PROPUESTA Y EVALUACIÓN DE UN MODELO DE RECONFIGURACIÓN DINÁMICA EN UN SUBSISTEMA DE ENTRADA/SALIDA REDUNDANTE PARA UN SISTEMA DE ARCHIVOS DISTRIBUIDO Y PARALELO.

JUAN PABLO GARCÍA OJEDA

Universidad Austral de Chile, Ingeniero Civil en Informática, jgarcia@inf.uach.cl, Casilla 567 - Valdivia.

RAIMUNDO VEGA VEGA

Universidad Austral de Chile, Doctor en Informática, rvega@uach.cl, Casilla 567 - Valdivia.

Resumen **- El presente trabajo estudia el problema del almacenamiento masivo de información centrándose en la disponibilidad de los datos. Para esto, se toma como base un simulador de un sistema de archivos distribuido y paralelo con tolerancia a fallos al cual se le añadió una nueva funcionalidad conocida como reconfiguración dinámica, es decir la característica que permite al sistema poder agregar más nodos de almacenamiento sin necesidad de detener la normal entrega de servicios. Por último se generan pruebas que permiten analizar los resultados y compararlos con otros estudios realizados anteriormente sobre el mismo sistema bajo condiciones que no incluyen reconfiguración.**

Abstract **- The current work studies the problem of the massive storage of information being centered in data availability . For this purpose, it takes as base a fault tolerance distributed and parallel system simulator which has been added a new functionality known as dynamic reconfiguration, it means the attribute that allows to the system to be able to add more storage nodes without need to stop normal services delivery. Finally, tests are generated that allow to analyze the results and to compare them with other studies carried out previously on the same system under conditions that don't include reconfiguration.**

Palabras clave **– sistemas distribuidos, paralelismo, tolerancia de fallos, sistemas de archivos, redundancia, reconfiguración.**

1. INTRODUCCIÓN

La informática se define como: “El conjunto de conocimientos científicos y técnicos que hacen posible el tratamiento automático de la información por medio de ordenadores” [1]. Con el pasar de los años, esta ciencia ha aprendido que el valor de la información obtenida depende en gran manera de la cantidad y calidad de los datos que la subyacen. Existen ocasiones en que la cantidad de datos es tan abrumadora que el acceso a estos no se puede obtener en un lapso razonable de tiempo por lo que la información que con ella se obtiene se considera obsoleta.

Con el transcurrir del tiempo, la velocidad de procesamiento superó considerablemente a la velocidad de entrada y salida a los datos proporcionada por los sistemas de archivos, lo que trajo consigo la llamada “Crisis de la E/S” [2], la cual se agrava aún mas en entornos paralelos donde el acceso concurrente a los archivos, es decir, dos o más procesos acceden a este, es muy frecuente [3,4] y tanto los sistema de archivos tradicionales como los distribuidos no están optimizados para este tipo de acceso [5]. De esta problemática aparecieron los Sistemas de Archivos Distribuidos y Paralelos, los cuales combinan soluciones tales como: paralelismo en el sistema de entrada y salida, interfaces paralelas y caché en el sistema de entrada y salida.

Estos últimos sistemas han sido hasta el momento la solución a la crisis de la E/S. Sin embargo aún queda por incorporar el tema de la disponibilidad de datos, cuestión que se torna crítica en los sistemas de archivos distribuidos y paralelos ya que, a medida que se incrementa el paralelismo y la cantidad de nodos de entrada/salida, también aumenta la probabilidad de fallo del sistema [6]. En vista del poco estudio que había en este campo, es que en [6] se propuso un modelo de redundancia de datos que además fue implementado en un simulador especialmente diseñado para el efecto [6][8].

A partir del estudio que ahí se realizó, este trabajo estudiará un método que permita incorporar en forma dinámica nuevos dispositivos sin detener la normal entrega de servicios del sistema de entrada/salida.

2. DESCRIPCIÓN DEL PROBLEMA

Un sistema de archivos distribuido y paralelo, es aquel en el cual los datos se encuentran repartidos en múltiples nodos dentro de una red de computadores y estos son accedidos de manera simultánea para mejorar el rendimiento. Con el objetivo de estudiar el desempeño de estos sistemas, se ha construido un sistema simulador de este tipo de arquitecturas que además permite medir el ancho de banda y los tiempos de lectura y escritura para cargas de tipo científica y transaccionales (OLPT). El sistema es configurable mediante el ingreso de distintos parámetros que permiten identificar el tipo de distribución que tienen los elementos de este, tales como: el número de nodos en la red, tamaño de las unidades de reparto, nivel de RAID, etc.

Este trabajo se centra en la formulación de un modelo que incorpore una nueva funcionalidad al sistema y este pueda así incorporar nuevos nodos a la red de manera que la información se redistribuya de forma balanceada y sin que el sistema este fuera de servicio.

