# Supporting RISC-V Full System Simulation in gem5

Prepared by Peter Yuen Ho Hin and Liao Xiongfei\*

www.huawei.com

June 2021

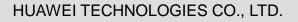
\*The full list of contributors can be found in paper



HUAWEI TECHNOLOGIES CO., LTD.

## Outline

- Introduction and Overview
- Target System
  - Hardware Configuration
  - Software Layers
  - Full System Linux Boot-up
- Challenges and Debugging Methodology
- Results
  - Diosix Boot-up
  - Benchmark Results





#### **RISC-V** and gem5

- RISC-V ISA
  - □ free, open-source ISA, popular in both the academia and the industry
  - □ simple, efficient yet future-proof, adopts a modular approach
- gem5 Simulator
  - open-source simulator widely used in computer architecture research
  - configurable CPU models, memory sub-systems and peripherals
  - Syscall Emulation (SE) mode
    - effects of system calls are emulated
    - suitable for quickly running benchmarks in user mode
  - Full System Simulation (FS) mode
    - cycle-accurate simulation of a full-fledged system: OS + kernel, peripherals, interrupts etc.
    - required for accurate and realistic analysis of system performance







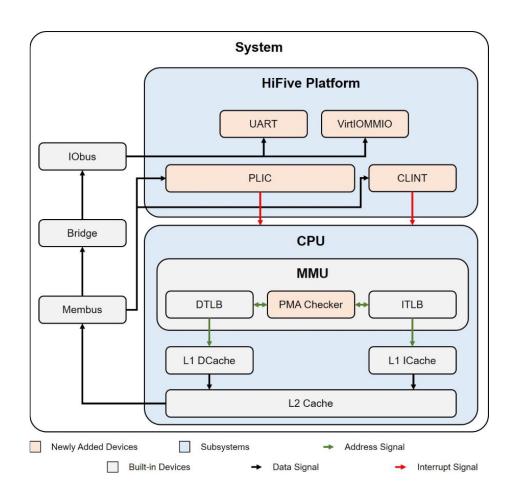
#### **RISC-V Full System Simulation in gem5**

- Need for gem5 RISC-V Full System Simulation
  - enables more research possibilities: virtual memory, virtualization, distributed system, storage stack performance etc.
  - several online feature requests through gem5-users mailing list and stackoverflow
  - several projects within Huawei can benefit from gem5 RISC-V Full System simulation capability
- Existing RISC-V ISA Support in gem5
  - "RISC5: Implementing the RISC-V ISA in gem5" @ CARRV 2017: supporting most RISC-V instructions and system calls in SE mode
  - "Simulating Multi-Core RISC-V Systems in gem5" @ CARRV 2018: supporting thread-related system calls and synchronization instructions
  - Our work: adding Full System simulation capabilities to gem5 RISC-V by developing peripheral models, fixing privileged instructions and interrupt handling as well as providing preliminary hypervisor support
- source code is available in official gem5-21.0 release (March 2021)

4

#### Hardware Configuration of Target System

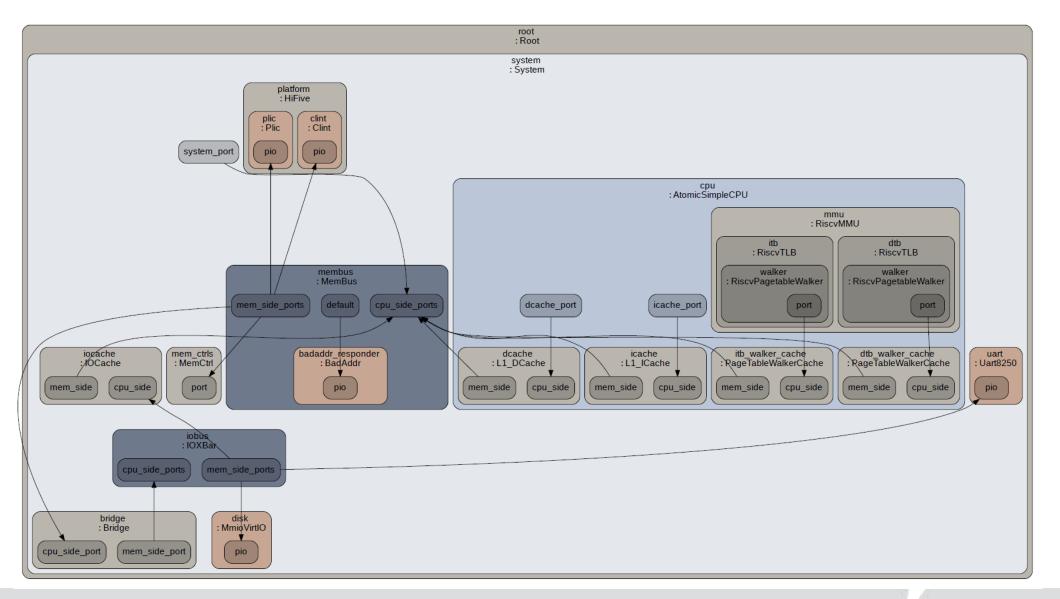
Target System: a baseline RISC-V system which can be easily extended based on user needs



