UNIVERSIDAD

UNIVERSIDAD AUTONOMA METROPOLITANA

DIVISIÓN DE CIENCIAS BASICAS E INGENIERIA

MAESTRIA EN CIENCIAS DE LA COMPUTACIÓN

PROPUESTA DE PROYECTO TERMINAL I

"Métodos Heurísticos para Solucionar Problemas con el Desempeño de Bases de Datos Relacionales"

ALUMNO
LI. Lazarini Castañeda Ulises

ASESORES

MC. BLANCA SILVA LOPEZ
DR. JAVIER RAMÍREZ RODRÍGUEZ

1. Introducción

Optimizar una Base de Datos (BD) es una de las actividades más complejas del Administrador de la Base de Datos (ABD), es necesario que las aplicaciones y procesos se ejecuten de la manera más eficiente. “Más eficiente” generalmente significa un rendimiento de procesamiento alto para dar mejores tiempos de respuesta, este es el principal punto de estudio de este trabajo.

Generalmente las BD que se construyen en muchas empresas lo hacen sin considerar los elementos que permiten tener un buen rendimiento. Por lo que el ABD tiene que realizar una serie de actividades en distintos niveles de afinación considerando los siguientes elementos

- Diseño de la BD.
- Estructuras de datos.
- Algoritmos involucrados en la explotación de la información.
- Parámetros del manejador de la BD.
- Configuración del sistema operativo (SO).
- Hardware
- Logística de la Aplicación – Usuario final

Los ABD que requieren solucionar problemas relacionados con las mismas, deberán de poseer un amplio conocimiento de diseño de BD, de métodos de optimización, de paradigmas de programación para poder modelar de manera consistente la aplicación para que cuando el ABD se encuentre inmerso en un problema de bajo desempeño, él pueda tener los fundamentos y herramientas necesarias para poder solucionar de manera correcta los problemas.

El afinar no es tarea fácil porque el administrador necesita considerar un gran número de escenarios y ambientes que existen en el mundo real, no existe ningún procedimiento que indique cómo realizar la afinación de un solo paso y en una sola sesión de trabajo. Muchos investigadores, académicos e industriales han intentado afinar las BD y el procesamiento de las consultas, generalmente sin un manejo formal de los procesos y las estructuras. Estos esfuerzos se han derrumbado generalmente por la falta de los elementos mencionados. Cualquier persona que realice o esté involucrada en la afinación, deberá considerar los diferentes niveles de análisis y ajuste para mejorar el rendimiento de la BD.

En la creación de una BD, su monitoreo constante puede ser controlado, puede ser flexible o abierto, estructurada, pero nunca caótica y sin método.

La idea de este estudio es novedosa ya que ayudará a seguir teorías y/o mejorarlas para resolver problemas actuales. Este trabajo cumplirá dos propósitos fundamentales:

> *a*) Diseñar procedimientos eficientes de búsqueda para optimizar los tiempos de respuesta de las consultas hacia la BD, una posible herramienta puede ser la generación de un árbol ponderado en base a un árbol de pesos que generaremos.

b) proponer métodos exactos o de búsqueda para determinar el mejor esquema de balanceo de cargas en varios discos: $\Sigma_{i,\ldots,j}$

"donde el árbol de búsquedas se hará dependiendo del número de discos en donde se monte una BD".

Se desarrollará un procedimiento de optimización claro y aplicable a cualquier sistema de BDR, con el fin de mantener un buen desempeño. Para lograr esto nos basaremos en modelos matemáticos de optimización de estructuras y optimización de código.

2. Antecedentes

Por años se han construido sistemas sin fundamentos formales (fundamentos matemáticos) que han dado como resultado el desarrollo de sistemas con un desempeño pobre y que causa la insatisfacción del cliente y la frustración del equipo de desarrollo y de los ABD.

Desde los años 1970 en los tratados de "Manejo de Bases de Datos Relacionales" [1], pocas han sido las empresas e industriales que le han inyectado un formalismo a sus sistemas. Utilizar técnicas eficientes de programación, algoritmos que hagan búsquedas eficaces con un grado de complejidad cuando más polinomial o logarítmica, es decir con menos ciclos iterativos y con menos accesos a disco, como se recomienda en diferentes tratados de Knuth [2], si no se aplica una formalidad, es muy probable que se tengan malos tiempos de respuesta.

