UNIVERSITY of York

BlueVisor: A Scalable Real-time Hardware Hypervisor for Many-core Embedded System

Zhe Jiang, Neil C Audsley, Pan Dong

Real-Time Systems Group
Department of Computer Science
University of York, United Kingdom

School of Computer
National University of Defense Technology
China

RTAS 2018

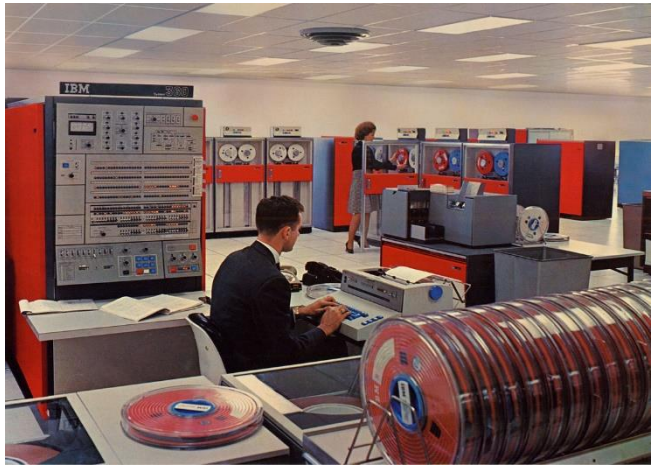
RTS York

Outline

- **Virtualization Technology**
- Networks-on-chip
- BlueVisor
- Experimental Evaluation
- Conclusion

Motivation

- Virtualization technology was invented by IBM in the 1960's on server platform
  - Resource sharing
- Currently, virtualization technology has became popular everywhere – server, desktop and embedded system
  - Resource utilization
  - System volume
  - Cost of hardware
  - Load Balance

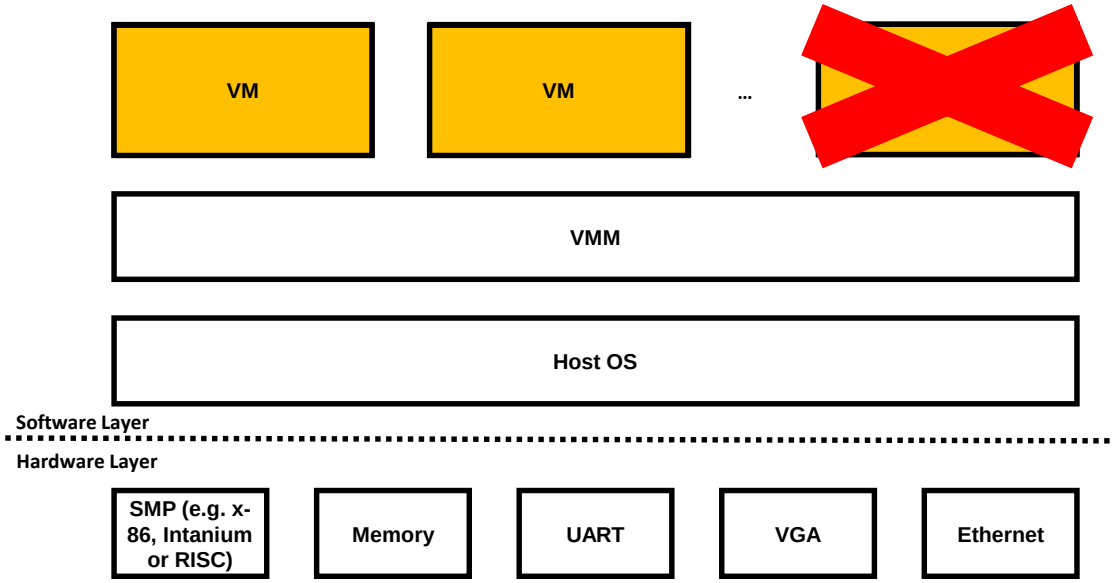


IBM CP-40 mainframe – the first system invented for server virtualization

In Real-time Systems

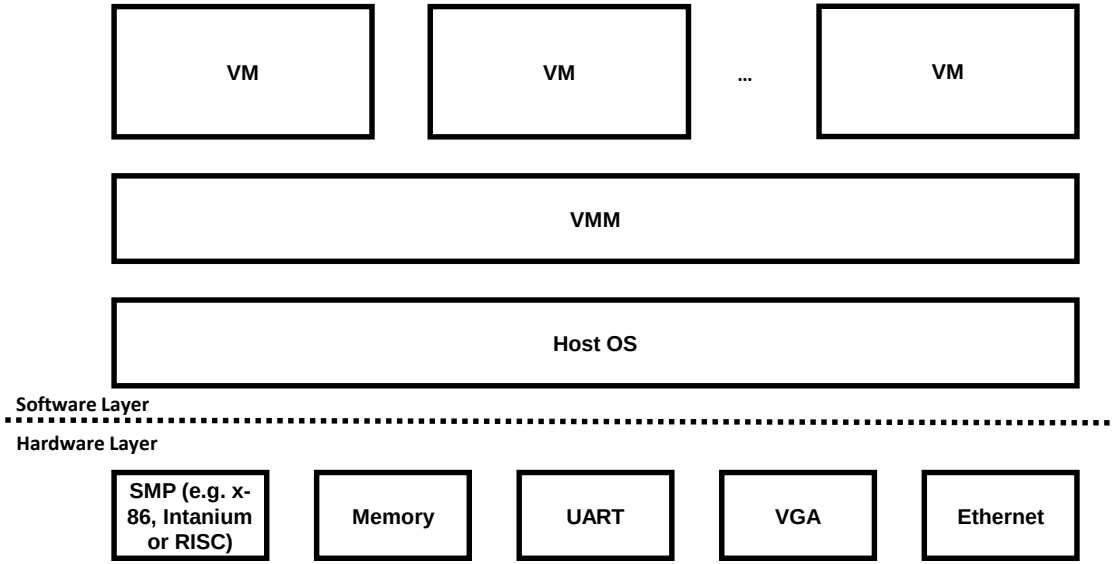
- Virtualization technology brings:
  - Consistent execution environment
  - Isolation

