

# FreeBSD bhyve projects in University POLITEHNICA of Bucharest

Mihai Carabas, Elena Mihailescu, Darius Mihai, Alexandru Elisei  
mihai@freebsd.org, elenamihailescu22@gmail.com,  
dariusmihaim@gmail.com, alexandru.elisei@gmail.com



bhyvecon Tokyo 2018 - The BSD Hypervisor Conference  
Tokyo University of Science  
Tokyo, Japan  
March 9th, 2017



# About me

- ▶ Assistant professor at University POLITEHNICA of Bucharest
  - ▶ Teaching Assistant: operating systems, systems architecture, networks



# About me

- ▶ Assistant professor at University POLITEHNICA of Bucharest
  - ▶ Teaching Assistant: operating systems, systems architecture, networks
- ▶ BSD world
  - ▶ DragonFly BSD: SMT aware scheduler - 2012, Intel EPT for vkernels - 2013
  - ▶ FreeBSD - bhyve: instruction caching - 2014, porting bhyve on ARM - 2015-2016
- ▶ Coordinating bhyve related diploma and master projects



# bhyve through diploma and master projects

- ▶ Promote and Coordinate
- ▶ Lot of work have been done, not too much yet committed to upstream
  - ▶ instruction caching
  - ▶ emulate NE2000 network device driver
  - ▶ emulate ATA disk controller
  - ▶ emulate HD-Audio device driver
  - ▶ porting bhyve on ARMv7/ARMv8
  - ▶ enabling virtio devices on bhyve ARMv7
  - ▶ bhyve x86 save-restore mechanism



# bhyve through diploma and master projects

- ▶ Promote and Coordinate
- ▶ Lot of work have been done, not too much yet committed to upstream
  - ▶ instruction caching
  - ▶ emulate NE2000 network device driver
  - ▶ emulate ATA disk controller
  - ▶ emulate HD-Audio device driver
  - ▶ porting bhyve on ARMv7/ARMv8
  - ▶ enabling virtio devices on bhyve ARMv7
  - ▶ bhyve x86 save-restore mechanism
- ▶ For ease of tracking we created a Github group:  
<https://github.com/FreeBSD-UPB>



# Outline

Instruction Caching

Emulated Devices

bhyve on ARMv7

bhyve on ARMv8

Virtio devices for bhyve ARM

bhyve save-restore

Conclusions



FreeBSD

# Instruction Caching

- ▶ Author: Mihai Carabas
- ▶ Coordinator: Neel Natu
- ▶ GSoC 2014



# Instruction Caching

- ▶ Author: Mihai Carabas
- ▶ Coordinator: Neel Natu
- ▶ GSoC 2014
  
- ▶ Not yet committed due to its low impact
- ▶ When we will support nested virtualization



# Outline

Instruction Caching

**Emulated Devices**

bhyve on ARMv7

bhyve on ARMv8

Virtio devices for bhyve ARM

bhyve save-restore

Conclusions



FreeBSD

# Emulated Devices

- ▶ Author: Alex Teaca
- ▶ Coordinator: Peter Grehan, Mihai Carabas
- ▶ Internal development in UPB and GSoC 2015/2016



# Emulated Devices - current status

- ▶ Not yet committed
- ▶ Peter is waiting for capsicum to come in before doing a new device driver model

# Emulated Devices - current status

- ▶ Not yet committed
- ▶ Peter is waiting for capsicum to come in before doing a new device driver model
- ▶ HD-Audio is the next candidate because it has the least dependencies



## Emulated Devices - current status

- ▶ Not yet committed
- ▶ Peter is waiting for capsicum to come in before doing a new device driver model
- ▶ HD-Audio is the next candidate because it has the least dependencies
- ▶ NE2000 is waiting for the netmap backend code (blocked on Peter)

## Emulated Devices - current status

- ▶ Not yet committed
- ▶ Peter is waiting for capsicum to come in before doing a new device driver model
- ▶ HD-Audio is the next candidate because it has the least dependencies
- ▶ NE2000 is waiting for the netmap backend code (blocked on Peter)
- ▶ ATA disk controller emulation needs reworking (Peter said that found a candidate)



