

目次

- CPUの動作クロックを設定する 1
 - コマンドラインから設定する 1
 - 起動スクリプトの設定ファイルで指定する 3
- 設定の例 4
 - CASE1 : 低消費電力設定 4
 - CASE2 : 最大パフォーマンス設定 4

CPUの動作クロックを設定する

MA-E3xxシリーズのCPUは、動的にCPUの動作クロック・駆動電圧を変更することが可能になっています。
デフォルトでは、

- 動作クロックが 300MHz~1GHz の間でソフトウェアの負荷により自動で変更
- CPU負荷が 95% 以上の時に動作クロックを動的に切り替える (ondemand governor)¹⁾

という設定にしてあります。

現在の動作クロックなどの情報は、cpufreq-info コマンドにより確認することが可能です。

```
root@plum:~# cpufreq-info
cpufrequtils 008: cpufreq-info (C) Dominik Brodowski 2004-2009
Report errors and bugs to cpufreq@vger.kernel.org, please.
analyzing CPU 0:
  driver: generic_cpu0
  CPUs which run at the same hardware frequency: 0
  CPUs which need to have their frequency coordinated by software: 0
  maximum transition latency: 300 us.
  hardware limits: 300 MHz - 1000 MHz
  available frequency steps: 300 MHz, 600 MHz, 800 MHz, 1000 MHz
  available cpufreq governors: conservative, userspace, powersave, ondemand,
performance
  current policy: frequency should be within 300 MHz and 1000 MHz.
                    The governor "ondemand" may decide which speed to use
                    within this range.
  current CPU frequency is 1000 MHz (asserted by call to hardware).
  cpufreq stats: 300 MHz:nan%, 600 MHz:nan%, 800 MHz:nan%, 1000 MHz:nan%
```

コマンドラインから設定する

cpufreq-set コマンドで設定します。

```
root@plum:~# cpufreq-set -h
cpufrequtils 008: cpufreq-set (C) Dominik Brodowski 2004-2009
Report errors and bugs to cpufreq@vger.kernel.org, please.
Usage: cpufreq-set [options]
Options:
  -c CPU, --cpu CPU          number of CPU where cpufreq settings shall be
modified
  -d FREQ, --min FREQ        new minimum CPU frequency the governor may select
  -u FREQ, --max FREQ        new maximum CPU frequency the governor may select
  -g GOV, --governor GOV    new cpufreq governor
  -f FREQ, --freq FREQ       specific frequency to be set. Requires userspace
governor to be available and loaded
```

```
-r, --related      Switches all hardware-related CPUs
-h, --help          Prints out this screen

Notes:
1. Omitting the -c or --cpu argument is equivalent to setting it to zero
2. The -f FREQ, --freq FREQ parameter cannot be combined with any other
   parameter
   except the -c CPU, --cpu CPU parameter
3. FREQuencies can be passed in Hz, kHz (default), MHz, GHz, or THz
   by postfixing the value with the wanted unit name, without any space
   (FREQuency in kHz =^ Hz * 0.001 =^ MHz * 1000 =^ GHz * 1000000).
root@plum:~#
```

それぞれのオプションの内容は下表のとおりとなります。
(指定する意味の無いオプションは省いています)。

option		内容	設定可能な値
short	long		
-d	--min	CPUの最低動作クロック	300M, 600M, 800M, 1000M(1G)
-u	--max	CPUの最高動作クロック	300M, 600M, 800M, 1000M(1G)
-g	--governor	CPUFreqガバナー 指定	conservative, userspace, powersave, ondemand, performance
-f	--freq	CPUの動作クロック固定指定 ²⁾	300M, 600M, 800M, 1000M(1G)

