Chinese Wall Isolation Mechanism and Its implementation on VMM

Guanhai Wang, Minglu Li and Chuliang Weng
Shanghai Jiao Tong University, China.

SVM09, Wuhan, China.

Agenda

- Scenarios
  - Where may covert channels happen between VMs?
- Proposal
  - What do we want to do?
- Design
  - How does it solve this kind of problems?
- Implementation
  - How to make it run on Xen?
- Evaluation
  - Are its overheads very high?
- Contributions

Scenarios I

- Zhenghong Wang and Ruby B. Lee[1] implemented a SMT/FU channel on a Pentium-4 processor with hyper-threading.

insider	observer

`int bit;`
`…`
`do {`
`  bit = get_bit();`
`  if ( bit == 1 )`
`  MULTIPLY();`
`  else`
`  NULL(); }`
`while ( !TX_end() );`
	`int time, dt;`
	`…`
	`time = 0;`
	`do {`
	`  dt = time;`
	`  RUN();`
	`  time = get_time();`
	`  STORE(time-dt); }`
	`while ( !RX_end() );`

Pseudo code for SMT/FU channel. This figure comes from [1]

[1] Zhenghong Wang and Ruby B. Lee. Covert and side channels due to processor architecture. In ACSAC '06: Proceedings of the 22nd Annual Computer Security Applications Conference, pages 473–482, Washington, DC, USA, 2006. IEEE Computer Society.

Scenarios II

- C. Percival[2] implemented a L1 cache missing channel on a Pentium-4 processor with hyper-threading .
  - The Trojan process access memory to evict cache lines owned by the spy process in L1 cache.
  - The spy process measures the amount of time needed to read many particular bytes.
- The two channels were not implemented on virtual machine systems, but we can use similar methods to implement channels between two VMs running on processors with hyper-threading.

[2] C. Percival. Cache missing for fun and profit. In BSDCan 2005, Ottawa, 2005.

