Google

Measuring the Effects of Turbo on VMs

KVM Forum - 27 October 2017

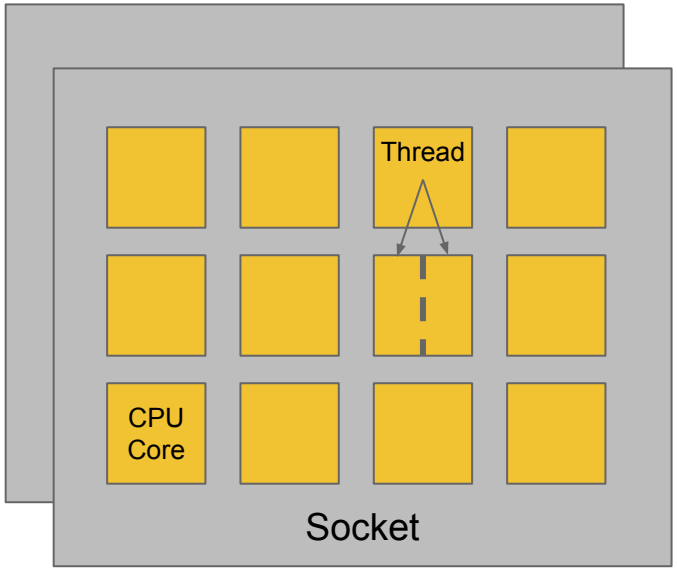
Ben Serebrin
serebrin@google.com

Background on CPU Turbo

What is CPU Turbo?

Modern computers can run components at different frequencies, to maximize performance tradeoffs.

CPU Turbo runs individual or groups of CPU cores at different frequencies, when power and thermal margins allow.



Example system:
2 sockets
12 2-threaded CPU cores per socket

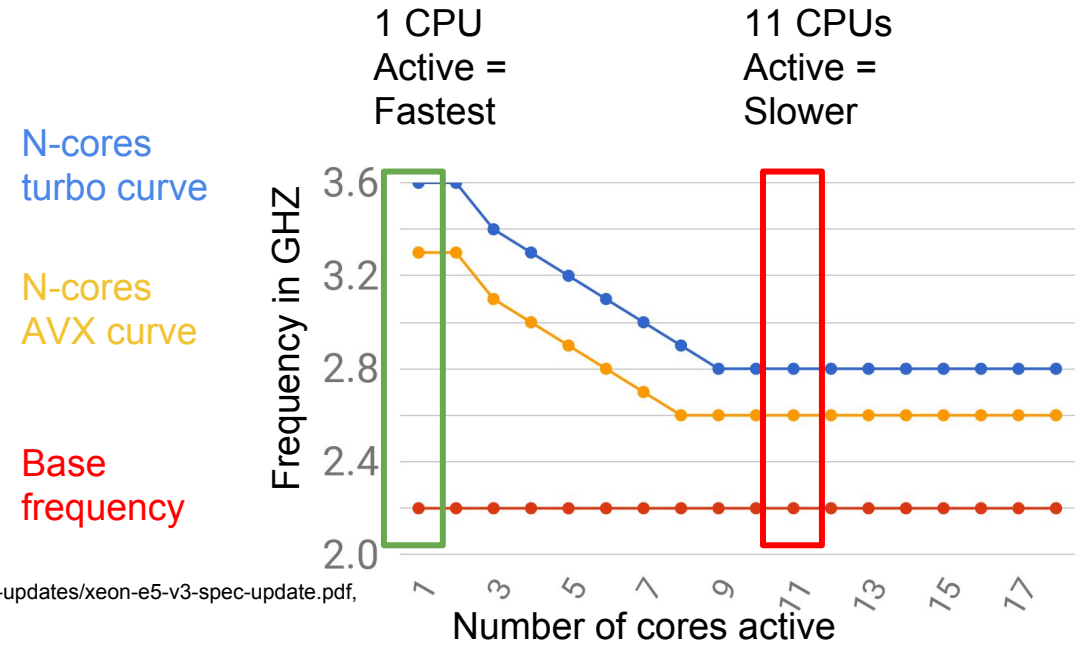
When can CPUs go fast?

