Research & Reading 1

What is Virtualization? - Virtual Machine Monitor (VMM) is a software-abstraction layer that partitions a hardware platform into one or more virtual machines, including CPU virtualization, memory virtualization and I/O virtualization.

CPU Virtualization - A CPU architecture is virtualizable if it supports the basic VMM technique of direct execution-executing the virtual machine on the real machine, while letting the VMM retain ultimate control of the CPU.

Challenge - Most modern CPU architectures were not designed to be virtualizable.

Techniques - Paravirtualization and direct execution combined with fast binary translation.

Future - With hardware support for x86 CPU VMMs, having a fully compatible virtual machine abstraction overrides any performance benefits from breaking compatibility.

Memory Virtualization - The VMM can page the virtual machine to a disk so that the memory allocated to virtual machines can exceed the hardware’s physical memory size.

Challenge - The VMM’s virtual memory subsystem constantly controls how much memory goes to a virtual machine, and it must periodically reclaim some of that memory by paging a portion of the virtual machine out to disk.

Future - Hardware-managed shadow page tables have long been present in mainframe virtualization architectures and would prove a fruitful direction for accelerating x86 CPU virtualization. Research mush look at resource management at the entire data center level, and the author expect significant strides will be made in this area in the coming decade.

I/O Virtualization - VMM in the virtual machine could directly read and write the device.

Challenge - Current computing environments, with their richer and more diverse collection of I/O devices make virtualizing I/O much more difficult.

Future - With adequate hardware support, safely passing these channel I/O devices directly to the software in the virtual machine should be possible, effectively eliminating all I/O virtualization overhead.

Research & Reading 2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | DDR4 SDRAM | GDDR5 SDRAM | MRAM | ZRAM |
| Speed | 2133 MHz | 7000 MHz | 667 MHz | 1000 MHz |
| Size | 64 GB | 4 GB | 128 MB | 512 MB |
| Access Time | Medium | Relatively Slow | Fastest | Fast |
| Availability | Available | Available | Available | Available |
| Cost | Medium | Medium | High | Medium |

Exercise 1

Answer

a) Offset = 4 bits, Index = 4 bits, Tag = 32 – 4 – 4 = 24 bits.

b) All number is in hexadecimal.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Address | Tag | Index | Offset | Hit/Miss | Type of miss |
| 0x100 | 1 | 0 | 0 | Miss | compulsory |
| 0x104 | 1 | 0 | 4 | Hit |  |
| 0x108 | 1 | 0 | 8 | Hit |  |
| 0x200 | 2 | 0 | 0 | Miss | compulsory |
| 0x204 | 2 | 0 | 4 | Hit |  |
| 0x410 | 4 | 1 | 0 | Miss | compulsory |
| 0x100 | 1 | 0 | 0 | Hit |  |
| 0x108 | 1 | 0 | 8 | Hit |  |
| 0x40C | 4 | 0 | C | Miss | conflict |
| 0x408 | 4 | 0 | 8 | Hit |  |
| 0x300 | 3 | 0 | 0 | Miss | conflict |
| 0x284 | 2 | 8 | 4 | Miss | compulsory |
| 0x280 | 2 | 8 | 0 | Hit |  |
| 0x304 | 3 | 0 | 4 | Hit |  |

c) Miss Rate = 6 / 14 = 43%

d) 4 blocks are occupied, including 2 blocks with index 0, 1 block with index 1, 1 block with index 8.

e) In this case, offset = 4 bits, index = 5 bits, then tag = 32 – 4 – 5 = 23 bits.

Exercise 2

Answer

a) Main Memory = 64 KB.

Offset = 3 bits, Index = 5 bits, Tag = 16 – 3 – 5 = 8 bits.

b)

|  |  |
| --- | --- |
| Address | Line |
| 0001 0001 / 0001 1 / 011 | 00011 / 3 |
| 1100 0011 / 0011 0 / 100 | 00110 / 6 |
| 1101 0000 / 0001 1 / 101 | 00011 / 3 |
| 1010 1010 / 1010 1 / 010 | 10101 / 21 |

c) They are

0001 1010 0001 1 / 000

0001 1010 0001 1 / 001

0001 1010 0001 1 / 011

0001 1010 0001 1 / 100

0001 1010 0001 1 / 101

0001 1010 0001 1 / 110

0001 1010 0001 1 / 111

7 bytes.

d) Total bytes = 32 \* 8 = 256 bytes.

Exercise 3

Answer

a) First Level Instruction, Offset = 3 bits, Index = 9 bits.

First Level Data, Offset = 3 bits, Index = 10 bits.

Second Level, Offset = 5 bits, Index = 16 – 1 = 15 bits.

b) CPI = 1 \* 1 + 1 \* 0.02 \* 10 + 1 \* 0.02 \* 0.1 \* 100 + 0.4 \* 1 + 0.4 \* 0.15 \* 10 + 0.4 \* 0.15 \* 0.1 \* 100 = 1 + 0.2 + 0.2 + 0.4 + 0.6 + 0.6 = 3

Exercise 4

Answer

a) Page Offset = 11 bits, 12 bits are needed to specify the physical page number, 21 bits are needed to specify the virtual page number.

b) Hit rate = 3 / 4 = 75%

Cache block size = 32B, each double variable = 8B, so consider every 4 references, the first miss and last 3 hit.

c) Hit rate = 0

4 double variables stored in one cache block, when we traversal A with step 16, every reference will cause cache miss.

d) Hit rate = 255 / 256 = 99.61%

Page size = 2KB, each page contains 256 double variables, so only 1 TLB miss in every continuous 256 references.

Exercise 5

Answer

8 physical frames

8KB page size, 4-entry fully associative TLB, 8 physical frames

a)

Virtual Address: 001 / 1 0001 0010 0011

Physical Address: 011 / 1 0001 0010 0011

TLB: Miss

Page: Hit

Page Replacement: Page Hit

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Valid | Dirty | Ref | Tag | Physical Page # |
| 1 | 0 | 1 | 2 | 6 |
| 1 | 1 | 1 | 5 | 1 |
| 1 | 0 | 1 | 1 | 3 |
|  |  |  |  |  |

b)

Virtual Address: 001 / 0 0101 0110 0111

Physical Address: 011 / 0 0101 0110 0111

TLB: Hit

Page: Hit

Page Replacement: TLB Hit

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Valid | Dirty | Ref | Tag | Physical Page # |
| 1 | 0 | 1 | 2 | 6 |
| 1 | 1 | 1 | 5 | 1 |
| 1 | 0 | 1 | 1 | 3 |
|  |  |  |  |  |

c)

Virtual Address: 011 / 1 0110 0101 0100

Physical Address: 000 / 1 0110 0101 0100

TLB: Miss

Page: Fault

Page Replacement: VPN-8 / PPN-0

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Valid | Dirty | Ref | Tag | Physical Page # |
| 1 | 0 | 1 | 2 | 6 |
| 1 | 1 | 1 | 5 | 1 |
| 1 | 0 | 1 | 1 | 3 |
| 1 | 0 | 1 | 8 | 0 |

d)

Virtual Address: 000 / 0 1010 1011 1100

Physical Address: 101 / 0 1010 1011 1100

TLB: Miss

Page: Hit

Page Replacement: Page Hit

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Valid | Dirty | Ref | Tag | Physical Page # |
| 1 | 0 | 1 | 0 | 5 |
| 1 | 1 | 1 | 5 | 1 |
| 1 | 0 | 1 | 1 | 3 |
| 1 | 0 | 1 | 8 | 0 |