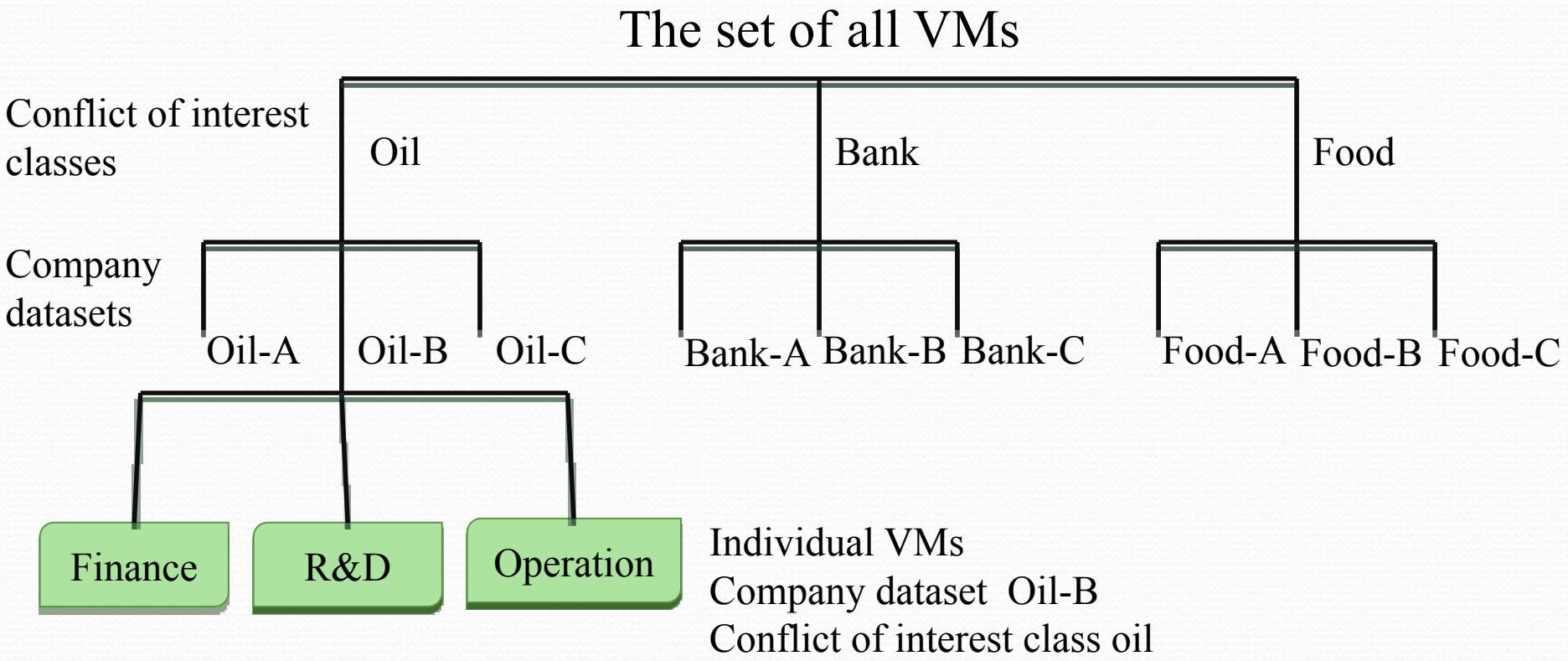
Proposal

- Build Chinese Wall Isolation(CWI) , a mandatory access control mechanism to block covert channel between VMs.
  - Prevents VMs belongs to different companies which are in competition from sharing hardware, then reduce the chance of building covert channels between VMs
- CWI based on Chinese Wall Policy regulates VMM allocating hardware to VMs.

Design I: Concepts

- Key concepts of CWI are based on Chinese Wall Policy.
  - VMs containing information of one company are defined as objects.
  - Hardware are defined as subjects.
  - Company datasets.
  - Conflict of interest class.
  - Session
  - Access rule
    - When a VM request an unit of hardware resources, if and only if the requested hardware was not used by its competitors, the request is granted, otherwise, it's denied.

Design II example of concepts



The idea of this figure comes from [3]

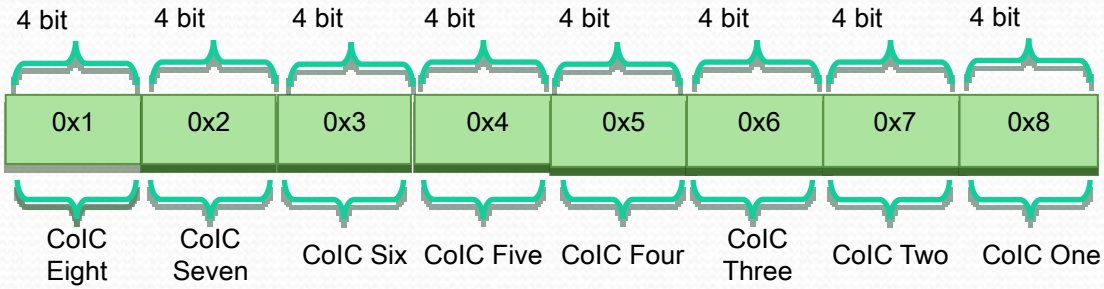
[3] D. F. C. Brewer and M. J. Nash. The chinese wall security policy. Proceedings of the 1989 IEEE Symposium on Security and Privacy, May 1989.

Design III: the basic idea

- Keeps all subjects' access histories to enforce the access rule.
  - A 32-bit record in VMM to keep one unit of hardware's access history.
  - Assign every company dataset a label and all VMs of the company have the label.
- CWI checks the access history of the requested hardware and the label of the requesting VM When VMM allocating hardware to aVM.