When they don't think they'll get too hot or consume too much power.

E.g.: When other CPUs on a socket are HLTed, MWAITing, or in deeper C-states

When do CPUs go slow?

E.g.: When AVX (wide vector FPU) is in use, lose a few 100MHz.



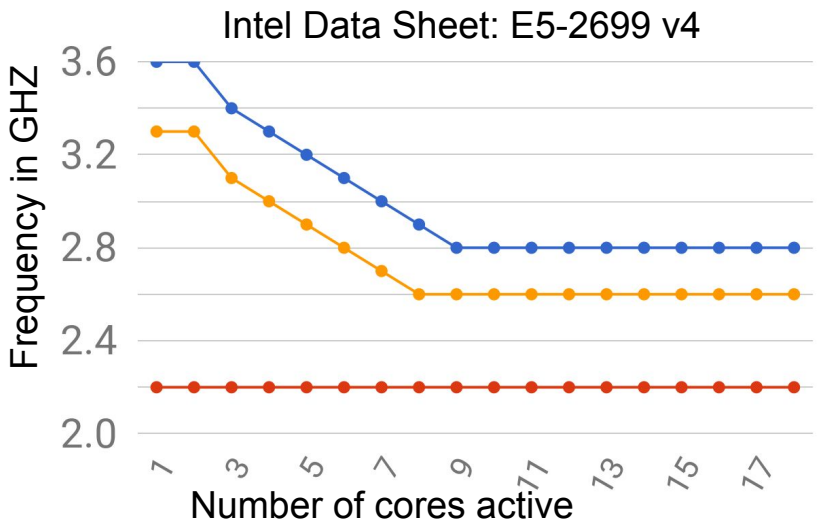
From Intel® Xeon® Processor E5-2600 v3 Product Family Spec Update, https://www.intel.com/content/dam/www/public/us/en/documents/specification-updates/xeon-e5-v3-spec-update.pdf, E5-2699 v3 entries in tables 1, 2, 3

Quiz: which CPU is faster?

CPU A: 2.3 GHz Broadwell E5-2673 v4 with 3.5 GHz turbo boost

CPU B: 2.3 GHz (base) Broadwell E5-2686 v4, 2.7 GHz (turbo)

CPU C: 2.2 GHz Broadwell E5 v4

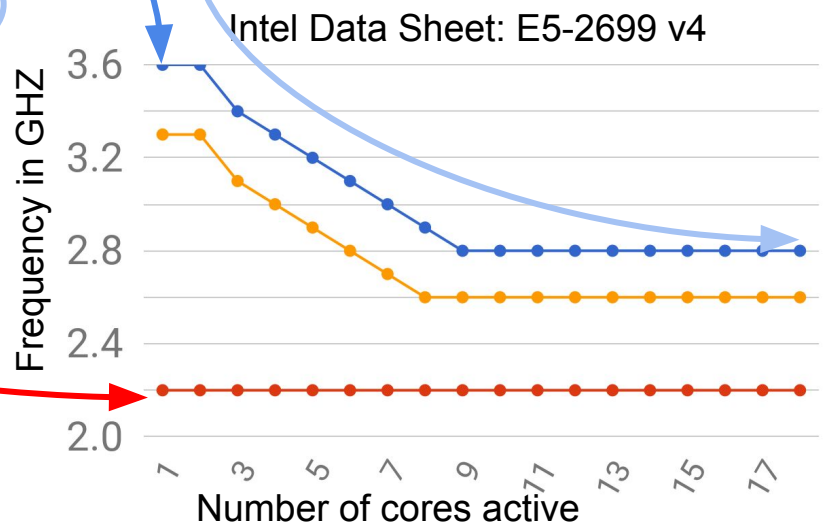


Quiz: which CPU is faster?