Los problemas de bajo desempeño se presentan principalmente por las malas técnicas y malas costumbres al modelar aplicaciones, sin conocimientos sólidos y sustentables en formalismos ya conocidos (algoritmos y modelos matemáticos, teoría de conjuntos, métodos heurísticos, programación dinámica, teoría de grafos), o bien por desarrollar sin un objetivo claro y alcanzable, sin manejo de excepciones y sin manejo de las nuevas expectativas que sufre la empresa y el sistema.

2.1. Estado del Arte.

En el ambiente de las BD se han tenido grandes evoluciones, desde las BD jerárquicas, reticulares pasando por las relaciones y últimamente las BD orientadas a objetos predominando en un gran porcentaje de utilización las BDR dentro de las empresas, es aquí en donde más estudios e investigaciones se han hecho.

Las investigaciones actuales con relación a la optimización dentro de las BDR son las siguientes:

Un estudio exhaustivo estudio del álgebra relacional para modelar de manera matemática todas aquellas consultas hechas hacia la BD utilizando una metodología de manera robusta como lo menciona Paredaens [3]. Por su parte Kossmann [4] propone algoritmos de programación dinámica iteractiva para la optimización de las consultas hechas a BD comerciales.

El manejo del E/S de una BD en producción es de tomar en cuenta, ya que esto puede afectar los niveles de aceptación con referencia a los tiempos de respuesta, en este punto nos vamos a apoyar del estudio Hsu, WW; Smith, AJ; Young, HC [5] que hicieron pruebas a diferentes BD.

La gran mayoría de los estudios que se han hecho sobre BDR son enfocándose a la optimización de las consultas hechas hacia la BD, muy pocos son los estudios que se tienen sobre el comportamiento global del la BD que involucra no solo la optimización de las consultas sino también la optimización de:

Memoria.

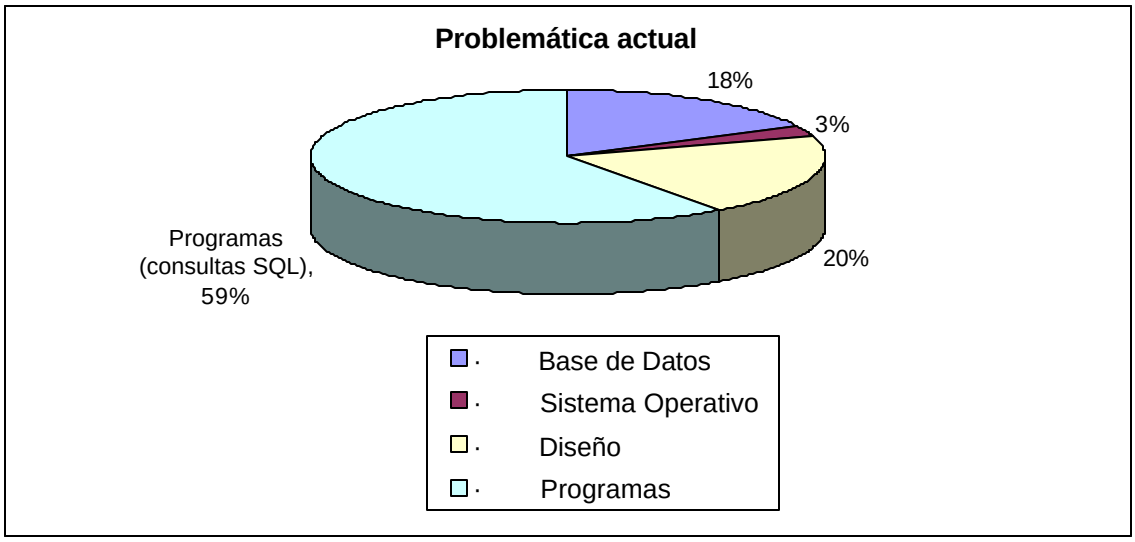
Usos de CPU's.

E/S de acceso a los datos.

