Propuesta y Evaluación de un Modelo de E/S redundante para un Sistema de Ficheros Distribuido y Paralelo

[1]Raimundo Vega Vega, Francisco Rosales García, Jesús Carretero Pérez

Resumen

Este artículo propone un subsistema de E/S redundante mediante esquemas de replicación como los usados en las Matrices de Discos Redundantes (Raid), con el objeto de que un sistema de ficheros distribuido y paralelo dé servicios tolerantes a fallos.

Se propone un subsistema jerárquico separando las funcionalidades sobre la gestión de nodos y la gestión de discos. La gestión de nodos será realizada por un dispositivo lógico redundante llamado VRAID que es posible configurar mediante esquemas de redundancia, como los usados en las Matrices de Discos redundantes de nivel 0, 4 y 5, y que corresponde en sentido jerárquico al nivel superior del subsistema de E/S y que proporciona las interfaces correspondientes para dar servicio a la capa superior, que en este caso es un sistema de ficheros distribuido y paralelo. La gestión de los discos la lleva a cabo un dispositivo redundante llamado Raid que es posible configurar mediante esquemas de redundancia tipo Raid de nivel 0, 4 y 5.

Para la implementación del subsistema de E/S redundante y su posterior evaluación se han elegido técnicas de simulación ya que no disponemos del hardware que permita implementaciones reales y debido a que los modelos analíticos que pudieran emplearse para el cálculo de rendimiento del tipo de arquitectura que se propone, son hasta ahora difíciles de formular, debido a que una sola operación de un cliente incluye fenómenos de espera en cola fork-join.

Palabras claves—Sistemas de Ficheros, Disponibilidad de Datos, Sistemas Distribuidos, Paralelismo, Redundancia.

[1] Dr. Raimundo Vega Vega
Facultad de Ingeniería
Campus Miraflores
Universidad Austral de Chile
Valdivia, Chile
rvega@uach.cl

Dr (c) Francisco Rosales García
Facultad de Informática
U. Politécnica de Madrid
Campus de Montegancedo
Boadilla del Monte
28660 Madrid /España
frosal@fi.upm.es

Dr. Jesús Carretero Pérez
Depto. Informática
U. Carlos III de Madrid
28911 Leganes, Madrid
España
jcarrete@arcos.inf.uc3m.es

I. Introducción

Un sistema de ficheros distribuido y paralelo (SFDP) almacena los datos a través de múltiples dispositivos de almacenamiento, permitiendo que un acceso a un fichero sea atendido por más de un dispositivo. Esto permite aumentar el ancho de banda con el número de dispositivo que participan en la transferencia. Los dispositivos de almacenamiento pueden ser discos, nodos de E/S en un computador paralelo, o servidores de ficheros en un sistema de ficheros en red.

La mayoría de los SFDP existentes Swift [Cabrera91], Zebra [Hartman95], ParFiSys[Carretero96] [Carretero95a] [Carretero95b], proporcionan pocos mecanismos de tolerancia a fallos. Un SFDP debería ofrecer, al menos, dos mecanismos importantes de tolerancia a fallos: uno que permita recuperar los datos de un dispositivo averiado y un segundo mecanismo que permita volver arrancar un dispositivo averiado y reconstruir sus datos sin tener que detener las actividades del sistema.

En este artículo se propone y evalúa un subsistema de E/S estructurado como una jerarquía de tres niveles. En el nivel más bajo, se encuentran los dispositivos hardware de almacenamiento (discos). En un segundo nivel, un dispositivo tipo RAID que controla el reparto de datos sobre el conjunto de los discos de un nodo de E/S. Por último, en el nivel más alto, el subsistema de E/S implementa un dispositivo RAID virtual (VRAID) que controla la distribución de datos entre el conjunto de los nodos de E/S. Para cada uno de los dispositivos descritos (disco, RAID, VRAID) se plantea un modelo de transición entre los siguientes estados: libre de fallos, fuera de servicio, servicio degradado, fase de reconstrucción.

