

Caractérisation de la performance des microprocesseurs pendant l'exécution de programmes : Régularité, chaos et autres dynamiques

H. Berry, D. Gracia Pérez, & O. Temam

INRIA Futurs, Alchemy, Parc Club Orsay Université, ZAC des vignes 4, rue Jacques Monod, Bat G. 91893 Orsay Cedex
hugues.berry@inria.fr

Résumé. Les microprocesseurs modernes sont composés de centaines de millions de transistors qui interagissent selon des protocoles complexes. L'évolution de leur performance durant l'exécution d'un programme peut donc être très variable et présente souvent des oscillations non périodiques. Nous avons étudié l'évolution temporelle de la performance au cours de l'exécution de programmes sur les microprocesseurs actuels, à l'aide d'une batterie de méthodes pour l'étude des séries temporelles non linéaires. Si l'évolution de la performance pour certains programmes est périodique et régulière, elle est extrêmement riche et variable pour de nombreux autres. De plus, bien que la grande variabilité observée pour certains programmes semble provenir de comportements apparemment stochastiques, la dynamique temporelle pour certains autres programmes montre des indications claires de chaos déterministe, avec une sensibilité aux conditions initiales comparable aux modèles classiques. Les applications de ces résultats dans le domaine de l'architecture des ordinateurs sont brièvement discutées.

Abstract. Modern computer microprocessors are composed of hundreds of millions of transistors that interact through intricate protocols. Their performance during program execution may be highly variable and present aperiodic oscillations. Here, we studied the temporal dynamics of the performance during program execution on current microprocessors, using a collection of methods for nonlinear time series analysis. While performance evolution for several programs is periodic and regular, it displays richer dynamics for many others. Furthermore, although the large variability observed for some programs seems to stem from apparently stochastic processes, temporal dynamics for several programs display clear indications of deterministic chaos, with sensitivity to initial conditions that is comparable to textbook models. Some applications of these results to computer architecture are briefly discussed.

Les architectures d'ordinateur modernes sont le résultat d'une évolution rapide qui a débuté dans les années soixante, quand Moore a observé que le nombre de transistors par circuit intégré croissait de façon exponentielle, et a prédit que cette tendance continuerait [1]. Cette fameuse "Loi de Moore" a en effet été vérifiée durant les 40 dernières années, la densité de transistor doublant environ tous les 18 mois. Une des principales conséquences en est que les microprocesseurs des ordinateurs actuels sont composés d'un nombre impressionnant de transistors : le familier Pentium® 4 d'Intel® contient déjà 42 millions de transistors, mais le plus récent Itanium® 2 en possède 410 millions. De plus, une caractéristique constante de cette évolution est que la vitesse du microprocesseur (et spécialement sa fréquence d'horloge) est toujours de loin plus performante que les opérations d'accès à la mémoire. Les avancées les plus récentes dans le domaine ont donc principalement visé à cacher cette latence mémoire à l'aide de solutions d'ingénierie (exécutions parallèles, pipelining, hierarchies de caches mémoires...). Mais cela s'est forcément accompagné d'une augmentation de la complexité du microprocesseur. En conséquence, il est de plus en plus difficile de prévoir la performance précise des microprocesseurs modernes lors de l'exécution d'un programme (i.e. le nombre d'instructions exécutées chaque seconde). Dans cet article, nous présentons une étude visant à quantifier l'évolution temporelle de la performance (sa dynamique) lors de l'exécution de plusieurs programmes sur des microprocesseurs modernes modèles.