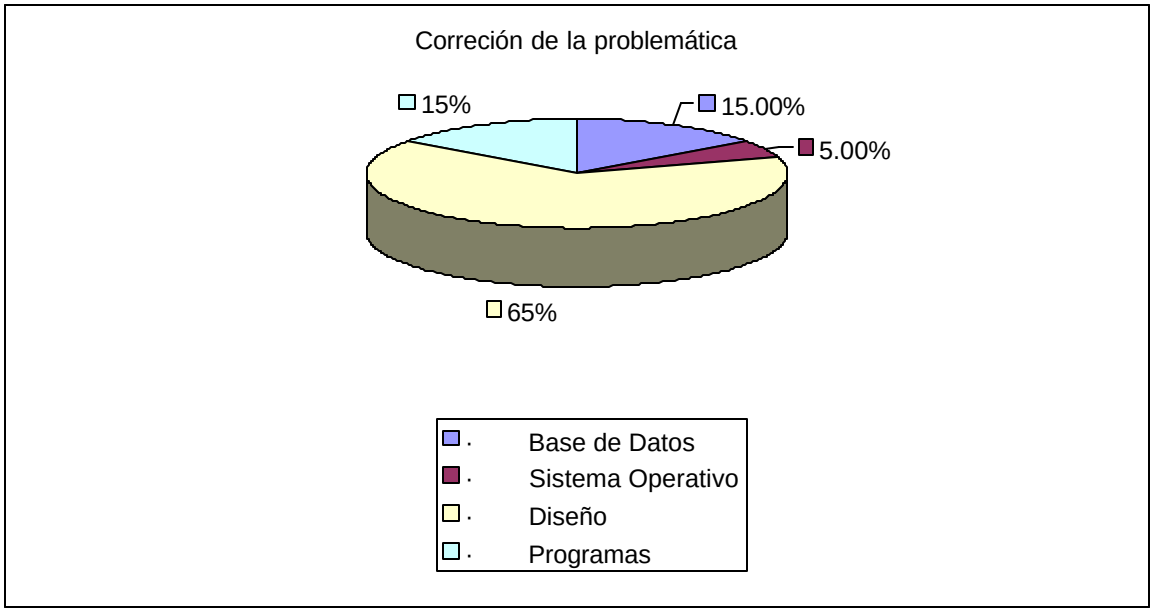
3. Justificación

Debido a que se encuentra poca información en el manejo de la optimización de una BDR, se propone una investigación formal de los componentes que engloban el funcionamiento de una BD.

Una de las principales razones para llevar acabo una metodología para un diseño formal y proporcionar mecanismos para optimizar una BD, se debe a que un gran número de empresas tienen problemas relacionados con sus sistemas de BD. Como resultado de estudios de trabajo Oracle Corporation nos muestra información de empresas que tienen problemas de desempeño [7].



El corregir estos problemas relacionados con el desempeño se distribuye de la siguiente manera:



Con estas gráficas podemos observar y comparar lo siguiente:

1. En la primera gráfica muestra que se debe de tener cuidado con los programas que se tienen y sus consultas SQL que se hacen hacia la BD, ya que la programación puede ser la causante de los problemas de optimización.
2. En la segunda gráfica es de notar que no necesariamente la programática es la culpable del mal desempeño de una BDR, debido a que se puede desperdiciar horas de trabajo tratando de optimizar las sentencias de control y las consultas SQL de un programa cuando el verdadero problema son sus algoritmos.

Implicaciones prácticas - Recompensas

Básicamente tendremos mejoras en los siguientes puntos:

- Beneficios relacionados al tiempo: Desarrollo de aplicaciones con uso de BDR eficientes con un mínimo de esfuerzo
- Eliminación de tiempos oscuros. Esto es, el no saber que pasa con la BD cuando esta tiene problemas de desempeño, y estar buscando por diferentes lugares los posibles problemas.

4. Objetivos

4.1 *Objetivo General*

Establecer algoritmos heurísticos que nos permitan estandarizar mecanismos de:

- Diseñar la distribución de los archivos de una BDR.
- Optimizar las consultas de una BDR.

4.2 Objetivos Específicos

- Diseñar diferentes métodos heurísticos para mejorar la distribución de los archivos físicos de una BDR dependiendo la naturaleza de la misma (esto es si es una BD de tipo: OTLP, DSS, o Híbrida).
- Generar un árbol de pesos en una tabla relacional para asignar un valor a las entidades dependiendo de su naturaleza y los registros que estas tengan (esto es, si es una tabla catalogo, histórica ó transaccional).