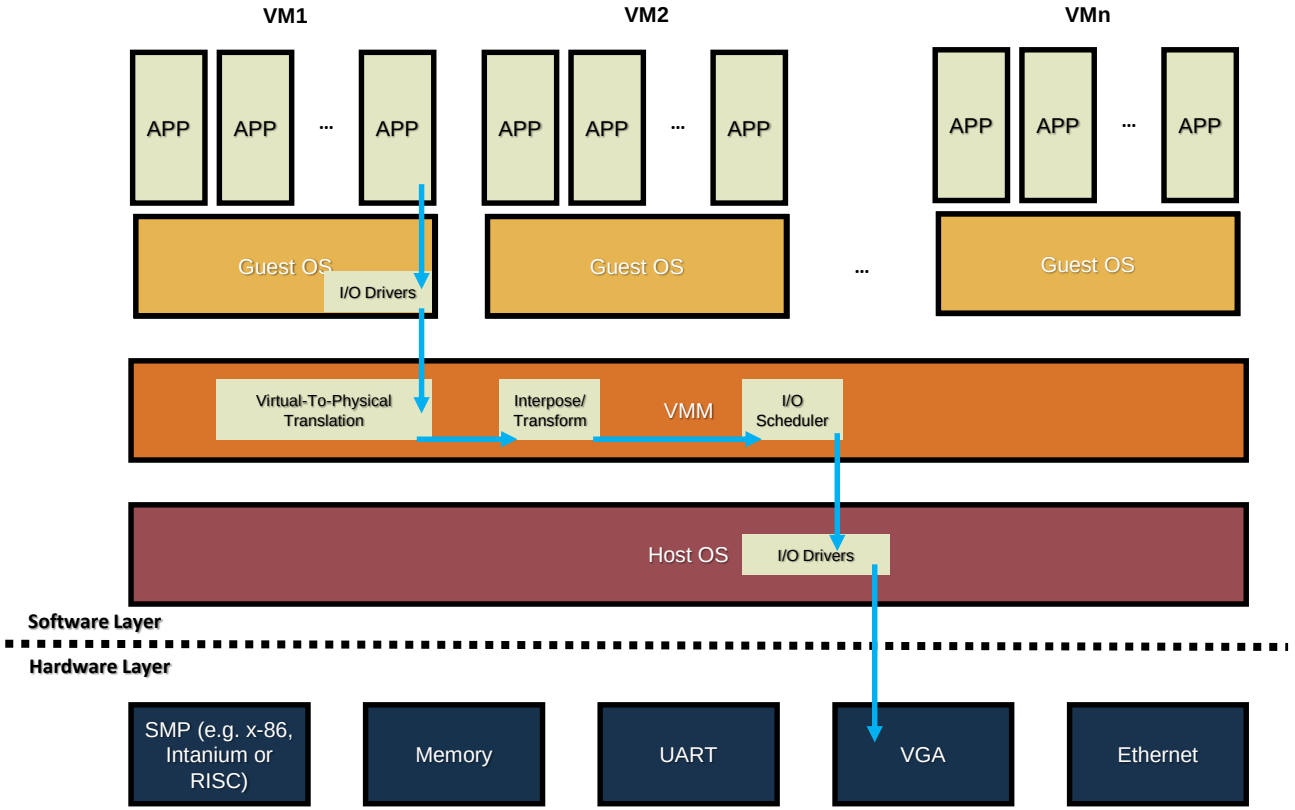
} Time Analysis

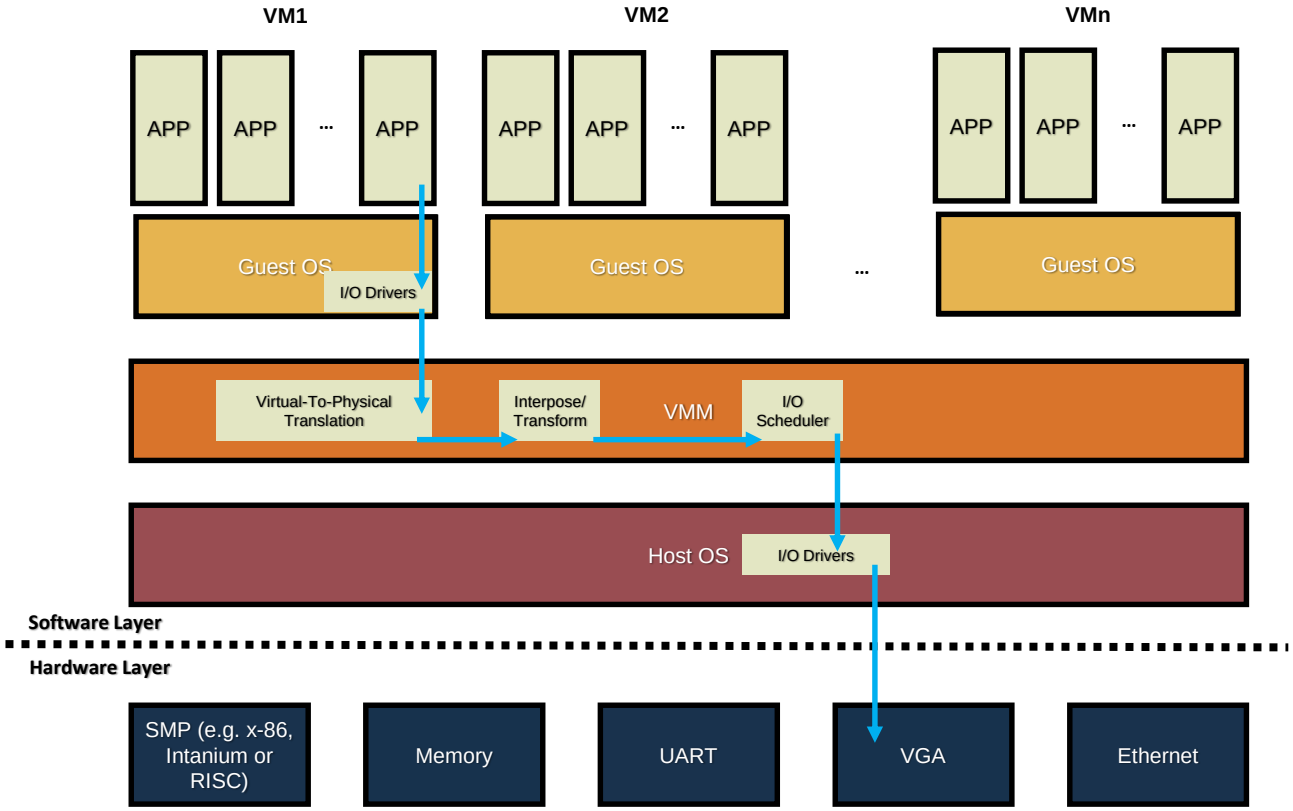


However

- Virtualization technology significantly conflicts to requirements of real-time systems, because of
  - Indirection and interposition of privileged instructions (e.g. I/O request)




VM1
VM2
VMn
APP
APP
...
APP
APP
APP
...
APP
APP
APP
...
APP
Guest OS
I/O Drivers
Guest OS
...
Guest OS
Virtual-To-Physical Translation
Interpose/ Transform
VMM
I/O Scheduler
Host OS
I/O Drivers
Software Layer
Hardware Layer
SMP (e.g. x-86, Intanium or RISC)
Memory
UART
VGA
Ethernet



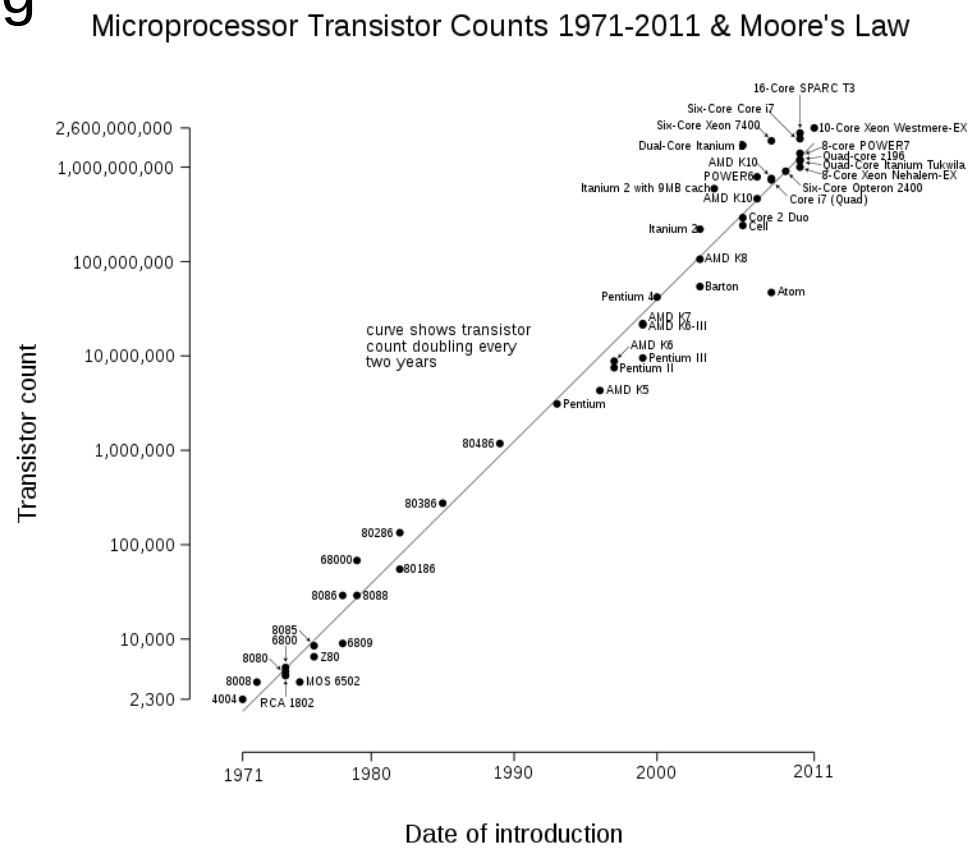
- Significant software overhead
- Longer response time of handling privileged instructions
- More uncertainty – Worse predictability

Outline

- Virtualization Technology
- **Networks-on-chip**
- BlueVisor
- Experimental Evaluation
- Conclusion

Move to Many-core Systems

- Breakdown of Dennard scaling
- In order to meet the year-on-year performance increment:
  - Before: Increase processor's clock speed / frequency
  - After: Increase the number of cores on a chip
- System:
  - Single core
  - Multi-core (point-to-point, bus)
  - Many-core (NoC)