II. Modelo de subsistema de E/S redundante para un SFDP

2.1. Propuesta del modelo de subsistema E/S redundante

Un sistema de E/S distribuido y paralelo permite el acceso simultáneo a datos distribuidos a través de múltiples discos conectados a múltiples nodos de E/S. Para explotar las ventajas de la arquitectura de E/S paralela se propone diseñar un subsistema de E/S paralelo redundante separando la funcionalidad en niveles. La funcionalidad del subsistema de E/S redundante se ha dividido en dos tipos de subconjuntos básicos [Rosales98], [Vega98]: gestión de los nodos y gestión de los discos dentro de los nodos. Formalmente, se puede caracterizar un subsistema de E/S básico mediante la siguiente tupla:

SS = {VRAID, ESTADO_GLOBAL}

Compuesta por dispositivo lógico VRAID que corresponde, en sentido jerárquico, al nivel superior del subsistema de E/S y que proporciona las interfaces correspondientes para dar servicio a la capa superior, que en este caso es un sistema de ficheros. Además, por un ESTADO_GLOBAL que lo determinan los estados de los dispositivos internos del

subsistema de E/S. La tupla que caracteriza el subsistema de E/S se detalla de la siguiente forma:

VRAID= {{ NODOS }, Tamaño unidad reparto, Tipo redundancia, Estado, Capacidad }
NODOS = { Raid }
Raid = {{ DISCOS }, Tamaño unidad reparto, Tipo redundancia, Estado, Capacidad}
DISCOS={Tamaño sector, Nº sectores, Nº cilindros, Vel. acceso, Latencia, Nº superficies}
ESTADO_GLOBAL = {Normal, Degradado, En Reconstrucción, Fuera de Servicio}

Los componentes modelados del subsistema de E/S redundante se pueden observar en la Fig 2.1. Su estructura jerárquica está constituida en un primer nivel por el dispositivo VRAID, una red de interconexión de alta velocidad, los dispositivos Raids y en cada dispositivo Raid se ubican discos o Matrices de Discos Redundantes (Raid). El dispositivo lógico VRAID provee las interfaces para establecer la comunicación con el sistema de ficheros. Tanto los dispositivos Raid y Vraid pueden ser configurados según los esquemas tipo RAID con nivel 0, 4 y 5, [Patterson88][Chen94].

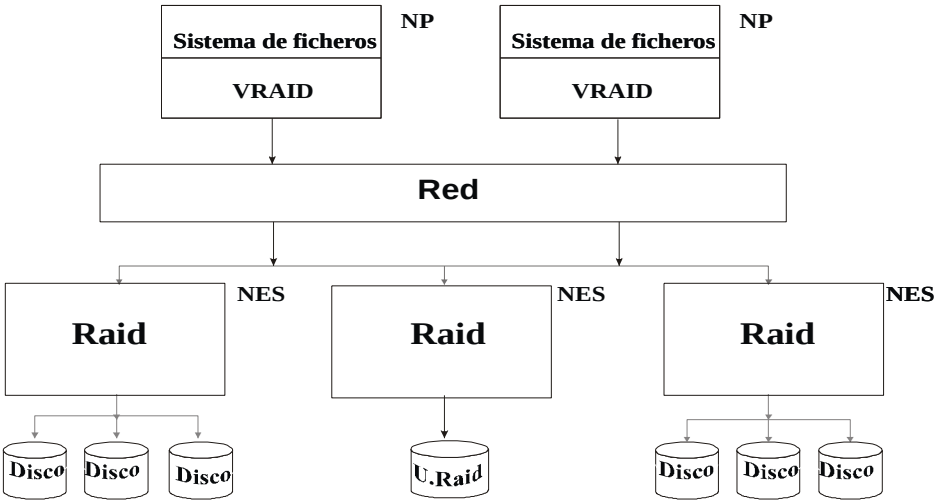

Fig. 2.1. Arquitectura de un Subsistma de E/S redundante

2.2. Selección del tamaño de la unidad de reparto

La unidad de reparto es el conjunto de datos lógicamente contiguos almacenados en un mismo nodo cuando se trata del dispositivo VRAID o en un mismo disco cuando se trata del dispositivo Raid. La selección del tamaño de la unidad de reparto para cada uno de los dispositivos tiene que hacer frente a dos objetivos:

- Maximizar la cantidad de información que se puede transferir en cada operación de E/S lógica.
- Utilizar de forma equilibrada todos los nodos que componen el VRAID y todos los discos que componen el Raid, para evitar que mientras algunos nodos y/o discos están cargados, otros estén ociosos.

