

# Linux Virtualization

*Kir Kolyshkin*  
*OpenVZ project manager*

*Kirill Korotaev*  
*OpenVZ kernel project manager*

Linux Tag, 5 May 2006, Wiesbaden, Germany  
<http://www.linuxtag.org/>

# What is virtualization?

Virtualization is a technique for deploying technologies. Virtualization creates a level of indirection or an abstraction layer between a physical object and the managing or using application.

<http://www.aarohi.net/info/glossary.html>

Virtualization is a framework or methodology of dividing the resources of a computer into multiple execution environments...

<http://www.kernelthread.com/publications/virtualization/>

A key benefit of the virtualization is the ability to run multiple operating systems on a single physical server and share the underlying hardware resources – known as **partitioning**.

<http://www.vmware.com/pdf/virtualization.pdf>

# Ways to Virtualize

- Hardware Emulation
- Para-Virtualization
- Virtualization on the OS level
- Multi-server virtualization

# Hardware Emulation

a.k.a. VM (Virtual Machine)

– VMware



– QEmu



– Bochs



## Pros:

- Can run arbitrary OS, unmodified

## Cons:

- Low density/scalability
- Slow/complex management
- Low performance

# Para-virtualization

- Xen
- UML  
(User Mode Linux)



## Pros:

- Better performance

## Cons:

- Needs modified guest OS
- Static resource allocation, bad scalability, bad manageability

# OS Level Virtualization

- OpenVZ
- FreeBSD jails
- Linux-VServer
- Solaris Zones



## Pros:

- Native performance
- Dynamic resource allocation, best scalability

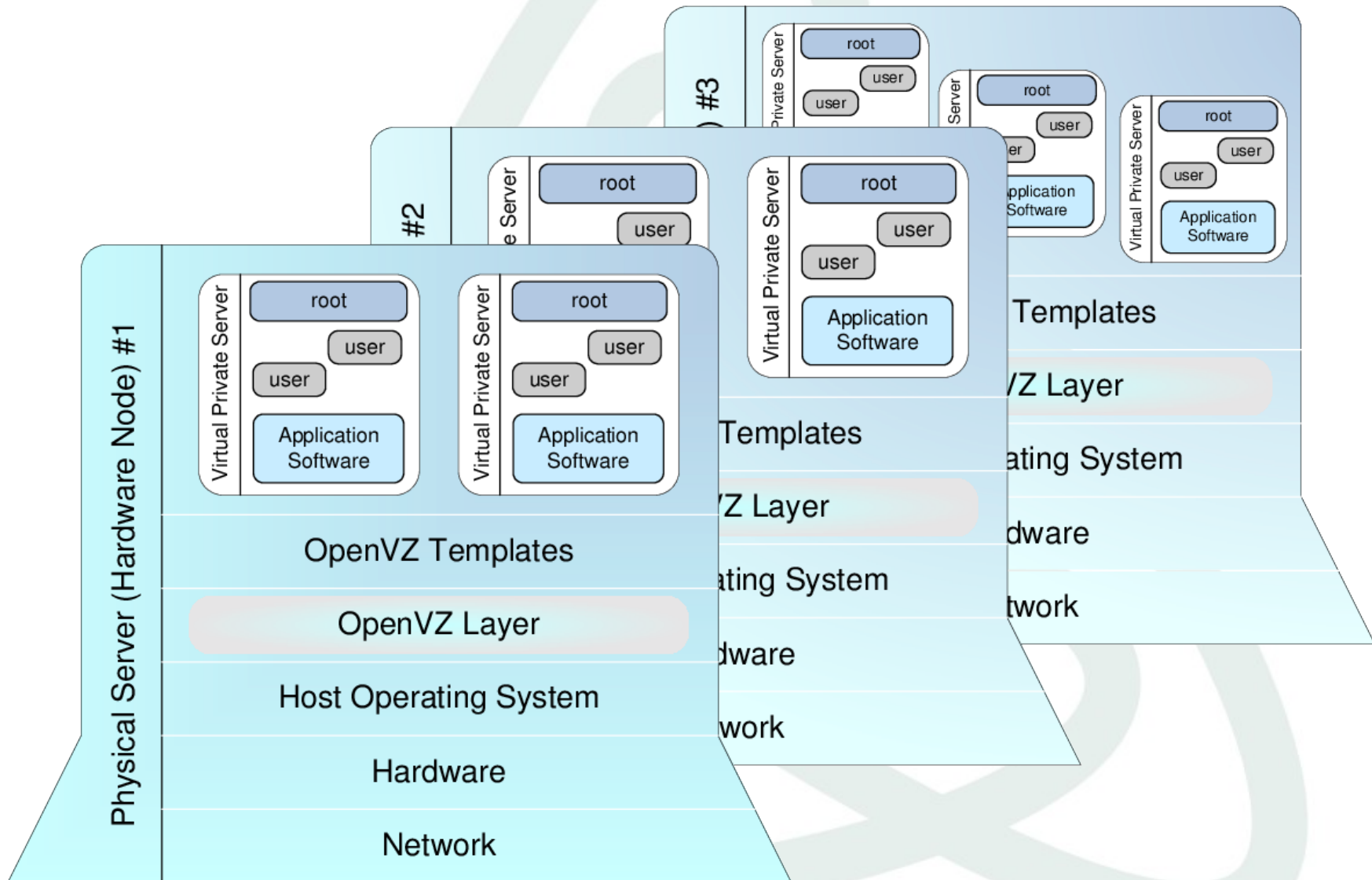
## Cons:

- Single (same) kernel per physical server

# OSs evolution

- **Multitask**  
many processes
- **Multuser**  
many users
- **Multiple execution environments**  
many virtual environments (VEs, VPSs, containers, guests, partitions, zones...)

# OpenVZ design approach





# OpenVZ: components

## Kernel

- Virtualization and Isolation
- Resource Management
- Checkpointing

## Tools

- vzctl: Virtual Environment (VE) control utility
- vzpkg: VE software package management

## Templates

- precreated VE images for fast VE creation

# Kernel: Virtualization & Isolation

Each virtual environment has its own

- **Files**  
System libraries, applications, virtualized /proc and /sys, virtualized locks etc.
- **Process tree**  
Featuring virtualized PIDs, so that the init PID is 1
- **Network**  
Virtual network device, its own IP addresses, set of netfilter and routing rules
- **Devices**  
Plus if needed, any VE can be granted access to real devices like network interfaces, serial ports, disk partitions, etc.
- **IPC objects**  
shared memory, semaphores, messages
- ...

# Kernel: Resource Management

Managed resource sharing and limiting.

- **Two-level disk quota** (first-level: per-VE quota; second-level: ordinary user/group quota inside a VE)
- **Fair CPU scheduler** (SFQ with shares and hard limits)
- **User Beancounters** is a set of per-VE resource counters, limits, and guarantees (kernel memory, network buffers, phys pages, etc.)

# Kernel: Checkpointing/Migration

- **Checkpoint:** VE state can be saved in a file
  - running processes, CPU registers, ...
  - opened files, signals, IPC, ...
  - network connections, buffers, backlogs, etc.
  - private/shared memory
- **Restore:** VE state can be restored later
- **Live migration:** VE can be restored on a different physical server

# Kernel: Security

- **Security in virtualization solutions**
- **What makes OpenVZ secure?**
  - Security model (deny by default)
  - Tracking mainstream security fixes
  - Stable kernel 2.6.8, code freeze, no **new** bugs
  - Security review by Solar Designer
  - Activity monitors
- **Practice (400,000 public VEs)**

# Tools: VE control

```
# vzctl create 101 --ostemplate fedora-core-5
# vzctl set 101 --ipadd 192.168.4.45 --save
# vzctl start 101
# vzctl exec 101 ps ax
```

PID	TTY	STAT	TIME	COMMAND
1	?	Ss	0:00	init
11830	?	Ss	0:00	syslogd -m 0
11897	?	Ss	0:00	/usr/sbin/sshd
11943	?	Ss	0:00	xinetd -stayalive -pidfile ...
12218	?	Ss	0:00	sendmail: accepting connections
12265	?	Ss	0:00	sendmail: Queue runner@01:00:00
13362	?	Ss	0:00	/usr/sbin/httpd
13363	?	S	0:00	\_ /usr/sbin/httpd
.....				
13373	?	S	0:00	\_ /usr/sbin/httpd
6416	?	Rs	0:00	ps axf

```
# vzctl enter 101
bash# logout
# vzctl stop 101
# vzctl destroy 101
```

