

# An overview of OpenVZ virtualization technology

Kir Kolyshkin <[kir@openvz.org](mailto:kir@openvz.org)>  
*OpenVZ project manager*

# 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/>

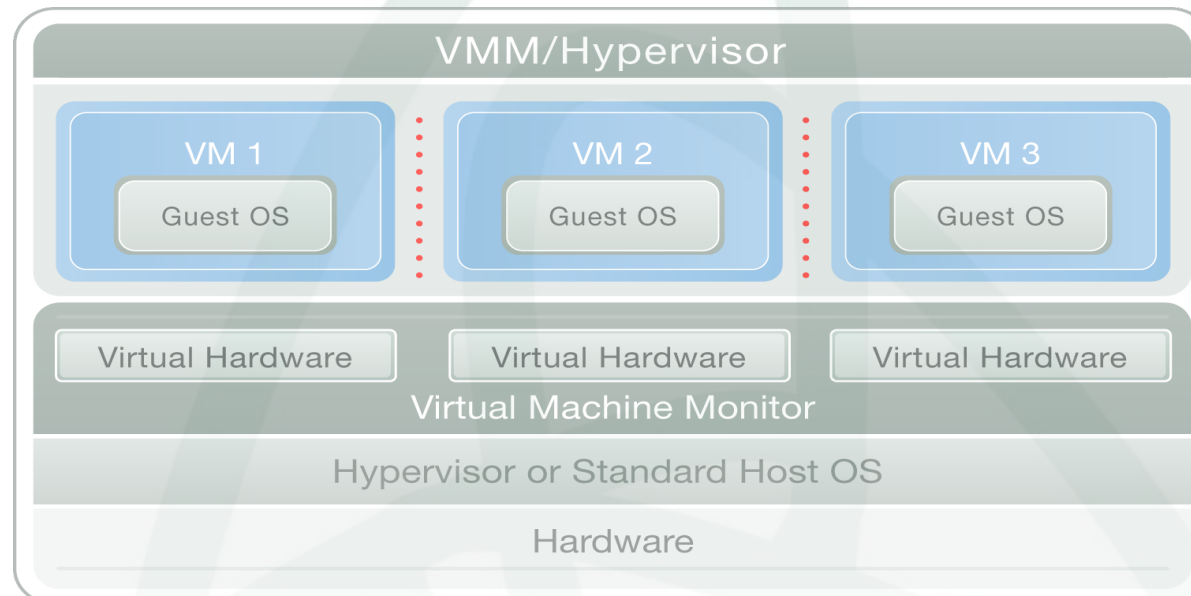
A key benefit of the virtualization is the ability to run multiple operating systems on a single physical server and share the underlying hardware resources – known as **partitioning**.

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

# Ways to Virtualize

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

# Emulation/Paravirtualization



- VMware
- Parallels
- QEmu
- Bochs



• Xen

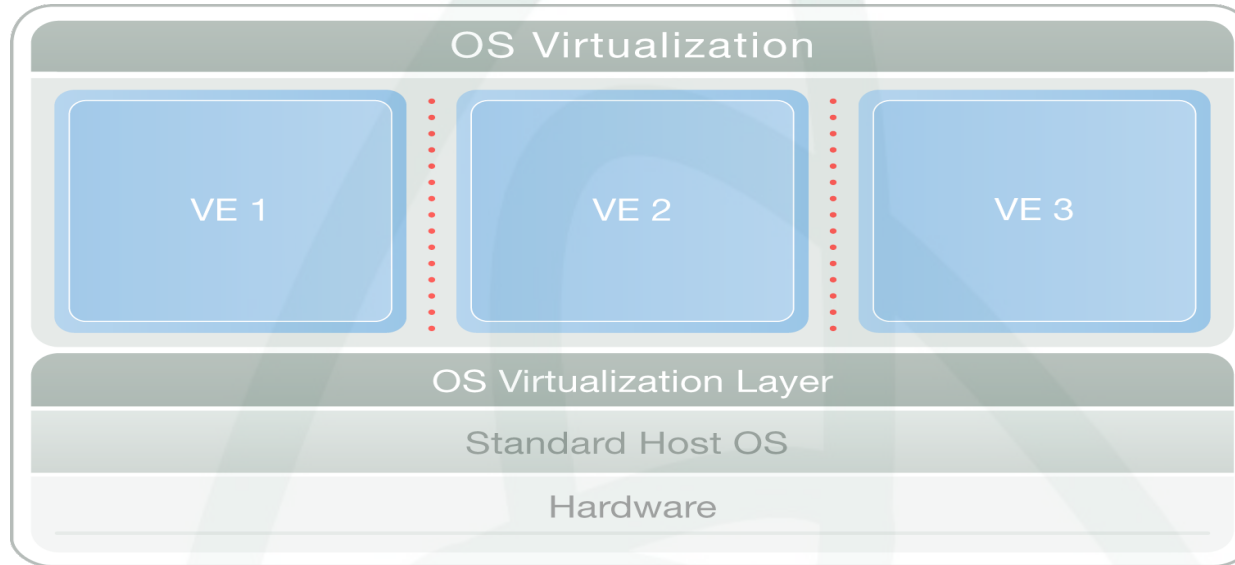


• UML  
(User Mode Linux)



