

Kap Temelli Özgür Sanallaştırma Çözümleri

Erdem Bayer
ebayer@bayer.gen.tr
ebayer@pardus.org.tr

Hosting Zirvesi '11 - İKÜ

Sunum Planı

- Kap (Container) Sanallaştırma
- Artıları / Eksileri / Limitleri
- Çözümler
 - Linux Vserver
 - OpenVZ
 - LXC (Linux Containers)
- Örnek: lxc

Kap Temelli Sanallaştırma

- İşletim sistemi kernel'inin aralarındaki izolasyonu sağlayarak birden fazla kullanıcı- alanı (userspace) yaratabilme yeteneğidir.
- İşletim sistemi kernel'i isimalanı ayırma yeteneğini kullanarak birden fazla init süreci başlatır.
- Her init süreci kendi adres alanına sahiptir, dolayısı ile bir kap içindeki yazılımlar diğer kap içindeki yazılımlara müdahale edemez.

Kap Temelli Sanallaştırma

- İleri seviye chroot mekanizması gibi düşünülebilir.
- Kernel kaynak yönetimi ile kapların kullandıkları kaynaklar kısıtlanabilir.
- Her kap temelli sanallaştırma çözümü kendi kernel yaması ve yönetim araçları ile gelir.

Kullanım Alanları

- VPS, Virtual Hosting
- Yarı / Tam Sanallaştırma çözümlerinin getirdiği yüke ihtiyaç olmayan, ancak yine de izolasyon ve kaynak yönetimi gerektiren tüm durumlar

Neden?

- Kartaca (<http://www.kartaca.com>) 'da çalışırken Linux-VServer ile tanıştım.
- Linux-Vserver kullanılmasının sebebi ihtiyaçlardan çok alışkanlıklar ile ilgiliydi.
- İzolasyon ayarları doğru yapılmadığından bir kap içinde çalışan uygulamalardaki hatalar diğer kapların kullanımını etkiliyordu.
- Pardus (<http://www.pardus.org.tr/>) projesi kapsamında kullanılan paket derleme çiftlikleri eldeki kaynakların atıl kullanılmasına sebep oluyordu.
- Hem kaynakları koruyacak, hem de dağıtıma minimum bakım ve işletim maliyeti getirecek bir sanallaştırma çözümüne ihtiyaç vardı.

Neden?

- Hipervizör tabanlı sanallaştırma çözümü – KVM
- Paket Derleme işlemlerinde ciddi zaman kaybı
- Testler:
 - Hyperthreading: açık / kapalı
 - Host IO Scheduler: cfg / deadline

Neden?

- Hyperthreading on – IO Scheduler cfg
 - Fiziksel Makine
 - User time (seconds): 2847.81
 - System time (seconds): 1371.26
 - Percent of CPU this job got: 244%
 - Elapsed (wall clock) time (h:mm:ss or m:ss): 28:48.32
 - Sanal Makine
 - User time (seconds): 2140.98
 - System time (seconds): 792.75
 - Percent of CPU this job got: 96%
 - Elapsed (wall clock) time (h:mm:ss or m:ss): 50:39.80

Neden?

- Hyperthreading off – IO Scheduler cfg
 - Fiziksel Makine
 - User time (seconds): 2792.07
 - System time (seconds): 1334.44
 - Percent of CPU this job got: 241%
 - Elapsed (wall clock) time (h:mm:ss or m:ss): 28:25.19
 - Sanal Makine
 - User time (seconds): 2162.67
 - System time (seconds): 794.07
 - Percent of CPU this job got: 96%
 - Elapsed (wall clock) time (h:mm:ss or m:ss): 51:07.56

Neden?

