

# XtremLab : une plate-forme pour l'observation et la caractérisation des grilles de PC sur Internet

Paul Malécot, Derrick Kondo, Gilles Fedak

INRIA / Laboratoire de Recherche en Informatique  
Université Paris XI,  
bâtiment 490,  
91405 ORSAY  
malecot@lri.fr

---

## Résumé

Les grilles de PC utilisent la puissance de calcul vacante de nombreux ordinateurs personnels mis à disposition par des volontaires pour exécuter des applications de grande taille ou fournir de grands espaces de stockage. Depuis près d'une décennie ces systèmes ont sans doute constitué parmi les plus grands systèmes distribués de l'Internet, agrégeant des puissances de calcul considérables pour une fraction du coût d'un supercalculateur. Si nous savons que les machines composant ces systèmes sont par essence volatiles et hétérogènes, il existe peu d'informations détaillées permettant de comprendre et quantifier leur volatilité et hétérogénéité. Pourtant cette caractérisation est essentielle pour la simulation et la modélisation de ces systèmes. Le projet que nous proposons, XtremLab, permettra d'obtenir une image détaillée, telle une radiographie, des ressources d'une grille de PC. À cette fin, nous déployons des sondes logicielles sur les machines des volontaires pour mesurer les différents niveaux de volatilité des machines (préemption par un utilisateur ou une application, redémarrage et pannes). Nous présentons une première analyse qualitative de ses données comparativement aux grilles de PC dans le contexte des réseaux locaux. Les données résultantes de ces mesures et leurs analyses techniques pourront être utilisées pour un large éventail de recherches scientifiques autour de l'étude des systèmes distribués à large échelle et pair-à-pair. (P2P)

**Mots-clés :** Grilles de PC, caractérisation de ressources

---

## 1. Introduction

Les grilles de PC (ou Desktop Grid) exploitent les ressources vacantes de calcul, de communication et de stockage d'ordinateurs personnels pour exécuter des applications parallèles de grande taille. Depuis la fin des années 90, ces systèmes de calcul tels que SETI@Home [1] impliquent des centaines de milliers de machines distribuées sur Internet et figurent parmi les plus grands systèmes distribués. L'avènement de plate-forme commerciales (Entropia [2] par exemple) ou libres (BOINC [3], OurGRID [4], XtremWeb [5]) permettent à une variété d'applications de différents domaines scientifiques (biologie moléculaire, climatologie, physique des particules, mathématiques, astronomie) d'atteindre des puissances de calcul considérables pour une fraction du coût d'un supercalculateur traditionnel.

Une caractéristique essentielle des grilles de PC est que les ressources qui les composent sont partagées avec les utilisateurs des machines. Il faut donc apporter le plus grand soin pour que les tâches fournies par la grille ne gênent pas les activités du propriétaire de la ressource. Donc une tâche de la grille peut être suspendue ou abandonnée dès que l'utilisateur reprend le contrôle de son ordinateur. De fait les ressources sont volatiles dans le sens que plusieurs facteurs peuvent conduire une tâche à ne pas se terminer correctement : activité clavier/souris, exécution d'une autre application de l'utilisateur, redémarrage de la machine, voir panne matérielle. De plus les ressources sont hétérogènes dans le sens qu'elles diffèrent par leurs systèmes d'exploitation, leur vitesse CPU, leur bande passante réseau, leur mémoire, et la taille de leur disque. Ces caractéristiques, volatilité et hétérogénéité, rendent difficile la conception de systèmes et d'applications capable d'exploiter efficacement ces ressources.

La question qui se pose alors est comprendre la nature et le degré de la volatilité et l'hétérogénéité de ces systèmes. Les études existantes sur les grilles de PC sur Internet ne fournissent que des statistiques agrégées ou brutes, telles que la moyenne, qui peuvent ne pas refléter correctement la distribution de la volatilité. Or, une caractérisation détaillée de cette volatilité est essentielle pour concevoir des grilles de PC efficaces pour différents types d'applications. Pour ce faire, il convient de simuler ou modéliser ces systèmes dans des conditions expérimentales qui soient contrôlées et reproductibles. Les mesures de disponibilité des ressources permettent en partie de reproduire cet environnement.