• KVM

# OS Level Virtualization



- **OpenVZ/Virtuozzo**



- **FreeBSD jails**



- **Linux-VServer**

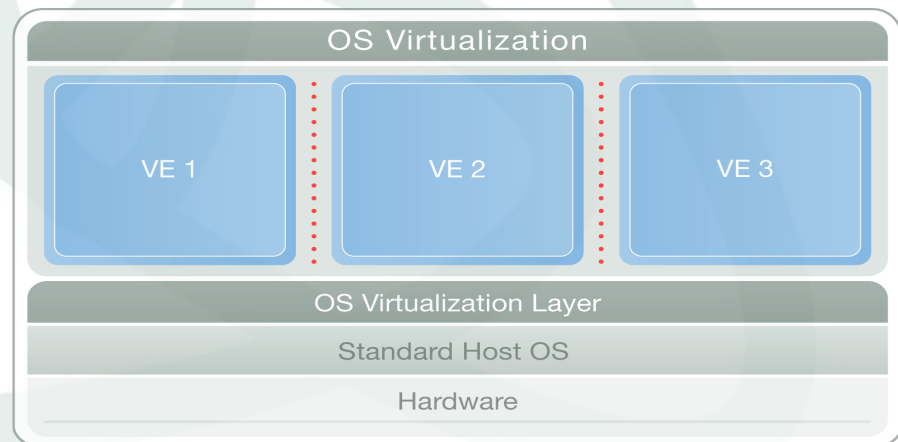
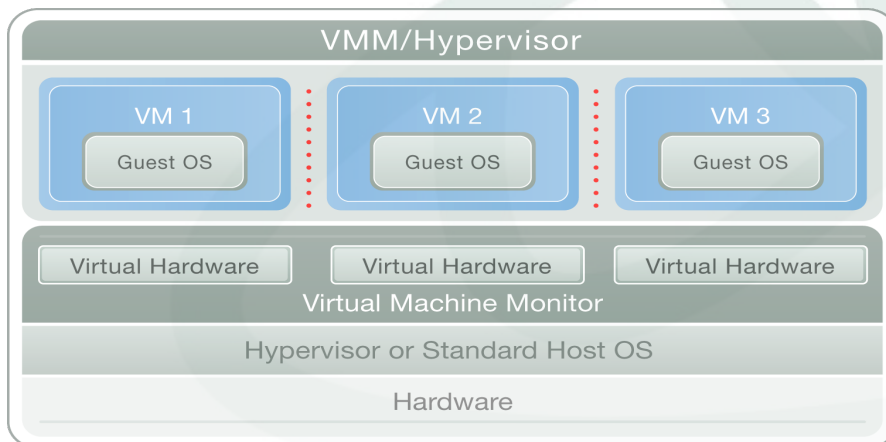


- **Solaris Zones**



# Comparison

- Can run different OSs on the same box
- Low density/scalability
- Slow/complex management – OS sprawl problem
- Low/moderate performance
- Native performance: no overhead
- Dynamic resource allocation, best scalability
- Single OS per box: easier to manage



# Evolution of Operating Systems

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

# 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.

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

Resource management is what makes OpenVZ different from other OS virtualization solutions.

# Kernel: Checkpointing/Migration

- Complete VE state can be saved in a file
  - running processes
  - opened files
  - network connections, buffers, backlogs, etc.
  - memory segments
- VE state can be restored later
- VE can be restored on a different server

