Automated on-execute test using VirtualBox

Junichi Murakami
Executive Officer, Director of Advanced Development Division

FFRI,Inc.
http://www.ffri.jp

Ver2.00.01

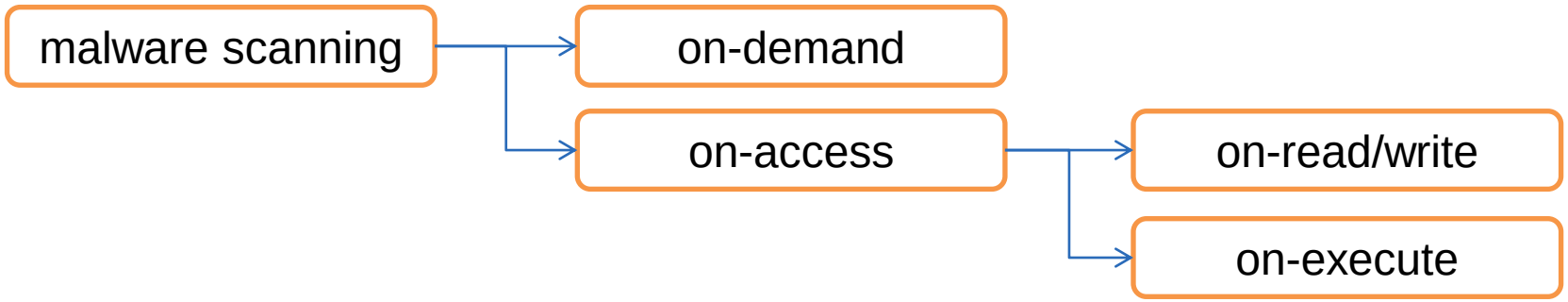
Agenda

1. Background and motivation
2. Overview of a test
   - automated on-execute test
   - virtualization software and automation methods
   - Oracle VM VirtualBox and its automation
   - example of VBoxManage
3. Automation script
   - FFRI AutoMonkey
   - design concept
   - throughput
   - performance
4. References
5. Contact information

1.Background and motivation

- Automated test against a large amount of malware is required to evaluate a malware detection engine
- Testing methods are classified into on-demand and on-access testing
- on-execute test which is a kind of on-access test has to execute malware one by one
- Therefore automation based on virtualization is required
- This slides describes automated on-execute test method using VirtualBox

* type of malware scanning



2.1.Automated on-execute testing

- Basic steps are following
  1. Copy malware into a guest
  2. Execute copied malware in the guest
  3. Analyze or detect malware in the guest
  4. Preserve the result after execution is terminated
  5. Revert the guest back to original condition
  6. Go to 1.

- Required functions to execute above are following
  a. Copying a file to a guest from a host (copy-to)
  b. Executing arbitrary a program in a guest from a host(exec)
  c. Copying a file from guest to a host (copy-from)
  d. Reverting a guest condition based on a snapshot(revert)

→ All functions can be achieved by making a communication interface between a host and a guest using TCP/IP. We considered the way we do not need to involve developing software as possible as we could

2.2.virtualization software and automation methods

- Use functions which virtualization software has natively
- VMware(licensed) and VritualBox have all the features we need
  → We considered using VirtualBox because of the cost advantage
- QEMU+KVM can be used by 3rd party software(ex: libguestfs + winexe)
  - "Malware Analysis: Collaboration, Automation & Tuning", Shmoocon 2013
    http://www.slideshare.net/xabean/malware-analysis-16674048

software	Licence	copy-to	copy-from	exec	revert	method
VMware Workstation	Proprietary	○	○	○	○	VIX API
VMware ESX(#1)	Proprietary	○	○	○	○	VIX API
Oracle VM VirtualBox	GPL2	○	○	○	○	VBoxManage
QEMU + KVM	GPL2(#2)	×	×	×	○	Libvirt

#1 ESXi can also use VIX API for 60days by registering a evaluation license.
#2 KVM's parts are licensed under various GNU licenses(GPL, GPL2, LGPL2, etc.)

