Three-hot Technologies and Their Usages at Huawei’s Public Cloud

Liu Jinsong, Huang Zhichao
Huawei
Agenda

• Online update requirements @ cloud

• Huawei’s 3-hot technologies
  – Hot patch
  – Hot replacement
  – Hot migration (live migration 😊)

• 3-hot usages @ Huawei Cloud
Cloud is complicated, need fix/update frequently
  - Bugs & security holes
    - Hundreds of CVE reports per year
    - High risk security holes
      - XSA-108
      - Intel security hole: spectre, meltdown, and … (it’s just 1 hole but …)
  - Components upgrade
    - Openstack components: nova, neutron, etc.
    - VM related components: libvirt, qemu, ovs, vims, etc.
    - Fast upgrade support newly-add features, say, once per month
  - Hostos upgrade
    - New CPU/Chipset support, i.e., Skylake adds ~40 hardware features
    - New kernel support, w/ better performance and newly-add features
  - CPU microcode upgrade, hardware broken
    - Microcode for Intel security hole
    - Memory error: UCNA, SRAO, SRAR
    - Other unbelievable hardware broken: i.e., CPU crazy fans 😞
Online update requirements @ cloud

• We have to fix/upgrade the SPEED car !!!
## Huawei’s 3-hot technologies

<table>
<thead>
<tr>
<th></th>
<th>Advantages</th>
<th>Disadvantages</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>hot patch</strong></td>
<td>• Bugfix and security holes</td>
<td>• Usually for small but critical fix</td>
</tr>
<tr>
<td></td>
<td>• Light-weight operation</td>
<td>• Do not support newly-add functions/features</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Some bugs/security holes are hard to fix via hot patch</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Troublesome for SRE to manage and verify patch branches</td>
</tr>
<tr>
<td><strong>hot replacement</strong></td>
<td>• Component replaced entirely</td>
<td>• Not good at kernel fix/update</td>
</tr>
<tr>
<td></td>
<td>• Support newly-add features</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Medium-weight operation</td>
<td></td>
</tr>
<tr>
<td><strong>hot migration</strong></td>
<td>• Kernel upgrade</td>
<td>• Cannot migrate vm w/ sr-iov</td>
</tr>
<tr>
<td>(= live migration in Chinese 😊)</td>
<td>• Not only for upgrade</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Solve problems what hot patch or hot replacement cannot handle</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Hot patch

- **Hot-patch for Xen**
  - xSplice-like solution (thanks Konrad @ Oracle)
  - Trampoline jump at the head of old func
    - Wait for all pCPUs to stop and apply together
    - clean stack ensure not running at any CPU
      - Idle
      - Before vmentry
    - cpuid serializing
  - Enhancement
    - Auto build from a patch and auto test
    - A framework to hot-patch a POD
      - Retry, revert, and reboot handler
    - Support hot-patching assembly code

- **Hot-patch for KVM & Linux**
  - livepatch combine consistency model of kGraft + kPatch
  - [https://www.slideshare.net/GlobalLogicUkraine/linux-kernel-live-patching](https://www.slideshare.net/GlobalLogicUkraine/linux-kernel-live-patching)

- **Hot-patch for usrspace processes**
  - Huawei’s Dopra, a framework
  - Patching qemu, ovs, vims, …
Hot patch use case @ Huawei cloud

- **Fix CVE-2017-5715 (Intel Spectre) at Xen hypervisor**
  - xSplice fix C function but cannot fix assembly code
  - xpatch/tools/create-diff-object.c
    - Define and handle special symbol (w/ prefix '_fix_')
    - Find correct **assembly address** to replace
  - Fix `vmx_asm_vmexit_handler`
    - `arch/x86/hvm/vmx/entry.S`
    - `arch/x86/hvm/vmx/entry.S`
    - `@@ -116,6 +116,81 @@ vmx_asm_vmexit_handler:`
      - `ALIGN`
      - `.globl _fix_vmx_asm_vmexit_handler`
      - `_fix_vmx_asm_vmexit_handler:` // special symbol w/ prefix '_fix_'
        - `push %rdi`
        - `push %rsi`
        ...
        - `push %r15`
      + `xor %edi,%edi` // fix assembly
      + `xor %esi,%esi`
      + ...
      + `xor %r15,%r15`
      `get_current(bx)`
      ...

