Containers and The Cloud
A Match Made in Heaven

James Bottomley
CTO, Server Virtualization

15 May 2013
Virtualisation History

• Shortly after the formation of VMware, the enterprise “discovered” virtualisation
  – Essentially began life as the way to run testing and production on the same systems

• Virtualisation was so much hyped that in 2005, the Linux Kernel was forked by the distributions to accommodate Xen.
  – The consequences of this fork remain with us to this day

• Hypervisor based Virtualisation blinkered everyone to other technologies
  – If it's not a hypervisor, the Enterprise isn't interested
Exploring other Virtualisation choices

• While the enterprise charged down the Hypervisor alley, the Hosting market turned to Containers

• Primary benefit was density:
  – Containers can support up to 3x the Virtual Private Server (VPS) density of hypervisors
  – In the hosting market, Margins are thin, so density makes the difference between a profit and a loss

• As the hosting market evolved, container technology became more sophisticated
  – And more dense
Also the Containers Technology Split

- Google saw the advantages of containers early on
- They hired the main Cgroup containers person (Paul Menage)
- And set about fully containerising their main data centres.
- So historically, the Hosting Containers (OpenVZ) and the Cgroups containers evolved separately
  - And actually large pieces of both are out of the mainline
Examining Containers and Hypervisors

• Containers
  - Share the same kernel
  - Communicate with the host via standard system calls
  - Are as isolating as hypervisors

• Hypervisors
  - Run separate copies of the kernel
  - Communicate with the host via virtual machine emulation
Why are Containers Denser?

• Partly it's simple mathematics of resource savings
  – Containers run only one kernel per physical system
  – Hypervisors run one kernel per virtual Machine (plus one for the host)
• But also it's to do with better management
  – The single host kernel in the Container case has complete visibility into the resource usage in the entire system (including what every container is doing)
  – Hypervisors have to invent all sorts of communication paradigms for the multiple kernels to inform the host what they're doing (ballooning, KSM, performance counters)
Chart of Densities

Parallels Plesk Panel density

- **Citrix XEN 6** up to 29 VMs
- **RHEL6 KVM** up to 37 VMs
- **OpenVZ** up to 57 CTs
- **PCS 6.0 [CT]** up to 121 CTs

Acceptable response time = 4 sec
Other Container Advantages

• It didn't really matter much to hosters, but containers are far more elastic than Hypervisors
  - Since the kernel is already booted, even if you have different operating systems (sharing the same kernel) going through a boot sequence, it's still faster than booting a kernel from scratch
  - If your operating systems are homogeneous (same version of Linux) then booting is instant and you can just start your applications.
Orchestration Benchmarks

Containers respond 7x faster than Hypervisors.
All of this is Tailor Made for the Cloud

• The prime requirements of cloud are
  – Elasticity
  – Density (for running a lean data centre)

• And the big disadvantage of Containers no longer matters
  – All cloud instances are homogeneous (it's about the applications)
  – So running different kernels (or even different operating systems) is not required

• Makes containers a much more ideal technology for the cloud
But there's more

• Main cloud PaaS/SaaS issue is Tenancy
  – How do you take an application and share it securely between multiple users (tenants) such that they're all completely isolated from each other

• Take Gmail: would you trust it if you could read other people's email?

• But this problem is what containers have been doing for years
  – Using the same kernel (and even same base OS) isolate applications from each other and even provide them with different isolated storage
Conclusion

• Containers not only provide up to three times the density of hypervisors
• And can be orchestrated orders of magnitude faster
• They also come with the tools and controls for solving the application tenancy problem easily
• And all this from a single image
• When it comes to the cloud, containers are an ideal technology
• Far more ideal than hypervisors
But Don't Take my Word for it

- All the smart Cloud Service installations are using Containers
  - Google Search, Gmail and web services is entirely containerised
    > When you fire up a search or log into Gmail, you get your very own container running in the google datacentres
  - So is Facebook
  - The distributions are jumping on the bandwagon too
    > Canonical/Ubuntu and Debian have had containers for a long time
    > SUSE has had the ability to build container images via studio for some time
    > Red Hat recently adopted them as part of its cloud strategy
Containers on Linux
A brief history of Containers in Linux

• Containers technology in the Linux Kernel came from three separate sources
  – Virtuozzo (OpenVZ) from Parallels
  – Cgroups from Google
  – Dpar from IBM
  – Namespaces from Eric Biederman

• Of those, Only OpenVZ, Cgroups and Namespaces are being pushed today.

• Have two competing control planes: LXC tools and VZ tools.
Where we're Going

• All containers contributors agree on selecting and enhancing the best available technologies
  - No re-run of the damaging Xen/Kvm wars which set Linux hypervisor technology back years.
  - There will be only one container technology in Linux and it will be the best

• All out of tree technologies will be moved in-tree (or replaced with better in-tree components)
  - Parallels is fully committed to pushing the remaining OpenVZ patches upstream
  - Most of OpenVZ was already upstream by 3.8 kernel (18 Feb 2013)
  - Memory density technologies targetted for 3.11
Next up: The Control Plane

- Now that we're all relying on identical kernel pieces to produce containers, it doesn't make much sense to use separate tool chains to achieve the same effect.
- Unifying lxc tools and vz tools will be discussed at Linux Plumbers Conference in September.
- Object is to see if we can produce a single tool set that will control all of containers in Linux.
- This would basically make containers technology in Linux completely unified for all use cases.
- It's not as important as the Kernel piece, but unification here is still useful.
The End Result

• Linux will have the best container technology available in any operating system
• Indeed, most Operating Systems have already missed the boat
  – Windows has no container technology
  – Neither does macOS (bsd)
  – Only legacy operating systems with containers are Solaris and AIX
Conclusions

• Containers are the new virtualisation technology for cloud services
• Anyone with cloud service plans should be looking into them
• Linux will have the best container technology of any operating system on the planet
• So Linux will be the best platform, both now and in the future, for cloud services.
Questions ?