# Tools: Templates

**# vzpkgls**

fedora-core-5-i386-default  
centos-4-x86\_64-minimal

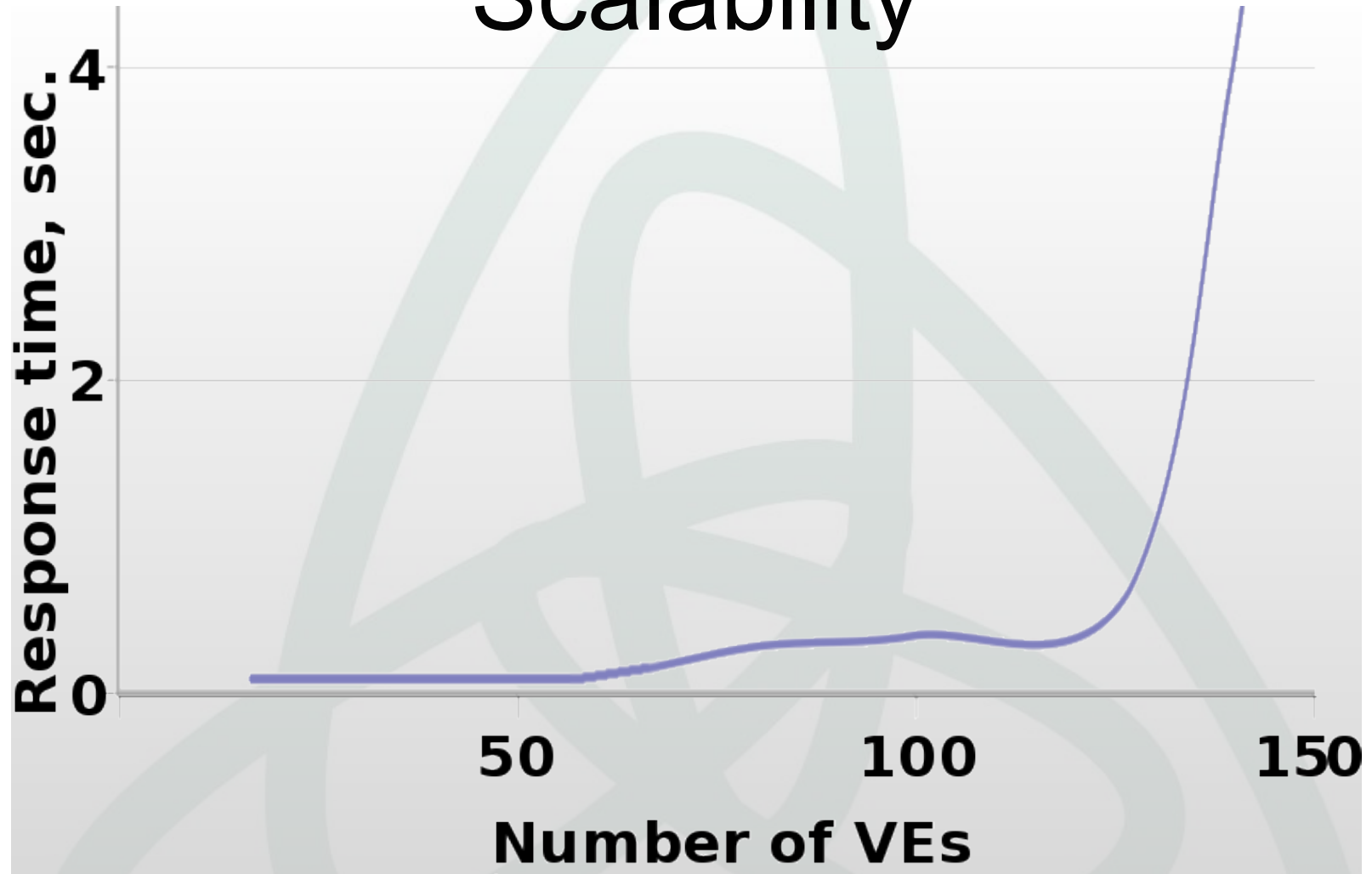
**# vzpkgcache**

(creates templates from metadata/updates existing templates)

**# vzyum 101 install gcc**

(installs gcc and its deps to VE 101)

# Scalability



768 ( $\frac{3}{4}$ ) MB RAM - up to 120 VEs  
2GB RAM - up to 320 VEs



# Usage Scenarios

- Server Consolidation
- Hosting
- Development and Testing
- Security
- Educational

# Server Consolidation

A bunch of servers:

- harder to manage
- upgrade is a pain
- eats up rack space
- high electricity bills

A bunch of VEs:

- uniform management
- easily upgradeable and scalable
- fast migration

# Hosting

- Web server serving hundreds of virtual hosts
  - Users see each other processes etc
  - DoS attacks
  - Unable to change/upgrade hardware
- Users are isolated from each other
  - VE is like a real server, just cheap
  - Much easier to admin