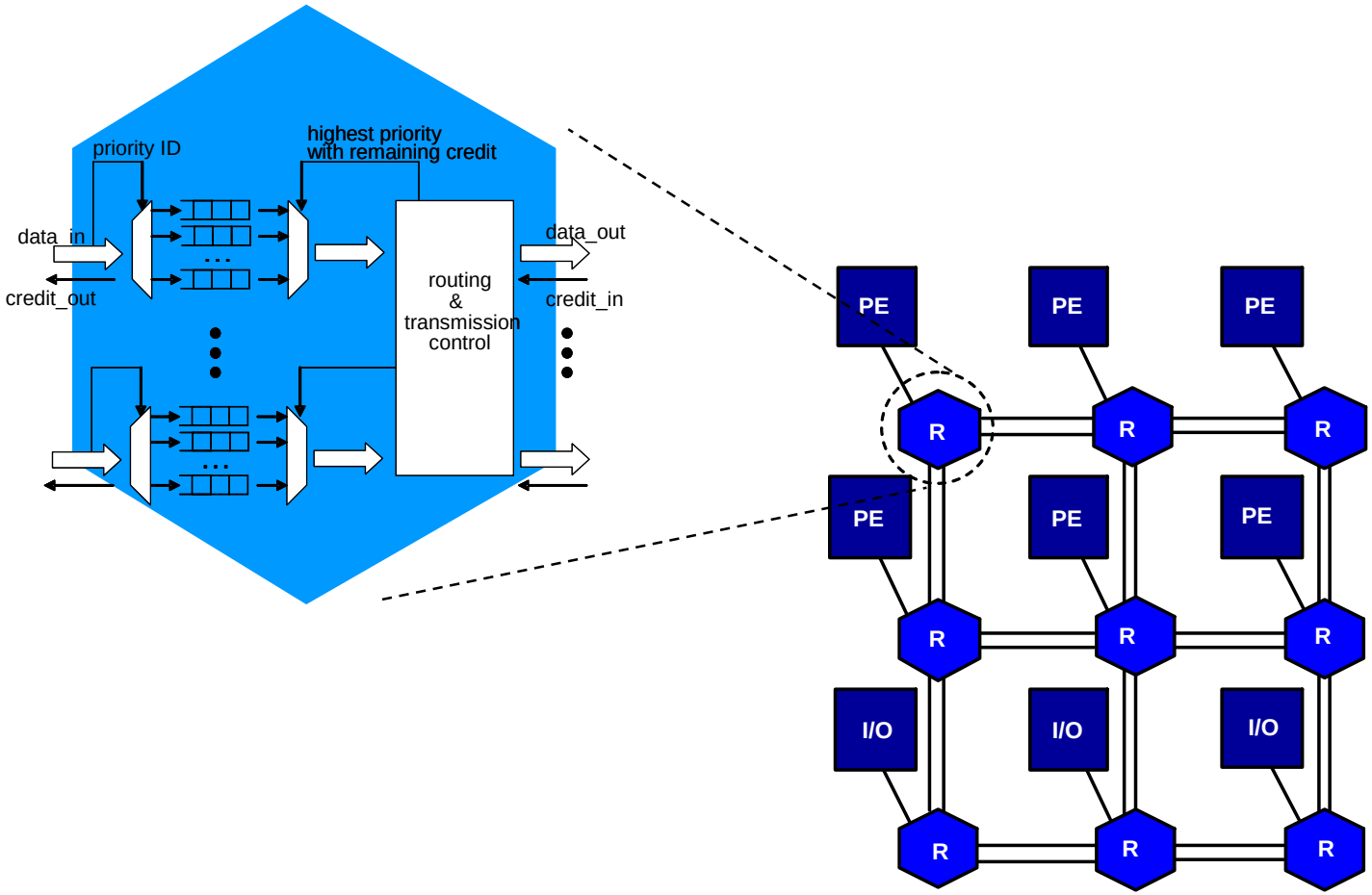
Network-on-Chip (NoC)

Parallella – 16 cores

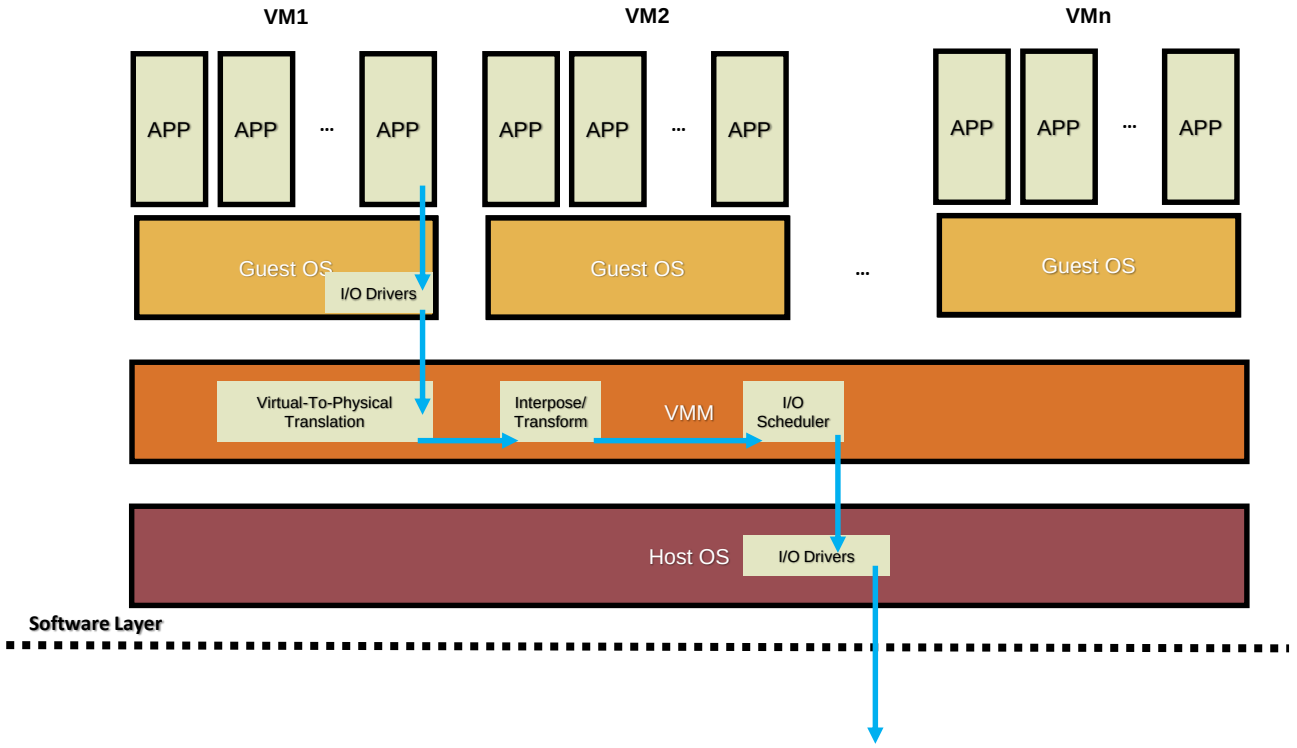
Intel's Knight Landing – 72 cores

MPPA – 256 cores

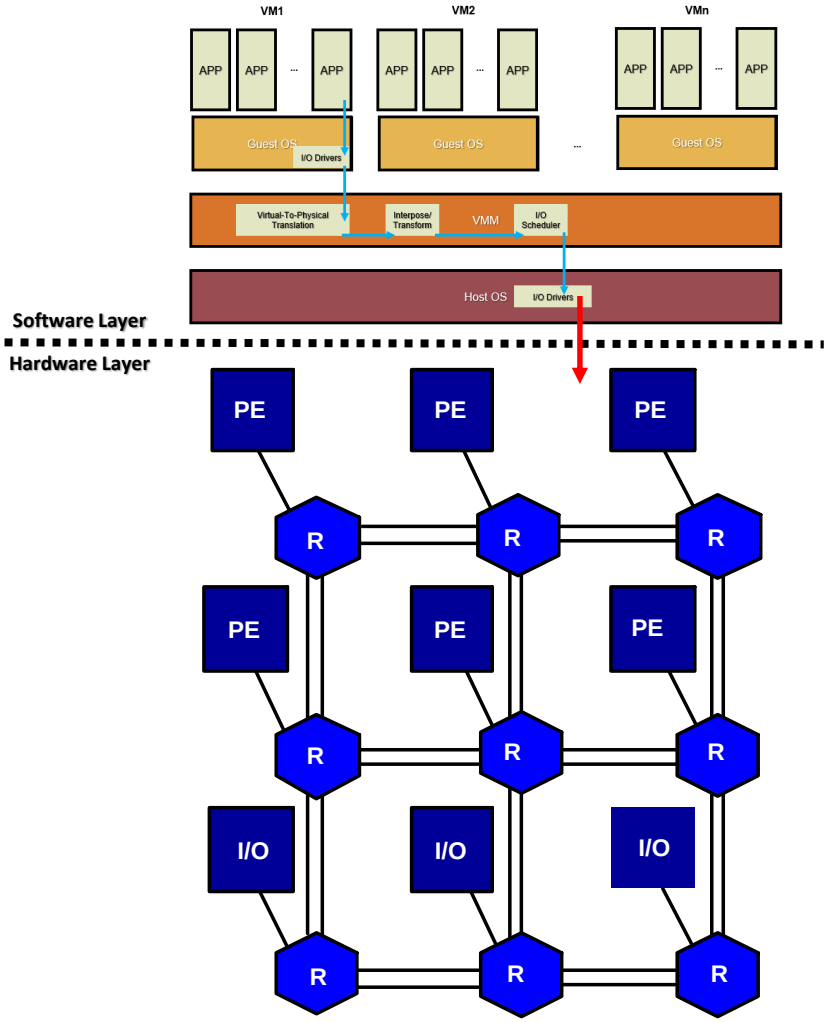
Network-on-Chip (NoC)



