PowerHalService Programming Guide – MT6765

Programming Guide

Customer Support

Doc No: CS6765-BD8E-PGD-V1.0EN
Version: V1.0
Release date: 2018-07-23
Classification: Standard

Keywords

PerfService, Common Part

MediaTek Inc.

Postal address

No. 1, Dusing 1st Rd. , Hsinchu Science Park, Hsinchu City, Taiwan 30078

MTK support office address

No. 1, Dusing 1st Rd. , Hsinchu Science Park, Hsinchu City, Taiwan 30078

Internet

http://www.mediatek.com/

Document Revision History

Revision	Date	Author	Description
A1	2018-03-20	Jimmy-YJ Huang	Initial Release

Table of Contents

Table of Contents

Table of Contents

Lists of Tables

Lists of Figures

1 Introduction

1.1 Purpose

This document provides the programming guidelines for MTK Power Hal and associated modules. It describes how to use MTK Power Hal Service API on the Android platform.

This document introduces data type and API of MTK Power Hal Service. User space program can use these API to control hotplug and DVFS. It also

This manual also elaborates the mechanism required to use MTK Power Hal.

1.2 Scope

The document provide the programming details of MTK Power Hal Service.

Table 1-1 presents the reference information of the modules which are used but beyond the scope.

Table 1-1. Reference Information beyond Scope

Modules	Reference information
Hotplug	Source code: kernel-4.4/drivers/misc/mediatek/base/power/hps_v3
DVFS	kernel-4.4/Documentation/cpu-freq/governors.txt 2.6 Interactive
PPM	Source code: kernel-4.4/drivers/misc/mediatek/base/power/ppm_v2
EAS	Source code: kernel-4.4/drivers/misc/mediatek/sched

1.3 Who Should Read This Document

This document is primarily intended for:

- Engineers with technical knowledge of performance and low power trade-off.
- Engineers who is responsible for programming in user space.

1.4 How to Use This Manual

This segment explains how information is distributed in this document, and presents some cues and examples to simplify finding and understanding information in this document. Table 1-2 presents an overview of the chapters and appendices in this document.

Table 1-2. Chapter Overview

#	Chapter	Contents
1	Introduction	Describes the scope and layout of this document.

1.4.1 Terms and Conventions

This document uses special terms and typographical conventions to help you easily identify various information types in this document. These cues are designed to simply finding and understanding the information this document contains.

Table 1-3. Conventions

Convention	Usage	Example
[1]	Serial number of a document in the order of appearance in the References topic	Look up Chapter 2: System Architecture in [1]
void xx(zz)	Source code	static int __stdcall cb_download_bloader_init(void *usr_arg){}
☞	Important	

2 References

The following documents contain provisions which, through reference in this text, constitute provisions of the present document.

[1] Hotplug strategy source code: kernel-4.4/drivers/misc/mediatek/base/power/hps_v3

[2] DVFS, kernel-4.4/Documentation/cpu-freq/governors.txt 2.6 Interactive

[3] PPM, Source code: kernel-4.4/drivers/misc/mediatek/base/power/ppm_v2

[4] EAS, Source code: kernel-4.4/drivers/misc/mediatek/sched

3 Definitions

For the purposes of the present document, the following terms and definitions apply:

Hotplug: A kernel module which controls CPU online and offline.

DVFS: A kernel module which controls CPU freq.

Thermal managerment: Thermal management is a feature to monitor the temperature on the device and take actions accordingly.

ACAO: All cores are always online. MTK new platforms which include mt6765 support ACAO. It means all CPU cores are always online. All CPU core related API of PowerHAL are invalid in ACAO platform.

EAS: Energy awared scheduler. It is MTK proprietary enhancement of scheduler.

4 Abbreviations

Please note the abbreviations and their explanations provided in Table 4-1. They are used in many fundamental definitions and explanations in this document and are specific to the information that this document contains.

Table 4-1. Abbreviations

Abbreviations	Explanation
MTK	MediaTek, Asia's largest fabless IC design company.
PerfService	Performance service
HPS	CPU hotplug strategy
DVFS	Dynamic Voltage Frequency Scaling
PPM	Performance and Power management
AMS	Activity manager service
WMS	Window manager service
ACAO	All cores are always online
EAS	Energy awared scheduler

5 Overview

This chapter first gives a brief description of the modules of the system and the relationship of the modules.

In general, CPU core number and CPU frequency is controlled by kernel driver. Hotplug driver is responsible for CPU core number. DVFS is responsible for CPU freq. They change cpu core number and freq dynamically according to system loading. If loading is high, they may enable more cpu and raise cpu freq. They may also disable cpu and lower freq if loading is no longer high.

There are many parameters in hotplug and DVFS policy. It is difficult to provide a policy which satisfies all scenarios. Fox example, if the policy favor low power, performance may drop. Besides, these policies need some time to make the decision. It may also cause performance drop. For example, the loading of app launch is high enough to enable two CPU cores. However, cpu1 is enabled too late to improve app launch time.

MTK Power Hal Service is a SW module that extend from google power hal service in user space. It provides interface between kernel driver and user program. User program can use these interface to boost up computing power itself. For example, app launch and exit can force more cpu online before hotplug driver.

This document introduces data type and interface of MTK Power Hal Service. It also provides sample codes. User can refer these sample codes to write your own program.

5.1 Architecture

We describe the software architecture of MTK Power Hal Service in this section.

5.1.1 Software concept

Concept of design and implement.

- MTK Power Hal Service bridges user scenarios to cpu cores and cpu freq.
- CPU hotplug / DVFS module provides interface for easily control multiple-cores and freq.
- Notify user scenarios to MTK Power Hal Service are done by Android Framework Services such as AMS and WMS
- Provide mechanism to register user's own scenario. User can define cpu core number and cpu freq itself.