1 Traces d'exécution

Les séries temporelles étudiées dans cet article ont été obtenues à l'aide d'un simulateur de microprocesseurs, appelé SimpleScalar [2], qui correspond à l'architecture d'un microprocesseur superscalaire moderne typique (comme le Pentium 4, par exemple). A l'aide de ce simulateur, nous avons exécuté les 26 programmes tests (benchmarks) de la série "Spec" (plus précisément, la version Spec2000). Un benchmark est un programme qui a été sélectionné comme étant "représentatif" d'un domaine d'application donné. La suite de benchmarks "Spec" est la plus couramment utilisée pour évaluer et comparer la performance de nouveaux ordinateurs et architectures de microprocesseurs. Pendant l'exécution, nous avons collecté trois mesures de la performance : l'IPC, et les taux de défaut des caches L1 et L2. L'IPC est le nombre moyen d'instructions exécutées par cycle (*Instructions Per Cycle*), et est la mesure typique de la performance globale des microprocesseurs superscalaires. L1 et L2 correspondent aux caches mémoires de premier et second niveau, respectivement, qui sont des mémoires rapides et de petite taille, dont le but est de cacher la latence de la mémoire principale. Le taux de défaut pour chaque niveau de mémoire est la fraction de requêtes mémoires issues du microprocesseur qui ne sont pas trouvées dans le cache en question. Les taux de défaut sont donc caractéristiques de l'efficacité d'utilisation des caches mémoires.

Les traces d'exécution étudiées ci-après présentent l'évolution de la moyenne de la mesure concernée sur un certain nombre τ_{av} d'instructions exécutées de façon consécutive. Ainsi, si $\{x(1), x(2), x(3), \dots, x(N)\}$ est la série temporelle en question, chaque valeur $x(n)$ de la série représente la moyenne de x entre l'exécution de l'instruction numéro $n\tau_{av}$ et celle de l'instruction numéro $(n+1)\tau_{av}$. Ces traces d'exécution ont été analysées à l'aide d'une batterie de méthodes d'analyse des séries temporelles non-linéaires. Pour nombre de ces méthodes, nous avons utilisé la package de routines TISEAN [3]. Cet article présente une partie des résultats obtenus. Une analyse plus complète peut être trouvée dans [4].

2 Résultats

La figure 1 présente la dynamique de l'évolution des mesures de la performance pour le programme bzip2 (un programme de compression). Pour des raisons de lisibilité, nous ne présentons sur la fig. 1A que les résultats correspondant aux 54 premiers milliards d'instructions exécutées (i.e. environ la moitié de l'exécution complète du programme). Les trois séries montrent deux phases distinctes : une première phase (jusqu'à environ 43 milliards d'instructions) présente une plus grande amplitude et une fréquence plus faible. La seconde phase (entre 43 et 54 milliards d'instructions) est caractérisée par une amplitude de variation plus faible et une fréquence supérieure. La seconde partie de la série totale (non montrée sur la fig. 1A) consiste essentiellement en la répétition de ces deux mêmes phases consécutives. Dans la suite de cette section, les résultats sont obtenus en traitant la trace entière (longue d'environ 110 milliards d'instructions) comme une seule entité. Néanmoins, nous avons aussi étudié les deux phases séparément (i.e. en restreignant la série temporelle à la première phase, ou à la seconde) et les résultats obtenus sont qualitativement identiques à ceux obtenus avec la trace entière (quoique la sensibilité aux conditions initiales semble plus importante au cours de la seconde phase).

Bien que ces séries temporelles montrent une certaine régularité, les deux phases forment clairement des dynamiques irrégulières ou bruitées. Nous avons tout d'abord recherché des corrélations à long terme dans les séries à l'aide de l'analyse spectrale et de l'analyse des fluctuations redressées. La fig. 1B montre les spectres de puissance en échelle log-log. Tout d'abord, il apparaît que le spectre a un aspect "diffus", typique des signaux chaotiques et stochastiques. De plus, et pour les trois mesures étudiées, le spectre de puissance décroît comme une loi de puissance d'exposant $\beta \approx 1.3$, pour des fréquences $f \geq 2 \times 10^{-9} \text{ instruction}^{-1}$ (i.e. pour des périodes plus petites que la période moyenne de répétition majeure, ≈ 0.6 milliards d'instructions). L'analyse des fluctuations redressées correspondante (Detrended Fluctuation Analysis, ou DFA [5]) est présentée fig. 1C. Là encore, pour des échelles n plus petites que la période moyenne de répétition majeure, on observe pour les trois séries temporelles, une relation en loi de puissance entre les fluctuations redressées $F(n)$ et l'échelle d'observation n , avec un exposant $\alpha \approx 1.13$. Les valeurs de ces deux exposants, obtenues de façon indépendante, satisfont à la relation $\alpha = (1 + \beta)/2$, ce qui est une indication de la cohérence des valeurs obtenues [6]. Ces résultats montrent que la dynamique de la