CPU A: 2.3 GHz Broadwell E5-2673 v4 with 3.5 GHz turbo boost

CPU B: 2.3 GHz (base) Broadwell E5-2686 v4, 2.7 GHz (turbo)

CPU C: 2.2 GHz Broadwell E5 v4 ??



Answer: You can't tell.

What frequency term indicates actual performance potential?

Most machines and vendors advertise base frequency, which can be confusing. Cores typically run faster than base frequency.

Base frequency == TSC[1] Frequency == Advertised Frequency

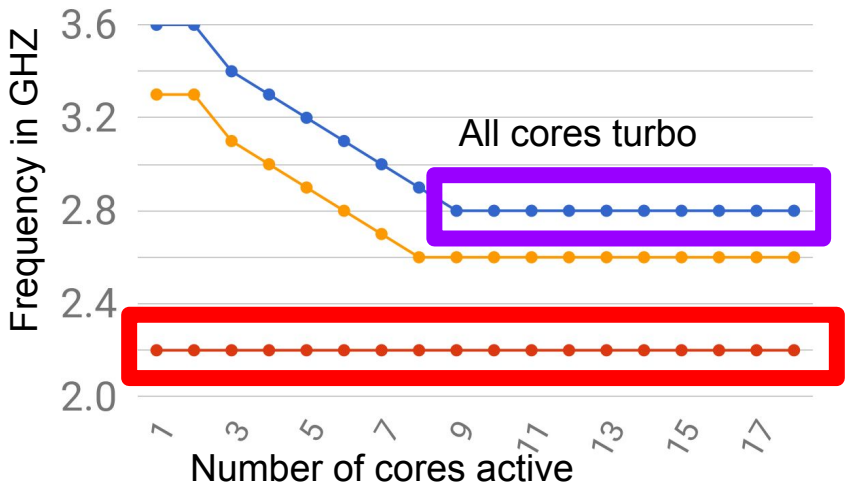
It would be much more useful if all vendors indicate **all-cores-turbo** instead of base frequency.

1: Time Stamp Counter (constant rate clock source)

N-cores turbo curve

N-cores AVX curve

Base frequency



Measuring Turbo

Measuring Turbo in non-virtualized machines

Extensive monitoring available per hyperthread

- APERF/MPERF MSRs[1] indicate ratio of total cycles (including bonus cycles due to Turbo) to constant-clock cycles: how much did I get turbo boosted?
  - Average frequency of last sample period = Δ APERF / Δ MPERF

- Turbostat tool: APERF/MPERF plus C-state residency
- MSR 0x198: MSR_PERF_STATUS Current frequency
- MSR 0x1AD-0x1AF: Model-specific non-AVX turbo curve values

Most or all of these MSRs are unavailable to most VMs.

1: MSR: Model Specific Register. See Intel SDM 3B section 14.2 for APERF/MPERF details.

Measuring frequency effects is difficult in a VM.

Most VMs don’t know much about their world:

- Socket size (CPU count)
- Which socket or hyperthread each VCPU is running on at the moment
- Actual CPU frequency curves or instantaneous frequency

Any of these contributes to unattributed performance variation and confuses cloud customers.

Why do hypervisors and guests care about measuring Turbo?

Diagnose anomalous performance

- Unlucky scheduling next to a thread using AVX, or on a heavily loaded socket
- Overly good scheduling in an underused socket
- Convenient aggregate of time stolen from a VM

Avoid disappointment

- Lucky benchmark results cannot be repeated later or under live traffic

Our Contribution

Relatively simple: Our code creates per-VCPU histograms of residency at each 100MHz bucket, and exports them via debugfs

- Buckets selected to cover common frequencies
  - < 1100 MHz
  - 1100-1199 MHz
  - 1200-1399 MHz
  - ...
  - >= 5000 MHz

We set the bucket sizes to be constant to avoid per-generation headaches in aggregation.

