Introduce And Status Update About COLO FT

Xie Changlong<xiecl.fnst@cn.fujitsu.com>

Zhang Hailiang <zhang.zhanghailiang@huawei.com>

HUAWEI FUJITSU

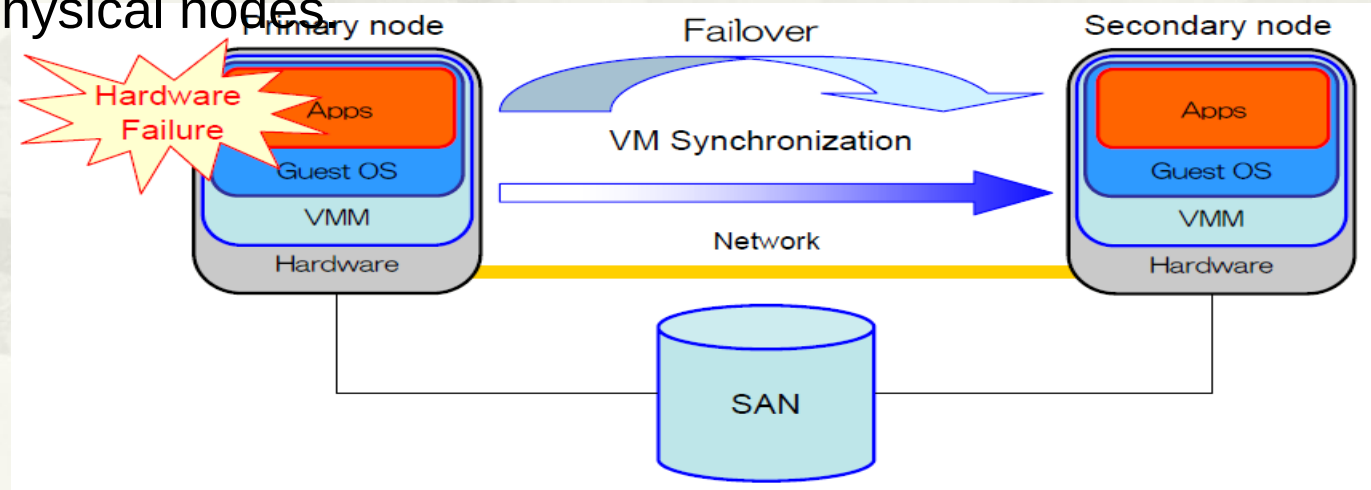
Agenda

- Introduce COarse-grain LOck-stepping
- COLO Design and Technology Details
- Current Status Of COLO
- Future Work About COLO

Non-Stop Service with VM Replication

Virtual Machine (VM) replication

- A software solution for business continuity and disaster recovery through application-agnostic hardware fault tolerance by replicating the state of primary VM (PVM) to secondary VM (SVM) on different physical nodes.



Existing VM Replication Approaches

- **Replication Per Instruction: Lock-stepping**
  - Execute in parallel for deterministic instructions
  - Lock and step for un-deterministic instructions
- **Replication Per Epoch: Continuous Checkpoint**
  - Secondary VM is synchronized with Primary VM per epoch
  - Output is buffered within an epoch

Problems

- **Lock-stepping**
  - Excessive replication overhead
    - ✂memory access in an MP-guest is un-deterministic
- **Continuous Checkpoint**
  - Excessive VM checkpoint overhead
  - Extra network latency

What Is COLO ?

- **VM and Clients model**
  - VM and Clients are a system of networked request-response system
  - Clients only care about the response from the VM
- **COarse-grain LOck-stepping VMs for Non-stop Service (COLO)**
  - PVM and SVM execute in parallel
  - Compare the output packets from PVM and SVM
  - Synchronize SVM state with PVM when their responses (network packets) are not identical

