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

Eddie Dong, Tao Hong, Xiaowei Yang

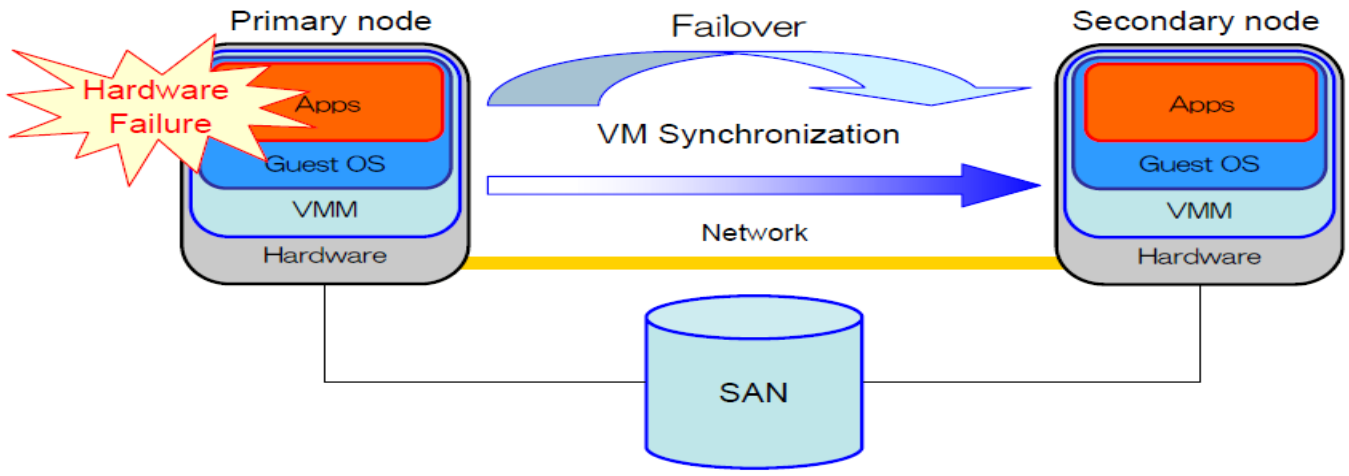
Legal Disclaimer

Agenda

- VM Replication & COLO
- COLO_KVM
- Performance Prediction
- Summary

Non-Stop Service with VM Replication

- Typical Non-stop Service Requires
  - Expensive hardware for redundancy
  - Extensive software customization
- VM Replication: Cheap Application-agnostic Solution



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
  - Extra network latency
  - Excessive VM checkpoint overhead

Why COarse-grain LOck-stepping (COLO)

- VM replication is an overly strong condition
  - Why do we care about the VM state ?
    - The client cares about response only
  - Can the control failover *without* "precise VM state replication"?
- Coarse-grain lock-stepping VMs
  - Secondary VM is a replica, if it has generated same response as the primary so far
    - If true, failover with no service stop

Non-stop service focus on server response, not internal machine state!

How COLO Works

- Response Model for C/S System

$$R_n = g_n(r_0, r_1, r_2, \ldots, r_n, u_0, \ldots, u_m)$$

  - r_i & u_i are the request and the execution result of an non-deterministic instruction
  - Each response packet from the equation is a semantics response
- Successfully failover at *kth* packet if

$$C = \{R_1^P, \ldots, R_k^P, R_{k+1}^S, \ldots\} \qquad \forall i \leq k, R_i^S = R_i^P$$

(C is the packet series the client received)