# Outline

Instruction Caching

Emulated Devices

**bhyve on ARMv7**

bhyve on ARMv8

Virtio devices for bhyve ARM

bhyve save-restore

Conclusions



# bhyve on ARMv7

- ▶ Author: Mihai Carabas, Nicolae-Alex Ivan, Darius Mihai
- ▶ Coordinator: Peter Grehan
- ▶ Internal development in UPB and GSoC 2015
- ▶ <https://github.com/FreeBSD-UPB/freebsd/tree/projects/bhyvearm>





## bhyve on ARMv7 - progress

- ▶ Virtualized interrupt controller
- ▶ Virtualized timer

## bhyve on ARMv7 - progress

- ▶ Virtualized interrupt controller
- ▶ Virtualized timer
- ▶ Currently you can boot a fully functional FreeBSD VM (interrupts and timer)
- ▶ Platforms:
  - ▶ Emulator from FastModels (Cortex A15)
  - ▶ Cubieboard2

## bhyve on ARMv7 - progress

- ▶ Virtualized interrupt controller
- ▶ Virtualized timer
- ▶ Currently you can boot a fully functional FreeBSD VM (interrupts and timer)
- ▶ Platforms:
  - ▶ Emulator from FastModels (Cortex A15)
  - ▶ Cubieboard2
- ▶ More tech details about interrupt controller virtualization on AsiaBSDCon2018 presentation (11th of March 2018)

## bhyve on ARMv7 - progress

- ▶ Virtualized interrupt controller
- ▶ Virtualized timer
- ▶ Currently you can boot a fully functional FreeBSD VM (interrupts and timer)
- ▶ Platforms:
  - ▶ Emulator from FastModels (Cortex A15)
  - ▶ Cubieboard2
- ▶ More tech details about interrupt controller virtualization on AsiaBSDCon2018 presentation (11th of March 2018)
- ▶ Timer virtualization was ment for BSDCan2018 but not accepted



# Outline

Instruction Caching

Emulated Devices

bhyve on ARMv7

**bhyve on ARMv8**

Virtio devices for bhyve ARM

bhyve save-restore

Conclusions



# bhyve on ARMv8

- ▶ Author: Alexandru Elisei
- ▶ Coordinator: Mihai Carabas
- ▶ Internal development in UPB in 2017
- ▶ <https://github.com/FreeBSD-UPB/freebsd/tree/projects/bhyvearm64>
- ▶ Sponsored-by: FreeBSD Foundation (in form of scholarship through our university)



# bhyve on ARMv8

- ▶ Author: Alexandru Elisei
- ▶ Coordinator: Mihai Carabas
- ▶ Internal development in UPB in 2017
- ▶ <https://github.com/FreeBSD-UPB/freebsd/tree/projects/bhyvearm64>
- ▶ Sponsored-by: FreeBSD Foundation (in form of scholarship through our university)
  - ▶ Thank you Deb, Ed and Sabine for making this possible!



FreeBSD

# bhyve on ARMv8

- ▶ Author: Alexandru Elisei
- ▶ Coordinator: Mihai Carabas
- ▶ Internal development in UPB in 2017
- ▶ <https://github.com/FreeBSD-UPB/freebsd/tree/projects/bhyvearm64>
- ▶ Sponsored-by: FreeBSD Foundation (in form of scholarship through our university)
  - ▶ Thank you Deb, Ed and Sabine for making this possible!
- ▶ Alex (4th year bachelor student in Computer Science) will present you the current status report for bhyve on ARMv8





