Prebúsqueda Hardware en Aplicaciones Comerciales

por

Javier Picorel Obando

Sometido como cumplimiento parcial de los requerimientos para la licenciatura de

Ingeniería Informática

en la

ESCUELA DE INGENIERÍA Y ARQUITECTURA

Agosto de 2011

Autor: Javier Picorel Obando

Director - EPFL: Babak Falsafi

Laboratorio - EPFL: Parallel Systems Architecture Laboratory

Director - EINA: José Luis Briz

Departamento - EINA: Informática e Ingeniería de Sistemas

Prebúsqueda Hardware en Aplicaciones Comerciales

Resumen

Las técnicas de prebúsqueda intentan solventar la diferencia entre la latencia de acceso a memoria y el tiempo de ciclo del procesador. Esta diferencia, conocida como “Memory Gap” o “Memory Wall”, es de dos órdenes de magnitud y continúa aumentando. Junto a factores térmicos y de alimentación constituye la principal la limitación para incrementar el rendimiento de los procesadores actuales. Técnicas propuestas recientemente aprovechan el hecho de que las secuencias de referencias a memoria se repiten a lo largo del tiempo y además en el mismo orden. El talón de Aquiles de estos prebuscadores es la imposibilidad de predecir accesos a memoria para direcciones que no se han visitado anteriormente. Este coste de oportunidad se magnifica en aplicaciones donde una gran mayoría del conjunto de datos del programa es leía una sola vez. Existe otro tipo de técnicas basadas en la observación de que los programas acceden al espacio de direcciones mediante patrones comunes alineados a regiones de memoria, y son predecibles mediante correlación en código. Una de las limitaciones más importantes de este tipo de prebuscadores es la necesidad de un primer acceso en la región para comenzar la predicción de los bloques que van a ser referenciados dentro de la región, el cual no puede ser amortizado con los bloques prebuscados correctamente más allá del tamaño de la región, que es finito. Otro es el mecanismo en sí, antes explicado que inicia las predicciones. Debido a que realiza la predicción para toda la región, si una predicción es incorrecta perdemos toda la oportunidad de predecir correctamente dentro de la misma. El estado del arte en prebúsqueda de datos correlaciona temporalmente los accesos que inician predicciones dentro de las regiones.

Este PFC muestra las ineficiencias de los prebuscadores más avanzados y propone dos técnicas para solventarlas: (1) mecanismo de predicción por secuencia de delta direcciones – iniciando las predicciones mediante la última secuencia de deltas observada en la secuencia de referencias a memoria, y (2) predicción por acceso – cada vez que el procesador envía una petición de datos a la jerarquía de memorias, se realiza una predicción. Los resultados indican que la correlación a través de deltas puede predecir patrones recurrentes de acceso a direcciones de memoria nunca antes referenciadas mejorando la predicción temporal de accesos a memoria, y que las predicciones por acceso mejoran los resultados obtenidos por los prebuscadores basados en regiones eliminando la limitación de coste de oportunidad de las predicciones incorrectas dentro de la región.

Contexto y Objetivos del PFC

El Grupo de Architectura de Computadores de Zaragoza (GAZ) ha publicado recientemente un prebuscador hardware de datos orientado a cargas de trabajo de sobremesa / workstation. El objetivo del proyecto es su evaluación en cargas de trabajo comerciales de tipo servidor. Para ello el prebuscador deberá ser implementado en un entorno de simulación del tipo procesador de memoria compartida multi-núcleo, simulando aplicaciones comerciales de servidores como Oracle Enterprise Server, IMB DB2 y Apache ejecutando cargas de trabajo TPC-C, TPC-H y SPECWeb. El único entorno de simulación en el que actualmente puede llevarse a cabo esta tarea es FLEXUS. Este simulador permite ejecutar aplicaciones comerciales sin ninguna modificación en un tiempo de ejecución asequible. FLEXUS ha sido desarrollado en el Parallel Systems Architecture Laboratory (PARSA), perteneciente a la École Polytechnique Federal de Lausanne (EPFL). El objetivo del PFC es la evaluación del prebuscador con este tipo de cargas de trabajo y proponer soluciones para aumentar su rendimiento si los resultados con el prebuscador base no son satisfactorios.

Con esta idea principal y tras la consulta de la literatura existente sobre prebúsqueda hardware en aplicaciones comerciales se han desarrollado las dos ideas que se presentan como resultado del PFC, *Delta Sequence Spatial Memory Streaming* (DS-SMS) y *Delta Sequence Temporal Memory Streaming* (DS-TMS).

Este proyecto ha sido realizado en el laboratorio PARSA de la École Polytechnique Federal de Lausanne y ha sido escrito y presentado originalmente en Inglés como tesis final de máster en la misma, obteniendo la calificación más alta.

ÍNDICE

1. INTRODUCCIÓN

Las aplicaciones comerciales son un auténtico desafío para los arquitectos de computadores que intentan diseñar chips eficientes para servidores de alto rendimiento. Estas aplicaciones utilizan complejos algoritmos multi-hilo, estructuras de datos irregulares, enormes conjuntos de datos y de instrucciones que limitan el rendimiento e imponen un cuello de botella en el sistema. Estudios anteriores han demostrado que entre un medio y dos tercios del tiempo total de ejecución, el procesador está esperando a recibir datos de la jerarquía de memoria [1].

Innovaciones tecnológicas acaecidas en el campo de la microarquitectura como la ejecución especulativa y las caches no bloqueantes no pueden mejorar el rendimiento global del sistema si los programas presentan un flujo de control irregular e impredecible y si el paralelismo a nivel de instrucción es casi inexistente debido a la frecuente presencia de instrucciones de memoria dependientes. Una forma de incrementar el rendimiento del sistema de memoria es paralelizar los accesos a la misma para enmascarar la latencia de las operaciones de memoria. Otra forma de incrementar el paralelismo a nivel de memoria es la técnica del multi-hilo. Sin embargo, para aplicar esta última es necesario que haya suficientes hilos disponibles para enmascarar la latencia de acceso a memoria, cosa que es posible que no ocurra en las aplicaciones comerciales [2]. Existen varias técnicas hardware y propuestas de investigación que intenta encontrar fragmentos de código independiente en aplicaciones de un solo hilo para ejecutarlas en hilos diferentes, como la técnica de *Dynamic Multithreading* de Intel [3]. Por desgracia, aunque este tipo de técnicas han resultado ser útiles en aplicaciones con un flujo de control predecible como las cargas de trabajo SPEC, no son factibles para las cargas de trabajo de servidores.

Otro enfoque muy bien estudiado es el de incrementar el paralelismo a nivel de memoria prediciendo los accesos que el procesador va a realizar antes que estos sean realizados. Hay gran cantidad de estudios que han conseguido solapar la latencia de memoria en aplicaciones científicas y de escritorio, aunque sin embargo son muy ineficientes cuando su uso se extrapola a aplicaciones comerciales debido a la complejidad de los accesos a memoria [4]. Un grupo más reciente de prebuscadores tienen como objetivo capturar patrones de acceso mucho más complejos obteniendo resultados muy prometedores en conductas de acceso altamente irregulares pero repetitivas. Una clasificación directa de estas técnicas separan estos prebuscadores en dos tipos, temporal y espacial. El primer grupo de prebuscadores [5, 6] se apoyan en el hecho de que los accesos a memoria tienden a repetirse a lo largo del tiempo y en el mismo orden, siendo más probables los patrones aparecidos más recientemente que los más antiguos. La principal limitación de

estos prebuscadores es la imposibilidad de predecir accesos a direcciones que nunca antes se han visitado, haciendo que la oportunidad para capturar los patrones este limitada a la recurrencia y estabilidad de las direcciones accedidas. Los prebuscadores espaciales [2, 7] por su parte, observan patrones alineados a regiones del espacio de direcciones y permiten extrapolar predicciones a otras regiones a su vez. Aunque los prebuscadores espaciales son más precisos que los temporales debido a que los patrones se repiten de forma más estable en cada región que en todo el espacio de direcciones y que permiten predecir direcciones nunca antes visitadas, tienen dos importante limitaciones. Primero, es necesario un acceso a la región para comenzar la predicción. Segundo, la predicción es realizada para toda la región, perdiendo toda la oportunidad de predecir correctamente los accesos dentro de la región si la predicción es incorrecta. El estado del arte en prebúsqueda de datos correlaciona temporalmente los accesos a regiones que producen predicciones espaciales.

En este PFC proponemos:

- **Delta direcciones para la predicción de secuencias de acceso temporal (DS-TMS):** Para solventar las ineficiencias de la predicción temporal permitiendo predecir accesos a direcciones no visitadas.

- **Predicciones por acceso en la predicción espacial (DS-SMS):** Para eliminar el coste oportunidad que aparece al predecir una única vez por región espacial.

Estudios recientes han demostrado que utilizar secuencias de delta direcciones producidas por la misma instrucción es útil para predecir patrones de accesos a memoria en cargas de trabajo SPEC. Desafortunadamente, esta técnica llamada Prefetcher based on Diferencial Finite Context Machine (PDFCM) no presenta buenos resultados cuando los accesos a memoria son muy complejos como en las aplicaciones comerciales para servidores.

Utilizando la idea de las delta direcciones y la oportunidad mostrada en otras técnicas enfocadas a cargas de trabajo comerciales como *Temporal Memory Streaming* (TMS) y *Spatial Memory Streaming* (SMS) se han desarrollado dos técnicas. La primera, denominada *Delta Sequence Temporal Memory Streaming* (DS-TMS) obtiene deltas de las secuencias de acceso a memoria correlacionadas temporalmente. La segunda, denominada *Delta Sequence Spatial Memory Streaming* (DS-SMS) divide el espacio de direcciones en regiones de memoria y extrae las deltas dentro de cada una de las regiones.

La técnica conocida como TMS se basa en observar que las referencias a memoria se repiten a lo largo del tiempo y que existe localidad en estos accesos ya que los más recientes tienen más probabilidad de repetirse en un futuro que los más antiguos. El

problema es que es incapaz de predecir accesos a memoria que no se hayan producido antes. DS-TMS soluciona este problema almacenando los patrones en delta direcciones relativas en vez de en direcciones globales.

SMS por su parte, aprovecha que los programas acceden repetitivamente con una disposición común a múltiples regiones del espacio de direcciones. Una limitación de esta técnica es que necesita un primer acceso a la región para comenzar la predicción de toda la misma y al ser la región de tamaño finito, ese acceso no puede ser amortizado idealmente más allá de la región. Otra limitación es que la predicción de patrones se realiza para toda la región tras el primer acceso a la misma y en consecuencia, toda la predicción es errónea para la región si esta es incorrecta. Mediante DS-SMS conseguimos realizar predicciones por acceso a la región con lo que si predecimos mal una secuencia de accesos podemos enmendarlo en el siguiente. Además debido a la forma en la que obtenemos la disposición de los accesos en la región podemos aumentar el tamaño de la región y amortizar las predicciones realizadas dentro de la región.

A partir de una simulación funcional de un procesador multi-núcleo con protocolo de coherencia de caches se obtiene que: DS-SMS consigue mejor cobertura que SMS y que DS-TMS permite solventar la limitación de TMS prediciendo patrones en direcciones nunca antes visitadas.

El talón de Aquiles de ambas técnicas es que a día de hoy tienen una complejidad en hardware demasiado elevada para su implementación en procesadores actuales.

El resto de este PFC está organizado de la siguiente forma. La sección 2 muestra la planificación llevada a cabo. La sección 3 explica las ineficiencias de la predicción temporal y espacial. En la sección 4 y 5 se exponen las ideas que hay detrás de las contribuciones que se realizan en este proyecto, en la predicción temporal y espacial respectivamente. El diseño propuesto se presenta en la sección 6. La metodología y la evaluación son presentadas en la sección 7. Finalmente la sección 8 presenta otros trabajos relacionados con el temario del proyecto y la sección 9 plasma las conclusiones finales del PFC.

2. PLANIFICACIÓN

La planificación se presentará mediante un diagrama PERT para conocer además las dependencias entre las tareas principales entre las que se han dividido el proyecto.

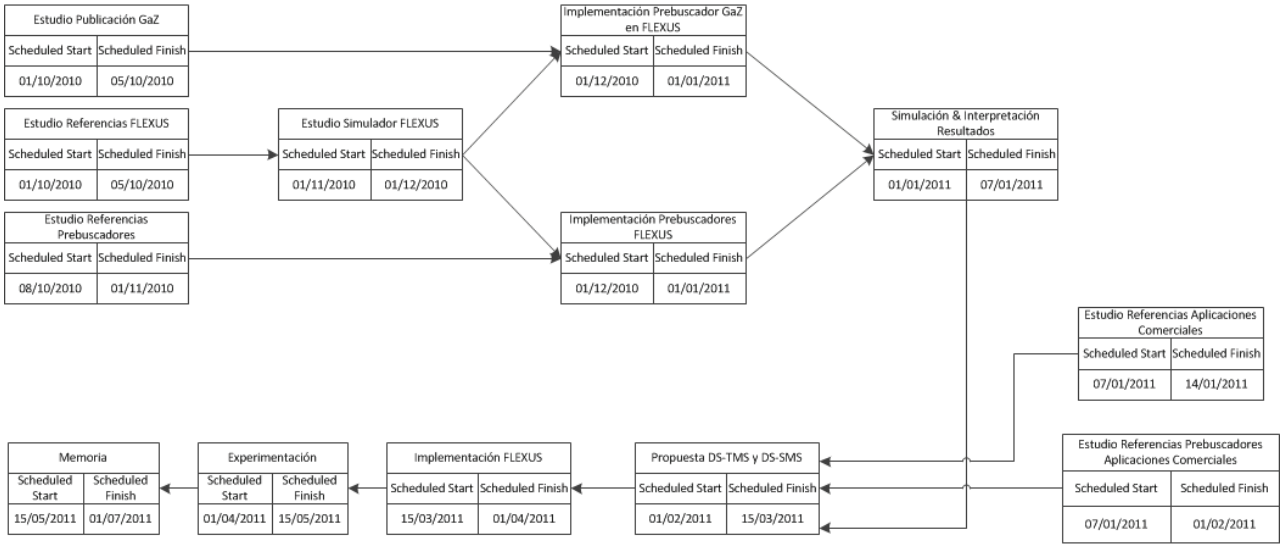

DIAGRAMA 1. Diagrama PERT de planificación.

3. ANTECEDENTES

Básicamente, existen dos grupos de prebuscadores diferenciados según en qué observación del comportamiento de las referencias a memoria se basen. La siguiente clasificación diferencia entre prebuscadores temporales y espaciales. La siguiente clasificación diferencia entre secuencias de acceso temporal y espacial que por supuesto referencia a cada uno de los tipos de prebuscadores indicando su comportamiento, oportunidades e ineficiencias.

3.1 Secuencias de acceso temporal

La intuición detrás de esta idea es la que los programas tienden a acceder a las estructuras de datos de una forma repetitiva. Otra observación es que la probabilidad de que dichos accesos sean repetidos en un tiempo cercado es mayor cuanto más recientes hayan sido, existe localidad temporal. Extrayendo secuencias de referencias a memoria temporales las largas cadenas de instrucciones de acceso a memoria dependientes, como en estructuras de persecución de punteros, se vuelven predecibles. Por ejemplo, los árboles B+ son estructuras de datos omnipresentes que permiten un tiempo de inserción, eliminación y búsqueda muy reducido. Hay un ejemplo directo que explica las secuencias de acceso temporal cuando se acceden a este tipo de estructuras. La Figura 1 muestra un ejemplo de un árbol B+.