Why is 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

Status

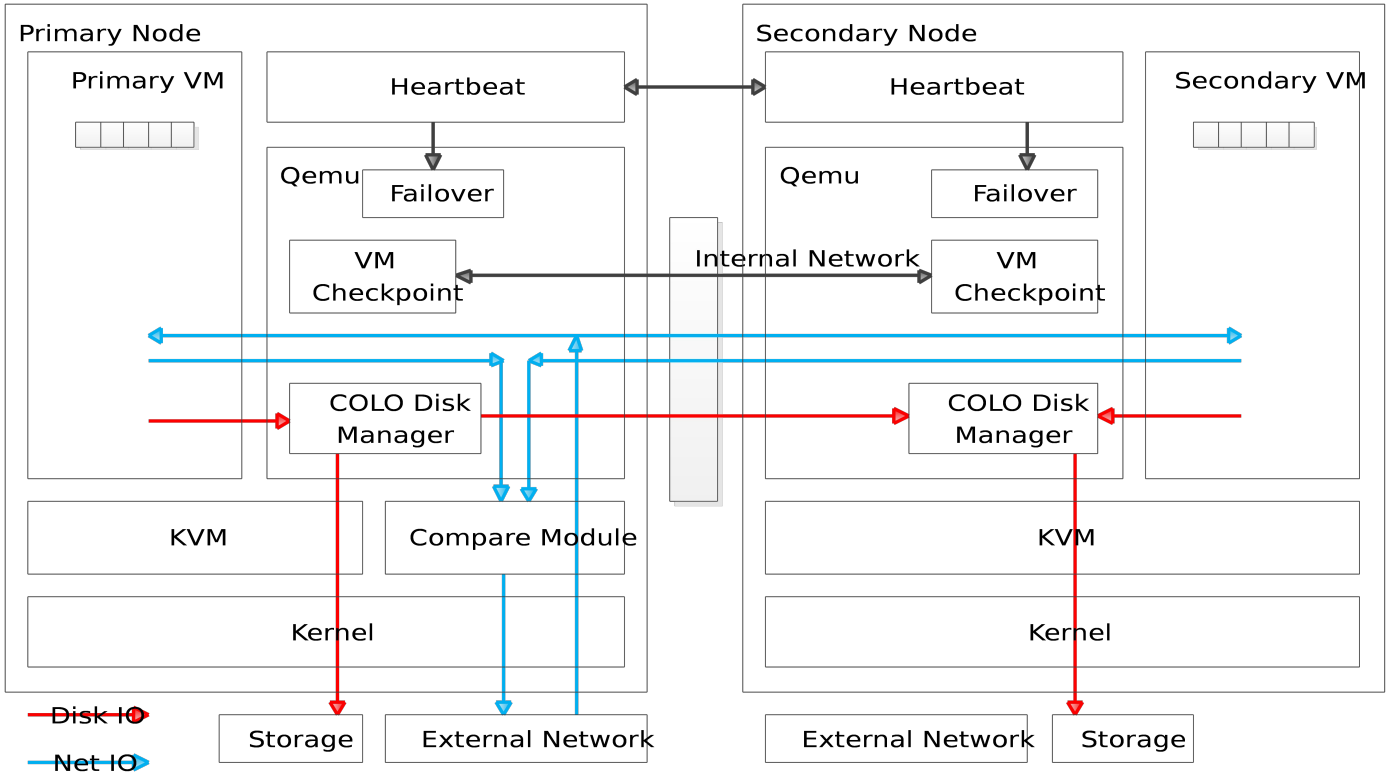
- Academia paper published at ACM Symposium on Cloud Computing (SOCC'13)
  - "COLO: COarse-grained LOck-stepping Virtual Machines for Non-stop Service"
    - http://www.socc2013.org/home/program
  - Refer to the paper for technical details

- Industry announcement
  - Huawei FusionSphere uses COLO
    - http://enterprise.huawei.com/ilink/enenterprise/about/news/news-list/HW_308817?KeyTemps=

Agenda

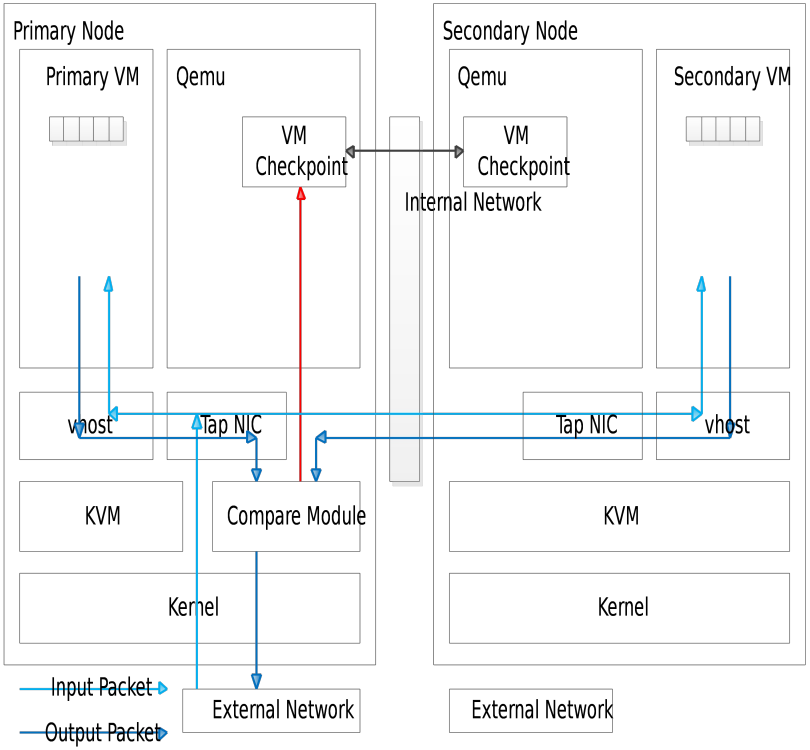
- VM Replication & COLO
- COLO_KVM
- Performance Prediction
- Summary

