

TP3 – Annexe

Utilisation du script Power Analysis Tools

Table des matières

I Introduction.....	1
II Dépendances.....	2
III Utilisation pratique.....	2
III.1 Enregistrement d'un fichier de log.....	2
III.1.a Avec s-tui.....	2
III.1.b Avec Intel® Power Gadget ou MX Power Gadget.....	3
III.2 Analyse (interactive) de l'enregistrement.....	4
III.2.a Utilisation (réglages par défaut).....	4
III.2.b Recours possibles en cas de variation poste à poste.....	5
III.2.c Remarque à propos de l'utilisation du zoom.....	6

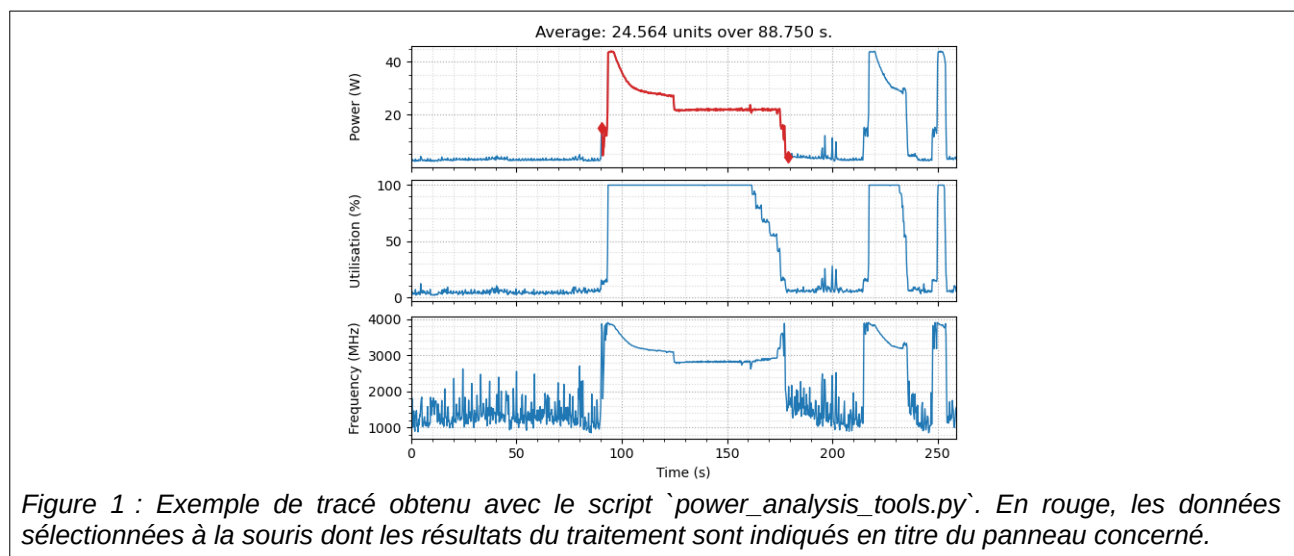
I Introduction

Un script ``power_analysis_tools.py`` a été écrit en Python pour vous aider à afficher et analyser les fichiers de log enregistrés par les logiciels de monitoring `s-tui`¹ (sous Linux), Intel® Power Gadget² (sous Windows ou MacOS, pour les processeurs Intel® uniquement) ou encore MX Power Gadget³ (pour les ordinateurs Apple récents équipés d'un processeur ARM®).

Le script permet notamment d'afficher sur une même figure :

1. une courbe de puissance consommée (par défaut par le processeur) ;
2. une courbe de taux d'utilisation (par défaut moyenné sur l'ensemble des cœurs) ;
3. une courbe de fréquence d'horloge (par défaut moyennée sur l'ensemble des cœurs).

À noter que les trois panneaux partageant le même axe des abscisses : zoomer sur l'un des panneaux mettra également à jour l'axe des abscisses des deux autres panneaux.



1 <https://amanusk.github.io/s-tui/>

2 <https://www.intel.com/content/www/us/en/developer/articles/tool/power-gadget.html>

3 <https://www.seense.com/menubarstats/mxpg/>

Le script permet en outre d'évaluer la moyenne des échantillons d'une des courbes compris entre deux points (sélectionnés chacun de manière interactive par un clic de souris). La valeur de la moyenne et de l'intervalle temporel concerné sont affichés dans le terminal ainsi qu'en titre du panneau pertinent. Les données sélectionnées sont par ailleurs indiquées en rouge.

