



PowerAPI: mesurer la consommation électrique des logiciels sans wattmètre

Aurélien Bourdon, Frédéric Bordage

► **To cite this version:**

Aurélien Bourdon, Frédéric Bordage. PowerAPI: mesurer la consommation électrique des logiciels sans wattmètre. GreenIT.fr, GreenIT.fr, 2013. <hal-00783426>

HAL Id: hal-00783426

<https://hal.inria.fr/hal-00783426>

Submitted on 1 Feb 2013

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Publié sur *GreenIT.fr* (<http://www.greenit.fr>)

PowerAPI : mesurer la consommation électrique des logiciels sans wattmètre

Par *fbordage*

Créé le 14/01/2013 - 07:54

L'éco-conception d'un logiciel ou d'un site web permet de réaliser d'importantes économies d'énergie au sein des centres informatiques tout en allongeant la durée de vie active des postes utilisateurs. Cette démarche nécessite de mesurer finement la consommation en ressources - cycles processeur, espace disque, quantité de mémoire vive - des différentes parties du code d'un logiciel. C'est ce que propose la librairie PowerAPI.

Mise au point par l'équipe ADAM de l'INRIA à Lille, PowerAPI permet de se passer d'un wattmètre. En libérant le développeur de la contrainte du wattmètre, cette librairie va accélérer la prise en compte des bonnes pratiques d'éco-conception logicielle. Echange avec Aurélien Bourdon qui pilote le projet au sein de l'équipe ADAM de l'INRIA

1) Qu'est-ce que PowerAPI ?



Aurélien Bourdon : PowerAPI est une librairie offrant une API permettant de mesurer la consommation énergétique d'un processus système. Le but étant de pouvoir répondre aux questions du type : "quelle est la consommation énergétique de mon application/processus ?". L'intérêt de PowerAPI repose à la fois sur son architecture et sa manière de mesurer la consommation énergétique :

- Le calcul se base sur la consommation énergétique des composants matériels utilisés par le processus (CPU, mémoire, disque...) et utilise pour cela des formules énergétiques, sans effort matériel, i.e, sans avoir besoin de wattmètre ou d'autres outils matériels de mesure/calibration pour calculer la consommation énergétique.
- Son architecture est modulaire, où chaque module représente une unité de calcul propre (capteur CPU, formule CPU...), permettant à l'utilisateur d'adapter la librairie à sa requête. L'utilisateur compose alors la librairie en ajoutant les modules dont il a besoin, évitant ainsi les surcoûts inutiles.

L'utilisation de PowerAPI se base sur une API écrite en Scala, définissant principalement une méthode, `PowerAPI#startMonitoring()`, prenant en paramètre le processus étudié, le temps de rafraîchissement du calcul et le type de rendu du résultat que l'on souhaite obtenir :

```
PowerAPI.startMonitoring(  
Process(123),  
500 milliseconds,  
classOf[fr.inria.powerapi.listener.console.CpuListener]  
)
```

Dans cet exemple, le processus a pour pid 123, le temps de rafraîchissement est à 500 millisecondes et les résultats seront affichés au sein de la console utilisateur. Un détail plus précis sur l'utilisation de PowerAPI est disponible au sein du site Web du projet [1].

2) Pourquoi avoir lancé ce projet ? N'existait-il pas de solution comparable ?

Le projet vient d'une motivation de l'équipe-projet ADAM, Inria Lille [2], de prolonger ses travaux d'adaptation logicielle dans le domaine du Green Computing [3]. Le but étant à terme de concevoir des applications moins consommatrices en énergie. Il nous fallait alors des outils nous permettant de mettre en évidence cette consommation, notamment au niveau des processus systèmes.