- CPU Sub-system
  - PMA checker (extra MMU component)
    - checking physical memory attributes such as atomicity, memory-ordering, coherence, cacheability and idempotency
- HiFive Platform
  - based on SiFive's HiFive series of board
  - Core Local Interrupter (CLINT): handles software and timer interrupts via an MMIO interface
  - Platform Level Interrupt Controller (PLIC): routing external interrupts to the hardware threads based on a priority scheme
  - UART: provides an interactive command line terminal
  - VirtIOMMIO: provides a copy-on-write root filesystem which contains the workload scripts and operating system binaries



### Hardware Configuration in gem5

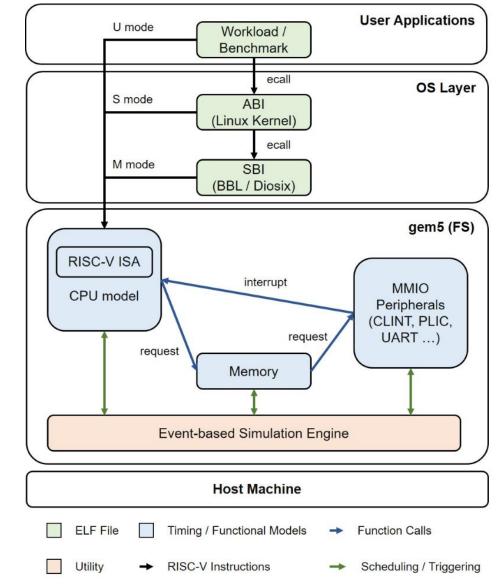


HUAWEI TECHNOLOGIES CO., LTD.



#### **Software Layers of Target System**

- gem5 (FS) Block
  - simulates functional and timing behaviour of hardware devices using an event-based approach
  - parses ELF workloads and sends machine instructions to CPU model
- CPU Model with RISC-V ISA Decoder
  - decodes the machine instructions and updates register state / performs memory requests

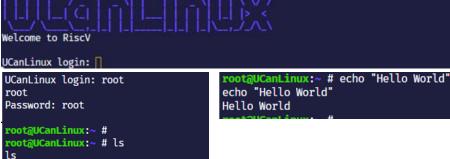




#### **Full System Linux Boot-Up**

- Setup
  - SMP with 4 CPU cores, each with one hardware thread
  - Berkeley bootloader (bbl) + Linux kernel v5.10
  - Filesystem: BusyBox disk image with PARSEC benchmark
- Features
  - Supports all four CPU models: Atomic Simple, Timing Simple, Minor and DerivO3
  - Multi-threaded workloads
  - Checkpoint and restoration functionalities for switching CPU models
  - m5 pseudo-instructions for dumping benchmark statistics / checkpoints
     via the terminal
  - Fully functional filesystem
  - Easy configuration through Python (e.g. automatic DTB generation)