# 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
```

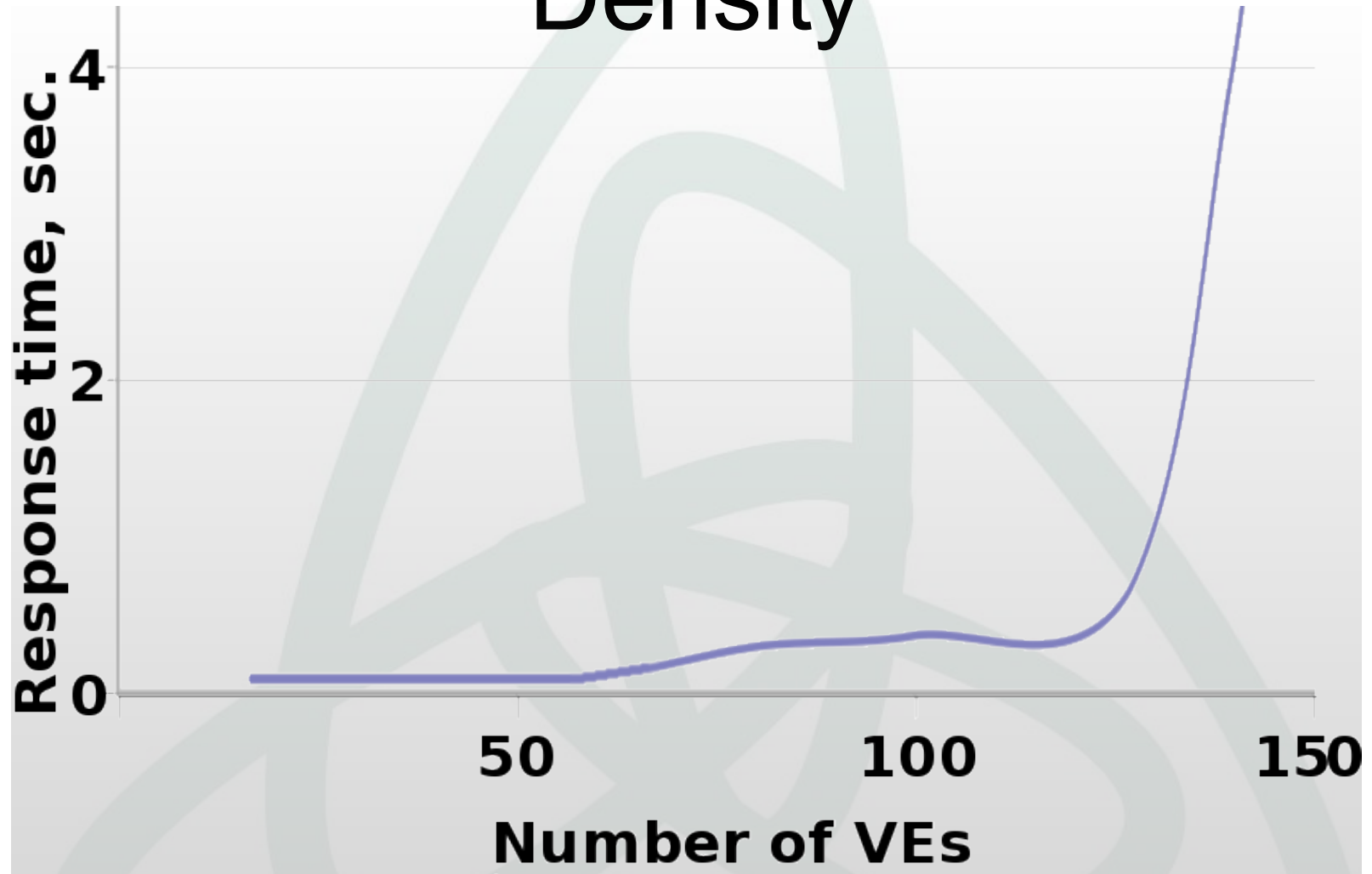
```
# vzpkgcache
```

```
(creates templates from metadata/updates existing  
templates)
```

```
# vzyum 101 install gcc
```

```
(installs gcc and its deps to VE 101)
```

# Density




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


# Users Feedback

Hello all, just downloaded and installed OpenVZ, and i must say its a big improvement over other VPS systems that i have tested IMHO.

[http://forum.openvz.org/index.php?t=msg&goto=646#msg\\_646](http://forum.openvz.org/index.php?t=msg&goto=646#msg_646)

I use virtuozzo in my day job and openvz is very much the same. Just no windows GUI which I hate using anyway! Virtuozzo and openvz are wonderful - I don't know why more people aren't using them. I hear a lot of hype for  and but virtuozzo/openvz is so great for many common needs. I'm very happy to be using openvz - very good for my side projects that I can't afford real virtuozzo for.

[http://forum.openvz.org/index.php?t=msg&goto=650#msg\\_650](http://forum.openvz.org/index.php?t=msg&goto=650#msg_650)

Last week when we were in limbo about what to do, it was decided to try out Virtualization. From what is written in the press the  system has alot of promise, <...> but was far too complicated to get working in our configuration. OpenVZ was the only virtual server system that was simple to install and get working.

[http://forum.openvz.org/index.php?t=msg&goto=568#msg\\_568](http://forum.openvz.org/index.php?t=msg&goto=568#msg_568)

# 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
- 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

# Recent achievements

- NFS and FUSE in VE
- VE I/O accounting and scheduling
- Checkpointing/live migration for IA64
- Port to RHEL5 kernel
- Port to vanilla 2.6.20

# Mainstream kernel integration

- Collaborative community effort:
  - OpenVZ
  - IBM (Metacluster)
  - Linux-VServer
  - Eric Biederman (namespaces)
  - Google (Paul Menage, containers)
- Current progress (as of linux-2.6.20):
  - IPC namespaces/virtualization
  - utsname() virtualization
  - preliminary support for PID namespaces
- More to come soon (networking, beancounters)

# How can you help?

- Use OpenVZ
- Contribute to OpenVZ, be a part of community:
  - Programmer
    - fixes
    - enhancements
    - new functionality
  - Non-programmer
    - bug reports
    - work with wiki
    - answer support questions



# What about Itanium?

- OpenVZ is platform-independent
  - as long as Linux support it, we support it
- The only arch-dependent piece is CPT
  - recently added checkpointing for IA64
- We support and care for Itanium for years
  - production quality, first released in Mar 2003
- No problems with scalability or disk IO
  - lots of memory, lots of CPUs no prob
  - native I/O speed

# Project Links

- Main site: <http://openvz.org/>
- Downloads: <http://download.openvz.org/>
- Wiki: <http://wiki.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)