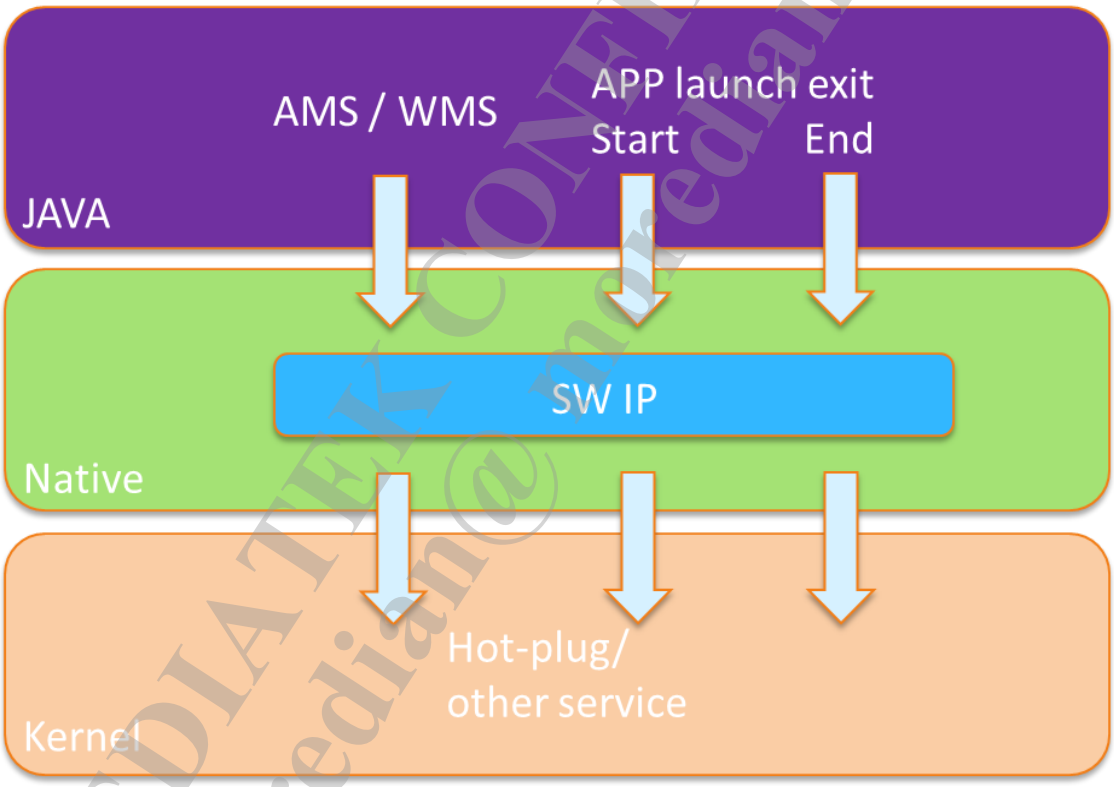

Figure 5-1. MTK Power Hal Service concept

5.1.2 Modules

MTK Power Hal Service consists of many components. Figure 5-2 shows block diagram to demonstrate these components. MTK Power Hal Service which user can get directly in java is **only supported in turnkey solution**, because we don't modify system server in BSP package.

5.1.2.1 Power Manager

Power Manager provide the interface which could use MTK power hal service directly.

5.1.2.2 MTK Power Hal Service

Provide service for handling performance/power management actions.

5.1.2.3 Native Client

Native Client means hals that would use MTK Power Hal Service.

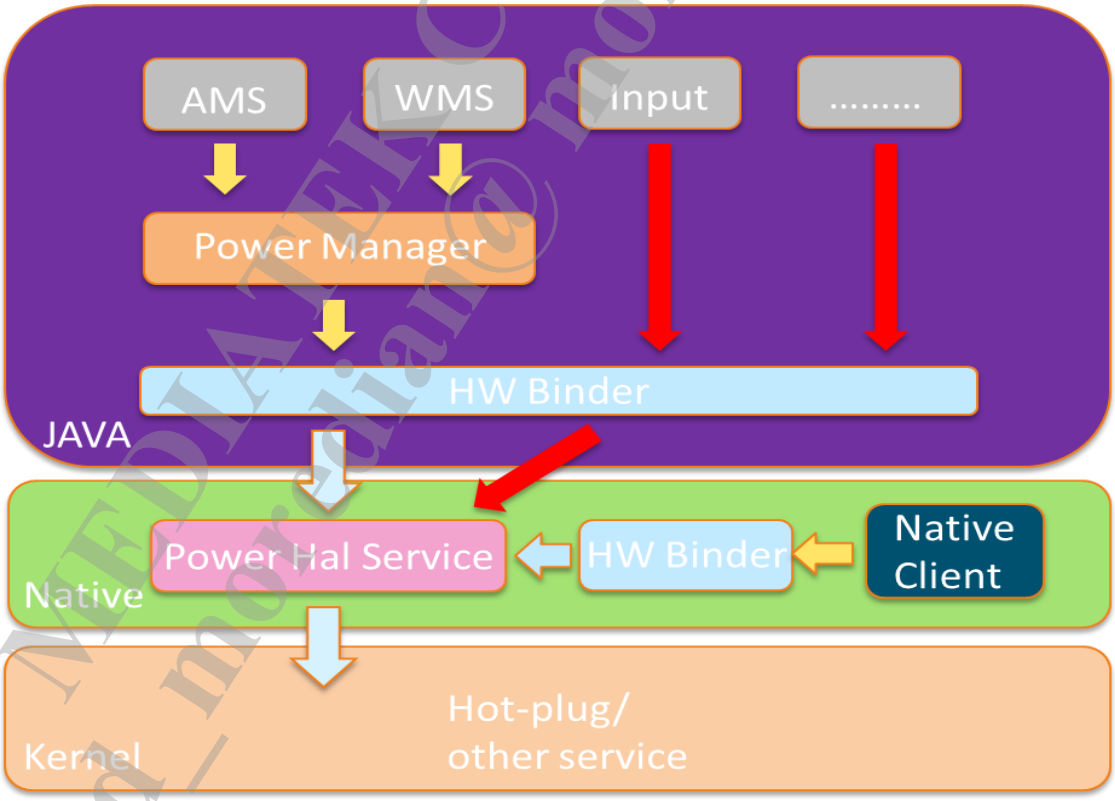

Figure 5-2. MTK Power Hal Service block diagram

5.2 Usage

Use MTK Power Hal Service API should follow the flow list bellowed.

Pre-defined Scenario

- **Enable Performance Boost (with / without timeout)**
- Run user's own behaviors
- **Disable Performance Boost**

Customized Power Hint

- **Add customized hint and config**
- **Enable Performance Boost (with / without timeout)**
- Run user's own behaviors
- **Disable Performance Boost**

User defined Scenario

- **Register user's own scenario**
- **Config scenario (if necessary)**
- **Enable Performance Boost (with / without timeout)**
- Run user's own behaviors
- **Disable Performance Boost**
- **Unregister user's scenario if no more use**

Note:

User shall disable performance boost after the job is finished. Otherwise, system may not enable deep sleep mode.

5.3 Decision Policy

Some users need more computing power to improve performance. Some others limit computing power to save power consumption. MTK Power Hal has to decide the actual setting if there are more than one scenarios enabled.

Though MTK Power Hal Service tries to satisfy all users, there may be some requirements conflicted. MTK Power Hal Service have corresponding handler to make the final decision to favor performance. Following sections describe the policy which the handler follow to make the decision.

5.3.1 Policy of CPU core

Decision flow of cpu core is listed below.

Step 1. Decide total cpu core

Step 2. Decide min cpu core of each cluster

Step 3. Decide max cpu core of each cluster

Step 4. Decide actual value of max cpu core (if max < min)

Following example demonstrates the decision flow.

5.3.1.1 Requirement

User A needs 3LL at least and no L online. Therefore, it sets LL min to 3 and L max to 0.

User B needs 1LL at most and 2L at least. Therefore, it sets LL max to 1 and L min to 2.