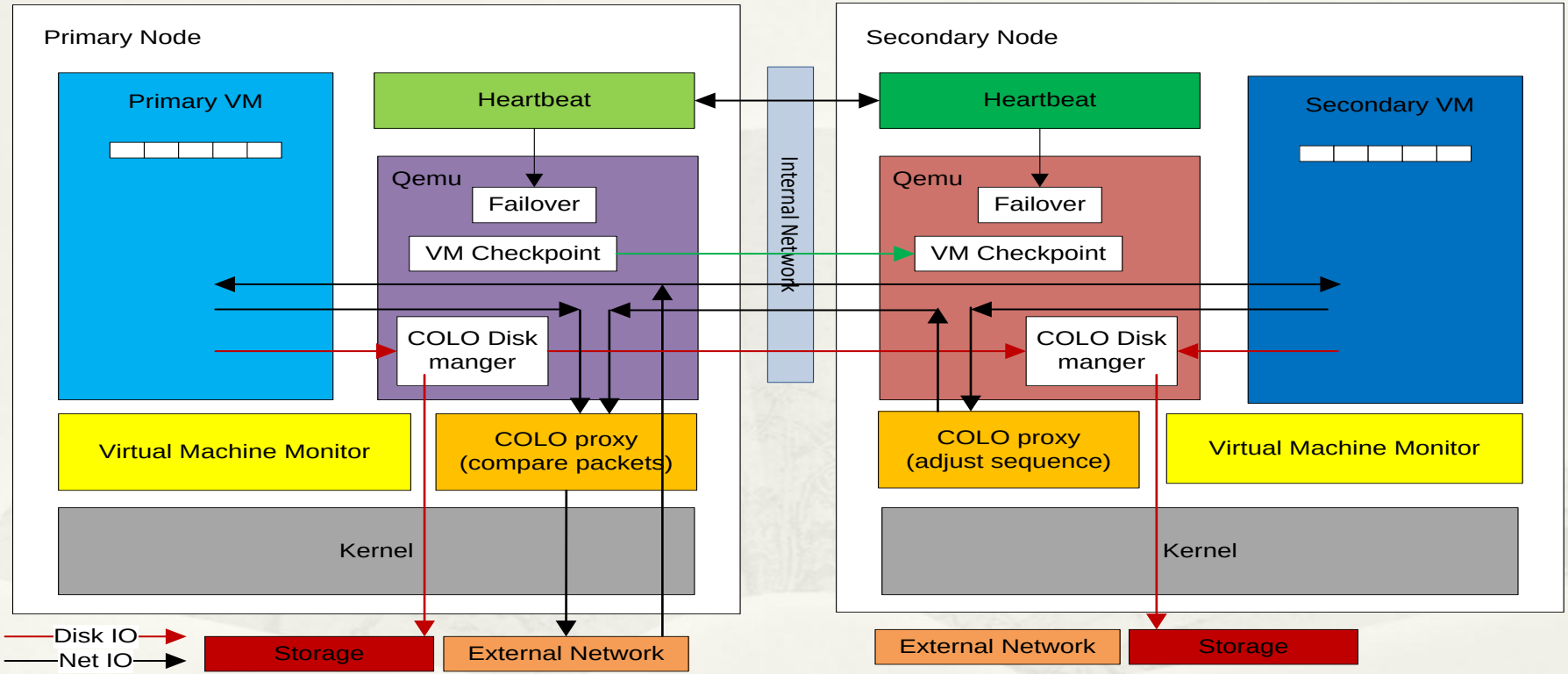
Why COLO Better

- **Comparing with Continuous VM checkpoint**
  - No buffering-introduced latency
  - Less checkpoint frequency
    - On demand vs periodic
- **Comparing with lock-stepping**
  - Eliminate excessive overhead of un-deterministic instruction execution due to MP-guest memory access

Agenda

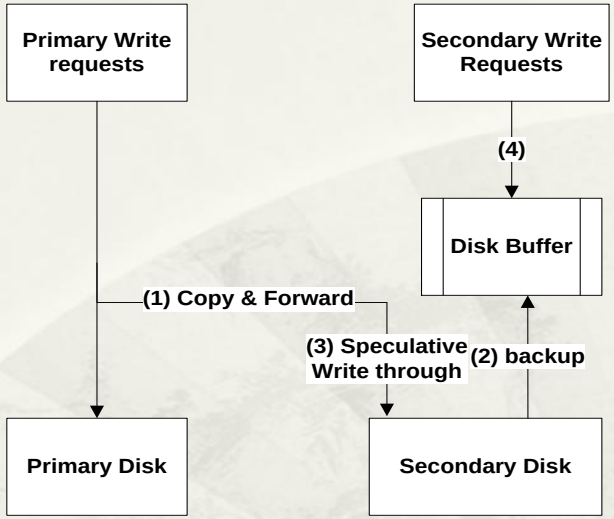
- Introduce COarse-grain LOck-stepping
- COLO Design and Technology Details
- Current Status Of COLO
- Future Work About COLO