3. MODELO DE UN SISTEMA DE ARCHIVOS DISTRIBUIDO Y PARALELO CON REDUNDANCIA DE DATOS.

Un sistema de archivos distribuido y paralelo con redundancia de datos, es aquel en el cual la información que se encuentra distribuida dentro de los nodos incorpora información redundante, que le proporciona al sistema características de tolerancia a fallos(fig.1)

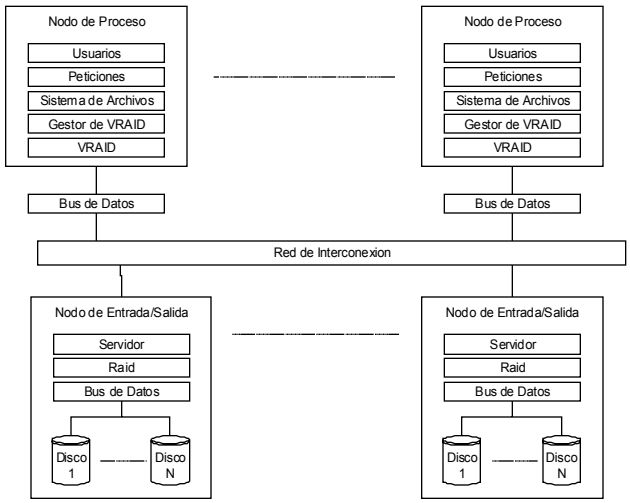

Figura 1. Esquema de un sistema de archivos distribuido y paralelo con redundancia de datos.

Es decir, el sistema es capaz de permitir el fallo de uno o más de los nodos dependiendo de la cantidad de información redundante que se desee incorporar sin perder la disponibilidad de la información. Una forma de alcanzar redundancia de datos en un sistema de archivo distribuido y paralelo es utilizando un esquema similar al empleado por las Matrices de Discos Redundantes, que se puede representar por una matriz de m filas y n columnas (Figura 2).

En este caso las n columnas representan el número de nodos de almacenamiento existentes dentro de la red y m el número de bloques de almacenamiento dentro de cada uno de los nodos. Cada fila de la matriz es conocida como Franja de Paridad y cada elemento de la matriz es un bloque dentro del sistema de archivos distribuido. En cada una de las franjas de paridad existe un bloque representado por una letra P que almacena la información redundante calculada en base al resto de los bloques que comparten la misma franja de paridad. De esta manera existen m*(n-1) bloques de almacenamiento.

	N0	N1	N2	N3
F0	0	1	2	P
F1	3	4	P	5
F2	6	P	7	8
F3	P	9	10	11
F4	12	13	14	P
F5	15	16	P	17
F6	18	P	19	20
F7	P	21	22	23
F8	24	25	26	P
F9	27	28	P	29

Figura 2. Matriz de un sistema de archivos distribuido y paralelo con tolerancia de fallos.

La arquitectura del modelo de sistema de archivos distribuido y paralelo con redundancia de datos con la cual se ha trabajado consta de las siguientes capas (fig3):

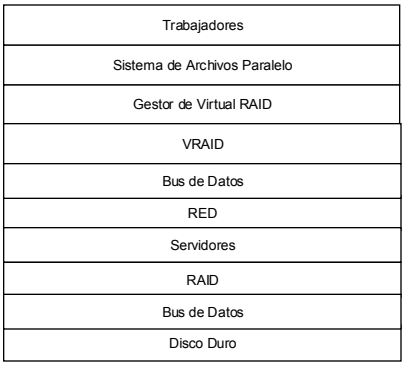

Figura 3. Bloques de un sistema de archivos distribuido y paralelo con redundancia de datos.

Trabajadores: Este módulo se encarga de simular la generación de todas las solicitudes de entrada/salida al sistema de archivos.

Sistema de Archivos Paralelo: Este módulo representa un sistema de archivos que lanza las peticiones hacia los dispositivos de las capas inferiores [7]. Para esto simula los procesos en que deben dividirse las solicitudes de entrada/salida con el fin de ser enviadas a los nodos respectivos.

Red: Modelo de una red de trabajo que interconecta nodos con CPUs, puede configurarse de diferentes formas, lo cual determina sus prestaciones. Interconecta y crea una cantidad de nodos que componen el sistema [7].

Servidores: Implementa un servidor de entrada/salida el cual esta encargado de monitorear los tiempos de transferencia por la red.

Raid: Modela un dispositivo Raid de niveles 0, 4 o 5, los que abarcan el conjunto de todos los discos de un nodo. Distribuye la operación solicitada en operaciones a los discos involucrados. Este módulo despacha las operaciones que se le solicitan en el orden en que llegan, de forma secuencial, pero explotando la concurrencia entre los discos [7].

Bus de Datos: Modelo de un bus de entrada/salida, puede configurarse de diversas formas que determinan sus prestaciones. Un bus transmite información desde un nodo a un disco o viceversa. Pueden existir varios buses por nodo [7].

Disco Duro: Modelo de un disco duro. Los discos pueden ser de distintos tipos lo cual determina sus prestaciones. Un disco ira conectado a un bus. Puede haber varios discos por bus [7].

Gestor de Virtual Raid: Un dispositivo gestionador de Raid que agrupa componentes de VRAID que permite al sistema redundante aumentar su disponibilidad de datos [7].

VRAID: Modela un dispositivo virtual Raid de niveles 0, 4 o 5. Este módulo despacha las operaciones que le solicitan los usuarios en el orden en que llegan, de forma secuencial, pero explotando la concurrencia entre los nodos[7].