Nous proposons de concevoir, d'implémenter et de déployer un projet de surveillance des ressources appelé *XtremLab* grâce au logiciel BOINC [3, 6]. Les projets BOINC sont sans conteste les plus grands projets de calculs distribués dans le monde et peuvent regrouper plus de 500 000 participants actifs. Le but d'*XtremLab* est de scruter l'activité d'une large fraction de ces ressources dans l'objectif d'obtenir une image détaillée de cet ensemble.

En particulier, l'application déployée sur les ressources permet de mesurer les différentes périodes de disponibilité ainsi que la puissance processeur utilisable et d'enregistrer ces informations dans un fichier de trace. Ces fichiers de traces sont collectés et assemblés pour former une série temporelle continue de l'activité de chaque ressource du système. Nous proposons une analyse de ces mesures pour en déduire des statistiques descriptives du système dans son ensemble et ressource par ressource. A ce jour, plus de 2000 machines se sont enregistrées sur le site *XtremLab* (<http://xtremlab.lri.fr>).

Le reste de cet article est construit ainsi : dans la section 2, nous décrivons la problématique de la caractérisation des grilles de PC sur Internet, dans la section 3, nous décrivons l'implémentation de *XtremLab*, puis dans la section 4, nous donnons l'analyse de résultats préliminaires obtenus grâce à la plate-forme, finalement nous concluons dans la section 5.

## 2. Contexte

### 2.1. Le système de calcul volontaire BOINC

BOINC (Berkeley Open Infrastructure for Network Computing) est une infrastructure logicielle qui facilite la mise en place d'applications parallèles massivement distribuées sur le modèle du calcul volontaire. C'est à dire que les ressources de calcul sont fournies essentiellement par des ordinateurs personnels répartis sur Internet. Il existe à ce jour une trentaine de projets BOINC différents. Les utilisateurs ont la possibilité de décider à quels projets et dans quelles proportions ils souhaitent faire contribuer leurs ressources. Il suffit à l'utilisateur d'exécuter un logiciel client et de l'attacher au serveur du projet pour recevoir des tâches de calcul en fonction du niveau de contribution souhaité.

De fait, il existe deux niveaux de partage de la ressource. Le premier entre l'utilisateur et le client BOINC, qui est déterminé par la politique de mise à disposition de la machine. Par exemple dans le cas typique d'une activation de type «économiseur d'écran», la politique consiste à définir un temps d'inactivité de périphérique de pointage (souris/clavier). Le second niveau se situe entre les projets BOINC. Lorsque le client est disponible, la répartition de la ressource se fait sur une base de *time slicing* entre les projets avec préemption de la tâche qui dépasse son quantum de temps.

### 2.2. Volatilité et disponibilité des ressources

Il existe différents niveaux de disponibilité, inclus les uns dans les autres :

**Disponibilité machine** Lorsque la machine est allumée et que le client BOINC fonctionne.

**Disponibilité processeur** Lorsque la politique d'activation attribue les ressources de calcul à une tâche BOINC. Notons cependant que l'utilisation du processeur n'est pas exclusive et qu'il est possible que d'autres applications utilisateur puissent tourner en tâche de fond.

**Disponibilité application** Lorsqu'une tâche BOINC tourne, il se peut qu'elle soit interrompue par le client BOINC pour que la ressource puisse être attribuée à un autre projet BOINC.

### 2.3. Travaux récents

Nos travaux complètent les nombreuses études sur la disponibilité des machines. Dans [7] les auteurs utilisent un senseur (mesure passive) qui enregistre périodiquement l'*uptime* des machines depuis le

pseudo système de fichier `/proc`. Dans [8, 9, 10] une méthode similaire est utilisée pour obtenir des traces de disponibilité processeur à partir de la mesure de la charge de la machine. Mais ces ensembles de traces n'enregistrent pas les événements pouvant conduire à l'interruption d'une tâche de l'application. Dans [11, 12], nous avons mesuré et caractérisé plusieurs grilles de PC de l'Université de Californie San Diego et de l'Université Paris XI. Nous avons obtenu plusieurs mois de traces de disponibilité de centaines de machines de bureau au sein de ces organisations. La caractérisation que nous avons obtenue forme la base du modèle décrivant l'utilité des grilles de PC pour différentes applications et pour développer des méthodes efficaces de répartition des tâches sur les nœuds de calcul [13]. Nous avons créé une archive en-ligne de ces traces<sup>1</sup>, disponible pour la communauté scientifique. La limitation de ces travaux, que nous proposons d'améliorer dans ce projet, est que nous n'avons pas de mesures prises depuis les PC domestiques qui contribuent pour une grande part aux projets de calcul volontaire sur Internet.