Architecture Of COLO

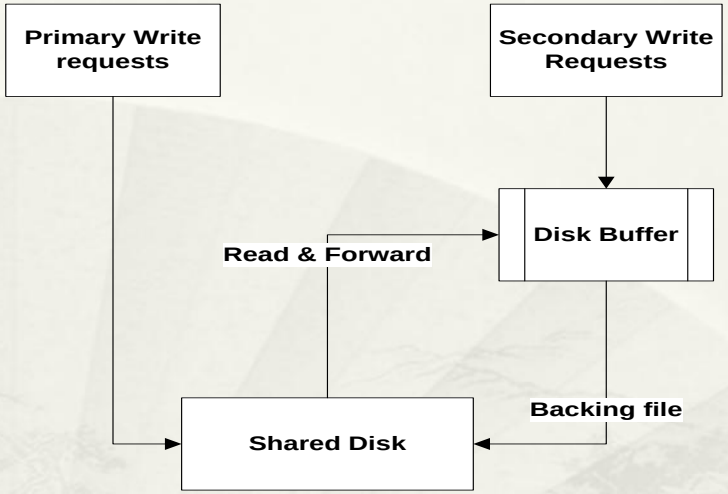


COarse-grain LOck-stepping Virtual Machine for Non-stop Service

How Block Replication Work



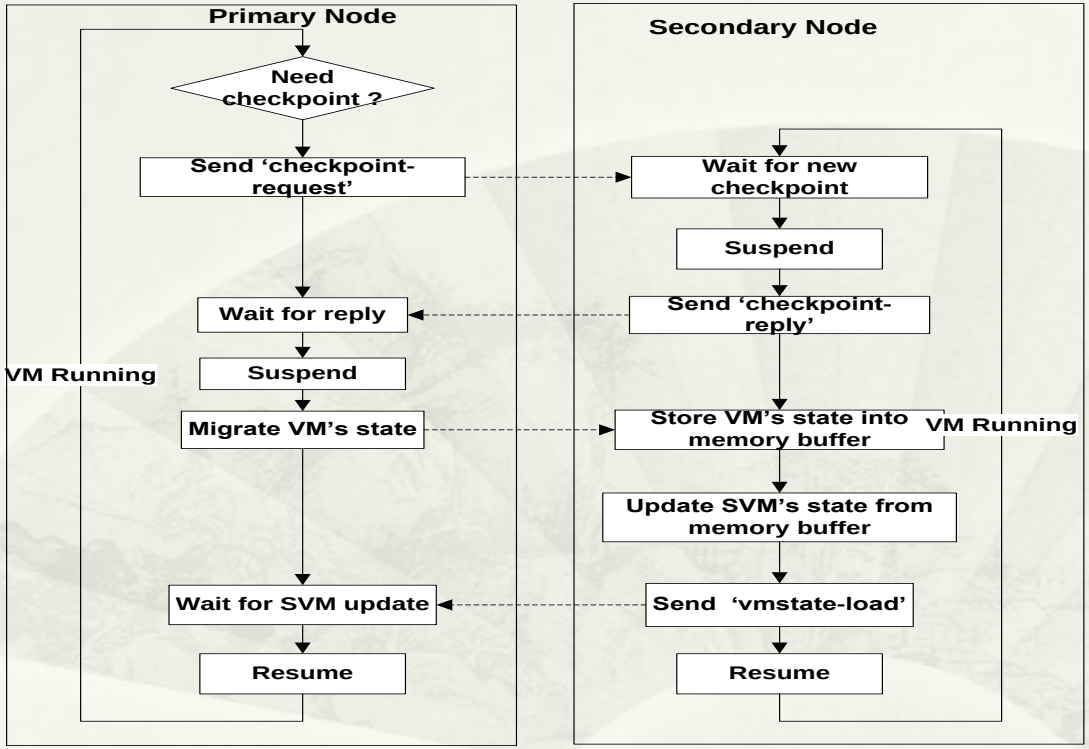
Non-Shared disk workflow



~~Shared disk workflow~~

Checkpoint: Disk buffer will be emptied to achieve block replication
Failover: Disk buffer will be written back to the ‘parent’ disk