3.1. Modelo del problema.

El problema se puede modelar como un arreglo de m filas y n columnas (Figura 4) en el cual las columnas representan los nodos del VRAID y las filas representan las franjas de paridad dentro de cada uno de nodos. Cada elemento de la matriz identifica un bloque con información, si el elemento esta representado por una P significa que ese bloque es utilizado para almacenar la información de paridad de la franja respectiva.

	N0	N1	N2	N3
F0	0	1	2	P
F1	3	4	P	5
F2	6	P	7	8
F3	P	9	10	11
F4	12	13	14	P
F5	15	16	P	17
F6	18	P	19	20
F7	P	21	22	23
F8	24	25	26	P
F9	27	28	P	29

Figura 4. Matriz de un sistema de archivos distribuido y paralelo con tolerancia de fallos.

El problema a resolver consiste en otorgarle al sistema la capacidad de incorporar nuevos nodos (Fig. 5), es decir agregar mas columnas a la matriz de manera que se reordenen los bloques dentro de la nueva matriz, cuidando que los bloques que almacenan información de paridad sean recalculados y no redistribuidos. Este proceso es el que se conoce como reconfiguración (Fig. 6).

	N0	N1	N2	N3	N4	N5
F0	0	1	2	P	–	–
F1	3	4	P	5	–	–
F2	6	P	7	8	–	–
F3	P	9	10	11	–	–
F4	12	13	14	P	–	–
F5	15	16	P	17	–	–

Figura 5. Matriz de un sistema de archivos distribuido y paralelo con tolerancia de fallos antes del proceso de reconfiguración dinámica.

	N0	N1	N2	N3	N4	N5
F0	0	1	2	3	4	P
F1	5	6	7	8	P	9
F2	10	11	12	P	13	14
F3	15	16	P	17	18	19
F4	20	P	21	22	23	24
F5	P	25	26	27	28	29

Figura 6. Matriz de un sistema de archivos distribuido y paralelo con tolerancia de fallos después del proceso de reconfiguración dinámica.

Con la incorporación de la reconfiguración dinámica el sistema queda con el siguiente diagrama de estados posibles (fig. 7). Donde:

- **Estado Normal**: Indica que todos los servicios del sistema se encuentran en correcto funcionamiento. Esto quiere decir que no existe ningún nodo que se encuentre con bloques de datos no disponibles debido a un fallo de los discos o de los mismos nodos.

- **Estado Degradado**: Indica que el sistema esta presentando un fallo en algún nodo o disco dentro de un nodo que impide el normal funcionamiento del VRAID. Sin embargo, esto no significa que el sistema no pueda entregar sus servicios, sino que estos tomarán un mayor tiempo debido al necesario cálculo de los datos que no se encuentran explícitos sino que deben ser reconstituidos con los códigos correctores de errores.

- **Estado Reconstrucción**: Este estado indica que existe un nodo o un disco en algún nodo que no permite la normal entrega de los servicios. En este estado el sistema se encuentra con la unidad que provoco el fallo reparada, pero sin los datos que debe contener. En consecuencia el sistema trabaja con un procesos reconstructor que genera y

almacena los datos en la unidad reparada en paralelo a otras solicitudes que le sean hechas que puedes ser respondidas en modalidad degradada o normal según sea el caso

- **Estado Fuera de Servicio**: Este estado indica que el sistema se encuentra fuera de servicio. La ocurrencia de este estado la determina el fallo de más de un nodo o disco dentro un nodo..

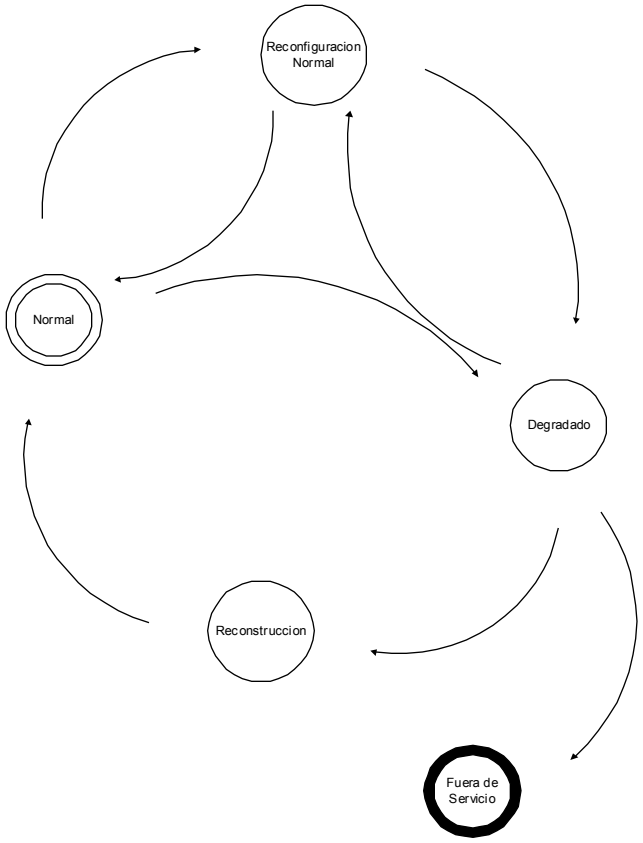

Figura 7. Estados de un sistema de archivos distribuido y paralelo con tolerancia a fallos y reconfiguración dinámica.