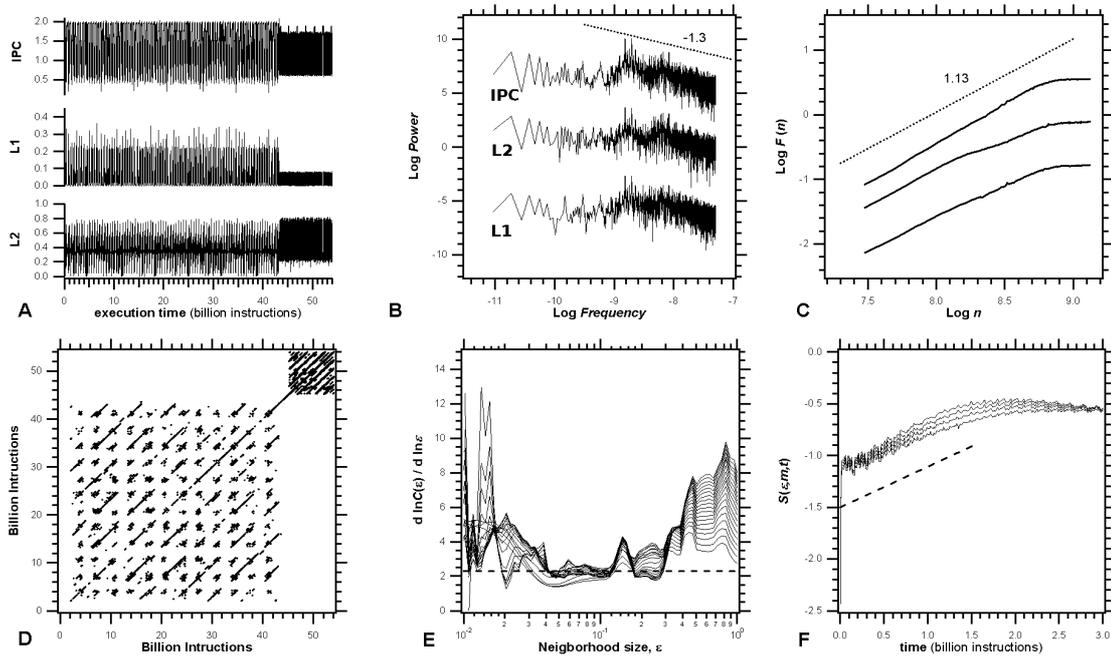


Fig.1. Études de la dynamique de la performance durant l'exécution du programme `bzip2`. (A) Évolution des trois mesures de performance durant les 54 premiers milliards d'instruction. Les spectres de puissance correspondant sont présentés en (B). La ligne discontinue indique une décroissance en loi de puissance, avec un exposant de 1.3. (D) Analyse des fluctuations redressées (DFA). Le graphe présente en échelle log-log les fluctuations redressées ($F(n)$) en fonction de l'échelle d'observation n . (D) Graphe de récurrence correspondant à (A) (seuillage à 0.136). (E) Pentes locales des sommes de corrélation $d \ln C(m, \epsilon) / d \ln \epsilon$, pour des dimensions de plongement m variant du haut en bas, entre 25 et 7 (par décrétement de 1). (F) Estimation du plus grand exposant de Lyapunov, λ_{max} à l'aide de la méthode de Kantz avec $m = 12$ à 15 (du haut vers le bas). La ligne discontinue indique une croissance linéaire de pente 0.42, correspondant à une estimation de $\lambda_{max} \approx 0.60$ bits/milliard d'instructions. Paramètres de plongement (D-F) : délai $d = 153$ million d'instructions, dimension $m = 14$.

performance lors de l'exécution de `bzip2` présente des caractéristiques typiques de signal en $1/f^\beta$. De plus, la valeur obtenue pour α est supérieure à 0.5, ce qui indique la présence de corrélations à longue portée persistantes. En d'autres termes, une grande valeur (par rapport à la moyenne) de l'IPC ou d'un taux de défaut de cache a plus de chance d'être suivie d'une grande valeur de l'IPC ou du taux de défaut en question (et vice-versa).