VM State Checkpointing



Execution and Checkpoint Flow in COLO

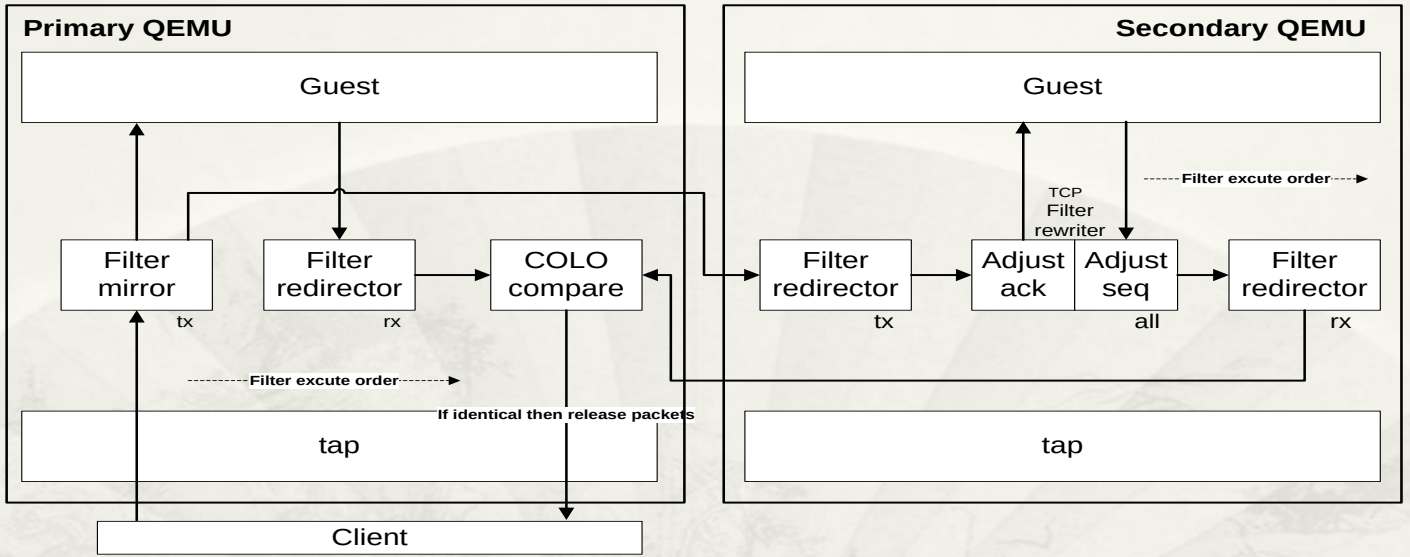
- Based on live migration
- PVM's memory/device data be stored in extra memory-buffer of SVM before be synchronized to SVM

COLO Proxy Design

Scheme:

- ~~Kernel scheme~~: (obsolete)
  - Based on kernel TCP/IP stack and netfilter component
  - Can support vhost-net, virtio, e1000, rtl8139, etc
  - Better performance but less flexible (Need modify netfilter/iptables and kernel)
- Userspace scheme:
  - Totally realized in QEMU
  - Based on QEMU's netfilter components and SLIRP component
  - Not support vhost-net, but e1000, rtl8139
  - More flexible

