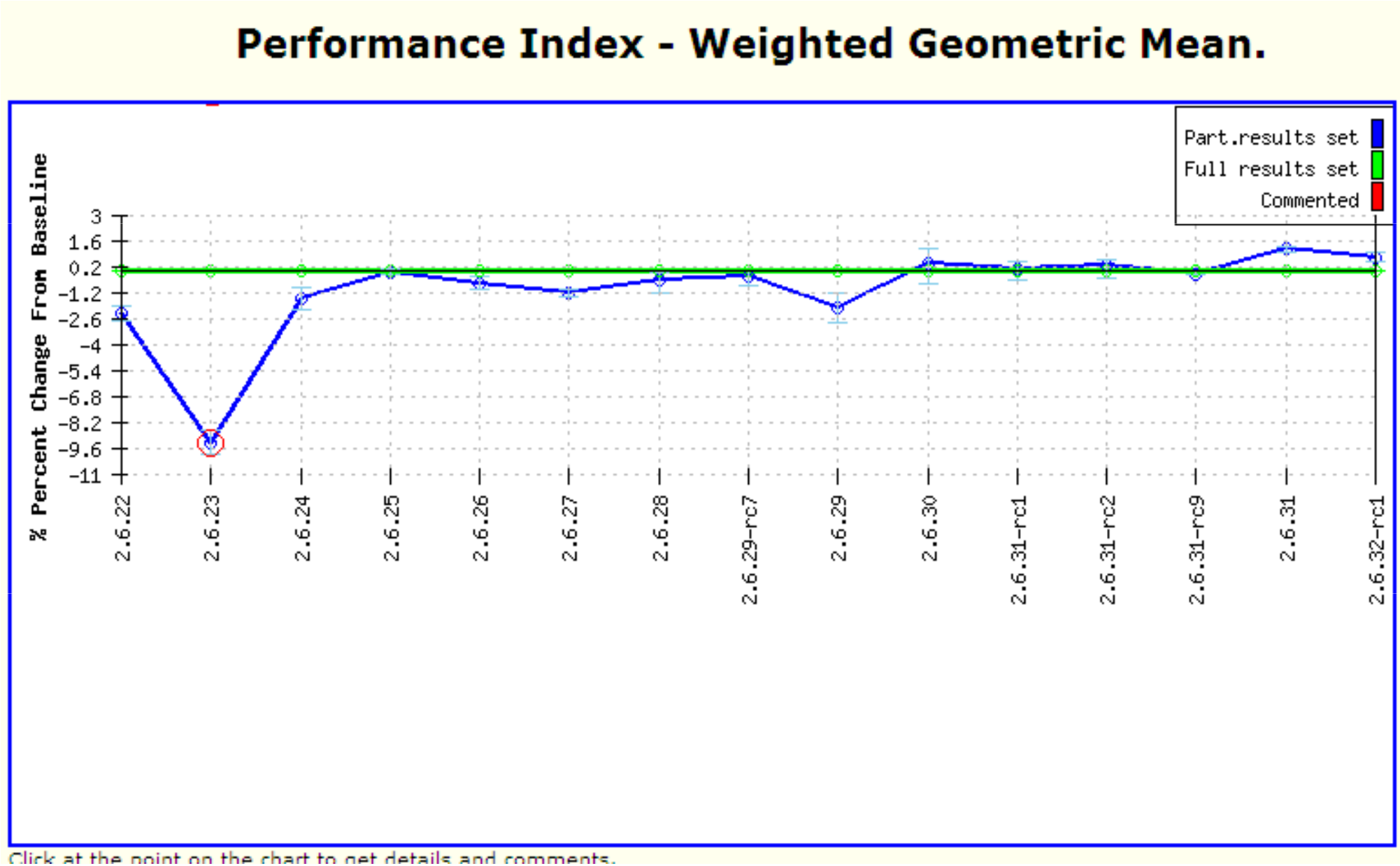
Linux Kernel Performance Tracking

Zhang Yanmin

Senior Software Engineer
Intel Linux Open Source Technology Center
（OTC）

LKP (Linux Kernel Performance)



C-state impact on performance

- **Fio sequential read downgrade**
  - Locate a patch which enables NONSTOP tsc;
  - Get the best result with poll=idle, so we think it's an issue of cpu C state transition;
  - Arjan worked out a patch to create a new idle governor. I worked out a patch against current menu governor.
  - (benchmark is "fio", "no cstates" is using "idle=poll")

	no cstates	current linux	new algorithm
1 disk	107 Mb/s	85 Mb/s	105 Mb/s
2 disks	215 Mb/s	123 Mb/s	209 Mb/s
12 disks	590 Mb/s	320 Mb/s	585 Mb/s

