Modelo de Costos para Bases de Datos en Memoria Principal

José Luis Martí L. y Horst H. von Brand
Departamento de Informática - Universidad Técnica Federico Santa María
Av. España 1680, Valparaíso – Chile
Teléfono: 56-32-654242 - Fax: 56-32-797513
{jmarti, vonbrand}@inf.utfsm.cl

Resumen

El modelo de costos de un optimizador de consultas es un componente esencial de cualquier sistema administrador de bases de datos y, en especial, para un sistema basado en la memoria principal, en el cual los factores de costos son mucho más, en número y complejidad, que aquéllos usados por los tradicionales sistemas basados en el disco. El presente trabajo ejemplifica una propuesta destinada a plantear un modelo de costos para tal entorno, la que se basa en expresiones genéricas construidas a partir de funciones analíticas de costos, cuyos coeficientes son determinados por muestreo y regresión múltiple; este esquema es fácil de implementar y de incorporar en un sistema comercial.

Palabras Claves: Bases de Datos en Memoria Principal, Modelos de Costos.

1 Introducción

Un sistema de bases de datos en memoria principal (BD-MP) se define como un sistema en el cual la copia principal de los datos se encuentra en la memoria principal, con lo cual todo el procesamiento de consultas y transacciones, y la administración del almacenamiento de datos, se hacen sin necesidad de usar el disco, salvo por el hecho de registrar en éste los registros de bitácora asociados con las operaciones de modificación de datos, dada la persistencia que presenta.

El diseño del sistema administrador de una base de datos en memoria principal (SABD-MP) contempla varias diferencias con respecto a una base de datos tradicional, basada en el disco. En el caso particular del procesamiento de consultas, el modelo de costos a usar debe considerar aspectos relacionados con parámetros propios de la forma de ejecutarlo y con la estructura de la memoria principal del computador. Surge, en lugar de una recuperación desde el disco, una que contempla el rescate de datos desde la memoria principal hacia los *cachés*, operación que se debe tomar en cuenta desde el diseño mismo de las estructuras de almacenamiento e indexación que se vayan a usar, pues de lo contrario el resultado puede ser perjudicial para el sistema.

Escasos son los trabajos que a la fecha se han centrado en el tema de un modelo de costos para BD-MP y que contemplen la influencia de la estructura jerárquica de la memoria principal. Sin embargo, la relevancia que los sistemas basados en la memoria principal tendrán a futuro, y el papel fundamental que juega el modelo de costos para un procesamiento de consultas adecuado, hacen necesario profundizar el tema y plantear una metodología de construcción que permita generar modelos de costos razonablemente precisos y simples.

El presente trabajo presenta un esquema para generar un modelo de costos para BD-MPs y obtener funciones de costo que apoyen adecuadamente la labor de optimización de consultas. La estructura del trabajo comienza con una descripción general de los sistemas de memoria principal de los computadores modernos, indicando el impacto que tienen en los tiempos de procesamiento, para luego pasar a la sección tres, donde se explica las etapas y pasos seguidos en la construcción de un modelo de costos para una BD-MP. La sección cuatro desarrolla la metodología propuesta, a

través de la construcción de las expresiones de costo para una operación propia de BD-MP. Finalmente se presentan las conclusiones del trabajo.

2 Trabajos Relacionados

Los trabajos de investigación recientes en el tema de las BD-MPs se han centrado, casi exclusivamente, en estudiar algoritmos y estructuras de datos que permitan sacarle mejor provecho al sistema de memoria, dando origen a técnicas bastante diferentes a las planteadas hace ya más de una década [Bonc99], [Mane00], [Rao00]. Los resultados permiten apoyar el procesamiento de una consulta, dado que representan alternativas a considerar tanto en la optimización como en la ejecución misma de la consulta. Sin embargo, la literatura sobre BD-MP incluye pocos trabajos relacionados con el modelo de costos, entendiéndose por éste al conjunto de funciones de costo que permiten estimar el costo de incluir y ejecutar una operación dentro del plan de ejecución de una consulta.

Estudios sobre modelos de costos para BD-MPs consideran los trabajos de [Swam89], [Whan90], [List96] y [Mane00], cada uno de los cuales sigue un esquema de construcción distinto, basándose en un conjunto de parámetros no del todo compatibles. [Swam89] y [Whan90] trabaja con un modelo analítico para representar el procesamiento llevado a cabo en la *CPU*, el que no toma en cuenta la estructura de sistema de memoria principal debido a que fueron planteados cuando la memoria principal era de un nivel y no existían niveles de *cachés*. En el trabajo de [List96] se genera un modelo construido mediante calibración, para obtener los coeficientes de expresiones lineales genéricas, para representar la misma situación considerada por los dos trabajos anteriores, sin entregar una base más completa para justificar el uso de dichas expresiones lineales, a la vez que éstas son dependientes de la BD-MP analizada. En [Mane00] se aplica un modelo híbrido que combina calibración, a través de un componente que resume todos los costos de la ejecución menos los relacionados con el sistema de memoria, obtenido con la medición de diversas ejecuciones hechas con consultas de prueba, con expresiones analíticas que modelan los costos ocasionados por los constantes traslados de datos entre los distintos niveles de la memoria.

De todos los señalados, sólo el trabajo de [Mane00] se acerca al nuestro, por tomar en cuenta la influencia de la memoria principal dentro de los costos del procesamiento, pero su método de trabajo contempla una etapa de calibración inicial que puede no ser simple de hacer. Así, el objetivo de nuestro estudio es plantear un esquema de construcción diferente (alternativo), más simple de obtener y de una precisión razonable.