2.3. Estados de un dispositivo

El estado de los dispositivos tanto físicos como lógicos (VRAID,Raids) puede ser definido de acuerdo al estado de sus dispositivos subyacentes y la consistencia de la paridad con los datos. Es así como los dispositivos pueden estar en uno de los siguientes estados:

- Libre de fallo. Los dispositivos subyacentes están libres de fallo y la información redundante es consistente con los datos almacenados.
- Degradado. Uno de los dispositivos subyacentes está fuera de servicio, pero la redundancia almacenada permite mantener los accesos a los datos.
- Fuera de servicio. Más de un dispositivo subyacente está fuera de servicio, por tal razón no hay redundancia disponible para reconstruir los datos inaccesibles.
- En Reconstrucción. El dispositivo subyacente averiado está retornando a un estado libre de fallo, pero los datos almacenados necesitan ser actualizados para que exista consistencia entre paridad y datos.

Los eventos significativos que reflejan un cambio de estado en los dispositivos son los siguientes: Fallo de un componente, inicio de reconstrucción, fin de la reconstrucción.

2.4. Estrategia para la gestión de los dispositivos

Para gestionar homogéneamente tipos de dispositivos heterogéneos vamos a ocultar los detalles bajo la interfaz de un dispositivo abstracto de almacenamiento. Un dispositivo abstracto de almacenamiento es el proveedor de un conjunto de acciones de E/S para un número de unidades de almacenamiento de un tamaño específico. Cada acción al menos contendrá una de las siguientes operaciones:

- LOCK bloquea la unidad especificada.
- LEER lee un número de unidades a partir de una unidad dada.
- XOR calcula el or-exclusivo de la franja de paridad indicada.
- ESCRIBIR escribe un número de unidades a partir de una unidad dada.
- UNLOCK desbloquea la unidad especificada.

Cuando un dispositivo abstracto realiza una operación de lectura o escritura, la descompondrá en un conjunto de acciones dirigidas a sus dispositivos subyacentes. La tabla 1 muestra el criterio aplicado para una lectura, escritura total, parcial y larga para cada franja de paridad.

	LOCK	LEER	XOR	ESCRIBIR	UNLOCK
Leer	NO	Unidades	NO	NO	NO
Escritura Total	Paridad	NO	Si	Unidades	Paridad
Escritura Parcial	Paridad	Unidades	Si	Unidades	Paridad
Escritura Larga	Paridad	Resto	Si	Unidades	Paridad

Tabla 1. Criterio de descomposición por franja de paridad

2.5. Mejorando el rendimiento

Dependiendo del tamaño, una acción de E/S puede descomponerse en un gran número de subacciones sobre un conjunto de unidades de almacenamiento de los dispositivos subyacentes. Para reducir la cantidad de subacciones individuales y para optimizar el acceso a los dispositivos subyacentes, este conjunto es reordenado juntando subacciones que son lógicamente contiguas, es decir: se refieren al mismo dispositivo de almacenamiento subyacente, son del mismo tipo de acción (LOCK, LEER, XOR, ESCRIBIR o UNLOCK), conciernen a un conjunto contiguo de unidades.

El conjunto resultante de subacciones es procesado ejecutándose en paralelo las acciones para cada uno de los dispositivos, haciéndolos en el siguiente orden: todo los bloqueos, todas las lecturas, el cálculo interno del XOR, todas las escrituras y finalmente los desbloqueos. Este método tiene las siguientes propiedades:

- Asegura la consistencia entre los datos y la paridad de cada franja de paridad.
- Minimiza el número de final de acciones y por tanto (caso de VRAID) el tráfico de red.
- Las acciones finales por dispositivo, son más compactas y puede ser realizadas más eficientemente.

En el ejemplo de la Fig. 2.2, podemos observar que un cliente hace una operación de escritura de las unidades 4 al 12, sobre un dispositivo compuesto por 4 dispositivos subyacentes y cuyo esquema de redundancia es Raid nivel 5. Las unidades son proyectadas desde la franja de paridad 1, a partir del dispositivo subyacente 1, a la franja de paridad 4 que incluye sólo el dispositivo subyacente 0.