Enablement

Currently implemented in Intel `vmx.c`, should be a straightforward port to AMD.

`kvm-intel.o` module param `measure_turbo` defaults to true unless the hardware does not report APERF/MPERF (expected only in nested virtualization or very old hardware).

How we measure: Tracking VCPUs, not Physical CPUs

The scheduler may move VCPUs to arbitrary Physical CPUs.

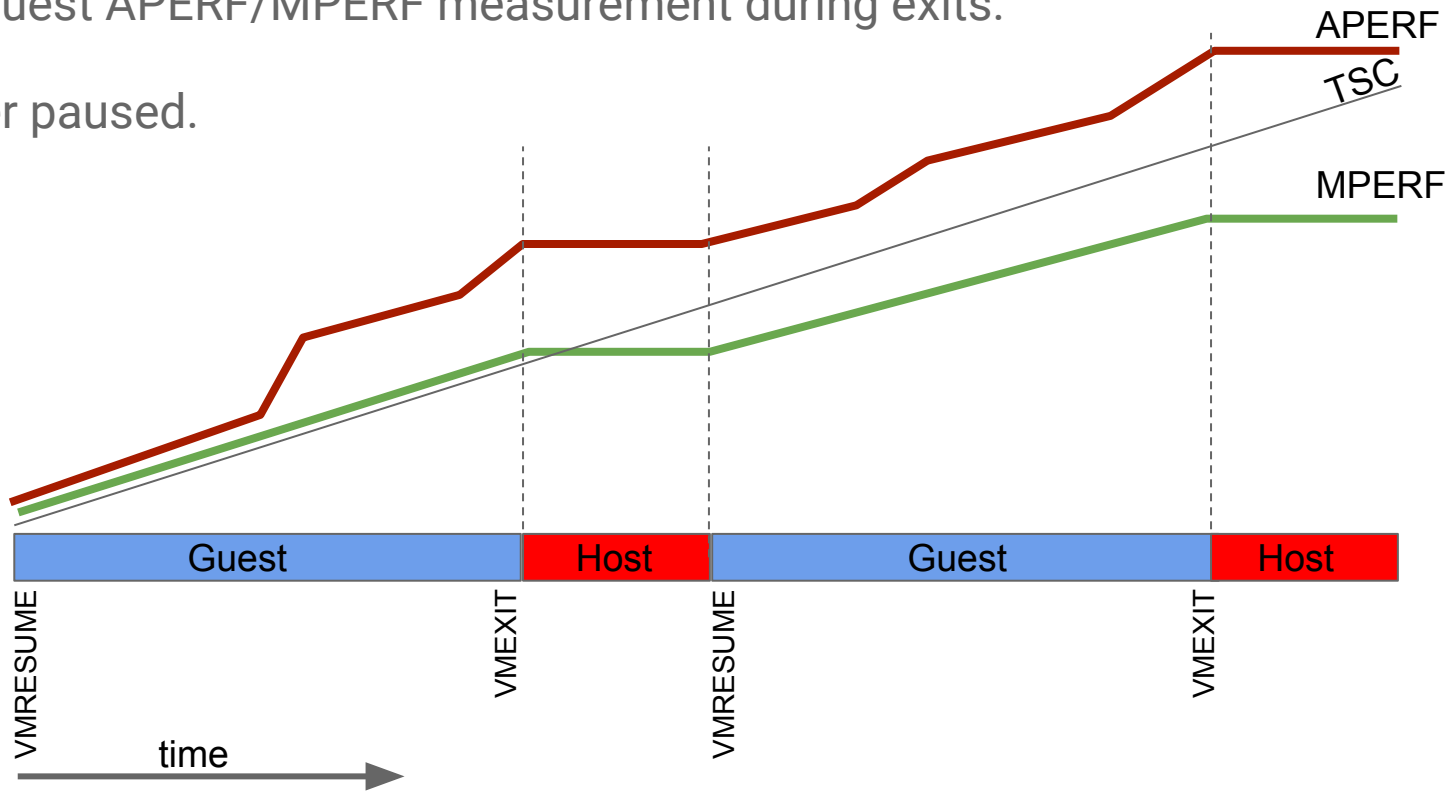
Host APERF and MPERF track Physical CPUs, so this code samples the APERF and MPERF deltas on the current Physical CPU while the VCPU is running there.