3 Aspectos de Diseño del Modelo de Costos

La construcción de un modelo de costos para BD-MPs es un trabajo que no puede aislarse de las características estructurales que presenta el sistema jerárquico de memoria del computador sobre el cual se estén ejecutando las consultas, el que se compone de cuatro componentes (ver figura 1):

- Un primer nivel de *caché,* llamado *caché* L1, incluido al interior del chip del procesador mismo.
- Un segundo nivel de *caché,* llamado *caché* L2, fuera del chip del procesador pero dentro de la tarjeta del sistema.
- La memoria principal misma.
- El sistema de memoria virtual.

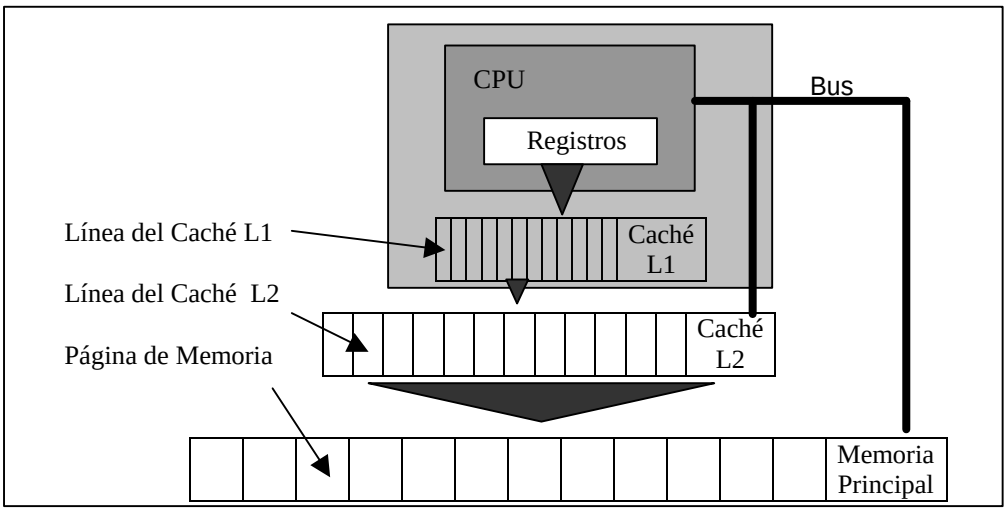

Figura 1: Estructura del sistema jerárquico de memoria de los computadores modernos.

Las mejoras a la memoria han contribuido a reducir el tiempo de ejecución, pero a costa de una mayor exigencia y complejidad del hardware, y a dificultar la obtención de expresiones destinadas a estimar en forma precisa dicho tiempo. De acuerdo a resultados obtenidos en [Mane00] y [Aila99], el tiempo de ejecución es la suma de cuatro términos:

- Tiempo de procesamiento o computación.
- Demoras relacionadas con el sistema jerárquico de memoria: que resume las demoras debido a *misses* en el *caché* L1 de datos (con un *hit* en el L2), en el *caché* L1 de instrucciones (con un *hit* en el L2), en el *caché* L2 de datos, en el *caché* L2 de instrucciones, en la memoria asociativa de datos y en la de instrucciones.
- Penalidades por mala predicción del siguiente camino de ejecución.
- Esperas por los recursos: ocasionados por la no disponibilidad de las unidades funcionales, dependencias entre las instrucciones y a características específicas de la plataforma.

4 Construcción del Modelo de Costos

La construcción de cualquier modelo de costos abarca dos etapas [List96]: diseño y validación. La primera consiste en la definición de los parámetros de costo y las funciones de costo para las operaciones de la máquina de ejecución, mientras que la segunda se centra en realizar una serie de pruebas experimentales para validar la exactitud del modelo planteado en la etapa anterior.

La etapa de diseño considera la utilización de diversas técnicas, como el método de construcción (modo de elaborar la estructura de las funciones de costos), la métrica de costo (criterio escogido para medir el costo de la operación asociada), los factores de costo (componentes de costos a usar) y el tipo de utilización que se le da al modelo de costos. Su adecuada inclusión va a la par con los diversos parámetros a incluir en los términos de las funciones de costos, parámetros que van a representar propiedades de los datos, de los componentes de la consulta y del ambiente de ejecución [Mart01].

Para construir el modelo de costo, nuestra propuesta se basa en la aplicación de una técnica híbrida, de dos etapas. El contexto sobre el cual se basa es uno centralizado y serial, y representa un primer paso a un estudio más amplio destinado a generar modelos de costos para sistemas de bases de datos en memoria principal, distribuidas y paralelas. Por esta razón es que los factores de costo a considerar son el tiempo de procesamiento y los retardos relacionados con los traspasos de datos dentro del sistema jerárquico de memoria. Dado que se pretende apoyar la labor de un optimizador

de consultas, no es necesario medir ni modelar en forma separada cada uno de los retardos de la memoria; así, su influencia de éstos se puede considerar ya sea en algún componente puntual o dispersa a lo largo de los términos de la función de costos. La métrica de costo a tomar en cuenta es el tiempo de respuesta, dado que el criterio que más se utiliza y porque es adecuado para determinar la forma de evaluación más conveniente desde el punto de vista de un usuario deseoso de tener respuestas rápidas.

El método propuesto se resume en dos etapas:
- Modelamiento analítico, de las expresiones correspondientes al tiempo de procesamiento y al tiempo relacionados con los retardos de la memoria.
- Regresión múltiple, para poder incorporar el efecto de la mala predicción del camino de ejecución y la no disponibilidad inmediata de los recursos de la memoria.

