In the Age of Software Defined Everything Does Your Device Architecture Matter?

Robert Haynes, 2014-02-07

With the big focus on abstraction and automation in cloud and next generation datacenter designs it sometimes feels like we've forgotten what it is that we are orchestrating and managing. Does the underlying architecture of devices still matter?

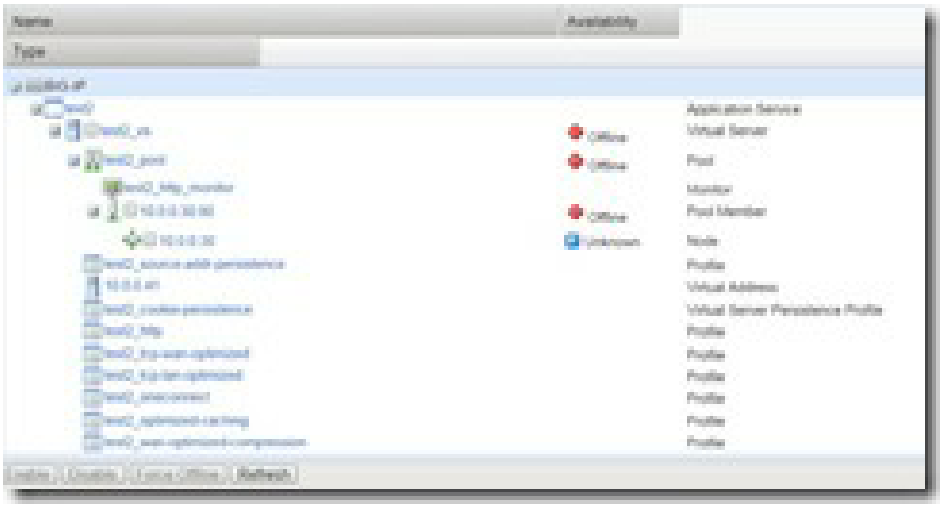
Who cares what's inside the box? It's a debate that I've been having within my corner of F5 (and to give you a clue about my life this corner isn't in executive boardrooms with picture windows and great views of Mount Rainier, but in beige cubicles surrounded by the cardboard coffins of long outdated BIG-IP models form the late '90's and empty packets of esoteric Japanese candy).

My position is that the under the hood stuff really matters – it's all about spin vs content. I'd like to say I've managed to convince some of my more "marketingingy" colleagues of the legitimacy of my position, but they keep asking why 'the business' should care and telling me I'm getting 'down in the weeds'. I've been looking for a strategy to convert them now that we've gone digital and simply holding their crayons to ransom won't work. Fortunately the marketing side of my role has come to my rescue (nothing like turning their own

Technorati Tags: archtecture
weapons against them). All great movements need a story behind them and all great stories need a hero and a villain.

Let me introduce you the hero of our tale, the F5 distributed compute architecture – clustered multi-processing (CMP), and it's ancient foe – Shared Memory Architecture (SMA).

Essentially a shared memory architecture is pretty much just that – a design where several processors access a globally shared memory:



On the other hand the F5 clustered multi-process architecture creates isolated microkernel instances with dedicated memory managed by an upstream disaggregation layer:

Each CPU hosts an independent Traffic Management Microkernel (TMM) instance and receives traffic to process from the disaggregation layer. Communication between TMM \instances is kept to a minimum.

I can see why you might not be as filled with anticipation as I am - this is low level architecture design stuff – far removed from the traditional topics of business value and application fluency I'm usually paid to talk about. And yet I'm convinced this really matters. The question is: can I prove it to you (and hence win a small side bet that will save me from buying beer at the next team meeting)? Can my hero, pulled from the massed ranks of technology details, prevail?

So let's work back from the *things you should care about* in your infrastructure and see what our champions can do with them.

Scalability

Organizations need infrastructure that scales and scales predictably, since traffic volumes and processing requirements are only going to grow. It's no good having systems that become inefficient at larger scale or that lock you into a maximum device size or an outmoded infrastructure design. That places barriers to expansion and innovation.

The trouble with shared memory design is that they just don't scale up to the levels you might need – at least that's what intel say. As CPU core count rises contention for access to the memory bus begins to choke performance. Shared memory designs are by their very nature monolithic – you are never going to be able to break out of a single device. Distributed compute designs have been shown to scale to the very largest workloads (Hadoop anyone?).

So what's better – a system that can't really scale beyond about 32 cores, can't break out of a single device and has diminishing returns as CPU count goes up, or an architecture that scales linearly (try saying that after a lunchtime 'planning session') and can be broken out into 'webscale' distributed compute models? Which one won't let you down when your needs grow?

Performance

If your infrastructure does not deliver the performance you need you're going to lose reputation, productivity and customers. No one likes slow applications . High performance means different things to different workloads. You might need very low, very predictable latency the ability to handle millions of connections or capacity to stream 100's of Gb of video content. What you don't need is an architecture that has bottlenecks and performance choke points.

CPU's in shared memory systems need to spend a lot of time and effort keeping in-sync with each other, and can end up waiting for the state to clear in other processors. It's like that project plan that has too many dependencies – one task is delayed and everything grinds to a halt. CMP fully parallelizes processing, and only needs the occasional check in between units. The end result is greater performance, especially at scale.

Reliability

Your IT infrastructure is probably mission-critical to your business. If it fails you are going to be, at the least, inconvenienced and it's probably going to be a lot worse.

Despite excellent levels of reliability, hardware components fail. That fact of life is mitigated by software systems that create clusters or other high availability configurations. Unfortunately software also fails – and that's harder to mitigate against. Assuming all manufacturers have rigorous QA and software quality practices two of the key factors contributing to software reliability are simplicity and ease of debugging. Software must be easy to debug and fix when it breaks.

Shared memory architectures are more complex and harder to debug due to synchronization and cache coherency issues. Distributed architectures are both simpler and have the ability to be more hardware fault tolerant (for example if a blade in a F5 Viprion chassis were to fail, the system can maintain service with only minimal disruption).

So if you want the best scalability, availability and performance, and the benefits that's bring to your business, then the grimy details of what's under the hood matter, and matter a lot.

F5 Networks, Inc. | 401 Elliot Avenue West, Seattle, WA 98119 | 888-882-4447 | f5.com

F5 Networks, Inc. Corporate Headquarters info@f5.com	F5 Networks Asia-Pacific apacinfo@f5.com	F5 Networks Ltd. Europe/Middle-East/Africa emeainfo@f5.com	F5 Networks Japan K.K. f5j-info@f5.com