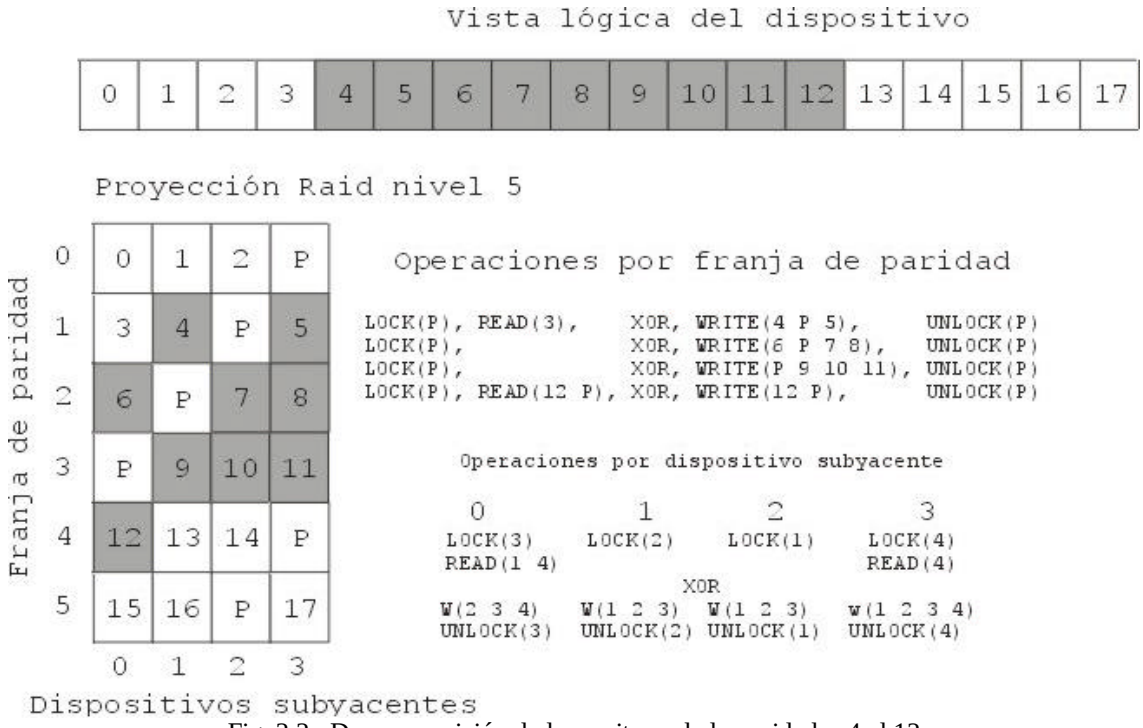

Fig. 2.2. Descomposición de la escritura de las unidades 4 al 12

2.6. Gestión distribuida de cerrojos en el VRAID

Toda operación que manipule la paridad deberá, una vez terminada la operación, mantener la coherencia en la franja de paridad, es decir, que las unidades de reparto de datos sean coherentes con la unidad de reparto que almacena la paridad. Por lo tanto, las operaciones de escritura en modo normal y degradado deberán usar cerrojos, así como la lectura en modo degradado. En cuanto a la operación de lectura, se ha decidido llevarla a cabo sin aplicar cerrojos, aun tomando en cuenta que se podría leer información obsoleta. Se impuso esta opción de cara a mantener un buen rendimiento en el sistema.

Para bloquear una franja de paridad se ha decidido bloquear solo la unidad de reparto que almacena la paridad. En el caso degradado, se ha decidido que si la paridad está almacenada en el dispositivo averiado, le será impuesto el cerrojo a la unidad de paridad de datos del dispositivo adyacente, de tal manera de garantizar el acceso exclusivo a la franja de paridad. Una vez que el dispositivo averiado se recupera se aplican los cerrojos en la unidad de reparto que almacena la paridad.