La primera etapa comienza con la definición de los parámetros de costo básicos para luego continuar con la formulación de las expresiones analíticas. Para simplificar este resultado inicial, a continuación se realiza un análisis de sensibilidad de los parámetros utilizados, el que se debe llevar a cabo haciendo diversas variaciones en los valores de los parámetros. A partir de este paso, se eliminan aquellos parámetros que no son muy influyentes y seguir con los que si tienen sentido, los que constituyen una expresión de costo genérica, acompañados de coeficientes C_i, donde i está comprendido dentro del intervalo [0..Número de parámetros].

La segunda etapa se basa en la determinación de los coeficientes de regresión y en el análisis de la bondad del ajuste obtenido. El método de determinación de coeficientes contempla un conjunto de consultas, cuyos tiempos de ejecución se usan para calcular el valor de dichos coeficientes, según las ecuaciones de regresión descritas en la literatura correspondiente [Nete90]. Varias corridas se llevan a cabo por cada algoritmo, de modo de tener algunos conjuntos de resultados, y así identificar aquél que presenta mayor confiabilidad en cuanto a los tiempos medidos, y lograr un mejor ajuste. Este último es calculado al llevar los datos a un paquete estadístico que calcula regresiones y las estadísticas relacionadas. De estas estadísticas, son de especial interés, las llamadas R^2 ajustado y *F*; la primera es una estadística refinada para estimar cuán bien se ajusta el modelo en la población, mientras que la segunda refleja la influencia que tiene la o las variables presentes en el modelo. Un buen modelo debe tener un valor cercano a 1 para R^2 ajustado y un valor alto para *F*.

La combinación de dos métodos (modelamiento analítico y regresión múltiple) obedece a que no hay un esquema único capaz de modelar los retardos producidos por la mala predicción del camino de ejecución y la no disponibilidad inmediata de los recursos de la memoria; dada la variabilidad que éstos presentan, que depende de las características de la arquitectura del computador y del compilador que se esté usando, su modelamiento es un punto difícil de atacar. Lo que se puede modelar en forma adecuada son el tiempo de procesamiento y el número de traspasos de datos dentro del sistema jerárquico de memoria; de ahí el uso de un esquema híbrido para considerar todos los aspectos presentes.

5 Un Ejemplo Práctico

Para ejemplificar la validez del método, en esta sección se desarrolla la expresión de costo de una operación propia de BD-MPs, que es la búsqueda indexada usando un árbol B^+ conciente del *caché* [Rao00]. Esta estructura de datos es una adaptación del clásico árbol B^+, que en lugar de almacenar claves y punteros a los sub-árboles de niveles inferiores, guarda sólo las claves y un puntero a un grupo de nodos, los que se direccionan por la posición relativa que tengan del primer

nodo del grupo (ver figura 2). La característica de conciente del *caché* está dada porque el nodo guarda, prácticamente, sólo claves, y no en una proporción del 50% para cada caso como en un árbol tradicional, con lo que en el mismo nodo, y por ende en una misma entrada de *caché*, se pueden tener más claves para la comparación, disminuyendo el número de desplazamientos de datos entre los niveles de la memoria. El recorrido sobre este índice es similar al que se realiza para el caso de un índice B^+ tradicional, con la diferencia de que el avanzar al siguiente nivel no se hace siguiendo un puntero directo al nodo descendiente. En su lugar, se sigue el puntero al grupo de nodos del siguiente nivel, y se accesa uno de los nodos que agrupa; este último se determina de la siguiente manera: si la clave de búsqueda es mayor que el valor de la clave de la posición *i* del nodo de nivel superior y, menor o igual que el valor de la clave de la posición *i+1*del mismo nodo, entonces se debe accesar el nodo de la posición *i+1* del grupo de nodos del nivel siguiente.

A pesar de que los árboles T han sido la estructura de indexación clásica de los sistemas de BD-MP, recientes estudios han cuestionado su eficiencia, dándole ventajas a los árboles B^+ con conciencia de *caché* [Lu00], [Mane00]. Así, un análisis basado en éstos es una buena decisión, sobre todo pensando en la construcción de sistemas de BD-MP a corto plazo.

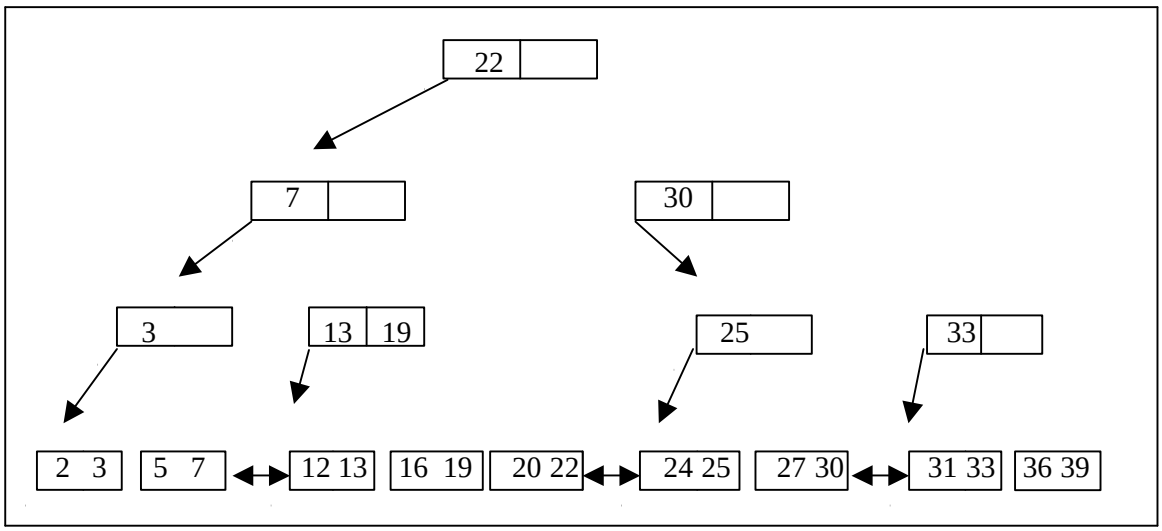

Figura 2: Ejemplo de árbol B^+ con conciencia del *caché*.