Pour étudier la dynamique plus avant, nous avons reconstruit l'attracteur par plongement de la série temporelle pour l'IPC. Le plongement par la méthode des délais consiste à partir de la série temporelle scalaire initiale $\{x(t)\}$ et de l'utiliser pour former une nouvelle série temporelle vectorielle $\{\mathbf{X}(t)\}$ dans un espace de plongement de dimension m selon $\mathbf{X}(t) = (x(t), x(t + \tau), \dots, x(t + (m - 1)\tau))$, où τ est le délai de plongement. Le théorème de plongement de Takens [7] montre que, pour des valeurs suffisantes de m , la géométrie de l'attracteur reconstruit $\{\mathbf{X}(t)\}$ dans l'espace de plongement, a les mêmes propriétés topologiques que l'attracteur original dans l'espace des phases naturel. La fig. 1D présente le graphe de récurrence seuillé [8] correspondant (le seuil est déterminé selon [9]). Il est d'abord visible sur cette figure que les deux phases de la dynamique (fig. 1A) sont reconnaissables sur le graphe de récurrence, ce qui indique que ces deux phases pourraient avoir des taux de récurrence assez différents. De façon plus intéressante, ces graphes montrent un grand nombre de diagonales interrompues, ainsi que des points isolés. Ce type de structure est typique des systèmes chaotiques [8]. De plus, des tests statistiques, à l'aide de données « surrogates » [3] (non montré), ont rejeté l'hypothèse que la dynamique d'évolution de l'IPC pourrait provenir d'un processus aléatoire Gaussien (possiblement redimensionné), ce qui confirme

la nature non-linéaire de cette série temporelle.

Nous avons ensuite caractérisé les propriétés géométriques de l'attracteur reconstruit. La dimension de

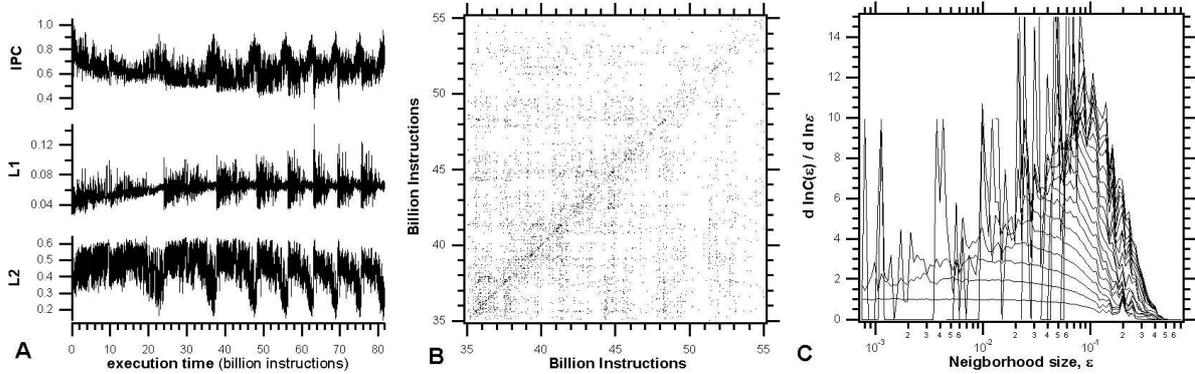


Fig.2. Études de la dynamique de la performance durant l'exécution du programme `vpr`. (A) Dynamique d'évolution des mesures de la performance au cours d'une exécution complète du programme. (B) Graphe de récurrence correspondant (seuillage à 0.079). (C) pentes locales des sommes de corrélations $d \ln C(m, \epsilon) / d \ln \epsilon$ pour une dimension de plongement m variant, du haut en bas, de 20 à 1 (par décrement de 1). Paramètres de plongement (B-C) : délai $d = 350$ million d'instructions, dimension $m = 5$.