User C needs 1 ~ 2LL and 3L. Therefore, it sets LL min to 1 and LL max to 2. It also set both L min and L max to 3.

5.3.1.2 Step 1. Decide total cpu core

1. Calculate total core number of each scenario
2. Choose the maximal value to satisfy all scenarios

Table 5-1. CPU core step 1

Scenario	LL min	L min	Total core
A	3	0	3 + 0 = 3
B	0	2	0 + 2 = 2
C	1	3	1 + 3 = 4
Final setting			max (3, 2, 4) = 4

5.3.1.3 Step 2. Decide min cpu core of each cluster

1. Choose the maximal value of each cluster to satisfy all scenarios.
2. Decide the actual value of L cluster.
3. Decide the actual value of LL cluster. (total core – L cpu core)

Table 5-2. CPU core step 2

Scenario	LL min	L min	Total core
A	3	0	
B	0	2	
C	1	3	
(1) Maximal value	max (3, 0, 1) = 3	max (0, 2, 3) = 3	
(2) Final setting of L		3	4 (refer to Table 5-1)
(3) Final setting of LL	4 – 3 = 1		

5.3.1.4 Step 3. Decide max cpu core of each cluster

Choose the minimal value of satisfy all scenarios

Table 5-3. CPU core step 3

Scenario	LL min	L min
A	-1	0
B	1	-1
C	2	3
Final setting	min (don't care, 1, 2) = 1	min (0, don't care, 3) = 0

5.3.1.5 Step 4. Decide actual value of max cpu core (if max < min)

1. If max >= min, we set both values to driver directly.
2. If max < min, we make the final decision to favor performance. Therefore, we will ignore max value in this case.

Table 5-4. CPU core step 4

Scenario	LL min	LL max	L min	L max
Final setting (refer to Table 5-2, Table 5-3)	1	1	3	0
Actual setting to driver	1	1	3	-1

5.3.2 Policy of CPU freq

Decision flow of cpu freq is listed below.

Step 1. Decide min cpu freq of each cluster

Step 2. Decide max cpu freq of each cluster

Step 3. Decide actual value of max cpu freq (if max < min)

Following example demonstrates the decision flow.

5.3.2.1 Requirement

User A needs LL runs @ 800MHz at least and L runs @ 1400MHz at most.

User B needs LL runs @ 600MHz at least and L runs @ 1200MHz at least.

User C needs LL runs @ 700MHz at most and L runs @ 1600MHz at most.

5.3.2.2 Step 1. Decide min cpu freq of each cluster

Choose the maximal value of each cluster to satisfy all scenarios.

Table 5-5. CPU freq step 1

Scenario	LL min	L min
A	800MHz	-1
B	600MHz	1200MHz
C	-1	-1
Final setting	max (800, 600, don't care) = 800MHz	max (don't care, 1200, don't care) = 1200MHz

5.3.2.3 Step 2. Decide max cpu freq of each cluster

Choose the minimal value of each cluster to satisfy all scenarios.

Table 5-6. CPU freq step 2

Scenario	LL max	L max
A	-1	1400MHz
B	-1	-1
C	700MHz	1600MHz
Final setting	min (don't care, don't care, 700) = 700MHz	min (1400, don't care, 1600) = 1400MHz

5.3.2.4 Step 3. Decide actual value of max cpu freq (if max < min)

1. If max >= min, we set both values to driver directly.
2. If max < min, we make the final decision to favor performance. Therefore, we will align max value to min value this case.

Table 5-7. CPU freq step 3

Scenario	LL min	LL max	L min	L max
Final setting (refer toTable 5-5, Table 5-6)	800MHz	700MHz	1200MHz	1400MHz
Actual setting to driver	800MHz	800MHz	1200MHz	1400MHz

5.4 Source Code Organization

Source codes of MTK Power Hal Service are listed in Table 5-8.

Table 5-8. Source code

Module	Path	Description
IPower	vendor\mediatek\proprietary\hardware\interface\power\IPower.hal	Interface definition
Types	vendor\mediatek\proprietary\hardware\interface\power\types.hal	Command and parameter definition
MTK Power Hal Service	vendor\mediatek\proprietary\hardware\power\service	MTK Power Hal implementation
Pre-Defined Scenario table	vendor\mediatek\proprietary\hardware\power\config\mt[xxxx]\scn_tbl	Pre-Defined Scenario table
Customized Scenario table	vendor\mediatek\proprietary\hardware\power\config\mt[xxxx]\cust_hint	Customized Scenario table

6 MTK Power Hal Service

MTK Power Hal service provides Java and C++ interface API that could manage performance of Android devices. The JAVA and C++ interface API are one-one mapping. We only use native API to introduce functionality in this section. Besides, we only introduce API for user-defined scenario.

6.1 Data type

Data types are defined in:

vendor/mediatek/proprietary/hardware/interfaces/power/

We only introduce user scenario related type in this section.

6.1.1 Google-defined scenarios

There are some google-defined scenarios (Table 6-1). Most of them are not configured in MTK Power Hal. So far, we only support power hint launch that could be configured. In order to manage google-defined and mtk pre-defined scenarios, we prefix MTK_POWER_HINT to google-defined scenarios .

Table 6-1. Google-defined scenarios

Enum	Rename for MTK scn table	Scenario	Description
VSYNC	MTK_POWER_HINT_VSYNC	vsync pulse request	Not support
INTERACTION	MTK_POWER_HINT_INTERACTION	interacting with the device	Not support
VIDEO_ENCODE	MTK_POWER_HINT_VIDEO_ENCODE	video encode	Not support
VIDEO_DECODE	MTK_POWER_HINT_VIDEO_DECODE	Video decode	Not support
LOW_POWER	MTK_POWER_HINT_LOW_POWER	low power mode	Not support
SP	MTK_POWER_HINT_SP	sustained performance	Not support
VR	MTK_POWER_HINT_VR	VR mode	Not support
LAUNCHING	MTK_POWER_HINT_LAUNCHING	App launching	
AUDIO_STREAMING	MTK_POWER_HINT_AUDIO_STREAMING		AOSP PowerHAL 1.2. Not support
AUDI_LOW_LATENCY	MTK_POWER_HINT_AUDIO_LOW_LATENCY		AOSP PowerHAL 1.2. Not support
CAMERA_LAUNCH	MTK_POWER_HINT_CAMERA_LAUNCH		AOSP PowerHAL 1.2. Not support
CAMERA_STREAMING	MTK_POWER_HINT_CAMERA_STREAMING		AOSP PowerHAL 1.2. Not support
CAMERA_SHOT	MTK_POWER_HINT_CAMERA_SHOT		AOSP PowerHAL 1.2. Not support

6.1.2 Pre-defined scenarios

There are some pre-defined scenarios (Table 6-2). Most of them are already configured in MTK Power Hal. Users could use these hint to manage performance directly without configuring CPU policy. If CPU policy of these scenarios didn't meet the requirement, user could modify CPU policy to meet.