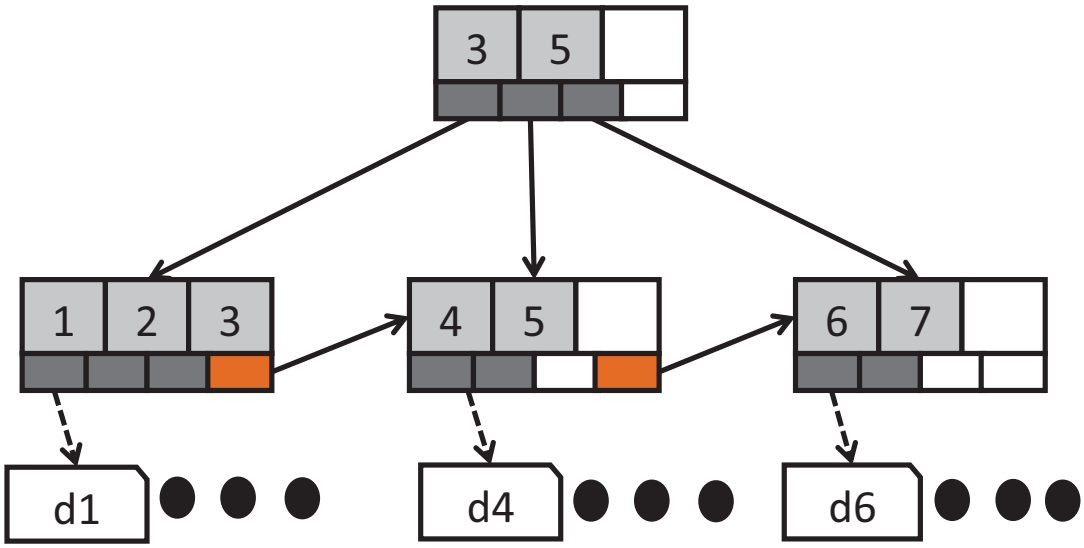

FIGURA 1. Estructuras de un árbol B+.

Una lista ordenada de claves por nodo, punteros en las hojas al siguiente nodo para el acceso secuencial de datos y punteros a los datos referenciados por las claves en cada una de las hojas.

Digamos que el programa necesita realizar un acceso secuencial a los registros apuntados por las claves del árbol, de una clave hasta otra. Para ello realizará una búsqueda binaria en cada uno de los nodos del árbol para encontrar la hoja que alberga la primera clave que apunta al registro con la clave más pequeña de las que buscábamos. Acto seguido atravesará las hojas del árbol aprovechando los punteros que concatenan las hojas realizando una búsqueda secuencial hasta la última clave que queríamos acceder. Como los datos apuntados por las hojas no están almacenados secuencialmente esto lleva a una secuencia de accesos a memoria temporal, pudiéndose repetir en un futuro próximo.

3.2 Temporal Memory Streaming

El prebuscador *Temporal Memory Streaming* (TMS) fue propuesto en [5] y extrae las secuencias de referencias a memoria temporales almacenando los fallos de cache en un gran buffer circular. Tras cada fallo, TMS localiza la última ocurrencia de dicha dirección en el buffer y pide a la jerarquía de memoria los bloques con las direcciones que suceden a dicha dirección en el buffer. A medida que los bloques son referenciados por el procesador, TMS continúa leyendo direcciones del buffer para sincronizar la velocidad con

la que la aplicación está referenciando bloques de memoria. La Figura 2 muestra el diagrama de funcionamiento de TMS.

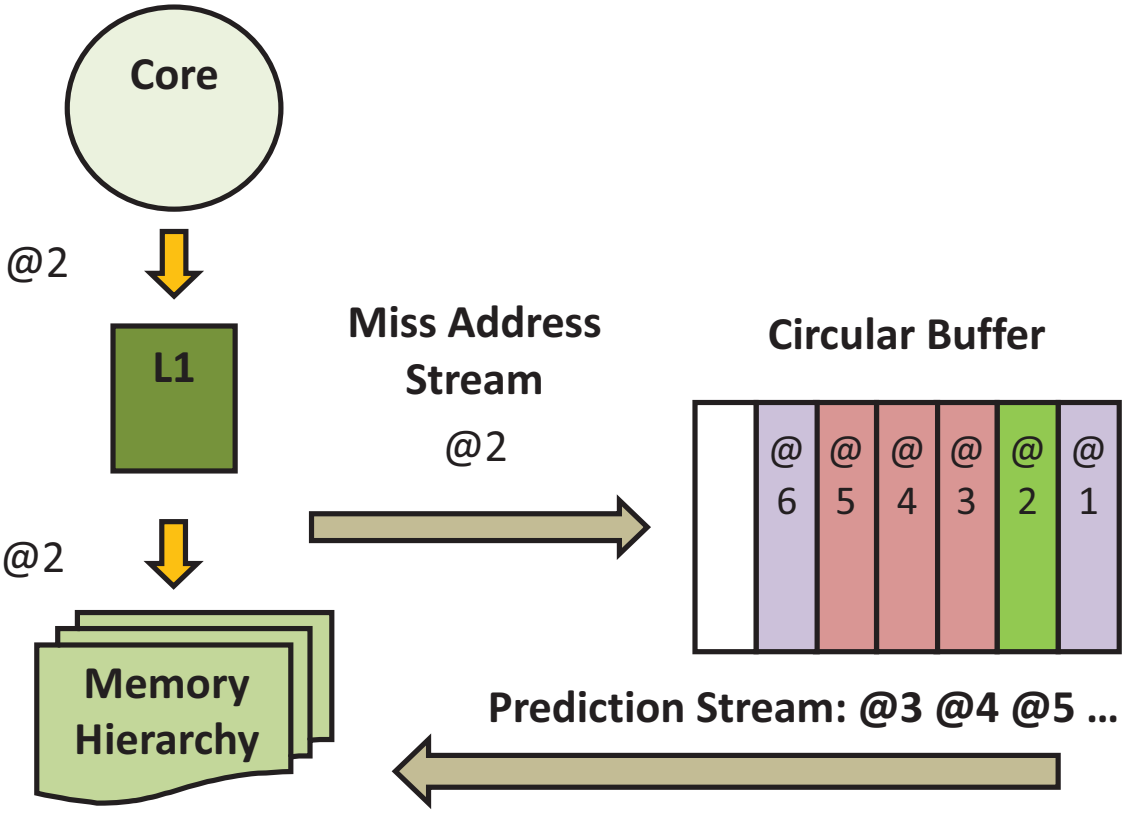

FIGURE 2. Diagrama en alto nivel del funcionamiento de TMS.

La secuencia de referencias a memoria es almacenada en un buffer circular. Cada vez que un dirección falle en cache (L1), la entrada más reciente que contiene la dirección en el buffer es localizada y las dirección sucesivas a esta son referenciadas a la jerarquía de memoria.

TMS es capaz de predecir largas cadenas de accesos a memoria dependientes (estructuras de persecución de punteros) y como las secuencias son generalmente largas [5] (cientos de fallos), el coste de empezar una nueva secuencia es amortizado con los sucesivos aciertos en cache. No obstante, una limitación existente es la imposibilidad de predecir secuencias de acceso que no se hayan visitado anteriormente llevando a un incremento en rendimiento despreciable si el porcentaje de datos referenciado una única vez por el programa es alto.

3.3 Secuencias de acceso espacial

Este fenómeno es debido a que la disposición de los accesos a memoria se repite de acuerdo a una dirección base, debido a que las estructuras de datos tienen una estructura común y la mayoría de las ocasiones en las que son accedidas existe una probabilidad elevada de que los mismos elementos sean accedidos. Por ejemplo, las páginas en un sistema de gestión de bases de datos (DMBS) comparten una estructura común la cual es atravesada de la misma forma cada vez que una página es leída o modificada. La Figura 3 muestra una página de un DBMS.

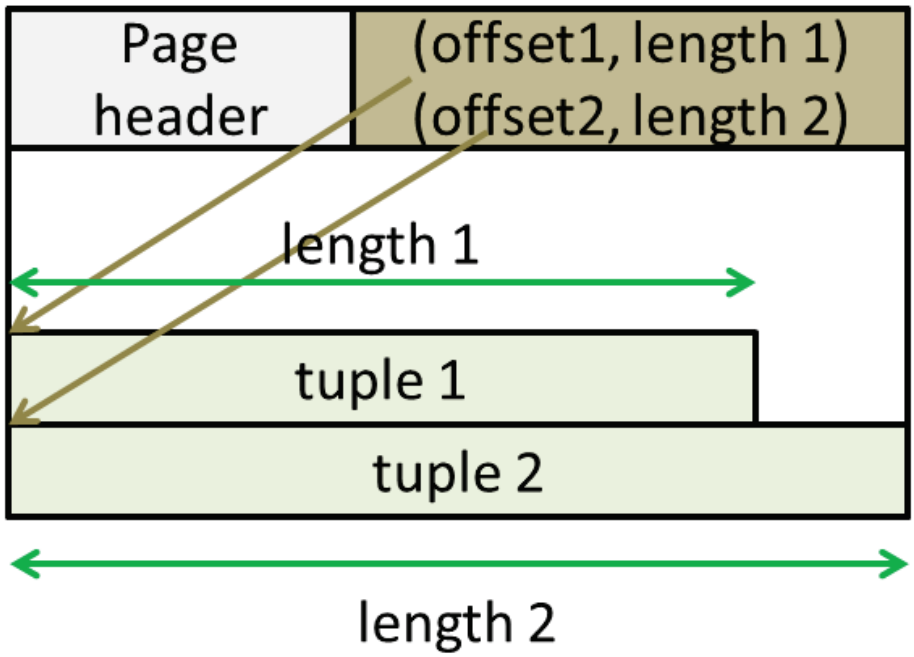

Figura 3. Ejemplo de correlación espacial en páginas de una DBMS.

En los DBMSs las tuplas son identificadas por (ID Página, offset). Debido a que utilizan para todas las páginas una estructura común, antes de insertar una nueva tupla o de recorrer una tabla, patrones espaciales comunes aparecen en cada uno de este tipo de accesos.

En los sistemas gestores de bases de datos modernos las tuplas están distribuidas a través de tablas de bases de datos y son identificadas por el identificador de la página y el offset dentro de la página donde esta tupla está almacenada. Para que una nueva tupla sea

almacenada el gestor tiene que mirar en las páginas donde hay espacio libre mirando los offsets y las longitudes de las tuplas dentro de las páginas. Para leer o modificar las tuplas de una tabla, se realiza el mismo proceso debido a como están estructuradas las páginas. Esto tipo de accesos conllevan unos patrones alineados a regiones del espacio de direcciones, aunque estas referencias aun no hayan sido nunca antes realizadas tienen la misma disposición en varias regiones

3.4 Spatial Memory Streaming

El prebuscador Spatial Memory Streaming (SMS) publicado en [7], aprovecha los patrones recurrentes de acceso a memoria que atraviesan múltiples regiones espaciales. Almacena patrones espaciales observando los accesos a la cache L1. Cuando un región es accedida por primera vez, SMS busca en los patrones almacenados utilizando alguna información como índice (contador del programa de la instrucción, dirección que referencia, …) para predecir los bloques que se van a pedir de la región a la que estamos accediendo en este primer acceso. La Figura 4 ilustra el comportamiento de SMS. El espacio de direcciones está dividido en regiones que son usadas por SMS para almacenar los patrones de acceso que se producen dentro de las mismas. SMS utiliza un vector de bits tan grande como el número de bloques que haya en una región (una región de 2KB con un tamaño de bloque de 64B tendrá vectores de 32 bits), almacenando la información de la siguiente manera – 0 bloque no referenciado, 1 bloque referenciado. Estos patrones se almacenan en una tabla indexada mediante el contador del programa de la instrucción que realizó el primer acceso que generó el registro del patrón almacenado. Cada vez que hay un acceso a una región, SMS busca si hay alguna entrada en la tabla de patrones almacenados que coincida con el contador del programa del acceso. Si existe, se utilizada la dirección base de la región a la que estamos accediendo y el vector de bits para pedir a la cache L1 los bloques marcados como referenciado.

SMS permite predecir patrones de acceso comunes que transcurren a lo largo de varias regiones, pero al contrario que TMS, las estructuras de seguimiento de punteros son imposibles de predecir ya que las estructuras de datos dinámicas pueden almacenarse en cualquier parte y de forma distinta en cualquier región del espacio de direcciones, sin existir ninguna correlación espacial entre punteros de dos regiones diferentes. Existen otras dos limitaciones en este tipo de prebuscadores. Primero, siempre se necesita un primer acceso a la región para empezar las predicciones, y como la región es finita, éste no puede ser amortizado más allá del tamaño de la región. Segundo, debido a que las predicciones se realizan tras el primer acceso para toda la región, si la predicción es

errónea perderemos toda la oportunidad para predecir correctamente los bloques que serán referenciados dentro de la región. La figura 4 presenta el funcionamiento general de SMS a grandes rasgos.

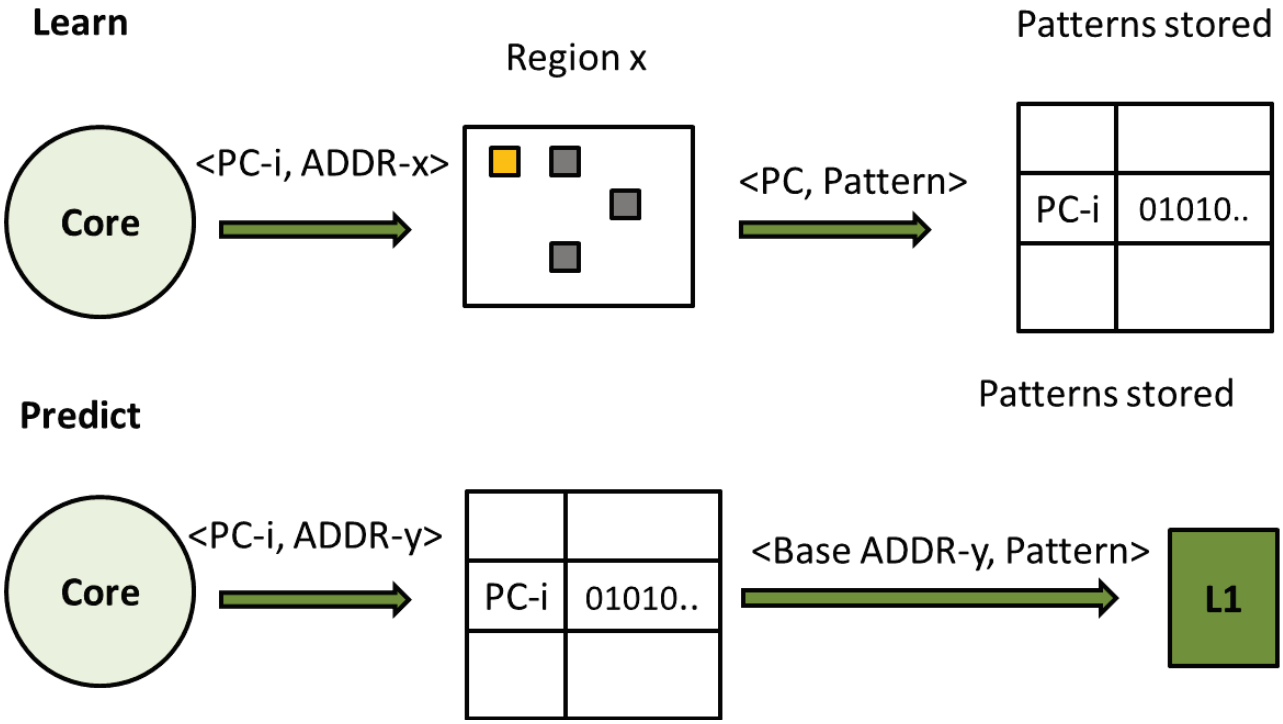

FIGURA 4. Diagrama de alto nivel del funcionamiento de SMS.