Vritualization

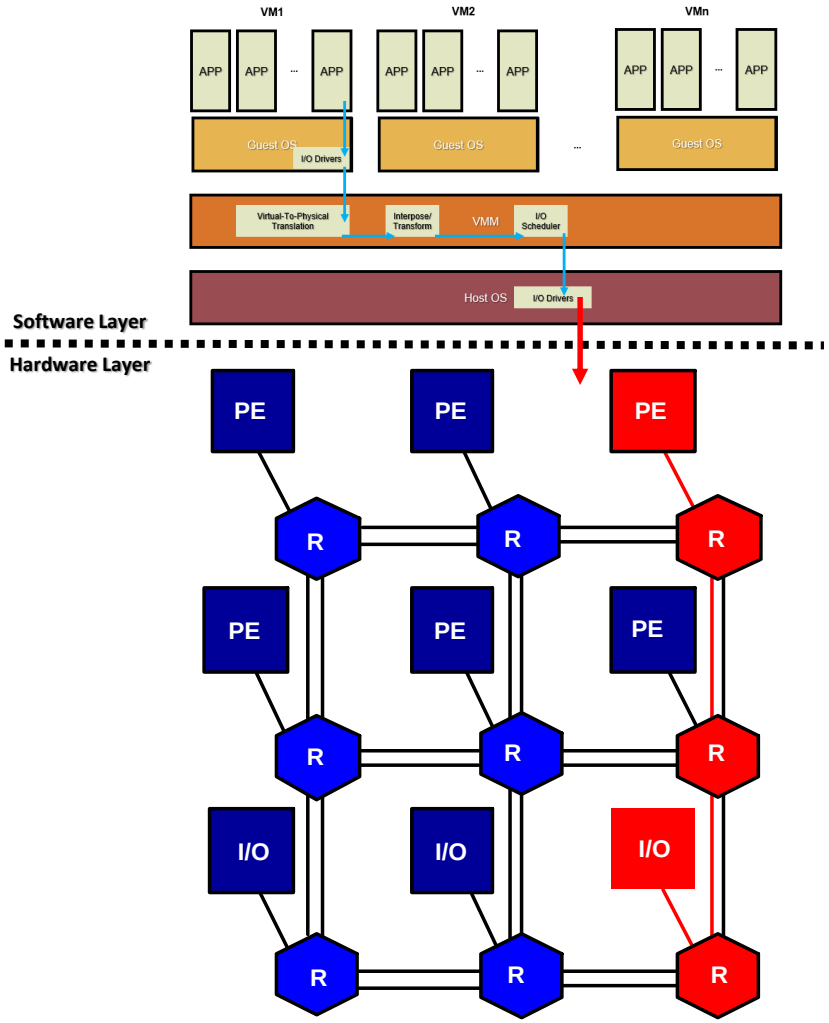


NoC + Virtualization



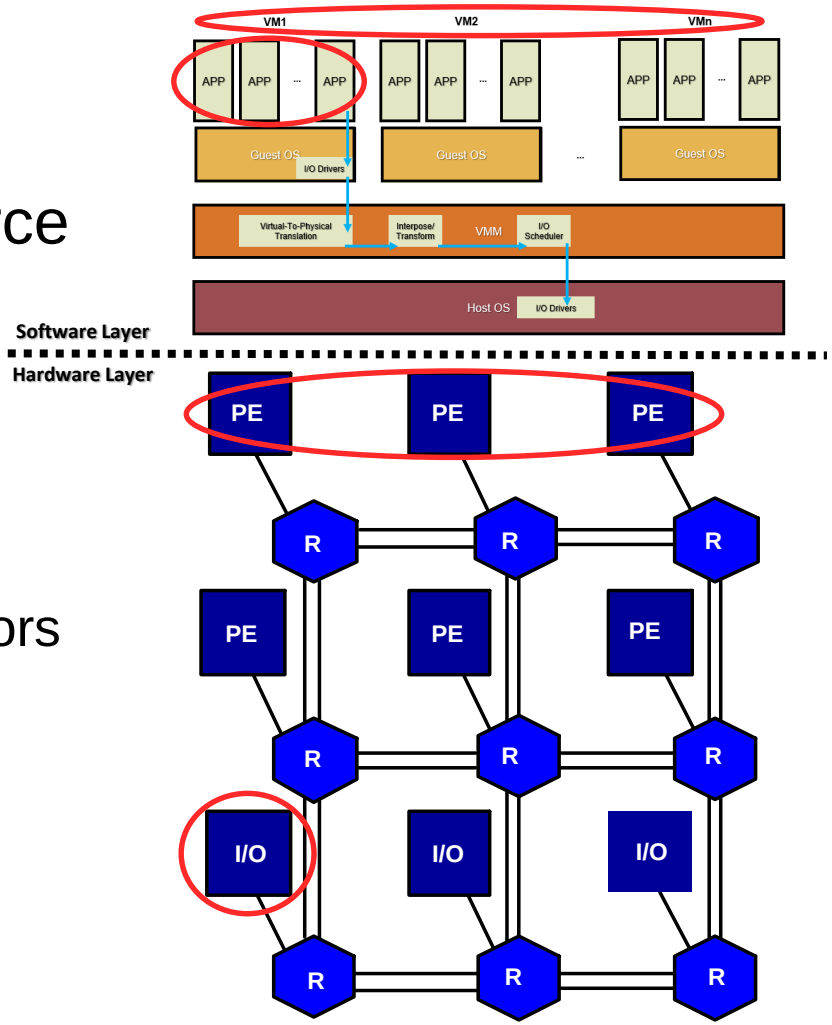
NoC + Virtualization

- With NoC, indirection and interposition of privileged instructions has became worsen
  - Longer response time – Decreased system performance
  - More uncertainty – Worsen predictability
  - Even some works are focusing on the predictability of NoC, e.g. Leandro *et al*. “Buffer-aware bounds …” DATE 2018.



NoC + Virtualization

- Moreover, the architecture also suffers from complicated resource management
  - Scheduling between the applications
  - Scheduling between the VMs
  - Scheduling between the processors
  - Contentions on I/Os
- Significant overhead, both software + hardware
- Decreased predictability



Therefore,

- We are looking for a hypervisor, which enables:
  - Predictability
  - Improved system performance
  - Decreased software overhead
  - Good Scalability
  - Isolation

A lot of intelligent are also working on this…

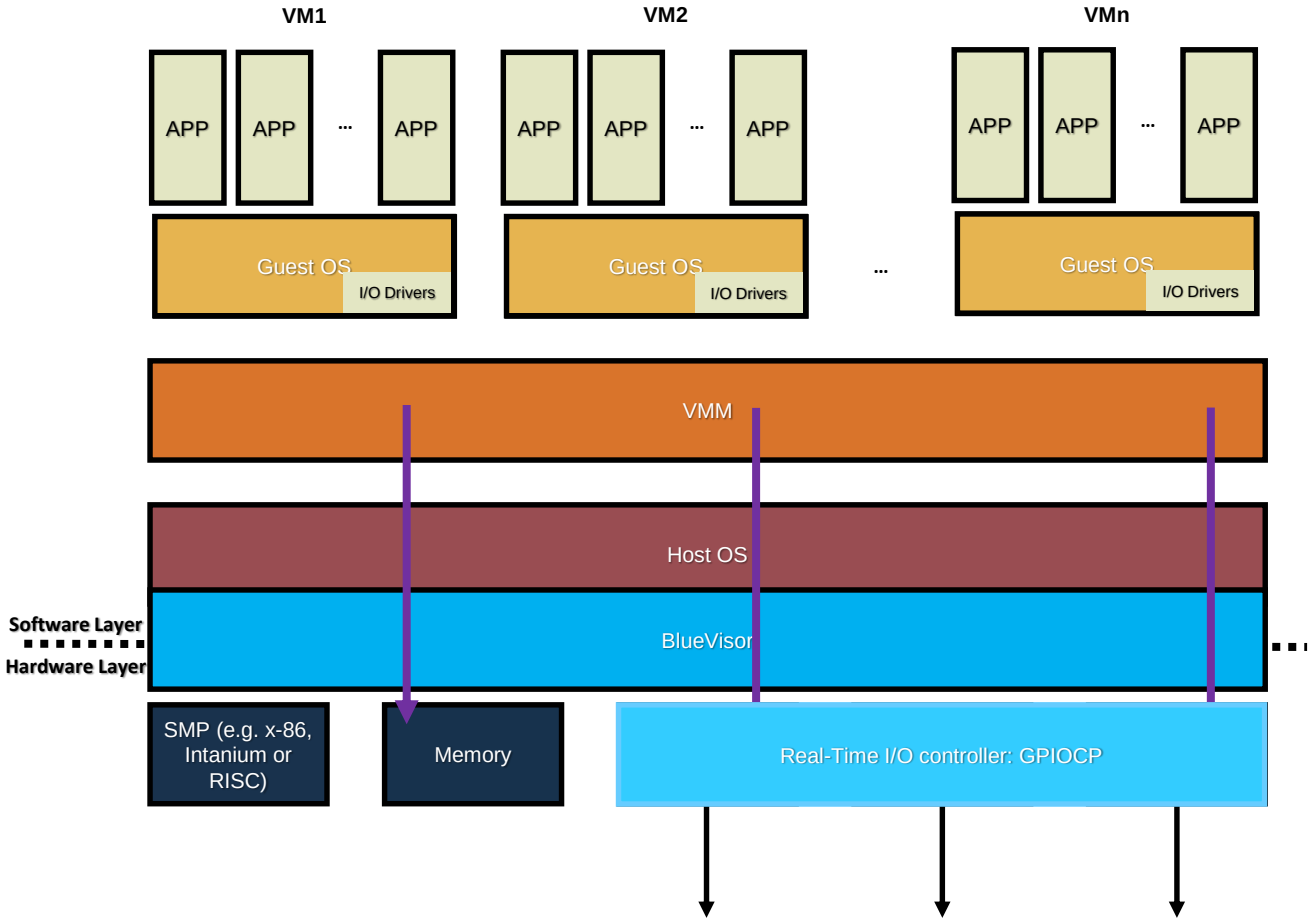
- Sisu Xi, Justin Wilson, Chengyang lu *et al*. “RT-Xen: towards real-time hypervisor scheduling in xen”