# bhyve on ARMv8 - boot log

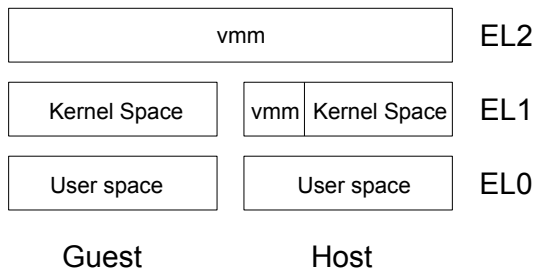
bhyvearm64

```
root@:~ # kldload vmm
root@:~ # bhyveload -k kernel.bin -m 128 -b 0x80000000 guest
root@:~ # bhyve -b guest
Copyright (c) 1992-2017 The FreeBSD Project.
Copyright (c) 1979, 1980, 1983, 1986, 1988, 1989, 1991,
1992, 1993, 1994
    The Regents of the University of California. All
rights reserved.
FreeBSD is a registered trademark of The FreeBSD
Foundation.
FreeBSD 12.0-CURRENT #65 b55ac7a322e(projects/bhyvearm64)-
dirty: Sun Nov 26
16:54:05 EET 2017
    alex@:/usr/home/alex/arm64-
workspace/obj/arm64.aarch64/usr/home/alex/arm64-
workspace/freebsd/sys/FOUNDATION_GUEST
arm64
FreeBSD clang version 4.0.0 (tags/RELEASE_400/final 297347)
(based on LLVM
4.0.0)
VT: init without driver.
arc4random: no preloaded entropy cache
random: entropy device external interface
kbd0 at kbdmux0
ofwbus0: <Open Firmware Device Tree>
simplebus0: <Flattened device tree simple bus> on ofwbus0
clk_fixed0: <Fixed clock> on simplebus0
clk_fixed1: <Fixed clock> on simplebus0
clk_fixed2: <Fixed clock> on simplebus0
gic0: <ARM Generic Interrupt Controller v3.0> mem
0x2f000000-0x2f00ffff,0x2f100000-0x2f2fffff,0x2c000000-
0x2c001fff,0x2c010000-0x2c011fff,0x2c02f000-0x2c030fff
irq 0 on ofwbus0
Unhandled memory access to 0x2f000000
Failed to emulate
instruction at
0xffff0000003102a0
```



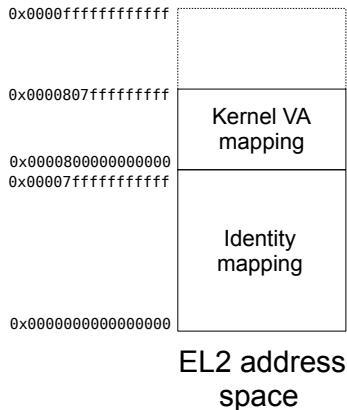
# bhyve on ARMv8 - privilege levels

bhyvearm64



# bhyve on ARMv8 - EL2 address space

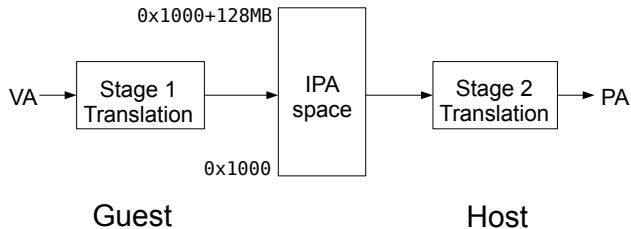
bhyvearm64



# bhyve on ARMv8 - multiple translations

bhyvearm64

```
bhyveload -k kernel.img -m 128 -b 0x1000 \
example_vm
```



# Outline

Instruction Caching

Emulated Devices

bhyve on ARMv7

bhyve on ARMv8

**Virtio devices for bhyve ARM**

bhyve save-restore

Conclusions



FreeBSD

# Virtio devices for bhyve ARM

- ▶ Author: Darius Mihai
- ▶ Coordinator: Mihai Carabas
- ▶ Internal development in UPB in 2017
- ▶ <https://github.com/FreeBSD-UPB/freebsd/commits/projects/bhyvearm>
- ▶ Sponsored-by: FreeBSD Foundation (in form of scholarship through our university)



# Virtio devices for bhyve ARM

- ▶ Author: Darius Mihai
- ▶ Coordinator: Mihai Carabas
- ▶ Internal development in UPB in 2017
- ▶ <https://github.com/FreeBSD-UPB/freebsd/commits/projects/bhyvearm>
- ▶ Sponsored-by: FreeBSD Foundation (in form of scholarship through our university)
  - ▶ Thank you Deb, Ed and Sabine for making this possible!