Para lograr estos resultado tomaremos principios matemáticos como son:

- El álgebra y el cálculo relacional.
- Evaluación de métodos para la optimización de las consultas como son:
    - Métodos heurísticos que permiten ordenar las operaciones de la consulta en un árbol.

5. Metodología de desarrollo

FASE 1.

- Revisión del estado del arte.
- Análisis y selección de modelos de optimización que sirvan como herramientas para el diseño del árbol de distribución de los archivos de la BD, para evitar la contención contra los dispositivos

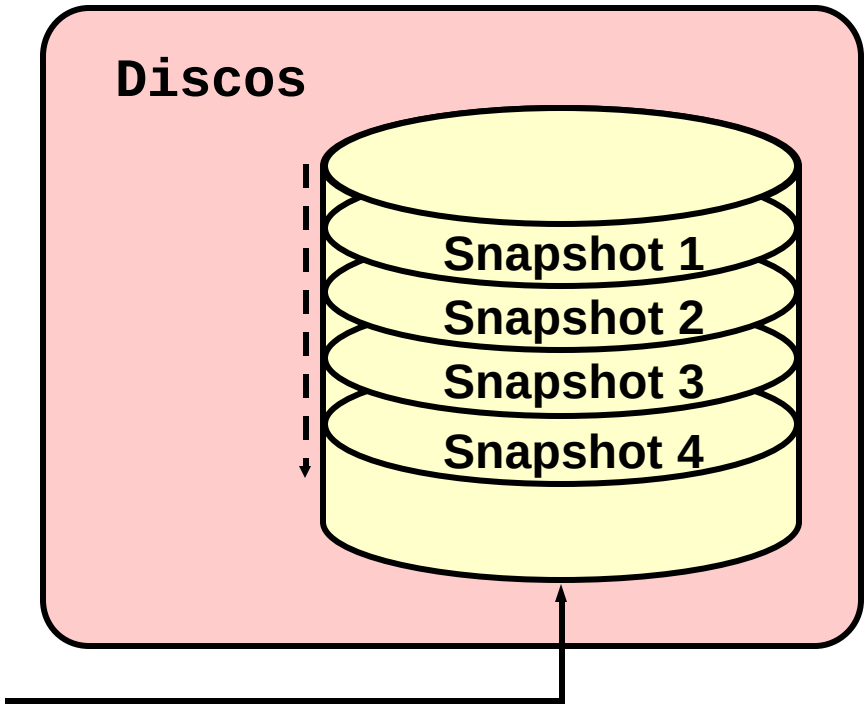
Trataremos de encontrar los mejores métodos de distribuciones de los archivos dependiendo del número de dispositivos físicos (discos duros) que se nos dé. Trataremos de manejar donde el total de discos sea mayor de 2.

$\Sigma \iota > 2$

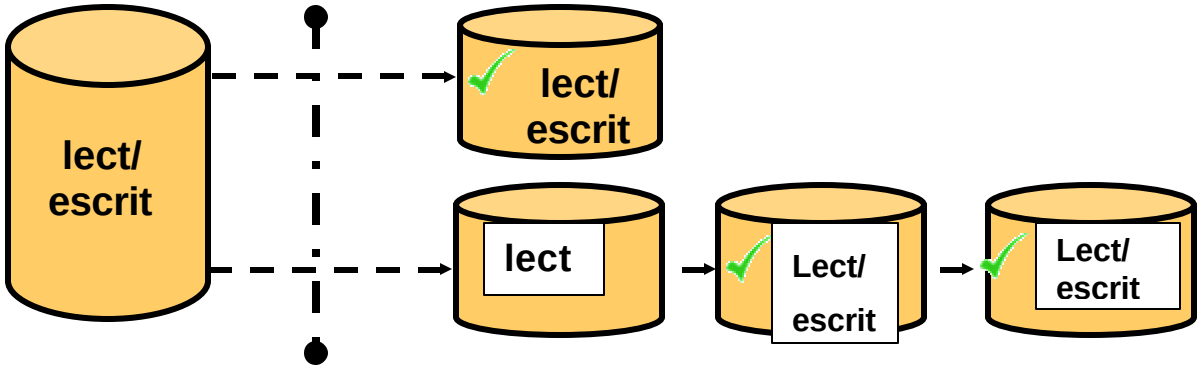
i es el número de discos.

