MEI: A Light Weight Memory Error Injection Tool for Validating Online Memory Testers

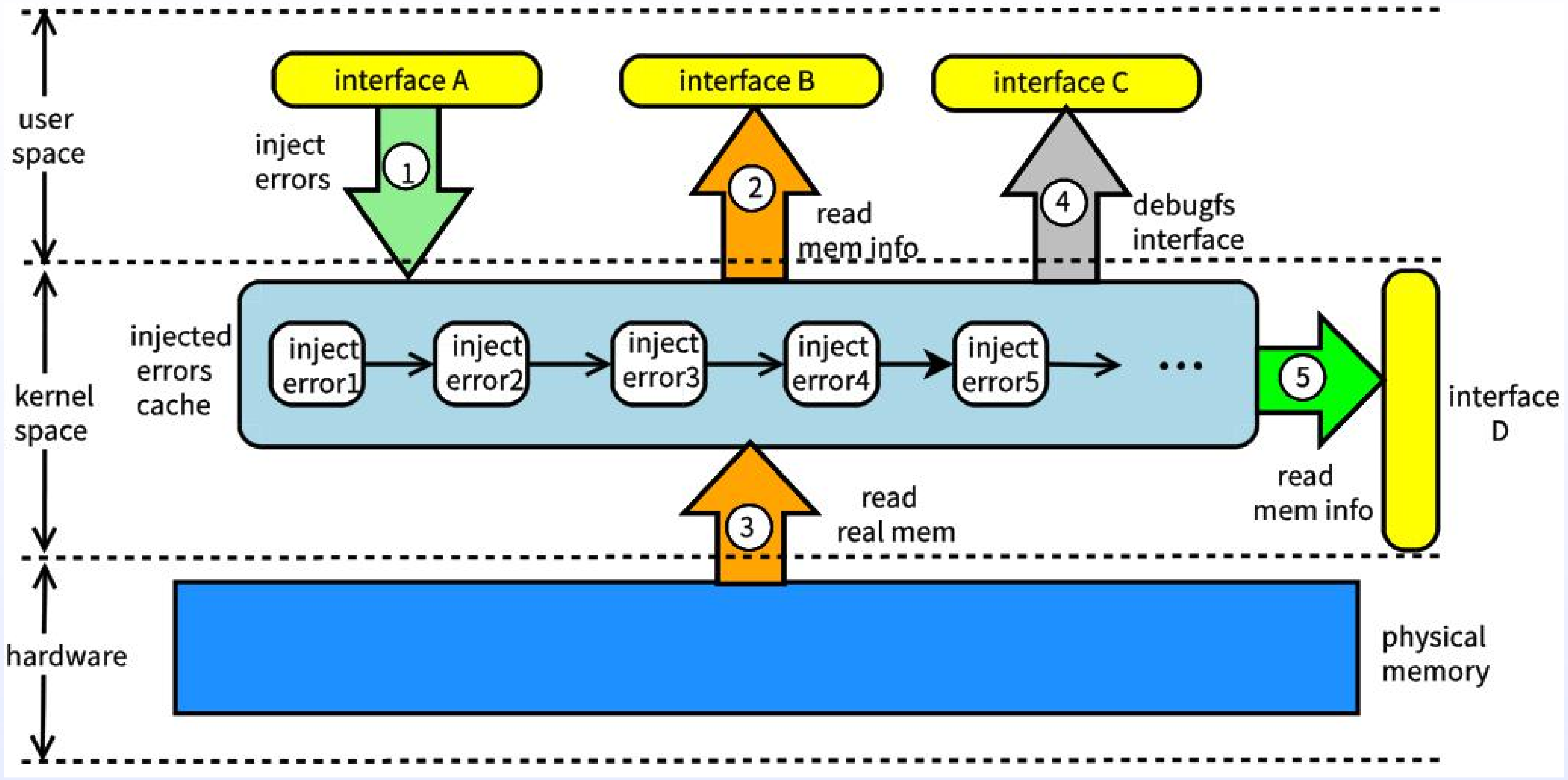
Wang Xiaoqiang <wangxq10@lzu.edu.cn>,
Wang Xuguo, Zhu Fangfang, Zhou Qingguo*, Zhou Rui
School of Information Science and Engineering,
Lanzhou University

Content

Introduction

- Memory chip bit-flip error rates are orders of magnitude higher than previously reported.
- A variety of methods to detect and handle memory errors are studied and developed, e.g. ECC, Memtest86+, RAMpage, COMeT+...
- Online memory testers can work with the OS (operating system) at the same time.
- Validation of these online memory testers is a hard work. e.g. real broken memory chips is hard to find and using a virtual machine to do this work is really complex.
- MEI - Memory Error Injection, implemented in software on Linux platform and easy to use.