For example, the CPU policy of scenario MTK_POWER_HINT_PACK_SWITCH are used 4LL at least. But customers could config its cpu policy to 2LL at least.

Table 6-2 Pre-defined scenarios

enum	Scenario	Description
MTK_POWER_HINT_PROCESS_CREATE	Process create	
MTK_POWER_HINT_PACK_SWITCH	Package switch	Package switch
MTK_POWER_HINT_ACT_SWITCH	Activity switch	Activity switch without package switch
MTK_POWER_HINT_GAME_LAUNCH	Game launch	
MTK_POWER_HINT_APP_ROTATE	Window rotate	
MTK_POWER_HINT_APP_TOUCH	Touch boost	
MTK_POWER_HINT_FRAME_UPDATE	Render aware boost	
MTK_POWER_HINT_GAMING	Gaming mode	
MTK_POWER_HINT_GALLERY_BOOST	Gallery Boost	
MTK_POWER_HINT_GALLERY_STEREO_BOOST	Gallery Stereo Boost	
MTK_POWER_HINT_SPORTS	sports mode setting	Setting apply to sports mode
MTK_POWER_HINT_PMS_INSTALL	APK install	

6.1.3 Query command

Thess commands are used to query system or cluster capability.

Table 6-3. query command

enum	Description	Note
CMD_GET_CLUSTER_NUM	Get cpu cluster number	CMD_GET_CLUSTER_NUM
CMD_GET_CLUSTER_CPU_NUM	Get cpu number of specified cluster	
CMD_GET_CLUSTER_CPU_FREQ_MIN	Get minimal cpu freq of specified cluster	
CMD_GET_CLUSTER_CPU_FREQ_MAX	Get maximal cpu freq of specified cluster	CMD_GET_CLUSTER_CPU_FREQ_MAX
CMD_GET_GPU_FREQ_COUNT	GPU freq count	
CMD_GET_FOREGROUND_PID	pid of foreground process	

6.1.4 Config command

These command are used to config scenarios.

Command can be extended in the config fillw which is defined in the following file:

vendor/mediatek/proprietary/hardware/power/config/mt[xxxx]/con_tbl/powercontable.cfg

Table 6-4. config command

enum	Description	Note
CMD_SET_CLUSTER_CPU_CORE_MIN	Minimal cpu core of specified cluster	Not support in ACAO
CMD_SET_CLUSTER_CPU_CORE_MAX	Maximal cpu core of specified cluster	Not support in ACAO
CMD_SET_CLUSTER_CPU_FREQ_MIN	Minimal cpu freq of specified cluster	
CMD_SET_CLUSTER_CPU_FREQ_MAX	Minimal cpu freq of specified cluster	
CMD_SET_GPU_FREQ_MIN	Minimal GPU freq	
CMD_SET_GPU_FREQ_MAX	Maximal GPU freq	
CMD_SET_SCREEN_OFF_STATE	Behavior for screen-off	
CMD_SET_CPUFREQ_HISPEED_FREQ	CPU hispeed freq of DVFS	It is only applicable in interactive governor. Please consult DVFS owner
CMD_SET_CPUFREQ_MIN_SAMPLE_TIME	min sample time of DVFS	It is only applicable in interactive governor. Please consult DVFS owner
CMD_SET_CPUFREQ_ABOVE_HISPEED_DELAY	Above hispeed delay of DVFS	It is only applicable in interactive governor. Please consult DVFS owner
CMD_SET_STUNE_THRESH	Internal only	Please consult scheduler owner
CMD_SET_SCHED_MODE	Internal only	Please consult scheduler owner
CMD_SET_ROOT_BOOST_VALUE	Boost value of root group	Please consult scheduler owner
CMD_SET_TA_BOOST_VALUE	Boost value of top-app group	Please consult scheduler owner
CMD_SET_FG_BOOST_VALUE	Boost value of foreground group	Please consult scheduler owner
CMD_SET_BG_BOOST_VALUE	Boost value of background group	Please consult scheduler owner
CMD_SET_OPP_DDR	Minimal DRAM opp	
CMD_SET_MTK_IDLE_PREFER	Internal only	Please consult scheduler owner
CMD_SET_STUNE_TA_PERFER_IDLE	Top-app tasks prefers idle CPU	
CMD_SET_STUNE_FG_PERFER_IDLE	Foreground tasks prefers idle CPU	
CMD_SET_IO_BOOST_VALUE	Boost value of IO group	Please consult IO driver owner
CMD_SET_SCHED_BOOST	Value 0: turn off Value 1: "all tasks" prefer to put on big core Value 2: foreground & top on big core via CPUSET	Please consult scheduler owner
CMD_SET_SCHED_MIGR_COST	Migration cost (unit: ns)	Please consult scheduler owner

6.1.5 Screen off state

These commands are used to control scenario state while screen off.

Table 6-5. screen off state

enum	Description	Note
SCREEN_OFF_DISABLE	The scenario will be auto-disabled when screen (backlight) is turned off	It's default setting for all scenarios
SCREEN_OFF_ENABLE	The scenario will be keep enabled.	
SCREEN_OFF_WAIT_RESTORE	Scenario will be disabled when screen is turned off. PerfService will enable it again after screen is turned on.	

7 Interface

Since we use power hint to manage performance, we will introduce power hint related API in this section.

7.1 Power Hint API

We introduce power hint api in this section.

7.1.1 Google-Defined scenario

Return<void> **PowerHint(PowerHint hint, int32_t data)**

[Description]

PowerHint() is called to pass google-defined hints on power/performance requirements which may result in adjustment of power/performance parameters of the cpufreq governor and other controls.

[Input parameter]

hint: power hint , refer to Table 6-1. Google-defined scenarios

data: additional information about the hint.

[Note]

1. So far, we only support power hint launch. If you want to use this hint to boost, you could configure the powerscntbl.cfg. Please also refer to 9.1.1.

[Example]

1. PowerHint(PowerHint::VR, 5000); /*Enable VR HINT 5000ms*/
2. PowerHint(PowerHint::VR, 0); /*Disable VR HINT*/

7.1.2 Pre-Defined scenario

Return<void> **mtkPowerHint(MtkPowerHint hint, int32_t data)**

[Description]

mtkPowerHint() is called to pass pre-defined hints on power/performance requirements which may result in adjustment of power/performance parameters of the cpufreq governor and other controls.

[Input parameter]

hint: power hint , refer to Table 6-2

data: additional information about the hint.