Este proceso se trabajará a nivel físico BD y SO.

En esta etapa nos vamos apoyar de la máxima divide y vencerás, al momento de hacer la consulta a los dispositivos será del orden N/2, esto usa un tiempo lineal que nos puede ayudar para que el coste de la entrada y salida de lecturas a disco (E/S) se reduzca por lo menos a la mitad.



Volumen de discos



Distribución de los archivos para las consultas de los datos.

- Una vez establecidas las distribuciones físicas de los archivos de la BD sobre los discos, trabajaremos sobre las relaciones y funciones de los objetos discretos que ahí se encuentran en la BD para optimizar:

Las proyecciones de atributos de la forma $\pi_x(R) = \{x \mid \exists \text{ y tal que } (x,y) \in R(X,Y)\}$,

Las selección de atributos de la forma $\sigma_p(R) = \{x \mid x \in R \wedge P_{(x)} \text{ es verdadero}\}$,

El join de las relaciones R y S sobre los atributos de la forma

$R \underset{X=Y}{\infty} R = \{(z,x,w) \mid (z,x) \in R \wedge (y,w) \in S \wedge x=y\}$,

La unión de dos relaciones $R \cup S = \{t|t \in R \vee t \in S\}$

La intersección de dos relaciones $R \cap S = \{t|t \in R \wedge t \in S\}$ y

La diferencia entre dos relaciones $R - S = \{t|t \in R \wedge t \notin S\}$

Complejidad algorítmica

Hay una gran número de parámetros para medir la complejidad, para nuestro desarrollo tomaremos los siguientes:

tiempo de ejecución principalmente,

número de CPU ≥ 2,

número de dispositivos físicos (discos duros) ≥ 2

cantidad de memoria.

Nuestra talla o punto de medida será el número de registros que tenga un archivo en función de T segundos, de tal forma que nos enfocaremos a medir de la siente manera:

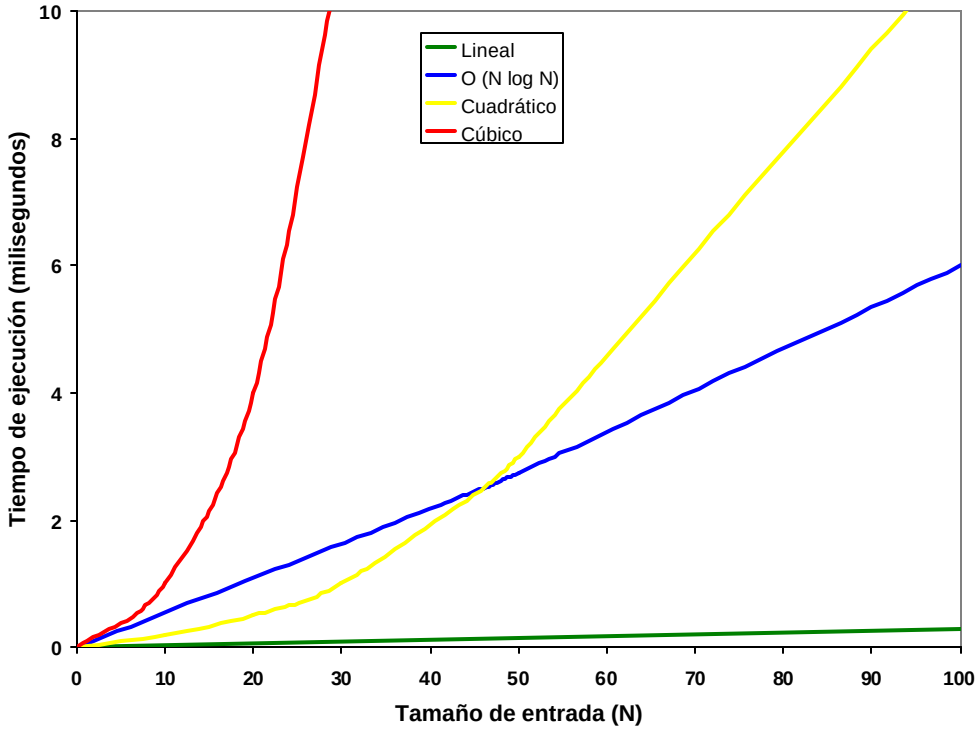
T(N) /* tiempo de ejecución */

T(N) = t1 + t2*N

Y nuestro coste será:

C(n) = Número de pasos de programa en función de n.