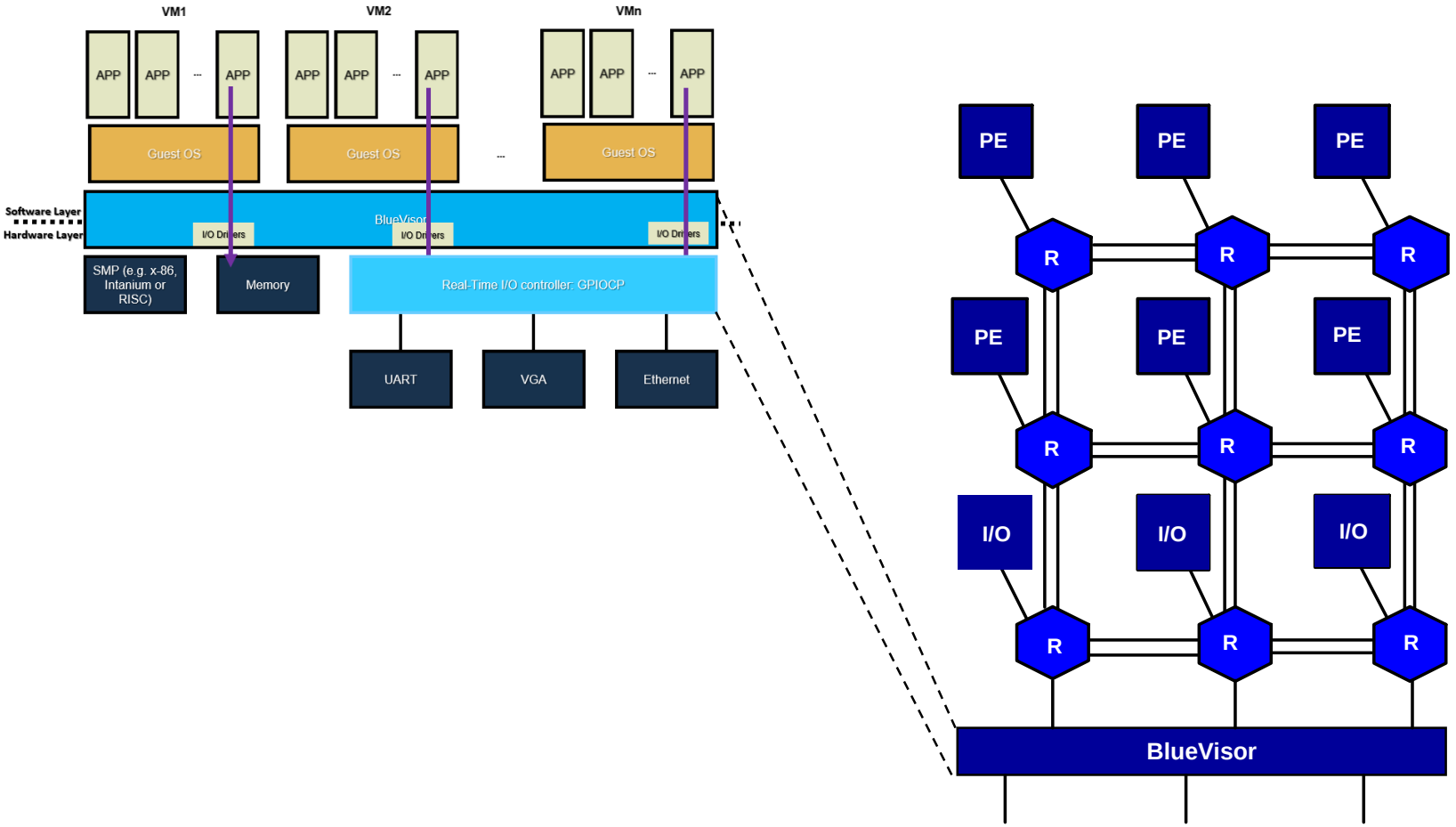
- Sandro Pinto, Jorge Pereira, Tiago Gomes *et al.* “ LTZVisor: TrustZone is the Key

- Ye Li, M Danish, R West *et al. “*Quest-V: A Virtualized Multikernel for High-Confidence Systems*”*

Outline

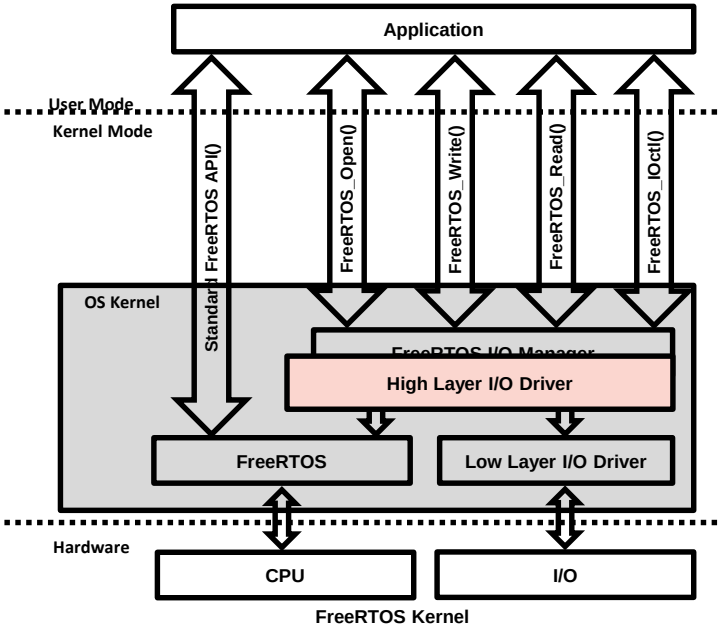
- Virtualization Technology
- Networks-on-chip
- **BlueVisor**
- Experimental Evaluation
- Conclusion


VM1
VM2
VMn
APP
APP
...
APP
APP
APP
...
APP
APP
APP
...
APP
Guest OS
I/O Drivers
Guest OS
I/O Drivers
...
Guest OS
I/O Drivers
VMM
Host OS
BlueVisor
Software Layer
Hardware Layer
SMP (e.g. x-86, Intanium or RISC)
Memory
Real-Time I/O controller: GPIOCP


VM1
VM2
VMn
APP
APP
APP
APP
APP
APP
APP
APP
APP
Guest OS
Guest OS
Guest OS
Software Layer
Hardware Layer
BlueVisor
I/O Drivers
I/O Drivers
I/O Drivers
SMP (e.g. x-86, Intanium or RISC)
Memory
Real-Time I/O controller: GPIOCP
UART
VGA
Ethernet
PE
PE
PE
PE
PE
PE
PE
PE
PE
R
R
R
R
R
R
R
R
R
I/O
I/O
I/O
BlueVisor