Design IV: record and label

- A 32-bit record of one unit of hardware.
  - Every 4-bit represents a company dataset ID. (CoIC denotes conflict of interest class)



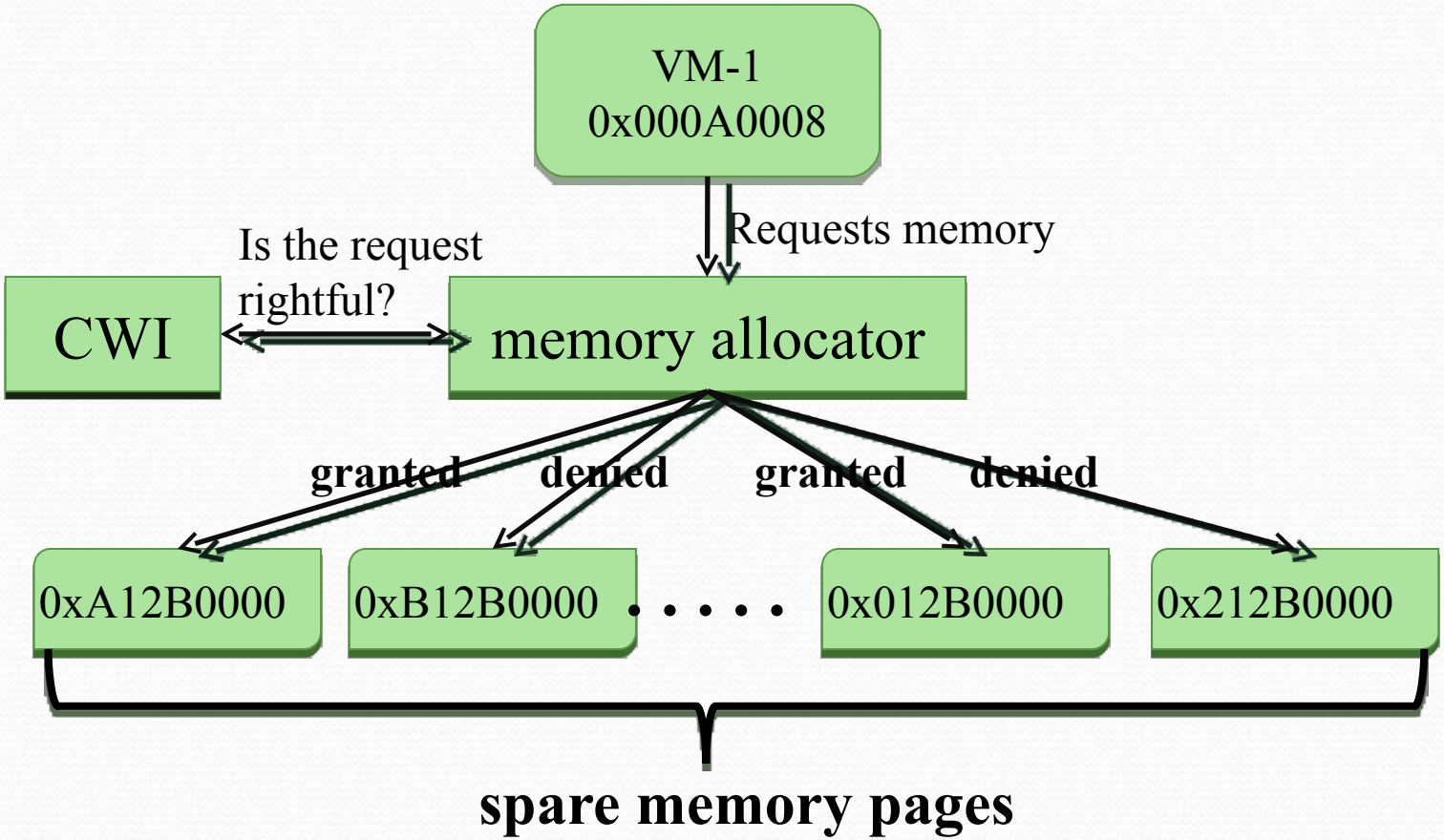
- A label comprises a dataset ID and a conflict of interest class ID.
  - A VM label $0x0002004$ means the VM is in the second dataset and belongs to the 4th conflict of interest class.