This has the desired effect of tracking each VCPU’s turbo history as it moves around the system’s CPUs.

When we measure

We pause guest APERF/MPERF measurement during exits.

TSC is never paused.



Limitations and Tradeoffs

Difficult to consistently define APERF/MPERF while VCPU isn't running. We chose to stop counting, so we do not represent time spent in guest emulation (eg. CPUID intercept, MMIO).

- Hypervisor time is ignored. (When should you stop timing an asynchronous path?) Not measuring long-latency hypervisor work is especially misleading.

We measure "Average frequency" as defined in turbostat; we can't measure time spent in each frequency bin directly.

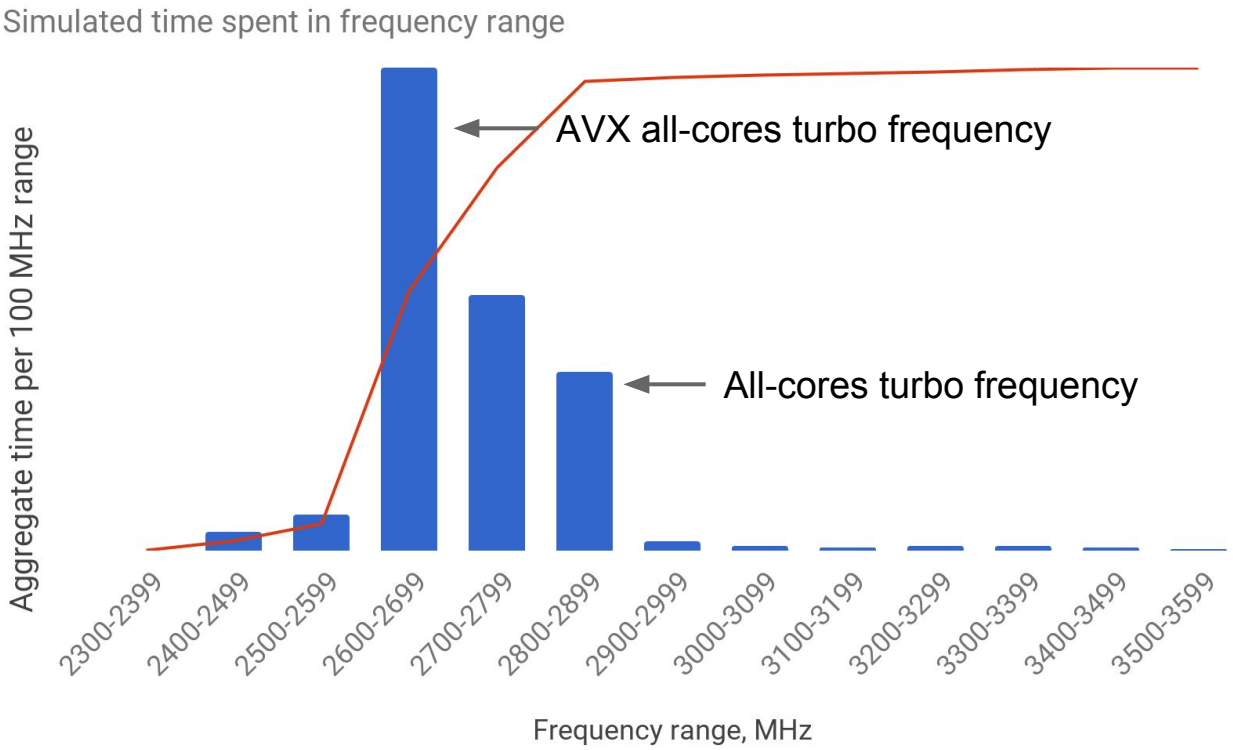
We cannot directly tell why we were throttled.