Architecture of COLO



Pnode: primary node; PVM: primary VM; Snode: secondary node; SVM: secondary VM

Network Process



No need to change exiting vNIC modules – use *tc* for packet redirect / mirror

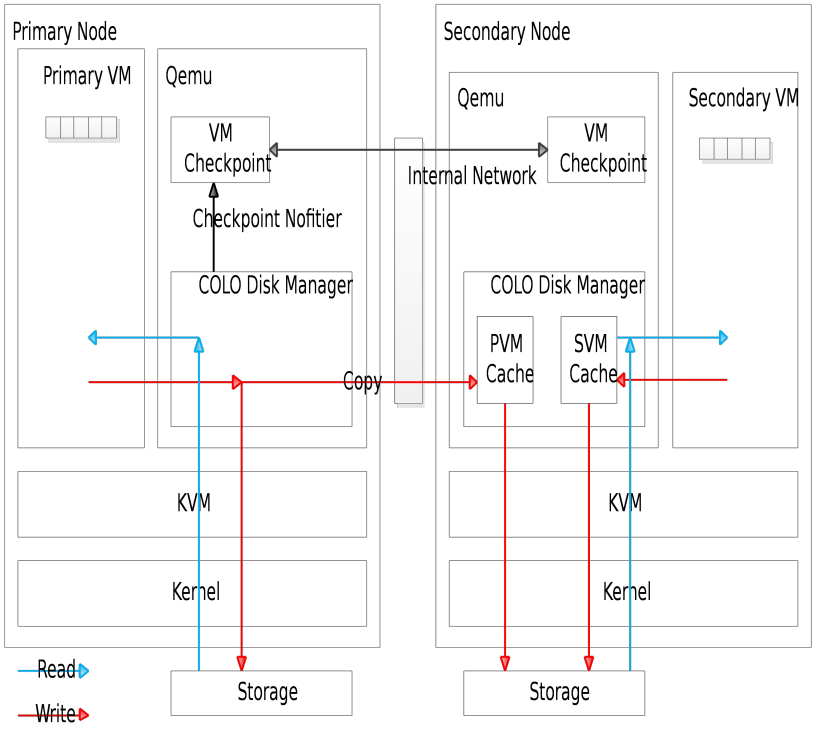
RX

- **Pnode**
  - Receive a packet from client
  - Mirror the packet and send to Snode
  - Send the packet to Tap NIC
- **Snode**
  - Receive the packet from Pnode
  - Redirect the packet to Tap NIC

TX

- **Snode**
  - Redirect the SVM packet to Pnode
- **Pnode**
  - Redirect the PVM packet to ifb0
  - Redirect the SVM packet to ifb1
  - CM compares PVM/SVM packet in ifb
    - Same: send the packet to client
    - Different: trigger checkpoint

Storage Process



Need modify Qemu vDisk IO path – intercepted by Colo Disk Manager (DM)

Write

- **Pnode**
  - DM sends the Write request (offset, len, data) to PVM cache in Snode
  - DM calls block driver to write to storage
- **Snode**
  - DM saves Write request in SVM cache

Read

- **Snode**
  - From SVM cache, or storage otherwise
- **Pnode**
  - From storage