L'observation des systèmes distribués à large échelle est une tâche difficile et a été peu faite sur les systèmes de calcul volontaire. Dans [14], la plate-forme BOINC est utilisée pour observer les différents types de pannes de machines tournant sous Windows. Ces mesures ne concernent qu'un type de volatilité et les résultats ne sont pas disponibles publiquement. Dans [15], nous avons analysé les statistiques agrégées de l'ensemble de la plate-forme BOINC. Une limitation de ce travail est que les mesures ne décrivent pas la structure temporelle de la disponibilité des ressources individuelles, il serait difficile d'utiliser ces statistiques pour une simulation détaillée ou de déduire un modèle mathématique. Notre travail propose de surmonter cette limitation en obtenant une série temporelle détaillée des mesures de la disponibilité de chaque ressource de la grille de PC.

### 3. Architecture d'XtremLab

Afin d'étendre les mesures effectuées en entreprise à Internet en général et de compléter les mesures déjà faites sur la plate-forme BOINC, nous avons mis en place un projet BOINC appelé XtremLab. Dans cette section, nous décrivons l'architecture d'XtremLab. La figure 1 présente un schéma de cette architecture.

#### 3.1. Mesure Active

Nous réalisons des mesures actives : nous demandons à la plate-forme BOINC d'exécuter un programme que nous connaissons et nous mesurons exactement quand il est exécuté et quelle puissance est utilisée. Ce programme utilise au maximum les ressources qui lui sont accessibles.

Pour cela, nous utilisons une boucle issue des tests de performance (benchmarks) utilisés par BOINC. Celle-ci comporte un nombre connu d'opérations flottantes et entières et est répétée jusqu'à l'arrêt de la tâche. Nous mesurons le nombre de tours de boucle effectués qui est directement proportionnel au nombre d'instructions réellement effectuées. Nous avons donc une mesure précise la puissance réellement disponible pour l'application, c'est-à-dire non utilisée par un autre programme.

Nos tâches ont une durée fixe de 10 minutes pendant lesquelles le résultat de la mesure est noté toutes les 10 secondes. La durée des tâches est constante pour conserver une fréquence de mesure identique quelque soit la puissance de la machine. Les heures précises de début de tâche sont connues. Ce comportement est celui qui était utilisé dans les études précédentes.

#### 3.2. Les tâches

Pour pouvoir utiliser la plate-forme BOINC, nous avons inclus notre mesure active ainsi que la collecte de quelques informations complémentaires à l'intérieur d'une application BOINC. Puis nous produisons une infinité de tâches utilisant cette application de manière à ce que toute machine demandant des tâches en obtienne.

Normalement, toutes les tâches exécutées par BOINC correspondant à une même application ont un nombre d'instructions constant. Leurs temps d'exécution varient donc en fonction des machines et de la charge de celles-ci. BOINC se base sur ce nombre d'instructions et la vitesse des processeurs pour remplir son cache de tâches. Ici, nos tâches ont une durée fixe et donc un nombre d'instructions variable, ce qui perturbe le fonctionnement du client. Nous tirons parti du système prévu par BOINC pour corriger les petites variations de durée des tâches ; et d'un réglage particulier des différentes limites du nombre

