs MCS now.

Although BriForum (and Synergy) are always busy times, I always try to catch a few sessions by some of my favorite presenters. One of them is [Jim Moyle](http://www.jimmoyle.com/) and he actually inspired this article. If you don’t know Jim, he is one of our CTPs and works for Atlantis Computing – he also wrote one of the most [informative papers on IOPS](http://www.jimmoyle.com/2011/05/windows-7-iops-for-vdi-deep-dive/) I’ve ever read. I swear there is not a month that goes by that I don’t get asked about PVS vs. MCS (pros and cons, what should I use, etc.). I’m not going to get into the pros and cons or tell you what to use since many folks like Dan Feller have done a good job of that already, even with [beautiful decision trees](http://virtualfeller.com/2011/03/09/pvs-or-mcs-%E2%80%93-what%E2%80%99s-your-decision-going-to-be/). I might note that Barry Schiffer has an [updated decision tree](http://www.barryschiffer.com/provisioning-services-vs-machine-creation-services-2013-revision/) you might want to check out, too. But I do want to talk about one of the main reasons people often cite for not using MCS – it generates about “**1.6x or 60% more IOPS compared to PVS**“. And ever since [Ken Bell sort of “documented” this](http://blogs.citrix.com/2011/02/17/mcs-or-pvs-what-should-i-be-using/) in passing about 2-3 years ago, that’s sort of been Gospel and no one had challenged it. But our CCS team was seeing slightly different results in the field and Jim Moyle also decided to challenge that statement. And Jim shared the results of his MCS vs. PVS testing at BriForum this year – I think many folks were shocked by the results.

What were those results? Here is a summary of the things I thought were most interesting:

* MCS generates **21.5% more average IOPS** compared to PVS in the **steady-state** (not anywhere near 60%)
* This breaks down to about **8% more write IO** and **13% more read IO**
* MCS generates **45.2% more peak IOPS** compared to PVS (this is closer to the 50-60% range that we originally documented)
* The read-to-write (R/W) IO ratio for **PVS** was **90%+ writes in both the steady-state and peak** (nothing new here)
* The R/W ratio for **MCS at peak** was **47/53** (we’ve long said it’s about 50/50 for MCS, so nothing new here)
* The R/W ratio for **MCS in the steady-state** was **17/83** (this was a bit of a surprise, much like the first bullet)

So how can this be?!?

I think it’s critical to understand where our initial “1.5-1.6x” or “50-60%” statement comes from – that takes into account not just the steady-state, but also the boot and logon phases, which are mostly read IOPS and absolutely drive up the numbers for MCS. If you’re unfamiliar with the typical R/W ratios for a Windows VM during the various stages of its “life” (boot, logon, steady-state, idle, logoff, etc.), then this picture, courtesy of [Project VRC](http://www.projectvrc.com/), always does a good job explaining it succinctly:



We were also looking at peak IOPS and average IOPS in a single number – we didn’t provide two different numbers or break it down like Jim and I did above in the results, and a single IOPS number can be very misleading in itself. You don’t believe me? Just check out my BriForum presentation on IOPS and I’ll show you several examples of how misleading statements like “1 million IOPS” can be!

So there you have it – things are looking up for MCS. And in reality, I think MCS sort of a got a bad name early-on and no one ever bothered to look at real-world data or re-test like Jim did. Score one for the CTPs.

Does this mean we should all start leveraging MCS? Not necessarily – we have thousands of customers using PVS in a production capacity and I just can’t say the same for MCS right now. But does this mean we should give MCS another look? Absolutely. Especially as companies like MSFT and VMW build native support for read caching into their hypervisors to handle the read IOPS…because once we get rid of those extra read IOPS, we’re just left with an almost-negligible 8% more write IOPS in the steady state…and that’s when the simplicity of MCS starts to look pretty attractive.

Hope this helps. Again, special thanks to Jim Moyle for most of this data and for helping me bust a long-standing myth!

Cheers, Nick

Nick Rintalan, Lead Architect, Citrix Consulting