Trataremos de manejar una complejidad de orden lineal O(n) ya que es todo lo bueno que podemos esperar, el algoritmo también puede ser de orden logarítmico $O(\log^n)$ en el mejor de los casos, pero quedando satisfechos si la complejidad del algoritmo lo podemos hacer de manera polinomial o logarítmica de la forma $O(N \log^n)$, pero nunca que está sea una complejidad cuadrática o cúbica.



- Redacción de resultados.

FASE 2.

- Creación de un ambiente de pruebas.

  Construcción del programa que nos genere el árbol que describa el proceso de búsqueda y a partir de aquí encontrar la mejor manera de realizar la búsqueda en base a las heurísticas diseñadas.

- Realización de pruebas de rendimiento con datos de una situación real.
- Evaluación de resultados obtenidos en el proceso del análisis del desempeño.
- Redacción de conclusiones finales.

Temario

Capitulo 1

Panorámica matemática /* Repaso de los fundamentos matemáticos que se requieren y que se van a utilizar.

Capitulo 2

Panorámica a las BDR.

- Conjuntos
- Álgebra relacional.
- Particionamiento.

Capitulo 3

Desarrollo de modelos /* Desarrollo del método de optimización.

Capitulo 4

Implementación del modelo.

- Instalación de un RDBMS.
- Programática del modelo.

Capitulo 5

Pruebas.

Capitulo 6

Conclusiones.

6. Calendarización

Proyecto de investigación en Computación I

ACTIVIDAD	SEMANA										
	1	2	3	4	5	6	7	8	9	10	11
Revisión del estado del arte											
Análisis de modelos para consultas a la BD											
Análisis de modelos para el manejo de los archivos ocupados por la BD a nivel SO											
Análisis de modelos para el manejo de las consultas hechas a la BD											

Selección de modelos											
Redacción de reporte											

Proyecto de investigación en Computación II

ACTIVIDAD	SEMANA										
	1	2	3	4	5	6	7	8	9	10	11
Creación de un ambiente de pruebas											
Construcción de un programa en base a modelos tudiados.											
Implementación del programa sobre la BD											
Realización de pruebas											
Evaluación de resultados obtenidos en el proceso del análisis del desempeño.											
Redacción de conclusiones finales.											

7. Recursos Necesarios

Hardware

Pentium IV 5

1GB memoria RAM (mínimo).

2 Discos duros 36 Gb (mínimo).

2 CPU

Software

Sistema Operativo de dominio libre (linux) [8]

Un sistema manejador de de bases de datos relacionales. (RDBMS) [9]

7.2 Recursos disponibles

Hardware

Equipos de laboratorio de maestría.

Software

8. Bibliografía

[1] C.J. Date,
Introducción a los sistemas de bases de datos
Volumen 1
Addison wesley

[2] Donald E. Knuth.
The art of computer programing
Fundamental algorithms
Volume I Third Edition
Addison Wesley

[3] Paredaens, j; vangucht,
Converting nested algebra expressions into flat algebra expressions
Acm transactions on database systems, 17 (1): 65-93 mar 1992

[4] Kossmann, D; Stocker, K
Iterative dynamic programming: A new class of query optimization algorithms
ACM TRANSACTIONS ON DATABASE SYSTEMS, 25 (1): 43-82 MAR 2000

[5] Hsu, WW; Smith, AJ; Young, HC
I/O reference behavior of production database workloads and the TPC benchmarks - An analysis at the logical level
ACM TRANSACTIONS ON DATABASE SYSTEMS, 26 (1): 96-143 MAR 2001

[6] Oracle Corpotation
Carry Millsap
Oracle Performance. 2003

[7] Modern Heuristic Techniques for Combinatorial Problems,
Reeves, C. (Editor), McGraw Hill,
London,1995.

[8] Mark Brunelli, Monica Kumar, Oracle senior manager of Linux product marketing.
LinuxWorld,
Database 10g update to feature more automation
31 Mar 2005 | SearchEnterpriseLinux.com

[9] Jack Loftus
Oracle retains database pole position
LinuxWorld, 15 Mar 2005 | SearchOracle.com