corrélacion de l'attracteur a été déterminée selon la méthode de Grassberg-Procaccia [9]. La figure 1E montre les pentes locales $d \ln C(m, \epsilon) / d \ln \epsilon$ mesurées à partir des représentations en log-log des sommes de corrélacion $C(m, \epsilon)$ en fonction de la taille du voisinage considéré, ϵ . Pour $0.03 \leq \epsilon \leq 0.3$ et $m \geq 9$, les pentes locales prennent des valeurs qui sont largement indépendantes de m et d' ϵ . La présence d'un tel régime de scaling est un signe fort de la présence d'un attracteur étrange. La valeur obtenue dans le régime de scaling donne une estimation de la dimension fractale (de corrélacion) de cet attracteur : $D_2 = 2.3 \pm 0.3$. La valeur non-entière de cette dimension pourrait être une indication de la nature fractale de cet attracteur chaotique. Néanmoins, comme dans la plupart de systèmes issus du monde réel, l'estimation de la dimension de corrélacion n'est pas assez précise pour exclure une valeur entière, si bien que la nature fractale de l'attracteur ne peut pas être assurée par les résultat présents. La faible valeur de D_2 n'en reste pas moins une forte indication que la dynamique de `bzip2` résulte de la présence de chaos déterministe de faible dimension. Les sommes de corrélacion peuvent aussi être utilisées pour

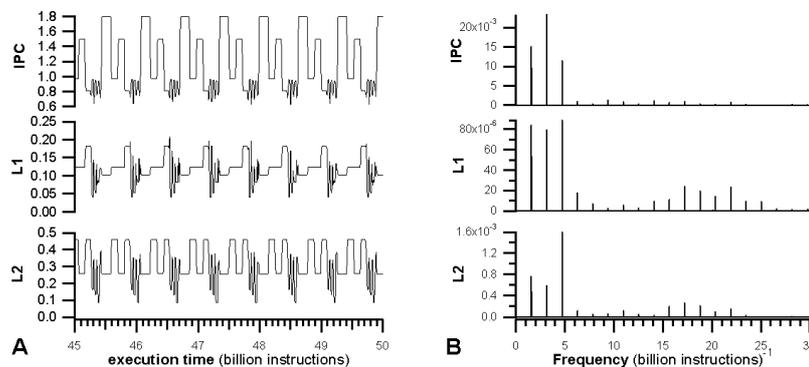


Fig.3. Études de la dynamique de la performance durant l'exécution du programme `applu`. (A) Dynamique d'évolution des mesures de la performance pendant 15 milliards d'instructions (le programme dans son ensemble consiste en environ 220 milliards d'instructions). Les spectres de puissance correspondants sont présentés en (B).

estimer l'entropie de corrélation correspondante, h_2 . Nos estimations donnent $h_2 \approx 1.2$ bits/milliard d'instructions (non montré).

Pour quantifier la sensibilité aux conditions initiales du système, nous avons ensuite mesuré la valeur du plus grand exposant de Lyapunov, λ_{max} (Figure 1F) à l'aide de la méthode de Kantz [11] (n.b. : la méthode voisine de Rössler, donne des résultats équivalents). Les courbes montrent clairement une croissance initiale linéaire, dont la pente donne une estimation de la valeur de $\lambda_{max} \approx 0.60$ bits/milliards d'instructions. Cette valeur est du même ordre de grandeur que la valeur trouvée à partir de cartes de Poincaré de la dynamique ($\lambda_{max} \approx 1.22$, non montré) et que la valeur de l'entropie de corrélation h_2 (qui est une borne inférieure pour la somme de tous les exposants de Lyapunov positifs du système). Ces données supportent la cohérence de nos mesures et renforcent la démonstration de l'existence de chaos déterministe dans la dynamique de `gzip2`. Etant donnée la durée d'une orbite moyenne autour de l'attracteur (≈ 430 millions d'instructions), la valeur de λ_{max} varie entre 0.26 et 0.52 bits/orbite moyenne. Cette valeur, quoique plus faible que celle obtenue pour le système classique de Lorenz ($\lambda_{max} = 1.36$ bits/orbite), indique une sensibilité aux conditions initiales comparable à celle obtenue pour le système de Rössler ($\lambda_{max} = 0.78$ bits/orbite).