2.3.Oracle VM VirtualBox and its automation

- A kind of x86 virtualization software, currently developed by Oracle
- Version 4.0 and later, fully open source software (GPL2)
- Supporting various host and guest environments
  - HostOS : Windows, Linux, Mac OS X, Solaris
  - GuestOS : Windows, Linux, FreeBSD, OpenBSD, Mac OS X Server, Solaris,etc.
- CLI is available (VBoxManage), friendly to automation
  - startvm , pause, resume, poweroff, clonevm, showvinfo
  - copyto, copyfrom, exec
  - taking snapshot and reverting
  - control virtual machine devices status, etc.

2.4.Example of VBoxManage

* starting a guest

```
% vboxmanage  startvm   vm
```

* power off a guest

```
% vboxmanage  controlvm  vm  poweroff
```

* reverting a guest based on a snapshot

```
% vboxmanage  snapshot  vm  restore  snapshot-1
```

* execute a program in a guest from a host

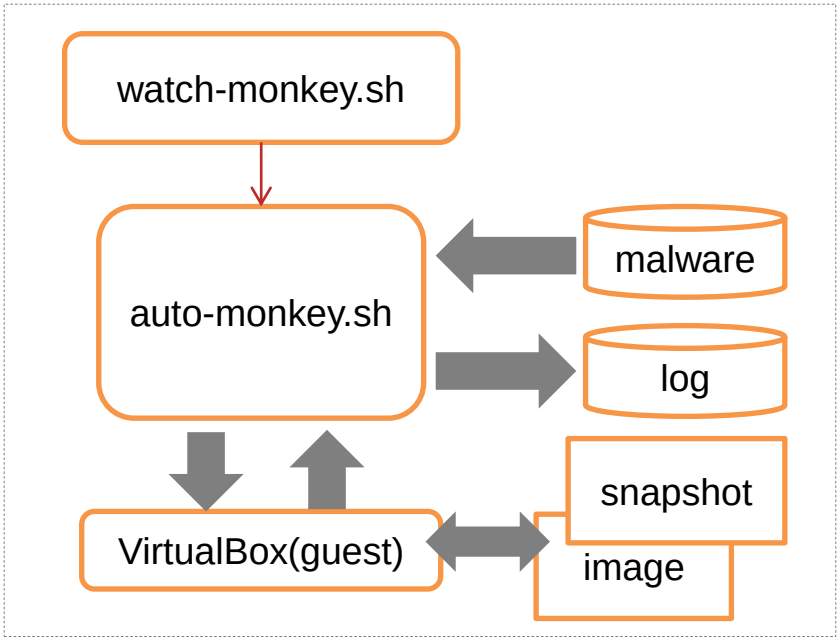
```
% vboxmanage  guestcontrol  exec  vm  --image “c:/windows/system32/calc.exe” ¥
    --username admin --timeout  60000  --wait-exit
```