---

<sup>1</sup> Desktop Grid Archive : <http://vs25.lri.fr:4320/dg/>

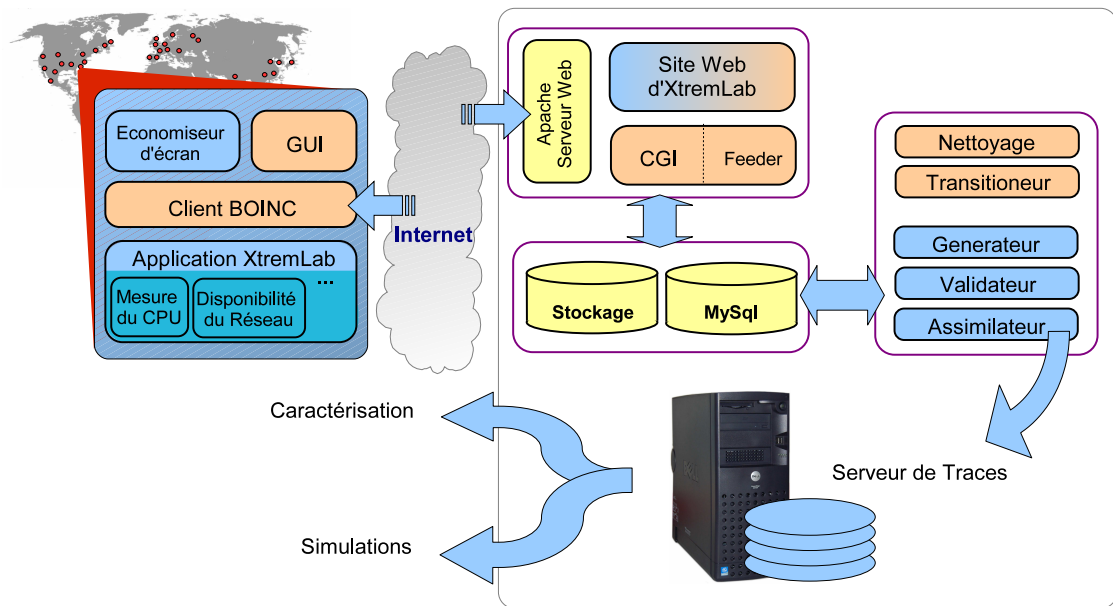


FIG. 1 – Architecture d’XtremLab. La partie gauche est installée chez chaque volontaire, la partie de droite est l’architecture de notre serveur.

de tâches distribuées afin de garantir que les machines aient toujours un nombre suffisant de tâches dans leur cache. En particulier, nous cherchons à éviter toute pénurie de tâches qui se traduirait dans nos résultats par une «fausse» période d’indisponibilité.

Outre la mesure active, notre outil consigne dans le fichier de résultat le temps depuis lequel la machine est en fonctionnement. Cela permet de savoir si entre deux tâches la machine est restée en fonctionnement ou si elle a subi un arrêt. Entre deux exécutions de nos tâches, il pourrait cependant y avoir plus d’une coupure du système. Par ailleurs, nous ne savons pas combien de temps la machine continue à fonctionner après la dernière exécution de l’une de nos tâches.

La plate-forme BOINC collecte elle-même un certain nombre d’informations à propos de la machine : version du système d’exploitation, type de processeur, mémoire, le réseau, les caractéristiques du disque sur lequel BOINC est installé. Mais aussi des informations sur le fonctionnement du client : dernier contact avec le serveur, dernier changement d’adresse IP et quelques statistiques sur le temps que BOINC passe à calculer au cours de son fonctionnement.

De nombreuses machines ne sont pas synchronisées à l’heure exacte. De plus, comme le projet est mondial, les machines se trouvent dans des fuseaux horaires différents. Pour pouvoir comparer ou calculer des statistiques sur plusieurs machines au même instant, il faut synchroniser les différentes traces obtenues. Pour cela nous avons mis en place, sur notre serveur, un serveur de temps. Nos tâches s’y connectent à chaque début d’exécution. Si l’heure est obtenue, on peut alors synchroniser cette trace avec les autres. Si l’on a réussi à obtenir l’heure, on sait aussi que le réseau était disponible à ce moment là. A l’inverse si l’heure n’a pas pu être obtenue, notre serveur n’était alors pas joignable et nous ne pouvons utiliser les résultats qu’individuellement.