# Development & Testing

- A lot of hardware
- Zoo: many different Linux distros
- Frequent reinstalls take much time
- Fast provisioning
- Different distros can co-exist on one box
- Cloning, snapshots, rollbacks
- VE is a sandbox – work and play, no fear

# Security

- Several network services are running
- One of them has a hole
- Cracker gets through
- Whoops..."all your base are belong to us"
- Put each service into a separate VE
- OpenVZ creates walls between applications
- Added benefit: dynamic resource management



# Educational

- No root access
  - Frequent reinstalls
  - DoS attacks
- Everybody and his dog can have a root access
  - Different Linux distros
  - No need for a lot of hardware

# Future plans

- Stable 2.6.17 kernel
- Support for IPv6 and bridged networking
- VCPU affinity
- I/O scheduling based on CFQ
- Distribution-specific kernels (SUSE10, FC5)
- Work on mainstream kernel virtualization

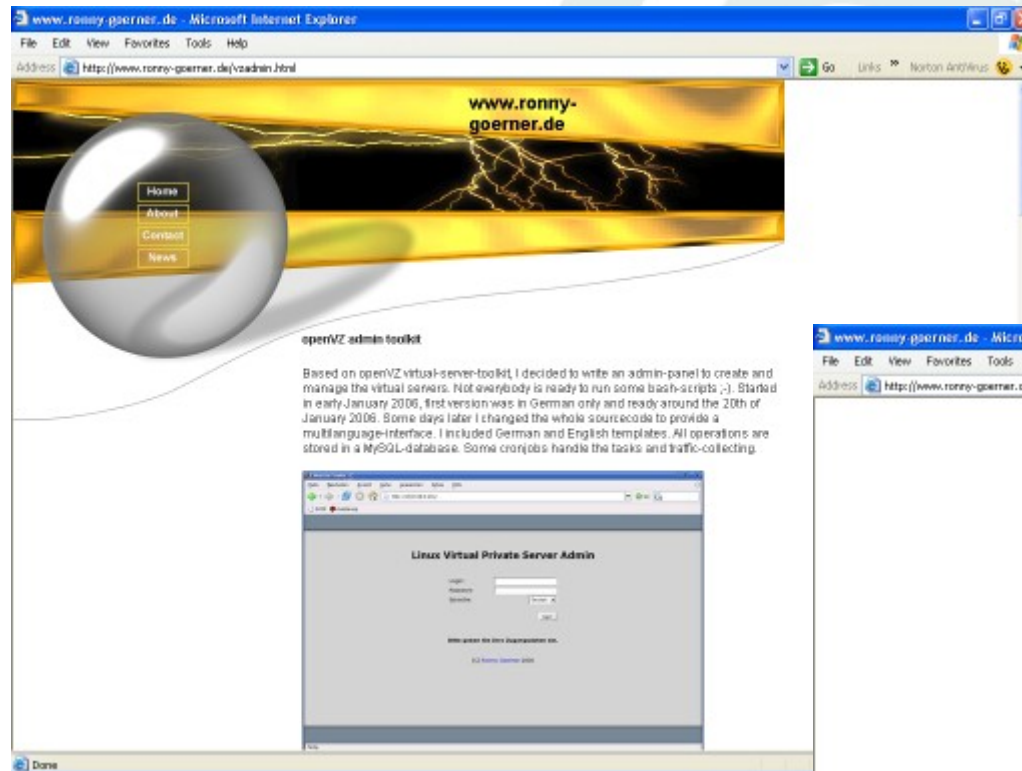
# Your role

- Use OpenVZ
- Contribute, be a part of the community:
  - Programmer
    - fixes
    - enhancements
    - new functionality
  - Non-programmer
    - bug reports
    - documentation, HOWTOs
    - answer support questions



# One example

## Web Control Panel for OpenVZ



# Project Links

- Main site: <http://openvz.org/>
- Downloads: <http://download.openvz.org/>
- Sources: <http://git.openvz.org/>
- Forum: <http://forum.openvz.org/>
- Bug Tracking: <http://bugzilla.openvz.org/>
- Blog: <http://blog.openvz.org/>
- Mailing lists:
  - [users@openvz.org](mailto:users@openvz.org)
  - [devel@openvz.org](mailto:devel@openvz.org)
  - [announce@openvz.org](mailto:announce@openvz.org)