

## User Guide

### Authors:

Martin Dimitrov, Carl Strickland, Seung-Woo Kim, Karthik Kumar, Kshitij Doshi

### Introduction :

Power Governor (power\_gov) is a software utility and library, which allows developers to (a) monitor power and (b) regulate power at very fine time granularities (few tens of milliseconds). Power monitoring/control is available for the package, core, graphics, uncore and DRAM domains, as illustrated in Figure 1 below. The tool is self-contained, easy to use, and available on Intel® Xeon® E5 series processors based on Intel® microarchitecture code-named Sandy Bridge EP/EN/E, 2nd Generation Intel® Core™ and newer processors. As a library, power\_gov allows developers to incorporate power monitoring/control into their own custom and dynamic solutions tailored to the needs of their application.

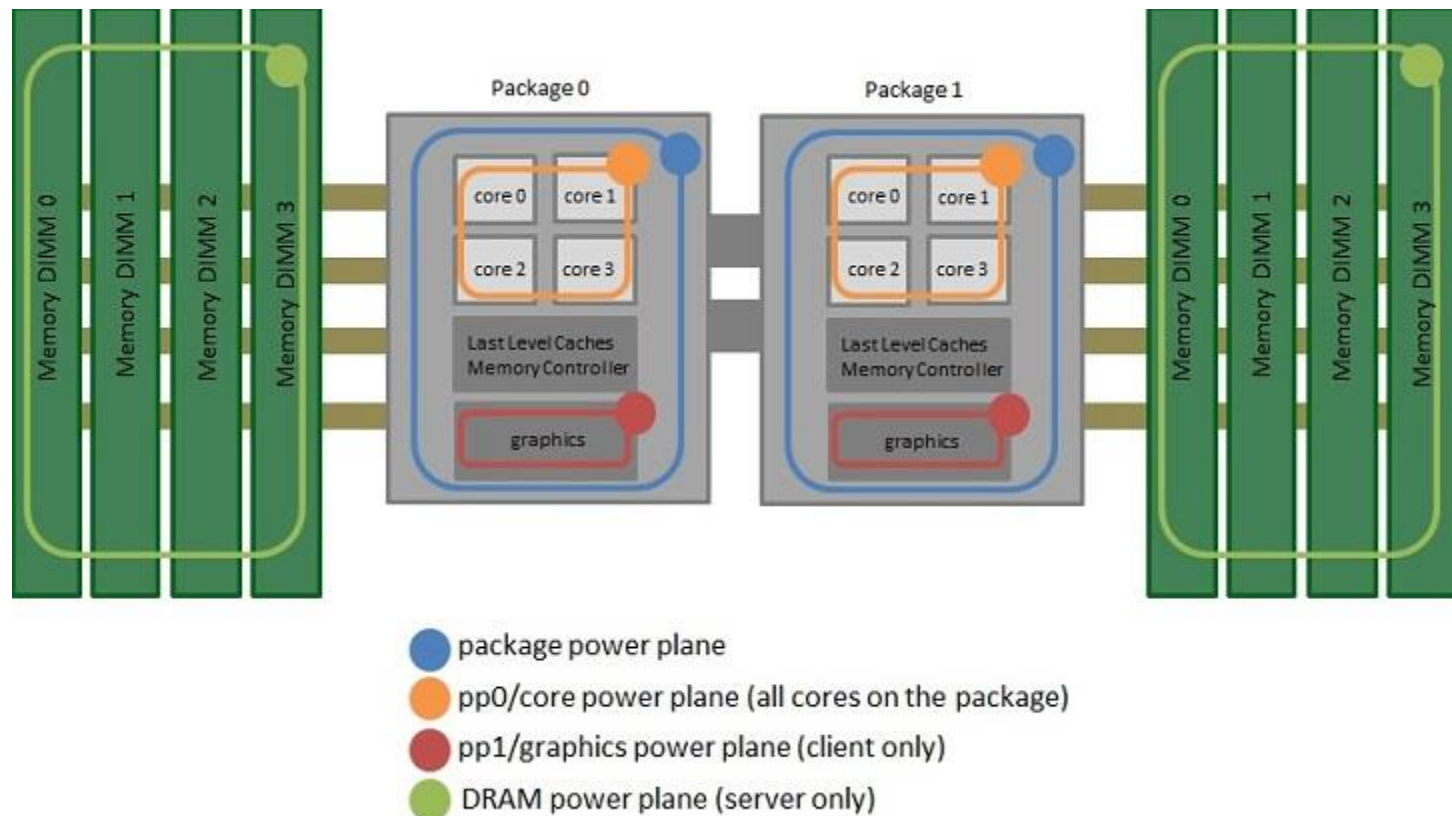


Figure 1. Power domains for which power monitoring/control is available. To get uncore (last level caches and memory controller) power, subtract the core and graphics from

package. Note: graphics power monitoring/control is only available on client parts, while the DRAM power monitoring/control is only available on server parts.