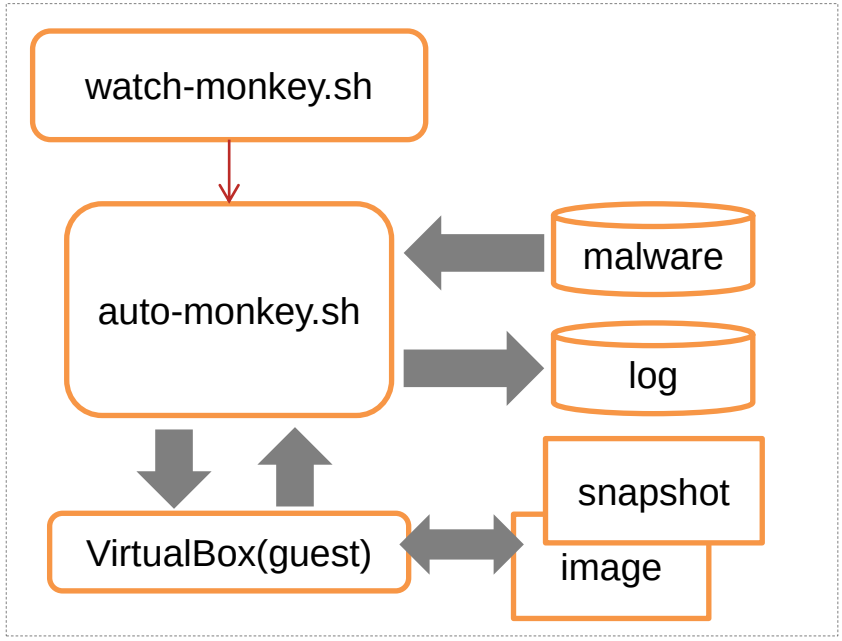
* copying a file to a guest from a host

```
% vboxmanage  guestcontrol  vm  copyto  “/some/file”  “c:/file.txt”  --username admin
```

3.1.FFRI AutoMonkey

- Automation script using VBoxManage, just a shell script
  - auto-monkey.sh:automation for copy, exec, copy, revert steps
  - watch-monkey.sh : watch dog script for the monkey
- It can execute multiple test simultaneously, works individually
- Published at our website below, see README for the detail (License: BSD)
  - http://www.ffri.jp/research/freeware.htm





3.2.Design concept

- conform to KISS principle
- Estimation of remaining time is important for this kind of test
  - we cannot determine when it would finish if the script hangs up
- Stability of VBoxManage (and VIX API) is the lifeline for the automation
- In fact, error occurs when it runs long time
  - Failure by error
    - exits immediately
    - resumed a test automatically by watch-monkey.sh
  - Hanging up(stuck) by error
    - watch-monkey.sh monitors lifetime of a VirtualBox process
    - if it is stuck, kill and resume