Structure


user space
kernel space
hardware
interface A
interface B
interface C
inject errors
1
2
read mem info
4
debugfs interface
injected errors cache
inject error1
inject error2
inject error3
inject error4
inject error5
...
5
interface D
read mem info
3
read real mem
physical memory

Usage

- Encode the injected errors in the error described file
- Use the user space error injection tool memerr-inject to inject the errors
- Use interface C in fig.1 provided by MEI to show the injected errors information
- Modify the source code of memory testers to use the read interfaces (interface B or D)

Injected Errors Cache

- This cache is implemented in the kernel space.
- The injected memory errors are recorded in a memory area organized as a list.
- Even though each read operation has to traverse the whole list to find that whether the related error information is in the cache with low efficiency, luckily the number of elements in the list is very small, and traverse it would not consume much time. e.g. 1.2us with 20 elements in the list.

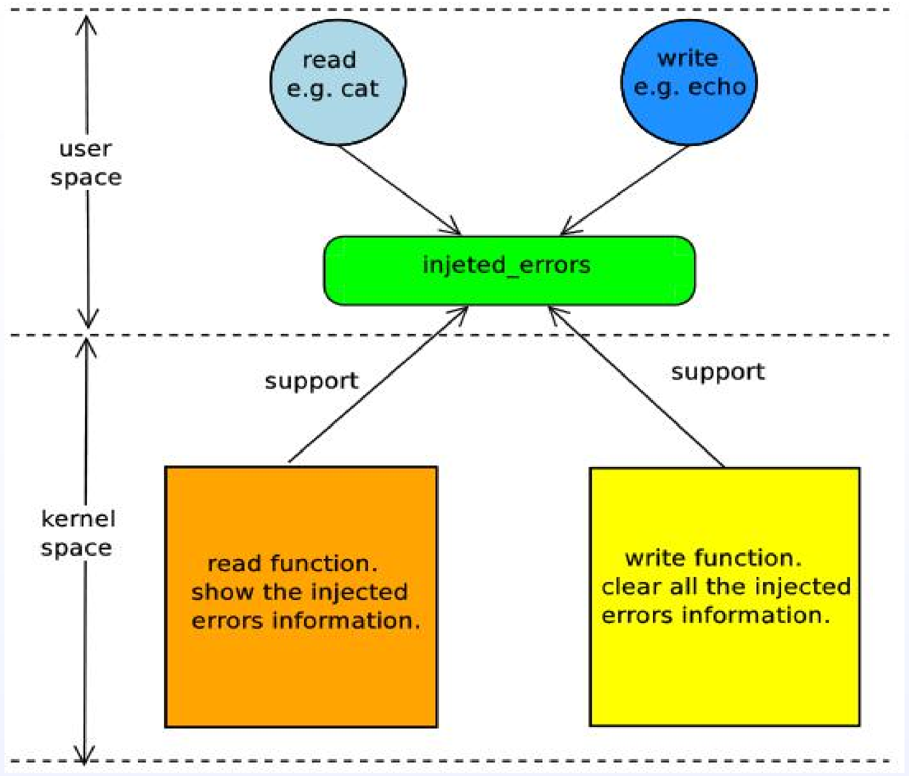
Injected Errors Format

- struct inject_memory_err {

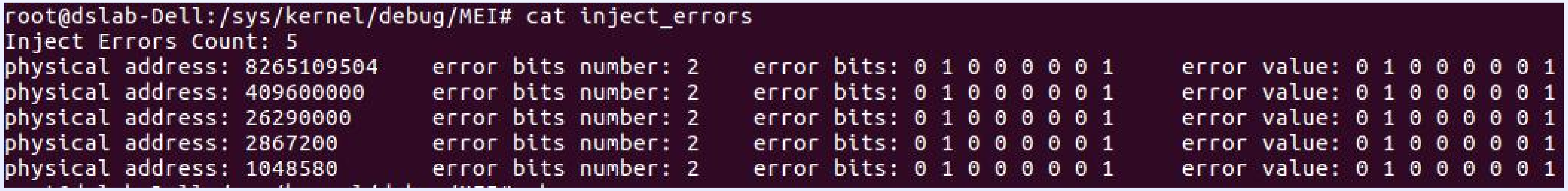
```
/* error injection physical memory address */
unsigned long phy_addr;
/* how many bits have errors */
int err_bit_num;
/* which bits have errors, the value is 0 or 1 */
int bit[BYTESIZE];
int bit_value[BYTESIZE];
/* the list pointer */
struct list_head lists;
};
```

User Space Interface

- read system call - Provide interface to online memory testers.
- write system call - Inject emulate errors into error injected cache.
- debugfs interface -
  Show and clear all injected errors.