En SMS los patrones son capturados dentro de regiones espaciales y almacenados en una tabla, utilizando correlación en código. En el primer acceso a la región la tabla es indexada con el PC de la instrucción que ha provocado el acceso. Si existe la entrada, la predicción comienza usando como la dirección base, la dirección de la región donde se ha producido este primer acceso y como offsets la posición de los bits del vector de bits que aparezcan referenciados.

3.5 Secuencias de acceso espacio-temporal

El estado del arte actual en prebúsqueda de datos [2], fusiona las ideas propuestas para la captura de patrones en secuencias de acceso temporal y espacial correlacionando los

accesos que producen predicciones espaciales en las regiones, temporalmente como en TMS. Esto permite eliminar los fallos producidos por los accesos que generan predicciones espaciales si están relacionados temporalmente. Un ejemplo que deriva en este tipo de patrón de acceso es la lectura de las tuplas de una tabla que está distribuida en múltiples páginas, almacenadas de forma consecutiva en el buffer de páginas. Este recorrido de páginas conlleva una secuencia de acceso temporal predecible, de los primeros accesos a cada página.

3.6 Spatio-Temporal Memory Streaming

El prebuscador *Spatio-Temporal Memory Streaming* (STeMS) [2], es el actual estado del arte en prebúsqueda de datos y el pionero en esta idea de explotar las secuencias de acceso espacio-temporales. La Figura 5 presenta a grandes rangos el funcionamiento de STeMS.

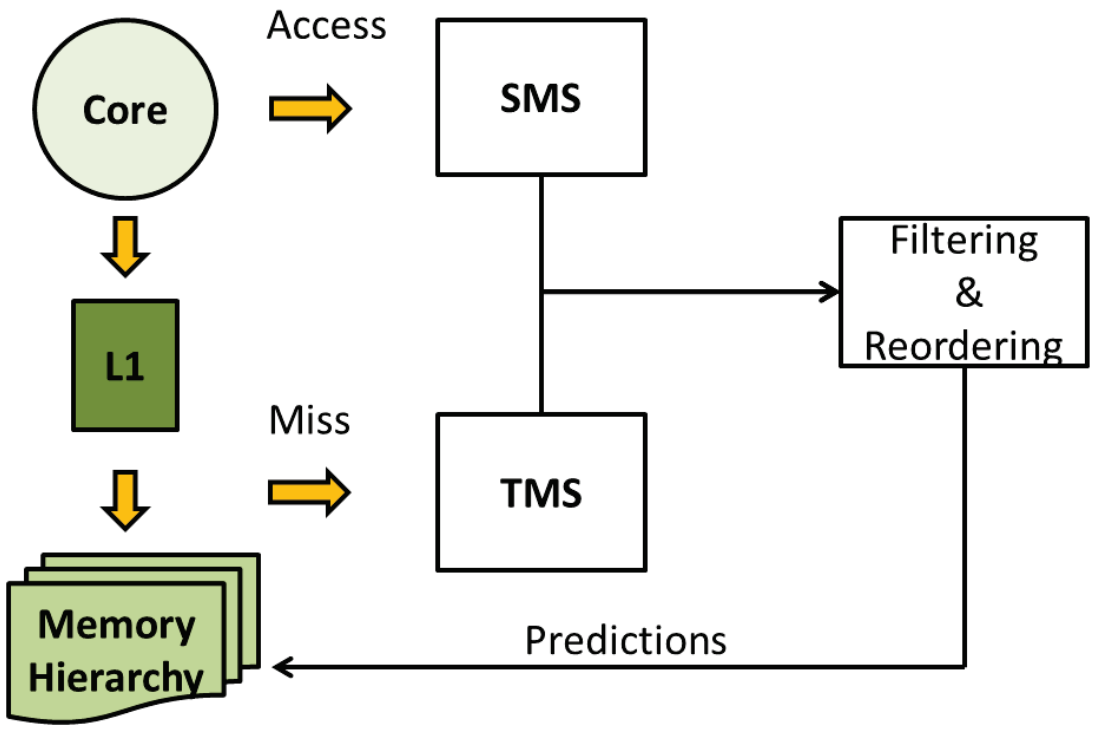

FIGURA 5. Diagrama de funcionamiento de STeMS.

Las predicciones espaciales y temporales son ordenadas a la vez para reconstruir la secuencia de accesos original.

SMS registra los patrones dentro de las regiones observando las referencias realizadas por el procesador a la cache L1, mientras que el predictor temporal (TMS) almacena los accesos realizados por L1 para almacenar la secuencia de fallos. Los fallos almacenados por TMS son filtrados y solamente se guardan los que no son predecibles de forma espacial por SMS. Para reconstruir la secuencia de accesos se reordenan las predicciones espaciales y temporales.

4. Delta direcciones

Líneas de investigación anteriores [5, 6] han demostrado que hay muchas referencias a memoria que tienden a accederse de forma conjunta y en el mismo orden, habiendo localidad temporal en estos accesos. Una de las mayores limitaciones de este tipo de técnicas es la imposibilidad de predecir secuencias de acceso que referencien direcciones nunca antes visitadas. Una forma de eliminar la atadura entre secuencias de acceso y direcciones, es calcular la diferencia entre direcciones consecutivas de los patrones. Esta idea, introducida la primera vez en [7] fue utilizada para la predicción en el TLB. La Figura 6 presenta una secuencia de direcciones y su secuencia en delta direcciones.

Address	5	10	15	17	19	49	79
Address Deltas	5	5	2	2	30	30	

FIGURA 6. Ejemplo de una secuencia de direcciones y su correspondiente secuencia de delta direcciones.

Todas las secuencias de direcciones pueden ser definidas como la primera dirección de la secuencia y las diferencias (delta direcciones) entre direcciones consecutivas.

Cualquier patrón puede ser descrito utilizado delta direcciones y éstas no están fijas a una dirección determinada. Los patrones que aparecen en otra dirección pero con la misma distancia entre direcciones pueden ser correctamente predichos. Esto ocurre cuando un procesador está atravesando diferentes estructuras de datos con un patrón de acceso común. El caso de observación es el mismo que para el predictor espacial debido a las secuencias temporales que puede aparecer antes de escanear o modificar una página en un sistema gestor de bases de datos. En este ejemplo no hay que llevar a error, no estamos hablando de regiones ni de predictores espaciales, simplemente a través de las delta direcciones intentamos encontrar secuencias de acceso temporal en estructuras de datos con una disposición de acceso idéntica, lo que es posible con SMS pero no con TMS. Tampoco hay que confundirse y pensar que estamos contradiciendo el hecho de que con los predictores espaciales no se pueden predecir las largas cadenas de accesos dependientes como en estructuras de datos con persecución de punteros. Estamos simplemente con esta idea permitiendo a los predictores temporales predecir accesos

hasta ahora solo posibles en los predictores espaciales. La figura 7 indica este comportamiento temporal.

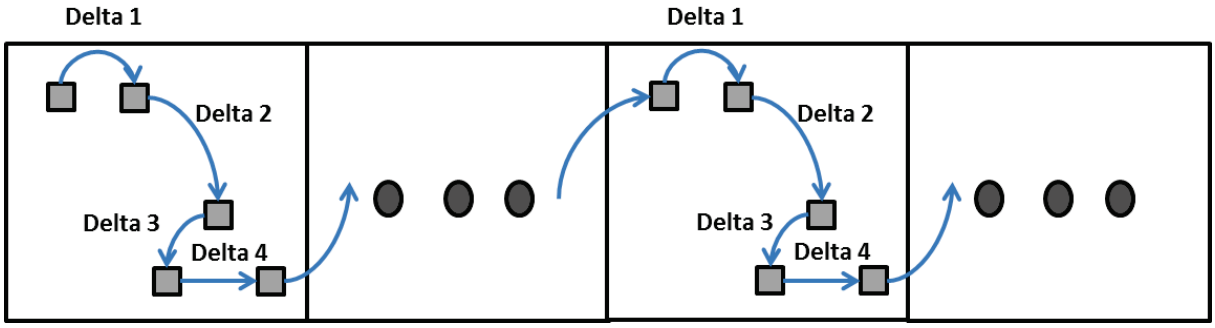

FIGURA 7. Secuencia de referencias a memoria repetitivas.

Si accedemos a las estructuras de datos de forma análoga, las delta direcciones son capaces de predecir accesos a direcciones nunca antes visitadas.

Es intuitivo pensar que aunque las delta direcciones pueden correlacionar patrones de direcciones nunca antes visitadas, una sola delta no es capaz de distinguir entre secuencias de accesos debido a la complejidad y al gran número de patrones que aparecen en un corto espacio de tiempo, típicas de las aplicaciones comerciales [5, 7]. Esto conlleva a una gran cantidad de predicciones incorrectas debido al solapamiento entre secuencias. En vez de mapear las predicciones por una sola delta, una secuencia de deltas es utilizada. Esto ayuda a desambiguar las secuencias de delta direcciones y obtenerlas en el mismo orden de aparición. Debido a la gran existencia de prebuscadores que se han diseñado a lo largo del tiempo, se ha convenido utilizar una abstracción para su clasificación. La notación elegida es la misma que ha sido utilizada en [4]. La Figura 8 muestra la clasificación:

Key / Mechanism	A/AC	D/DC	DS/DC
Prefetcher	TMS Markov	Delta Markov	DS-TMS

Figura 8. Clasificación de los prebuscadores temporales

Los prebuscadores temporales son clasificados de acuerdo a como indexan las predicciones y de qué tipo es la predicción que realizan (direcciones, diferencias, …).

La taxonomía utilizada para la clasificación de los prebuscadores tiene en cuenta el tipo de clave que se utiliza para identificar los distintos patrones almacenados y la forma en la que los patrones son almacenados (posición fija o relativa). Los del tipo A/AC guardan la secuencia de referencias como direcciones fijas (AC – Address Correlated) e indexan los patrones almacenados utilizando las direcciones que pide el procesador a memoria (A – Address). Otros, como los del tipo D/DC, guardan la secuencia de accesos como delta direcciones (DC – Delta Correlated) y diferencian los patrones mediante la diferencia de las direcciones entre dos operaciones de memoria consecutivas (D – Delta). Por otra parte, los prebuscadores DS/DC, siguen la misma idea que los del tipo D/DC pero en vez de utilizar para indexar las predicciones una única delta, utilizan una secuencia de delta direcciones. En la sección 6 se muestran los resultados de cada tipo de prebuscador.

5. PREDICCIÓN POR ACCESO

Estudios recientes [2, 7] se basan en la idea de extraer patrones dentro de regiones del espacio de direcciones. Debido a que los programas tienden a acceder a estructuras de datos comunes y de la misma manera, los patrones guardados para una región pueden aparecer en muchas otras. Técnicas híbridas aprovechan la correlación temporal a través de los accesos que inician predicciones espaciales solventando de cierta manera el coste de oportunidad de este primer acceso necesario en cada región, haciendo posible encadenar varias predicciones en distintas regiones con un único acceso. Los predictores basados en regiones actuales tienen una limitación principal basada en como predicen dentro de las regiones. Debido que cada predicción se inicia con un primer acceso dentro de la misma y ésta es para toda la región, una predicción errónea desemboca en una total pérdida de oportunidad de las predicciones dentro de la región. Reduciendo la granularidad de las predicciones a una por acceso dentro de la región en vez de una en el primer acceso a la región, conseguimos predecir patrones espaciales aunque estos no sean totalmente iguales. No estamos atados al primer acceso dentro de la región nunca más ya que todos los accesos inician predicciones. La figura 9 muestra un ejemplo explicando la limitación de los predictores espaciales actuales y como sería solventado por nuestra propuesta.

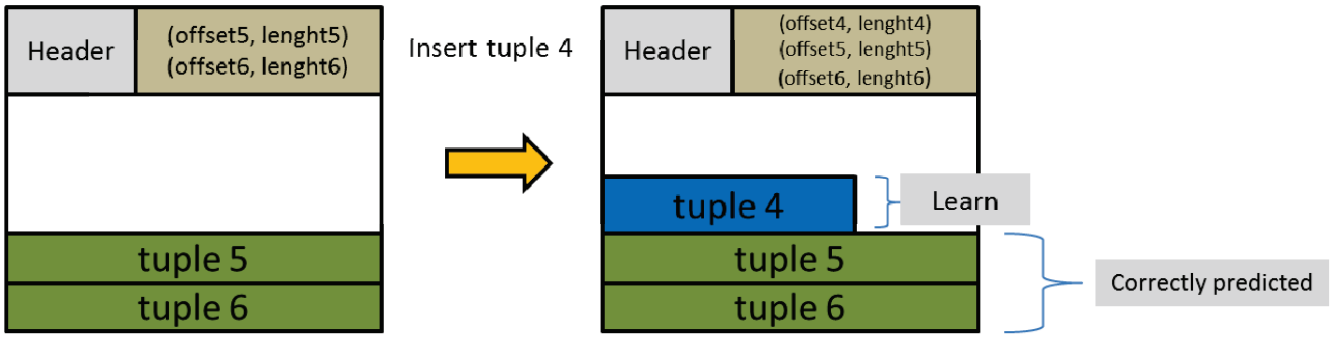

FIGURA 9. Ejemplo de limitación de los predictores espaciales actuales.

Parte de los patrones espaciales concuerdan después de la modificación de una página de un DBMS. Sin embargo, si el primer acceso a la página se modifica o si hay patrones entrelazados pero con un primer acceso común, los predictores espaciales actuales pierden toda la oportunidad para predecir correctamente en la región. Eliminando toda conexión entre predicción y primer acceso a la región, podemos guardar nuevas secuencias a la vez que predecimos para las ya aparecidas, todo a la vez.

Imagínese que tenemos una tabla que consiste en varias tuplas distribuidas a lo largo de unas cuantas páginas de un DMBS. En este caso, las tuplas están ordenadas por una clave primaria. El usuario añade una tupla nueva a la tabla pero debido a que la tabla está ordenada de acuerdo a una clave primaria, el gestor no puede colocar la tupla en cualquier sitio. El DMBS analiza las cabeceras y los offsets para acceder a las claves y colocar la tupla 4 entre la 3 y la 5. Como podemos observar en la figura, el recorrido de la página cuando la tupla 4 es insertada hace que la disposición espacial sea diferente aunque tiene partes comunes con el patrón antiguo. Si el acceso que inicia la predicción espacial cambia, nuestra técnica es capaz de predecir patrones comunes que aparecen en la página modificada antes de aprender la nueva secuencia. Si aunque el primer acceso a la página no cambia, pero al recorrer páginas aparecen patrones distintos, nuestro predictor es capaz de predecir correctamente todos ya que los distinguirá a medida que haya más accesos en la página. Por ejemplo, si estamos barriendo una tabla de una base de datos como en el ejemplo anterior, en algunas páginas existirán dos y en otras tres tuplas. Como acceden utilizando el mismo primer acceso relativo a la página, nuestra oportunidad está limitada a cuantas páginas de cada tipo son escaneadas de forma consecutiva y cuanta similitud existe entre ambos patrones. Este caso se presenta gráficamente en la figura 10.