5.1 Primera Etapa: Diseño del Modelo mediante Modelamiento Analítico

El primer paso para el diseño del modelo es la identificación de los parámetros de costo. Para este caso se consideran los siguientes:

- Parámetros que representan propiedades de los datos: **número de registros del archivo** (**C**), factor básico por cuanto mientras más grande sea dicho número, más tiempo cuesta ejecutar cualquier operación sobre el archivo, y más memoria puede ser necesaria usar; ***fan-out* del índice** (**f**), número máximo de entradas que contiene cada nodo del índice, a mayor *fan-out*, mayor es el número de comparaciones que se deberán realizar al interior del nodo; y **el número de niveles** (**n**): factor que incorpora latencia para alcanzar el o los registros requeridos desde el archivo.
- Parámetros de las propiedades de los componentes de la consulta: **cardinalidad de la selección** (**s**): corresponde a un valor que determina el número de registros dentro de un archivo que cumplen la condición impuesta por la operación de búsqueda.

- Parámetros del ambiente de ejecución: **tiempo de comparación en un nodo** (t_{NODO}), necesario para considerar el tiempo que tarda el sistema con cada entrada de un nodo del índice; **tiempo para accesar el contenido de una página de** datos ($t_{CHEQUEO}$); y el **tiempo para escribir un registro en la salida** ($t_{ESCRITURA}$), que permite modelar el costo unitario de incluir un registro en la salida de la operación.

Definición de las Expresiones Analíticas para el Costo

Los costos de procesamiento y de acceso a la memoria asociados a la operación aparecen en la tabla 1. Para esta labor, hay que considerar dos componentes desde el punto de vista del procesamiento de CPU, uno relacionado con el acceso al índice y otro que señala la recuperación de datos desde el archivo. En el primer caso, hay que recuperar tantos nodos como niveles tenga el índice, por cuanto todos las hojas están en el mismo nivel, y es preciso llegar a este punto para recién alcanzar los datos del archivo. La recuperación de datos desde el archivo es de *s* páginas para la búsqueda sobre un atributo con duplicados.

En las expresiones aparecen, también, algunos términos que corresponden a aspectos relacionados con la estructura de memoria. Éstos se refieren a tres latencias, que corresponden al tiempo para accesar un dato desde la memoria asociativa (l_{TLB}), el tiempo para accesar un dato desde la memoria misma (l_{MEM}) y el tiempo para accesar un dato desde el *caché* L2 (l_{L2}).

Notar que el proceso de búsqueda presenta un grado de aleatoriedad no bajo, lo cual es un punto importante al momento de ya tener la expresión de costo, pues dado que existen duplicados del valor de la búsqueda, pueden aparecer uno o varios registros asociados.

Costos de CPU	
Recorrido del árbol	$n * f/2 * t_{NODO}$
Acceso al Archivo de Datos	$s * t_{CHEQUEO}$
Escritura del resultado	$s * t_{ESCRITURA}$
Costos de Acceso a Memoria	
Carga de las páginas de datos a la memoria asociativa	$(n + s) * l_{TLB}$
Carga de los datos al *caché* L2	$(n + s) * l_{MEM}$
Carga de los datos al *caché* L1	$(n + s) * l_{L2}$
Escritura del resultado al *caché* L2	$s * l_{L2}$
Escritura del resultado a la memoria	$s * l_{MEM}$
Escritura sobre la memoria asociativa	$s * l_{TLB}$

Tabla 1: Costos de la búsqueda indexada sobre un atributo ordenado ascendentemente, con duplicados.

Análisis de Sensibilidad de las Expresiones de Costo

Esta actividad se concretiza con la codificación y ejecución de un programa escrito en C, el cual contiene los términos de costos presentes en la expresión analítica. El programa es ejecutado con diversos tipos de variaciones en los valores de los parámetros, de modo de medir los cambios en el resultado final. Los valores de los parámetros y los resultados son capturados en tablas y gráficos que permiten confirmar o rechazar la influencia de un parámetro en la expresión de costo general. La versión de compilador, la arquitectura de computador, y los retardos y capacidades de cada nivel de la memoria que se usan en la codificación del algoritmo están señalados en la tabla 2. El dominio utilizado para los valores de los parámetros se indican en la tabla 3.

Configuración Básica

Sistema Operativo	Linux 2.2.5
Tamaño de la RAM	512 MB
Procesador	Intel Pentium III
Velocidad Procesador	450 Mhz
Compilador	gcc 2.91.66

Retardos de la Memoria

Latencia *caché* L2	42.2 ns
Latencia de memoria	93.3 ns
Latencia TLB	11.1 ns

Sistema de Memoria

Tamaño *caché* L1	16 Kb
Tamaño línea *caché* L1	32 bytes
Líneas *caché* L1	512
Tamaño *caché* L2	512 KB
Tamaño línea *caché* L2	32 bytes
Líneas *caché* L2	16.384
Número Entradas TLB	64
Tamaño de la Página	4 KB
Tamaño Entrada TLB	256 KB

Tabla 2: Parámetros del sistema a considerar en el análisis de sensibilidad.