Nos tâches peuvent être interrompues de plusieurs manières. Notre application nous permet de discerner différentes situations :

- Fin normale de la tâche : Le nombre de mesures dans le fichier de résultat est atteint. BOINC passe à la tâche suivante et commence une nouvelle série de mesures.
- Attribution du processeur au projet suivant, réquisition des ressources par l’utilisateur ou utilisation de la machine sur batteries : Dans ce cas une période sans mesures apparaît dans les résultats correspondant à la tâche concernée. Si l’utilisateur l’a demandé, l’application est conservée en mémoire tant que le client BOINC n’est pas lui-même arrêté. La durée de cette conservation en mémoire, mais sans

calcul, est notée.

- Période d'arrêt de la machine ou crash du système : ce cas peut être détecté dans la mesure ou les mesures sont interrompues en plus d'une remise à zéro du temps de fonctionnement du système. (*uptime*)
- Crash ou arrêt de BOINC : Dans le cas où le client est configuré pour conserver les applications en mémoire, un retrait de la mémoire signifie que BOINC a subi une période d'arrêt.
- Retrait de la machine du projet : BOINC ne prévoit pas de retrait explicite d'une machine d'un projet. Cependant, nous pouvons considérer une machine comme retirée après plusieurs semaines sans contact avec le serveur.

Normalement, BOINC effectue une réplication des tâches entre différentes machines pour s'assurer de l'exactitude du résultat du calcul scientifique exécuté. Pour cela chaque projet fournit un *validateur* (*validator*) chargé de comparer les résultats obtenus et de déclarer s'ils sont valables. (au minimum pour récompenser les utilisateurs) Nous avons désactivé cette réplication. La validation se contente de vérifier la syntaxe du fichier afin de détecter les tentatives manifestes de fausser nos statistiques. Les résultats non valides sont tout de même conservés, à part, afin d'être étudiés.

Les différents projets BOINC doivent fournir un fichier exécutable différent pour chaque plate-forme supportée, nous en fournissons pour l'instant pour Linux et Windows. Une version pour MacOS X est à l'étude, pour l'instant seul un faible nombre de ces machines sont enregistrées. Les exécutables et les plates-formes étant légèrement différents, des précautions doivent être prises lors de la comparaison directe des valeurs obtenues entre les différentes machines. Par contre, on peut faire des comparaisons en se basant sur la valeur maximum obtenue par la machine.

Si notre mesure permet de compléter les mesures passives, notre méthode a tout de même quelques limitations. Tout d'abord notre projet, comme tout autre projet BOINC est jugé par les utilisateurs et décident de lui attribuer plus ou moins de temps machines en fonction de s'ils pensent que l'étude de BOINC est utile.

Sur les machines multi-processeurs, BOINC peut, selon les préférences de l'utilisateur exécuter plusieurs tâches en même temps, une par processeur. Il est alors possible que plusieurs tâches d'XtremLab soient exécutées en même temps. Les différents «slots» utilisés par XtremLab sont différenciés dans les résultats obtenus. Par contre, aucune information n'est donnée sur le processeur exact qui exécute notre application à un moment donné.

#### 4. Premiers résultats

Plus de 2000 machines appartenant à plus de 900 utilisateurs ont été enregistrées auprès du projet XtremLab. Nous nous attendons à ce que ce nombre progresse au fur et à mesure que le projet sera considéré comme stable. Nous avons publié la version Linux le 1er mars 2006. La version pour Windows a suivie le 3 avril 2006. Depuis l'ouverture du projet nous avons accumulé l'équivalent d'environ 2,2 années de temps processeur pour une machine fonctionnant à 1GHz. Au 18 avril 2006, nous avons collecté 13 millions de mesures en provenance de 900 machines actives, c'est-à-dire de machines ayant fourni au moins un résultat.

##### 4.1. Distribution de la vitesse des machines

La plate-forme BOINC collecte, entre autres, la marque et le modèle des processeurs des machines qui sont enregistrées auprès d'XtremLab. Nous avons extrait de ses informations la vitesse pour Intel et l'indice de puissance pour AMD. Ce sont des données constructeurs, dans certains cas la fréquence réelle d'utilisation du processeur peut être différente (*overclocking* ou économie d'énergie). Quelques rares processeurs, tels les anciens modèles ne fournissant pas ces informations ou encore les processeurs Power PC et VIA, ne sont pas représentés ici.