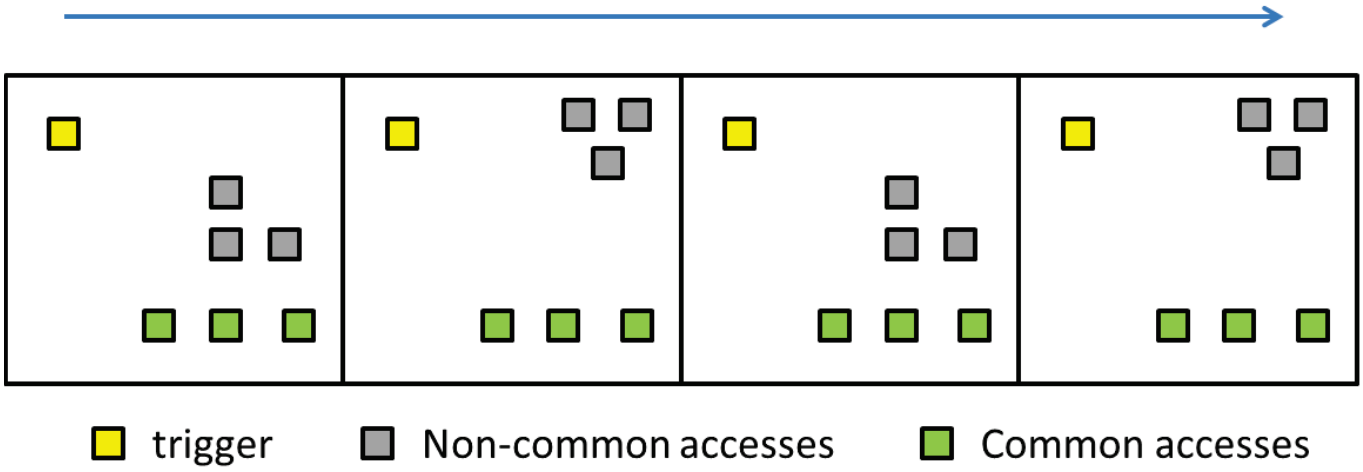

FIGURA 10. Patrones entrelazados con un primer acceso relativo a la región común.

SMS y STeMS son incapaces de predecir casos en los que el primer acceso a la región es común pero los patrones espaciales son diferentes.

Los prebuscadores basados en regiones como SMS o STeMS (Híbrido) utilizan el PC de la instrucción que ha causado el primer acceso a la región para indexar las predicciones espaciales almacenadas e iniciar una si alguna coincide. Utilizar el PC de la instrucción no es la solución si queremos realizar predicciones por acceso dentro de las regiones, debido a que es altamente probable que accedan a la región más de una instrucción en el periodo en el que se almacena el patrón espacial. Si diseñamos un prebuscador que prediga por acceso en vez de por región utilizando el PC como la clave para diferenciar patrones, nuestra oportunidad de predicción estará limitada al número de bloques referenciados por la misma instrucciones por región, necesitando un primer acceso por cada PC por región. Como el número de bloques referenciados dentro de una región es generalmente bajo [7] no hay ningún motivo por el que diseñar un prebuscador de este tipo. La figura 8 muestra el comportamiento de los accesos a una región según el tipo de instrucciones.

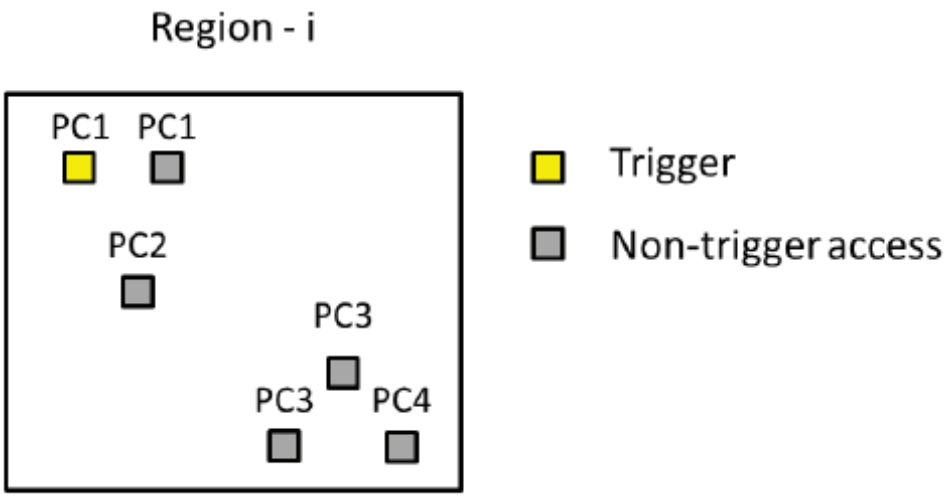

FIGURA 11. Comportamiento de accesos en un región.

El número de bloques referenciados dentro de una región es generalmente bajo y es muy probable que estos accesos sean debidos a más de una instrucción.

Analizando trazas de referencias a memoria de aplicaciones comerciales hemos comprobado que alrededor del 40% del tiempo, la instrucciones que produce el primer acceso a la región, accede inmediatamente después a otra distinta, siendo otras instrucciones las que producen el resto del patrón dentro de la región. Una solución directa a este problema sería el indexar las predicciones mediante las direcciones referenciadas dentro de la región, debido a contienen la disposición exacta de los bloques referenciados dentro de la región. Aunque es cierto que de esta forma podemos predecir

por acceso dentro de las regiones, estamos limitándonos a predecir patrones en las mismas regiones dónde han aparecido, ya que las direcciones corresponden con posiciones fijas del espacio de direcciones. Esto elimina la posibilidad de predecir secuencias comunes de acceso espacial que aparecen a través de múltiples y diferentes regiones, lo que es la principal ventaja de los prebuscadores espaciales.

Aquí es donde las delta direcciones entran en juego, ya que se basan en diferencias relativas entre direcciones en vez de en posiciones fijas del espacio de direcciones. Lo que equivale a que si existen patrones comunes a regiones distintas, estos poseerán la misma secuencia en delta direcciones. La figura 12 explica el siguiente comportamiento de forma gráfica.

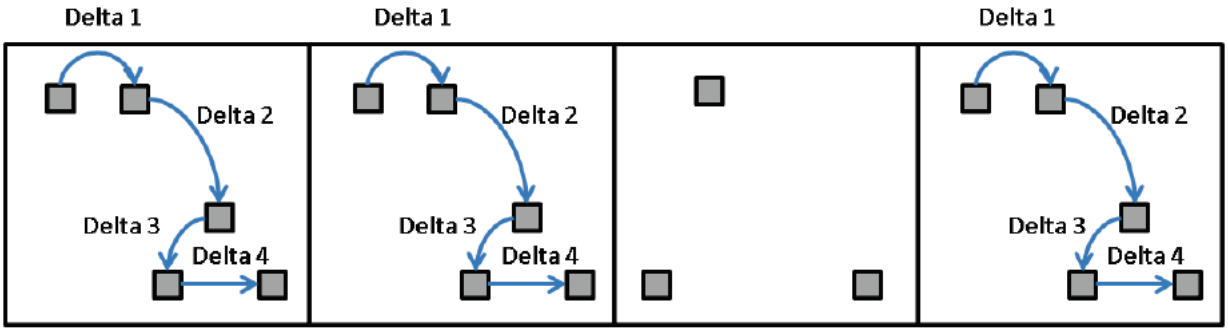

FIGURA 12. Delta direcciones en secuencias de acceso espacial.

Los rectángulos grises representan los bloques referenciados dentro de la región por el patrón de acceso. A medida que los patrones recurren en otras regiones las deltas se repiten.

Es obvio que una única delta no es capaz de distinguir entre todos los patrones espaciales que aparecen en distintas regiones. En vez de utilizar una única delta, una secuencia de deltas es utilizada para indexar los patrones almacenados. Utilizando la misma abstracción usada en la sección anterior para la clasificación de los prebuscadores temporales, la figura 9 presenta las características de los prebuscadores basados en regiones más avanzados y nuestra propuesta.

Key / Mechanism	PC/DC	(PC + A)/(DC + AC)	DS/DC
Prefetcher	SMS	STeMS	DS-TMS

FIGURA 13. Clasificación de los prebuscadores espaciales.

Los prebuscadores son clasificados de acuerdo a como indexan las predicciones almacenadas y de qué forma almacenan dichos patrones.

SMS utiliza para la representación de patrones un vector de bits que es en realidad una forma de representar posiciones relativas (deltas) dentro de una región. En SMS las predicciones son iniciadas por el PC de la instrucción que produce el primer acceso a la región, mientras que STeMS utiliza el mismo razonamiento para las predicciones espaciales PC/DC y A/AC para correlacionar esos accesos a regiones de forma temporal. Otros como D/DC almacenan los patrones como diferencias relativas (DC – Delta Correlated) pero para indexar las predicciones almacenadas utilizan una única delta lo que no es suficiente para desambiguar patrones en aplicaciones comerciales. Nuestra propuesta sigue esta misma última idea pero indexa las predicciones con una secuencia de delta direcciones DS/DC. Este esquema es similar al propuesto en la anterior sección, sin embargo en este caso debido a que las predicciones se obtienen en cada región, las diferencias relativas están acotadas por el tamaño de la región.

6. PROPUESTA

Esta sección describe ambas propuestas para la captura de secuencias de acceso temporal y espacial utilizando delta direcciones. Este trabajo está enfocado para aplicaciones comerciales de servidores en un entorno de procesador multi-núcleo.

En resumen, el diseño de ambas propuestas comprende dos estructuras hardware. La *Learning Table* (LT) recoge patrones directamente de las peticiones del procesador a la jerarquía de memoria (accesos a la cache L1) en forma de secuencia de delta direcciones. La *Prediction Table* (PT) almacena la delta que sigue a la secuencia de accesos con la que indexamos la tabla. Las próximas dos secciones describen ambos prebuscadores.

6.1 Delta Sequence - Temporal Memory Streaming

La propuesta llamada *Delta Sequence Temporal Memory Streaming* (DS-TMS) se explica en esta sección. Para mayor claridad la explicación estará dividida en dos secciones, refiriéndose a las dos fases que componen el funcionamiento de cualquier prebuscador: aprendizaje y predicción.

6.1.1 El aprendizaje en DS-TMS

Esta sección explica la forma en la que DS-TMS extrae y guarda los patrones de referencias a memoria. Este prebuscador posee dos estructuras hardware, la LT y la PT. La LT extrae la secuencia de deltas de los accesos del procesador y los utiliza para indexar la tabla PT y almacenar allí la delta que sigue a la secuencia de deltas vista anteriormente y que hemos utilizado para el indexado. Para extraer las deltas, la LT tiene que almacenar en cada acceso la dirección pedida por el procesador y calcular la diferencia con la petición actual. La última secuencia de deltas observada será otra entrada en la tabla (a más bits destinados, más precisión, menos solapamiento entre patrones). Cada vez que se calcula una nueva delta, la LT indexa la PT utilizando el historial de deltas (el historial aún no contiene la delta más recientemente calculada). La última delta calculada y la obtenida al indexar la PT son comparadas, actualizando el contador de la LT, que es una entrada utilizada para controlar que las predicciones son correctas. La delta obtenida de la PT es la delta que apareció la última vez, después de la misma secuencia que acabamos de utilizar para el indexado. Si los ambas deltas coinciden aumentamos el contador en una unidad, sino el contador es decrementado en otra unidad. Después de actualizar el contador, la PT es indexada de nuevo con el historial de deltas para almacenar la delta recientemente

calculada. Esta delta es utilizada para predecir que delta seguirá a la secuencia de deltas que hemos utilizado para indexar la PT en la fase de predicción. Entonces, el historial de deltas se desplaza para eliminar la delta más antigua y dejar paso a la más nueva. El historial de deltas almacena las delta direcciones de forma hash, utilizando una función de dispersión como la usada en [9], llamada FS-5. Para terminar, el campo de la LT de la última dirección es actualizado con el valor de la última dirección referenciada para el cálculo de la futura siguiente delta. La figura 14 muestra la fase de aprendizaje del prebuscador DS-TMS.

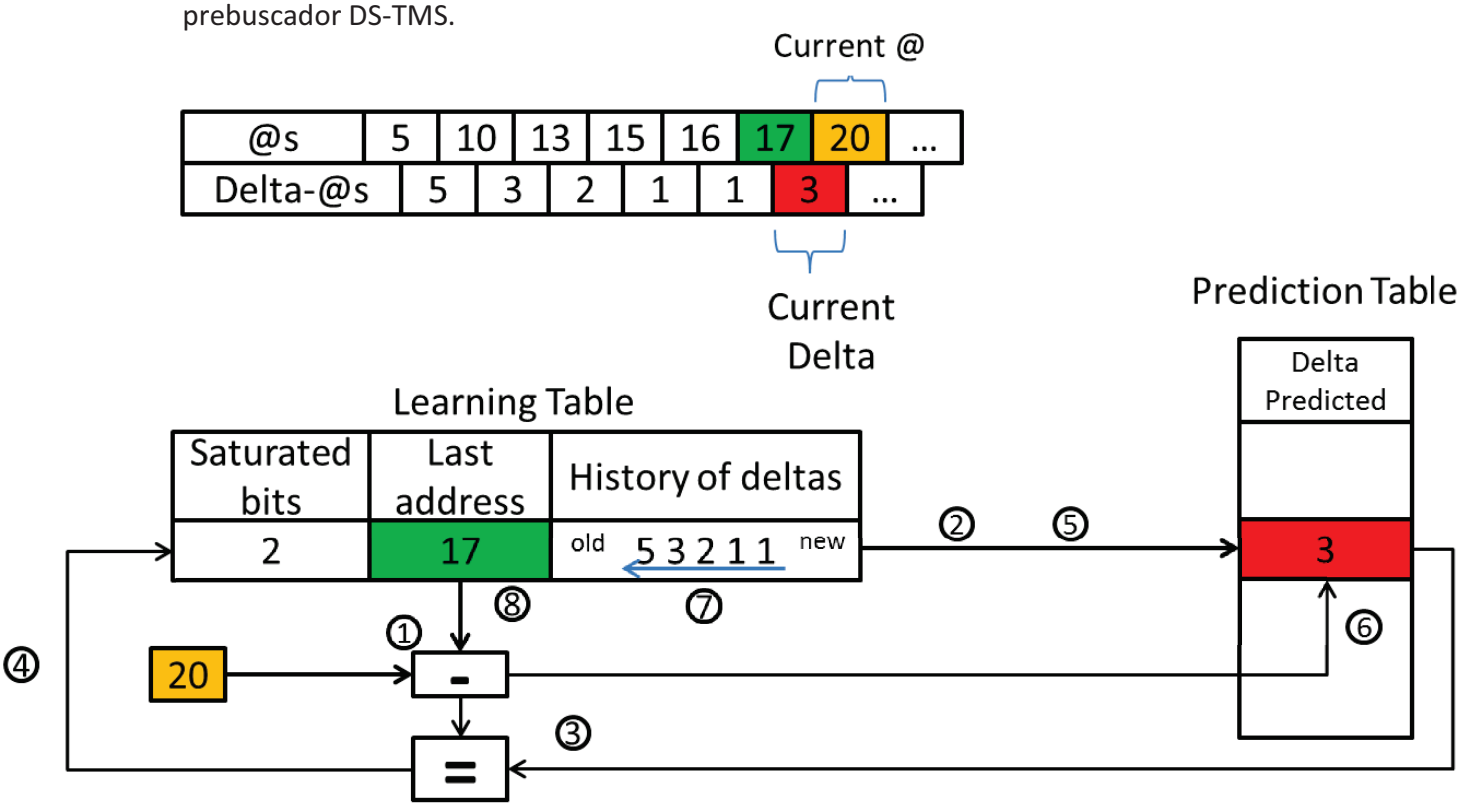

FIGURA 14. Fase de aprendizaje en el prebuscador DS-TMS

En el periodo de aprendizaje, siempre que la LT recibe una referencia de memoria del procesador, una nueva delta es calculada y la PT es indexada utilizando el historial de las n-últimas deltas recientemente observadas en el patrón de acceso. Después de comparar las deltas y actualizar el contador, se indexa la PT y la última delta calculada es insertada en la tabla. Por último el historial es actualizado.

Siguiendo el orden que aparece en la figura:

1. Una nueva delta es calculada.
2. La PT es indexada utilizando el historial, que posee la secuencia de deltas más recientes.
3. La delta obtenida de la PT es comparada con la recientemente calculada.
4. El contador es actualizado. Si las deltas coinciden, el valor del contador de la LT es incrementado. Si por el contrario no coinciden, se decrementa en una unidad.
5. La PT es indexada de nuevo con el mismo historial de deltas.
6. La nueva delta es almacenada en la PT.
7. El historial de deltas es desplazado para eliminar la delta más antigua. La delta más recientemente calculada es insertada en el historial.
8. La LT actualiza el campo de la última dirección referenciada para el siguiente cálculo de la delta.