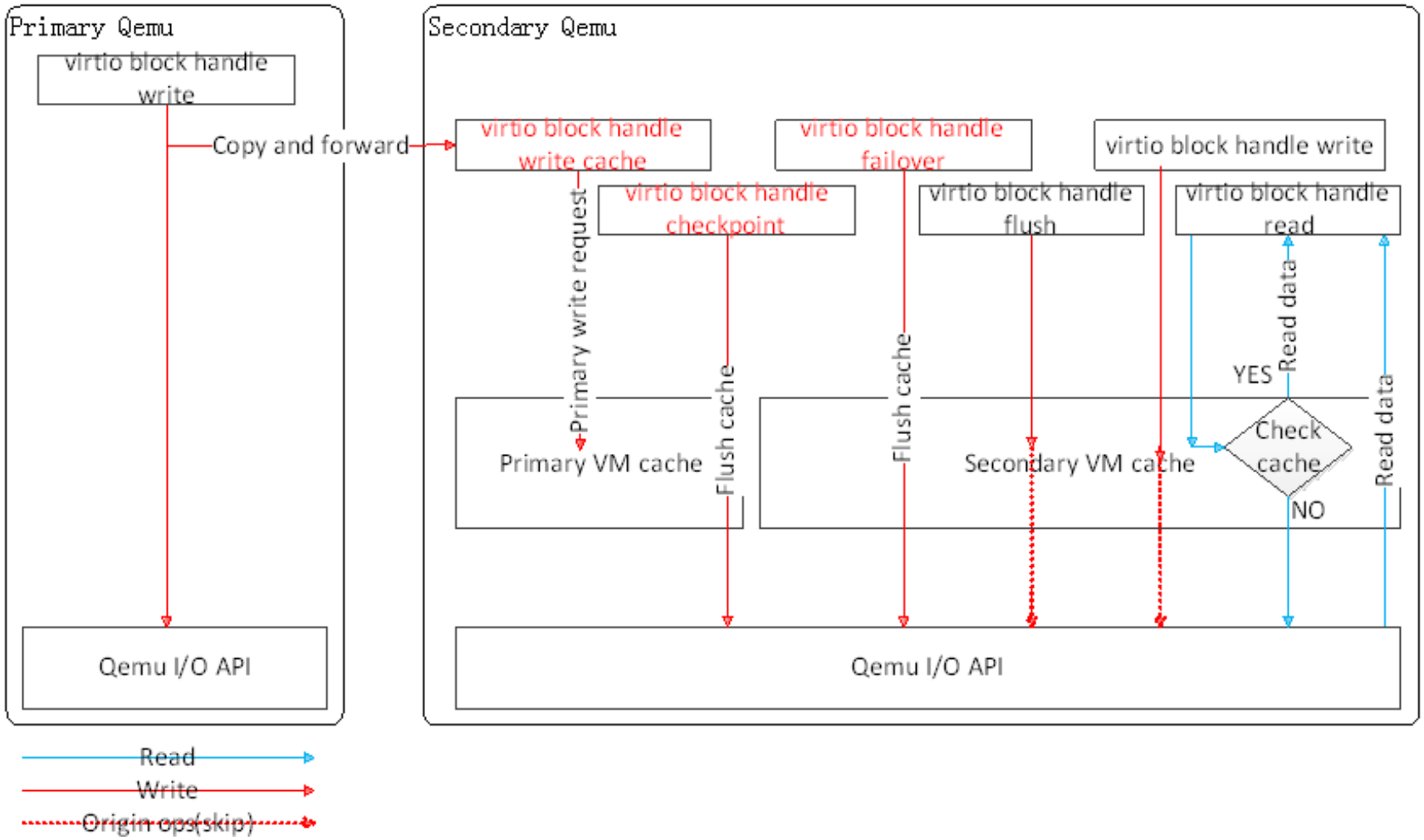
Checkpoint

- DM calls block driver to flush PVM cache

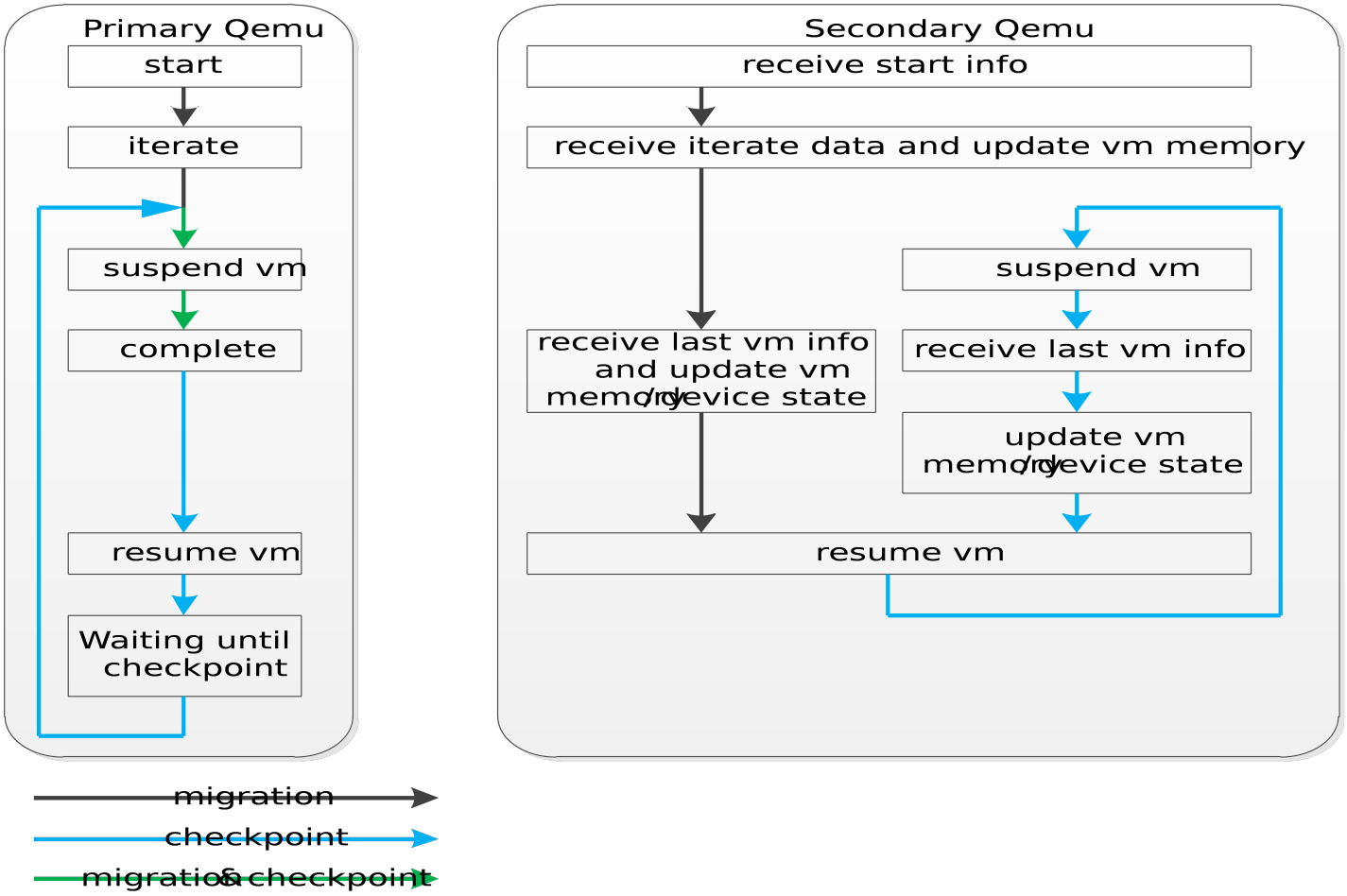
Failover

- DM calls block driver to flush SVM cache

Storage Process (2)