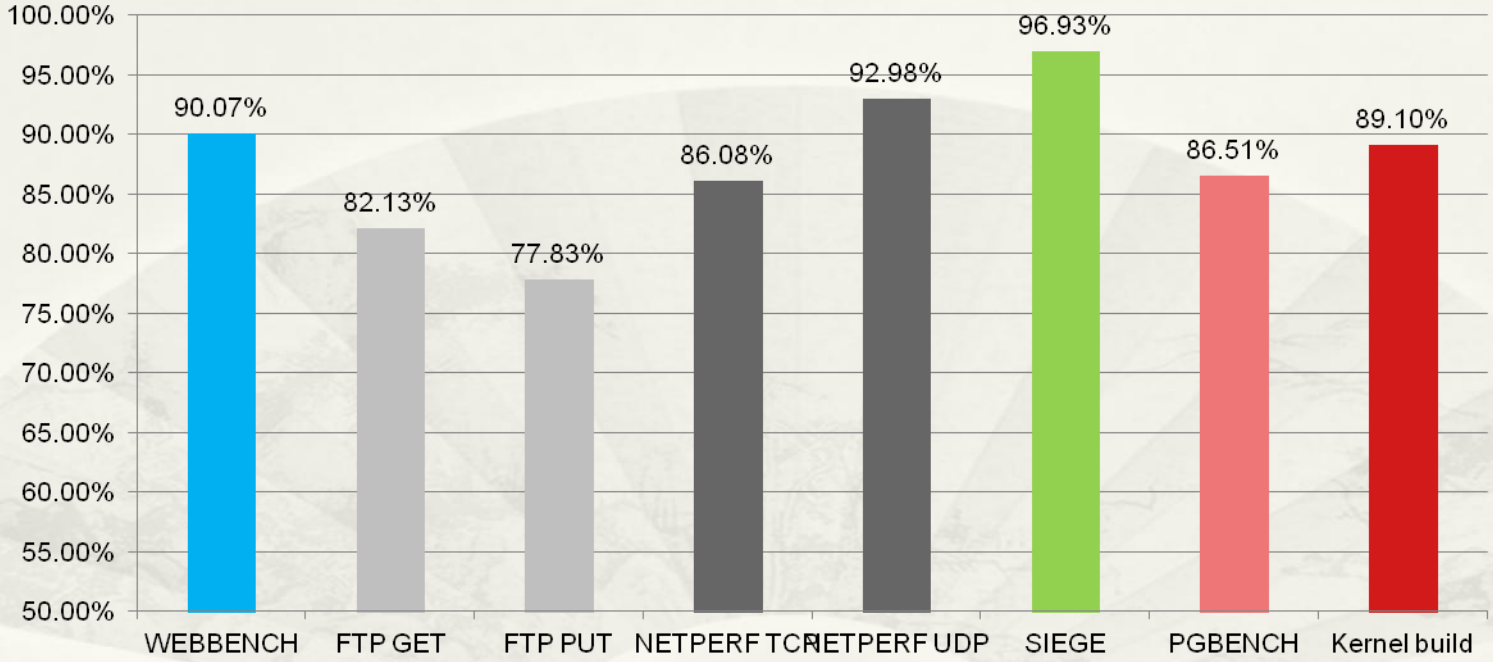
Proxy Design (Userspace scheme)



- **Filter mirror:** copy and forward client's packets to SVM
- **Filter redirector:** redirect net packets
- **COLO compare:** compare PVM's and SVM's net packets;
- **Filter rewriter:** adjust tcp packets' ack and tcp packets' seq

COLO Performance In KVM

* Performance (Based on kernel proxy)



The experimental data is normalized to the native system

Agenda

- Introduce COarse-grain LOck-stepping
- COLO Design and Technology Details
- Current Status Of COLO
- Future Work About COLO

Status of COLO In KVM

COLO Framework:

- Include VM state checkpoint process, failover process
- Patch v21 had been post, under review

COLO block replication:

- Only include non-shared storage replication scheme
- Already been merged to master branch

COLO proxy:

- Include netfilter base/buffer/mirror/packets compare plugins
- Already been merged to master branch

Status of COLO In Xen

COLO Framework:

- Already been merged to master branch

COLO block replication:

- Only include the old non-shared storage replication scheme
- Need to be sync with the last qemu branch

COLO proxy:

- Abandoned implementation scheme based on kernel proxy
- Need to be sync with the last qemu branch

Agenda

- Introduce COarse-grain LOck-stepping
- COLO Design and Technology Details
- Current Status Of COLO
- Further Work About COLO

Future Work

- Revise patches according review feedbacks, get patches accepted into upstream
- Continuous VM replication development
- Support shared storage
- Optimizations
- Libvirt support

Thank You