2.7. Algoritmo para la reconstrucción de un dispositivo

El algoritmo propuesto en este trabajo, divide el espacio de almacenamiento en zonas constituidas por un conjunto de franja de paridad. Una vez recuperada una zona, los procesos clientes las pueden acceder en modalidad normal. De esta forma, mientras se ejecuta el algoritmo de reconstrucción, habrá algunos procesos accediendo al espacio de almacenamiento en modalidad normal y otros en modalidad degradada. Independiente del dispositivo averiado (nodo o disco), el algoritmo bloquea la zona que va a regenerar con el fin de que ningún proceso cliente las pueda acceder. Para regenerar la información, el algoritmo de reconstrucción lee todas las unidades de reparto de una franja de paridad, incluyendo la que contiene a la paridad, y ejecuta la operación Xor para obtener la información que será escrita en el dispositivo recuperado.

Para no degradar el tiempo de respuesta de los clientes mientras se ejecuta el algoritmo de reconstrucción y a la vez alcanzar tiempos prudentes de reconstrucción, se ha decidido que el proceso reconstructor mantenga la misma prioridad de ejecución que los procesos clientes.

III. Evaluación de prestaciones

Para caracterizar el subsistema de E/S redundante se han elegido los dispositivos físicos que podrían ser usados en una arquitectura real.

Discos	Nº Cilindros	Nº pistas	Nº Sectores	Rev. Min.	T. Busqueda			Cache	Bus	
					Min	Prom	Max		Tipo	A/B
Seagate	2627	21	99	5400	1.7	11.0	22.5	1024	SCSI2	20MB

Para una simulación aproximada al funcionamiento real de una matriz de discos, se considera que éstos no están sincronizados y el lugar de inicio de cada disco es aleatorio

según una distribución de probabilidad uniforme. Los datos podrán estar almacenados sobre discos según una distribución uniforme o bien según una distribución exponencial.

La Red de interconexión tendrá un ancho de banda de 100 Mbytes/seg y la topología que se usará preferentemente será full-connect. Se ha decidido considerar para la simulación sólo el tiempo de transmisión, ignorando el "overhead" de emisor y receptor.

Los esquemas de redundancia usarán los niveles Raid 4 y 5, distribuyendo la paridad en forma simétrica hacia la izquierda. Se evaluará el sistema usando 5 discos por nodos y el número total de clientes será igual a un factor del número de discos en el sistema. Se simulará una carga de tipo transaccional en línea [Holland94] y otra científica, con las siguientes características:

Procesamiento transaccional en línea			Simulación científica
	4KB	24KB	50% de acceso de 1MB a un fichero de 100 MB
Lecturas	80%	2%	Tipo de acceso secuencial:
Escrituras	16%	2%	90% lecturas, 10% escrituras
			50% de acceso 1MB a 5 ficheros de 5 MB
Ficheros distribuidos uniformemente, acceso según una distribución exponencial.			Tipo de acceso secuencial: 10% lecturas y 90% escrituras Ficheros distribuidos uniformemente.

3.1. Cálculo de la unidad de reparto

Cada cliente realizó durante el transcurso de la simulación peticiones intensivas sobre los dispositivos, es decir, sin tiempo de retardo entre petición. Se eligió el esquema de redundancia (VRAID/Raid) 5/5 para hacer la evaluación. Para el caso de carga OLPT, se seleccionaron los tamaños de unidades de reparto VRAID/Raid: 4/1, 8/2, 16/4 y 4/4.

En el gráfico de la Fig. 3.1, muestra un alto rendimiento del esquema redundante 4/4, esto se debe a que el tamaño de la unidad de reparto, tanto del VRAID como Raid, coincide con el tamaño de las peticiones de 4KB. Esto hace que un alto porcentaje de las peticiones obtenga el máximo paralelismo del subsistema.

Para el caso de carga Científica, se seleccionaron los tamaños de unidades de reparto VRAID/Raid 256/64, 128/32, 64/16, 32/8, dicha selección estuvo basada en el tamaño de las peticiones en este tipo de carga. Se observa que la combinación 256/64 alcanza el mejor rendimiento.