F.g.2 debugfs interface



```
root@dslab-Dell:/sys/kernel/debug/MEI# cat inject_errors
Inject Errors Count: 5
physical address: 8265109504    error bits number: 2    error bits: 0 1 0 0 0 0 0 1     error value: 0 1 0 0 0 0 0 1
physical address: 409600000     error bits number: 2    error bits: 0 1 0 0 0 0 0 1     error value: 0 1 0 0 0 0 0 1
physical address: 26290000      error bits number: 2    error bits: 0 1 0 0 0 0 0 1     error value: 0 1 0 0 0 0 0 1
physical address: 2867200       error bits number: 2    error bits: 0 1 0 0 0 0 0 1     error value: 0 1 0 0 0 0 0 1
physical address: 1048580       error bits number: 2    error bits: 0 1 0 0 0 0 0 1     error value: 0 1 0 0 0 0 0 1
```

F.g.3 Output from debugfs interface

Kernel Space Interface

- The kernel space interface D is implemented in kernel module.
- The kernel module uses a mechanism in Linux to export interface D as a global variable in kernel space.
- Kernel space interface is used for online memory testers implemented kernel space. e.g. COMeT+

Evaluation Environment

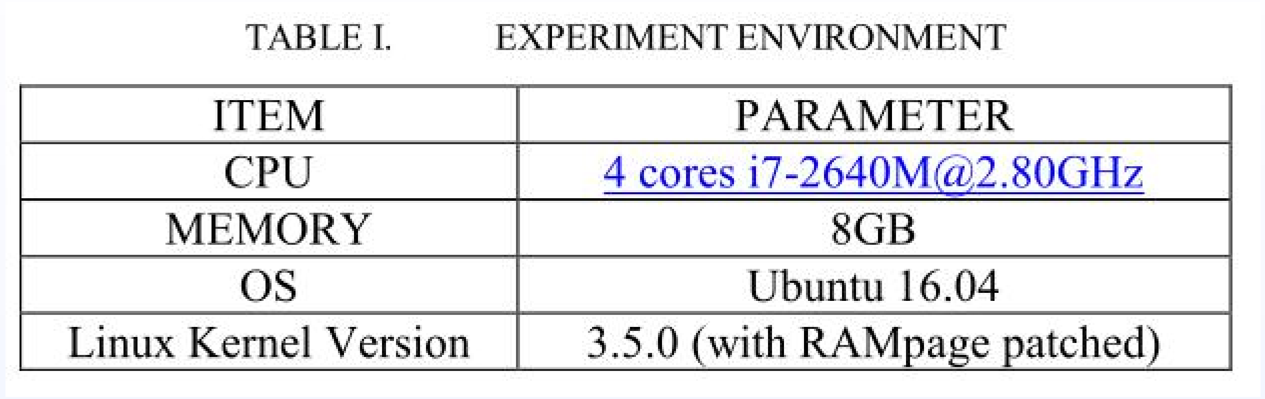
- 

TABLE I. EXPERIMENT ENVIRONMENT

ITEM	PARAMETER
CPU	4 cores i7-2640M@2.80GHz
MEMORY	8GB
OS	Ubuntu 16.04
Linux Kernel Version	3.5.0 (with RAMpage patched)

- RAMpage - User space online memory tester
- COMeT+ - Kernel space online memory tester