Fort de cette motivation, nous sommes partis à la recherche de tels outils. Cependant, ceux existant ne répondaient pas exactement à nos exigences :

- pTop [4], développé par l'université de Wayne (US), librairie de monitoring énergétique étant assez proche de nos besoins, dans le sens où le calcul énergétique se base au niveau des différents composants matériels (CPU, mémoire, disque...). Cependant, cet outil n'offre pas de support de configuration. Tout se faisant au sein du code et donc avec une nécessité de recompiler à chaque fois l'outil après chaque nouvelle configuration. La librairie ne possède pas non plus de support à l'interopérabilité. Cette difficulté nous a fortement motivé à concevoir l'architecture de PowerAPI de manière modulaire, fournissant ainsi un support à son extensibilité et son interopérabilité.
- JouleMeter [5], développé par Microsoft, est un outil de monitoring énergétique des composants matériels, offrant la possibilité d'obtenir à la fois la consommation énergétique d'un composant matériel donné et à la fois celle d'une application. Cependant, cet outil est exclusivement développé pour les plateformes Windows, et nécessite un composant matériel de calibration lorsqu'il est utilisé sur un PC standard.
- PowerTop [6], développé par Intel, fournissant des astuces pour l'utilisateur dans le but de réduire la consommation énergétique de sa machine. Ici encore, il s'agit d'une application s'exécutant sur un environnement précis (Linux) et n'offrant pas d'interopérabilité sur d'autres environnements.
- EnergyChecker [7], développé par Intel, offrant un moyen de rapporter la consommation énergétique au sein du code source de l'application en se servant de "compteur". Cependant, le calcul de la consommation énergétique nécessite un outil matériel, freinant ainsi son utilisation dans le cas d'une étude sur un ensemble de machines.

3) Quelles sont les applications concrètes de PowerAPI ? 4) N'importe quelle entreprise peut-elle utiliser ce logiciel ?

PowerAPI est sous licence GPL [13]. N'importe quelle personne/entreprise peut donc l'utiliser et la modifier sans restriction. Cependant, l'utilisation de la GPL implique le passage à la GPL

pour tout outil voulant intégrer PowerAPI dans son cœur de métier.

5) L'estimation de la consommation étant basé sur une analyse fine de la configuration technique sous-jacente (CPU, mémoire, etc.) comment la base de données de configuration sera-t-elle mise à jour ?

Comme tu le soulignes, PowerAPI a besoin des spécifications techniques des composants matériels analysés. Dans un souci d'évolution, cette configuration est définie au sein de fichiers extérieurs à la librairie. De plus, afin de faciliter au mieux cette configuration, PowerAPI essaie de se baser sur des données accessibles par l'utilisateur (provenant des constructeurs ou de sites Web spécialisés).

Actuellement, ces fichiers doivent être renseignés manuellement par l'utilisateur. Cependant, nous réfléchissons à la mise en place d'une génération automatique de ces fichiers en se basant sur l'environnement d'exécution. L'idée étant de développer une base de données distante, pouvant être peuplée par la communauté, et servant de support de déploiement pour PowerAPI.

6) L'algo d'estimation de la conso électrique est-il public ou appartient-il à l'INRIA ?

Un algorithme ne peut être breveté en France. Néanmoins, pour protéger le travail réalisé, le code de PowerAPI a fait l'objet d'une publication à l'Agence de Protection des Programmes [14]. Cependant, outre ces problèmes de protection de droits d'auteur, tout le monde peut utiliser la librairie s'il respecte la licence GPL.

Sources :

[1] Site Web de PowerAPI, <http://abourdon.github.com/powerapi-akka> [1]

[2] Equipe-projet ADAM, <http://adam.lille.inria.fr> [2]

[3] Green Middleware, <http://adam.lille.inria.fr/pmwiki.php/Topics/Green> [3]

[4] pTop, http://www.sigops.org/sosp/sosp09/papers/hotpower_13_do.pdf [4]

[5] JouleMeter,

<http://research.microsoft.com/en-us/downloads/fe9e10c5-5c5b-450c-a674-da...> [5]

[6] PowerTop, <https://01.org/powertop> [6]

[7] EnergyChecker,

<http://software.intel.com/en-us/articles/intel-energy-checker-sdk> [7]

[8] Tours de Hanoi, http://fr.wikipedia.org/wiki/Tours_de_Hano%C3%AF [8]

[9] A Preliminary Study of the Impact of Software Engineering on GreenIT (Adel Nouredine, Aurélien Bourdon, Romain Rouvoy, Lionel Seinturier), <http://hal.inria.fr/hal-00681560> [9]

[10] Runtime Monitoring of Software Energy Hotspots (Adel Nouredine, Aurélien Bourdon, Romain Rouvoy, Lionel Seinturier), <http://hal.inria.fr/hal-00715331> [10]