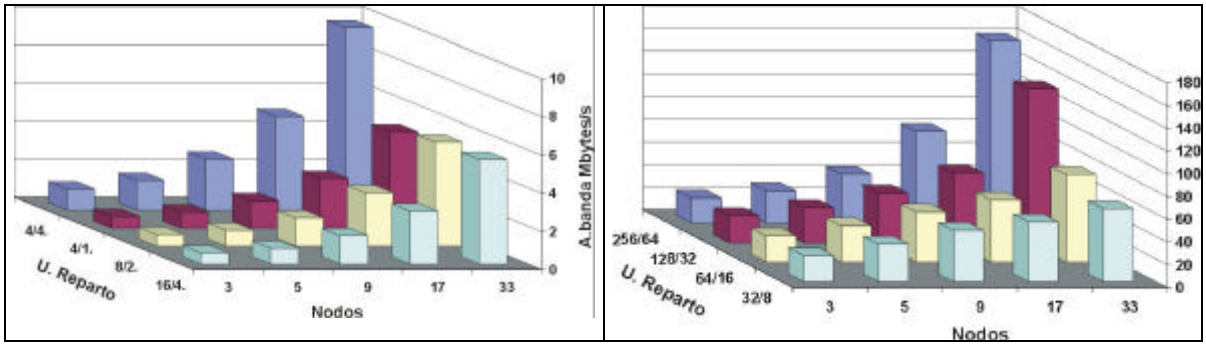

Fig. 3.1. Estimación del tamaño de la Unidad de Reparto OLPT-Científica

3.2. Evaluación de prestaciones bajo carga OLPT

Con el fin de investigar la capacidad máxima que el subsistema de E/S puede lograr con este tipo de carga, se consideraron configuraciones con 3, 5, 9, 17 y 33 nodos para evaluar la productividad. Para observar el punto de saturación del subsistema se analizó el comportamiento con un número de clientes igual a 1, 2, 3, 4 y 5 veces el número total de discos en el sistema haciendo peticiones de lectura y escrituras intensivas.

Al analizar el comportamiento de la productividad del subsistema, se observó que el rendimiento se incrementa en la medida que aumentamos el número de clientes por discos hasta 4. Por lo tanto, el subsistema estabiliza su productividad a partir de un número de clientes igual a 5 por disco. La Fig. 3.2 muestra la productividad obtenida con un número de clientes de 4 y 5 por discos.

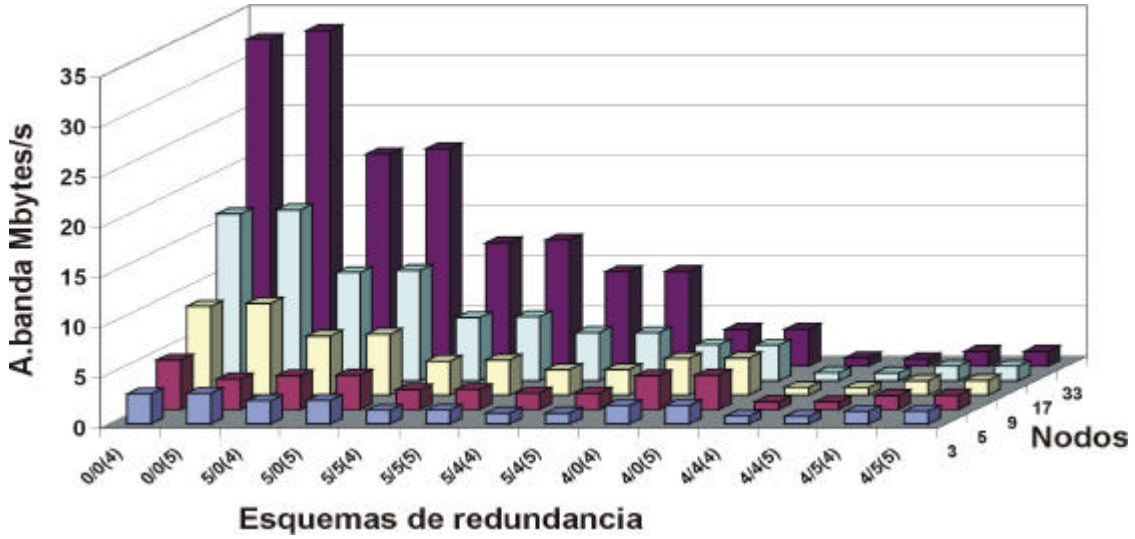

Fig. 3.2. Productividad del subsistema carga OLPT

Se observa por otro lado, que toda configuración que usa esquemas de redundancia nivel 5 en el dispositivo VRAID es escalable ya que en la medida que aumenta el número de nodos en las configuraciones y el número de clientes en el sistema, incrementa la productividad.