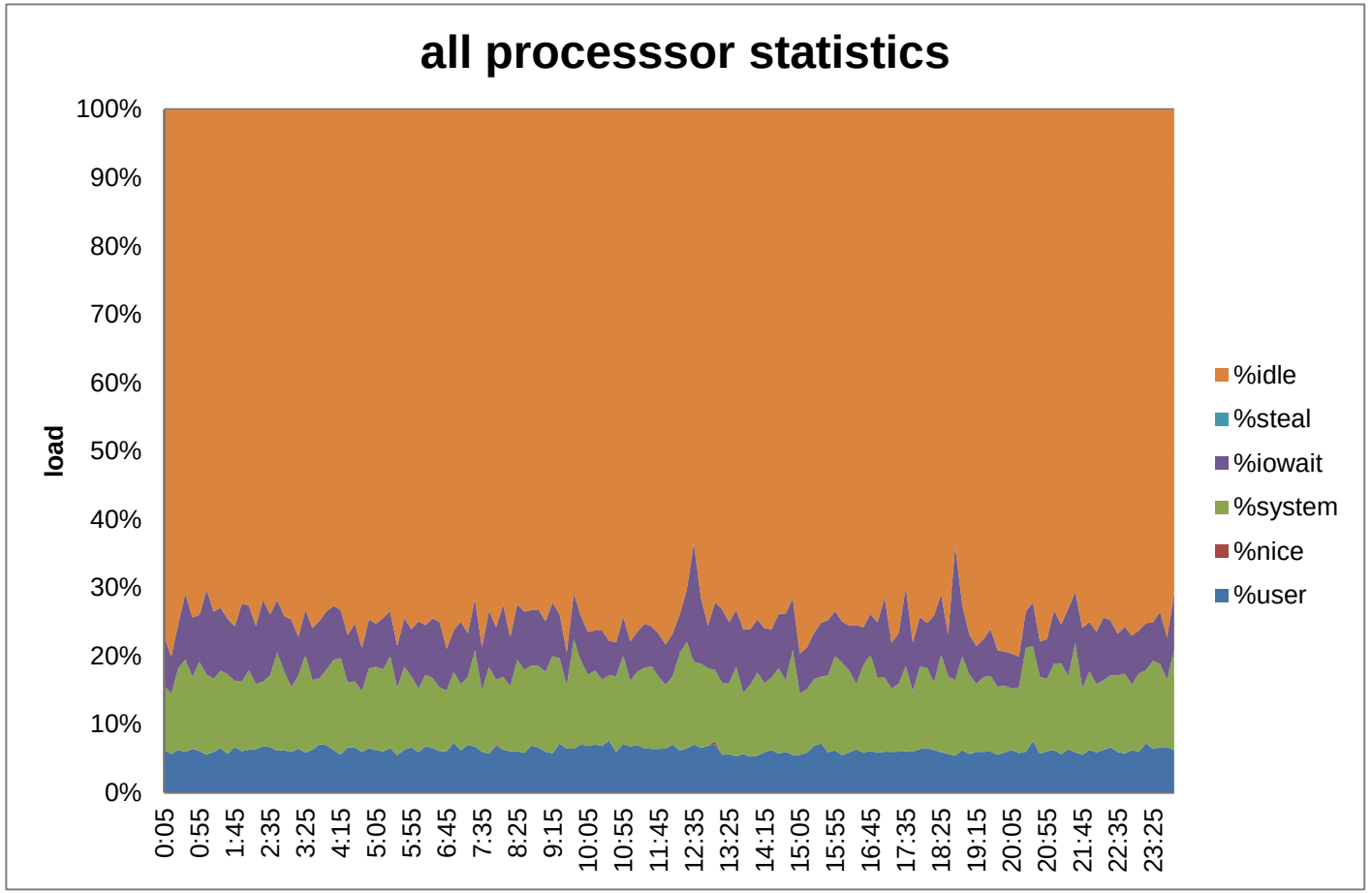
3.3.Throughput

- Testing under 1host and 7guest environment
- Processed 20,000 malware, each execution time was 60 seconds
  - total elapsed time: 37h15m
  - throughput : 8.95 malware/minute
    # if malware execution terminated less than 60 seconds, the script processes next item.
- Host and guest environment is following

Hardware	CPU: Intel(R) Core(TM) i5-4570 CPU @ 3.20GHz Memory: 8GB HDD: 1.8TB x 1
Host OS	Ubuntu 13.04 + VirtualBox 4.2
Guest OS	Windows XP SP3(x86) + FFR yarai 2.3 CPU:1 CPU Memory:750MB