De la sorte, il est par exemple possible d'estimer la puissance moyenne consommée lors de l'exécution d'une tâche où ladite puissance varierait significativement au cours de la tâche. Cela peut notamment se produire avec certains modèles de processeurs dont l'enveloppe thermique assez limitée nécessite parfois de réduire à la volée la fréquence d'horloge afin de faire baisser la puissance consommée et la dissipation thermique associée. Un exemple d'un tel phénomène est présenté à la Figure 1.

II Dépendances

Le script nécessite que les logiciels ou modules Python suivants :

- python (≥ 3.6) ;
- numpy (≥ 1.19) ;
- matplotlib (≥ 3.3) ;
- argparse (≥ 1.1).

Les numéros de versions indiqués entre parenthèses sont les versions les plus anciennes accessibles à l'auteur et avec lesquelles le script a été testé avec succès.

III Utilisation pratique

III.1 Enregistrement d'un fichier de log

Quel que soit le logiciel utilisé, notez que l'enregistrement de données de consommation de puissance requièrent de disposer des droits d'administration de la machine sur laquelle l'enregistrement est effectué, à moins d'avoir procédé à une configuration particulière comme pour les machines de l'ENSEIRB-MATMECA.

III.1.a Avec s-tui

Dans un terminal, déplacez-vous jusqu'au répertoire où se situe le script `power_analysis_tools.py` puis lancez le logiciel `s-tui` une première fois à l'aide de la commande :

```
'''
```

```
sudo s-tui
```

```
'''
```

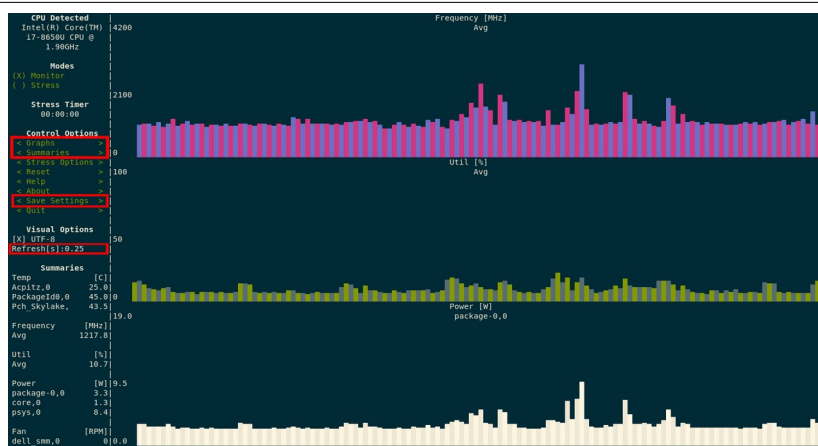


Figure 2 : Capture d'écran de `s-tui` en cours d'exécution. Les éléments encadrés en rouge sur la gauche permettent (de haut en bas) : de modifier les courbes et les éléments textuels affichés, d'enregistrer les réglages actuels (dans un fichier de configuration qui sera rechargé lors des exécutions ultérieures) et enfin de modifier l'intervalle de rafraîchissement des données.

Modifiez la configuration du logiciel de manière à obtenir quelque chose de similaire à la Figure 2. Pour cela, vous pouvez utiliser les onglets < Graphs > et < Summaries > afin de modifier les éléments affichés dans le terminal, ainsi que le champ `Refresh[s]` : pour modifier l'intervalle de rafraîchissement des données. À noter que pour ce dernier élément, une valeur trop faible peut entraîner une charge significative du processeur, perturbant potentiellement les mesures que vous souhaitez effectuer. Veillez donc à choisir une valeur raisonnable. Après quelques essais, l'auteur du présent document a par exemple retenu une valeur de 0,2 s lors des mesures sur sa propre machine.

Lorsque le résultat vous convient, enregistrez la configuration (< Save Settings >) et quittez le logiciel.