Parámetro	Valores Usados
Número de registros del Archivo	16 K, 32 K, 128 K, 256 K, 512 K, 784 K. 1 M .. 4M, con una separación de 0.25 M 4 M .. 16 M, separados por 1 M
Cardinalidad de la Selección	1, 16, 32, 64, 128, 256, 512, 1024
Tamaño de la Entrada (en bytes)	8, 16, 32

Tabla 3: Dominio de los parámetros a usar en el análisis de sensibilidad.

Se comienza el análisis estudiando el parámetro relacionado con el número de entradas de un nodo del índice. Este es un factor de relativa relevancia en el tiempo de respuesta, según muestran las curvas de la figura 3, las que corresponden a casos con distintos número de niveles. Cada curva es prácticamente una recta que cruza en diagonal el plano dibujado, lo cual permite concluir la necesidad de incluir este parámetro dentro de la expresión de costo.

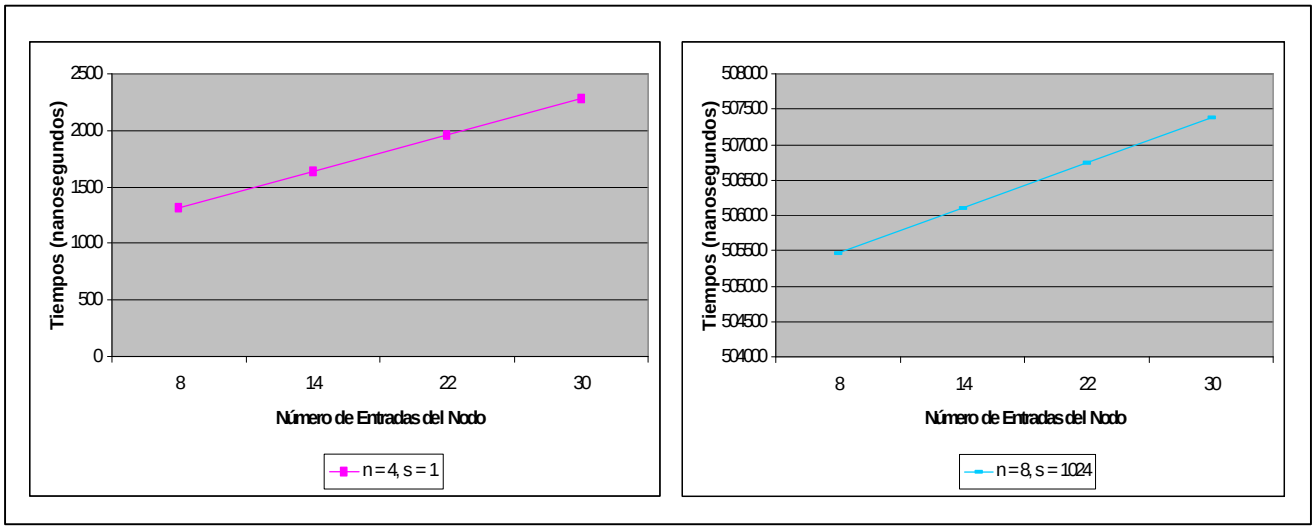

Figura 3: Diferentes relaciones entre el número de entradas de un nodo y el tiempo de respuesta.

Para corroborar lo anterior, la siguiente tabla muestra la variación porcentual asociada con este parámetro, la cual no es despreciable.

			Diferencia Porcentual
Número de Entradas del Nodo	8	14	75%
Tiempos obtenidos	2140	2780	30%

En relación a la cardinalidad de la selección, ésta guarda una relación de dependencia lineal con el tiempo de respuesta. A modo de ejemplo, la figura 4 muestra la curva correspondiente para un índice de cuatro niveles. La relación es lineal, no cuadrática, pues los valores usados para la cardinalidad de la selección son sólo potencias de dos. Se visualiza una diferencia importante entre el tiempo menor y el mayor, producto de la variación en el parámetro referido, por lo que se justifica su inclusión en la ecuación de regresión.

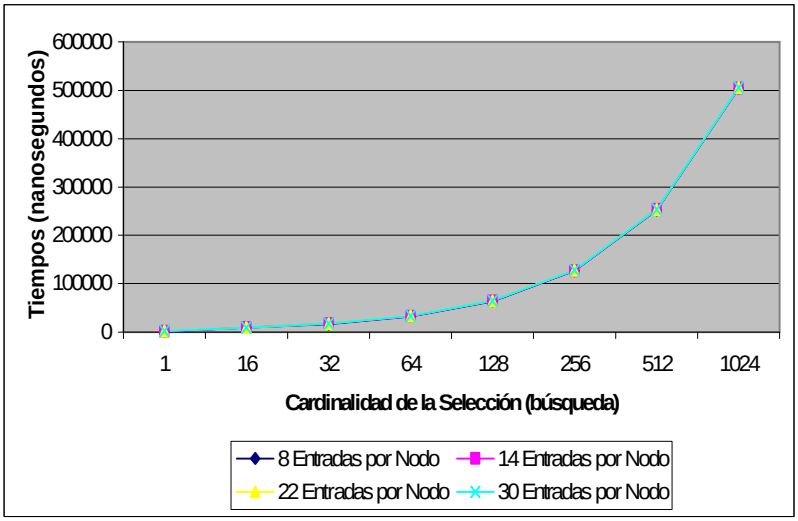

Figura 4: Ejemplo de relación para la cardinalidad de la selección y el tiempo de respuesta.

En cuanto al parámetro del número de niveles, también se visualiza una relación lineal (ver figura 5). Al analizar las variaciones porcentuales, expresadas en la tabla 4, se desprende que la influencia es relativamente fuerte si la cardinalidad es baja, pero a medida que ésta aumenta, la influencia disminuye paulatinamente.

