

De nos jours, des processeurs à mémoire partagée cohérente ayant jusqu'à 100 cores intégrés sur la même puce sont une réalité et des processeurs many-cores ayant plusieurs centaines, voire, un millier de cores sont à prévoir prochainement. Dans ces architectures, la question de la localité du trafic lié aux miss de caches L1 (data, instruction et TLB) est primordiale à la fois pour passer à l'échelle et pour réduire la consommation électrique (énergie consommée par bit transféré). Notre thèse est que : (i) la gestion de la localité des accès mémoire doit être prise en compte au niveau du noyau du système d'exploitation et elle doit être effectuée d'une manière transparente aux applications utilisateur; et (ii) les noyaux monolithiques actuels sont incapables de renforcer la localité des accès mémoire des threads d'une même application parallèle, car la notion de threads dans ces noyaux est intrinsèquement inadaptée pour les processeurs many-cores. Par conséquent, nous pensons que la démarche suivie jusqu'à présent pour faire évoluer les noyaux monolithiques n'est pas suffisante et qu'il est impératif de mettre la question de la localité des accès mémoire au centre de cette évolution.

Pour prouver notre thèse, nous avons conçu et réalisé ALMOS (Advanced Locality Management Operating System), un système d'exploitation expérimental à base de noyau monolithique distribué. ALMOS dispose d'un nouveau concept de thread, nommé Processus Hybrides. Il permet à son noyau de renforcer, d'une manière transparente, la localité des accès mémoire liés à l'exécution de chaque thread. La gestion des ressources (cores et mémoires physiques) dans le noyau d'ALMOS est distribuée renforçant la localité des accès mémoire lors de la réalisation des services systèmes. La prise de décision concernant l'allocation mémoire, le placement des tâches et l'équilibrage de charge dans le noyau d'ALMOS est décentralisée, multi-critères et sans prise de verrou. Elle repose sur une infrastructure distribuée coordonnant d'une manière scalable l'accès aux ressources.

En utilisant le prototype virtuel précis au cycle et au bit près du processeur many-core TSAR, nous avons expérimentalement démontré que : (i) les performances (scalabilité et temps d'exécution) du schéma d'ordonnancement distribué du noyau d'ALMOS sur 256 cores dépassent celles des noyaux monolithiques existants; (ii) la réalisation répartie de l'appel système fork permet de passer à l'échelle ce service système sur 512 cores; (iii) le coût de la mise à jour de l'infrastructure distribué de prise de décisions du noyau d'ALMOS ne nécessite que 0.05% de la puissance de calcul totale du processeur TSAR; (iv) les performances (scalabilité, temps d'exécution et trafic distant) de la stratégie d'affinité mémoire du noyau d'ALMOS, nommé Auto-Next-Touch, dépassent celles des deux stratégies First-Touch et Interleave sur 64 cores; (v) le modèle de processus hybrides d'ALMOS permet de passer à l'échelle deux applications hautement multi-threads existantes sur 256 cores et une troisième sur 1024 cores; et enfin (vi) le couple ALMOS/TSAR (64 cores) offre systématiquement une bien meilleure scalabilité que le couple Linux/AMD (Interlagos 64 cores) pour 8 applications de classe HPC et traitement d'images.