Vous pourrez par la suite relancer `s-tui` en lui demandant cette fois-ci également d'enregistrer les données (affichées mais aussi celles cachées) dans un fichier au format CSV (*Comma Separated Values*) :

```
'''
sudo s-tui --csv-file logfile.csv
'''
```

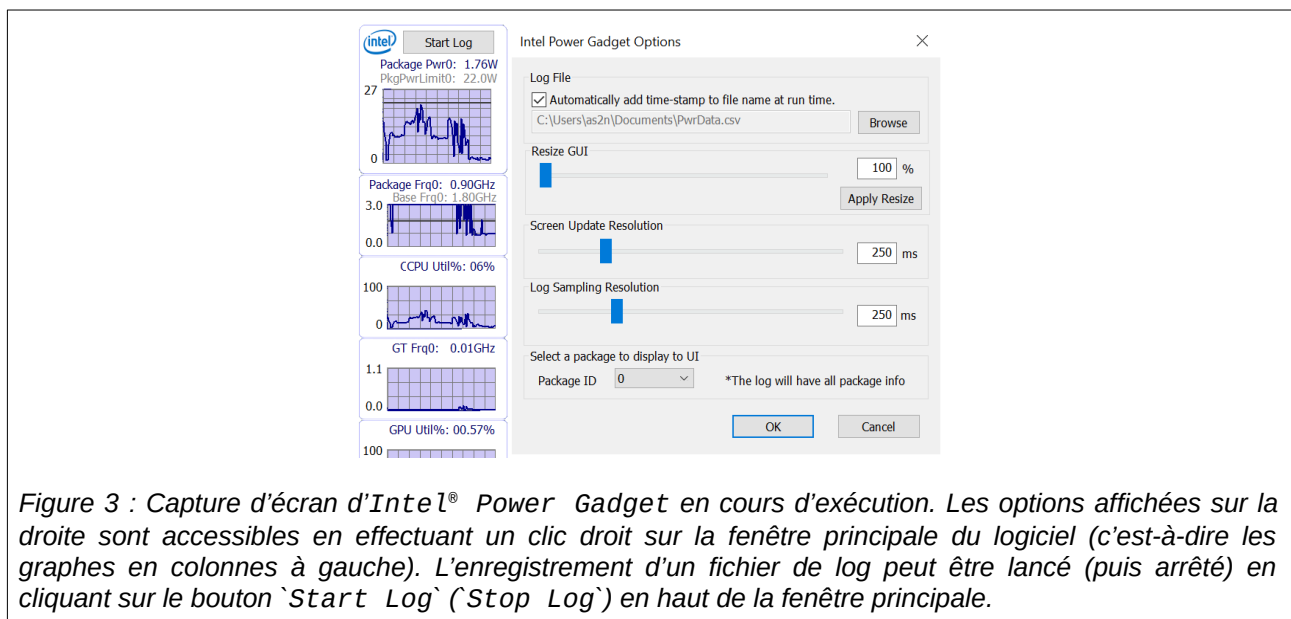
Les réglages antérieurs sont automatiquement rechargés. Dans la commande ci-dessus `logfile.csv` est le nom du fichier de log que vous pouvez modifier à votre guise. Dans le cas de l'exemple utilisé, le fichier est enregistré dans le répertoire courant du terminal.

Pour procéder aux relevés du TP 3 à l'aide de `s-tui` et des fichiers de log de ce dernier, vous pouvez par exemple recourir aux étapes suivantes :

1. lancer `sudo s-tui --csv-file logfile_scenario_X.csv` ;
2. lancer la tâche de calcul à analyser ;
3. quitter `s-tui` une fois l'étape 2 achevée, afin d'interrompre l'enregistrement des données ;
4. analyser à l'aide du script `power_analysis_tools.py` le contenu pertinent du fichier de log `logfile_scenario_X.csv` (voir les détails ci-après) ;
5. compléter votre tableau d'analyse et de synthèse ;
6. recommencer à l'étape 1.

III.1.b Avec Intel® Power Gadget ou MX Power Gadget

Lancez le logiciel et configurez éventuellement à votre guise l'intervalle de rafraîchissement de l'affichage ainsi que celui d'échantillonnage des données (Figure 3). Notez que des considérations similaires à celles faites pour le logiciel `s-tui` s'appliquent au choix de l'intervalle de rafraîchissement de l'affichage : méfiez vous donc d'un intervalle trop court.