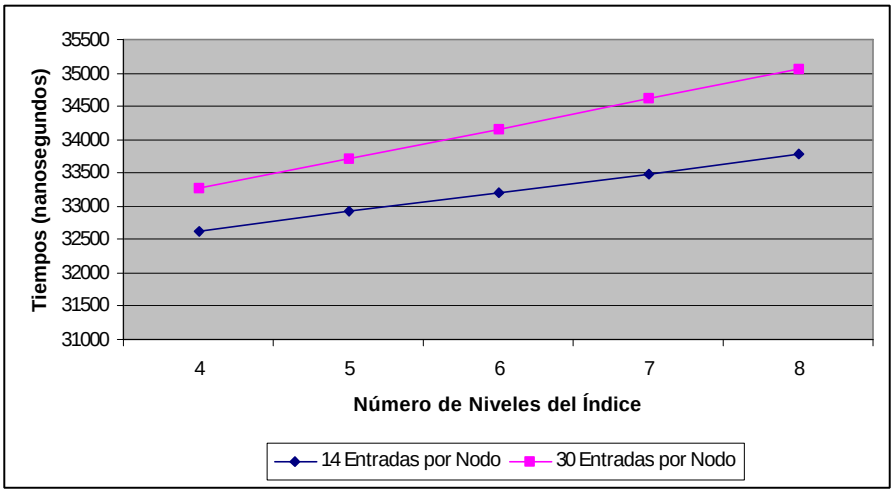

Figura 5: Ejemplo de relación entre el número de niveles del índice y el tiempo de respuesta, para una cardinalidad de selección de 64 registros.

Cardinalidad de la Selección	Número de Niveles del Índice		Número de Entradas del Nodo: de 14 a 30 (114%)
1	4		39 %
1	6		43 %
1	8		46 %
64	6		3 %
64	8		4 %

Tabla 4: Variaciones de los parámetros de la búsqueda indexada.

Luego, a pesar de que su influencia decrece con una cardinalidad de selección creciente, es un factor que debe seguir formando parte del modelo a trabajar.

Finalmente, la ecuación de regresión asociada tiene la forma: $T = C_0 + C_1 * n + C_2 * f + C_3 * s$, la cual es lineal en todos los parámetros incluidos, que son los mismos que se incorporaron desde un comienzo. La aleatoriedad de la operación está centrada en el último parámetro, pues su valor puede variar desde 1 hasta un límite no determinado.

5.2 Segunda Etapa: Validación del Modelo mediante Regresión Múltiple

Esta etapa consiste en la codificación el algoritmo de búsqueda indexada mismo, usando el árbol B^+ conciente del *caché*, y técnicas de programación destinadas a sacar mejor provecho del uso del *caché* [Shat94]. Para realizar la medición de los tiempos, se hace uso de un sistema de dominio público, denominado PERFCTR [Perf00], y que implementa *drivers* de bajo nivel a través del cual es posible rescatar las medidas de rendimiento sobre un sistema de computación. PERFCTR es una familia de parches para un kernel tipo UNIX, con una biblioteca de funciones destinadas a medir de rendimiento del hardware. Todas las funciones son invocables desde programas escritos en C, condición necesaria realizar el trabajo, dado que es el lenguaje de programación en uso. En cuanto a los dominios para los parámetros, corresponden a los ya usados para el análisis de sensibilidad.

Tras la ejecución del algoritmo correspondiente, se obtienen las estadísticas de la tabla 5, asociadas a diversas curvas de ajuste que se pueden obtener combinando los parámetros involucrados, y que se muestran para efectos de comparar las medidas obtenidas. El mejor ajuste es aquél que contiene sólo el número de niveles del índice y la cardinalidad de la selección; sin embargo, la curva propuesta para el estudio no se aleja de la bondad que presenta la primera; de hecho las estadísticas del R^2 ajustado son prácticamente iguales. Por lo mismo, cualquiera de las dos curvas es un buen ajuste, con la salvedad de que la ecuación que contiene sólo a *n* y a *s* es más simple.

Variables incluidas	**R^2 Ajustado**	**F**
n	0,006	1,414
f	0,003	0,800
s	0,730	176,876
n, f	0.008	0,756
n, s	0,748	97,522
f, s	0,739	92,836
n, f, s	0,746	64,637

Tabla 5: Estadísticas asociadas a diversas curvas de ajuste.

Al llevar a un gráfico las relaciones de los parámetros con el tiempo de ejecución se derivan las siguientes observaciones. En el caso del número de niveles, la relación gráfica presente en la

figura 6 no es muy clara, pues los puntos tienden a repartirse en forma homogénea dentro de un rectángulo, lo que no deja entrever una recta representativa de la relación entre ambos factores.

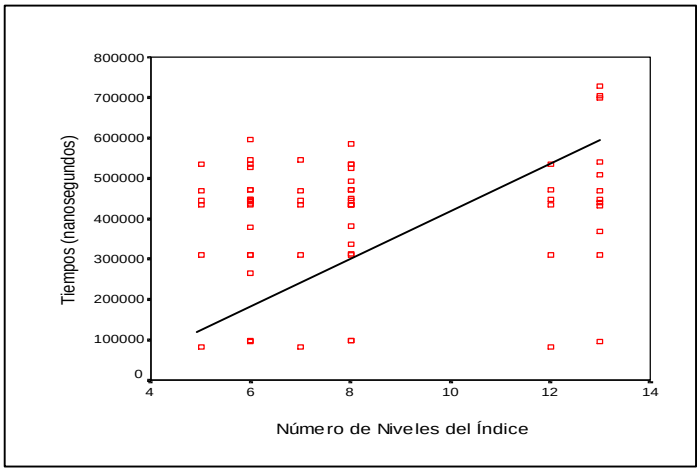

Figura 6: Relación entre el número de niveles del índice y tiempo de ejecución.