# Virtio devices for bhyve ARM

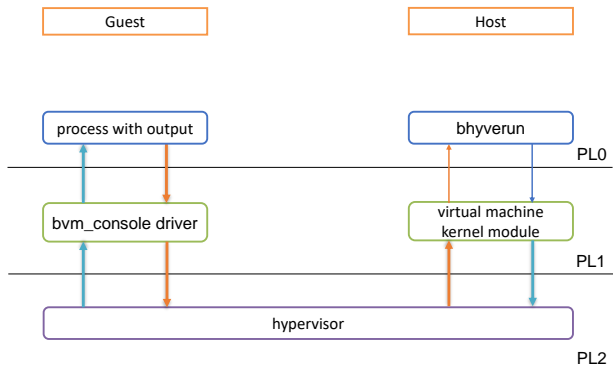
- ▶ Author: Darius Mihai
- ▶ Coordinator: Mihai Carabas
- ▶ Internal development in UPB in 2017
- ▶ <https://github.com/FreeBSD-UPB/freebsd/commits/projects/bhyvearm>
- ▶ Sponsored-by: FreeBSD Foundation (in form of scholarship through our university)
  - ▶ Thank you Deb, Ed and Sabine for making this possible!
- ▶ Darius (1st year master student in Network Security) will present the current status for virtio devices in bhyve ARM





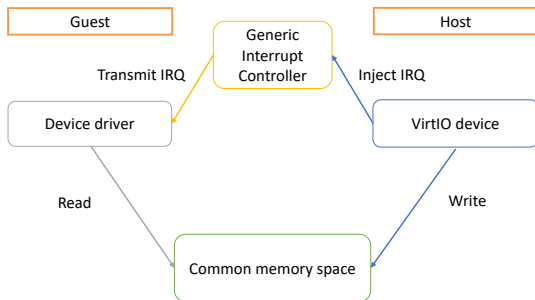
# Virtio on bhyve ARM - transfer flow

## Typical Data Transfer Workflow



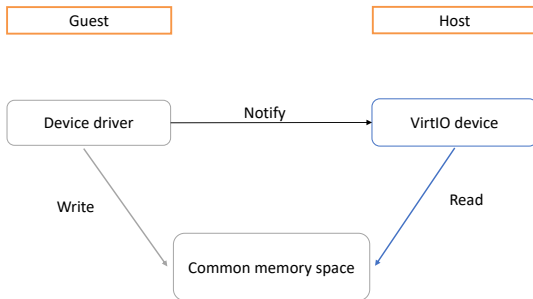
# Virtio on bhyve ARM - device communication

## VirtIO Device Communication



# Virtio on bhyve ARM - device communication (2)

## VirtIO Device Communication



# Outline

Instruction Caching

Emulated Devices

bhyve on ARMv7

bhyve on ARMv8

Virtio devices for bhyve ARM

**bhyve save-restore**

Conclusions



FreeBSD

## bhyve save-restore feature

- ▶ Author: Mihai Tiganus, Flavius Anton, Elena Mihailescu
- ▶ Coordinator: Mihai Carabas, Peter Grehan
- ▶ Internal development in UPB started from Summer 2016 and is on-going
- ▶ [https://github.com/FreeBSD-UPB/freebsd/tree/projects/bhyve\\_save\\_restore](https://github.com/FreeBSD-UPB/freebsd/tree/projects/bhyve_save_restore)
- ▶ Sponsored-by: Matthew Grooms (in form of scholarship for the Master students)

## bhyve save-restore feature

- ▶ Author: Mihai Tiganus, Flavius Anton, Elena Mihailescu
- ▶ Coordinator: Mihai Carabas, Peter Grehan
- ▶ Internal development in UPB started from Summer 2016 and is on-going
- ▶ [https://github.com/FreeBSD-UPB/freebsd/tree/projects/bhyve\\_save\\_restore](https://github.com/FreeBSD-UPB/freebsd/tree/projects/bhyve_save_restore)
- ▶ Sponsored-by: Matthew Grooms (in form of scholarship for the Master students)
  - ▶ Thank you Matthew for making this possible!



