JudgePenguin

基于Linux的应用程序稳态测试系统

致理-信计01 单敬博 2020012711

2023/4/16

选题背景

- 原始需求：**编程题目评测**
  - 选手需要编写能够在规定**时间、空间限制**内解决给定问题的程序
  - 时间限制：user time；内存限制：最大驻留集
- 测试系统的任务
  - 为用户程序提供**输入**；收集用户程序**输出**
  - 尽可能**准确**测量用户程序**时间**、**空间**使用情况
  - 防止用户程序进行连接网络、破坏系统等**恶意行为**

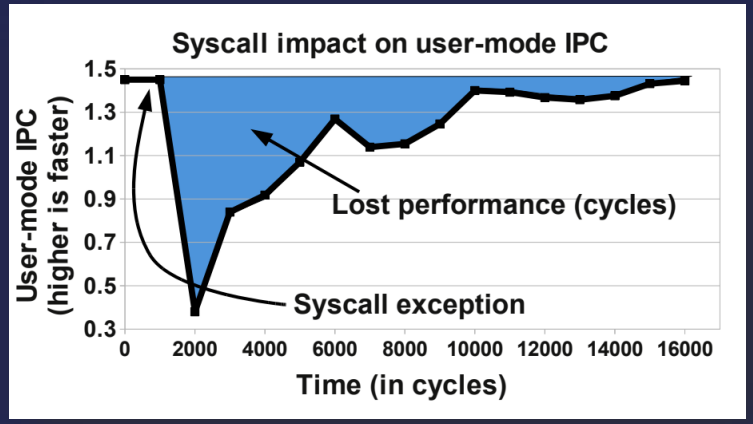
选题背景：现有测试系统及其问题

- 在系统中直接运行：Lemon, Cena, Arbiter, …
  - 无法有效防范用户程序的攻击（LemonF**cker：直接窃取答案）
- 基于docker / sandbox：LOJ, TUOJ, UOJ, …
  - 受虚拟化技术影响，时间测量结果波动较大（误差可能高达100%!）

#	用户	题目	语言	状态	分数
57706	[redacted]	A	python3	Time Limit Exceeded	90
57706	[redacted]	A	python3	Accepted	100

选题背景：现有测试系统及其问题

- OS中断与调度
  - 用户程序执行过程中OS仍会收到来自外设、网络、时钟等的中断
  - 用户程序也可能因为OS调度而暂停执行
  - 处理中断、任务切换不增加user time，但会对cache及TLB造成影响



from FlexSC

选题背景：现有测试系统及其问题

- 内存分配不连续
  - 常见OS的内存分配结果难以预测，用户程序访存时cache命中率不同
- 其他进程共享资源
  - 多核OS中其他进程对内存、L3cache等共享资源产生无法预测的影响

选题背景：现有测试系统及其问题

- 自研操作系统：JudgeDuck-OS
  - 屏蔽全部外部中断、为用户程序分配连续的内存、用户程序独享全部系统资源
  - **硬件驱动**需要自行编写，依赖特定硬件（duck依赖E1000网卡）

项目目标

- 只需在运行被测用户程序时提供稳定无干扰的环境
- 网卡驱动等不是本质需求 -> 借助Linux完善的驱动支持

- **目标**：在x86-64Linux中实现与JudgeDuck相似的稳定、准确的应用程序测试系统
- **实现方式**：Linux内核模块

项目目标

- OS中断与调度
  - 进入内核模块时暂停任务调度、关闭全部中断
  - 保存相关上下文，退出内核模块时恢复
- 内存分配不连续
  - 设法取得一段连续的**物理**内存，分配给用户程序

项目目标

- 避免与文件系统交互
  - 将输入/输出文件保存在内存中，在用户态完成输入输出等操作
- 对恶意行为的防护
  - 恶意行为大多是非法系统调用
  - 必要的库函数在用户态完成，不提供sys_exit外的系统调用