En cuanto al número de entradas del nodo interno de un índice, la relación tampoco es muy evidente (ver figura 7a); de hecho la gráfica no deja ver algún tipo de relación, quedando sólo el respaldo estadístico para saber como es ésta. Finalmente, al graficar la relación con la cardinalidad de la selección, si se tiene una evidente relación lineal; la figura 7b confirma esta afirmación.

Como conclusión, el ajuste obtenido es bueno. La estadística del R^2 ajustado tiene un valor igual a 0,746, lo que se ajusta a la aleatoriedad que tiene el valor del parámetro de la cardinalidad de la selección.

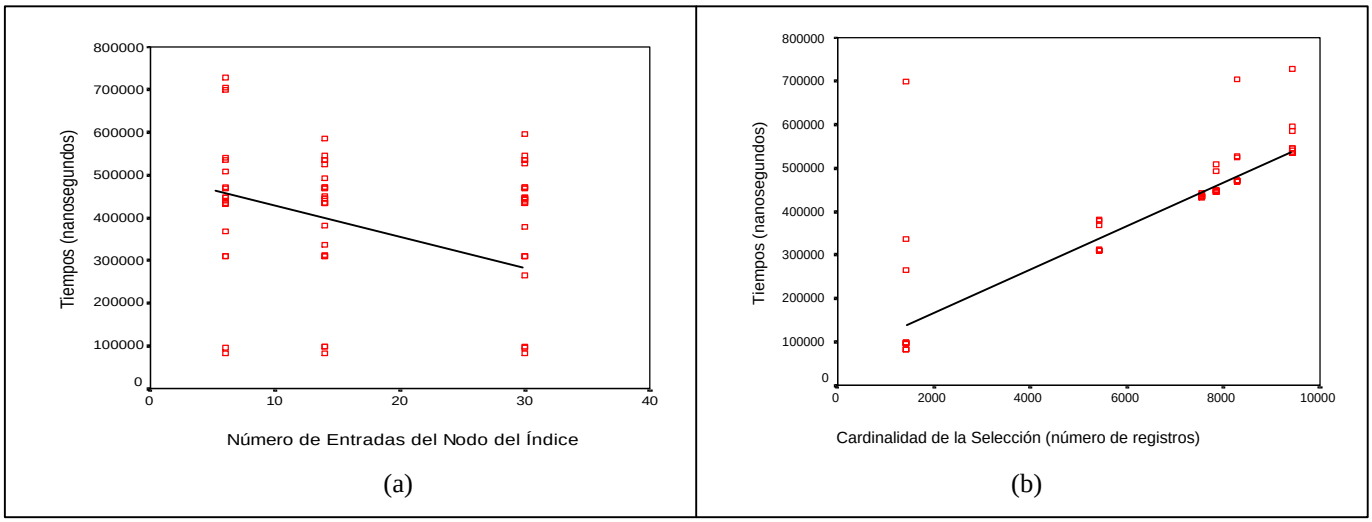

Figura 7: Relación entre el número de entradas del índice y el tiempo de ejecución (a), y entre la cardinalidad de la selección y el tiempo de ejecución (b).

5.3 Análisis de los Resultados y del Método

El trabajo desarrollado finaliza con el conjunto de ecuaciones de costo requeridas por el optimizador de consultas, el cual puede realizar su trabajo de forma conveniente, contando con fórmulas que calculan el costo de ejecutar cada tipo de operación posible, con un grado de exactitud razonable. De este modo, se tiene un método simple para obtener las expresiones necesarias, y que

sin caer en complejos cálculos para la estimación de los tiempos relacionados con el movimiento de datos dentro del sistema de memoria, es capaz de apoyar la labor del optimizador de buena forma.

Cabe mencionar en este punto que si bien los ajustes entregados no tienen un R^2 ajustado cercano al valor 1, el método es fiable. Los resultados entregados se aplican a ejemplos generalizados de una búsqueda indexada, en los que se incluyen todos los casos posibles que puedan presentarse, pero a pesar de esto los resultados son bastante buenos. Esto quiere decir que de ser más específico con los casos, dividiéndolos por tipo de operador o por una estimación más precisa de la cardinalidad de la operación, por ejemplo, las fórmulas alcanzarían una bondad mayor. Sin embargo, no hay que olvidar que el optimizador no requiere, necesariamente, una exactitud total en la estimación de los costos, sino que una estimación que le permita descartar planes malos y escoger uno bueno, con lo que el método y sus resultados permiten apoyar esta labor sin problemas. Así, el estudio profundiza un tema que ha sido muy poco trabajado, y que en comparación con las propuestas señaladas en la sección dos, resulta simple de entender y aplicar.

La incorporación del método a un software comercial debe hacerse sobre la base de expresiones de costos con coeficientes generales, las que deberán ser posteriormente afinadas a partir de los tiempos de ejecución reales registrados por el sistema, en su bitácora, los que reflejan el rendimiento real de la BD-MP en explotación.

6 Trabajo Futuro

El método es susceptible de ser aplicado a sistemas de BD-MP más avanzados. La extensión natural es su aplicación a ambientes paralelos, en donde el método abarca los costos de un nodo, existiendo la necesidad de ampliar su contexto para abarcar el paralelismo de la ejecución entre todos los nodos. La extensión debe considerar los costos de los bloqueos y desbloqueos sobre los datos (y cuya influencia no es despreciable [Cha01]), de las barreras de sincronización, y la adecuada identificación del costo de ejecutar una operación de modo paralelo. Esto último debido a que, por ejemplo, la ejecución paralela de un recorrido lineal sobre un archivo particionado horizontalmente, tiene un tiempo de ejecución equivalente al del procesador que consume más tiempo con la porción de datos asignada. La integración de todos los costos parciales tendrán que basarse en métricas propias de sistemas paralelos, como el trabajo y la profundidad [Blel96], y/o readecuar propuestas anteriores en las que el contexto usado no era, necesariamente, el de una BD-MP [Gang96].