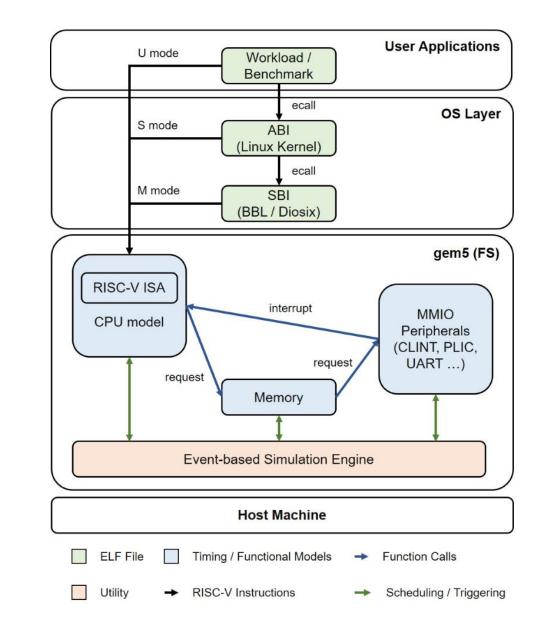
```
0.311007] virtio_blk virtio0: [vda] 524288 512-byte logical blocks (268 MB/256 MiB)
    0.311116] vda: detected capacity change from 0 to 268435456
    0.317180] libphy: Fixed MDIO Bus: probed
    0.319758] e1000e: Intel(R) PRO/1000 Network Driver
    0.319827] e1000e: Copyright(c) 1999 - 2015 Intel Corporation.
    0.320173] ehci_hcd: USB 2.0 'Enhanced' Host Controller (EHCI) Driver
    0.320256] ehci-pci: EHCI PCI platform driver
    0.320423] ehci-platform: EHCI generic platform driver
    0.320640] ohci hcd: USB 1.1 'Open' Host Controller (OHCI) Driver
    0.320768] ohci-pci: OHCI PCI platform driver
    0.320935] ohci-platform: OHCI generic platform driver
    0.321827] usbcore: registered new interface driver uas
    0.322055] usbcore: registered new interface driver usb-storage
    0.322445] mousedev: PS/2 mouse device common for all mice
    0.324000] usbcore: registered new interface driver usbhid
    0.324073] usbhid: USB HID core driver
    0.326590] NET: Registered protocol family 10
    0.328790] Segment Routing with IPv6
    0.329065] sit: IPv6, IPv4 and MPLS over IPv4 tunneling driver
    0.330865] NET: Registered protocol family 17
    0.331615] 9pnet: Installing 9P2000 support
    0.331828] Key type dns_resolver registered
    0.332146] debug vm pgtable: [debug vm pgtable
                                                          ]: Validating architecture page table hel
pers
    0.334822] EXT4-fs (vda): mounting ext2 file system using the ext4 subsystem
    0.338688] EXT4-fs (vda): mounted filesystem without journal. Opts: (null)
    0.338844] VFS: Mounted root (ext2 filesystem) readonly on device 254:0.
    0.340905] devtmpfs: mounted
    0.341456] Freeing unused kernel memory: 220K
    0.345704] Run /sbin/init as init process
   can't find device 'eth0'
ip: SIOCGIFFLAGS: No such device
ip: can't find device 'eth0'
```

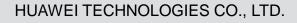




#### Challenges

- A fault can occur due to numerous reasons
  - DTB configuration error
  - wrong privileged ISA implementation
  - interrupt triggering mechanism or interrupt handling logic
     within CPU and interrupt controller
  - errors due to CPU pipeline and memory access of peripheral devices etc.
- Debugging involves 2 parts
  - kernel payload (in RISC-V assembly language, debugged using remote GDB)
  - gem5 implementation (in C++, debugged using host GDB)







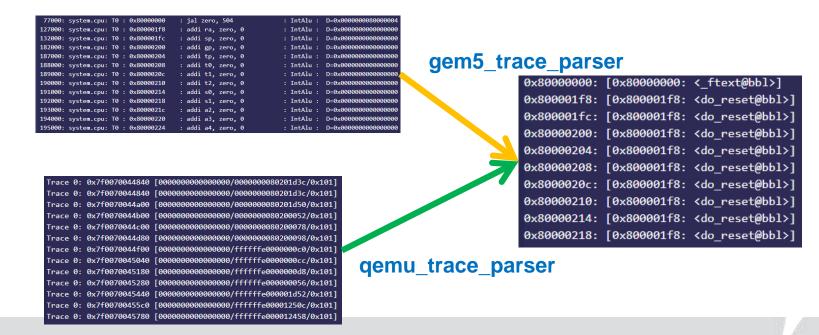
#### **Challenges (cont.)**

- Kernel payload: difficult to identify the instruction triggering the fault
  - large code size of kernel / bootloader
  - debugging mostly in assembly language
  - errors in kernel ends up in an infinite loop instead of exiting with an error
  - simple remote GDB is not suitable
- **gem5 implementation**: event-driven nature complicates debugging process
  - call-stack information is limited to calls within the same simulation tick
  - events such as memory read request and response would not be visible under the same call stack
  - a method of analysing beyond the scope of current tick is needed
  - relied heavily on debug logging (inefficient due to frequent rebuilding of gem5 binary)



#### **Debugging Methodology**

- Enhanced remote GDB support for RISC-V in gem5
  - added support for getting and setting values of FP and CSR registers for checking of privileged instruction implementation and interrupt register states
- Trace analysis and debugging using Python toolkit
  - boot QEMU and gem5 FS side-by-side using the same system setup and collect both execution traces
  - parse both traces and perform comparisons on the execution paths to find out where two traces diverge
  - allows for programmatic control of remote GDB instance (inserting breakpoints, comparing register states etc.)