CPUFreqガバナー とは

CPUFreqガバナーとはCPUfreqの設定のことです。
CPUFreqガバナーを指定することでCPUの動作クロックの調整を行います。

ondemandガバナー

ondemandガバナーは、負荷に応じてCPUの動作クロックを最低クロックから最大クロックの間で変化させます。
後述するconservativeガバナーと比較し、ondemandガバナーは積極的に動作クロックを変化させます。
短時間に低負荷な状況と高負荷な状況が頻繁に切り替わる場合、省電力効果は落ちます。

conservativeガバナー

conservativeガバナーは、負荷に応じてCPUの動作クロックを最低クロックから最大クロックの間で変化させます。
動作クロックが変化する範囲はondemandガバナーと同等ですが、変化の程度がondemandガバナーよりも緩やかです。

powersaveガバナー

powersaveガバナーは、CPUの動作クロックを最低クロック³⁾で固定します。
最も省電力を期待できますがCPUのパフォーマンスは一番低くなります。
また、高負荷な状況が続くと逆にシステム全体の消費電力が上がってしまう可能性があります。

performanceガバナー

performanceガバナーは、CPUの動作クロックを最高クロック⁴⁾で固定します。
高負荷な状況が続きパフォーマンスを優先したい時に指定するとよいと思われます。

userspaceガバナー

ユーザーランドから設定を指定できるガバナーです。

起動スクリプトの設定ファイルで指定する

デフォルトでは用意されていませんが/etc/default/cpufrequtils ファイルに設定をすることで、cpufrequtils の起動スクリプトにより設定が行われます。
実運用ではこちらの方法で設定するほうが良いと思います。

下記は、cpufrequtils起動スクリプトに記載されているコメントになります。

```
# Which governor to use. Must be one of the governors listed in:
#   cat /sys/devices/system/cpu/cpu0/cpufreq/scaling_available_governors
#
# and which limits to set. Both MIN_SPEED and MAX_SPEED must be values
# listed in:
#   cat /sys/devices/system/cpu/cpu0/cpufreq/scaling_available_frequencies
# a value of 0 for any of the two variables will disabling the use of
# that limit variable.
#
# WARNING: the correct kernel module must already be loaded or compiled in.
#
# Set ENABLE to "true" to let the script run at boot time.
#
# eg:   ENABLE="true"
#       GOVERNOR="ondemand"
#       MAX_SPEED=1000
#       MIN_SPEED=500
```

設定の例

用途によって、

- 消費電力を極力下げたいので、クロックを最低(300MHz)に固定したい
- 常に最大のパフォーマンスで動作させたいので、クロックを最高(1GHz)に固定したい
- クロックを更新するポリシーを変更したい

といったケースがあると思いますので、いくつかのケースの設定方法を紹介します。

CASE1 : 低消費電力設定

太陽電池+バッテリー駆動のような、極力消費電力を下げたい、という場合の設定です。

/etc/default/cpufrequtils を下記のように編集して保存します。

cpufrequtils

```
ENABLE="true"
GOVERNOR="powersave"
MAX_SPEED=0
MIN_SPEED=0
```

cpufrequtilsサービスを再起動します。

```
root@plum:~# service cpufrequtils restart
* CPUFreq Utilities: Setting powersave CPUFreq governor...
* CPU0...
[ OK ]
root@plum:~#
```

CASE2 : 最大パフォーマンス設定

CPUの動作クロックが負荷の増加により高く設定されますが、変更されるまでの間はパフォーマンス低くなり、
例えばルーターとして使用している場合、パケットが落ちてしまう場合があります。
それを避けるため、常に最高クロックで動作させる設定です。

/etc/default/cpufrequtils を下記のように編集して保存します。

cpufrequtils

```
ENABLE="true"
GOVERNOR="performance"
```

```
MAX_SPEED=0  
MIN_SPEED=0
```

cpufrequtilsサービスを再起動します。

```
root@plum:~# service cpufrequtils restart  
* CPUFreq Utilities: Setting performance CPUFreq governor...  
* CPU0...  
[ OK ]  
root@plum:~#
```

1)

[CPU frequency and voltage scaling code in the Linux kernel](#)

2)

CPUfreqガバナーが“userspace”になっている必要があります。

3)

300MHz

4)

1GHz

From:

<https://centurysys.jp/> - **MA-X/MA-S/MA-E/IP-K Developers' Wiki**

Permanent link:

https://centurysys.jp/doku.php?id=mae3xx_tips:cpufreq:start

Last update: **2018/12/29 14:33**