

Resumen

Los clústers de computadores representan una solución alternativa a los supercomputadores. En este tipo de sistemas, se suele restringir el espacio de direccionamiento de memoria de un procesador dado a la placa madre local. Restringir el sistema de esta manera es mucho más barato que usar una implementación de memoria compartida entre las placas. Sin embargo, las diferentes necesidades de memoria de las aplicaciones que se ejecutan en cada placa pueden dar lugar a un desequilibrio en el uso de memoria entre las placas. Esta situación puede desencadenar intercambios de datos con el disco, los cuales degradan notablemente las prestaciones del sistema, a pesar de que pueda haber memoria no utilizada en otras placas. Una solución directa consiste en aumentar la cantidad de memoria disponible en cada placa, pero el coste de esta solución puede ser prohibitivo.

Por otra parte, el hardware de acceso a memoria remota (RMA) es una forma de facilitar interconexiones rápidas entre las placas de un clúster de computadores. En trabajos recientes, esta característica se ha usado para aumentar el espacio de direccionamiento en ciertas placas. En este trabajo, la máquina base usa esta capacidad como mecanismo rápido para permitir al sistema operativo local acceder a la memoria DRAM instalada en una placa remota. En este contexto, una planificación de memoria eficiente constituye una cuestión crítica, ya que las latencias de memoria tienen un impacto importante sobre el tiempo de ejecución global de las aplicaciones, debido a que las latencias de memoria remota pueden ser varios órdenes de magnitud más altas que los accesos locales. Además, el hecho de cambiar la distribución de memoria es un proceso lento que puede involucrar a varias placas, así pues, el planificador de memoria ha de asegurarse de que la distribución objetivo proporciona mejores prestaciones que la actual. La presente disertación pretende abordar los asuntos mencionados anteriormente mediante la propuesta de varias políticas de planificación de memoria.

En primer lugar, se presenta un algoritmo ideal y una estrategia heurística para asignar memoria principal ubicada en las diferentes regiones de memoria. Adicionalmente, se ha diseñado un mecanismo de control de Calidad de Servicio para evitar que las prestaciones de las aplicaciones en ejecución se degraden de forma inadmisibles. El algoritmo ideal encuentra la distribución de memoria óptima pero su complejidad computacional es prohibitiva dado un alto número de aplicaciones. De este inconveniente se encarga

la estrategia heurística, la cual se aproxima a la mejor distribución de memoria local y remota con un coste computacional aceptable.

Los algoritmos anteriores se basan en *profiling*. Para tratar este defecto potencial, nos centramos en soluciones analíticas. Esta disertación propone un modelo analítico que estima el tiempo de ejecución de una aplicación dada para cierta distribución de memoria. Dicha técnica se usa como un predictor de prestaciones que proporciona la información de entrada a un planificador de memoria. El planificador de memoria usa las estimaciones para elegir dinámicamente la distribución de memoria objetivo óptima para cada aplicación que se esté ejecutando en el sistema, de forma que se alcancen las mejores prestaciones globales.

La planificación a granularidad más alta permite políticas de planificación más simples. Este trabajo estudia la viabilidad de planificar a nivel de granularidad de página del sistema operativo. Un entrelazado convencional basado en hardware a nivel de bloque y un entrelazado a nivel de página de sistema operativo se han tomado como esquemas de referencia. De la comparación de ambos esquemas de referencia, hemos concluido que solo algunas aplicaciones se ven afectadas de forma significativa por el uso del entrelazado a nivel de página. Las razones que causan este impacto en las prestaciones han sido estudiadas y han definido la base para el diseño de dos políticas de distribución de memoria basadas en sistema operativo. La primera se denomina *on-demand* (OD), y es una estrategia simple que funciona colocando las páginas nuevas en memoria local hasta que dicha región se llena, de manera que se beneficia de la premisa de que las páginas más accedidas se piden y se ubican antes que las menos accedidas para mejorar las prestaciones. Sin embargo, ante la ausencia de dicha premisa para algunos de los benchmarks, OD funciona peor. La segunda política, denominada *Most-accessed in-local* (Mail), se propone con el objetivo de evitar este problema.