#### **Debugging Process**

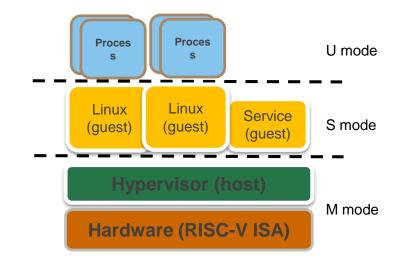
- Incorrect return value from a load instruction
  - incorrect implementation of a peripheral device
  - accidentally caching of the MMIO address range
- Interrupt trigger mechanism
  - requires both logging and remote GDB
  - remote GDB
    - examine the CPU's interrupt handling logic (e.g. Minor CPU repeatedly enters interrupt handler function)
  - logging
    - examine internal state of gem5 device models (e.g. PLIC's input and output registers)
    - investigate the sequence of events (verify timing behaviour of interrupt controllers)
- gem5's built-in debug logs
  - inspect detailed activities of built-in device models (CPU models, TLBs, MMU etc.)
  - identified an issue of CLINT's RTC accidentally triggering squashes in DerivO3CPU



#### **Diosix Boot-up**

- Diosix
  - an M mode hypervisor
  - can be used to perform hypervisor studies before H mode support is available
- System configuration of gem5 for bringing up DioSix
  - Using full system mode in gem5
    - 2 guest kernels (Linux kernel + UART service)
  - Using Linux Kernel 5.8
    - 5.10 has jump table bugs during kernel start-up (self-patching)
- Result
  - all guest are scheduling and running to idle
  - benchmarks can be run in the guest OS
  - interactive terminal using system service guest

HUAWEI TECHNOLOGIES CO., LTD.