Una visión similar se puede tener para el modelo de costos de una BD-MP distribuida; sin embargo, ahora es la mensajería la que tiene asociada el mayor costo, de modo que los tiempos de envío, recepción y sincronización de los mensajes se convierten en el factor principal del modelamiento. A primera vista, el método podría dejar de ser válido tal como está, pues la presencia de un número amplio de mensajes hace que los tiempos locales de cada nodo sean insignificantes para el sistema global. No obstante, si el grado de distribución de los datos no es muy alto, el uso de los mensajes es bajo, entonces los costos locales pueden adquirir un papel más influyente, por lo que debiera ser el contexto a considerar para extender el modelo en esa área.

7 Conclusiones

Se puede ver que el método aplicado es adecuado, pues es capaz de entregar buenas estimaciones para apoyar a un optimizador de consultas. Esto, porque los principales aspectos que influyen en el costo de las operaciones, y la estimación que se realiza a partir de una ecuación que los integra, entrega un costo de ejecución aceptable, según lo respalda la base estadística. Es

importante notar, eso sí, que el análisis de sensibilidad, por si solo, no es suficiente para determinar la forma exacta de la ecuación; esta primera actividad del proceso permite tener una aproximación a cada uno de los factores que influyen en la ejecución de una operación, pero es en el contexto de la ejecución misma cuando se conoce la real influencia de cada parámetro. Luego, la combinación de un estudio de sensibilidad de expresiones analíticas con una validación experimental es la forma adecuada para modelar funciones de costo con una bondad aceptable, con términos simples y fáciles de obtener.

Bibliografía

[Aila99] **DBMSs on a Modern Processor: Where does Time go?.** A. Ailamaki, D. DeWitt, M. Hill y D. Wood. Proceedings of the 25th VLDB Conference, Edimburgo, Escocia, Septiembre de 1999. Págs. 266-277.

[Blel96] **Programming Parallel Algorithms**. G. E. Blelloch. Communications of the ACM, Vol. 39, Nro. 3, Marzo de 1996. Págs. 85-97.

[Bonc99] **Database Architecture Optimized for the New Bottleneck: Memory Access.** P. Boncz, S. Manegold y M. Kersten. Proceedings of the 25th VLDB Conference, Edimburgo, Escocia, Septiembre de 1999. Págs. 54-65.

[Cha01] **Cache-Conscious Concurrency Control of Main-Memory Indexes on Shared-Memory Multiprocessor Systems.** S. Cha, S. Hwang, K. Kim y K. Kwon. Proceedings of the 27th VLDB Conference, Rome, Italy, September 2001. Pages 181-190.

[Gang96] **Efficient and Accurate Cost Models for Parallel Query Optimization.** S. Ganguly, A. Goel y A. Silberschatz. Proceedings of the ACM Symposium on Principles on Database Systems, 1996. Págs. 172-181.

[List96] **Modelling Costs for a MM-DBMS.** S. Listergarten y M. A. Neimat. Proceedings of the International Workshop on Real-Time Databases (RTDB): Issues and Applications, Newport Beach, California, Marzo 1996. Págs. 72-78.

[Lu00] **T-Tree or B-Tree: Main Memory Database Index Structure Revisited.** H. Lu, Y. Y. NG y Z. Tian. Online Proceedings of the Australian International Conference, Enero/Febrero del 2000. Págs. 65-73.

[Mane00] **What Happens during a Join? Dissecting CPU and Memory Optimization Effects.** S. Manegold, P. Boncz y M. Kersten. Proceedings of the 26th VLDB Conference, El Cairo, Egipto, Septiembre del 2000. Págs. 339-350.

[Mart01] **Modelo de Costos para un Sistema de Bases de Datos en Memoria Principal.** José Luis Martí. Tesis para optar al Grado de Magister en Ingeniería Informática, Departamento de Informática, Universidad Técnica Federico Santa María, Valparaíso, Chile.

[Nete90] **Applied Linear Statistical Models**. J. Neter. Richard D. Irwin, Inc. Tercera Edición, 1990.

[Perf00] **Linux x86 Performance Monitoring Counters Driver.** http://www.csd.uu.se/?mikpe/linux/perfctr/.

[Rao00] **Making B+ Cache Conscious in Main Memory.** J. Rao y K. Ross. Proceedings of the ACM Sigmod International Conference on Management of Data, Dallas, Texas, Junio del 2000. Págs. 475-486.

[Shat94] **Cache Conscious Algorithms for Relational Query Processing**. A. Shatdal, C. Kant y J. Naughton. Proceedings of the 20th VLDB Conference, Santiago, Chile, Septiembre de 1994. Pág. 522-533.

[Swam89] **Optimization of Large Join Queries: combining Heuristic and Combinatorial Techniques.** A. Swami. Proceedings of the ACM SIGMOD International Conference on Management of Data, Portland, Oregon, Mayo/Junio de 1989. Págs. 367-379.

[Whan90] **Query Optimization in a Memory Resident Domain Relational Calculus System**. K.Y. Whang y R. Krishnamurthy. ACM Transactions on Database Systems, Vol. 15, Nro. 1, Marzo de 1990. Págs. 67-95.