## Usages :

Software power meter: power\_gov can be used to report power consumption on the different power planes. The example in Figure 2 shows power\_gov executing on a two socket Intel® Xeon® Processor E5-based machine, and reporting the average power consumption of the package, core, uncore and DRAM for each socket at 1 second intervals.

```
mementest:/opt/power_gov/trunk # ./power_gov -e 1000
Node,PKG,Core,Uncore,DRAM
0,12.43,2.15,10.29,3.69
1,12.63,2.30,10.33,3.67
0,13.04,2.28,10.76,3.58
1,13.21,2.44,10.77,3.53
0,12.35,2.21,10.14,3.63
1,12.51,2.36,10.15,3.66
0,11.89,2.05,9.84,3.63
```

Figure 2. power\_gov reporting power consumption at 1-second intervals for package, core, uncore, DRAM domains on a two-socket system.

Optimize for power/performance target: power\_gov can be used to enforce power limits on the different power domains. In the example from Figure 3, we executed an OLTP workload, while at the same time we varied the power limit enforced on the package power domain (from 130W down to 30W on the x-axis). At the same time, on the y-axis we observed how the performance of the workload (in terms of average response time of transactions) varied with the enforced power limit. Assuming that our response time target was 0.015ms, we could limit the power consumption of the CPU socket to 50W while still satisfying the performance requirement.

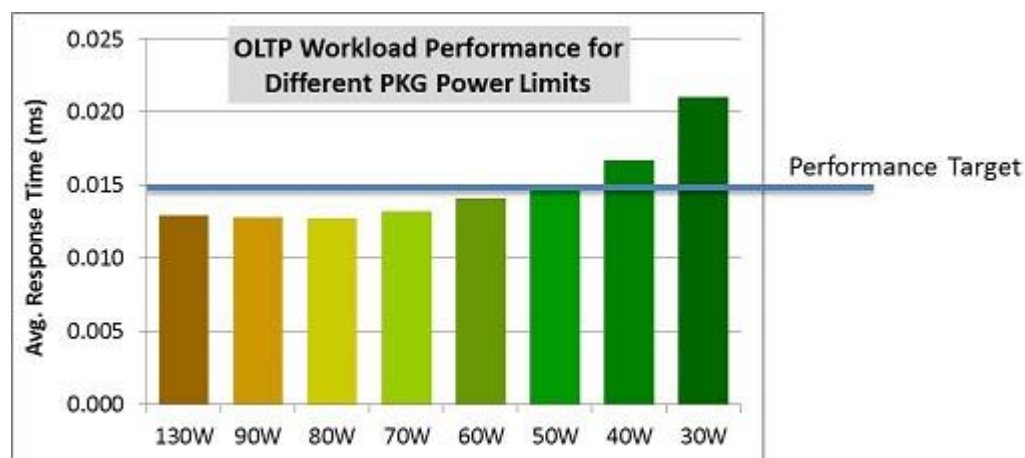


Figure 3. Using power\_gov to optimize for a power/performance target.

Dynamic power/performance optimization: Using the `power_gov` library allows developers to create dynamic power monitoring/control solutions, which respond to changes in workload demand, time of day, etc. The example in Figure 4 depicts a scenario in which the customer software monitors energy consumption, in addition to its own quality of service (QoS) metrics and responds to changes in workload demand by dynamically adjusting the power limits, see ref[1]. The control algorithms used to dynamically adjust the power limits depend on the usage scenario and can be arbitrarily complex.

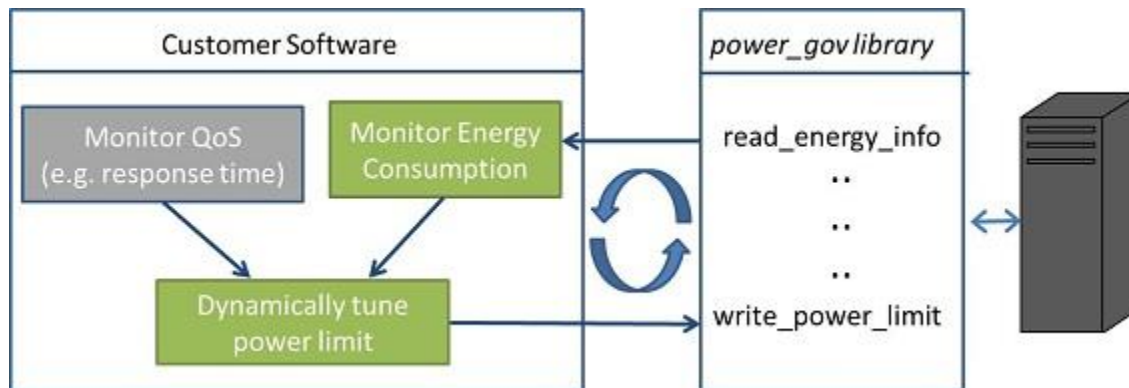


Figure 4. Using the `power_gov` library in order to design dynamic power/performance optimization solutions.

## References:

- [1] V. Anagnostopoulou, M. Dimitrov, K. Doshi, "SLA-Guided Energy Savings for Enterprise Servers", ISPASS 2012 poster, full version to appear in ITJ 2012
- [2] Intel® 64 and IA-32 Architectures Software Developer's Manual Volume 3B: System Programming Guide, Part 2