Implementation I

- Assigns labels to VMs, and store labels in their configuration files.
- Creates all hardware's access histories during VMM booting up.
- CWI checks the VM's label and the access history of the hardware when a VM requests hardware.
  - 3 places in which CWI checking them
    - Memory allocator when allocating memory to VM
    - CPU allocator when allocating processors to VMs
    - CPU scheduler when Virtual CPU migrating from one processor to another

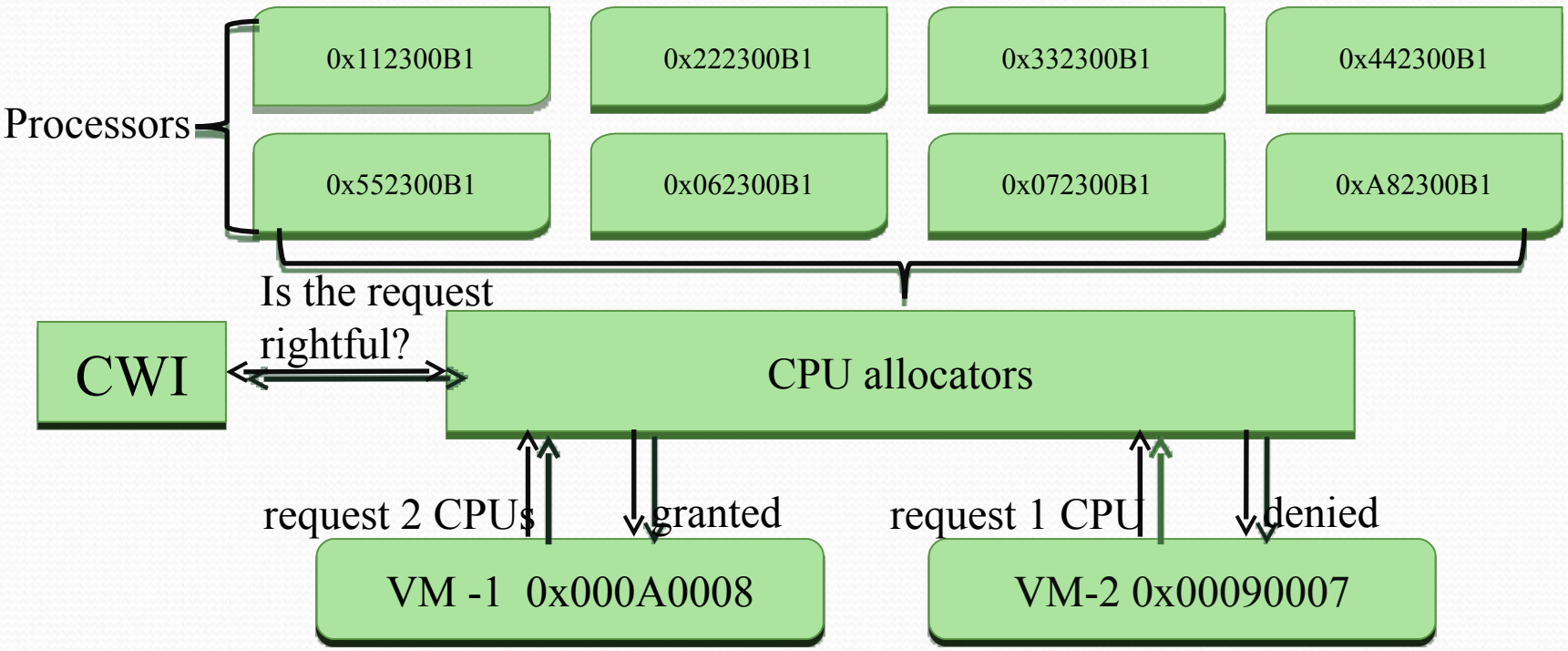
Implementation II

- Allocating memory



Implementation III

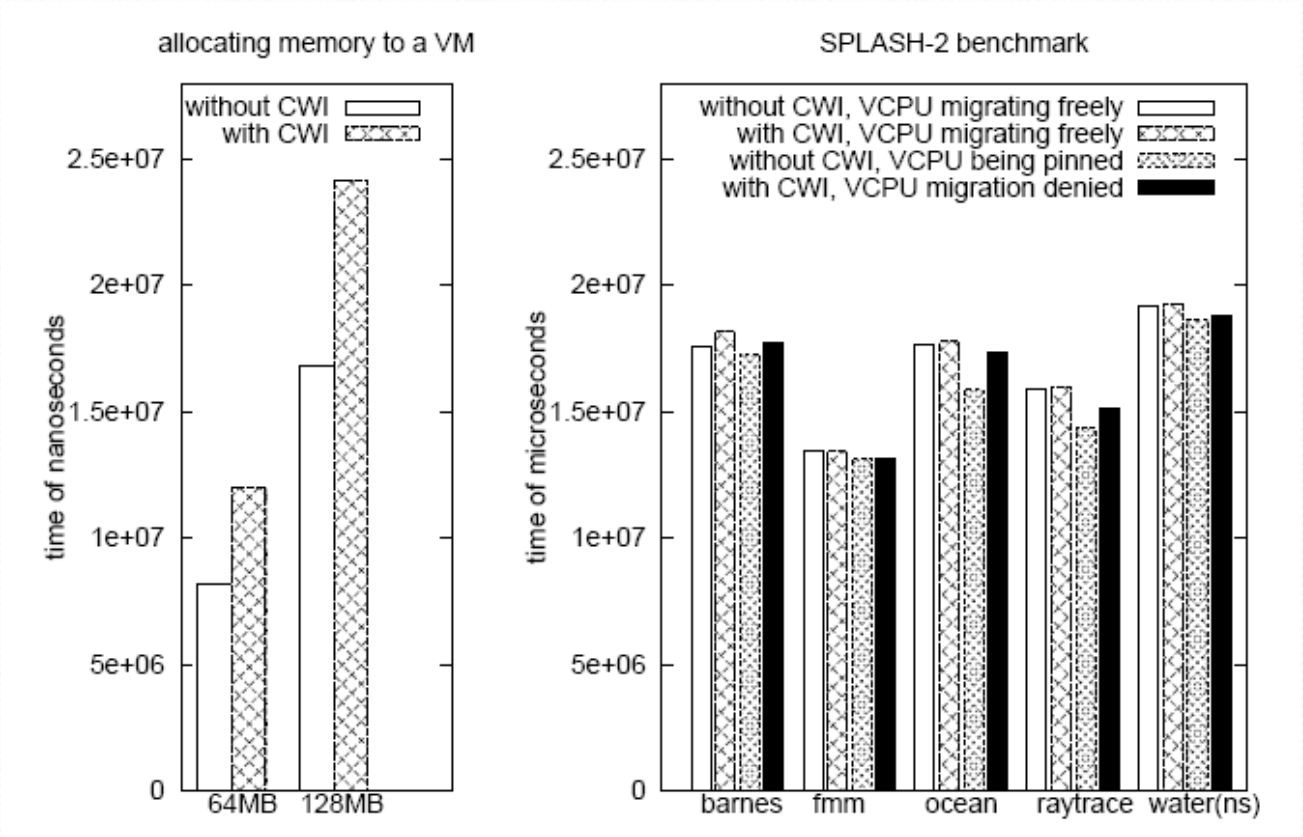
- Allocating Processors
  - Suppose there a 8 processors, each one has one core.
  - The CPU scheduler works quite when VCPU migrating.



Evaluation I

- CWI needs some memory to keep hardware's access histories.
  - Memory for physical processors is very small.
  - Memory for memory pages is considerable.
- Constructed a testbed to test CWI overheads on VMM performance
  - Measured the time increase of allocating memory and the computing time increase of SPALSH-2 application.
  - The testbed : a Dell server , 2 xeon quad core processors, 2 GB memory, Xen 3.2.1, and Debian Linux for both host and guest OSes.

Evaluation II



Evaluation III

- Overheads on performances are very low
  - Allocating Memory and CPUs are not on the critical path
    - The time of allocating memory increases by approximately 50%
  - VCPU migration is on the critical path
    - A small increase in computing time of SPLASH-2 application is about 5% on average.

Contributions

- Provides stronger isolation than VMM does.
  - It knows something which VMM doesn't know.
- Gets better hardware resources utilization than other mandatory access control(MAC) mechanisms do.
  - Defines every unit of hardware as one subject, and is a fine-grained access control mechanism.
  - Other MAC mechanisms on VMM define the whole system as one subject, and are coarse-grained access control mechanisms.

Thank you for your attention