Kernel Team KVM Contributions
...in support of resource overcommitment

Tim Pepper – tpepper@us.ibm.com
Kernel Team
IBM Linux Technology Center

May 13, 2010
Overview

- Background
- Work items
- Competitive analysis
- Current ability
- Test / acceptance plan
- Upstream outlook
- Value proposition
Background

- Kvm / cloud background assumed known
- Resource overcommitment:
  - Basic workload consolidation / virtualization
  - Multiple guests total resource “usage” greater than actual host resources
    - Eg: 2 guest cpus share 1 host cpu
    - Eg: 2 guest systems each allocated 1.4GB of memory, on host with 2GB total RAM
  - Minimizes hardware expense
  - Enhances viability of cloud business model
Overcommit

- Non-consolidated environment:
Overcommit

- Consolidation w/o overcommit
Overcommit

- Consolidation w/ overcommit
Memory Overcommit

- some IT resources are "renewable", some are “fixed”

- Eg: cpu timeslices vs. memory bytes
  - New cpu time is available all the time
  - New memory bytes don't just appear

- Must balance between extremes
  ...seek efficient sharing of scarce commodity
LI 10-0315.01 (12PM)

- **Memory & CPU overcommitment study - kernel support** (Tim Pepper)

- Add'l memory pressure stats to userspace (per guest), eg:
  - Page unmap rate
  - Swap out rate
  - Swap in rate
  - Page minor refault rate
  - Page major refault rate
  - Refault interval
Virtualization: Cooperative Memory Management support (Dave Hansen)

- **Guest → Host page usage hints eg:**
  - Free pages
  - Clean page cache pages
  - Clean swap cache pages
- **Host → Guest page usage hints eg:**
  - Resident pages
  - Non-resident pages
- ...rocky past; s390 proposal repeatedly shot down by community.
LI 10-0538 (9PM)

- **Adaptive spin locks and scheduling policy optimization for Linux as a guest OS** (Srivatsa Vaddagiri)

- Para-virtual scheduling:
  - Guest hints when in critical section, host does not pre-empt virtual cpu

- Swap locking:
  - Past testing showed swap lock perf issues
  - Diagnosis underway on more recent kernels
Guest friendly policy for page and buffer cache (Balbir Singh)

- Guest cache authoritative?
- Host cache authoritative?
- Double caching?
- Can guest cache in a more “friendly” way?
- Can host + guests cache in synergistic fashion (or at least not parasitic)?
Competitive Landscape

• Vmware:
  – claim 1.7x overcommit via balloon
• Amazon:
  – unknown...
• Oracle:
  – Xen, balloon, "transcendental memory"
  – TM: not upstream, but shipping product
  – Actual overcommit ratio?
• Other:
  – ???
Current Ability

- Balloon and kernel same-page merging (ksm) today give...
- With caveats...
  - 1.7x memory overcommit
  - expect 2x is ok

- Perhaps with additional technologies (ie: this year's line item work)...3x and higher
Current Ability

- Caveats, caveats, caveats: This is not magic! You can't store two bits in one bit...
- Highly workload dependent...generally:
  - Unused memory can be repurposed
  - Homogenous workloads may be able to share pages with the same contents
  - May "steal" mem from idle guests (with penalty if/when it is later needed back)
  - Page cache is good
  - Swap is good

- Best case is idle, homogenous systems (achieves 10x & higher mem overcommit)
Test / Acceptance Plan

• Prototypes tested against synthetic LAMP-based benchmark:
  - Benchmark has quality of service metrics
  - If prototype allows more overcommitment w/o QoS damage...then propose for upstream kernel

• Current kernel landscape leads to expectation of significant push back on the changes we're liable to propose
Upstream Outlook

• High risk for push back, eg:
  – Why not just use: ksm, balloon, placement of VMs, more physical memory?
  – Linus Torvalds recently, "....my only input to this is: numbers talk, bullsh*t walks."
  – Community unhappy w/ “benchmark special”

• Way forward?
  – Agile, co-development with product?
  – Demonstrate abstract benefit?
Value Proposition

- Not workload optimized systems: no workload
- No product interlock
- All work targeted for open source:
  - Upstream kernel patches
  - Open source “Memory Overcommit Manager” (MOM) to tweek tunables
- IBM value add is MOM policies
  - currently no delivery vehicle
  - hope Director (or services?) will choose to implement / ship policy for their markets