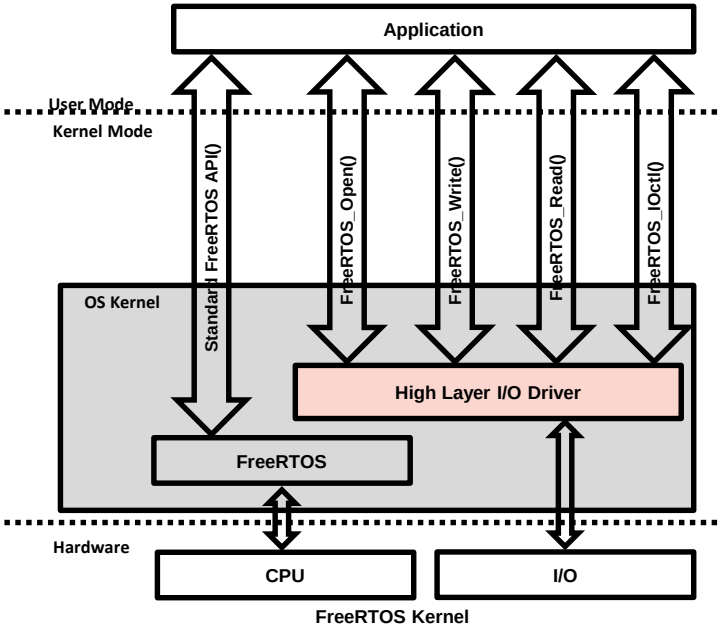
CPU Virtualization

- Each processor (no matter the architecture) is set as an individual guest VM.
  - Bare-metal Virtualization
  - Para-Virtualization



CPU Virtualization

- Each processor (no matter the architecture) is set as an individual guest VM.
  - Bare-metal Virtualization
  - Para-Virtualization

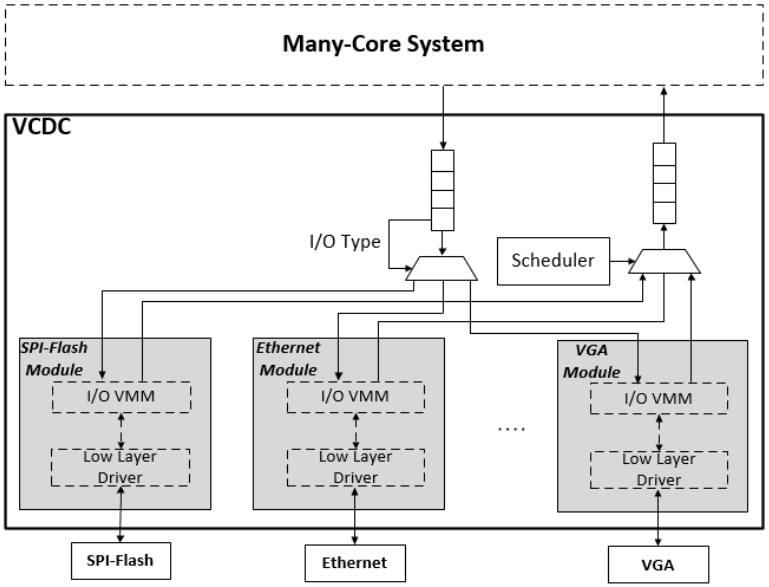


CPU Virtualization

- Currently, BlueVisor system supports the following OS kernels:
  - FreeRTOS
  - uCOSII
  - XilKernel
- Customized interface to support a new OS is also provided.

I/O Virtualization – VCDC

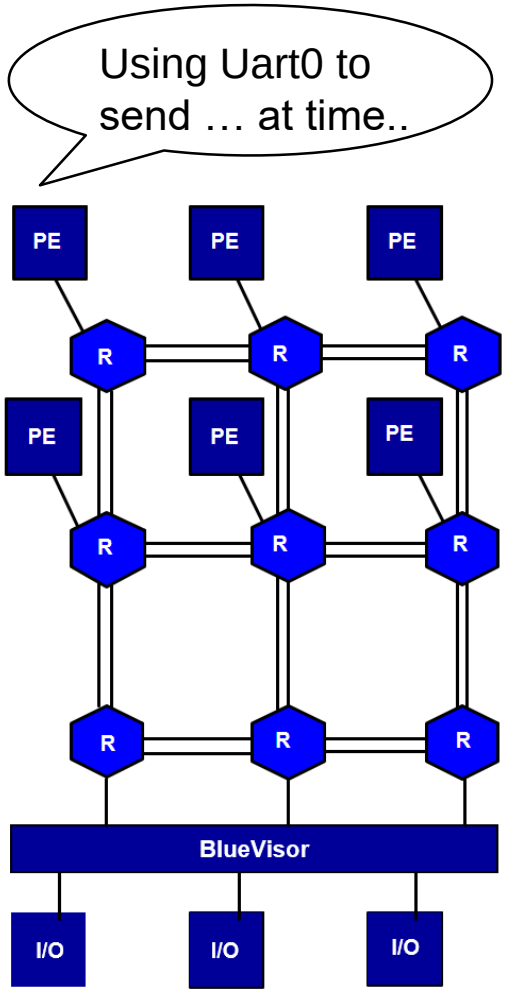
- Virtualized Complicated Device Controller (VCDC) integrates most functionalities of I/O virtualization and I/O drivers
  - Reduce software overhead significantly
  - Increase I/O performance significantly, i.e. I/O throughput & response time
  - Scalability



- Jiang, Zhe, and Neil C. Audsley. "VCDC: …" ECRTS 2017

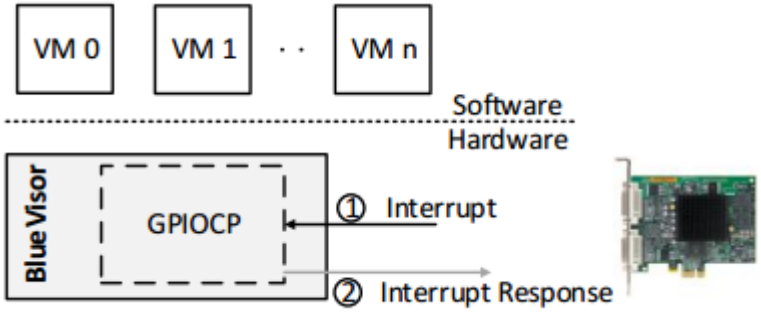
Real-time I/O Controller – GPIOCP

- General-purposed resource efficient co-processor
- Pre-programable, E.G. Using Uart0 to send “Hello World” at a specific time.
  - Remove the transmission
  - Enables predictable I/O operations
  - Enables Timng-Accurate I/O operations
- Jiang, Zhe, and Neil C. Audsley. “GPIOCP: …” DATE 2017

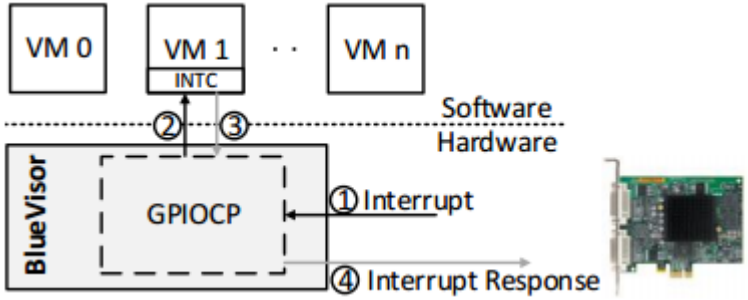


Interrupt Management

- Two types of interrupt management based on GPIOCP are provided:
  - Type1: Fast Interrupt Handler

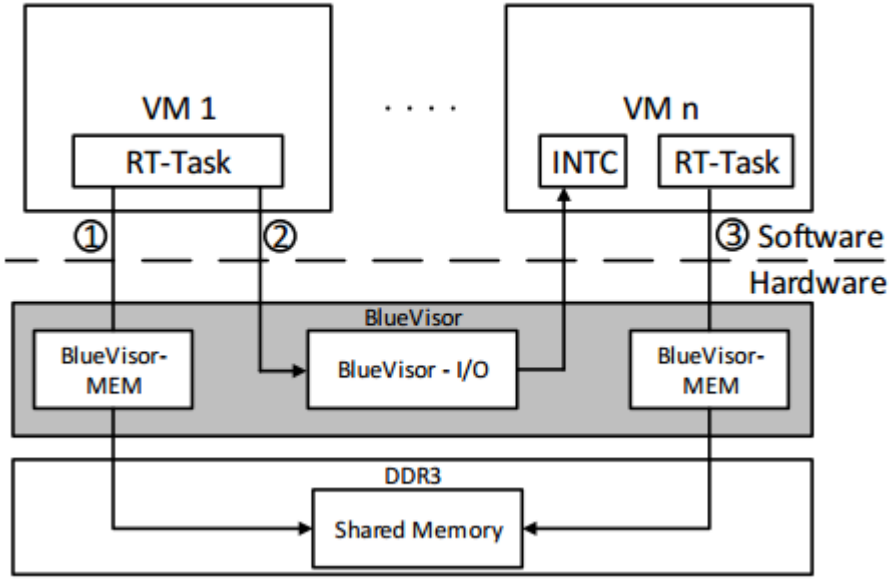


  - Type2: Normal Interrupt Handler



Inter-VM Communication

- Two types of Inter-VM communication
  - Packet-based communication (Simple and brief communication)
  - Memory-based communication (Complicated and long communication)