6.1.2 La predicción en DS-TMS

Las predicciones en DS-TMS son iniciadas por una secuencia de delta direcciones. En el mecanismo de aprendizaje, siempre que se produce un acceso a la cache L1, la tabla LT es accedida y una nueva delta es calculada. Entonces, después de actualizar el contador, la PT es indexada y la nueva delta se guarda en la tabla, tal y como ha sido explicado en la anterior sección. En la fase de predicción, si el valor del contador es mayor que un umbral (en nuestro caso, como hemos utilizado para el contador uno de 2 bits saturado, el umbral es el valor 1) a nuestra elección, se inicia la fase de predicción. El nuevo historial calculado es almacenado en un registro y dependiendo de nuestro grado de prebúsqueda (número de predicciones lanzadas por evento de prebúsqueda) utilizaremos ese registro para acceder a la PT, obtener la delta y calcular, con la última dirección referenciada, la predicción a realizar. Utilizaremos el registro para indexar la PT y obtener las sucesivas deltas, ya que desplazaremos el historial del registro a medida que vayamos obteniendo más deltas de la PT, tantas veces como el indicado por el grado de prebúsqueda. La figura 15 explica el funcionamiento de las predicciones.

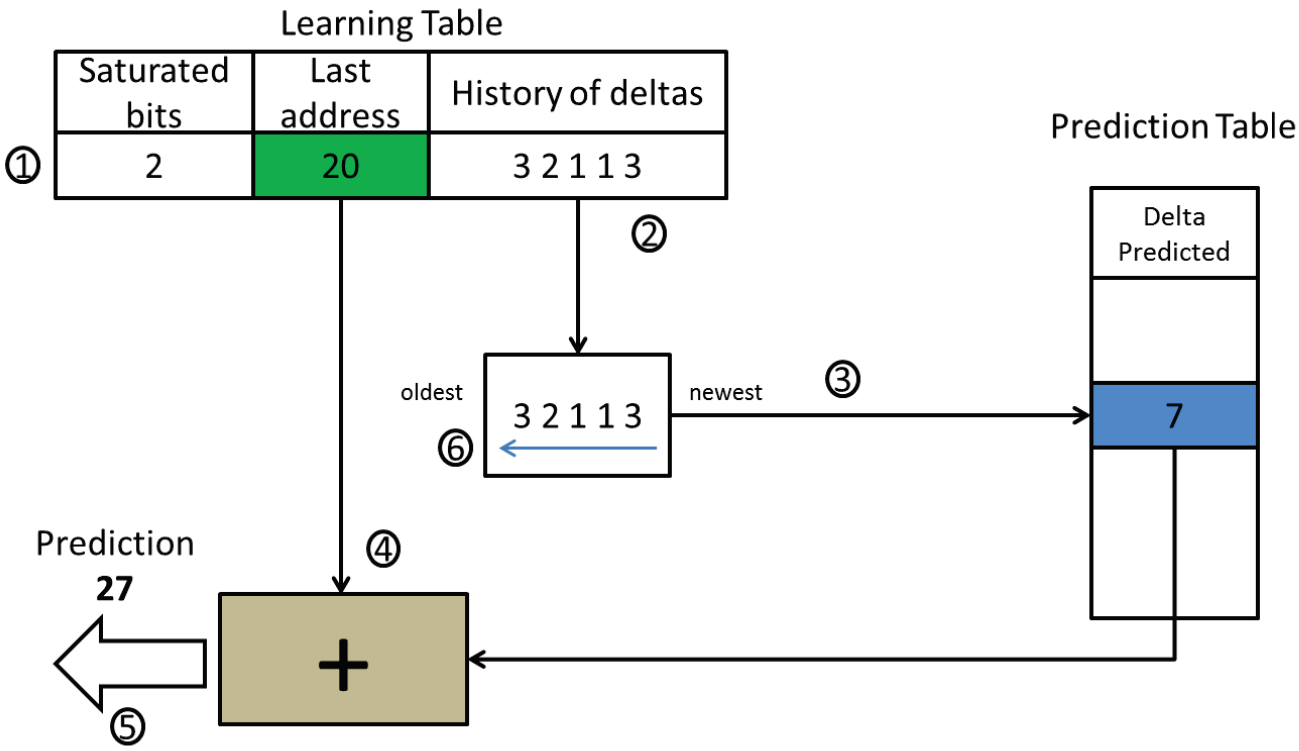

FIGURA 15. Fase de predicción en el prebuscador DS-TMS

Las predicciones comienzan si el valor del contador está por encima de un umbral, durante el periodo de predicción. Estas predicciones se realizan utilizando el historial almacenado en un registro y la última dirección referenciada por el procesador. Si el grado de prebúsqueda es mayor que uno, el registro se desplaza para tener en cuenta la delta recién utilizada en la predicción y volver a indexar la PT para obtener la delta de la siguiente predicción. Este paso se realiza tantas veces como diga el grado de la prebúsqueda.

Los pasos que aparecen numerados en la figura presentan la fase de predicción como un proceso de seis pasos:

1. Si el contador está por encima de un umbral, el mecanismo de predicción comienza.
2. El historial es almacenado en el registro.
3. El registro es utilizado para indexar la PT.

4. La delta almacenada en la PT y la última dirección referenciada por el procesador se suman para calcular la dirección de la predicción.
5. Realizamos la primera predicción.
6. Si el grado de prebúsqueda es mayor que uno, el registro que almacena el historial es desplazado para almacenar la delta recién utilizada en la predicción y el mecanismo se repite.

6.2 Delta Sequence - Spatial Memory Streaming

El prebuscador *Delta Sequence Spatial Memory Streaming* se explica en esta sección. Para enseñar de una forma más clara y precisa el funcionamiento de DS-SMS vamos a dividir su explicación en dos mecanismos diferenciados, aprendizaje y predicción.

6.2.1 El aprendizaje en DS-SMS

Este prebuscador posee dos estructuras hardware, la tabla LT y la tabla PT. La LT es la encargada de extraer las secuencias de delta direcciones provenientes de las referencias del procesador, para realizar la indexación de la tabla PT que guarda la predicción de la secuencia de deltas observada. A más bits utilizamos para el historial, más deltas y menos solapamiento entre patrones. La diferencia con DS-TMS es que en vez de calcular las deltas de todo el espacio de direcciones, dividimos el espacio de direcciones en regiones y calculamos las deltas dentro de cada región por separado. Para ello (en el caso ideal para evitar colisiones) tendremos tantas entradas en la tabla de direcciones como regiones de memoria en el que hayamos dividido el espacio de direcciones. De esta manera la historia de las últimas n-deltas será independiente para cada región. Cada vez que el procesador envía una petición a la cache L1, la dirección es utilizada para indexar la tabla LT y acceder a la entrada de la región a la que estamos referenciando. Una vez que hemos encontrado la entrada buscada, el campo dirección es utilizado para calcular la nueva delta, calculando la diferencia con la última dirección accedida. Existe un campo donde se almacena el historial de las últimas deltas observadas en la secuencia de accesos a la región y es utiliza para indexar la tabla PT y obtener la delta que apareció la última vez después de esa misma secuencia de delta direcciones. Comparamos ambas deltas para actualizar el contador de la entrada de la LT. Este contador se utiliza para controlar la probabilidad de que las predicciones que hagamos sean correctas. Si ambas deltas coinciden aumentamos el valor del contador en una unidad, sino el valor se decrementa en la misma cantidad. Después,

utilizando la misma secuencia de deltas anterior, volvemos a indexar la PT y guardamos el valor de la nueva predicción, la nueva delta. La próxima vez que aparezca la misma secuencia de delta con la que hemos indexado la PT, la predicción será la delta que acabamos de almacenar. Entonces, el historial de deltas es desplazado para eliminar la delta más antigua del historial e insertar la más nueva, actualizamos el historial. El historial de deltas se guarda en forma hash, utilizando una función de dispersión usada en [9], llamada FS-5. Al final, el campo de la entrada LT de la última dirección referenciada es actualizada con la dirección del acceso actual para calcular la delta del próximo acceso del procesador a la jerarquía de memoria. Algo remarcable es que, ya que para indexar la PT solamente utilizamos las últimas n-deltas observadas en la región y ninguna información sobre una región fija en sí, la tabla PT es compartida por todas la regiones. Esto se traduce en que un patrón aprendido en una región puede ser utilizado para predecir en otras regiones nunca antes visitadas (positive aliasing). Por motivos de simplicidad, vamos a dividir el espacio de direcciones en cuatro regiones (0, 1, 2 y 3) para explicar el funcionamiento de D-SMS, aunque este mecanismo puede extrapolarse a un número arbitrario de regiones. La figura 16 muestra el mecanismo de aprendizaje del prebuscador DS-SMS.

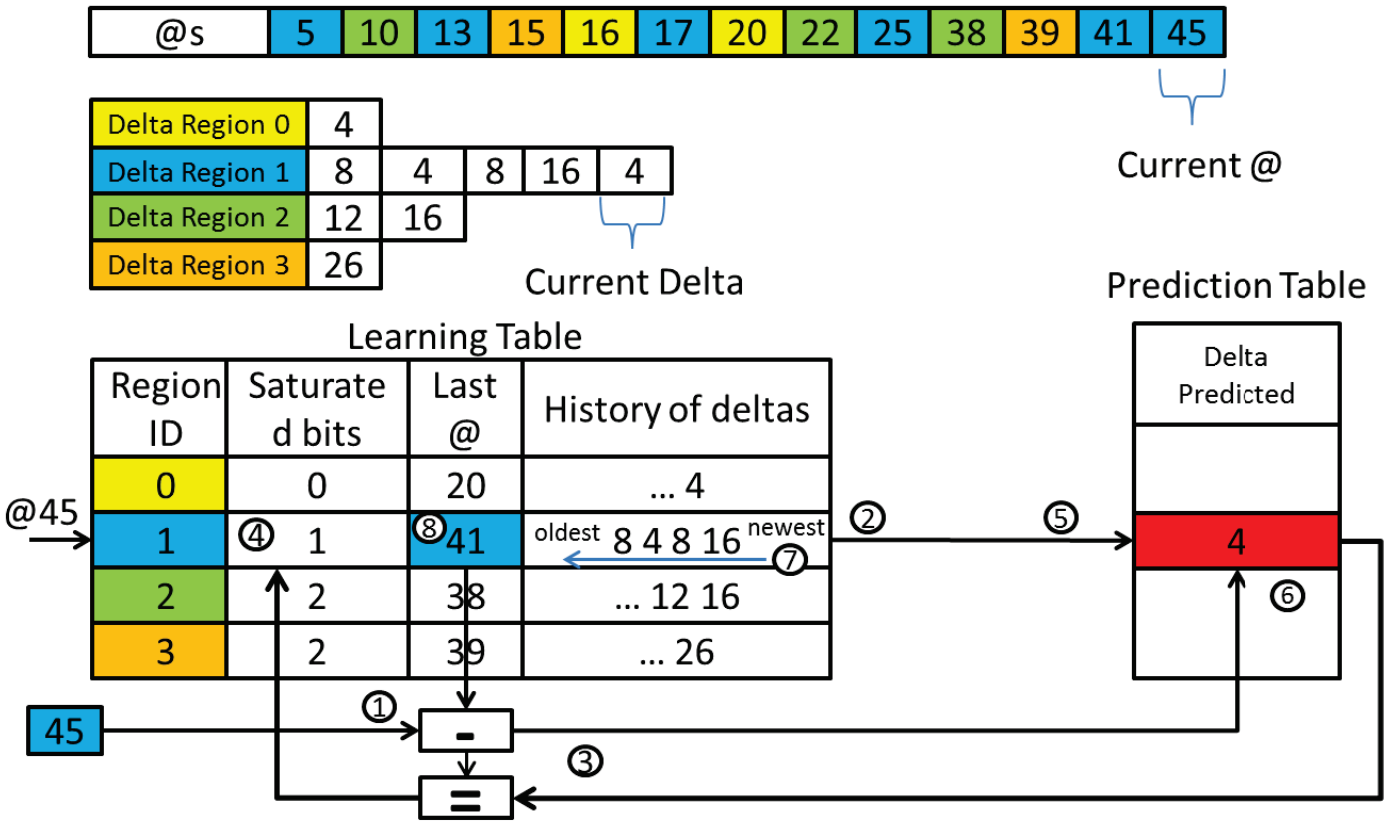

FIGURA 16. Periodo de aprendizaje del prebuscador DS-SMS

En el periodo de aprendizaje, cada vez que la LT recibe una petición del procesador, se calcula una nueva delta para la entrada correspondiente a la región accedida. Utilizando el historial de delta direcciones de la región indexamos la PT y actualizamos el contador con la delta de predicción. Acto seguido, volvemos a indexar la PT y almacenamos la nueva delta calculada.

Siguiendo el orden de la figura, el mecanismo consta de ocho pasos diferenciados:

1. Indexando la LT, una nueva delta es calculada.
2. La última secuencia de deltas observadas es utilizada para indexar la tabla PT.
3. La delta especulativa de la PT y la nueva delta calculada son comparadas.
4. Dependiendo de la comparación, el contador es actualizado. Si ambas son iguales, el contador es incrementado, sino es decrementado en una unidad.
5. La PT es indexada de nuevo.

6. La delta recién calculada es almacenada en la tabla PT.
7. El historial de la última secuencia de deltas observadas se desplaza para dejar sitio a la nueva delta y eliminar a la más antigua.
8. La última dirección que ha sido referenciada de la región es almacenada en el campo de última dirección.

6.2.2 La predicción en DS-SMS

En el prebuscador DS-SMS, las predicciones son iniciadas por una secuencia de deltas. En la fase de aprendizaje, cada vez que hay un acceso a la cache L1, la tabla LT es accedida y una nueva delta es calculada. La entrada de la LT corresponde a la región que estamos accediendo en ese momento. Después, el historial de deltas de la entrada se utiliza para indexar la tabla PT y obtener la delta de la predicción. Comparamos ambas deltas, actualizamos el contador y almacenamos la nueva delta en la tabla PT, tal y como hemos explicado en la sección anterior. En la fase de predicción, si el contador después de la actualización está por encima de un umbral a nuestra elección (el valor 1 es en nuestro caso ya que hemos utilizado un contador de 2 bits saturado), el mecanismo de predicción se inicia. El historial actualizado se almacena en un registro, que se utilizará para actualizar el historial con las deltas obtenidas de la PT de forma especulativa. Mediante la delta obtenida de la PT y la última dirección referenciada realizaremos la primera petición a la jerarquía de memoria. Si el grado de prebúsqueda en mayor que la unidad, el historial del registro se desplaza para dar paso a la delta de la predicción y volver a obtener otra delta especulativa. La siguiente predicción utilizara la última dirección referenciada (la anterior predicción) y la delta recién obtenida, para volver a realizar la segunda predicción. La figura 17 muestra esta fase.