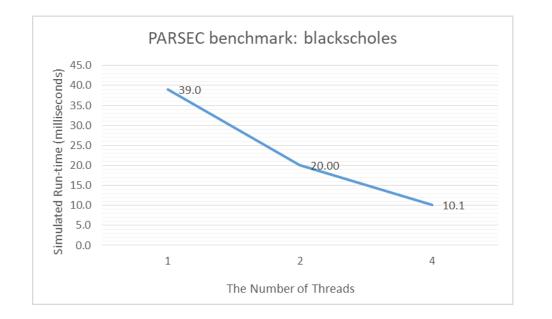


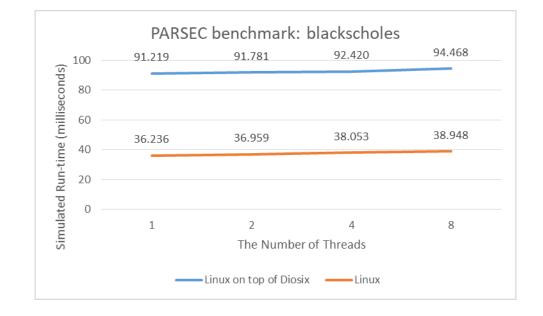
	`Cassadacasadacasadacasadacasadacasadacasadacasadacasadacasadacasadacasadacasadacasadacasadacasadacasadacas
	·/s5355535353535353
	yyyyyyyyyye+: : :333333333333333333333333333
	992002000002001001: 18853888888888888889899 90200200200020020015 + 184585888885888888989
	2010/02/02/02/02/02/0/ issssssssssssss
	4260460460460464y
	(36C036C36C36C36C36C3C) :ssssssssssss)^
	084008008008008/dy 4=83893838385850- 09409080908098/dy = 45838383838587-
	+* *+**********************************
	221:
	17111/,,//tubs/38538338385.5. 1101 0712016/0 - 0385383853853858585.0. 1101702/
	2010/12/2d/ `/05555555555555+. +110/210/
	1940424241n/5355555555555555
	989498202180'y, - + ssssssss, - +1/10/2820804 9892802280261 camiss: 1:202280280380
	GREGREGREGREGAELd: 1 ssso. + IEGREGREGREG
	1920-926-926-926-926-926-926-926-926-926-926
	decugedagedagedage(s) * //Nageodagedagedagedage)
na)	
ng)	
•	SS. SSSSSSSS. SSS ''' SSSS YSSSP
	888,o8P18887888, ,68P 88b dP888 oP*``*Yo,
	1603EF* 1660 **11602615* *116411* 5081,=* ****
	<pre>sttps://dicsix.org :: The Rust-RISC-V bare-metal hypervisor</pre>
	See REALME and LICENSE for usage, copyright, and distribution
	[2] CFU 0: 134217728 134217728 requested
	[?] CFU 0: region 0 size 488628224
	[2] CPU 0: capsule 1s started from 2550136832 134217728
	(?) CPU 0: Created system service goocy (console interface) 2170360 bytes (capsule 0) (?) CPU 0: Interface) CPU 2172 sequence 4
	[?] CPU 0: region 0 size 354410496
	(?) CPU 0: capsule is started from 2415919104 134217728
	(7) CDV 0: Created guest OS riscvid-linux-basylos-autoropythms (46-bit RISC-V linux with Busylos, Micropython, sth, less) 10280472 bytes (capsuls 1) (7) CDV 0: Physical CDC core DV64IBETC (Devel SiFiev/Thmuno); ready to roll
	(7) UN UT INTEGED LID CODE NOVEMBLED (DEBUSITE/ONEDODE) TEADY TO TOLI 0.000000] Linux yearin 5.10.13 (Artisédiativa) (isseré-baildacot-linux-gnu-gco.br real (Buildacot 2021.02-rc2-40-g42e60b515a) 10.2.0, GNU ld (GNU Binutile) 2.36.1) #4 SND Sun Feb 28 17:17:12 UTC 2021
	[ 0.000000] efi: UEFI not found.
	[ 0.000000] Zone ranges:
	[ 0.00000] Nxxxa] [see 0x3000006090000000-xx300000000-xx300000000]
	0.000000 Hovable cane start for each node
	[ 0.000000] Early memory node ranges
	[ 0.00000] node 0: [mem 0x0000000000000000000000000000000000
	[ 0.00000]Initmem setup node 0 [tmem 0x0000000000000000000000000000000000
	[ 0.000000] SBI specification v0.2 detected
	0.000000] SBI implementation ID=0x5 Version=0x1
	[ 0.000000] SBI v1.2 TIME extension detected 0.0000000] SBI v1.2 TEME extension detected
	0.000000) risovi ISA extensions avoifim
	[ 0.000000] riscv: ELF capabilities addfim
	[ 0.00000] percepu: Embedded 17 pages/cpu s3230 r1022 d2900 u59632 [ 0.000000 Built 2 constitus, mobility grouping on. Total pages: 3320
	c. OUVDUD DUILT 1 ZUMEISES, MODILICY GROUPING ON. IOCAL pages: 32320 G. OLDBODS Kernel Commond Line: Console=hvcD
	[ 0.000000] Dentry cache hash table entries: 16384 (order: 5, 131072 bytes, linear)
	[ 0.000000] Inode-cache hash table entries: 8192 (order: 4, 65536 bytes, linear)
	[ 0.000000] Sorting _extala [ 0.000000] men auto-init: stackioff, heap ilocroff, heap freesoff
	[ 0.000000] Memory: 39308K/131072K available (7079K kernel code, 4959K rwdata, 4096K rodata, 223K init, 343K bss, 91764K reserved, 0K cma-reserved)
	[ 0.000000] Virtual kernel memory layout:
	[ 0.00000] fixmap: cmffffcefe00000 - 0xffffceff000000 (2048 H8) [ 0.000000] pci io: cmffffceff000000-0xfffffceff0000000 (16 HB)
	or obtaining parties and interfection of the second s

#### **Benchmark Results**

- PARSEC blackscholes benchmark
  - FS simulation
    - simulated run-time scales well with the number of threads
    - 4 CPU cores, each with one hardware thread

- Linux vs Linux on top of Diosix hypervisor
  - check the overheads due to hypervisor
  - 1 CPU core with one hardware thread









#### Copyright©2011 Huawei Technologies Co., Ltd. All Rights Reserved.

The information in this document may contain predictive statements including, without limitation, statements regarding the future financial and operating results, future product portfolio, new technology, etc. There are a number of factors that could cause actual results and developments to differ materially from those expressed or implied in the predictive statements. Therefore, such information is provided for reference purpose only and constitutes neither an offer nor an acceptance. Huawei may change the information at any time without notice.