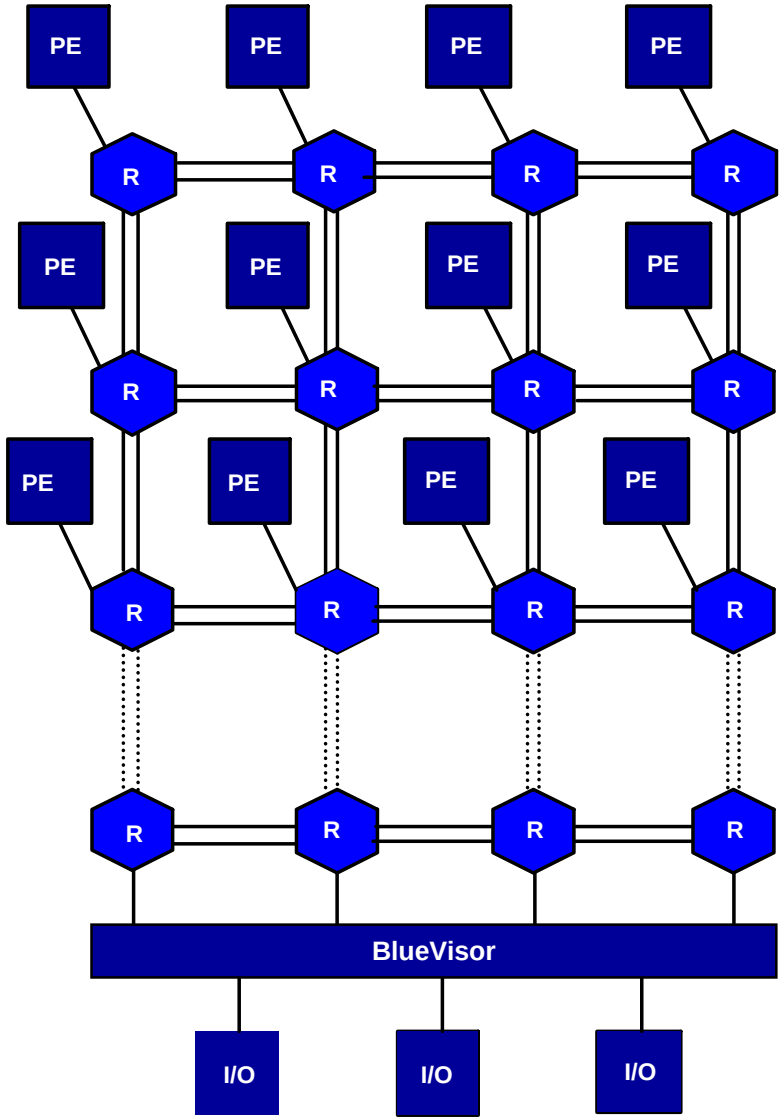
Outline

- Virtualization Technology
- Networks-on-chip
- BlueVisor
- **Experimental Evaluation**
- Conclusion

Experimental Platform

Platform	Many-Core System	Number of CPUs	System Frequency	Kernel Version
Xilinx VC709	4x5 size 2D mesh Type NoC	16	100 Mhz	FreeRTOS (V9.0.0)

- To enable comparison, a similar hardware architecture was built, but without the BlueVisor and virtualization technology.



Experimental Metrics

- Memory Footprint
- I/O performance
  - I/O Response Time
  - I/O Throughput
  - Timing-accuracy
- Interrupt Handling

Memory Footprint

- ***nFreeRTOS***: native FreeRTOS;
- ***FreeRTOS + I/O:*** FreeRTOS + I/O drivers (e.g. UART, VGA, etc.);
- **vFreeRTOS:** simply implemented software virtualized FreeRTOS;
- ***BV_vFreeRTOS:*** virtualized FreeRTOS in BlueViosr system.

	.text	.data	.bss	Total
BlueVisor	0	0	0	0
nFreeRTOS	121,309	1,728	35,704	158,741
nFreeRTOS+ I/O	179,652	1,852	36,250	217,754
vFreeRTOS + I/O	189,556	1,882	36,450	227,888
BV_vFreeRTOS +I/O	131,969	1,732	35,723	169,424

I/O Performance

- I/O Response Time
- I/O Throughput
- Timing-Accuracy

I/O Performance – Response Time

- Aims to evaluate the performance of the I/O system while CPU and I/O are fully loaded in a BlueVisor and non- BlueVisor system.

Number of Active CPUs	Evaluated I/O	I/O Requests	Scheduling Policy in non-BlueVisor	Scheduling Policy in BlueVisor
16	SPI nor-flash	Read 4, 8, 16, 32 … Bytes	Round-Robin (Global) FIFO (Local)	Round-Robin

I/O Performance – Response Time

Worst Case I/O Response Time (unit: clock cycle)

Written Bytes	Non-BlueVisor (FIFO)	Non-BlueVisor (Round-Robin)	BlueVisor (Round-Robin)
1	9,357	65,885	403
4	58,844	327,813	1,569
8	936,166	4,555,159	23,032
16	3,702,565	17,345,151	89,708

Variation of I/O Response Time (unit: clock cycle)

Written Bytes	Non-BlueVisor (FIFO)	Non-BlueVisor (Round-Robin)	BlueVisor (Round-Robin)
1	1,541	59,736	46
4	7,061	286,733	276
8	98,026	3,823,104	3,542
16	284,142	15,475,355	13,711

	Non-BlueIO System Scheduling Policy: FIFO			Non-BlueIO System Scheduling Policy: RoundRobin			BlueIO System		
CPU Index	Min	Max	Mean	Min	Max	Mean	Min	Max	Mean
Read 1 Byte									
(0, 0)	9357	9357	9357	6149	65885	36060	285	285	285
(0, 1)	7425	8989	8915	7073	65849	35860	380	403	396
(0, 2)	7057	8591	8[illegible]	7[illegible]	6[illegible]	3[illegible]	380	403	395
(1, 0)	7057	8207	8205	7096	65826	36237	357	403	391
(1, 1)	9748	9748	9748	7073	65826	36410	403	403	403
(1, 2)	7425	8966	8[illegible]	7[illegible]	6[illegible]	3[illegible]	334	334	334
(2, 0)	7034	8598	7467	7073	65826	36741	357	403	366
(2, 1)	7057	8207	8[illegible]	7096	65826	36930	357	403	377
(2, 2)	6121	6121	6121	7073	65803	37102	334	334	334
Read 4 Bytes									
(0, 0)	58002	58477	58021	29515	316248	173091	1066	1123	1093
(0, 1)	29611	3628[illegible]	3[illegible]	3[illegible]	3[illegible]	1[illegible]	1[illegible]	1[illegible]	1356
(0, 2)	29657	37017	36191	34770	322560	176642	1293	1569	1398
(1, 0)	28875	36258	3526[illegible]	3[illegible]	3[illegible]	1[illegible]	1[illegible]	1[illegible]	1412
(1, 1)	58361	58844	5[illegible]	3[illegible]	3[illegible]	1[illegible]	1[illegible]	1[illegible]	1325
(1, 2)	29588	35499	30208	34657	322547	179222	1247	1408	1270
(2, 0)	29979	37040	3[illegible]	3[illegible]	3[illegible]	1[illegible]	1247	1569	1322
(2, 1)	28139	36235	34785	32641	302799	169887	1293	1431	1369
(2, 2)	57579	58062	57599	32535	302693	170670	1247	1270	1249
Read 64 Bytes									
(0, 0)	[illegible]07744	929955	918905	408908	4381352	2398035	18770	19245	18935
(0, 1)	[illegible]50935	478696	460279	393536	4216640	2307883	19007	20272	19521
(0, 2)	[illegible]79501	579758	538170	476993	4426369	2423243	19053	22549	20808
(1, 0)	[illegible]73268	571294	520525	476993	4424823	2435851	19145	23032	21203
(1, 1)	[illegible]09739	936166	921822	488037	4541994	2512343	19076	19398	19188
(1, 2)	[illegible]49348	473636	456782	475305	4423507	2446804	19007	20157	19418
(2, 0)	[illegible]74027	579068	535487	475305	4423507	2469029	19007	22043	20535
(2, 1)	[illegible]72095	565429	518137	489451	4555159	2542512	19007	22549	20895
(2, 2)	[illegible]00332	920618	907492	468232	4356158	2456170	19007	19237	19073
Read 256 Bytes									
(0, 0)	[illegible]628902	3702565	3674076	1586442	16998330	9303655	75609	78231	76046
(0, 1)	[illegible]810819	1897023	1826232	1848174	17206343	9370227	75839	79841	77648
(0, 2)	[illegible]897828	2181970	2119170	1830492	17041721	9280577	75885	88305	83101
(1, 0)	[illegible]890399	2132060	2046512	1862700	17279325	9512215	75997	89708	84212
(1, 1)	[illegible]631085	3708365	3679649	1848508	17147673	9620444	75908	78484	76336
(1, 2)	[illegible]808220	1897000	1823103	1842516	17147673	9528055	75839	79542	77494
(2, 0)	1897391	2180659	2116159	1828370	17016021	9497681	75839	87040	82616
(2, 1)	1890422	2131301	2044241	1869796	17345151	9731236	75839	89202	83631
(2, 2)	3616296	3682191	3641053	1826248	16990334	9579806	75839	78346	76212