[Note]

1. Set data parameters as 0 means “disable power hint”.
2. Set data parameters as Others means “timeout value”. This hint will be disabled after timeout (unit:ms).
3. It’s recommended to add a MTK power hint if your scenario is AOSP module

[Example]

1. mtkPowerHint(MtkPowerHint::APP_ROTATE, 5000); /*Enable ROTATE HINT 5000ms*/
2. mtkPowerHint(MtkPowerHint::APP_ROTATE, 0); /*Disable ROTATE HINT*/
3. mtkPowerHint(MtkPowerHint::APP_ROTATE, (int)MtkHintOp::MTK_HINT_ALWAYS_ENABLE); /*Always Enable HINT*/

7.1.3 Customized scenario

Return<void> **mtkCusPowerHint(MtkCusPowerHint hint, int32_t data)**

[Description]

There are two advantages of mtkCusPowerHint API. **It’s recommended to use mtkCusPowerHint** instead of user scenario API.

1. Hint (handle) is allocated by PowerHAL process. Users don’t need to allocate (register) and free (unregister) handle themselves.
2. Scenario config is defined in header file. PowerHAL process config it in its context. Users don’t need to call several binder calls themselves.

mtkCusPowerHint() is called to pass customized hints on power/performance requirements which may result in adjustment of power/performance parameters of the cpufreq governor and other controls.

[Input parameter]

hint: customized power hint.

data: additional information about the hint.

[Note]

1. Set data parameters as 0 means “disable power hint”.
2. Set data parameters as Others means “timeout value”. This hint will be disabled after timeout (unit:ms).
3. mtkCusPowerHint provides the simple way to customize user scenario. User didn’t need to configure their scenario step by step. User just add their own scenario in the config table (Table 5-8) and use this simple api to adjustment performance.

[Example]

1. mtkCusPowerHint((int32_t) MtkCusPowerHintInternal::MTK_CUS_AUDIO_POWER_DL, 2000);
   /*Enable HINT 2000ms*/
2. mtkCusPowerHint((int32_t) MtkCusPowerHintInternal::MTK_CUS_AUDIO_POWER_DL, 0);
   /*Disable HINT*/
3. mtkCusPowerHint((int32_t) MtkCusPowerHintInternal::MTK_CUS_AUDIO_POWER_DL,
   (int)MtkHintOp::MTK_HINT_ALWAYS_ENABLE);
   /*Always Enable HINT*/

7.2 User Scenario API

We introduce user scenario api that also could let user to set their scenario step by step.

7.2.1 Register user scenario

Return<int32_t> **scnReg**()

[Description]

Register user scenario. It returns the handle. User should use this handle to access other functions.

[Return value]

-1: fail.

Other: handle

[Note]

Register scenario returns fail if MTK Power Hal Service is not ready.

7.2.2 Unregister user scenario

Return<void> **scnUnreg**(int32_t hdl)

[Description]

Unregister user scenario

[Input parameter]

Handle

[Note]

Please unregister the scenario before your process exits, otherwise handles may be leaked.

7.2.3 Enable user scenario

Return<void> **scnEnable**(int32_t hdl, int32_t timeout)

[Description]

Enable user scenario with timeout. MTK Power Hal Service will disable this scenario automatically after timeout value is expired.

[Input parameter]

handle: user handle

timeout: unit is ms.

[Note]

1. Set data parameters as 0 means “disable power hint”.
2. Set data parameters as Others means “timeout value”. This hint will be disabled after timeout (unit:ms).

7.2.4 Disable user scenario

Return<void> **scnDisable**(int32_t hdl)

[Description]

Disable user scenario.

[Input parameter]

handle: user handle

[Note]

It is recommended to calls disable API no matter you use timeout API or not.

7.2.5 Config user scenario

Return<void> **scnConfig**(int32_t hdl, MtkPowerCmd cmd, int32_t param1, int32_t param2, int32_t param3, int32_t param4

[Description]

Config user scenario.

[Input parameter]

handle: user handle

others: refer to following table (Table 7-1)

Table 7-1. config command

Command	Param_1	Param_2	Note
CMD_SET_CLUSTER_CPU_CORE_MIN	Cluster id	Min cpu core	Not support in ACAO
CMD_SET_CLUSTER_CPU_CORE_MAX	Cluster id	Max cpu core	Not support in ACAO
CMD_SET_CLUSTER_CPU_FREQ_MIN	Cluster id	Min cpu freq	
CMD_SET_CLUSTER_CPU_FREQ_MAX	Cluster id	Max cpu freq	
CMD_SET_GPU_FREQ_MIN	GPU freq index		0 is GPU freq opp 0
CMD_SET_GPU_FREQ_MAX	GPU freq index		0 is GPU freq opp 0
CMD_SET_SCREEN_OFF_STATE	Screen off state		Please refer to 6.1.5
CMD_SET_CPUFREQ_HISPEED_FREQ	Hispeed freq		It is only applicable in interactive governor. Please consult DVFS owner
CMD_SET_CPUFREQ_MIN_SAMPLE_TIME	Min sample time		It is only applicable in

Command	Param_1	Param_2	Note
			interactive governor. Please consult DVFS owner
CMD_SET_CPUFREQ_ABOVE_HISPEED_DELAY	hispeed delay		It is only applicable in interactive governor. Please consult DVFS owner
CMD_SET_STUNE_THRESH	Internal only		
CMD_SET_SCHED_MODE	Internal only		
CMD_SET_ROOT_BOOST_VALUE	Boost value		
CMD_SET_TA_BOOST_VALUE	Boost value		
CMD_SET_FG_BOOST_VALUE	Boost value		Please consult scheduler owner. It is only applicable to foreground task.
CMD_SET_BG_BOOST_VALUE	Boost value		
CMD_SET_IO_BOOST_VALUE	Boost value		
CMD_SET_OPP_DDR	DRAM opp index		0 means the highest freq.
CMD_SET_MTK_IDLE_PREFER	Internal only		
CMD_SET_STUNE_TA_PERFER_IDLE	1: enable 0: disable		
CMD_SET_STUNE_FG_PERFER_IDLE	1: enable 0: disable		
CMD_SET_SCHED_BOOST	Value 0: turn off Value 1: "all tasks" prefer to put on big core Value 2: foreground & top on big core via CPUSET		
CMD_SET_SCHED_MIGR_COST	Migration cost		Unit: ns
CMD_SET_FBT_FLOOR_BOUND	FPSGO variance control		Please consult FPSGO owner
CMD_SET_FBT_KMIN	FPSGO variance control		Please consult FPSGO owner
CMD_SET_FBT_BHR_OPP	CPU DVFS headroom for FPSGO		Please consult FPSGO owner
CMD_SET_DISP_IDLE_TIME	Threshold for enter display idle mode		Unit: ms
CMD_SET_CM_MGR	Internal only		

7.3 Query System Info API

User should use these API to query the system info.

7.3.1 Get system capability

Return<int32_t> **querySysInfo**(MtkQueryCmd cmd, int32_t param)

[Description]

Get system capability

[Input parameter]

cmd: refer to following table (Table 6-2)

[Return value]

Refer to following table (Table 6-2)

8 How to use MTK Power Hal Service

This section describes how to use MTK Power Hal Service. It provides sample codes of both native and JAVA layer.

8.1 Native and Java layer

In both native and java layer, the use flow and api of MTK Power Hal Service are the same.