Sur la figure 2, nous avons représenté le pourcentage cumulé du nombre de machines en fonction de leurs vitesse d'horloge. L'indice de performance d'AMD a été transformé en une vitesse équivalent et additionné à la vitesse fournie par Intel. Seules 10% des machines ont une vitesse supérieure à 3,2 GHz. La moitié des machines sont situées entre 2,5 et 3,2 GHz. Il y a un facteur 10 entre la machine la plus rapide et la moins rapide, ce qui dénote une hétérogénéité élevée en terme de fréquence processeur.

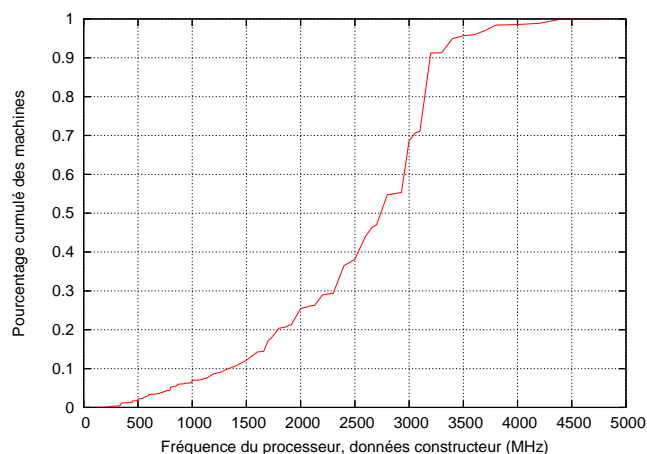


FIG. 2 – Pourcentage cumulé des machines en fonction de la vitesse indiquée par leur constructeur

#### 4.2. Étude des coupures

Notre méthode impliquant la soumission de tâches réelles, il y a inévitablement des coupures dans nos mesures. Une coupure est une période sans mesure entre deux mesures. La plupart des coupures observées correspondent aux périodes où le client, après avoir terminé la tâche précédente, se prépare à exécuter la tâche suivante.

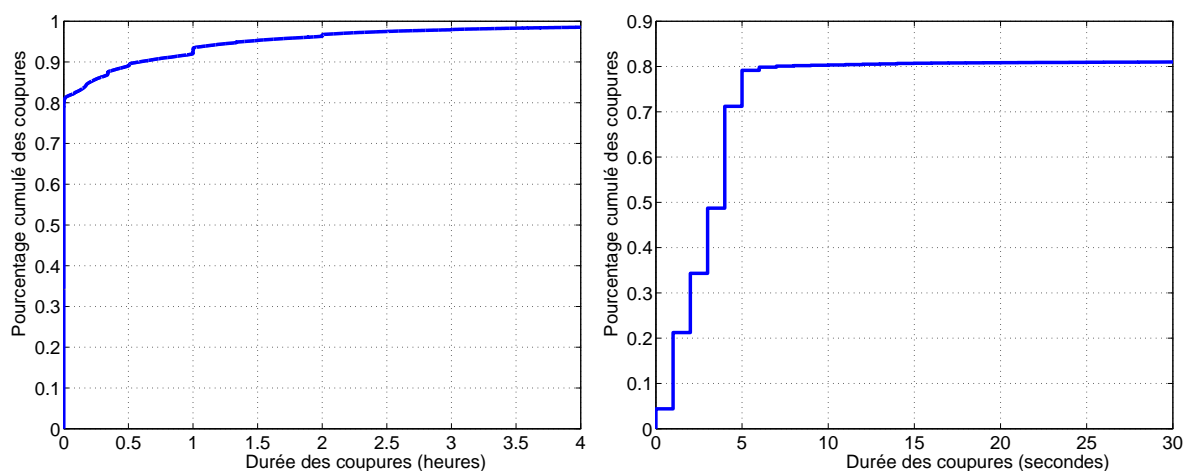


FIG. 3 – Répartition des coupures : (intervalle entre deux tâches) d'une durée inférieure à 4 heures à gauche, d'une durée inférieure à 30 secondes à droite.