EVALUATION Result

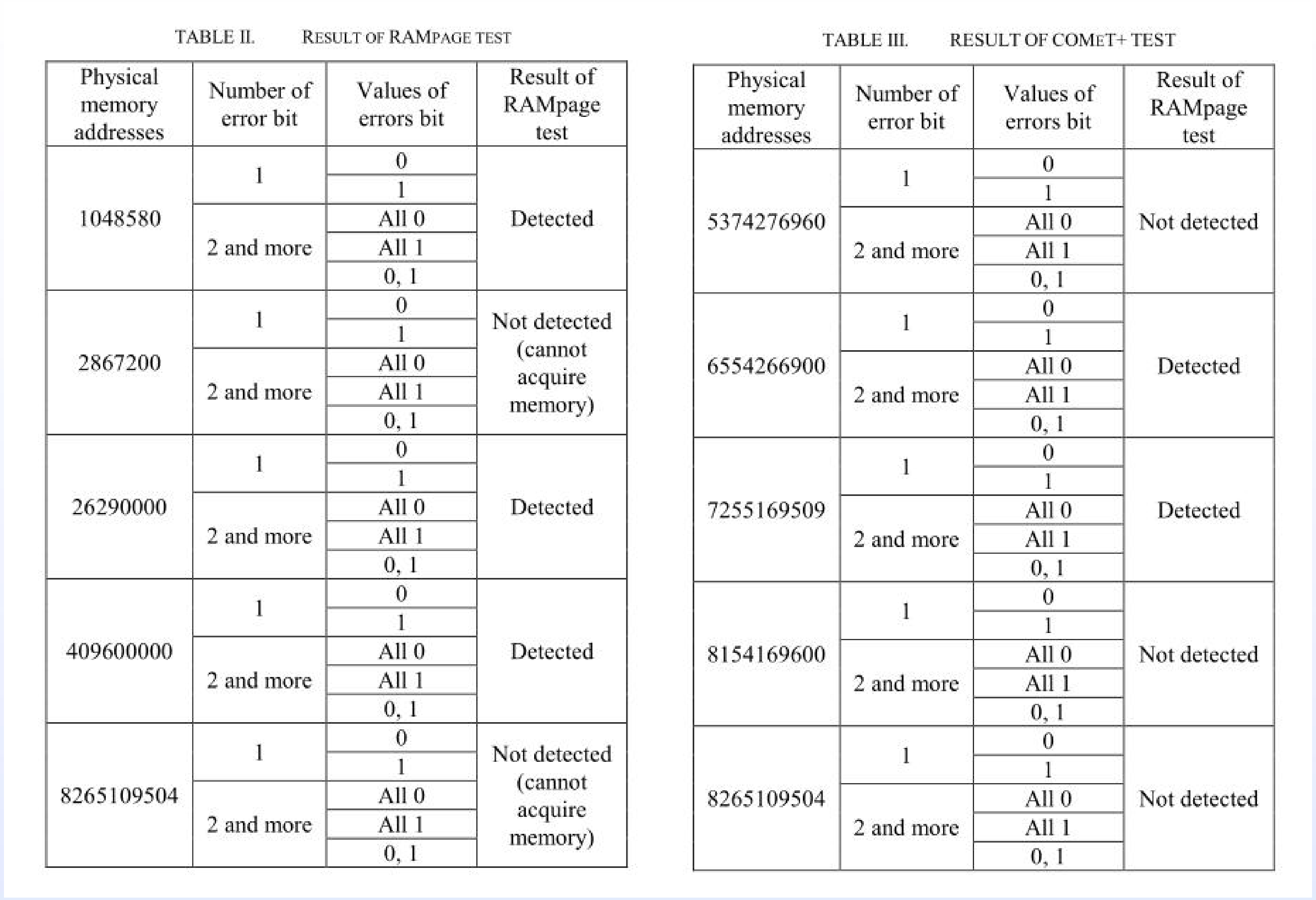


TABLE II. RESULT OF RAMPAGE TEST

Physical memory addresses	Number of error bit	Values of errors bit	Result of RAMpage test
1048580	1	0	Detected
		1	
	2 and more	All 0	
		All 1	
		0, 1	
2867200	1	0	Not detected (cannot acquire memory)
		1	
	2 and more	All 0	
		All 1	
		0, 1	
26290000	1	0	Detected
		1	
	2 and more	All 0	
		All 1	
		0, 1	
409600000	1	0	Detected
		1	
	2 and more	All 0	
		All 1	
		0, 1	
8265109504	1	0	Not detected (cannot acquire memory)
		1	
	2 and more	All 0	
		All 1	
		0, 1	

TABLE III. RESULT OF COMET+ TEST

Physical memory addresses	Number of error bit	Values of errors bit	Result of RAMpage test
5374276960	1	0	Not detected
		1	
	2 and more	All 0	
		All 1	
		0, 1	
6554266900	1	0	Detected
		1	
	2 and more	All 0	
		All 1	
		0, 1	
7255169509	1	0	Detected
		1	
	2 and more	All 0	
		All 1	
		0, 1	
8154169600	1	0	Not detected
		1	
	2 and more	All 0	
		All 1	
		0, 1	
8265109504	1	0	Not detected
		1	
	2 and more	All 0	
		All 1	
		0, 1	

CONCLUSION

- MEI is a light weight and easy-to-use memory errors injection tool aimed at simplifying the validation of online memory testers implemented in software level.
- The experiments show that most of the injected errors can be detected by RAMpage and COMeT+.
- The code of MEI has been released at https://github.com/wangxiaoq/MEI

THE END

THANKS

Q & A