8.1.1 Use flow

User should follow this flow to use MTK Power Hal API.

8.1.1.1 Use flow of MTK Power Hint

1. Get the MTK Power Hal Service
2. Use MTK Power Hint to enable or disable

8.1.1.2 Use flow of MTK Cus Power Hint

1. Get the MTK Power Hal Service
2. User add their own scenario in the table (Table 5-8)
3. Use MTK Cus Power Hint to enable or disable

8.1.1.3 Use flow of registering scenario dynamically

1. Get the MTK Power Hal Service
2. Register user own scenario
3. Enable user scenario
4. Run user behavior
5. Disable the scenario

8.1.2 Normal API

Config API can control all functionality. If you want enable big core, please use these API.

[Procedure]

- mtkPowerHint: enable or disable mtk power hint.

- mtkCusPowerHint: enable or disable mtk customized power hint.
- scnReg: register user scenario and get the handle.
- scnUnreg: unregister user scenario.
- scnEnable: enable the scenario.
- scnDisable: remember to disable the scenario.
- scnConfig: config the scenario. Set the cmd which you need.

8.2 Sample code

Here are sample of both native and Java layer.

8.2.1 Sample code of native layer

8.2.1.1 Make file

Android.mk

```
LOCAL_SHARED_LIBRARIES := \
    libhidlbase \
    libhidltransport \
    libhwbinder \
    android.hardware.power@1.0 \
    vendor.mediatek.hardware.power@1.1_vendor
```

8.2.1.2 Source code

```
#include <vendor/mediatek/hardware/power/2.0/IPower.h>
#include <vendor/mediatek/hardware/power/2.0/types.h>
using namespace vendor::mediatek::hardware::power::V2_0;
```

Get Power Hal Service

```
android::sp<IPower> gPowerHal;
gPowerHal = IPower::getService();
```

Usage: mtkPowerHint

gPowerHal->mtkPowerHint(MtkPowerHint::MTK_POWER_HINT_APP_TOUCH, 3000); // enable 3000ms at most

// Your own operations …

gPowerHal->mtkPowerHint(MtkPowerHint::MTK_POWER_HINT_APP_TOUCH, 0); // disable power hint

Usage: mtkCusPowerHint

gPowerHal->mtkCusPowerHint(MtkCusPowerHint::MTK_CUS_CAMERA_PREVIEW, 3000); // enable 3000ms at most

// Your own operations …

gPowerHal->mtkCusPowerHint(MtkCusPowerHint::MTK_CUS_CAMERA_PREVIEW, 0); // disable power hint

8.2.2 Sample code to config scenario

Source code

```
#include <vendor/mediatek/hardware/power/2.0/IPower.h>
#include <vendor/mediatek/hardware/power/2.0/types.h>
using namespace vendor::mediatek::hardware::power::V2_0;
```

Get Power Hal Service

```
android::sp<IPower> gPowerHal;
gPowerHal = IPower::getService();
```

Usage: at least 4 cores, fix at 1469000KHz

```
handle = gPowerHal->scnReg();          // get handle

gPowerHal->ScnConfig(handle, CMD_SET_CPU_CORE_MIN, 4, 0, 0, 0);  // 4 LL  (it only set cpu number of cluster 0)
gPowerHal->ScnConfig(handle, CMD_SET_CPU_FREQ_MIN, 1066000, 0, 0, 0);  // cpu freq lower bound: 1066000
gPowerHal->ScnConfig(handle, CMD_SET_CPU_FREQ_MAX, 1066000, 0, 0, 0);  // cpu freq upper bound: 1066000
```

gPowerHal->ScnEnable(handle);

user's behavior

gPowerHal->ScnDisable(handle);

Usage: enable 3LL + 1L, L core at least 1872000KHz

handle = gPowerHal->ScnReg (); // get handle

gPowerHal->ScnConfig(handle, **CMD_SET_CLUSTER_CPU_CORE_MIN**, 0, 3, 0, 0); // cluster 0, 3 ore => 3LL

gPowerHal->ScnConfig(handle, **CMD_SET_CLUSTER_CPU_CORE_MIN**, 1, 1, 0, 0); // cluster 1, 1 core => 1L

gPowerHal->ScnConfig(handle, **CMD_SET_CLUSTER_CPU_FREQ_MIN**, 1, 1872000, 0, 0); // L 1.872GHz at least

gPowerHal->ScnEnable(handle);

user's behavior

gPowerHal->ScnDisable(handle);

Usage: set each cluster to at least 4 cores

handle = gPowerHal->ScnReg (); // get handle

max_idx = gPowerHal-> querySysInfo (**MtkQueryCmd::CMD_GET_CLUSTER_NUM, 0**);

gPowerHal->ScnConfig(handle, **CMD_SET_CLUSTER_CPU_CORE_MIN**, 0, 4, 0, 0); // set cluster 0 to max value

.....

gPowerHal->ScnConfig(handle, **CMD_SET_CLUSTER_CPU_CORE_MIN**, max_idx, 4, 0, 0); // set cluster max_idx to max value

gPowerHal->ScnEnable(handle);

user's behavior

gPowerHal->ScnDisable(handle);

8.2.2.1 Use case 1: how to improve performance

If user want to improve performance, user could force more CPU online and runs at higher frequency. Following sample codes force 4L runs at 2GHz or higher freq.

// Please refer to 8.2.1.2 for loading API

handle = gPowerHal->ScnReg (); // get handle

gPowerHal->ScnConfig(handle, **CMD_SET_CLUSTER_CPU_CORE_MIN**, 1, 4, 0, 0); // min of cluster 1 = 4 core => 4L

gPowerHal->ScnConfig(handle, **CMD_SET_CLUSTER_CPU_FREQ_MIN**, 1, 2000000, 0, 0); // L runs @ 2GHz at least

gPowerHal->ScnEnable(handle); // After enable the scenario, HPS enable 4L online and DVFS set cpu freq to 2GHz at least

user's behavior

gPowerHal->ScnDisable(handle);

8.2.2.2 Use case 2: how to save power consumption

If user want to improve performance, user could force less CPU online and runs at lower frequency. Following sample codes force all cpu of L cluster offline.

// Please refer to 8.2.1.2 for loading API

handle = gPowerHal->ScnReg (); // get handle

gPowerHal->ScnConfig(handle, **CMD_SET_CLUSTER_CPU_CORE_MAX**, 1, 0, 0, 0); // max of cluster 1 = 0

gPowerHal->ScnEnable(handle); // After enable the scenario, HPS never enable cpu of cluster 1

user's behavior

gPowerHal->ScnDisable(handle);

8.2.3 Sample code of JAVA layer

We use ActivityStack.java to describe how to use PerfService in JAVA layer.