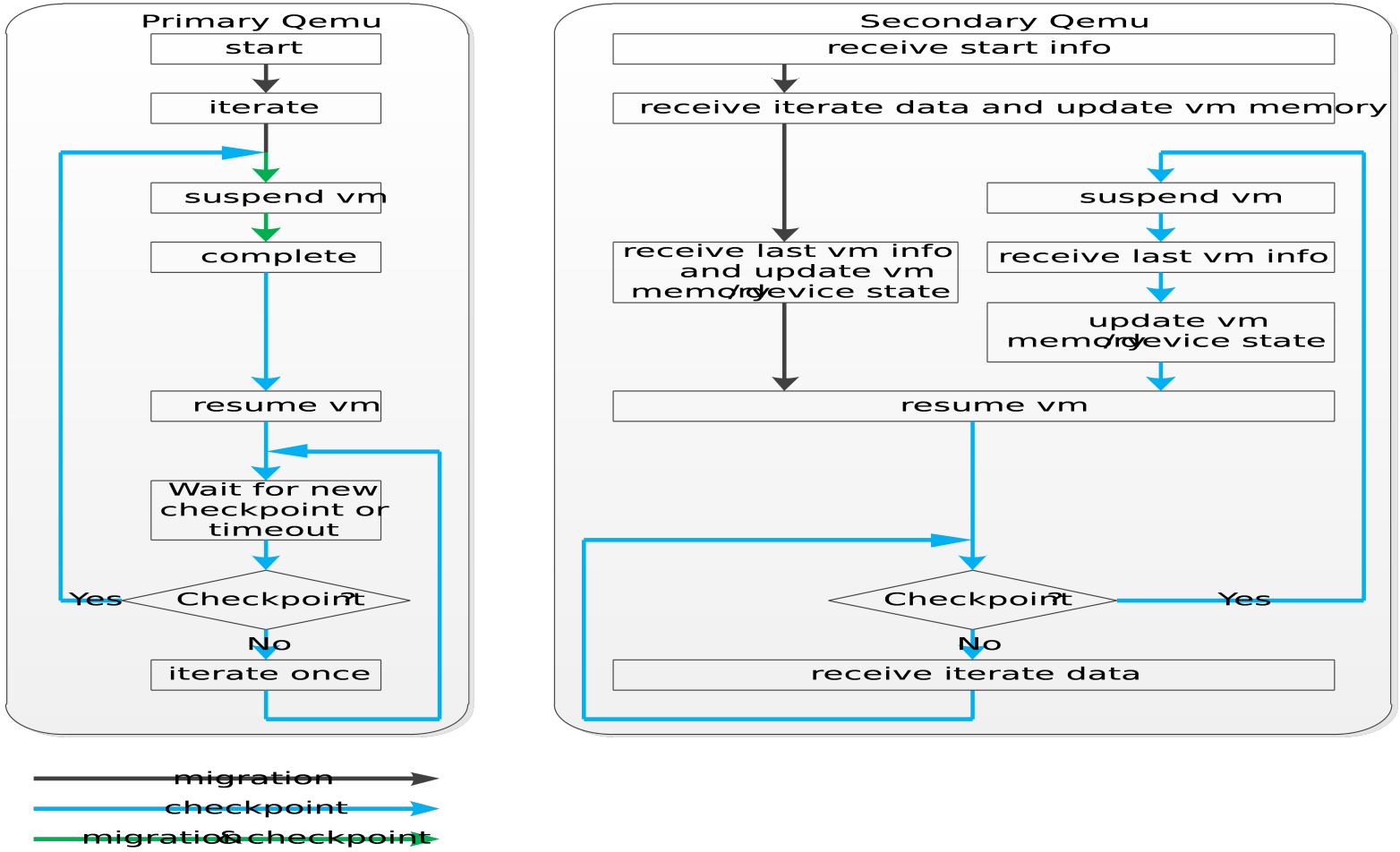


Checkpoint Process



Need modify migration process in Qemu to support checkpoint

Checkpoint + Memory Sync Process

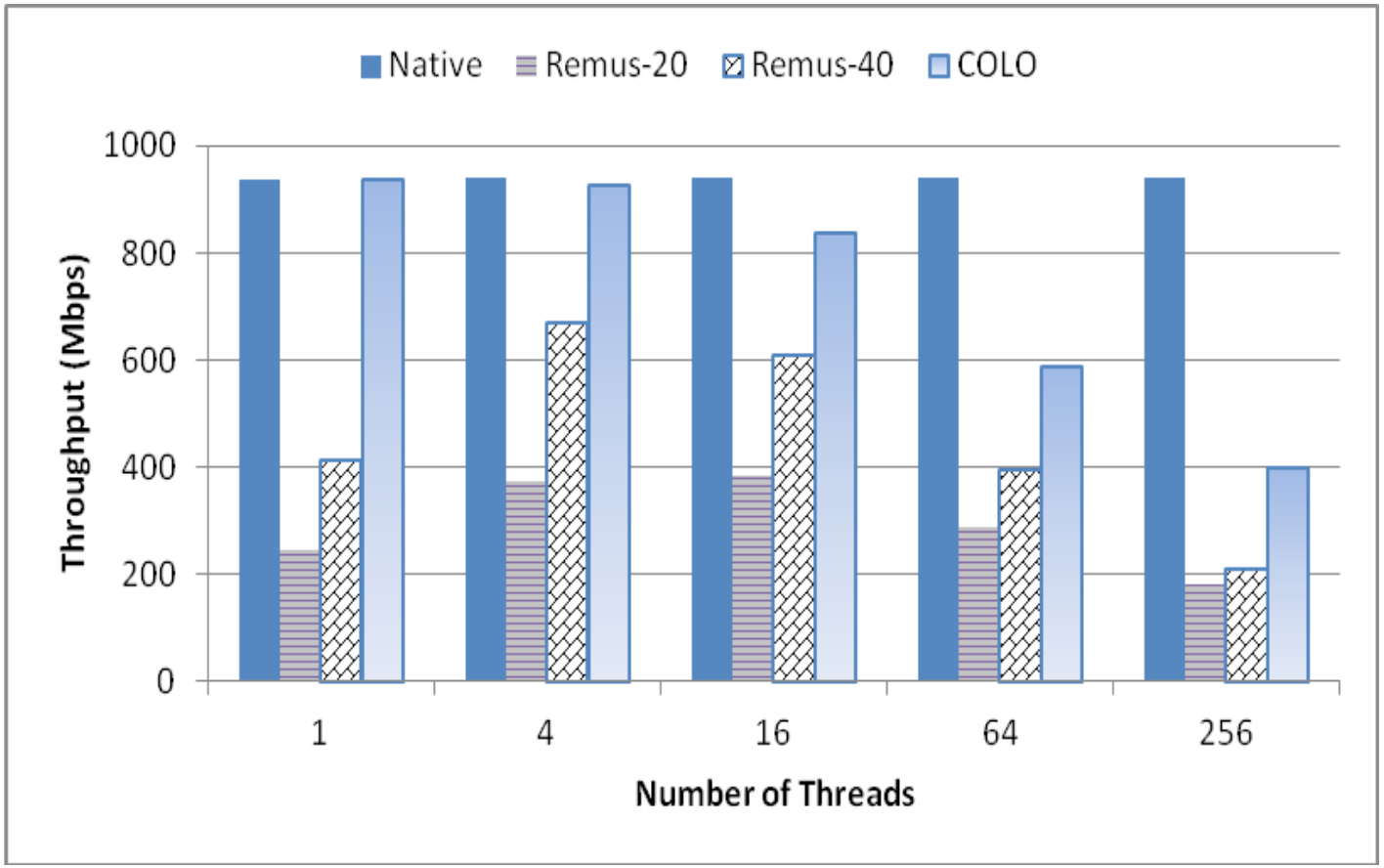


