El acceso a la memoria principal en los procesadores actuales supone un importante cuello de botella para las prestaciones, dado que los diferentes núcleos compiten por limitado ancho de banda de memoria, agravando la brecha entre las prestaciones del procesador y las de la memoria principal. Distintas técnicas atacan este problema, siendo las más relevantes el uso de jerarquías de caché multinivel y la prebúsqueda. Las cachés jerárquicas aprovechan la localidad temporal y espacial que en general presentan los programas en el acceso a los datos, para mitigar las enormes latencias de acceso a memoria principal. Para limitar el número de accesos a la memoria DRAM, fuera del chip, los procesadores actuales cuentan con grandes cachés de último nivel (LLC). Para mejorar su utilización y reducir costes, estas cachés suelen compartirse entre todos los núcleos del procesador. Este enfoque mejora significativamente el rendimiento de la mayoría de las aplicaciones en comparación con el uso de cachés privados más pequeños. Compartir la caché, sin embargo, presenta una problema importante: la interferencia entre aplicaciones. La prebúsqueda, por otro lado, trae bloques de datos a las cachés antes de que el procesador los solicite, ocultando la latencia de memoria principal. Desafortunadamente, dado que la prebúsqueda es una técnica especulativa, si no tiene éxito puede contaminar la caché con bloques que no se usarán. Además, las prebúsquedas interfieren con los accesos a memoria normales, tanto los del núcleo que emite las prebúsquedas como los de los demás.

Esta tesis se centra en reducir la interferencia entre aplicaciones, tanto en las caché compartidas como en el acceso a la memoria principal. Para reducir la interferencia entre aplicaciones en el acceso a la memoria principal, el mecanismo propuesto en esta disertación regula la agresividad de cada prebuscador, activando o desactivando selectivamente algunos de ellos, dependiendo de su rendimiento individual y de los requisitos de ancho de banda de memoria principal de los otros núcleos. Con respecto a la interferencia en cachés compartidos, esta tesis propone dos técnicas de particionado para la LLC, las cuales otorgan más espacio de caché a las aplicaciones que progresan más lentamente debido a la interferencia entre aplicaciones. La primera propuesta de particionado de caché requiere hardware específico no disponible en procesadores comerciales, por lo que se ha evaluado utilizando un entorno de simulación. La segunda propuesta de particionado de caché presenta una familia de políticas que superan las limitaciones en el número de particiones y en el número de vías de caché disponibles mediante la agrupación de aplicaciones en clústeres y la superposición de particiones de caché, por lo que varias aplicaciones comparten las mismas vías. Dado que se ha implementado utilizando los mecanismos para el particionado de la LLC que presentan algunos procesadores Intel modernos, esta propuesta ha sido evaluada en una máquina real.

Los resultados experimentales muestran que el mecanismo de prebúsqueda selectiva propuesto en esta tesis reduce el número de solicitudes de memoria principal en un 20%, cosa que se traduce en mejoras en la equidad del sistema, el rendimiento y el consumo de energía. Por otro lado, con respecto a los esquemas de partición propuestos, en comparación con un sistema sin particiones, ambas propuestas reducen la iniquidad del sistema en un promedio de más del 25%, independientemente de la cantidad de aplicaciones en ejecución, y esta reducción en la injusticia no afecta negativamente al rendimiento.