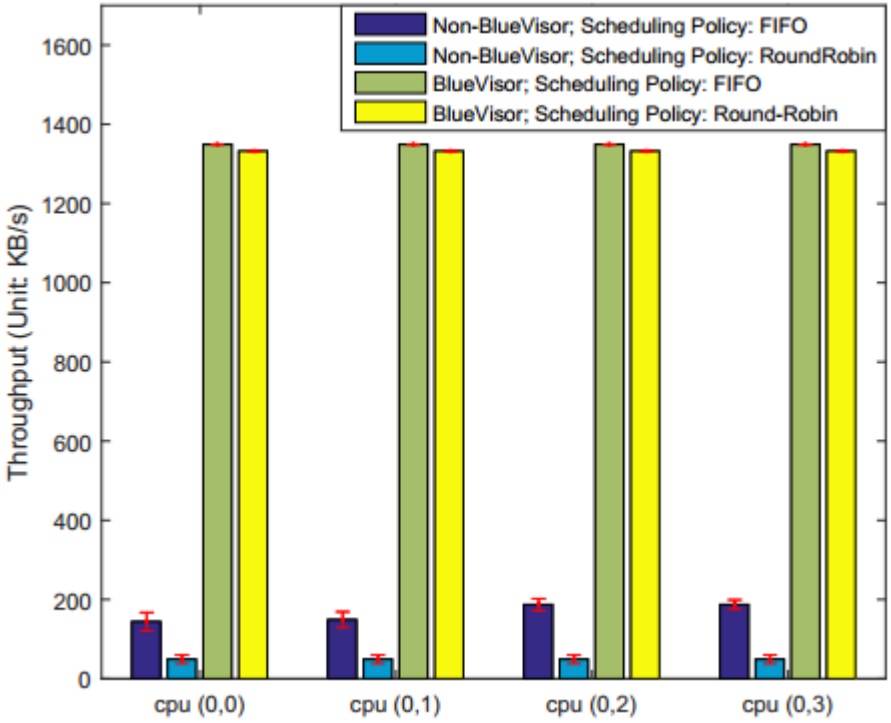
- As shown in the results:
  - The worst case of I/O response time (*max*) is smaller;
  - The I/O response time is more stable (*max* – *min*) - more predictable;

I/O Performance – Throughput

- Aims to evaluate the performance of the I/O system while CPU and I/O are fully loaded in a BlueVisor and non- BlueVisor system.

Number of Active CPUs	Evaluated I/O	I/O Requests	Scheduling Policy in non-BlueVisor	Scheduling Policy in BlueVisor
4	SPI nor-flash	Write	Round-Robin (Global) FIFO (Local)	Round-Robin FIFO

I/O Performance – Throughput



- From the results, we can see that, in BlueIO system:
  - The I/O throughput is higher;
  - The variance of I/O throughput is smaller.

I/O Performance – Timing-accuracy

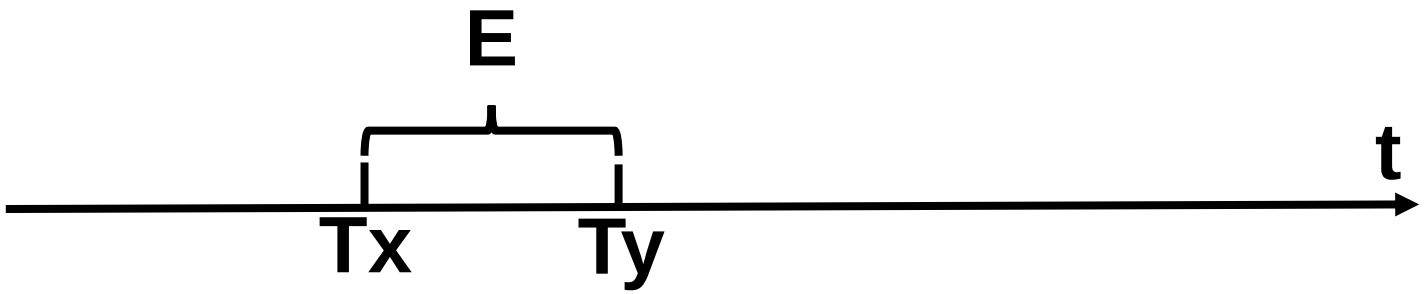
- Aims to evaluate the timing-accuracy of the I/O operations in a BlueVisor and non- BlueVisor system.

Number of Active CPUs	Evaluated I/O	I/O Requests	Scheduling Policy in non-BlueVisor	Scheduling Policy in BlueVisor
9	GPIO	Write I/O pins	Round-Robin (Global)	Round-Robin

I/O Performance – Timing-accuracy

- We define the error of I/O operations timing accuracy as:
  - **E = | Tx − Ty|**

- **Tx:** time at which I/O operation is required;
  Ty: the actual time that the I/O operation actually occurs.



- If ***E*** = 0, this I/O operation occurs exactly at the expected time – we say it is totally timing-accurate.

I/O Performance – Timing-accuracy

CPU Index	Non-BlueVisor								BlueVisor							
	E (unit: ns)				E (unit: clock cycle)				E (unit: ns)				E (unit: clock cycle)			
	Min	Med	Mean	Max	Min	Med	Mean	Max	Min	Med	Mean	Max	Min	Med	Mean	Max
(0,0)	3140.0	3140.0	3145.8	3160.0	314	314	315	316	0.0	0.0	0.0	0.0	0	0	0	0
(0,1)	3000.0	3000.0	3005.8	3020.0	300	300	301	302	0.0	0.0	0.0	0.0	0	0	0	0
(0,2)	2790.0	2790.0	2795.8	2810.0	279	279	280	281	0.0	0.0	0.0	0.0	0	0	0	0
(1,0)	2720.0	2720.0	2795.8	2810.0	279	279	280	281	0.0	0.0	0.0	0.0	0	0	0	0
(1,1)	3070.0	3070.0	3075.8	3090.0	307	307	308	309	0.0	0.0	0.0	0.0	0	0	0	0
(1,2)	2860.0	2880.0	2899.4	2940.0	286	288	290	294	0.0	0.0	0.0	0.0	0	0	0	0
(2,0)	2580.0	2580.0	2585.8	2600.0	258	258	259	260	0.0	0.0	0.0	0.0	0	0	0	0
(2,1)	2650.0	2650.0	2655.8	2670.0	265	265	266	267	0.0	0.0	0.0	0.0	0	0	0	0
(2,2)	2860.0	2930.0	2902.0	2950.0	286	293	290	295	0.0	0.0	0.0	0.0	0	0	0	0

Interrupt Handling

- Evaluate the response time of interrupt handling in BlueVisor and non-BlueVisor architectures

Number of Active CPUs	Evaluated I/O	I/O Requests	Scheduling Policy in non-BlueVisor	Scheduling Policy in BlueVisor
1	GPIO	Response Interrupt	None	None

Interrupt Handling

Interrupt handling (unit: clock cycles)

	Best Case	Worst Case	Mean
Native FreeRTOS	520	652	577
BS_vFreeRTOS (Fast IRQ)	10	10	10
BS_vFreeRTOS (Normal IRQ)	544	682	592

- Fast IRQ can be always completed at 10 cycles
- Normal IRQ takes more time than the IRQ in native FreeRTOS

Outline

- Virtualization Technology
- Networks-on-chip
- BlueVisor
- Experimental Evaluation
- **Conclusion**

Conclusion

- We are looking for a hypervisor, which enables:
  - Predictability
  - Improved system performance
  - Decreased software overhead
  - Good scalability
  - Isolation

Limitation

- Extra hardware overhead
- Communication overhead in the entrance of BlueVisor (Traffic congestion)
- Hard to be upgraded.

In the future

- Memory Virtualization
  - cache coherence
  - Isolation
- Support SMP OS
- Support More I/O Devices

UNIVERSITY of York

BlueVisor: A Scalable Real-time Hardware Hypervisor for Many-core Embedded System

Zhe Jiang, Neil C Audsley, Pan Dong

Real-Time Systems Group
Department of Computer Science
University of York, United Kingdom

School of Computer
National University of Defense Technology
China

RTAS 2018

RTS York