Need modify migration process in Qemu to support checkpoint

Agenda

- VM Replication & COLO
- COLO_KVM
- Performance Prediction
- Summary

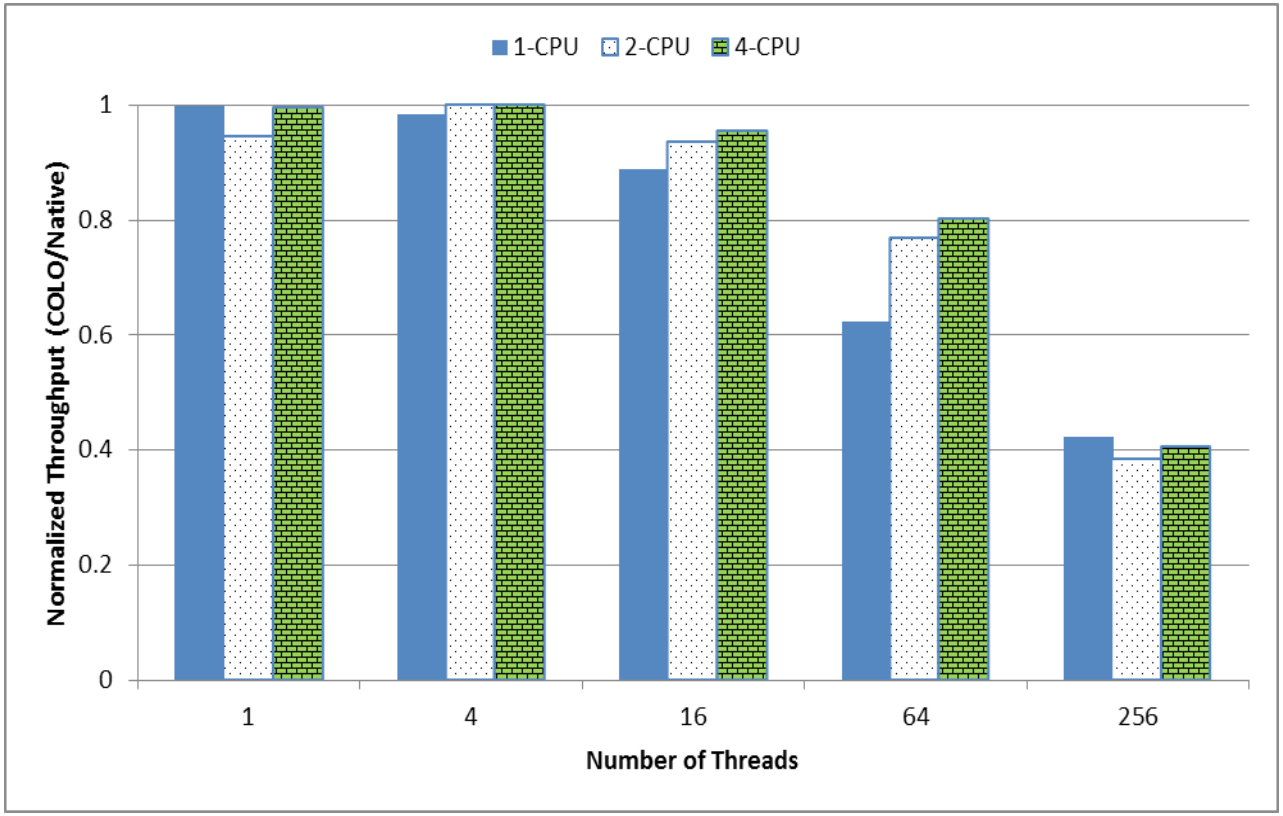