Sur la figure 3, nous avons représenté le pourcentage cumulé de ces coupures en fonction de leurs durées. Les coupures durent entre 0 et environ 35 jours. Ceci indique que certaines machines ne participent à XtremLab qu'une fois par mois, ce qui est très peu. Ceci implique, que ces machines sont soit très peu utiles soit très volatiles.

Mais la plupart des coupures, 99%, durent moins de 4 heures. Ceci indique que la plupart des machines participe à leurs différents projets au moins une fois toutes les 4 heures. Les dates limites (*deadlines*) des

tâches doivent donc être d'au moins 4 heures pour que les machines puissent renvoyer leurs résultats à temps.

Sur la courbe de droite, il apparaît que 80% des coupures sont inférieures à 10 secondes. La somme des durées de ces coupures inférieures à 10 secondes ne représente cependant que 0,17% du temps de mesure total. Si l'on souhaite disposer d'une trace continue de la disponibilité des machines, on peut donc réaliser une interpolation linéaire de la puissance disponible pendant ces coupures. Cette méthode n'éliminera que les indisponibilités réelles du processeur d'une durée inférieure à 10 secondes. La précision de nos traces reste tout de même suffisante.

### 4.3. Vitesse d'horloge et puissance réelle

Sur la figure 4, nous avons représenté à la fois la puissance réelle et les vitesses des machines. Pour obtenir la puissance réelle, nous avons calculé le nombre moyen d'opération réalisé en se basant sur la durée entre la première et la dernière mesure de chaque machine ; et la somme totale des mesures réalisées pendant cette période. Les machines sont ensuite triées de la moins puissante à la plus puissante.

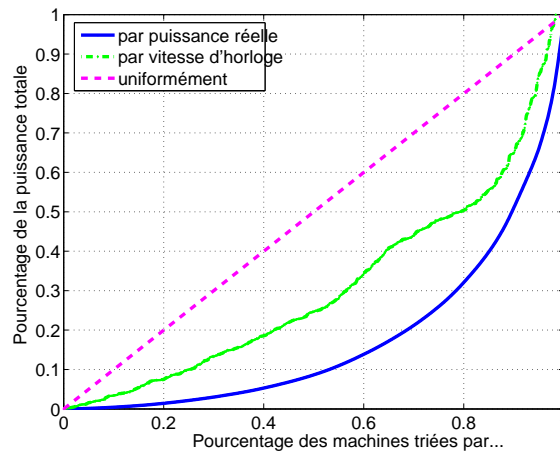


FIG. 4 – Pourcentage cumulé de la puissance totale de la machine en fonction des machines triées par puissance réelle et vitesse d'horloge

A l'aide de cette courbe de la puissance réelle, on peut remarquer que 70% de la puissance totale est fournie par les 20% machines les plus puissantes. D'autre part, les 50% machines les moins puissantes ne produisent que 10% de la puissance totale.

La deuxième courbe représente la puissance fournie par les machines en fonction de leur vitesse d'horloge. L'écart entre les deux courbes est significatif. En particulier, en se basant sur les vitesses, les machines les moins rapides semblent plus contribuer. Pour mesurer cet écart, nous avons calculé le coefficient de Kendall. Il correspond, intuitivement, à la différence de probabilité que deux variables prises au hasard soient classées dans le même ordre par rapport à la probabilité qu'elles ne soient pas dans l'ordre. Il est de 0,267 entre la puissance réelle et la vitesse d'horloge. Il y a donc une probabilité de 0,267 que le classement selon les vitesses d'horloge soit dans le même ordre que le classement par puissance réelle. L'utilisation de la vitesse pour classer les machines n'est donc peut être pas pertinente utilisée seule. Dans le cas des grilles d'entreprise, la vitesse d'horloge pouvait être utilisée. Ici, de nombreux autres phénomènes interviennent dans la disponibilité des machines telles que l'activité de l'utilisateur ou la disponibilité de la machine.

## 5. Conclusion

Dans cet article, nous avons décrit XtremLab, un projet utilisant la plate-forme BOINC. Ce projet permet d'obtenir une image détaillée des ressources disponibles sur les grilles de PC. Il réalise des mesures actives qui permettent d'observer avec précision quand l'application est exécutée et quelle puissance est mise à sa disposition.

