experiments1

prateek sharma

05 September 2011

Contents

1 Page Sharing

Survey of everything. Some history?

1.1 Scanning vs Disk based sharing

2 competing approaches. (2)

2 KSM

Diagram of operations, history, and some implementation details Focus on generic nature

2.1 Exp 1: KSM effectiveness

Small experiment which shows that KSM shares large % of same pages. Done earlier in fingerprinting project. Expected result : $>$ 90% sharing of KSM, so good enough. Reason : establish some ground truths: KSM works . Setup : Random workload (doesnt matter) and take fingerprint and see KSM shar% . Run with 1,2,3 VMs. (1)

3 Analysis of shared pages

3.1 Exp 2: Pages shared by flag type

Run some benchmarks (static VMs just booted up ; Kernbench ; HTTP-perf) and see what kinds of pages are shared by flag type. Reason: establish some ground truths : sharing is feasible . Also answer : what kinds of pages are shared? Setup: (1,2,3,5) VMs with different OS running same benchmarks. (diff VMs ; same kernel ; same /var/www)

3.2 Exp 2.1 : KSM with no pagecache pages

Run KSM but skip all guest pagecache pages.

3.3 Exp 2.2: Ftrace Overhead for KSM

Since we are recording all KSM events (just for few experiments only) whats the overhead of that? 1GB/minute data collected.

3.4 Exp 3: Page sharing over time

Run some benchmarks and record pages shared over the duration of benchmark. Also record **KSM overhead**. Reason: Show that KSM overhead is significant enough, thus implying the need for some optimizations. Setup: (2) VMs running benchmarks. KSM being profiled using perf.

(1)

4 Lookahead optimization

4.1 Exp 4: Lookahead success

Run benchmarks on VMs (1,2,3) to on and record lookahead successes. Also record **KSM overhead Compare vanilla KSM overhead with lookahead-optimization**

4.2 Exp 5: Substrings in shared-map.

Record consecutive pages being shared in some benchmarks. Reason : justify why lookahead works. Setup: tracedump analysis simple python script

(3)

5 Problem of double-caching

5.1 Exp 6: Memory savings with exclusive caches

How many pages are there in both places? Setup : Benchmarks on VMs.

5.2 Exp 7: Overhead of ksm-exclusive-cache

Run benchmarks on VMs to record KSM overhead (with ex cache) Reason : scanning vast host page cache could be significant overhead.Also savings might help.

Some caching theory references. (1)

6 Qualitative survey of dynamic memory management for VMs.

6.1 Exp 8: Memory mountains

Look at this problem like L1/2 cache , and build mem-mountains in these cases: (normal ; no guest cache ; no host cache ; swap as ramdisk) Setup : could use IOZone or randal bryant's simple program. Reason : Demonstrate the latencies/throughput of various caches. **This depends on lots of factors like IO schedulers, FS, virtual disk layout etc. Do for any one, for now.**