Enfin, les mesures et analyses présentées jusqu'ici on essentiellement été obtenues par une reconstruction de l'attracteur basée sur la série temporelle de l'IPC. Nous avons aussi effectué la plupart de ces mesures avec des attracteurs reconstruits sur la base des deux autres séries temporelles (les taux de défaut de cache L1 et L2). D'autre part, nous avons varié la fenêtre de moyennisation τ_{av} ($\tau_{av} = 10^6, 10^7$ or 10^8 instructions, voir 1). Toutes ces conditions ont donné des valeurs comparables et confirmé la présence de chaos déterministe. Enfin, nous avons étudié la performance lors de l'exécution de `gzip2` sur un fichier d'entrée d'un autre type. Tous les indicateurs testés ont là aussi confirmé la présence de chaos déterministe. Enfin, signalons que ce type de dynamique n'est pas spécifique à `gzip2`. Parmi les benchmarks Spec testés, nous avons retrouvé un comportement clairement chaotique pour d'autres programmes, comme `galgel` et `fma3d`, et obtenu des indications de sa présence (mais de façon non concluante) pour `gzip` et `ammp`.

L'évolution de la dynamique lors de l'exécution du programme `vpr` (un programme d'aide au design de circuits intégrés) est montré Figure 2A. En comparaison de `gzip2`, la dynamique en bien plus variable et ne montre pas vraiment de comportement régulier. La reconstruction de l'attracteur putatif par plongement de la série IPC a été ici aussi effectuée par la méthode des délais. La fig. 2B présente de graphe de récurrence seuilé correspondant. A la différence du graphe obtenu pour `gzip2`, le graphe de récurrence de `vpr` ne montre que des points isolés (pas de lignes diagonales, sauf près de la principale), dont la distribution est bien plus homogène. De plus, après génération de données "surrogates", les tests statistiques n'ont pu décider en faveur de la présence de non linéarité dans la série originale (non montré). La figure 2C montre les pentes locales des sommes de corrélation. Là encore en contraste fort avec les données obtenues pour `gzip2`, aucun régime de scaling n'est observé, quelque soit la gamme considérée pour la taille du voisinage ϵ ou la dimension de plongement m . Dans leur ensemble, ces résultats sont typiques des séries temporelles stochastiques. Ils montrent que, contrairement à `gzip2`, l'irrégularité et la grande variabilité observée lors de l'évolution de la dynamique de la performance pendant l'exécution de `vpr` ne proviennent pas de la présence de chaos déterministe, mais d'un processus pseudo-stochastique, ou en tout cas, de grande dimension. Parmi les benchmarks Spec que nous avons étudiés, un comportement similaire a été aussi clairement observé pour `art`, et suspecté pour d'autres programmes, comme `crafty`.

Notre dernier exemple concerne `applu`, une application de calcul scientifique. La simple inspection visuelle des séries temporelles correspondantes est suffisante pour se convaincre de la régularité et la périodicité des trois mesures de la performance (Figure 3A). Les spectres de puissance (Figure 3B) pour les trois mesures confirment cette régularité, avec une structure typique des signaux périodiques. Une fréquence majeure ($f \approx 1.6 \times 10^{-9}$ instructions correspondant à une période d' ≈ 0.6 milliards d'instructions) et ses harmoniques dominant le spectre et toute fréquence différente est quasiment inexistante. Ces résultats prouvent la présence de programmes dont la dynamique d'exécution est périodique et très régulière. En plus de `applu`, ce type de comportement a été observé pour d'autres benchmarks Spec, comme `apsi`.