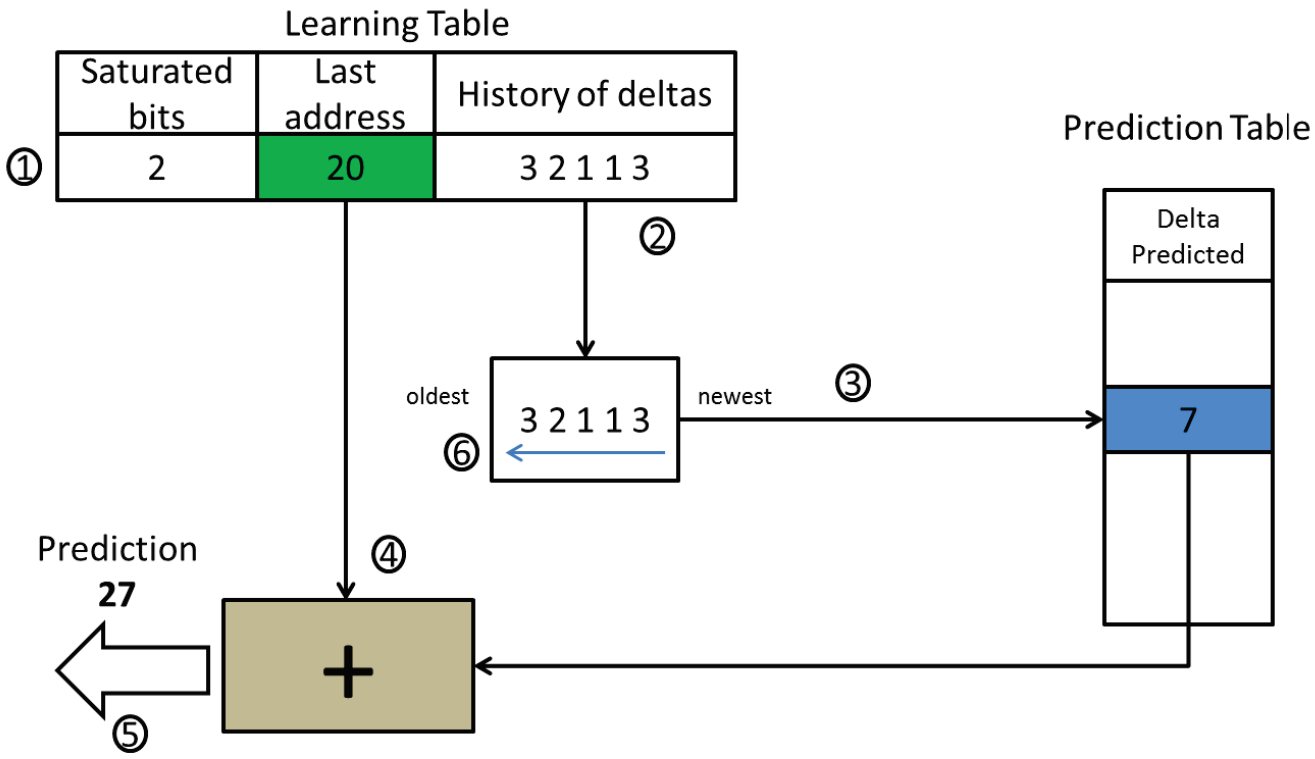

FIGURA 17. Fase de predicción del prebuscador DS-SMS

En el periodo de aprendizaje, si después de calcular el nuevo valor del contador este es mayor que un umbral, el periodo de predicción se inicia. El nuevo historial de deltas se almacena en un registro que se utiliza para indexar la PT. La delta especulativa se suma a la última dirección referenciada para obtener la dirección de la predicción. Después, si el grado de prebúsqueda es mayor que la unidad, el historial del registro se desplaza para dejar espacio a la nueva delta especulativa y el mecanismo se repite.

Los pasos del diagrama son los siguientes (Hemos obviado por simplicidad el hecho de que la LT tiene tantas entradas como regiones, simplemente hemos dibujado la entrada de la región que se accede):

1. Después de la fase de aprendizaje, si el valor del contador está por encima de un umbral, la fase de predicción se inicia.
2. El historial es guardado en un registro.
3. Indexamos la PT usando el valor del registro.
4. Añadimos la delta que obtenemos en el paso 3 y la añadimos a la última dirección referenciada.

5. Realizamos la petición a la jerarquía de memoria.
6. Si el grado de prebúsqueda es mayor que uno, el mecanismo se repite actualizando el historial con la delta especulativa y volviendo a acceder a la tabla PT. En este caso la última dirección referenciada corresponde con la primera predicción realizada.

7. EVALUACION

Esta sección está dividida en dos subsecciones. La primera explica la metodología utilizada y la segunda presenta los resultados, comparando ambas propuestas con los prebuscadores que más éxito han tenido hasta el momento.

7.1 Metodología

La evaluación de los prebuscadores se ha realizado mediante simulación funcional de un procesador multi-núcleo utilizando el simulador FLEXUS [10]. FLEXUS permite ejecutar aplicaciones comerciales sin ninguna modificación. FLEXUS trabaja encima del simulador *Virtutech Simics*, incorporando modelos de procesadores modernos, jerarquías de memoria, protocolos de comunicación y redes de comunicación. El entorno de simulación es un procesador de 16 núcleos con 3GB de memoria física ejecutando el sistema operativo *Sun Solaris 8*. Otros parámetros relevantes se indican a continuación, en la tabla 1.

System Parameters	
Core Node	UltraSPARC III ISA In-order IPC = 1.0 L1 Private - L2 Shared (LLC)
L1 Caches	Split Instructions/Data 64KB 2-way LRU Replacement Policy
L2 Cache	Unified 8MB 16-way LRU Replacement Policy
Main Memory	3 GB total memory 64B coherence unit

TABLA 1. Parámetros del sistema multi-núcleo.

Se simulan dos tipos de cargas de trabajo de bases de datos en dos gestores de bases de datos (*IMB DB2* y *Oracle 10g Enterprise Database Server*), un benchmark de servidores web (*SPECWeb 2005*) ejecutándose en *Apache HTTP server* y dos aplicaciones científicas. La tabla 2 proporciona más información de las cargas de trabajo simuladas.

Online Transaction Processing (TPC-C)	
Oracle	100 warehouses (10 GB), 16 clients, 1.4 GB SGA
DB2	100 warehouses (10 GB), 64 clients, 450 MB buffer pool
Online Transaction Processing (TPC-H on DB2)	
Qry 2	Join-dominated, 450MB buffer pool
Qry 16	Join-dominated, 450MB buffer pool
Qry 17	Balanced scan-join, 450MB buffer pool
Web Server	
Apache	Join-dominated, 450MB buffer pool
Zeus	16K connections, FastCGI
Scientific	
Ocean	1026x1026 grid, 9600s relaxations, 20K res., err tol 1e-07
Em3D	4096x4096 matrix

TABLA 2. Información adicional de las cargas de trabajo.

Nuestro análisis funcional utiliza trazas de accesos a memoria obtenidos de FLEXUS con ejecución en orden, sin bloqueos de memoria y con un IPC fijo de 1.0. Para las aplicaciones comerciales y científicas, calentamos la memoria principal con la simulación funcional durante 4 billones de instrucciones antes de tomar cualquier medición. Luego se ejecutan otros 4 billones de instrucciones de las que obtenemos los resultados. Todos los resultados de este PFC utilizan esta metodología.

7.2 Resultados

Para entender mejor los resultados vamos a realizar las siguientes aclaraciones: 1) La función de dispersión FS-5 es utilizada para indexar la tabla PT. Utiliza tantas deltas como el número de bits elegido para almacenar el historial en la tabla LT, dividido por 5. Como en este estudio queremos analizar la oportunidad de ambos prebuscadores, hemos elegido un número elevado de bits para almacenar un número elevado de deltas. Al elegir 64 bits podemos técnicamente tener hasta 2^64 secuencias diferentes, lo que siguiendo la fórmula se traduce en secuencias de delta direcciones de una longitud de 13. 2) El prebuscador A/AC es una aproximación a TMS. Mantiene la idea principal de un gran buffer circular de varios MBs para extraer y almacenar secuencias de acceso temporal, sin embargo en vez de utilizar varias colas de predicción utilizamos solo una. 3) Los resultados obtenidos para STeMS se han obtenido añadiendo a SMS un gran buffer circular para predecir los accesos a regiones espaciales que producen predicciones y que se encuentran correlacionados temporalmente. Lo que es una aproximación como en el anterior caso, aunque ajustándonos a la idea principal del diseño original. 4) La cobertura (coverage), en la terminología de prebúsqueda, es el porcentaje de fallos que son eliminados debido al prebuscador. Las predicciones incorrectas o no utilizadas (overpredictions) son la fracción de las peticiones de prebúsqueda iniciadas por el prebuscador pero que son expulsados de la cache antes de que sean referenciadas por el procesador. 5) En las gráficas, la cobertura y la fracción que sigue siendo un fallo en la cache (uncovered) representan el total de fallos de lectura en la cache L1 de las referencias del procesador.

7.2.1 Resultados del prebuscador DS-TMS

Los siguientes resultados presentan la oportunidad de DS-TMS para predecir patrones temporales utilizando una tabla PT de tamaño infinito. Debido a que en DS-TMS no existen regiones, la tabla LT solamente posee una única entrada.

La principal ineficiencia de los prebuscadores temporales es la imposibilidad de predecir referencias a direcciones nunca antes visitadas. Esto es especialmente ineficiente en aplicaciones donde un gran porcentaje del conjunto de datos es leído solamente una única vez. Esto es muy común en las cargas de trabajo DSS. Como las cargas de trabajo de bases de datos utilizan estructuras de datos con una disposición fija, esta ineficiencia estará solventada si un prebuscador temporal es capaz de conseguir una cobertura similar a un prebuscador espacial en este tipo de cargas de trabajo. La figura 18 muestra la cobertura de un prebuscador temporal, uno espacial y nuestra solución temporal.

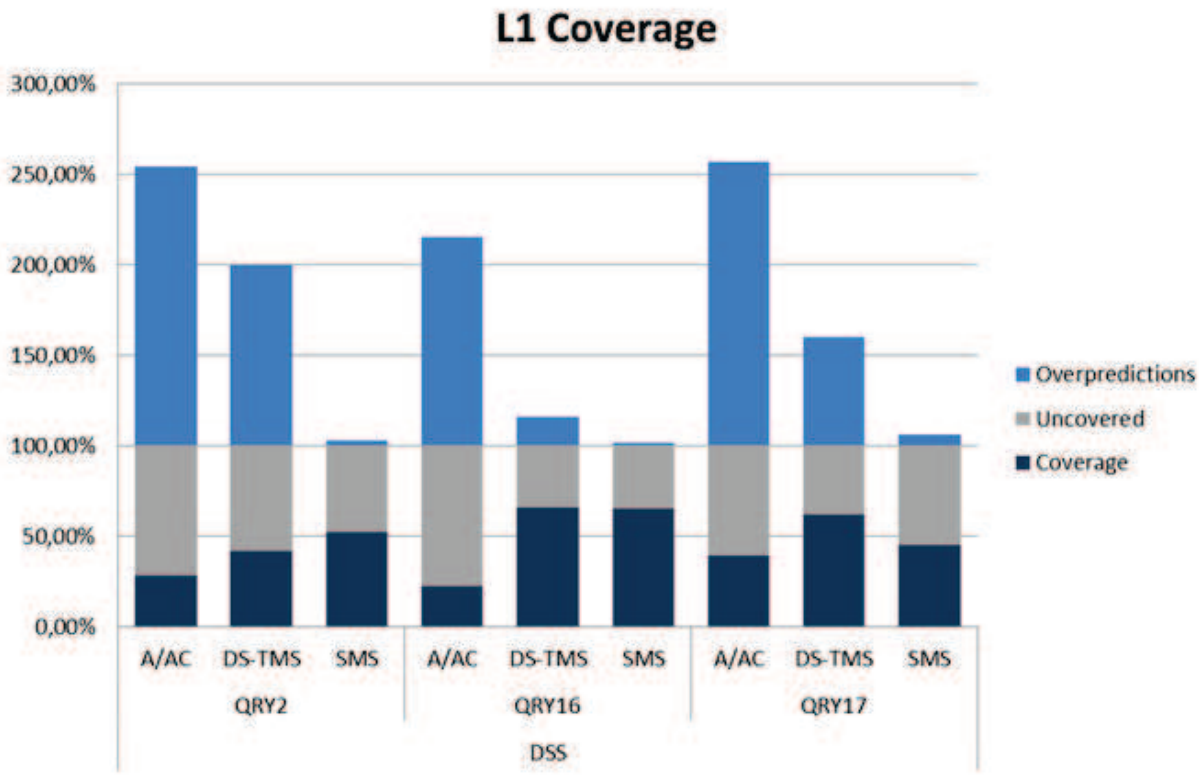

FIGURA 18. Cargas de trabajo DDS.

Esta gráfica muestra por carga de trabajo la oportunidad de DS-TMS de predecir patrones de acceso temporal en comparación con un prebuscador espacial.

Como podemos ver en la gráfica, DS-TMS es capaz de predecir referencias a direcciones nunca antes visitadas en la cargas de trabajo QRY2 y QRY16 donde A/AC no es capaz. En QRY17 superamos incluso a SMS debido a la existencia de patrones de acceso temporal que atraviesan varias regiones del espacio de direcciones y son repetitivos. Esto es corroborado por la diferencia entre A/AC y SMS en QR17 en comparación con las otras dos cargas de trabajo. Este repunte respecto a SMS es debido a que si las referencias de acceso temporal atraviesan varias regiones y son repetitivas, en SMS necesitaremos un acceso por región, mientras que en DS-SMS solamente necesitaremos un primer acceso e iremos prediciendo sin limitarnos a los límites de la región. Es cierto que una ventaja en los prebuscadores espaciales es que el número de predicciones incorrectas es más reducido que la de los temporales [7]. Esto es debido a que los patrones dentro de las regiones son más estables que los obtenidos en todo el espacio de direcciones. Aunque como hemos dicho antes A/AC es una aproximación de TMS, estos experimentos corroboran resultados previos ya que A/AC tiene una cobertura menor que SMS en cargas de trabajo DSS.

Por otro lado, las cargas de trabajo OLTP y WEB recurren continuamente a las mismas estructuras de datos produciendo secuencias de acceso temporal. La figura 19 presenta los resultados de este tipo de cargas de trabajo utilizando prebuscadores temporales.

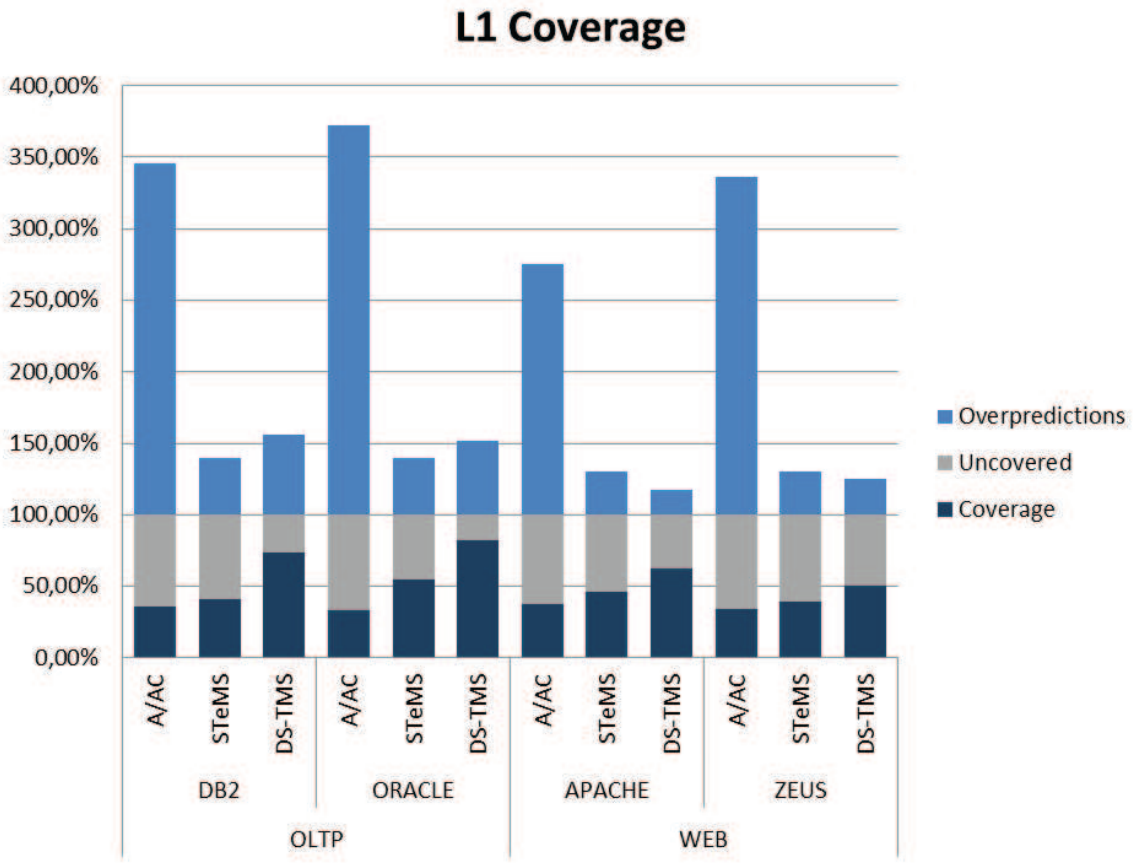

FIGURA 19. Cargas de trabajo OLTP y WEB.

