VESPER
(Virtual Embraced Space ProbER)

Sungho Kim <sungho.kim.zd@hitachi.com>
Satoru Moriya <satoru.moriya.br@hitachi.com>
Satoshi Oshima <satoshi.oshima.fk@hitachi.com>

Hitachi, Ltd., Systems Development Laboratory
Linux Technology Center
agenda

1. background
2. concept & structure
3. demo
4. evaluation
5. conclusion
1. Background
Trends on Server Systems

Server systems are required for

- **High availability & serviceability**
  - Mission-critical enterprise server
  - Cloud computing
  - Solution: HA cluster

- **Better resource utilization**
  - Hardware (CPU, memory) performance higher
  - Investment cost lower
  - Solution: Virtualization

Clustering Virtual Machines
Clustering VMs

We consider clustering VMs from 2 viewpoints

• Cluster model
  – How to set up cluster with VMs

• Monitoring Vms
  – How to health-check VMs
  – Likely to be dependent to the model

Solution: Linux-HA (Heartbeat)
  – Well-known cluster manager
  – Manage VMs
Brief Introduction to Heartbeat

• What is Heartbeat? (*) http://www.linux-ha.org/
  – A cluster management tool
  – Fail-over, load-balancing support

• What is the main feature in model?
  – Cluster resource manager which deploys resources onto proper cluster node if needed

• What is the cluster resource?
  – Every components needed to provide services to users
  – IP address, any server process like Web, DNS, and other servers, etc.
Cluster Model in Heartbeat

In Heartbeat, VMs are treated as resources because of simple manageability.

Conventional Cluster

- Cluster managers run on guests & monitor httpd as a resource.
- It is difficult to identify which guest running on which physical machine from the manager.

New Concept

- Cluster manager runs on host & monitors guests as a resource.
- It is clear that which guests running on which machine from the manager.

In Heartbeat, VMs are treated as resources because of simple manageability.
Monitoring in Heartbeat

Heartbeat monitors VMs with the conventional way

- **Polling**
  - Send health-check msg periodically

- **Failure detection depends on polling cycle**
Consideration on the Monitoring VMs

Heartbeat way of monitoring efficient enough?
--The same shortcomings exist

• Response latency
  – Periodic polling way of message passing
  – Deadtime allowance to ensure failure

• No method to find out why fail over
  – Service-bound failure?
  – System-wide(memory pressure ,etc.) failure?

Other health-check source to improve on the latency and failure analysis?
Probing Virtual Machines

Probing VMs can provide

- **Response latency improvement**
  - Event (trigger of fail-over) driven way
  - Immediate detection on failure

- **Failure analysis facility**
  - Memory pressure?
  - Network congestion?
  - etc.

![Diagram of Host, Guest, Cluster Manager, Immediate report, Failure, VMM]
VESPER

What features of probing technology needed for probing VMs in HA?

- Probe anywhere we want
- Insert probe dynamically
- Insert/delete probe from host
- Access probed data from host

VESPER (Virtual Embraced Space ProbER);
The framework to handle Kprobes on the virtual environment.
2. Concept & Structure
Concept of VESPER

VESPER is a framework to probe VMs

- **Controllability of probing modules**
  - **Load** probing modules from host to guest

- **Accessibility to the probed data**
  - **Transfer** probed data from guest to host via Virtual FS (relayfs-like)
Design Decision

• Split driver model
  – Comm way of implementing drivers in virtualization technology
  – Backend/frontend driver

• Don't use guest user space when loading modules
  – Insert probes even though user space is corrupt

• SystemTap *) http://sourceware.org/systemtap
  – Easy to build probing modules
Basic Structure

Host

User Space

Cluster Manager

SystemTap

VESPER-UI

VMM

Backend

Probe Listener

Probe Loader

Guest

User Space

Kernel Space

Probing Module

Probe Loader

Probe Listener

Load Probing Module

Share Recorded Data
3.

Demo
SystemTap

- Introduce VESPER options
  - `-D VSP_MODULE -D VSP_EM_MOD`
- Demo
  - Probing “send_signal” for monitoring httpd service

(1) Building Module

> stap -vvv -p4 -k -m probe_httpd -D VSP_MODULE -D \ VSP_EM_MOD httpd.stp

(2) Loading the module to guest

> ./vsp_to guest_id insmod probe_httpd.ko

(3) Start/stop httpd service

> /sbin/service httpd start/stop

(4) Check msg on relayfs with `/sys/kernel/debug/vesper/...`

> cat /sys/kernel/debug/vesper/...
4.

Evaluation
Testbed – Web Service

Cluster system to evaluate

Active
(Physical Server #1)

Host

User Space

Heartbeat

vesper

Kernel Space

Xen

Guest1

service

Guest2

service

probe

Not Used

Active
(Physical Server #1)

Guest1

service

Guest2

service

probe

Not Used

Standby
(Physical server #2)

Guest1

service

Guest2

service

probe

Not Used

Host

User Space

Heartbeat

vesper

Kernel Space

Xen

Copyright (c) 2009 Hitachi LTD., Systems Development Lab. All right
Use Case 1

Something wrong happened in Httpd, then SIGTERM

- **Guest1**: Heartbeat monitors service with 10sec period
- **Guest2**: VEPSER probes "send_signal"
  - Monitor signal sending
  - Record signal number
  - Record sender process
Fail-over Latency Improvement I

Time delay of failure notification

Average on 100 tries

- Response latency improved 50%
- VESPER can find out who sent signal and which signal sent
Use Case II

Guests used up the memory

• Guest1: Heartbeat monitors service with 10sec period
• Guest2: VEPSER probes "out_of_mem"
  – Monitor calling OOM-killer
Time delay of failure notification

In different situations, Heartbeat might find out failure of service. However, VESPER can surely show the better performance in any case.
5. Conclusion
Conclusion

• VESPER is the framework to gather guest information effectively in virtualized env.
  – Use probing modules (SystemTap) to gather guest information
  – Load probing modules from host to guest
  – Share the record buffer between host and guest
  – Provide evaluation criteria to cluster manager
  – Improve fail over latency
  – Improve failure analysis
Legal Statement

- Linux is a registered trademark of Linus Torvalds
- Other company, product, and service names may be trademarks or service marks of others
Questions?
Thank you!

For more information:
http://vesper.sourceforge.net/