Web Server Performance - Web Bench



Predication base on data in SOCC'13 paper

Source: Intel

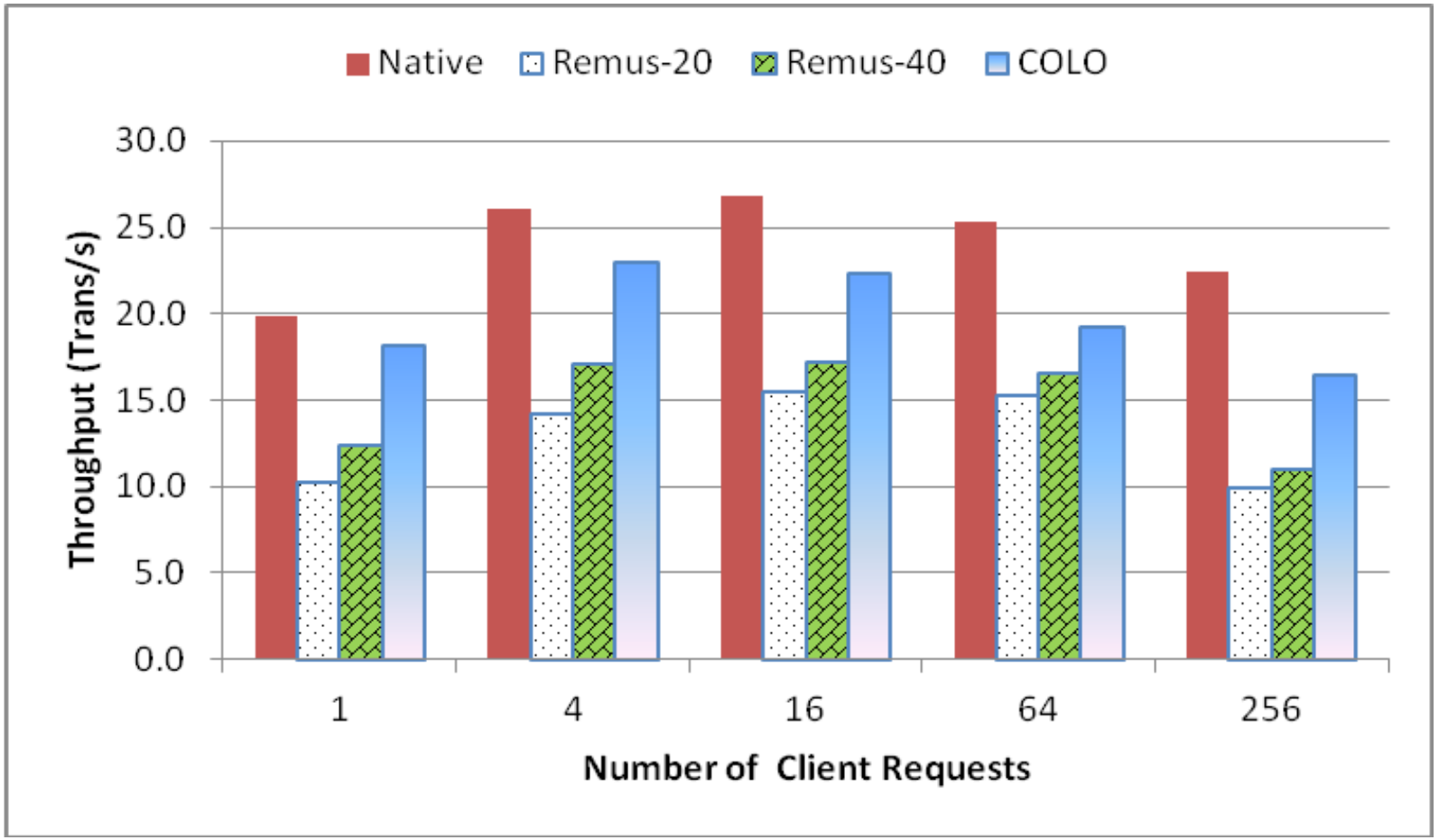
Web Server Performance - Web Bench (MP)