Esta gráfica muestra por carga de trabajo la oportunidad de DS-TMS para predecir patrones de acceso temporal comparado con el estado del arte en prebúsqueda de datos.

Estos resultados prueban que existe una gran oportunidad para predecir patrones temporales en cargas de trabajo OLTP y WEB. Como han indicado resultados de investigaciones anteriores [19], la cobertura de TMS es un poco menor que la de STeMS en este tipo de cargas de trabajo. No está del todo claro cuánto se aproxima A/AC a TMS debido al alto número de predicciones incorrectas. TMS utiliza varios buffers donde se almacenan varias secuencias de acceso para elegir el camino correcto en momentos donde

existe más de un camino a seguir desde una dirección determinada. Puede pensarse que debido a que son resultados aproximados la cobertura de A/AC y STeMS en comparación con DS-TMS no es tan distante como indican las gráficas. Aunque esto es cierto, nunca han sido publicados en ningún prebuscador unos resultados de cobertura cercanos al 85% en OLTP.

La figura 20 presenta una comparativa final entre nuestra propuesta DS-TMS y STeMS, que es el estado del arte en prebúsqueda de datos.

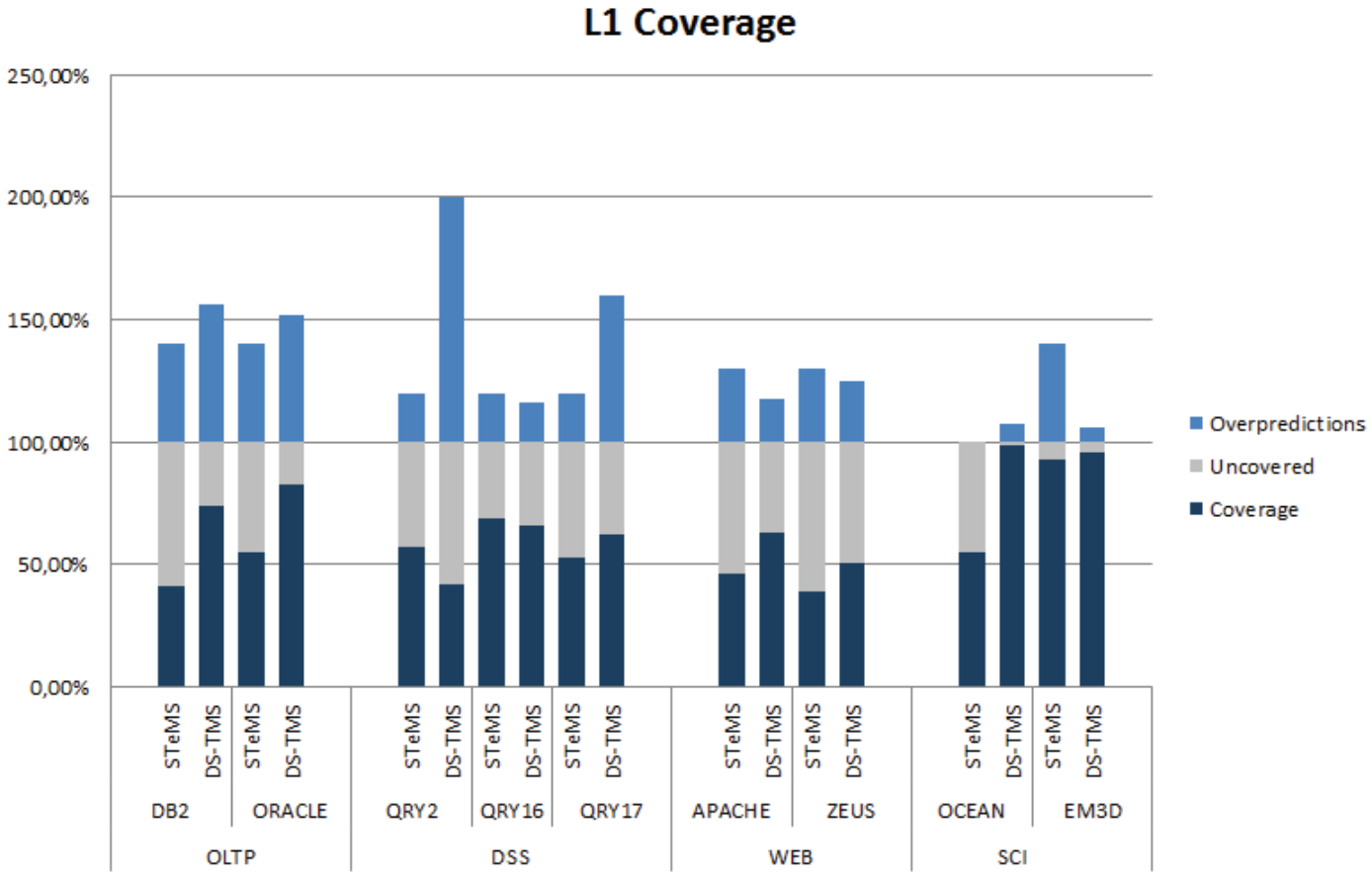

FIGURA 20. Resultados de cobertura de STeMS y DS-TMS

Comparativa entre DS-TMS y el estado del arte en prebúsqueda de datos.

Los resultados indican que hay una gran oportunidad para las delta direcciones en las secuencias de acceso temporal, incluido en aquellas donde un gran porcentaje del conjunto de datos es leído una sola vez, como en las cargas de trabajo DSS. De todas maneras, podemos observar que en alguna carga de trabajo como QRY2, los patrones

temporales entre regiones no son tan estables como los obtenidos dentro de la región, a pesar de utilizar delta direcciones. Las predicciones erróneas en este tipo de prebuscadores son generalmente grandes. DS-TMS muestra de media un 42% de predicciones de este tipo, estando en el mismo orden de magnitud que otros prebuscadores temporales. En cuanto a la complejidad hardware, este diseño tiene un coste de unos 100 MB por núcleo, lo que lo hace a día de hoy inviable en relación al almacenamiento dedicado a un prebuscador. Esta situación indica que deberemos realizar un diseño diferente si queremos implementar un prebuscador de delta direcciones en los sistemas actuales.

7.2.2 Resultados del prebuscador DS-SMS

Los siguientes resultados indican la oportunidad de DS-SMS para predecir patrones espaciales utilizando una tabla LT y PT infinitas.

Una de las limitaciones principales de estos prebuscadores es la propia forma en la que predicen. Si los patrones cambian su primer acceso dentro de la región o varios patrones tiene el mismo primer acceso dentro de la región, la cobertura de DS-SMS será mayor comparada con SMS. Es bastante improbable que las cargas de trabajo DSS se beneficien de DS-SMS debido a que los JOINS y los SCANS se aplican a todas las páginas de la misma forma. La figura 21 compara SMS con DS-SMS simulando cargas de trabajo DSS.

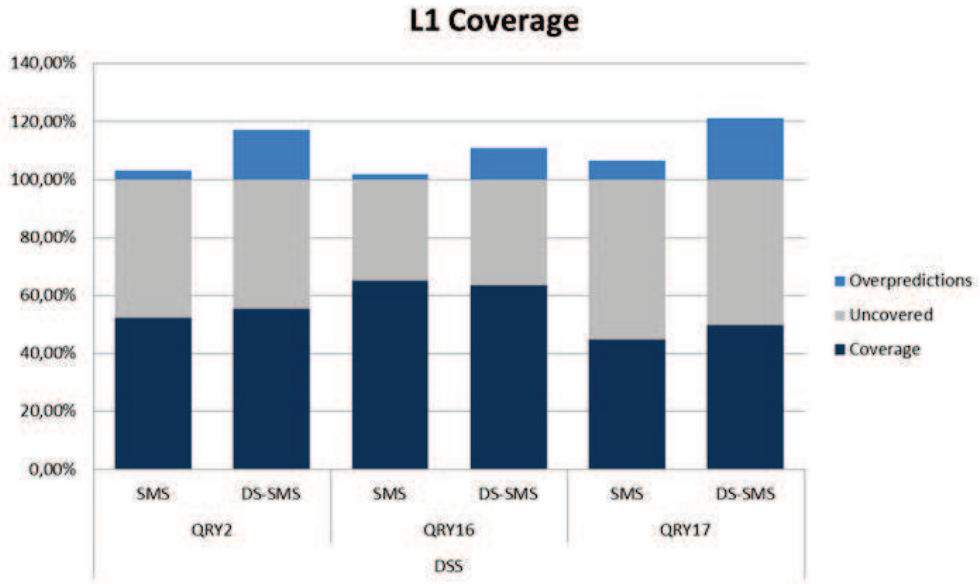

FIGURA 21. SMS comparado con DS-SMS en cargas de trabajo DSS.

La cobertura de SMS y DS-SMS en DSS.

Tal y como esperábamos, los beneficios de DS-SMS son casi despreciables debido a la gran estabilidad que tienen los patrones espaciales en este tipo de cargas de trabajo.

En cargas de trabajo OLTP y WEB, es posible que DS-SMS obtenga mayor cobertura que SMS. Esto es debido a que los patrones dentro de las regiones son más inestables debido a que se producen modificaciones de las páginas del gestor de bases de datos y diversos accesos a estructuras en árbol B+. La figura 22 muestra los resultados de cobertura con los prebuscadores DS-SMS y SMS en OLTP y WEB.

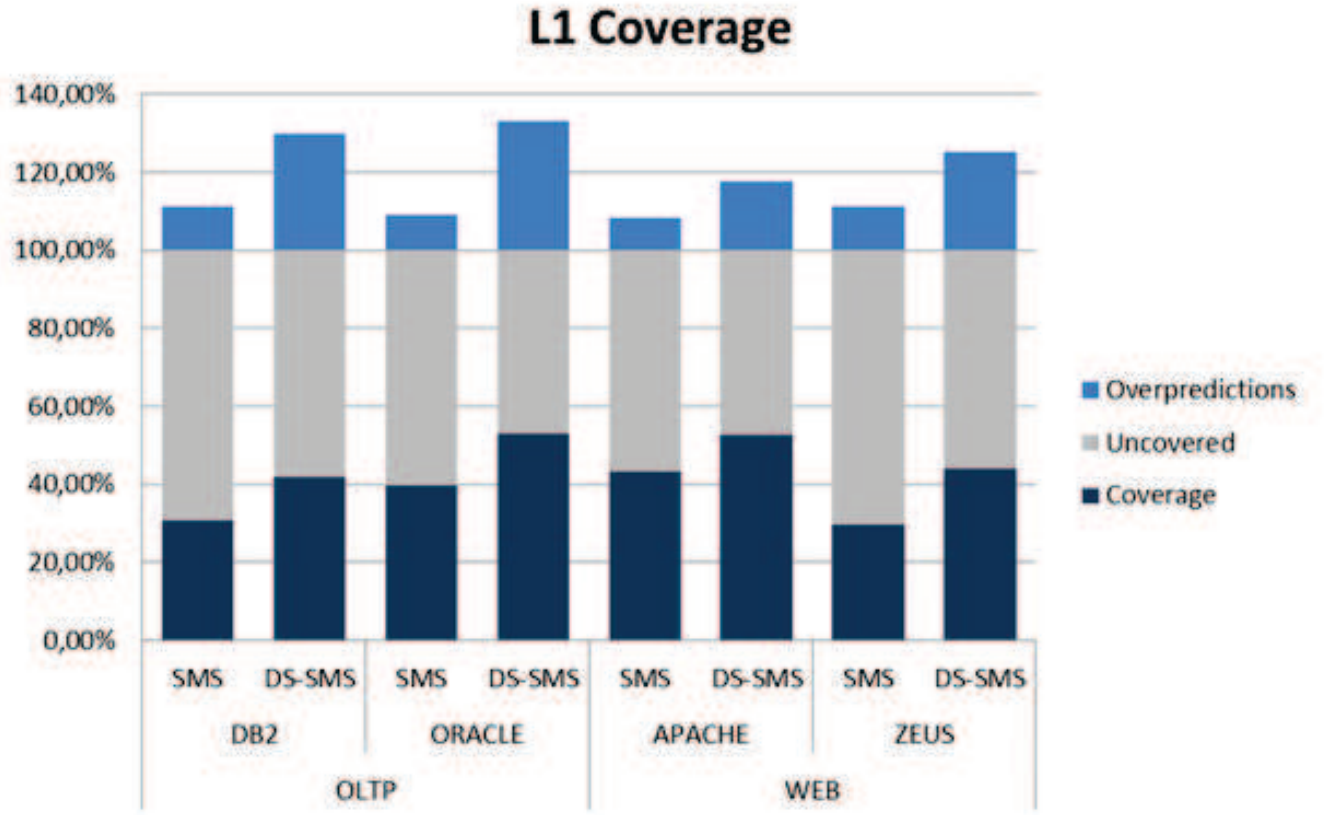

FIGURA 22. SMS comparado con DS-SMS en cargas de trabajo OLTP y WEB

La cobertura de SMS y DS-SMS en OLTP y WEB

DS-SMS es capaz de extraer patrones que SMS no es capaz en este tipo de cargas de trabajo. La razón de este incremento en la cobertura es que los patrones dentro de las regiones son menos estables que en las cargas de trabajo DSS. Mientras que en DSS los SCANS o JOINS acceden continuamente al mismo tipo de registros, los patrones espaciales

son siempre iguales en distintas regiones del espacio de direcciones. En las cargas de trabajo OLTO por ejemplo, se realizan accesos a árboles B+ y algunos registros son modificados en las páginas de los DBMSs. Debido a que los patrones recurren en los mismos registros, la disposición espacial de los accesos es diferente pero tienen muchas similitudes.

La figura 23 compara DS-SMS con la actual referencia en prebúsqueda de datos, STeMS. Esta última es una técnica híbrida que permite extraer patrones espaciales y concatenar temporalmente los primeros accesos a las regiones que son los que producen las predicciones espaciales.