Scalability investigation

- **We investigate 4*8*2 cpu machine scalability**
  - Aim7: captured anon_vma->lock contention and worked out a patch to fix it. HP guys has a better patch. We work together to push the patch to community.
  - Hackbench: we found numa domain is important on hackbench workload. Slqb allocator is better than slub (default configuration). But with a manual slub_min_order=6 configuration, we get a better result.
  - Sysbench+mysql: we located mysql uses a big lock LOCK_open during open_table and close_table. Mysql developers are trying to remove the big lock.

Performance work with community

- We tested Nick Piggin's SLQB patches and compared with SLUB. Provided some suggestions and bug fixes.
- BDIflusher patches: Jens Axboe released a series of patches on disk I/O to replace pdflush. We tested his patches in time and found many bugs. Other guys in community reviewed the patches and tested them, but not so thorough as what we did.
- Process group scheduling: worked with maintainers when they added group scheduling. Caught many issues and pushed developers to change many thresholds rationally.

Regression investigation

- Aim7 45% regression with 2.6.27-rc8: Both tigerton and Nehalem machines captured the same issue. Quickly located the culprit patch about timer implementation and communicated with LKML
- cpu2000 13% regression with 2.6.28-rc1: located the bad patch which causes Nehalem cpu couldn't enter C3 and Turbo mode is not getting activated.
- dbench 15% regression with 2.6.28-rc1 : Quickly located the culprit patch as rework wakeup preemption patch written by Peter. Communicated with him and he reverted it in the new kernel.

Regression investigation (Cont.)

- Hackbench 50% regression and oltp 3.8% regression with 2.6.29-rc3: Peter‘s patch which optimized process timer by allocating per-cpu time var for every processes a spinlock to protect the time var.
- Sysbench+mysql(oltp) 10% regression with 2.6.29-rc4: located the bad patches around sync wake up.
- SLUB optimization: netperf UDP-U-4k loopback is 20% worse than SLQB. Located SLUB doesn’t support large object. Worked with community to add a 4K big object allocation into SLUB.
- Investigated bad swap performance with SLUB: located it as a bad page allocator bypass issue of SLUB.
- SPECJBB2005 7% regression and aim7 1.7% regression: located the bad small patch which is to fix a latency issue.

Regression investigation (Cont.)

- iozone rand-write 30%~60% regression with 2.6.29-rc1: located Nick Piggin's patch that is to fix write_cache_pages cyclic issue. Nick Piggin fixed it later.
- tiobench read 50% regression with 2.6.30-rc1: located Jens' patch which doesn't start queue when getting the first request.

Regression investigation (Cont.)

- **Buffered I/O issue on Nehalem: a user reported the issue. Their applications have plentiful buffered I/O. On Nehalem machine, the performance is dropped. They found the free memory is always more than 2GB while the total memory is 12GB. I checked kernel source codes and did some experiments. The root cause is kernel set sysctl.vm.zone_reclaim_mode=1.**

Regression investigation (Cont.)

- Ffsb file create 16% regression with 2.6.31-rc1: located a fsync cleanup patch
- Tbench 6%~30% regression with 2.6.32-rc1: located the bad patch of SD_PREFER_LOCAL;
- Hackbench 7%~70% regression with 2.6.32-rc1: located 2 bad patches around scheduler (SD_PREFER_LOCAL and stop buddies from hogging the system)
- Disk I/O rand read/write 35% regression with 2.6.32-rc3: located Jens' patch which is to improve desktop interactivity.

VM Enabling

- **prioritize mapped executable pages**
  - page faults directly adds to user perceived delay
  - improves responsiveness by 50% under memory pressure
- **page-types**
  - a handy tool for querying page flags
- **hardware memory failure (cooperative work)**
  - Linux used to panic on memory failure
  - now only affected tasks will be killed

Disk I/O Writeback

- **lumpy pageout (reclaiming dirty pages)**
  - to avoid seek storm and fragmentation
- **dirty throttle wait queue**
  - fast write processes start interleaved and seeky writeback IO
  - now they will wait on the flush thread(s) to do work for them
- **huge writeback chunk size (4MB => 128MB)**
  - to improve IO efficiency and reduce fragmentation
  - avoid negative effects on slow devices (the challenging part)

Acknowledgement