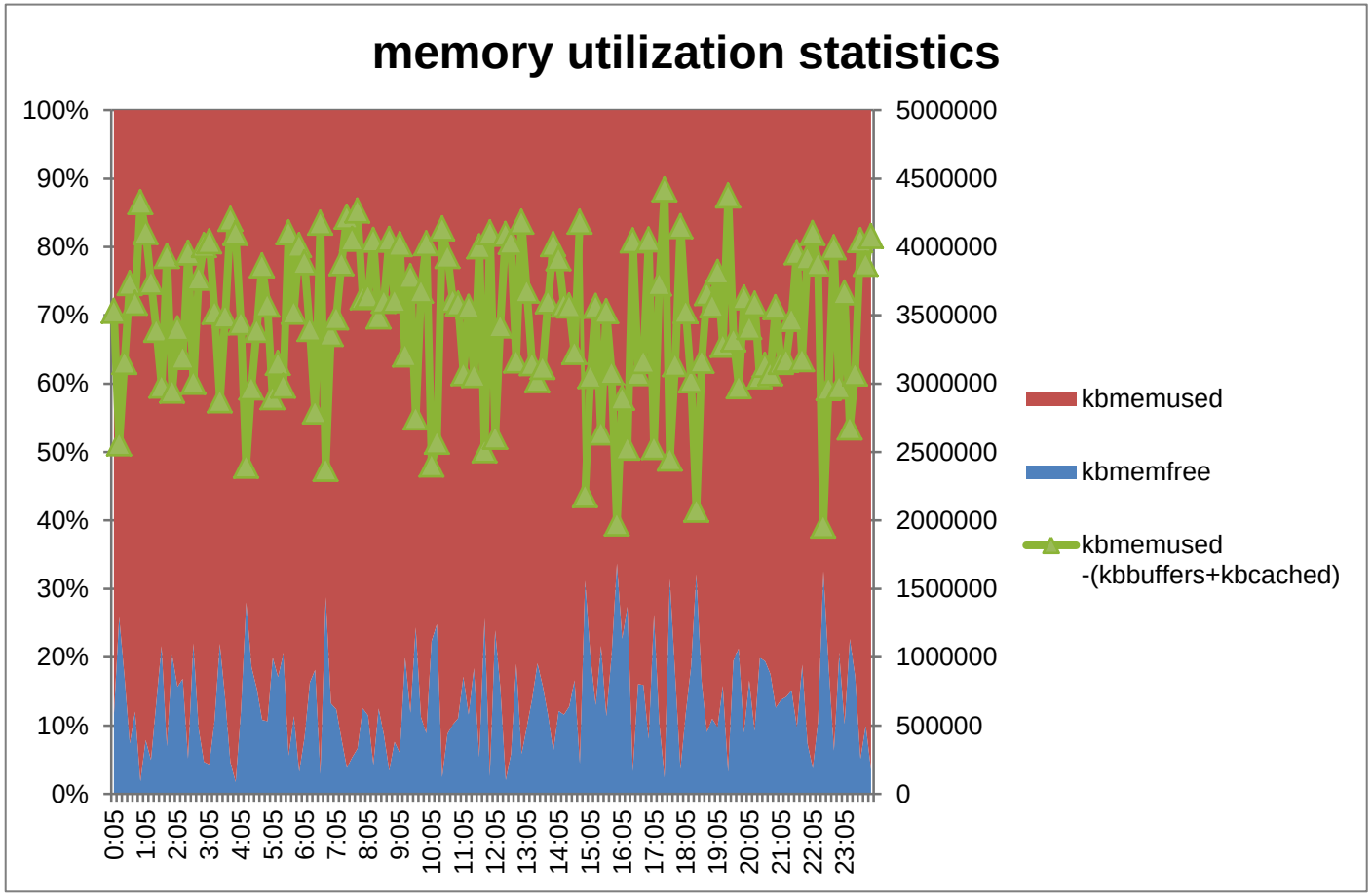
3.4.performance - processor

- About 70% of total processor is idle state(each core also indicates same trend)



