Linux®
Server™

Resource Efficient OS-Level Virtualization

DI Herbert Pötzl

1 Introduction

Computers have become sufficiently powerful to use virtualization to create the illusion of many smaller virtual machines, each running a separate operating system instance.

- Virtual Machines
- System Emulators
- Partitioning

2 The Concept

Virtual Servers do not necessarily require a separate operating system for each instance

Resources directly map to money – more servers require more CPU power, RAM, disk space, network bandwith and general I/O throughput.

Isolation allows to put several Servers on a Host, which will share the available resources efficiently.

2.1 Advantages

- ✘ Minimal Overhead
- ✘ Hardware Abstraction
- ✘ Shared Resources

2.2 Possible Drawbacks

- ✘ Kernel as Single Point of Failure?
- ✘ Kernel Security Issues?

3 Nomenclature

Host: the real or virtual machine running the Linux-VServer enabled Kernel.

Guest: the virtual private server (or short VPS) composed of a chrooted environment, isolated processes, and restricted IP ranges.

Context: the isolated and partially virtualized environment to which processes are *confined*.

4 Isolation vs. Virtualization

★ IP Layer **Network Isolation**

…instead of **Virtual Network Stacks**

★ **Namespaces** and **Shared Partitions**

…instead of **Virtual Filesystems**

★ **Accounting**, **Limits**, and **TB Scheduling**

…instead of **vResources** and **vCPUs**

4.1 Lightweight Guests

Isolation allows to have very small Guests (down to a single process) without creating measurable overhead.

4.2 Shared Services

Isolation areas can overlap (to some extend) and services can be shared between Guests

4.3 Flexible Resources

Because there is a common pool of Resources, and no static allocation to the Guests, they can be easily . . .

- ✘ adjusted and shared
- ✘ monitored on the Host System
- ✘ limited or extended

5 Optional Virtualizations

- ✘ Init PID(1) *[pstree, init]*
- ✘ Network Interface Information
- ✘ Memory Information *[free, meminfo]*
- ✘ Available Disk Space *[df]*
- ✘ System Uptime *[guest start]*
- ✘ System Load *[guest processes]*
- ✘ System Time *[adjustable]*

6 Field of Application

- ✘ Virtual Server Hosting
- ✘ Administrative Separation
- ✘ Service Separation
- ✘ Enhancing Security
- ✘ Easy Maintenance
- ✘ Fail-over Scenarios
- ✘ Simplified Testing

7 Existing Infrastructure

- ✘ Linux Capability System
- ✘ Resource Limits (ulimit)
- ✘ File Attributes (xattr)
- ✘ The chroot(1) Command
- ✘ Private Namespaces

8 Required Modifications

- ✘ Context Separation
- ✘ Network Separation
- ✘ The Chroot Barrier
- ✘ Upper Bound for Caps
- ✘ Resource Isolation
- ✘ Filesystem XID Tagging

9 Additional Modifications

- ✘ Context Flags
- ✘ Context Capabilities
- ✘ Context Accounting
- ✘ Context Limits
- ✘ Virtualization
- ✘ Improved Security
- ✘ Kernel Helper

10 Features and Bonus Material

- ✘ Unification
- ✘ CoW Link Breaking
- ✘ The Linux-VServer Proc-FS
- ✘ TB Per CPU Scheduler
- ✘ Context Disk Limits
- ✘ Context Quota and VRoot Proxy
- ✘ Information Isolation

11 Intrusiveness

patch	lines	chars	hunks	new
vs1.00	2845	95567	178	997
vs1.20	4305	131922	216	1857
vs2.00	19673	557988	856	8987
vs2.01	20300	572752	898	9362
vs2.02	21330	602493	977	9464
vs2.1.0	25948	759709	1222	10394
vs2.1.1	27754	809344	1279	11350
patch-2.6.11Δ	682454	21905964	23506	108544
patch-2.6.16Δ	937133	28148252	35302	162334

12 Non Intel x86 Hardware

- ✔ ia64, x86_64
- ✔ alpha, arm
- ✔ hppa, hppa64
- ✔ ppc, ppc64
- ✔ sparc, sparc64
- ✔ mips o/n32, mips64
- ✔ s390, s390x
- ✔ um, xen

Q & A

www: http://linux-vserver.org
irc: #vserver @ irc.oftc.net