Este diagrama muestra en la parte superior el nuevo estado que tendrá el sistema denominado "**Reconfiguración Normal**" cuyo propósito es indicar a cada uno de los componentes que el sistema se esta reconfigurando con la finalidad de incorporar los nuevos nodos. A su vez, este estado de reconfiguración podría estar trabajando tanto en modo normal como en modo degradado (denominandose "**Reconfiguración degradada**"), sin embargo en esta investigación solo se ha trabajado con la reconfiguración en modo Normal.

En base a lo mostrado, se han planteado dos nuevos modelos que permitan implementar la solución al problema.

3.2. Primer Modelo de Reconfiguración Dinámica.

Este primer modelo incorpora una nueva capa de software al sistema de archivos distribuido y paralelo con redundancia de datos (Figura 8).

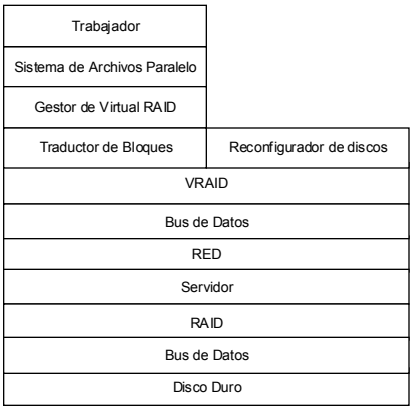

Figura 8. Bloques de un primer modelo de sistema de archivos distribuido y paralelo con tolerancia a fallos y reconfiguración dinámica

Esta nueva capa contiene 2 procesos que trabajan en forma concurrente, el primero de ellos denominado *Traductor de Bloques* se encarga de transformar las solicitudes de lectura/escritura que vienen dirigidas a nodos determinados desde los trabajadores a nuevas solicitudes dirigidas a la nueva configuración del raidmap. El segundo bloque llamado *Reconfigurador de discos*, es un proceso que tiene como misión realizar todas las operaciones necesarias para que el sistema incorpore los nuevos nodos y redistribuya la información. La descripción de los nuevos bloques es la siguiente:

Traductor de Bloques: Este proceso se encarga de redireccionar las solicitudes de entrada/salida provenientes desde los trabajadores. La motivación para crear este proceso viene de la problemática consistente en que el sistema de archivos no debe detener la entrega de los servicios mientras se encuentre en un estado de reconfiguración. A medida que el proceso de reconfiguración va avanzando por la matriz, esta queda particionada en tres zonas. La *primera zona* contiene las franjas de paridad que han sido reconfiguradas y cuya distribución abarca la totalidad de los nodos. La *segunda zona* esta conformada por las franjas de paridad que se encuentran bloqueadas puesto que el proceso reconfigurador esta trabajando sobre ellas. Finalmente la *tercera zona* esta compuesta por las franjas de paridad que aún no han sido reconfiguradas y que todavía se encuentran distribuidas como en un principio.

Debido a que la matriz se encuentra dividida en tres zonas (figura 9), el proceso traductor de direcciones debe ser capaz de discriminar a cual zona va dirigida la solicitud para de esta manera redireccionar la petición. Si la solicitud va dirigida a la zona reconfigurada el traductor de direcciones enviará esta solicitud a una nueva posición que será calculada en base a las expresiones de la ecuación (1).

$$j\_proy = \mathrm{int}\left(\frac{bloque}{n-1}\right)$$

$$i\_proy = (bloque\%(n-1)) + \mathrm{int}\left(\frac{bloque}{((n-2)*(j\_proy+1))+1+\left(n*\mathrm{int}\left(\frac{j\_proy}{n}\right)\right)}\right) \quad (1)$$

donde:

n: número de nodos del sistema.

bloque: número del bloque a trasladar [0..(m*(n-1))-1]

j_proy: número de la franja de paridad en la cual será trasladado el bloque [0..m-1].

i_proy: número del nodo en el cual será trasladado el bloque[0..n-1].

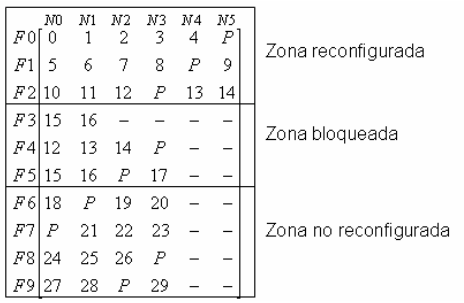

Figura 9. Zonas de reconfiguración.

Si la solicitud va dirigida a la zona bloqueada, el traductor de direcciones se encargará de postergar la solicitud hasta que dicha zona sea reconfigurada, para esto se utiliza un cerrojo de tal forma de proteger esa zona de datos.

Finalmente si la solicitud va dirigida a la zona no reconfigurada, el traductor de direcciones solo se encarga de retransmitir la petición tal cual fue realizada puesto que los bloques aún se encuentran en la misma posición.