El bajo rendimiento del subsistema usando esquemas de redundancia nivel 4, se debe a que este esquema de redundancia usa un único dispositivo para almacenar la paridad y éste actualizados por los procesos escritores generándose una alta competencia por dicho recurso.

La Fig. 3.3. muestra la productividad del subsistema con carga científica. En este caso, el número total de clientes necesario en el sistema para alcanzar el máximo de rendimiento, es el equivalente a dos por disco. También se puede observar que, las configuraciones con esquemas de redundancia nivel 4 tienen un bajo rendimiento debido a la competencia por

el acceso al nodo de paridad. El esquema de redundancia nivel 5 es el que supera en rendimiento al resto.

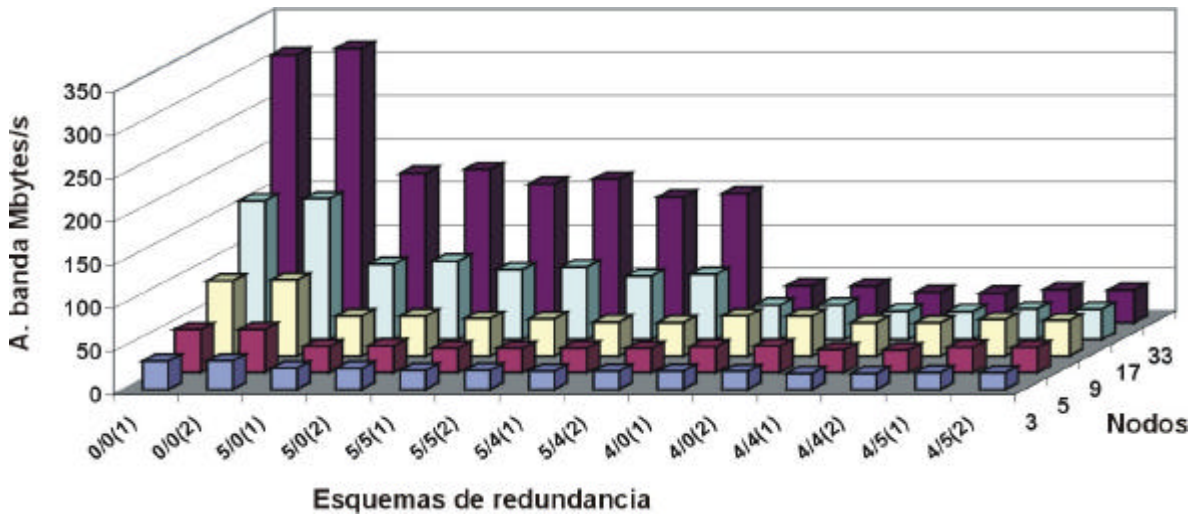

Fig. 3.3. Productividad del subsistema carga Científica

3.3. Tiempos de lectura y escrituras.

Para estimar los tiempos de lectura y escrituras promedio de los clientes en modalidad normal, degradado y en reconstrucción, sé hicieron evaluaciones con clientes que realizaron peticiones de forma intensiva (sin retardo entre peticiones) y evaluaciones en que los clientes realizaron peticiones según una distribución uniforme, con intervalos de tiempo entre petición de: 0-250, 0-500, 0-750, 0-1000, 0-1500 y 0-2000ms. De cada una de estas evaluaciones, obtuvimos el número de peticiones por segundo y el tiempo de respuesta promedio de la petición. Esto nos permite observar en los gráficos siguientes el comportamiento el subsistema en la medida que disminuye la intensidad de trabajo.

La Fig. 3.4. muestra los tiempos de respuesta de lecturas y escrituras en relación con el número de peticiones por segundo de los clientes en las modalidades de trabajo normal, degradado y en reconstrucción, para una carga OLPT.

En la modalidad de trabajo degradado y en reconstrucción, el subsistema de E/S redundante simula tener averiado un componente del dispositivo redundante VRAID. El esquema de redundancia usado en el dispositivo VRAID y Raid es de nivel 5. Se hicieron evaluaciones considerando zonas de tamaño 10 y 100 franjas de paridad.