A l'aide de cet outils, nous avons obtenus quelques premiers résultats. En étudiant la vitesse des machines participante, nous avons caractérisé leur hétérogénéité. Puis nous avons étudié les coupures interrompant les calculs. Enfin, en calculant la puissance réelle des différentes machines, nous avons vu que l'utilisation seule de la vitesse des horloges des machines ne suffit à rendre compte de cette puissance.

Le projet XtremLab est toujours ouvert à de nouveaux participants et au fur et à mesure qu'il deviendra plus populaire auprès des volontaires, nous devrions obtenir des résultats plus précis. Les données collectées seront disponibles sur le serveur et de nouvelles mesures concernant la manière dont l'ordonnement est effectué dans le client BOINC seront ajoutées.

## Bibliographie

1. W. T. Sullivan, D. Werthimer, S. Bowyer, J. Cobb, G. Gedye, D. Anderson, A new major SETI project based on Project Serendip data and 100,000 personal computers, in : Proc. of the Fifth Intl. Conf. on Bioastronomy, 1997.
2. A. Chien, B. Calder, S. Elbert, K. Bhatia, Entropia : Architecture and Performance of an Enterprise Desktop Grid System, *Journal of Parallel and Distributed Computing* 63 (2003) 597–610.
3. T. B. O. I. for Network Computing, <http://boinc.berkeley.edu/>.
4. N. Andrade, W. Cime, F. Brasileiro, P. Roisenberg, OurGrid : An Approach to Easily Assemble Grids with Equitable Resource Sharing, in : Proceedings of the 9th Workshop on Job Scheduling Strategies for Parallel Processing, 2003.
5. G. Fedak, C. Germain, V. N'eri, F. Cappello, XtremWeb : A Generic Global Computing System, in : Proceedings of the IEEE International Symposium on Cluster Computing and the Grid (CC-GRID'01), 2001.
6. D. Anderson, BOINC : A System for Public-Resource Computing and Storage, in : proceedings of the 5th IEEE/ACM International Workshop on Grid Computing, Pittsburgh, USA, 2004.
7. J. Brevik, D. Nurmi, R. Wolski, Quantifying Machine Availability in Networked and Desktop Grid Systems, Tech. Rep. CS2003-37, Dept. of Computer Science and Engineering, University of California at Santa Barbara (November 2003).
8. W. Bolosky, J. Douceur, D. Ely, M. Theimer, Feasibility of a Serverless Distributed file System Deployed on an Existing Set of Desktop PCs, in : Proceedings of SIGMETRICS, 2000.
9. P. Dinda, D. O'Hallaron, An Evaluation of Linear Models for Host Load Prediction, in : Proceedings of the The Eighth IEEE International Symposium on High Performance Distributed Computing, 1999, p. 10.
10. R. Wolski, N. Spring, J. Hayes, Predicting the CPU Availability of Time-shared Unix Systems, in : Proceedings of 8th IEEE High Performance Distributed Computing Conference (HPDC8), 1999.
11. D. Kondo, M. Taufer, C. Brooks, H. Casanova, A. Chien, Characterizing and Evaluating Desktop Grids : An Empirical Study, in : Proceedings of the International Parallel and Distributed Processing Symposium (IPDPS'04), 2004.
12. D. Kondo, G. Fedak, F. Cappello, A. A. Chien, H. Casanova, Resource Availability in Enterprise Desktop Grids, Tech. Rep. 00000994, Dept. of Computer Science, INRIA (January 2006, submitted).
13. D. Kondo, A. Chien, C. H., Rapid Application Turnaround on Enterprise Desktop Grids, in : ACM Conference on High Performance Computing and Networking, SC2004, 2004.
14. A. Ganapathi, Why Does Windows Crash?, Master's thesis, The Berkeley/Stanford Recovery-Oriented Computing (ROC) Project (2005).
15. D. Anderson, G. Fedak, The Computational and Storage Potential of Volunteer Computing, Tech. rep., Space Sciences Laboratory, UC Berkeley (November 2005).