Predication base on data in SOCC'13 paper

Source: Intel

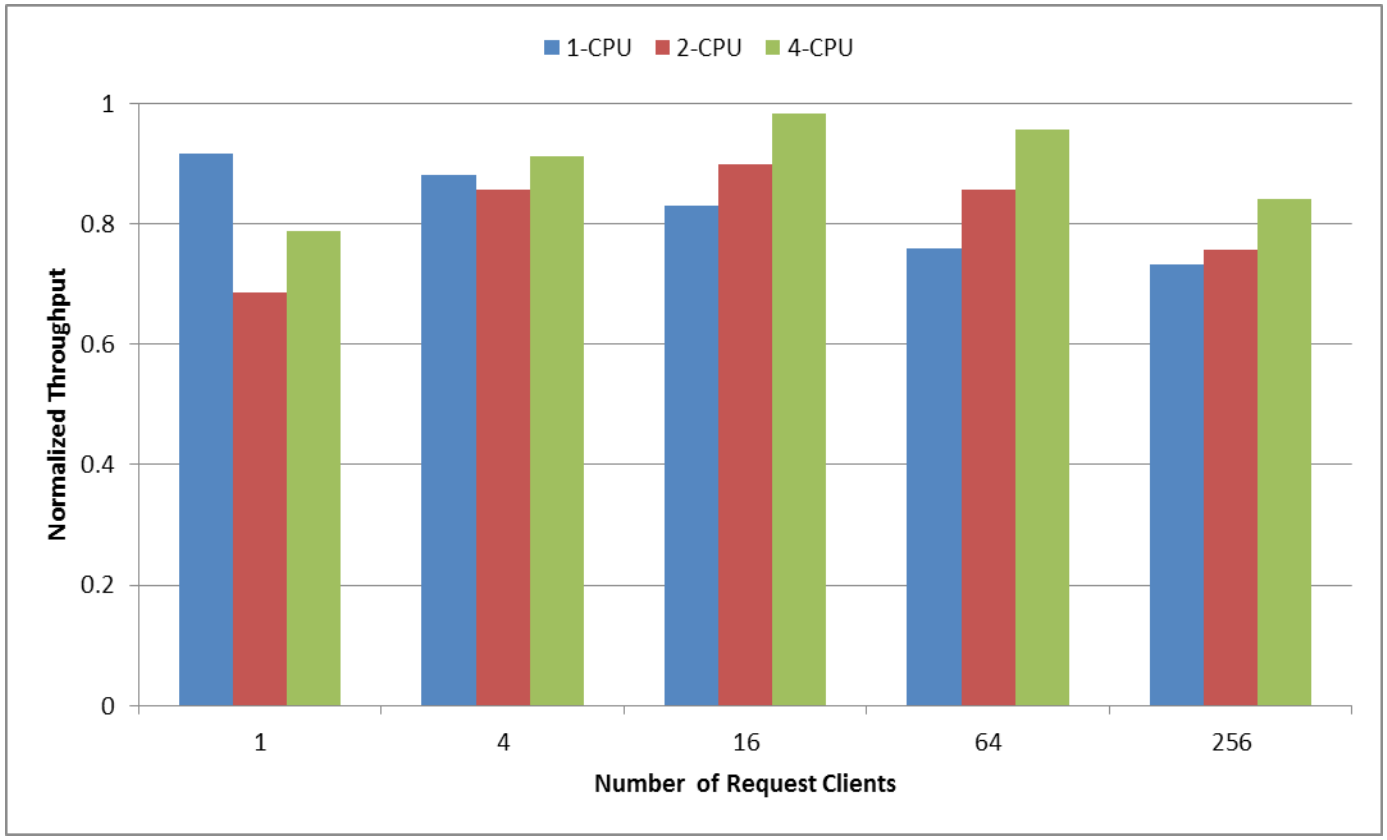
PostgreSQL Performance - Pgbench



Predication base on data in SOCC'13 paper

Source: Intel

PostgreSQL Performance - Pgbench (MP)



Predication base on data in SOCC'13 paper

Source: Intel

Agenda

- VM Replication & COLO
- COLO_KVM
- Performance Prediction
- Summary

Summary

- COLO is an ideal Application-agnostic Solution for Non-stop service
  - Web server: 67% of native performance
  - CPU, memory and netperf: near-native performance
- Next steps
  - Redesign based on feedback
  - Develop and send out for review
  - Optimize performance