Reconfigurador de Discos: Este módulo se encarga de redistribuir la información contenida en los nodos de forma tal que ahora incorpore a los nuevos nodos que se han agregado al sistema. Para esto hace un recorrido por la matriz, bloqueando las franjas de paridad que contienen a los bloques que en ese momento se están reconfigurando, con el fin de que ningún otro proceso lea o escriba en dichos bloques. Un parámetro importante para el rendimiento del sistema durante el funcionamiento de este módulo, es el número de franjas de paridad que deben bloquearse (figura 10) la cual esta determinada por el número de bloques que contendrá el grupo a reconfigurar.

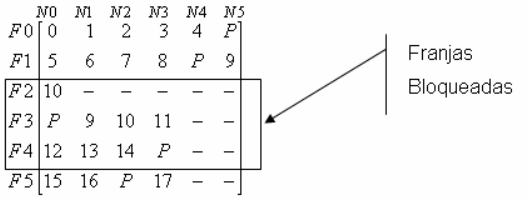

Figura 10. Bloqueo de franjas de paridad.

Independiente del modelo a implementar, existe un problema que debe solucionarse que tiene que ver con el bloqueo del acceso a los bloques. Para esto existen dos alternativas:

1) **Restringir el acceso a bloques específicos**: Esta opción significa que solo se impedirá el acceso los bloques que estén siendo relocalizados en ese momento, lo cual trae como ventaja minimizar la posibilidad que el bloque que se está solicitando desde los trabajadores se encuentre no disponible en ese instante. Sin embargo, esto provoca un aumento en el número de operaciones que debe realizarse y con esto el tiempo que tarda la reconfiguración. Por ejemplo, para mostrar esto de una manera mas sencilla se puede llevar esta situación al extremo, es decir, se puede pensar que el proceso reconfigurador solo irá tomando un bloque y lo llevará a su nueva posición, luego tomará el siguiente y así sucesivamente. De lo anterior se puede ver que cada vez que se relocalize un bloque será necesario calcular nuevamente la paridad tanto de la franja de origen como de la franja de destino, lo que implica un considerable aumento en el número de cálculos necesarios.

2) **Bloquear el acceso a franjas específicas**. Esta opción propone que el bloqueo de acceso abarque un número determinado de franjas completas, produciéndose así un aumento en la probabilidad de encontrar en un determinado momento bloqueado el bloque que se está solicitando desde los trabajadores ya sea para lectura o escritura. Por otro lado, la ventaja de este método se encuentra en que se reduce la cantidad de veces que debe calcularse la paridad a solo una vez por franja, lo que en comparación con la opción anterior en la cual era necesario calcular la paridad una vez por cada bloque de la franja lleva a un menor tiempo de procesamiento.

En base a lo explicado en los puntos anteriores, se decidió utilizar la segunda alternativa puesto que aunque se aumenta la probabilidad de que encontrar bloqueado un bloque solicitado por un trabajador, esta se puede seguir

considerando despreciable debido a que el número total de bloques dentro del sistema de archivos es considerablemente mayor que el número de bloques dentro de una franja de paridad.

Finalmente, para agrupar todos lo conceptos anteriormente descritos, es decir, las franjas de paridad, tamaños de las unidades de paridad, operaciones de bloqueo/desbloqueo, cálculos de paridad, etc., es necesario generar un algoritmo (figura 11) que permita de una manera ordenada y metódica aplicar cada una de estas ideas de una forma conjunta. Para el sistema de archivos distribuido y paralelo con redundancia de datos desarrollado en [6] se tiene el siguiente algoritmo:

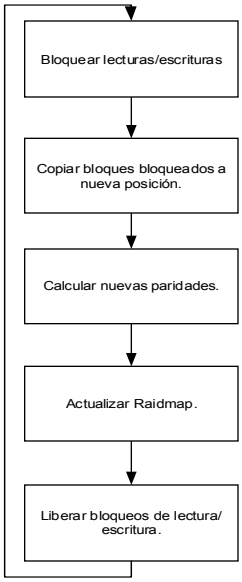

Figura 11. Diagrama de flujos del algoritmo de reconfiguración.

Bloquear lecturas/escrituras: El primer paso bloquea todas las franjas de paridad que serán trasladadas a su nueva posición, así también como las franjas de paridad que se encuentran en las posiciones de destino de los bloques a trasladar.

Copiar bloques bloqueados a nueva posición. Como segundo paso, se copian todos los bloques que pertenecen a la franja de paridad de origen hacia la franja de paridad destino.

Calcular nuevas paridades . El tercer paso es calcular y escribir las unidades de paridad de las franjas de origen y de destino.

Actualizar Traductor de Bloques. El cuarto paso es indicar en el traductor de bloques la nueva posición del puntero de reconfiguración con el fin de que este pueda redireccionar las acciones solicitadas a los bloques de manera correcta.

Liberar bloqueos de lectura/escritura. El quinto paso dentro del proceso de reconfiguración es liberar las franjas de paridad que se encontraban bloqueadas puesto que estas ya se encuentran actualizadas y entregando un normal servicio para los trabajadores que las soliciten.