Example output for single VM

```
turbo_time_hist      :
0,0,0,[...],0,22879,470581,882145,11976814,6326829,4422419,216405,11958
8,58441,107977,99586,68693,34747,0,[...],0
(0,0,0,[...],0,22879,470581,882145,11976814,6326829,4422419,216405,1195
88,58441,107977,99586,68693,34747,0,[...],0)

turbo_tsc            : 23687783 (23687783)

turbo_mperf          : 19739820 (19739820)

turbo_aperf          : 24787105 (24787105)
```

Example measurement

We sample the histograms for each VM every few minutes.



Using the metrics

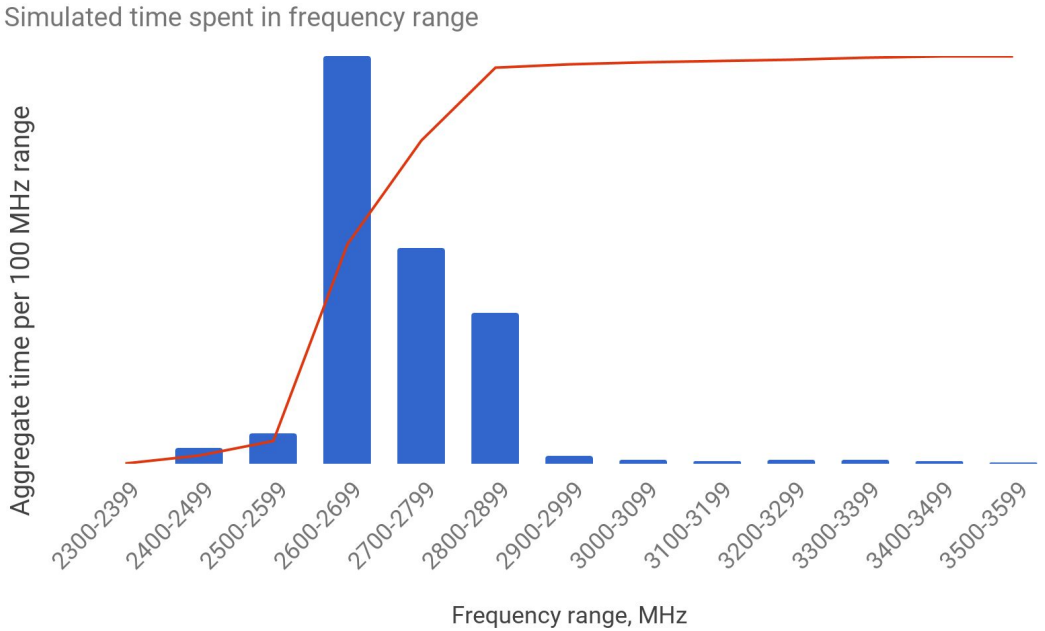
Example forensics:

Customer reports anomalous performance at 18:30 on 25-October.

- Admin looks at Turbo histograms for that VM and all co-located VMs
- Correlated low speeds? Implications of AVX or other throttling workloads?

This cluster shows significant time at the AVX turbo frequency.

Some VMs also get abnormally high performance bursts.



Possible future work

Consider exposing APERF/MPERF to guests

- But live migration and other concerns. E.g. migration to different base frequency CPU, or to different CPU generation can cause the APERF and MPERF MSRs to work inconsistently with the pre-migration state.

Connect with virtualized performance counters to enable deeper in-guest profiling and tuning

Conclusion

Measuring per-VM and aggregate turbo residency allows a user new insight to diagnose CPU-level performance anomalies.

Metrics could include indications to a guest that it received better-than-typical (or worse-than-typical) frequencies over a period of time, to set expectations during workload tuning.

We also suggest using all cores turbo rather than base frequency, as a more intuitive metric to enable realistic comparisons of different CPU versions.