- Hyperthreading off – IO Scheduler deadline
 - Fiziksel Makine
 - User time (seconds): 2792.07
 - System time (seconds): 1334.44
 - Percent of CPU this job got: 241%
 - Elapsed (wall clock) time (h:mm:ss or m:ss): 28:25.19
 - Sanal Makine
 - User time (seconds): 2162.67
 - System time (seconds): 794.07
 - Percent of CPU this job got: 96%
 - Elapsed (wall clock) time (h:mm:ss or m:ss): 51:07.56

Artıları

- Hipervizör veya emülatör katmanının getirdiği yük yoktur. (Tüm izolasyon kontrolleri context switch gerekmeden kernel seviyesinde yapılır)
- Aynı dosya sistemini paylaşabilme yeteneği (cow hard linkler ile)
- Kaplar aynı fiziksel kaynakları kullanır.
- Ağ yönetimi izolasyon temellidir. (sanallaştırmanın getirdiği fazladan paket yükü yoktur.)
- Ortak kernel, ortak modüller
- Kaplar farklı linux dağıtımları olabilir.

Artıları

- Kaynak kontrolü kernel tarafından sağlandığı için kullanılmayan kaynaklar diğer kaplar arasında paylaşılabilir.
- Kaplar kolayca yedeklenebilir ve taşınabilir.
- Kullanıcı açısından bakıldığında kap içinde çalışan işletim sisteminin fiziksel kurulumdan farkı yoktur. (hemen hemen)
- Fiziksel kurulumda çalışan çoğu yazılım kap içine kurulduğunda da çalışabilir.

Eksileri

- Ortak kernel, ortak modüller
- Tüm kaplar linux olmak zorundadır.
- Donanım sanallaştırması yerine izolasyon kullanılır.
- Ağ yönetimi izolasyon temellidir. (Kaplar kendi ağ, yönlendirme ve paket filtreleme mekanizmalarını kullanamazlar)
- Kap içinde çalışacak kurulumlar elden geçirilmelidir.

Eksileri

- Kaplar içinde çalışacak linux kurulumları çalışan kernel ile uyumlu olmak zorundadır.
- Çoğu kap teknolojisi kernel yaması şeklinde gelir.
- Her kap teknolojisi kendi yönetim araçları ve ayar dosyaları ile gelir.
- Kaplar içinde donanım kullanımı kısıtlıdır.
(udev, hwclock, vs)

Yönetim Araçları

- Her kap teknolojisinin kendi komut seti ve ayar isimlendirmesi vardır.
- OpenVZ ve LXC kapları Libvirt Sanallaştırma API'si (<http://www.libvirt.org/>) ile yönetilebilmektedir.

Linux-VServer

- Jacques Gélinas tarafından 2001 yılında başlatılmış.
- Şu an Herbert Pötzl tarafından geliştiriliyor.
- Kernel yaması ve yönetim araçlarından oluşur.
- Stabil sürüm 2.6.22.19, geliştirme sürümü 2.6.38-rc2 kernel yaması.

Linux-VServer

- Hashify uygulaması ile kaplar arasında dosya paylaşımı (Kaplar içinde içeriği aynı olan dosyalardan sadece bir tane tutulur.)
- Ipv6 uyumluluğu için fazladan yama gerekiyor. (util-vserver >= 0.30.212)
- Sorunlu uygulamalar: hylafax, screen, asterix, samba, zimbra, vs

OpenVZ

- Parallels Virtuozzo Containers ürününün temeli
- 2.6.16 – 2.6.32 kernel yamaları
- Checkpoint ve Canlı göç desteği

Cgroups (Kontrol Grupları)

- Linux Kernel'inin bir grup sürecin kullanacağı kaynak (cpu, ram, disk io) miktarını sınırlama yöntemidir.
- 2006 yılında başlayan proje 2.6.24'den beri kernel içindedir.

LXC

- Diğer kap temelli sanallaştırma çözümlerine göre daha genç.
- Kontrol grupları (cgroups) ile kaynak yönetimi ve kernel isim alanları (namespaces) ile kaynak izolasyonu sağlar.
- 2.6.29'dan beri vanilla kernel içinde geliştiriliyor.