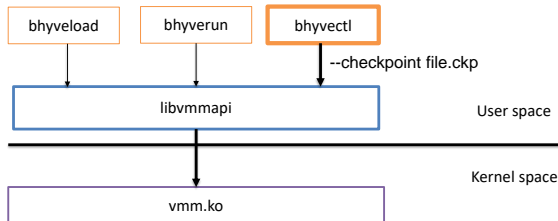
## bhyve save-restore feature

- ▶ Author: Mihai Tiganus, Flavius Anton, Elena Mihailescu
- ▶ Coordinator: Mihai Carabas, Peter Grehan
- ▶ Internal development in UPB started from Summer 2016 and is on-going
- ▶ [https://github.com/FreeBSD-UPB/freebsd/tree/projects/bhyve\\_save\\_restore](https://github.com/FreeBSD-UPB/freebsd/tree/projects/bhyve_save_restore)
- ▶ Sponsored-by: Matthew Grooms (in form of scholarship for the Master students)
  - ▶ Thank you Matthew for making this possible!
- ▶ Elena (1st year master student in Network Security) will present the current status of bhyve save-restore feature



# bhyve save-restore - save mechanism

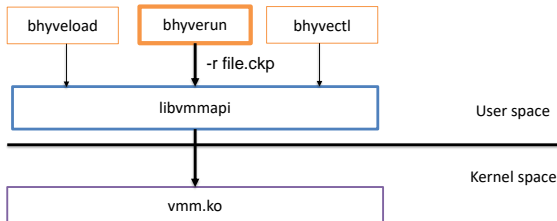
## Save Mechanism





# bhyve save-restore - restore mechanism

## Restore Mechanism



## Virtual Devices

- VATPIC - Virtual Programmable Interrupt Controller
- VATPIT - Virtual Programmable Interval Timer
- VRTC - Virtual Real Time Clock
- VMPTR - Virtual ACPI Power Management Timer



# bhyve save-restore - AMD support

## AMD CPUs

- AMD's Virtual Machine Representation
  - SVM
  - VMCB
- AMD's Specific Operations
  - VMLOAD
  - VMRUN
  - VMEXIT

## Current Status and Future Work

What we have implemented

- Guest's State Save and Restore

What we do now

- Warm Migration

What we aim to do next

- Live Migration



# Outline

Instruction Caching

Emulated Devices

bhyve on ARMv7

bhyve on ARMv8

Virtio devices for bhyve ARM

bhyve save-restore

Conclusions



FreeBSD

# Conclusions

- ▶ **Last year I was stating:**

- ▶ There is a great potential in developing core code for bhyve with students
- ▶ The satisfaction are from both perspectives (especially from them because they are doing low-level programming)
- ▶ Is hard to have results if you do not ensure a minimal scholarship from them



# Conclusions

- ▶ **Last year I was stating:**
  - ▶ There is a great potential in developing core code for bhyve with students
  - ▶ The satisfaction are from both perspectives (especially from them because they are doing low-level programming)
  - ▶ Is hard to have results if you do not ensure a minimal scholarship from them
- ▶ After my presentation I managed to talk to Deb and engage FreeBSD Foundation in two projects (virtio and ARMv8)



# Conclusions

- ▶ **Last year I was stating:**
  - ▶ There is a great potential in developing core code for bhyve with students
  - ▶ The satisfaction are from both perspectives (especially from them because they are doing low-level programming)
  - ▶ Is hard to have results if you do not ensure a minimal scholarship from them
- ▶ After my presentation I managed to talk to Deb and engage FreeBSD Foundation in two projects (virtio and ARMv8)
- ▶ Financed on-going projects: bhyve save-restore, bhyve on ARMv7/ARMv8





# Conclusions

## ▶ Last year I was stating:

- ▶ There is a great potential in developing core code for bhyve with students
  - ▶ The satisfaction are from both perspectives (especially from them because they are doing low-level programming)
  - ▶ Is hard to have results if you do not ensure a minimal scholarship from them
- ▶ After my presentation I managed to talk to Deb and engage FreeBSD Foundation in two projects (virtio and ARMv8)
- ▶ Financed on-going projects: bhyve save-restore, bhyve on ARMv7/ARMv8
- ▶ Personal perspective:
- ▶ We need to integrate ARM work in the main repo because is very hard to keep up with all the changes (we have a big codebase)

Thank you for your attention!

*ask questions*