相关工作：JudgeDuck-OS

- 自研的应用程序稳态测试操作系统

- 依赖硬件的问题已在前文讨论
- 项目实现过程中的主要参考资料

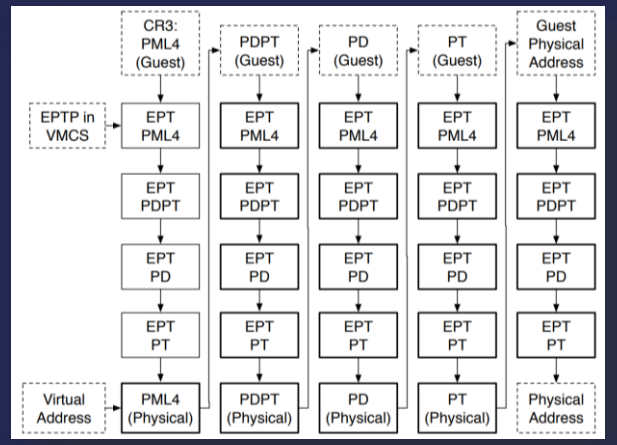
相关工作：RTAI / Xenomai

- Real Time Application Interface：Linux内核硬实时扩展
- 主要解决有较强**实时性**要求的任务调度与进程通信问题

- 只保证了任务执行的实时性(解决了任务切换问题)
- 外部中断、内存分配、资源共享等问题仍未解决
- 没有对用户程序恶意行为进行防护！

相关工作：RVM1.5 / JailHouse

- RVM1.5：Rust编写的Type-1.5 Hypervisor
- 从宿主OS启动的Hypervisor：启动其他OS，支持OS间通信



from Intel SGX Explained

- 验证了内核模块相关技术可行
- 引入Hypervisor，带来额外开销
- 提供了过多与本项目无关的功能，不如自己实现一份精简的

本周进展

- QEMU启动JudgeDuck-OS

```
[0.565091][INFO] Running tests
[0.565654][DEBUG] start = 0x12f8b8, len = 73400
[1.831513][INFO] time 0.000000 ms, memory 3104 KiB (3.0 MiB) (A: 3156 KB)
[1.833936][INFO] tsc 4262606788, trap 255, retcode 123
[1.834528][INFO] stdout size 120, stderr size 34
[1.835093][INFO] >>> stdout content (first 256 bytes) <<<
Hello world from x86-64!  curr tsc = 9061448456
a + b = 1087 (from stdin), e = 2.718281828459046, pi = 3.141592653589793
[1.836865][INFO] >>> stderr content (first 256 bytes) <<<
stderr working, memset 50.0 MiB ok
[1.838033][INFO] ======================================
[3.022298][INFO] time 0.000000 ms, memory 3096 KiB (3.0 MiB) (A: 3140 KB)
[3.022948][INFO] tsc 3996103798, trap 255, retcode 321
[3.023494][INFO] stdout size 119, stderr size 34
[3.024040][INFO] >>> stdout content (first 256 bytes) <<<
Hello world from i386!  curr tsc = 13616381806
a + b = 1087 (from stdin), e = 2.718281828459045, pi = 3.141592653589793
[3.025710][INFO] >>> stderr content (first 256 bytes) <<<
stderr working, memset 50.0 MiB ok
[3.026778][INFO] ======================================
[3.027345][INFO] Welcome to JudgeDuck-OS-64 !!!
[3.027884][INFO] ABI Version 0.04
[3.028428][INFO] Starting duck server
```

```
shanjb0221@shanjb0221-mini
Hello world!
e = 2.718281828459046
pi = 2 * atan2(1, 0) = 3.141592653589793
[tsc 6675678926][DEBUG] void Multiboot2_Loader::load() start
[tsc 6679411308][DEBUG] cmdline:
[tsc 6681561436][DEBUG] base 00000000 (0.0 MiB), len 0009fc00 (0.6 MiB), type 1
[tsc 6684531076][DEBUG] base 0009fc00 (0.6 MiB), len 00000400 (0.0 MiB), type 2
[tsc 6686586544][DEBUG] base 000f0000 (0.9 MiB), len 00010000 (0.1 MiB), type 2
[tsc 6688665626][DEBUG] base 00100000 (1.0 MiB), len 7fee0000 (2046.9 MiB), type 1
[tsc 6692600944][DEBUG] base 7ffe0000 (2047.9 MiB), len 00020000 (0.1 MiB), type 2
[tsc 6696265788][DEBUG] base fffc0000 (4095.8 MiB), len 00040000 (0.2 MiB), type 2
[tsc 6700032674][DEBUG] void Multiboot2_Loader::load() done
[tsc 6702082984][DEBUG] void PIC::init() start
[tsc 6704186194][INFO] Enabled Interrupts: 2
[tsc 6706347780][DEBUG] void PIC::init() done
[tsc 6708302704][DEBUG] void Timer::init() start
[tsc 6710248860][DEBUG] CPU brand string: [GenuineIntel]
[tsc 6712288304][WARN] Assume clk_freq Hz = round(tsc_freq, 100M)
[0.000007][DEBUG] tsc_freq = 3600000000, ext_freq = 1000000000
[0.000595][DEBUG] Userspace performance counters enabled
[0.001126][DEBUG] void Timer::init() done
[0.001677][DEBUG] void LAPIC::init() start
[0.002362][DEBUG] Switched to ACPI mode
[0.002884][DEBUG] remapped lapic = 0xffffff00fee00000
[0.003470][DEBUG] void LAPIC::init() done
[0.004029][DEBUG] void Memory::init() start
[0.004593][INFO] Kernel memory used: 3.1 MiB
[0.005180][DEBUG] n_huge_pages = 1021
[0.005700][DEBUG] void Memory::init_page_tables() start
[0.006247][DEBUG] void Memory::init_page_table_break() start
[0.006832][INFO] page_table_break = a00000 (10.0 MiB)
[0.007375][DEBUG] void Memory::init_page_table_break() done
[0.007931][DEBUG] uint64_t Memory::init_page_table_4k() start
[0.008480][DEBUG] uint64_t Memory::init_empty_kernel_page_table() start
[0.009385][DEBUG] uint64_t Memory::init_empty_kernel_page_table() done
[0.014644][DEBUG] uint64_t Memory::init_page_table_4k() done
[0.015252][INFO] vaddr_break = 7f800000 (2040.0 MiB)
[0.015809][DEBUG] void Memory::init_page_tables() done
[0.016367][DEBUG] void Memory::init() done
[0.016927][DEBUG] void Trap::init() start
[0.017536][DEBUG] void Trap::init() done
[0.018086][DEBUG] int PCI::init() start
[0.018740][DEBUG] PCI: 00:00.0: 8086:1237: class: 6.0 (Bridge device) irq: 0
[0.019396][DEBUG] PCI: 00:01.0: 8086:7000: class: 6.1 (Bridge device) irq: 0
```

