Qualcomm
TECHNOLOGIES, INC

CPU Governor Parameters

Application Note

80-NR256-3 A

September 4, 2014

Submit technical questions at:
https://support.cdmatech.com/

Confidential and Proprietary – Qualcomm Technologies, Inc.

Qualcomm Technologies, Inc.
5775 Morehouse Drive
San Diego, CA 92121
U.S.A.

Contents

Tables

Revision history

Revision	Date	Description
A	Sep 2014	Initial release

1 Introduction

1.1 Purpose

The purpose of this document is to understand the governor parameters of interactive governor and how they work. This document also explains CPU boost parameters.

This document is written for engineers who need to understand the process for collecting system profiling data for QoS-related issues. The scope is limited to the Android™ platform. Steps have been validated to be working on Qualcomm Technologies, Inc. (QTI) MSM™ chipsets on the Android platform.

1.2 Conventions

Function declarations, function names, type declarations, and code samples appear in a different font, e.g., `#include`.

1.3 References

Reference documents are listed in Table 1-1. Reference documents that are no longer applicable are deleted from this table; therefore, reference numbers may not be sequential.

Table 1-1 Reference documents and standards

Ref.	Document	
Qualcomm Technologies		
Q1	*Application Note: Software Glossary for Customers*	CL93-V3077-1

1.4 Technical assistance

For assistance or clarification on information in this document, submit a case to QTI at https://support.cdmatech.com/.

If you do not have access to the CDMATech Support website, register for access or send email to support.cdmatech@qti.qualcomm.com.

1.5 Acronyms

For definitions of terms and abbreviations, see [Q1].

2 Governor Parameters

2.1 Interactive governor parameters

The node to read these parameters is located at /sys/devices/system/cpu/cpufreq/interactive.

- target_loads − The CPU frequency is adjusted to achieve this load. target_loads also accepts strings as arguments such that it can be different for different values of current frequency.

  For example, the string ‘85 1000000:90 1700000:99’ would mean:

  target_loads = 85, if cur_freq < 1 GHz

  90, if 1 GHz < cur_freq < 1.7 GHz

  99, if cur_freq > 1.7 GHz

  The higher the target_loads value for a particular frequency, the lower would be the next frequency picked so that the load is achieved till the next.

  The lower the target_loads, the more often the governor will raise CPU speeds to bring the load below the target.

- Hispeed_freq − This is the intermediate frequency to jump in case the load exceeds ‘go_hispeed_load’. If the load stays high for the amount of time specified in above_hispeed_delay, then the speed may be bumped higher.
- Go_hispeed_load − If the load exceeds this value, then the next frequency chosen is at least hispeed.
- Above_hispeed_delay − Keep the CPU frequency at hispeed_freq (or above) for min_sample_time before ramping up the frequency.
- Min_sample_time – This is the minimum time interval to wait at any frequency before dropping to lower frequencies.
- Sampling_rate – This is the sampling rate of interactive governor. This is how often you want the kernel to look at the CPU usage and to make decisions on what to do about the frequency.

  *Note:This parameter is not applicable for 8916
- Sampling_down_factor – This parameter controls the rate at which the kernel makes a decision on when to decrease the frequency while running at top speed. When set to 1 (the default) decisions to reevaluate load are made at the same interval regardless of current clock speed. But when set to greater than 1 (e.g. 100) it acts as a multiplier for the scheduling interval for reevaluating load when the CPU is at its top speed due to high load. This improves performance by reducing the overhead of load evaluation and helping the CPU stay at its top speed when truly busy, rather than shifting back and forth in speed.

- Sync_Freq Feature – This feature will cause a CPU frequency to stay above a particular value (sync_freq) if certain conditions (determined by the two nodes up_threshold_any_cpu_freq and up_threshold_any_cpu_load) are satisfied.
  - Up_threshold_any_cpu_freq – If the maximum frequency across all the CPUs is higher than or equal to this frequency value, do not let the current CPU fall below sync_freq. The higher this value, the less the chances to go to sync_freq.
  - Up_threshold_any_cpu_load – If the maximum load across all the CPUs is higher than or equal to this load value, do not let the current CPU fall below sync_freq. The higher this value, the less the chances to go to sync_freq.
  - Sync_freq – Only when *both* of the above conditions are satisfied will the CPU *not* drop below this frequency value. The higher this value, the higher the frequency to jump will be when the above conditions are satisfied.

  *Note:This parameter is not applicable for 8916.

2.2 CPU boost parameters

The node to read these parameters is located at /sys/module/cpu_boost/parameters.

- Boost_ms − This is the value in milliseconds for which the scaling_min will stay at the decided value (by the cpu-boost driver) after thread migration.

  For example, if boost_ms is 40 ms, then the scaling_min frequency of the CPU will remain at the decided value by the cpu-boost driver for 40 ms.
- Input_boost_freq − This is the frequency value to which the scaling_min of all online cores will be raised to on an input event (currently just touch events).

  For example, if input_boost_freq is 1.5 GHz, as soon as the touch event occurs, the scaling_min frequency of the CPU will be 1.5 GHz.
- Input_boost_ms − This is the time in milliseconds for which the input boost is effective. This parameter plays a vital role in the touch operations. At any touch operation the CPU stays at a minimum of input_boost_freq for input_boost_ms amount of time. If this is reduced, the latencies might increase and vice versa.

  For example, if input boost_ms is 40 ms, then as soon as the touch event occurs, the CPU scaling_min will remain at a minimum of input_boost_freq for 40 ms.
- Sync_threshold − This is an upper limit on the frequency sync on the task migration such that if the source frequency is above this value, the destination CPU's scaling_min will be set to this value (rather than the source core's frequency as was the case before this feature).

  For example, if sync_threshold is at 1.72 GHz and the thread migrates from CPU0 to CPU1, and if the destination CPU is running at a frequency higher than 1.72 GHz, then scaling_min of CPU1 will be set to 1.72 GHz. If CPU1 is running at lower than the 1.72 GHz, then scaling_min freq of CPU1 will remain the same.