Containers—Not Virtual Machines—Are the Future Cloud

Jun 17, 2013 By David Strauss (/users/davis-strauss)
in

Cloud infrastructure providers like Amazon Web Service sell virtual machines. EC2 revenue is expected to surpass $1B in revenue this year. That's a lot of VMs.

It's not hard to see why there is such demand. You get the ability to scale up or down, guaranteed computational resources, security isolation and API access for provisioning it all, without any of the overhead of managing physical servers.

But, you are also paying for lot of increasingly avoidable overhead in the form of running a full-blown operating system image for each virtual machine. This approach has become an unnecessarily heavyweight solution to the underlying question of how to best run applications in the cloud.

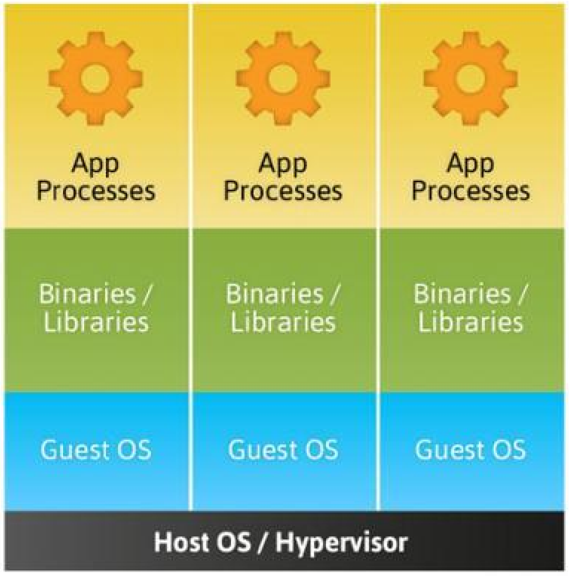

Figure 1. Traditional virtualization and paravirtualization require a full operating system image for each instance.

Until recently it has been assumed that OS virtualization is the only path to provide appropriate isolation for applications running on a server. These assumptions are quickly becoming dated, thanks to recent underlying improvements to how the Linux kernel can now manage isolation between applications.

Containers now can be used as an alternative to OS-level virtualization to run multiple isolated systems on a single host. Containers within a single operating system are much more efficient, and because of this efficiency, they will underpin the future of the cloud infrastructure industry in place of VM architecture.

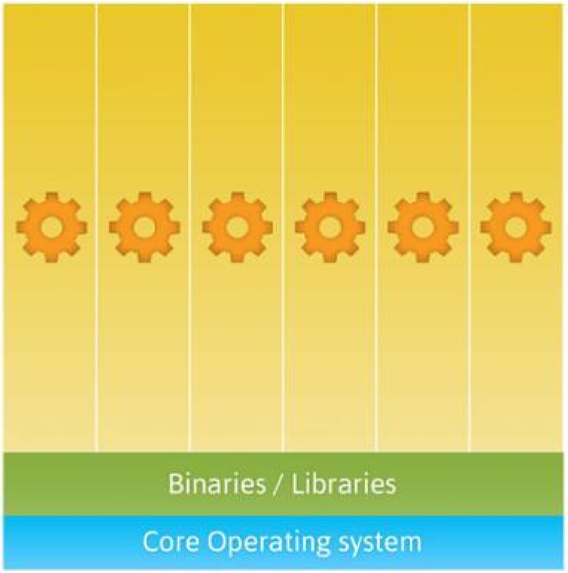

Figure 2. Containers can share a single operating system and, optionally, other binary and library resources.

How We Got Here

There is a good reason why we buy by the virtual machine today: containers used to be terrible, if they existed in any useful form at all. Let's hop back to 2005 for a moment. "chroot" certainly didn't (and still doesn't) meet the resource and security isolation goals for multi-tenant designs. "nice" is a winner-takes-all scheduling mechanism. The "fair" resource scheduling in the kernel is often too fair, equally balancing resources between a hungry, unimportant process and a hungry, important one. Memory and file descriptor limits offer no gradient between normal operation and crashing an application that's overstepped its boundaries.

Virtual machines were able to partition and distribute resources viably in the hypervisor without relying on kernel support or, worse, separate hardware. For a long time, virtual machines were the only way on Linux to give Application A up to 80% of CPU resources and Application B up to 20%. Similar partitioning and sharing schemes exist for memory, disk block I/O, network I/O and other contentious resources.

Virtual machines have made major leaps in efficiency too. What used to be borderline-emulation has moved to direct hardware support for memory page mapping and other hard-to-virtualize features. We're down to a CPU penalty of only a few percent versus direct hardware use.

David Strauss is the CTO and co-founder of Pantheon, whose all-for-one-and-one-for-all improvements to the Drupal infrastructure have made the largest Drupal Web sites in the world more scalable and secure.