4. EVALUACIÓN.

Para la evaluación definitiva del modelo propuesto, se procedió a modificar el simulador del sistema de archivos distribuido y paralelo con redundancia de datos desarrollado en [6] para incorporar los nuevos módulos. En primer lugar se detectaron cuales son las librerías sensibles a las nuevas funcionalidades para su posterior modificación. Principalmente, debieron modificarse aquellas secciones que tienen que ver con el funcionamiento de los procesos trabajadores que en este sistema se encuentran aleatoriamente repartidos con tiempo entre llegadas correspondientes a una distribución uniforme.

Las modificaciones realizadas a estos módulos corresponden a la adición de operaciones que les indiquen a los trabajadores que deben detener su ejecución hasta que el bloque sobre el cual deben actuar se encuentre liberado del proceso de reconfiguración. Además fue necesario agregar una funcionalidad, correspondiente al traductor de

bloques, que le indique a los nodos cual es el verdadero bloque sobre el cual debe trabajar. Por otro lado, estas nuevas funcionalidades provocan un retardo que es detectado por el sistema y permite generar las estadísticas finales que se muestran en los gráficos de los próximos capítulos.

4.1. Selección del tamaño de la Unidad de Reparto.

En definitiva, se ha optado trabajar con los siguientes tamaños de unidades de reparto:

Tamaño Gsuperstripe: Este es el tamaño de la unidad de reparto para el gestor de vraids, el cual se ha fijado en 4KB para las cargas OLPT y de 256KB para las cargas Científicas.

Tamaño Supertripe: Esto corresponde al tamaño de la unidad de reparto de las unidades VRAID y se ha fijado en 4KB para las cargas OLPT y de 64KB para las cargas Científicas.

Tamaño Stripe: Esta es unidad de reparto de los RAID que pertenecen a cada uno de los nodos del sistema. Los valores que se han fijado en este caso son de 4KB para las cargas OLPT y de 4KB para las cargas de tipo Científica.

El criterio utilizado para seleccionar los distintos tamaños de las unidades de reparto fue el hecho que es necesario comparar los resultados de este estudio con otros experimentos anteriores realizados en [6] y en [7], estudios en los que se utilizaron los valores anteriormente mencionados. Sin embargo es bueno mencionar que los valores seleccionados en esos estudios garantizaban que la división de la petición del cliente permitía el máximo del paralelismo dado en el sistema[7].

4.2. Selección de la Carga de Trabajo.

Tomando en consideración los estudios realizados por empresas y universidades se ha considerado trabajar con las cargas de tipo transaccional y las de tipo científica ya que representa a un porcentaje alto de la actividad de un Sistema de Archivo Distribuido y Paralelo.

Carga de Trabajo OLPT:

Las principales características de las cargas transaccionales con las que se trabajaron en este estudio se muestran en la Tabla 1.

Tabla 1. Perfil de una carga de trabajo OLPT.

Distribución Probabilística	Uniforme
Lecturas de 4 KB	8%
Escrituras de 4KB	88%
Lecturas de 24KB	2%
Escrituras de 24KB	2%

Carga de Trabajo Científica:

Para la carga de tipo científica se tiene el perfil de la Tabla 2.

Tabla 2. Perfil de una carga de trabajo científica.

Distribución Probabilística	Uniforme
Accesos	50% a 10 archivos de 5MB
Accesos	50% acceso secuencial a 1 archivo de 100MB
Lecturas a los archivos de 5MB	10% lecturas de 5MB
Escrituras a los archivos de 5MB	90% escrituras de 5MB
Lecturas al archivo de 100MB	90% lecturas de 1MB
Lecturas al archivo de 100MB	10% escrituras de 1MB

Al igual que en la selección del tamaño de la unidad de reparto y con el fin de poder realizar una comparación posterior, se ha decidido continuar con la línea planteada en los estudios realizados en [6] y en [7] y se ha optado por generar las pruebas en bases a las cargas de tipo OLPT y las cargas de tipo Científica. Además, se han realizado las pruebas utilizando tiempos entre peticiones generadas por las cargas de trabajo distribuidos de manera uniforme con tiempo de 250, 500, 750, 1000 t 1250 ms. en promedio.

4.3. Selección de la configuración del VRAID.

En cuanto a la configuración del VRAID, las pruebas se han realizado tomando en cuenta que se debe trabajar al igual que en [7], por lo que para las primeras pruebas que tienen que ver con la medición de la productividad en el sistema, se han utilizados configuraciones de dos sistemas VRAID con 3, 5 y 9 nodos tanto para las cargas OLPT como para las de tipo científica. Luego, para las pruebas que tienen que ver con la medición de los tiempos de lectura y escritura, se utilizaron configuraciones de dos VRAID de 5 nodos cada uno.

Conviene decir que para todas las configuraciones generadas para las distintas pruebas se han distribuido los nodos de los VRAID en una red de topología bus de 100Mbps y además cada uno de estos contenía una matriz de discos RAID.

4.4. Selección de la configuración de los RAID.

Para la selección de las configuraciones de las matrices de discos RAID contenidas en cada uno de los nodos, se utilizó el mismo criterio que para las configuraciones anteriores, es decir aquellas que permitiesen ser comparadas con los trabajos realizados en [6] y en [7].

