

## *Resum*

Els clústers d'ordinadors representen una solució alternativa als superordinadors. En aquest tipus de sistemes, se sol restringir l'espai d'adreçament de memòria d'un processador donat a la placa mare local. Restringir el sistema d'aquesta manera és molt més barat que usar una implementació de memòria compartida entre les plaques. No obstant això, les diferents necessitats de memòria de les aplicacions que s'executen en cada placa poden donar lloc a un desequilibri en l'ús de memòria entre les plaques. Aquesta situació pot desencadenar intercanvis de dades amb el disc, els quals degraden notablement les prestacions del sistema, tot i que puga haver-hi memòria no utilitzada en altres plaques. Una solució directa consisteix a augmentar la quantitat de memòria disponible en cada placa, però el cost d'aquesta solució pot ser prohibitiu.

D'altra banda, el hardware d'accés a memòria remota (RMA) és una forma de facilitar interconnexions ràpides entre les plaques d'un clúster de computadors. En treballs recents, aquesta característica s'ha usat per a augmentar l'espai d'adreçament en certes plaques. En aquest treball, la màquina base fa servir aquesta capacitat com a mecanisme ràpid per a permetre al sistema operatiu local accedir a la memòria DRAM instal·lada en una placa remota. En aquest context, una planificació de memòria eficient constitueix una qüestió crítica, ja que les latències de memòria tenen un impacte important sobre el temps d'execució global de les aplicacions, pel fet que les latències de memòria remota poden ser diversos ordres de magnitud més altes que els accessos locals. A més, el fet de canviar la distribució de memòria és un procés lent que pot involucrar a diverses plaques, així doncs, el planificador de memòria ha d'assegurar-se que la distribució objectiu proporciona millors prestacions que l'actual. La present dissertació pretén abordar els assumptes esmentats anteriorment mitjançant la proposta de diverses polítiques de planificació de memòria.

En primer lloc, es presenta un algorisme ideal i una estratègia heurística per a assignar memòria principal situada en les diferents regions de memòria. Addicionalment, s'ha dissenyat un mecanisme de control de Qualitat de Servei per tal d'evitar que les prestacions de les aplicacions en execució es degraden de forma inadmissible. L'algorisme ideal troba la distribució de memòria òptima però la seua complexitat computacional és prohibitiva donat un alt nombre d'aplicacions. D'aquest inconvenient s'encarrega

l'estratègia heurística, la qual s'aproxima a la millor distribució de memòria local i remota amb un cost computacional acceptable.

Els algorismes anteriors es basen en *profiling*. Per tractar aquest defecte potencial, ens centrem en solucions analítiques. Aquesta dissertació proposa un model analític que estima el temps d'execució d'una aplicació donada per a certa distribució de memòria. Aquesta tècnica s'usa com un predictor de prestacions que proporciona la informació d'entrada a un planificador de memòria. El planificador de memòria usa les estimacions per a triar dinàmicament la distribució de memòria objectiu òptima per a cadascuna de les aplicacions que s'estan executant en el sistema, de manera que s'aconseguisquen les millors prestacions globals.

La planificació a granularitat més alta permet polítiques de planificació més simples. Aquest treball estudia la viabilitat de planificar a nivell de granularitat de pàgina del sistema operatiu. Un entrellaçat convencional basat en hardware a nivell de bloc i un entrellaçat a nivell de pàgina de sistema operatiu s'han pres com a esquemes de referència. De la comparació de tots dos esquemes de referència, hem conclòs que solament algunes aplicacions es veuen afectades de forma significativa per l'ús de l'entrellaçat a nivell de pàgina. Les raons que causen aquest impacte en les prestacions han estat estudiades i han definit la base per al disseny de dues polítiques de distribució de memòria basades en sistema operatiu. La primera es denomina *on-demand* (OD), i és una estratègia simple que funciona col·locant les pàgines noves en memòria local fins que aquesta regió s'omple, de manera que es beneficia de la premissa que les pàgines més accedides es demanen i se situen abans que les menys accedides per tal de millorar les prestacions. No obstant això, davant l'absència d'aquesta premissa per a alguns dels benchmarks, OD funciona pitjor. La segona política, denominada *Most-accessed in-local* (Mail), es proposa amb l'objectiu d'evitar aquest problema.