8.2.3.1 Make file

Android.mk

```
LOCAL_JAVA_LIBRARIES := \
vendor.mediatek.hardware.power-V2.0-java \
LOCAL_STATIC_JAVA_LIBRARIES := \
vendor.mediatek.hardware.power-V2.0-java-static \
```

8.2.3.2 Source code

Import package

```
import vendor.mediatek.hardware.power.V2_0*; /// M: Power Hal Service


class MyClass {

/// M: PerfBoost include @ {
private IPower mPowerProxy = null;
/// @}
```

Use mtkPowerHint

```
public MyClass() {
mPowerProxy = IPower.getService();
...
```

```
}
```

Enable scenario

```
public void MyBoostStart() {
        if(mPowerProxy != null && mPowerProxy == -1) {
            mPowerProxy.mtkPowerHint(MtkPowerHint.MTK_POWER_HINT_PACK_SWITCH, 2000);
        }
        ....
}
```

Disable scenario

```
Public void MyBoostStop() {
        if(mPowerProxy != null && mPowerProxy == -1) {
            mPowerProxy.mtkPowerHint(MtkPowerHint.MTK_POWER_HINT_PACK_SWITCH, 0);
        }
        ....
    }
……
}
```

8.2.4 Sample code of JAVA layer for using Power Hal as System Service

We use ActivityStack.java to describe how to use PerfService in JAVA layer.

8.2.4.1 Source code

Import package

```
import com.mediatek.powerhalmgr.PowerHalMgr; /// M: Power Hal Service

class MyClass {

/// M: PerfBoost include @ {
private PowerHalMgr mPowerProxy = new PoweHalMgr();
```

```
/// @}

Use scnReg
public MyClass() {
handle = mPowerProxy.scnReg();
...
}

Config scenario
public void MyBoostConfig() {
    if(mPowerProxy != null && mPowerProxy == -1) {
        /* min of cluster 1 = 4 core => 4L */
        mPowerProxy.scnConfig(handle, CMD_SET_CLUSTER_CPU_CORE_MIN, 1, 4, 0, 0);
        /* L runs @ 2GHz at least */
        mPowerProxy.scnConfig(handle, CMD_SET_CLUSTER_CPU_FREQ_MIN, 1, 2000000, 0, 0);

    }
    ....
}

Enable scenario
public void MyBoostStart() {
    if(mPowerProxy != null && mPowerProxy == -1) {
        mPowerProxy.scnEnable(handle, 2000);
    }
    ....
}
```

Disable scenario

```
Public void MyBoostStop() {
      if(mPowerProxy != null && mPowerProxy == -1) {
          mPowerProxy.scnDisable(handle, 0);
      }
      ....
  }
……
}
```

8.2.4.2 Code Path

Table 8-1. Source code

Module	Path	Description
Powerhalmgr service	vendor\mediatek\proprietary\frameworks\base\services\powerhalservice\	Implement the service
Powerhalmgr proxy	vendor\mediatek\proprietary\frameworks\base\core\java\com\mediatek\powerhalmgr	Implement the proxy
Powerhalmgr proxy	frameworks\base\core\java\com\mediatek\powerhalmgr	Interface

8.2.5 Sample code of JAVA layer for using Power Hal as System Service (MTK customized power hint)

We use ActivityStack.java to describe how to use PerfService in JAVA layer.

8.2.5.1 Source code

Import package

```
import com.mediatek.powerhalmgr.PowerHalMgr; /// M: Power Hal Service
import  com.mediatek.powerhalmgr.PowerHalMgrFactory;
```

Use mtkCusPowerHint

```
private PowerHalMgr mPowerHalService = PowerHalMgrFactory.getInstance().makePowerHalMgr();
mPowerHalService.mtkCusPowerHint((int)MtkCusPowerHintInternal::MTK_CUS_AUDIO_POWER_DL, 3000);
/*enable hint 3000ms at most*/
```

9 Customization

MTK Power Hal Service provides some customization mechanisms. Customer can config their own scenarios. We will introduce these mechanisms in this section.

9.1 Config scenario

There are some pre-defined scenarios provided by MTK. Users could control scenarios and config cpu policy of theses scenarios themselves.

9.1.1 Config google-defined scenario

Config file

vendor/mediatek/proprietary/hardware/power/config/mt[xxxx]/scn_tbl/powerscntbl.cfg

Format

command, scenario, setting

command: refer to Table 6-4. config command.

Example

CMD_SET_CLUSTER_CPU_CORE_MIN**, MTK_POWER_HINT_**LAUNCH, 0, 4

CMD_SET_CLUSTER_CPU_FREQ_MIN, **MTK_POWER_HINT_**LAUNCH, 0, 1638000

1. App Launch will force 4 LL core online and set cpu freq to 1.638GHz at least.

9.1.2 Config pre-defined scenario

Config file

vendor/mediatek/proprietary/hardware/power/config/mt[xxxx]/scn_tbl/powerscntbl.cfg

Format

command, scenario, setting

command: refer to Table 6-4. config command

Example

CMD_SET_CLUSTER_CPU_CORE_MIN, MTK_POWER_HINT_APP_TOUCH, 0, 3

CMD_SET_CLUSTER_CPU_FREQ_MIN, MTK_POWER_HINT_APP_TOUCH, 0, 1066000

CMD_SET_CLUSTER_CPU_CORE_MIN, *MTK_POWER_HINT*_PACK_SWITCH, 0, 0

CMD_SET_CLUSTER_CPU_FREQ_MIN, *MTK_POWER_HINT*_PACK_SWITCH, 0, 1638000

CMD_SET_CLUSTER_CPU_CORE_MIN, *MTK_POWER_HINT*_PACK_SWITCH, 1, 4

CMD_SET_CLUSTER_CPU_FREQ_MIN, *MTK_POWER_HINT*_PACK_SWITCH, 1, 2340000

2. touch will force 3 LL core online and set cpu freq to 1.066GHz at least.
3. Package switch force 4 L core online and set cpu freq to 2.34GHz

9.2 Parameter config table

Programmers can define / add their own parameters in this config table. Then user program can use MTK Power Hal Service API to control these parameters. In the other word, programmer can use this table to extend config command. (Please also refer to Table 7-1. config command)

Config file

vendor/mediatek/proprietary/hardware/power/config/mt[xxxx]/con_tbl/powercontable.cfg

Format

cmd, cmd id, file entry, compare, max value, min value, normal value, sports value

Table 9-1. format description

Field	Description	Note
cmd	Command to access this parameter. This command can be used in perfservscntbl.txt and perf_whitelist_cfg.xml	It should be sync with perfservice_types.h
cmd id	Command id to access this parameter. This id can be used in PerfServiceNative_userRegScnConfig	It should be sync with perfservice_types.h
file entry	PerfService uses file operations to control parameters. Therefore, kernel driver should provide sysfs, procfs or debugfs.	Programming may also need to add file permission in init.mtxxxx.rc
compare	PerfService should know hwo to compare this parameter. For example, more is better or less is better.	less: PerfService chooses smaller value more: PerfService chooses bigger value
max	Maximal value of this parameter.	
min	Minimal value of this parameter.	
normal value	Default value of this entry.	Not necessary. If normal value is set, PerfService set this value in init function.
Sports value	Setting of sports mode.	Not necessary. PerfService set this value if benchmark app is in foreground.

