Performance and Scalability of Server Consolidation

August 2010

Andrew Theurer
IBM
Linux Technology Center
Agenda

- How are we measuring server consolidation?
- SPECvirt_sc2010
- How is KVM doing in an enterprise release?
- How is KVM doing in development release?
- What can we do to improve performance?
How are we measuring Server Consolidation?

- Not many benchmarks that model server consolidation
  - VMmark
    - Really designed for ESX
    - Lacking QoS requirements
  - Home grown
    - May not be easily reproduced by someone else
  - SPECvirt
    - Just released
    - Can be a little overwhelming to run (at first)
    - Costs $$$
    - Restrictions on reporting results

- Some things important to server consolidation workload
  - Lots of VMs of different sizes running many server types
  - Monitor response times
  - Variability in each of the VM's workload
  - Decent amount of I/O
  - Reproduce-able
SPECVirt_sc2010 - What IBM uses right now

- Will likely become the industry standard
- Therefore, will likely be how KVM and others are compared (for performance/capacity)

- KVM is the first and only hypervisor used to date for a published result!
  http://www.spec.org/virt_sc2010/results/specvirt_sc2010_perf.html
  KVM Score: 1169 @ 72 VMs
  - on 2 socket, 12 core Intel Westmere @3.33 GHz (IBM x3650M3)
  - RHEL5.5 host and guests
  - Key optimizations: hugepages, SR-IOV, and node binding

- ESX score: I am forbidden to tell anybody....
**SPECvirt_sc2010 What does it do?**

- Run as many VMs until any of the workloads fail any of the Quality of Service requirements
- VMs are added in sets of six, called a Tile
- VMs: Web (http), App (Java Enterprise), DB (for App), Idle, Infra (NFS for Web), and Mail (imap)
- Three SPEC workloads drive one Tile: SPECweb, SPECjApp, and SPECmail
- Each workload is throttled (there are think times between requests)
- SPECjApp workload has peaks/valleys to greatly very resource usage in App & DB VMs
Let's break down how KVM did… RHEL5.5 default

Server Consolidation

Not in any way official SPECvirt data
Let's break down how KVM did... Add hugepages

Server Consolidation

Not in any way official SPECvirt data

Tiles vs. CPU (not official SPECvirt data)
Lets break down how KVM did... add SR-IOV

Server Consolidation

Not in any way official SPECvirt data

Tiles

CPU

vm_exits reduced by 33%
Lets break down how KVM did... add node binding

Server Consolidation

Not in any way official SPECvirt data

Tiles vs CPU

- RHEL55
- +hugepage
- +sr-iov
- +sr-iov+high-int
- +node-bind
Lets break down how KVM did... Recap

- A 54% improvement from baseline to fully tuned
- This is a *lot of manual tuning* to get there
  - Hugepages: figure out how many pages you need, reserve them, mount hugetlbfs, add -mem-path option, etc.
  - SR-IOV: decide which VMs get the virtual functions and assign them. Interrupt coalescing for VF driver is critical.
  - Binding: study resources usage of your VMs, hope that does not change, assign VMs to nodes
- Can we expect users to do this level of tuning? Usually not.

- Let's try SPECvirt again on some newer code
- 2.6.32 +/- few thousand patches
- Qemu 0.12.x
- We should try to get these optimizations without manual tuning
  - Transparent hugepages
  - Vhost-net instead of SR-IOV
  - Automatic node [re]assignment?
Baseline (no hugepage, no vhost, etc)

Server Consolidation

Not in any way official SPECvirt data
Baseline + transparent hugepages

Server Consolidation

Not in any way official SPECvirt data
Baseline + transparent hugepages + vhost_net

- We don't have data for this because
  - We are in the middle of evaluating vhost
  - Some observations:
    - Single-thread vhost is nowhere close enough for this workload
      - Just 2.2 Gbps can saturate the vhost thread
    - Multi-thread vhost evaluation underway
    - Seeing issues with guests that don't have MSI-X for virtio_net
Baseline + transparent hugepages + vhost_net + automatic node binding

- We are not there yet (not implemented)
- Considering the potential gain, we think this deserves a look
- Would like to discuss how to do this
  - Picking the right node on VM start
  - Re balancing VMs: maybe a user-space daemon
Final Thoughts

- Performance
  - KVM can compete well in industry standard benchmarks
  - We should make optimizations automatic when possible

- Benchmarks
  - Should we come up with a server consolidation benchmark of our own?
    - Free
    - Easy to use
    - Easy to share data
Thanks