En definitiva, cada una de las matrices RAID estaba compuesta por 5 discos y con una configuración de tipo RAID5 con simetría izquierda, además los discos estaban conectados con un bus de tipo SCSI2 con un ancho de banda de 20 MB/s y concurrencia activada. Resta decir que los discos simulaban discos de marca IBM, modelo ULTRA2 con 3534 cilindros, 21 pistas por cilindro, 110 sectores por pista, 10000 revoluciones por minuto, tiempo mínimo de búsqueda de 2.4 ms, tiempo promedio de búsqueda de 9 ms, tiempo máximo de búsqueda de 19 ms, caché de 2048 KB y 4GB de tamaño.

4.5. Resultados Carga OLPT

Después de ejecutar el sistema simulador con las modificaciones necesarias, se midieron los tiempos de respuesta del sistema para peticiones de lecturas y escrituras encontrándose lo siguiente:

- *Tiempos de Lectura (figura 12):*

En este caso, se graficaron los valores obtenidos para peticiones cuyos tiempos entre llegadas se distribuyen de manera uniforme entre 0 y 250, 500, 750, 1000 y 1250 milisegundos. Se puede apreciar que los valores obtenidos tanto para el modo normal (puntos marcados con cuadrados) como para el modo de reconfiguración (puntos marcados con triángulos), no existe una gran diferencia. Esto se puede explicar debido a que el proceso de reconfiguración adiciona tan solo una pequeña cantidad de proceso que es despreciable si se compara con los tiempos totales del sistema.

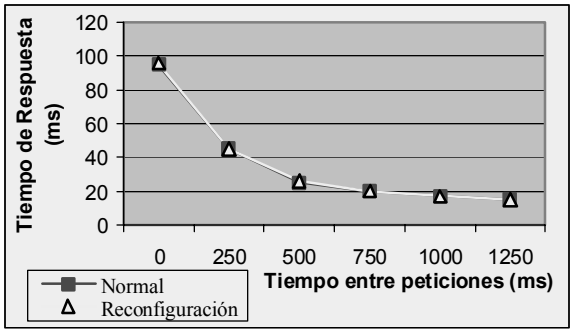

Figura 12. Gráfico de tiempos de lecturas del sistema para el modo normal y en reconfiguración con cargas de tipo OLPT.

- *Tiempos de Escritura (figura 13):*

Al igual que el gráfico anterior, este presenta los valores obtenidos para el sistema en los estados normal y en reconfiguración para los tiempos entre llegadas de las peticiones de lectura/escritura distribuidas de manera uniforme entre 0, 250, 500, 750, 1000 y 1250 milisegundos. De la misma manera, se aprecia que el tiempo de respuesta del modo reconfiguración cuando los tiempos entre llegadas de las peticiones es pequeño es levemente superior a los tiempos del estado normal, tendiendo estos tiempos de respuesta a igualarse a medida que se incrementan los tiempos entre llegadas. Si se toma en cuenta que todo experimento considera un rango de incertidumbre, puede considerarse que la diferencia de los primeros valores es prácticamente nula.

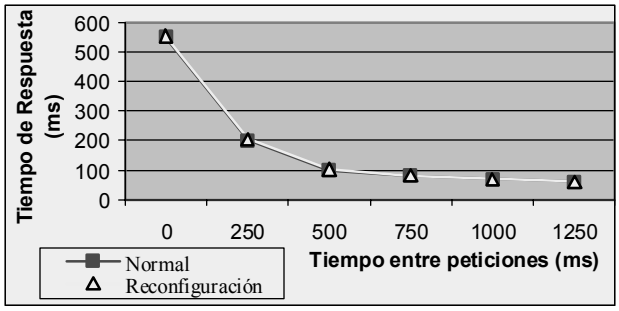

Figura 13. Gráfico de tiempos de escrituras del sistema para el modo normal y en reconfiguración con cargas de tipo OLPT.

4.6. Resultados Carga Científica

Para el caso de las cargas de tipo científica, se realizo el mismo tipo de pruebas que para las cargas de tipo OLPT. Los resultados obtenidos para los distintos tipos de peticiones se muestran en la fig. 1.4

Los tiempos de lectura de las cargas de tipo científica se realizaron con los mismos valores que para las carga de tipo OLPT. En este gráfico se puede apreciar que los tiempo de lectura para el tipo de cargas de tipo científica es mayor que los tiempos de lectura para cargos de tipo OLPT tendiendo a igualarse a medida que aumentan los tiempos entre peticiones. Sin embargo, al igual que en las cargas de tipo OLPT, la diferencia de tiempos entre las modalidades normal y reconfiguración son prácticamente los mismos estos se muestran en la fig. 1.5

Se puede apreciar al igual que el gráfico anterior que los tiempos de respuesta son mejores para las cargas de tipo OLPT que para las de tipo científica sobre todo cuando los tiempos entre llegadas de las peticiones son bajos. Además también se puede ver que de la misma manera que los gráficos anteriores, el tiempo de respuesta del sistema para los modos normal y reconfiguración es bastante similar.