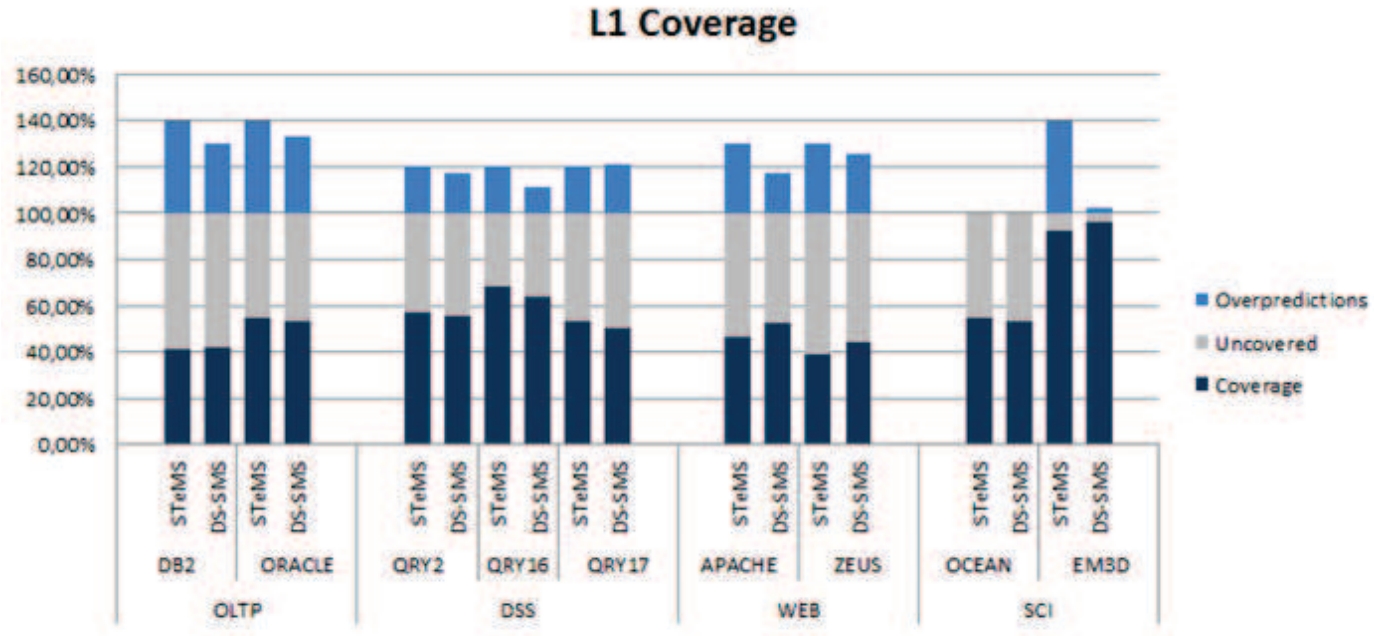

FIGURA 23. Resultados de cobertura de STeMS y DS-SMS

Estos resultados comparan DS-SMS con el estado del arte en prebúsqueda de datos

Los resultados indican que hay una fracción de patrones espaciales que no son predecibles por anteriores prebuscadores como SMS. Además, mirando a la gráfica podemos comprobar que la fracción de cobertura que STeMS puede conseguir en comparación con SMS es muy parecida a la que consigue DS-SMS en comparación con SMS. Esto es sorprendente ya que en la mayoría de casos esto se cumple a pesar de que, en DS-SMS las predicciones se limitan dentro de la región. Si bien es cierto que al disminuir la

granularidad de las predicciones a una por acceso, el número de predicciones no utilizadas es mayor, ya que el número de bloques referenciados por región es reducido. En media, las predicciones incorrectas representan solo el 17% en relación a los fallos de cache de L1 lo que es más que aceptable por el incremento en cobertura obtenido. La complejidad hardware necesaria para este diseño es de 10 MB por núcleo de procesador. Esto es un coste muy elevado para ser implementado en el interior del procesador, pero podría ser implementado en la memoria principal. No está del todo claro cuánto representa esta técnica en términos de rendimiento (SpeedUp) en relación a SMS y STeMS debido a que la lógica hardware está implementada dentro del procesador en estos dos últimos casos (en STeMS sólo la lógica que extrae patrones espaciales). Existen varios estudios que permiten reducir la lógica necesaria para la implementación de prebuscadores temporales y espaciales, utilizando meta-datos [12] y virtualización [13] con una penalización en rendimiento mínima.

7.2.3 Resultados finales de cobertura

Los resultados de cobertura de todos los prebuscadores en media en aplicaciones comerciales y científicas se presentan en la tabla 3.

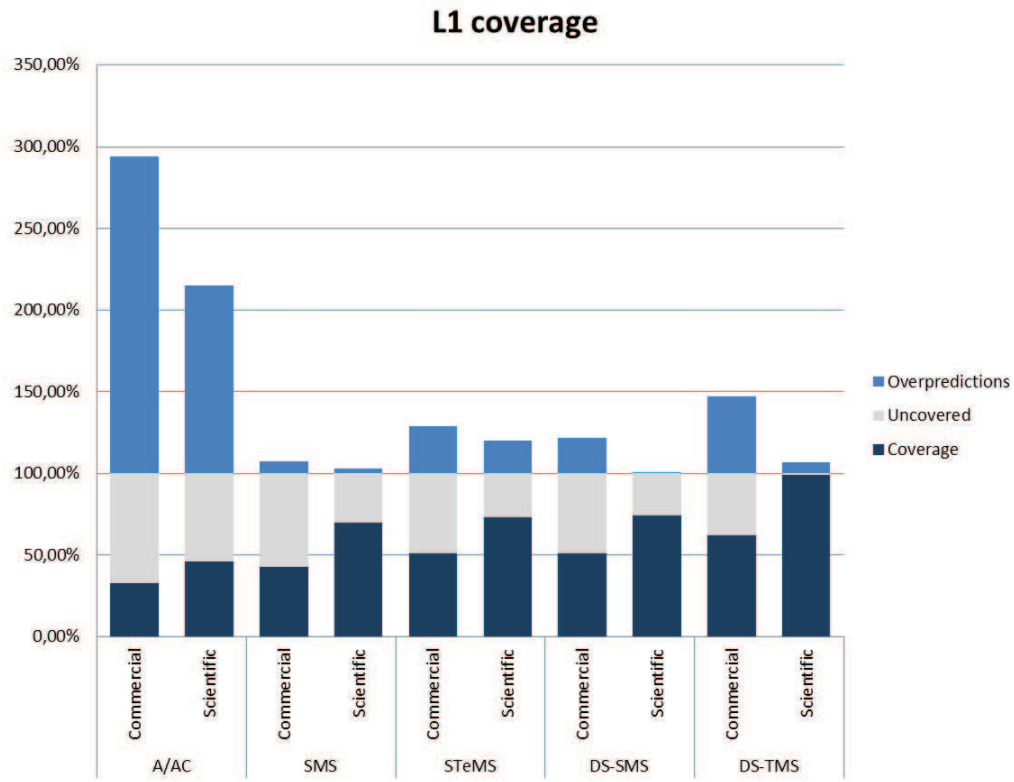

TABLA 3. Resumen de coberturas en media.

En resumen, SMS ofrece la mejor relación de cobertura/predicciones inútiles debido a la estabilidad de los patrones en regiones del espacio de direcciones y a su forma de predecir. DS-SMS obtiene mejor cobertura llegando al nivel de STeMS (estado del arte) debido a que la predicción por acceso obtiene mejores resultados en los prebuscadores espaciales permitiendo predecir secuencias de acceso que SMS es incapaz. Este trabajo demuestra que existen patrones espaciales que SMS no es capaz de predecir llegando a igualar a los fallos por correlación temporal entre regiones que STeMS si es capaz de predecir debido a su carácter temporal y espacial. Como punto negativo, las predicciones inútiles son superiores en DS-SMS debido a que se realizan cada vez que hay un acceso a la región y dado que el número de bloques que se referencian en una región es reducido [7], se realizan predicciones inútiles aunque el grado de prebúsqueda sea reducido. DS-TMS por su parte, obtiene mejores resultados de cobertura que la referencia en prebúsqueda de datos (STeMS) obteniendo inclusive resultados nunca antes publicados en OLTP. Si bien es cierto que DS-TMS tiene un porcentaje más elevado que STeMS de predicciones no referenciadas, estas se encuentran cercanas entre sí. Esto es debido a que la parte espacial de STeMS aglomera la mayor parte de predicciones realizadas por el prebuscador relegando a la parte temporal un mínimo número de referencias lo que ayuda a reducir el número de predicciones incorrectas.

Este trabajo demuestra el potencial de la indexación y predicción con delta-direcciones en aplicaciones comerciales llevando a resultados nunca antes visto, sin embargo, la complejidad hardware de DS-SMS y DS-TMS invita a rediseñar o aplicar técnicas para reducir la lógica necesaria en ambos diseños para ser asequible su implementación en procesadores actuales.

8. TRABAJOS RELACIONADOS

Propuestas recientes [2, 5, 7] han tratado de predecir patrones de acceso recurrentes en aplicaciones comerciales. Investigaciones anteriores [4] en prebúsqueda de datos se han centrado en aplicaciones científicas y de escritorio pero ninguna de ellas ha conseguido buenos resultados en aplicaciones de servidores debido a la enorme cantidad y complejidad de las secuencias de acceso a memoria. Fundamentalmente, se han utilizado dos ideas para predecir y extraer patrones en aplicaciones comerciales, la extracción de secuencias de acceso temporal y espacial. La primera se basa en el hecho de que las referencias tienen a repetirse a lo largo del tiempo y la segunda que los accesos a memoria poseen una disposición alineada a regiones del espacio de direcciones. Mediante la extracción de secuencias de acceso temporal podemos predecir las largas cadenas de operaciones de memoria dependientes como las que aparecen al atravesar un árbol B+. La principal limitación de estos predictores es que son incapaces de predecir referencias a memoria que no se hayan efectuado con anterioridad. El prebuscador DS-SMS por su parte, se basa en diferencias entre direcciones en vez de en posiciones fijas con lo que es capaz de predecir secuencias de acceso vistas con anterioridad o nuevas pero con la misma diferencia entre direcciones. Por otro lado, los predictores espaciales capturan patrones en regiones del espacio de direcciones y las extrapolan a otras. Este prebuscador permite buenos niveles de cobertura cuando los programas acceden a estructuras de datos con una disposición espacial fija como las páginas de gestores de bases de datos. La principal limitación de este tipo de prebuscadores es la unión que existe entre el primer acceso a la región y la predicción. Otra limitación importante es la forma de predecir, se realiza tan solo en el primer acceso a la región, si la predicción es errónea perderemos toda la oportunidad dentro de la región. DS-SMS por su parte, predice una vez por acceso. Lo que hace elimina la unión entre predicción y primer acceso, y además permite encaminar la predicción de forma más precisa cuantos más accesos existan en la región. Esto permite por un lado predecir patrones aunque el primer acceso a la región sea distinto o existen patrones entrelazados con un primer acceso común a la región, y por el otro, corregir la predicción en los sucesivos accesos a la región.

El punto de inicio del PFC era un nuevo prebuscador llamado *Prefetcher based on Differential Finite Context Machine* (PDFCM) [11], un prebuscador basado en delta direcciones obtenidas calculando la diferencia entre direcciones de memoria consecutivas referenciadas por la misma instrucción (correlación en código). De hecho, PDFCM, DS-TMS y DS-SMS tienen dos estructuras hardware, una para extraer patrones, otra para realizar las predicciones y su gestión es muy parecida. Aunque se ha demostrado que PDFCM

obtiene buenos resultados en cargas de trabajo SPEC comparables a otras técnicas como el PC/DC de Nesbit [4], obtiene una cobertura muy discreta en aplicaciones comerciales, muy lejos de otros como SMS o TMS. Uniendo la idea de extraer patrones mediante delta direcciones con la de los prebuscadores temporales y espaciales, se desarrollaron las ideas de finales de DS-TMS y DS-SMS.

9. CONCLUSIONES

Este PFC demuestra que existe oportunidad para extraer patrones temporales y espaciales en aplicaciones comerciales corroborando resultados previos en prebúsqueda de datos. Además, se establece la secuencia de delta direcciones como una herramienta poderosa debido a que permite predecir accesos a memoria nunca antes visitados y permite predecir por acceso en regiones espaciales. Utilizando dos estructuras hardware, somos capaces de extraer y predecir patrones temporales y espaciales. A través de una simulación funcional de un procesador multi-núcleo, con jerarquía de memoria y con un protocolo de coherencia de caches, demostramos que: (1) el prebuscador DS-TMS obtiene mejor cobertura que STeMS manteniendo las predicciones inútiles en el mismo orden de magnitud, siendo este último el estado del arte en prebúsqueda de datos en aplicaciones comerciales y científicas, y (2) el prebuscador DS-SMS es capaz de aprovechar mejor los patrones espaciales que SMS, obteniendo resultados simulares a STeMS, el cual posee también correlación temporal. Sin embargo, la complejidad hardware necesaria para la implementación de ambas propuestas hace imposible su implementación en procesadores actuales. Diversas técnicas utilizando meta-datos y virtualización en las caches de último nivel sería el siguiente paso a realizar para un diseño adaptado a las restricciones tecnológicas actuales.

Personalmente el proyecto me ha aportado una visión mucho más clara de lo que la investigación realmente es. Además de descubrir cuáles son los retos actuales en arquitectura de computadores, la forma de realizar una investigación es lo que más me va a ayudar en mis planes futuros. Las incontables horas trabajando en el laboratorio PARSA de la EPFL han hecho incrementar mis conocimientos de una forma que no hubiera sido posible de ningún otro modo. Voy a seguir trabajando en este proyecto como estudiante de doctorado en la EPFL.

REFERENCIAS

[1] A. Ailamaki, D. J. DeWitt, M. D. Hill, and D. A. Wood. DBMSs on a modern processor: Where does time go? In The VLDB Journal, pages 266–277, Sept. 1999.

[2] Stephen Somogyi, Thomas F. Wenisch, Anastasia Ailamaki and Babak Falsafi. Spatio-Temporal Memory Streaming In the 36th Annual International Symposium on Computer Architecture (ISCA), June 2009.

[3] Haitham Akkary, Michael A. Driscoll. A Dynamic Multithreading Processor.

[4] K. J. Nesbit and J. E. Smith. Data cache prefetching using a global history buffer. In Proceedings of the Tenth Symposium on High-Performance Computer Architecture, Feb. 2004.

[5] T. F. Wenisch, S. Somogyi, N. Hardavellas, J. Kim, A. Ailamaki, and B. Falsafi. Temporal streaming of shared memory. In Proceedings of the 32nd International Symposium on Computer Architecture, June 2005.

[6] Pedro Díaz, Marcelo Cintra. Stream Chaining: Exploiting Multiple Levels of Correlation in Data Prefetching in the 36th Annual International Symposium on Computer Architecture (ISCA), June 2009.

[7] Stephen Somogyi, Thomas F. Wenisch, Anastassia Ailamaki, Babak Falsafi, and Andreas Moshovos. Spatial memory streaming. In Proceedings of the 33rd International Symposium on Computer Architecture, June 2006.

[8] G. Kandiraju and A. Sivasubramaniam. “Going the distance for TLB prefetching: an application-driven study”, in Proceedings of the 29th Annual International Symposium on Computer Architecture, 2002.

[9] Y. Sazeides and J. E. Smith, “Implementations of context based value predictors”, TRECE97-8, Dept. of Electrical and Computer Engineering, Univ. Wisconsin/Madison, Dec. 1997.

[10] Thomas F. Wenisch, Roland E. Wunderlich, Michael Ferdman, Anastassia Ailamaki, Babak Falsafi, and James C. Hoe. SimFlex: statistical sampling of computer system simulation. IEEE Micro, 26(4):18–31, July-Aug. 2006

[11] Ramos, L.M, Briz, J.L., Ibáñez, P. and Viñals, V. "Multi-level Adaptive Prefetching based on Performance Gradient Tracking". Journal of Instruction Level Parallelism (JILP) Vol. 13, January 2011 (ISSN 1942-9525).

[12] Thomas F. Wenisch, Michael Ferdman, Anastasia Ailamaki, Babak Falsafi and Andreas Moshovos. Practical Off-chip Meta-data for Temporal Memory Streaming. Proc. of the 15th International Symposium on High Performance Computer Architecture (HPCA) , Feb. 2009.

[13] Ioana Burcea, Stephen Somogyi , Andreas Moshovos, Babak Falsafi, “Predictor Virtualization”. In Proceedings of the 13th International conference on Architectural Support for Programming Languages and Operating Systems.