3 Conclusion

Une des principales conclusions de cette étude est que la très grande variabilité observée durant la dynamique de performance de nombreux programmes peut être due à la présence de chaos déterministe. Ce résultat semble important parce qu'il révèle la grande complexité des mécanismes qui, dans les microprocesseurs, déterminent la performance. Néanmoins, il doit être manipulé avec soin. En particulier, il ne s'applique qu'à la performance instantanée, et non à d'autres aspects du fonctionnement du microprocesseur (en particulier, il n'indique en aucun cas que le résultat final du programme pourrait être imprévisible). D'autre part, ces résultats impliquent que prédire la performance de façon détaillée, sur la base d'un court échantillon de la dynamique pourrait être impossible. Néanmoins, là encore, ce type d'interprétation est à tempérer par le fait que l'amplitude des exposants de Lyapunov quantifie seulement les vitesses de divergence (ou de convergence) *moyennes* (sur tout l'attracteur étrange), alors que la prédictabilité peut varier grandement d'une partie à l'autre de l'attracteur. Il est donc possible de perdre la prédictabilité exponentiellement vite à certains endroits de la dynamique, pour la regagner plus tard. En revanche, prédire la dynamique de la performance des programmes au comportement pseudo-stochastique (comme vpr) semble plus facilement accessible. En particulier, la méthode dite des "surrogates" (utilisée pour tester la non-linéarité des séries temporelles) permet de générer des séries temporelles synthétiques (généralement générées pas de procédés de type moyenne mouvante auto-régressive [3]) dont les caractéristiques sont très proches de la série temporelle réelle. Il semble donc possible, pour ces programmes, d'obtenir une bonne estimation de la valeur moyenne de l'IPC ou du taux de défauts de cache L1, par exemple, durant l'exécution réelle du programme, en se basant sur une trace synthétique, générée à un coût computationnel faible. D'une façon générale, ces résultats montrent que la performance des microprocesseurs modernes lors de l'exécution de certains programmes forme un système suffisamment compliqué (voire complexe) pour que l'utilisation des méthodes récentes des sciences du non-linéaire et de la complexité, puisse nous en fournir de nouvelles perspectives de compréhension.

Références

1. G. E. MOORE, Cramming more components onto integrated circuits, *Electronics* **38**, 114-117 (1965).
2. D. BURGER, T. M. AUSTIN, S. BENNETT, The SimpleScalar Tool Set, Version 2.0, *Computer Architecture News*, **25** (3), 13-25 (1997).
3. R. HEGGER, H. KANTZ, T. SCHREIBER, Practical implementation of nonlinear time series methods : The TISEAN package, *CHAOS*, **9** (18), 413-434 (1999)
4. H. BERRY, D. GRACIA-PÉREZ, O. TEMAM, Chaos in computer performance, *CHAOS*, **16** (1), 013110 (2006)
5. C.-K. PENG, S. HAVLIN, H.E. STANLEY, A.L. GOLDBERGER, Quantification of scaling exponents and crossover phenomena in nonstationary heartbeat time series, *CHAOS*, **5**, 82-87 (1995)
6. G. RANGARAJAN, M. DING, Integrated approach to the assessment of long range correlation in time series data, *Phys. Rev. E*, **61**, 4991-5001 (2000)
7. F. TAKENS, *Detecting strange attractors in turbulence*, in *Dynamical Systems and Turbulence*, Lecture Notes in Math., pp. 366-381, Springer-Verlag (1981)
8. J.-P. ECKMANN, S. O. KAMPHORST, D. RUELLE, Recurrence Plots of Dynamical Systems, *Europhysics Lett.*, **5**, 973-977 (1987)
9. J.P. ZBILUT, N. THOMASSON, C.L. WEBBER, Recurrence quantification analysis as a tool for nonlinear exploration of nonstationary cardiac signals, *Med. Eng. Phys.*, **24**, 53-60 (2002)
10. P. GRASSBERGER, I. PROCACCIA, Measuring the strangeness of strange attractors, *Physica D*, **9**, 189-208 (1983)
11. H. KANTZ, A robust method to estimate the maximal Lyapunov exponent of a time series, *Phys. Lett. A*, **185**, 177-187 (1994)