Pour procéder aux relevés du TP 3 à l'aide d'Intel® Power Gadget (ou MX Power Gadget) et des fichiers de log de ce dernier, vous pouvez par exemple recourir aux étapes suivantes :

1. lancer Intel® Power Gadget (ou MX Power Gadget) ;
2. lancer l'enregistrement d'un fichier de log `logfile_scenario_X.csv` ;
3. lancer la tâche de calcul à analyser ;
4. interrompre l'enregistrement du fichier de log ;
5. analyser à l'aide du script `power_analysis_tools.py` le contenu pertinent du fichier de log `logfile_scenario_X.csv` (voir les détails ci-après) ;
6. compléter votre tableau d'analyse et de synthèse ;
7. recommencer à l'étape 2.

III.2 Analyse (interactive) de l'enregistrement

III.2.a Utilisation (réglages par défaut)

Description des arguments du script `power_analysis_tools.py`

Un récapitulatif complet des arguments disponibles peut être obtenu à l'aide de la commande :

```
python3 ./power_analysis_tools.py --help
```

Dans tous les cas, au moins deux options sont requises :

1. `-s <logiciel>` (ou `--software <logiciel>`) où `<logiciel>` est respectivement « `intelpowergadget` », « `mxpowergadget` » ou « `stui` » selon que le fichier de log à analyser a été enregistré par Intel® Power Gadget, MX Power Gadget ou s-tui ;
2. `-f <logfile>` (`--filepath <logfile>`) où `<logfile>` est le chemin d'accès au fichier de log à analyser.

Avec un fichier de log produit par s-tui

Les fichiers de log produits par s-tui ne comportant pas d'enregistrement du temps écoulé avec une granularité suffisamment précise, il est indispensable d'utiliser également l'argument `-dt <valeur>` où `<valeur>` correspond à l'intervalle de rafraîchissement des données, exprimé en secondes. Une commande typique ressemble donc à (pour un intervalle de 0,2 s) :

```
python3 ./power_analysis_tools.py -s stui -dt 0.2 -f ./logfile_scenario_X.csv
```

Avec un fichier de log produit par Intel® Power Gadget ou MX Power Gadget

Une commande typique pour analyser un fichier de log produit par Intel® Power Gadget ou MX Power Gadget ressemble à :

```
# Intel Power Gadget
python3 ./power_analysis_tools.py -s intelpowergadget -f ./logfile_scenario_X.csv
# MX Power Gadget
python3 ./power_analysis_tools.py -s mxpowergadget -f ./logfile_scenario_X.csv
```

Les fichiers de log produits par Intel® Power Gadget ou MX Power Gadget comportant un enregistrement du temps écoulé avec une granularité suffisamment précise, il n'est pas nécessaire d'utiliser également l'argument `-dt <valeur>` comme dans le cas des fichiers de log produits par s-tui.

III.2.b Recours possibles en cas de variation poste à poste

Dans le cas où les noms des colonnes visées (à savoir le temps écoulé, la puissance consommée, le taux d'utilisation et la fréquence d'horloge) enregistrées dans vos fichiers de log ne correspondraient pas aux valeurs par défaut utilisées par le script, une (première) solution envisageable est de modifier directement les valeurs pertinentes dans le code source du script `power_analysis_tools.py` :

```

...
# The following dictionnaires define the names of the columns that
# one is going to look after in the log files. If those names were
# actually different in logs recorded on different computers, one can
# modify them either directly here in the source code or using the
# relevant command line arguments. See the output of the command
# `python3 ./power_analysis_tools.py -- help` for more information on
# the latter option.
#
# Regarding the case of ARM processors for Apple computers (thus for
# MX Power Gadget logs), one focuses on the P-core (Performance) rather
# than the E-core (Efficiency) as that is the one that is the more
# likely to be heavily used during the ENSEIRB-Matmeca labworks (about
# computation intensive-tasks).
#
# Names of the columns to target regarding y-axis data
DEFAULT_TARGETED_CSV_YDATA_COLUMN_NAMES = {
    STUI_KEY: { # s-tui
        POWER_KEY: "Power:package-0,0",
        UTILIZATION_KEY: "Util:Avg",
        FREQUENCY_KEY: "Frequency:Avg",
    },
    INTEL_PG_KEY: { # Intel Power Gadget
        POWER_KEY: "Processor Power_0(Watt)",
        UTILIZATION_KEY: "CPU Utilization(%)",
        FREQUENCY_KEY: "CPU Frequency_0(MHz)",
    },
    MX_PG_KEY: { # MX Power Gadget
        POWER_KEY: "Package Power (Watt)",
        UTILIZATION_KEY: "P-Core Utilization (%)",
        FREQUENCY_KEY: "P-Core Frequency (Mhz)",
    },
}
# Name of the column to target regarding x-axis data
DEFAULT_TARGETED_CSV_XDATA_COLUMN_NAME = {
    STUI_KEY: None, # actually not available in s-tui logs
    INTEL_PG_KEY: "Elapsed Time (sec)", # for Intel Power Gadget logs
    MX_PG_KEY: "Elapsed Time (sec)", # for MX Power Gadget logs
}
...