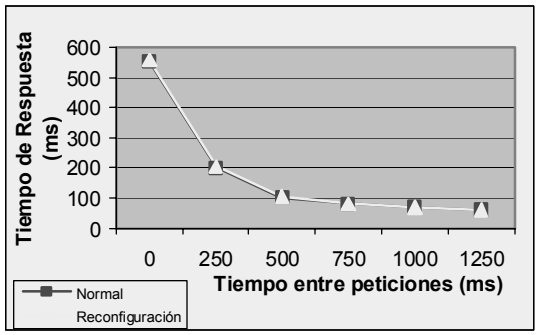

Figura 14. Gráfico de tiempos de lecturas del sistema para el modo normal y en reconfiguración con cargas de científica.

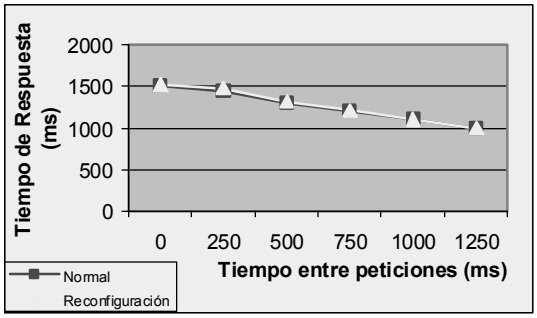

Figura 15. Gráfico de tiempos de escrituras del sistema para el modo normal y en reconfiguración con cargas científica.

5. CONCLUSIONES

Del trabajo desarrollado en esta tesis se puede concluir lo siguiente:

1. A medida que el tiempo entre peticiones aumenta, el sistema tiende a comportarse de la misma manera en modalidades normal y de reconfiguración.

2. El modelo seleccionado para realizar las pruebas es una alternativa viable para ser implementada en la realidad puesto que cumple con el objetivo de reconfigurar el sistema desde un estado inicial de desbalance a uno final en el que los datos se encuentran repartidos de manera homogénea.

3. Cuando el sistema se encuentra en modo de reconfiguración existe una carga de trabajo adicional que se puede considerar despreciable respecto a la carga de trabajo total cuando el sistema se encuentra en modo normal.

Esto se explica por medio del siguiente análisis: En las pruebas realizadas se utilizó un sistema compuesto por 50 discos de 4GB cada uno por lo tanto el tamaño total del sistema es de 208,98 GB. El tamaño de cada franja de paridad es de 640KB, existiendo por lo tanto 326541 franjas. El proceso de reconfiguración bloquea dos franjas, una para la lectura de la franja que se esta trasladando y otra que es la franja que se esta escribiendo. Esto implica que la probabilidad de que una petición se efectúe sobre una franja bloqueada se obtiene en la ecuación (2):

$$P = \frac{2}{326541} = 6{,}12 * 10^{-6} \qquad (2)$$

De aquí se puede apreciar que el valor de P es extremadamente pequeño, sin embargo si se considera que el tiempo entre peticiones es de 0, y el número de peticiones que se ejecutan es de 150 como en el caso de las pruebas realizadas, entonces suponiendo que todas las peticiones han bloqueado una franja la ultima petición tiene una probabilidad igual a la que se encuentra en la ecuación (3) de encontrar una franja ocupada:

$$P = \frac{152}{326541} = 0{,}0004593 \qquad (3)$$

De lo anterior se aprecia que las probabilidades de que una petición encuentre ocupada la franja sobre la que realizará la operación es baja. De todas maneras es bueno hacer notar que estos cálculos suponen que las peticiones son independientes lo que en la realidad no es así puesto que las peticiones tanto de lectura y escritura tienden a agruparse sobre franjas contiguas.

REFERENCIAS

[1] Real Academia Española. Diccionario de la Lengua Española. Pagina 822. Junio de 1992.

[2] D. Patterson, G. Gibson, and R. Katz. A Case for Redundant Arrays of Inexpensive Disks (RAID). *In Proc of ACM SIGMOD* p 109-116. ACM, June 1988.

[3] N. Nieuwejaar, D. Kotz, A. Purakayastha, C.S. Ellis and M. Best, "File access characteristics of parallel scientific workloads" Tech. Rep. PCS-TR95-263, Dartmouth College, Aug. 1995.

[4] D. Kotz and N. Ñieuwejaar. File System Workload on a Scientific Multiprocessor. IEEE Parallel and Distributed Technology. Systems and Aplications, pages 134-154, Spring 1995.

[5] M. Ñelson, B. Welch, and J. Ousterhout. Caching in the Sprite Network File System. *ACM Transactions on Computer Systems*, 6(1):134-154, February 1988.

[6] Rosales 98 Rosales F, Vega R., Achieving Data Availability on Parallel and Distributed File Systems, Proceding 3° International Meeting On vector and Parallel Processing, Facultadade de Engenharia da U. do Porto Portugal Jun 1998, pag 645-650.

[7] Vega[01] Vega R., Romo C., Rosales F., Mejora de la disponibilidad de datos de un subsistema de E/S paralelo mediante agrupación de componentes. XXVII Conf. Latinoamericana de Informática (CLEI 2001), Sep. 2001, U. de Los Andes, Mérida, Venezuela.

[8] Vega[00] Vega R., Rosales F, Carretero J, Propuesta y Evaluación de un Modelo de E/S Redundante para un Sistema de Ficheros Distribuido y Paralelo. XXVI CLEI 2000, Septiembre 2000, Tecnológico de Monterrey, México. Elegido entre 10 mejores artículos