Example

CMD_SET_HPS_RUSH_BOOST, 20, /proc/hps/rush_boost_enabled, more, 1, -1, 1, 1

1. Add CMD_SET_HPS_RUSH_BOOST in powercontable.cfg. This command id is 20.
2. These command and id should be sync with types.hal (Table 5-8). It the other word, programmer should not use command and id which is already defined in types.hal (Table 5-8).
3. Then we can add following statement in powercontable.cfg (or perf_whitelist_cfg.xml). It means “Gaming” will set 0 to */proc/hps/rush_boost_enabled*.
   CMD_SET_HPS_RUSH_BOOST*, MTK_POWER_HINT_GAMING, 0*

9.3 White list

Config file

vendor/mediatek/proprietary/hardware/power/config/mt[xxxx]/app_list/power_whitelist_cfg.xml

Purpose

Use white list to set cpu policy of specific package. If foreground application is listed in whitelist, MTK Power Hal Service will apply this policy.

Format

```
<Package name="PACKAGE_NAME1">
    <Activity name="ACTIVITY_NAME1">
    <COMMAND1 Param1="value1"/>
    <COMMAND2 Param1="value1" Param2=”value2”/>
    ……
    </Activity>
    <Activity name="ACTIVITY_NAME2">
    ……
    </Activity>
    <Activity name="Common">
    ……
    </Activity>
</Package>
<Package name="PACKAGE_NAME2">
……      </Package>
```

Example

```
<Package name="com.imangi.templerun2">
        <Activity name="Common">
        <CMD_SET_CLUSTER_CPU_CORE_MIN Param1="0" Param2=”3”/>
        </Activity>
</Package>
```

It’s means all activities of templerun2 needs 3LL online at least.

Note: Common means all activities apply this policy.

9.4 App launch customization in white list

Config file

vendor/mediatek/proprietary/hardware/power/config/mt[xxxx]/app_list/perf_whitelist_cfg.xml

Command

Use CMD_SET_PACK_BOOST_TIMEOUT to decide boost duration. MTK Power Hal Service will boost system if app is first launch (with process create)

Example

If 王者榮耀 is launched with process create, Power Hal Service will apply boost policy "MTK_POWER_HINT_PRCOCESS_CREATE" 25 sec.

```
<Package name="com.tencent.tmgp.sgame">
  <Activity name="Common">
    <CMD_SET_PACK_BOOST_TIMEOUT Param1="25"/>
    <CMD_SET_FSTB_FPS Param1="60" Param2="30"/>
  </Activity>
</Package>
```

10 Tuning Guideline

10.1 Command guideline

Table 10-1. config command

Command	Effect	Purpose	Note
CMD_SET_CLUSTER_CPU_FREQ_MIN	Set floor of cpu freq	Performance	
CMD_SET_CLUSTER_CPU_FREQ_MAX	Set ceiling of cpu freq	Low power	
CMD_SET_GPU_FREQ_MIN	Set floor of GPU freq	Performance	
CMD_SET_GPU_FREQ_MAX	Set celing of GPU freq	Low power	
CMD_SET_ROOT_BOOST_VALUE	Set boost value of root	Performance	Scheduler uses more computing power for root
CMD_SET_TA_BOOST_VALUE	Set boost value of top-app	Performance	Scheduler uses more computing power for top-app
CMD_SET_FG_BOOST_VALUE	Set boost value of foreground	Performance	Scheduler uses more computing power for foreground
CMD_SET_BG_BOOST_VALUE	Set boost value of background	Performance	Scheduler uses more computing power for background
CMD_SET_OPP_DDR	Set floor of dram freq	Performance	
CMD_SET_STUNE_TA_PERFER_IDLE	Enable “prefer idle” of top-app	Performance	Tend to use idle CPU for top-app
CMD_SET_STUNE_FG_PERFER_IDLE	Enable “prefer idle” of foreground	Performance	Tend to use idle CPU for foreground
CMD_SET_IO_BOOST_VALUE	Set boost value of IO tasks	Performance	
CMD_SET_SCHED_BOOST	Control schedule boost	Performance	Tend to use big core
CMD_SET_SCHED_MIGR_COST	Set migration cost	N/A	

10.2 Use case

10.2.1 Case 1: If limit CPU or GPU freq can improve low power

Use CMD_SET_CLUSTER_CPU_FREQ_MAX / CMD_SET_GPU_FREQ_MAX to limit CPU / GPU freq.

10.2.2 Case 2: If force higher GPU freq can improve performance

Use CMD_SET_GPU_FREQ_MIN to force GPU runs at higher freq.

10.2.3 Case 3: If force higher CPU freq can improve performance

We recommend that use boost value instead of CPU freq to get more computing power. EAS will control CPU freqeucy according to boost value and task loading.

Please cat /proc/[pid]/cgroup to check which cgroup should be boosted. For example,

If we want to give launcher3 more computing power, we can set boost value of top-app.

 # ps -ef | grep launcher3

u0_a18 1708 508 1 00:00:21 ? 00:00:01 com.android.launcher3

 # cat /proc/1708/cgroup

4:cpuset:/top-app

3:cpu:/

2:schedtune:/top-app

1:cpuacct:/uid_10018/pid_1708

10.2.4 Case 4: Tasks runs at small core.

1. Use boost value to boost scheduler. EAS may tend to use big core.
2. Enable schedule boost.
   2.1 If set schedule boost to 1, scheduler prefer put all task on big core.
   2.2 If set schedule boost to 2, scheduler prefer pull top-app and foreground on big core.

10.2.5 Case 5: There are still idle cpu but some tasks are in runnable state.

1. Enable prefer idle.
   Scheduler will tend to put runnable task on idle CPU.
2. Decrease migration cost.
   Scheduler will migratre task to idle CPU if CPU idle time is more than this threshold. If we decrease migration cost, scheduler will tend to migration task to idle CPU.

11 Frequently Asked Questions

11.1 Policy Related Questions

11.1.1 How to use command CMD_SET_CLUSTER_CPU_CORE_MIN, CMD_SET_CLUSTER_CPU_CORE_MAX, CMD_SET_CLUSTER_CPU_FREQ_MIN and CMD_SET_CLUSTER_CPU_FREQ_MAX?

They support a parameter to specify the cluster id. Please also refer to 7.2.5

11.1.2 If max core value is less than min core value, what happens?

MTK Power Hal Service have corresponding handler to make the final decision to favor performance. Therefore, it ignores max core value temporarily. Please also refer to 5.3.1

11.2 Synchronization Related Questions

11.2.1 1. All power hal service APIs are thread-safe?

All mtk power hal service functionality are protected by mutex. Therefore all mtk power hal service API are thread-safe.