本周进展

- QEMU启动RVM1.5

```
ubuntu@ubuntu:~$ ./test
Execute VMCALL failed.
You are in the Host mode.
ubuntu@ubuntu:~$ ./enable-rvm.sh
JAILHOUSE_DISABLE: Invalid argument
ubuntu@ubuntu:~$ ./test
Execute VMCALL OK.
You are in the Guest mode.
ubuntu@ubuntu:~$ ./disable-rvm.sh
ubuntu@ubuntu:~$ ./test
Execute VMCALL failed.
You are in the Host mode.
```

```
Initializing hypervisor...
config_signature = Ok("RVMSYS")
config_revision = 10
build_mode = release
log_level = info
arch = x86_64
vendor = intel
stats = off

[ 542.743433 INFO  0] Heap alloca
[ 542.749946 INFO  0] Hypervisor
    signature: Ok(
        "RVMIMAGE",
    ),
    core_size: 0x204e000,
    percpu_size: 0x80000,
    entry: 0x11d20,
    max_cpus: 0x4,
    online_cpus: 0x4,
}
[ 542.753929 INFO  0] Frame alloc
[ 542.755571 INFO  0] Hypervisor
[ 542.757175 INFO  0] Root cell i
```

```
[ 542.762752 INFO  0] CPU 0 init...
[ 542.762752 INFO  3] CPU 3 init...
[ 542.762752 INFO  2] CPU 2 init...
[ 542.762752 INFO  1] CPU 1 init...
[ 542.763849 INFO  0] successed to turn on VMX.
[ 542.764033 INFO  3] successed to turn on VMX.
[ 542.764347 INFO  2] successed to turn on VMX.
[ 542.764803 INFO  1] successed to turn on VMX.
CPU 3 init OK.
CPU 2 init OK.
CPU 1 init OK.
CPU 0 init OK.
[ 542.771606 INFO  0] Primary CPU init late...
Activating hypervisor on CPU 2...
Activating hypervisor on CPU 3...
Activating hypervisor on CPU 0...
Activating hypervisor on CPU 1...
[ 576.346545 WARN  0] Hypercall not supported: 2333
Deactivating hypervisor on CPU 1...
Deactivating hypervisor on CPU 3...
Deactivating hypervisor on CPU 2...
Deactivating hypervisor on CPU 0...
[ 578.016471 INFO  3] successed to turn off VMX.
[ 578.016634 INFO  1] successed to turn off VMX.
[ 578.016637 INFO  2] successed to turn off VMX.
[ 578.016934 INFO  0] successed to turn off VMX.
```

本周进展

- 学习《The Linux Kernel Module Programming Guide》
- 编写了内核模块Hello world

```
shanjb0221@shanjb0221-minipc:~/lkmpg/hello-1$ sudo insmod hello-1.ko
shanjb0221@shanjb0221-minipc:~/lkmpg/hello-1$ sudo rmmod hello-1.ko
shanjb0221@shanjb0221-minipc:~/lkmpg/hello-1$ sudo dmesg | tail -n 2 | grep world
[ 1887.564719] Hello world 1.
[ 1890.737601] Goodbye world 1.
```

下周计划

- 尝试关闭调度、关闭中断，使内核模块能够独占CPU运行
- 从RVM1.5中学习在Linux中预留物理内存的方法

感谢聆听 & 欢迎提问

致理-信计01 单敬博 2020012711