```

Il est possible d'utiliser le nom "none" pour ne pas analyser l'un ou l'autre des types de relevés parmi la puissance consommée, le taux d'utilisation ou encore la fréquence. Le nombre de panneaux d'affichage sera alors adapté en fonction du nombre de courbes restantes.

À noter qu'il est également envisageable de recourir aux arguments en ligne de commande prévus à cet effet (donc sans avoir à altérer le code source du script...). Ci-dessous quelques exemples qui produisent un résultat similaire aux réglages par défaut (pour Intel® Power Gadget) :

```

...
python3 ./power_analysis_tools.py -s intelpowergadget -f ./logfile_scenario_X.csv \
    --time_col_name "Elapsed Time (sec)" \ # not available in s-tui logs
    --power_col_name "Processor Power_0(Watt)" \
    --util_col_name "CPU Utilization(%)\" \
    --freq_col_name "CPU Frequency_0(MHz)"

# Equivalent to the previous command (but with shorter arguments)
python3 ./power_analysis_tools.py -s intelpowergadget -f ./logfile_scenario_X.csv \
    -tcn "Elapsed Time (sec)" \ # not available in s-tui logs
    -pcn "Processor Power_0(Watt)" \
    -ucn "CPU Utilization(%)\" \
    -fcn "CPU Frequency_0(MHz)"
...

```

Enfin, dans le cas où les fichiers de logs produits par Intel® Power Gadget sur votre machine comporteraient un nombre de lignes de synthèse (à la fin du fichier) différent de 19, il est possible de

modifier la valeur pertinente (`skip_footer`) dans le dictionnaire `DEFAULT_CSV_OPTIONS` défini dans le code source du script `power_analysis_tools.py` :

```
...  
# The following dictionary defines the options of the CSV log files  
# produced by the different logging softwares. In case the number of  
# summary lines at the end of Intel Power Gadget logs were actually  
# different in logs recorded on different computers, one can modify  
# the skip_footer value accordingly to the relevant amount of lines.  
DEFAULT_CSV_OPTIONS = {  
    STUI_KEY: { # s-tui  
        "delimiter": ",",  
        "skip_header": 1,  
    },  
    INTEL_PG_KEY: { # Intel Power Gadget  
        "delimiter": ",",  
        "skip_header": 1,  
        "skip_footer": 19, # modify if more or less lines in your logs  
                           # May actually be 15 lines on MacOS (Collin's log)  
                           # vs. 19 lines on Windows (Jego's log)  
    },  
    MX_PG_KEY: { # MX Power Gadget  
        "delimiter": ",",  
        "skip_header": 1,  
    },  
}  
...
```

Contrairement au cas précédent, il n'est pas encore possible de corriger une telle situation simplement à l'aide d'un argument supplémentaire du script `power_analysis_tools.py`.

III.2.c Remarque à propos de l'utilisation du zoom

Après l'utilisation du zoom dans la figure interactive affichée à l'écran, il peut être nécessaire de désélectionner ledit outil de zoom (en cliquant une nouvelle fois sur l'icône) afin de réactiver la possibilité de sélectionner à la souris un intervalle de temps sur lequel estimer la moyenne d'une courbe.