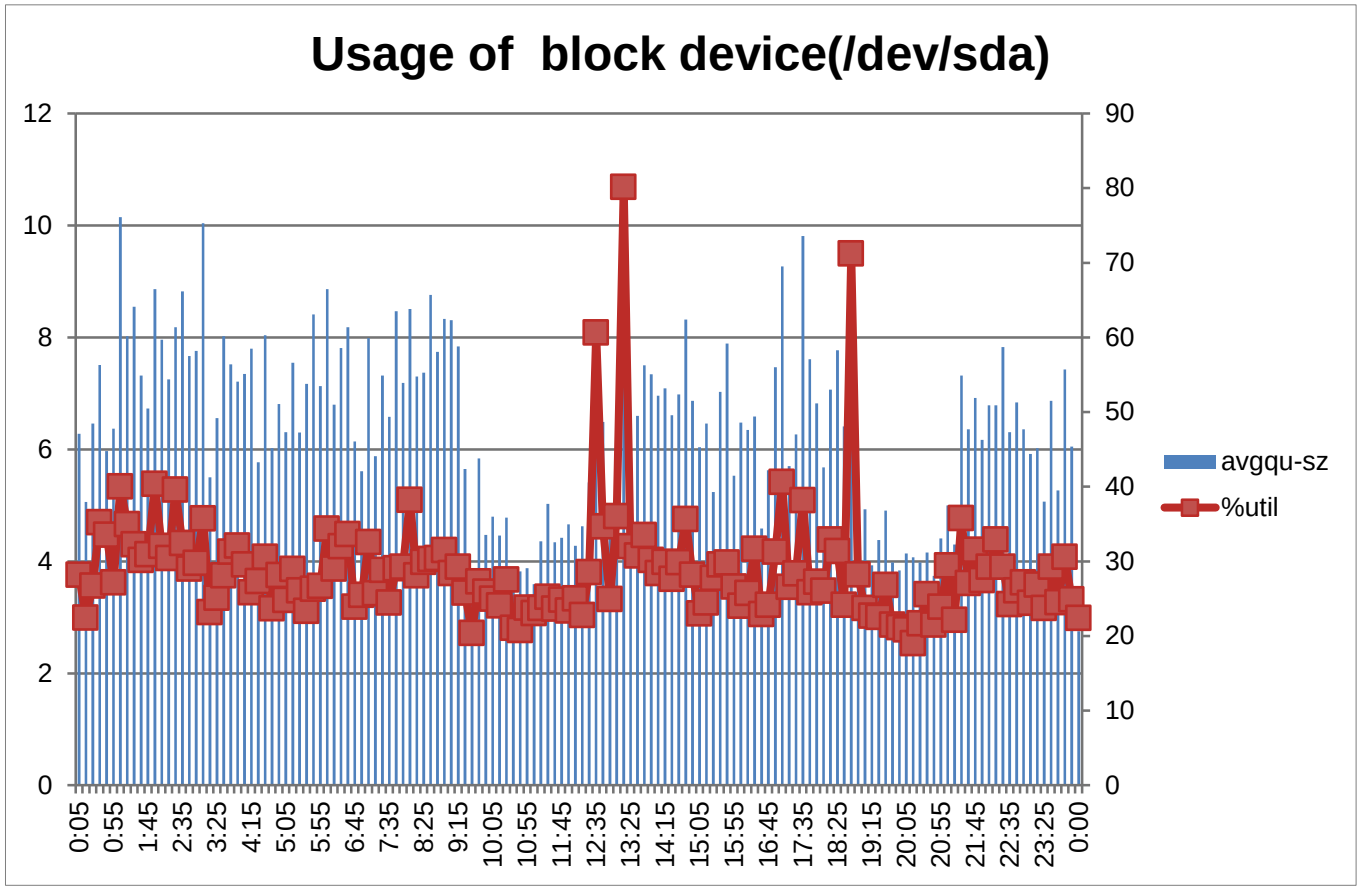
3.4.performance - memory

- Consuming about 80% - 90% memory steadily
  - real memory usage is between 2.5 and 4.0GB



3.4.performance – Disk IO

- Disk busy ratio(%util) stays around 30% steadily
- The number of queued requests is between 4 and 8



3.4.performance - consideration

- None of CPU, memory and IO wasn’t bottleneck under 1host and 7guest environment
- It seems we can add some more guests up to around 10 VMs according to memory usage
- However, we have to consider requirement of a process which is executed in a guest (cpu, memory)

References

- http://www.ffri.jp/assets/files/research/freeware/FFRIAutoMonkey-1.0.tgz
- https://www.virtualbox.org/manual/UserManual.html
- http://www.slideshare.net/xabean/malware-analysis-16674048
- http://www.youtube.com/watch?v=peHdyUlchSM
- http://libguestfs.org/
- http://sourceforge.net/projects/winexe/files/

Contact Information

- E-Mail
  - research-feedback@ffri.jp
- Twitter
  - @FFRI_Research