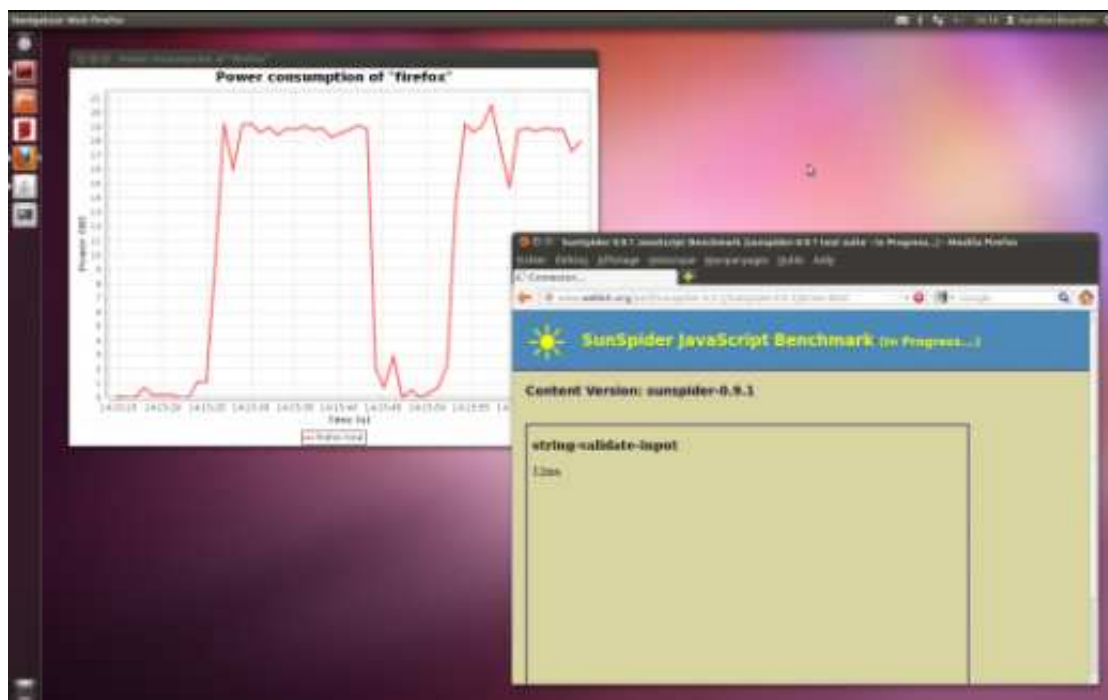
[11] Green Code Lab, <http://www.greencodelab.fr> [11]

[12] HTTP Archive, <http://httparchive.org> [12]

[13] Licence GPL, <http://www.gnu.org/licenses/gpl.html> [13]

[14] Agence de Protection des Programmes (PowerAPI publié sous le numéro IDN.FR.001.400015.000.S.P.2012.000.10000), <http://www.app.asso.fr> [14]

[15] Stress (Linux), <http://linux.die.net/man/1/stress> [15]



[16]

PowerAPI évalue la consommation du navigateur Firefox.

— Envoyer cette page à un ami —

Logiciels

URL source: <http://www.greenit.fr/article/logiciels/powerapi-mesurer-la-consommation-electrique-des-logiciels-sans-wattmetre-4705>

Liens:

- [1] <http://abourdon.github.com/powerapi-akka>
- [2] <http://adam.lille.inria.fr>
- [3] <http://adam.lille.inria.fr/pmwiki.php/Topics/Green>
- [4] http://www.sigops.org/sosp/sosp09/papers/hotpower_13_do.pdf
- [5] <http://research.microsoft.com/en-us/downloads/fe9e10c5-5c5b-450c-a674-daf55565f794>
- [6] <https://01.org/powertop>
- [7] <http://software.intel.com/en-us/articles/intel-energy-checker-sdk>
- [8] http://fr.wikipedia.org/wiki/Tours_de_Hanoi
- [9] <http://hal.inria.fr/hal-00681560>
- [10] <http://hal.inria.fr/hal-00715331>
- [11] <http://www.greencodelab.fr>
- [12] <http://httparchive.org>
- [13] <http://www.gnu.org/licenses/gpl.html>
- [14] <http://www.app.asso.fr>
- [15] <http://linux.die.net/man/1/stress>
- [16] <http://www.greenit.fr/sites/greenit.fr/files/images/PowerAPI-firefox-total.small.png>