

# Analyse et comparatif de benchmarks d'ordonnanceurs de tâches à l'heure du multi-cœur

Xavier de Rochefort

7 octobre 2013

L'ordonnanceur de tâches est au cœur de nos systèmes d'exploitation. Chargé de déterminer sur quel processeur et pour combien de temps chaque programme utilisateur ou tâche du système doit s'exécuter, son implémentation est déterminante pour les performances globales du système.

L'implémentation d'une politique d'ordonnement dans un système existant est une tâche complexe. L'étude de ses performances soulève la question du choix du type de charges à soumettre au système, et des métriques à en retirer.

En prenant pour point de départ les travaux de Kuz et al. [2] et Boyd et al. [1], et en vous documentant sur les problématiques liées à l'ordonnement sur les architectures parallèles [3, 4] il vous est demandé de proposer un protocole de test des performances de l'ordonneur d'une version récente du noyau Linux sur machines multi-cœurs.

**Mots clefs** : *scheduling, benchmark, performance analysis, processor architecture, task assignment, computer architecture*

## Références

- [1] S. Boyd-Wickizer, A. T. Clements, Y. Mao, A. Pesterev, M. F. Kaashoek, R. Morris, and N. Zeldovich. An analysis of linux scalability to many cores. In *Proceedings of the 9th USENIX conference on Operating systems design and implementation, OSDI'10*, page 1–8, Berkeley, CA, USA, 2010. USENIX Association.
- [2] I. Kuz, Z. Anderson, P. Shinde, and T. Roscoe. Multicore OS benchmarks : we can do better. In *Proceedings of the 13th USENIX conference on Hot topics in operating systems, HotOS'13*, page 10–10, Berkeley, CA, USA, 2011. USENIX Association.
- [3] S. Peter, A. Schüpbach, P. Barham, A. Baumann, R. Isaacs, T. Harris, and T. Roscoe. Design principles for end-to-end multicore schedulers. In *Proceedings of the 2nd USENIX conference on Hot topics in parallelism, HotPar'10*, page 10–10, Berkeley, CA, USA, 2010. USENIX Association.
- [4] S. Zhuravlev, S. Blagodurov, and A. Fedorova. Addressing shared resource contention in multicore processors via scheduling. In *Proceedings of the fifteenth edition of ASPLOS on Architectural support for programming languages and operating systems, ASPLOS XV*, page 129–142, New York, NY, USA, 2010. ACM.