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

Eddie Dong, Tao Hong, Xiaowei Yang
INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH INTEL® PRODUCTS. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. EXCEPT AS PROVIDED IN INTEL’S TERMS AND CONDITIONS OF SALE FOR SUCH PRODUCTS, INTEL ASSUMES NO LIABILITY WHATSOEVER, AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO SALE AND/OR USE OF INTEL® PRODUCTS INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT. INTEL PRODUCTS ARE NOT INTENDED FOR USE IN MEDICAL, LIFE SAVING, OR LIFE SUSTAINING APPLICATIONS.

Intel may make changes to specifications and product descriptions at any time, without notice.

All products, dates, and figures specified are preliminary based on current expectations, and are subject to change without notice.

Intel, processors, chipsets, and desktop boards may contain design defects or errors known as errata, which may cause the product to deviate from published specifications. Current characterized errata are available on request.

Intel and the Intel logo are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

*Other names and brands may be claimed as the property of others.

Copyright © 2012 Intel Corporation.
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 undeterministic 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 undeterministic

- 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
  \[ R_n = g_n(r_0, r_1, r_2, ..., r_n, u_0, ..., u_m) \]
  - \( r_i \) & \( u_i \) 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
  \[ C = \{ R^P_1, ..., R^P_k, R^s_{k+1}, ... \} \]
  \[ \forall i \leq k, R^s_i = R^P_i \]
  \( C \) 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 undeterministic instruction execution due to MP-guest memory access
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](http://www.socc2013.org/home/program)
  - Refer to the paper for technical details

Industry announcement

- Huawei FusionSphere uses COLO
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

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

No need to change exiting vNIC modules - use `tc` for packet redirect / mirror
Storage Process

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

Need modify Qemu vDisk IO path – intercepted by Colo Disk Manager (DM)
Storage Process (2)
Need modify migration process in Qemu to support checkpoint
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)

Source: Intel

Predication base on data in SOCC’13 paper
PostgreSQL Performance - Pgbench

Source: Intel

Predication base on data in SOCC’13 paper
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