Advantages and disadvantages of hot patch

- Hot patch
  - Light-weight operation for cloud SRE
  - But troublesome for SRE to manage baseline branches
  - Some fix are hard to be hot-patched
    - data structure (shadow variable after kernel 4.15)
    - .rodata
    - cannot change function api and semantic
    - unsafe to fix ftrace handler w/ infinite loop risk
    - unsafe to fix NMI handler
    - booting stage bugfix
    - inline function
    - should be very careful about deadlock
    - do not support newly-add functions
    - ......
Hot replacement

- **Components entirely upgrade**
  - Reboot-able components: VM runtime-unrelated
    - nova, neutron, libvirt, etc.
  - Non reboot-able components: VM runtime-related
    - compute (qemu), storage (vims), network (ovs), etc.
Hot replacement framework

- Unified replacement framework for OVS (network) and VIMS (storage)
  - Preload and lazy-offload, fast switching (less than 100ms)
  - State vs. stateless design
  - Add component agent connecting qemu (if possible) so that no disconnect and no re-connect

- Qemu is another story
Hot replacement - qemu

- **Qemu hot replacement**
  - **Way 1:** migrate vm locally
    - may fail since insufficient memory
    - may fail for VM under high dirty page speed
  - **Way 2:** share page
    - Zero copy
    - Performance impact by transparent huge pages
  - **Way 3:** share page table, cover old qemu VMAs except that of VM
    - Zero copy
    - keep pid unchange
    - Much bigger switch downtime, kill old qemu then covered by new qemu VMAs
    - Cannot revert if new qemu fail
  - **Way 4:** share page table, but exec new qemu process
    - Zero copy
    - Preload new qemu sharing VM PUD with old qemu
    - Pause old qemu and unpause new qemu
    - Lazy-offload old qemu if new qemu success, or, revert old qemu if new qemu fail
    - Different pid but acceptable
Hot migration -- challenges

• Live migration @ virtualization
  – Xen live migration
    • PV is unfriendly to live migration
      – Buggy PV disconnect and re-connect
      – Ecosystem issue, work around by guest whitelist but >15% guest cannot migrate
    • Support migration among different CPUs via emulated tsc but w/ performance issue

  – KVM live migration
    • Not support migration among different CPUs because of native tsc (until Skylake tsc scaling)

  – SR-IOV migration
    – Giant VM migration under huge memory dirty ratio
Hot migration -- challenges

- **Live migration @ cloud**
  - Cloud environment challenges
    - Cloud environment is very complicated and unfriendly to live migration
      - Different software version and configuration
      - Different hardware types: CPU, MSRs
      - Even buggy network switch may result in migration error !!
    - Different storage/network types
  - Performance challenges
    - Network breaktime, growing w/ VPC scale (10S->10 minutes)
    - Communication among cloud components
      - Nova, neutron, libvirt, etc.
  - Reliability challenges
    - Migrating VM may dead or brain-split
    - Ensure vm 100% survive when migrate fail
  - Large scale parallel migration challenges
    - Server congestion, network congestion, etc.
    - Gratuitous ARP may not accepted by parallel migrating vms
    - Malfunction server isolation
  - Blablabla ……
Hot migration design @ Huawei cloud

- **De-couple**
  - Event mechanism and publisher-subscriber model
  - Support different storage/network types

- **Reliability**
  - Shakehands and roll-back when anything wrong (vm will survive)
  - How about shakehands broken (say, network issue)?
    - **image lock**: who get the image lock will survive (vm will not brain-split)

- **Performance**
  - Fast event channel for performance-critical ops
  - Network trampoline when VPC path not ready

- **Giant vm migration**
  - Support any giant vm migration under any dirty page ratio
    - If only transfer ratio > dirty page ratio
Hot migration result @ Huawei cloud

- **Live migration for OS upgrade at all Huawei cloud sites**
  - **Reliability**
    - 99.99% migration success
    - 100% vm survive when migration fail for whatever reason
  
  - **Performance**
    - CPU downtime: ~25ms
    - VPC network breaktime:
      - 82% breaktime < 50ms
      - 99% breaktime < 200ms
      - 100% breaktime < 500ms
  
  - **Degree of parallelism**
    - Upgrade > 2000 servers per night
    - Technically support much higher parallelism but no enough free servers
  
  - Support all giant vm live migration
Hot migration use case @ Huawei cloud

- MCE/Disk error/Filesyststem readonly …..
  - ~1% server crash per day, while ~48% hardware issue
- Dynamic resource scheduling
- Distributed power management
- Fix CVE-2017-5715 (Intel Spectre) at KVM
  - Better performance than upstream: 30% -> 10%-
  - Retpoline optimization: remove unnecessary retpoline (no vcpus)
  - IBPB/IBRS optimization: remove unnecessary IBPB/IBRS (novcpus, A->Idle->A)
  - Microcode update, so that guest upgrade by itself
“Quote Placeholder”
LINUX CON
containercon
CLOUD OPEN
CHINA

THINK OPEN
开放性思维