Se analizan configuraciones con 3 y 33 nodos, cada nodo tiene asociado 5 discos. Se observa que los clientes en modalidad reconstrucción tienen, a partir de aproximadamente 150 peticiones por segundo, mejores tiempos de respuesta que en modalidad degradada.

Este comportamiento se observa tanto en lecturas como escrituras y en ambas configuraciones ya que en la configuración con 33 nodos, a partir de 1600 peticiones por

segundo, se observa un comportamiento similar. Esto se debe al hecho que el proceso reconstructor una vez que regenera la información de una zona la hace disponible a los clientes para que la accedan en modalidad de trabajo normal.

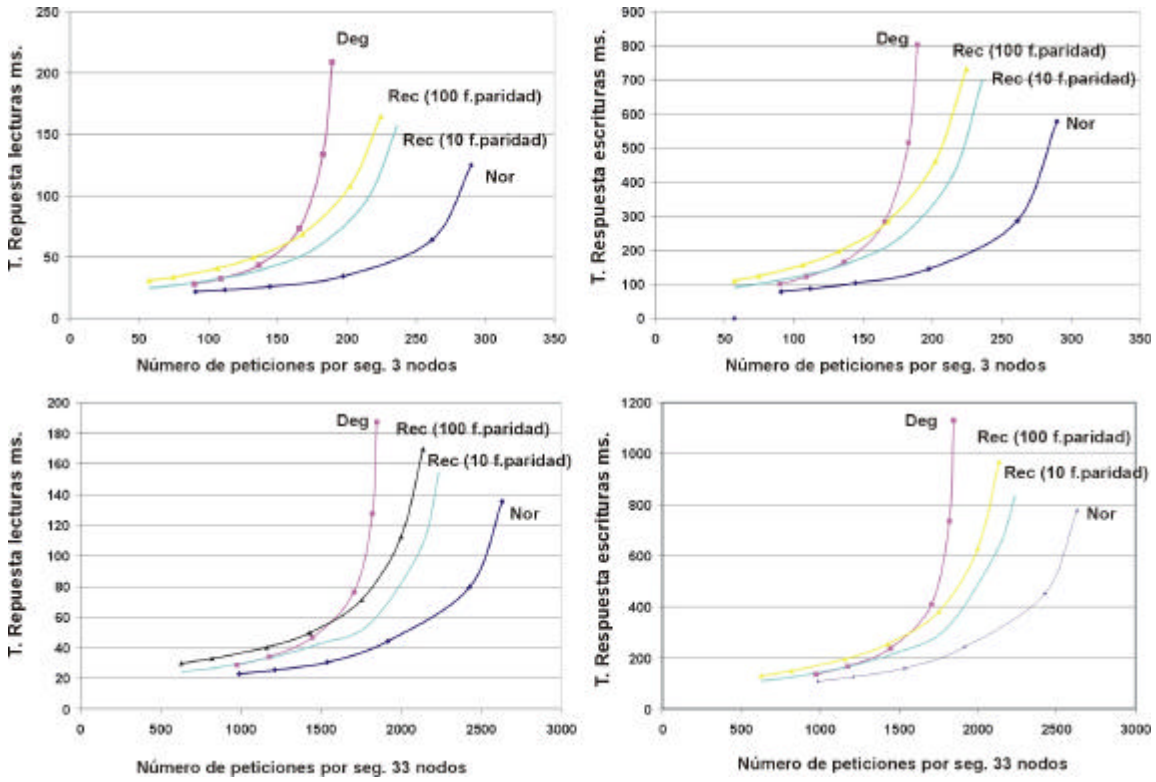

Fig. 3.4. Tiempos de Lectura/Escritura normal, degradado y en reconstrucción.

La Fig. 3.5, muestra los tiempos de respuesta de lecturas y escrituras en relación con el número de peticiones por segundo de los clientes, en las modalidades de trabajo normal, degradado y en reconstrucción, para una carga Científica.

Se observa que los tiempos de respuestas en modalidad reconstrucción considerando zonas de 100 franjas de paridad es muy alto, tanto en la configuración de 3 y 33 nodos. Esto se debe al tamaño de la petición y el tamaño de la zona de reconstrucción. En cambio, la modalidad reconstrucción con zonas igual a 10 franjas de paridad, tiene buenos tiempos de respuesta. En el caso de la configuración con 3 nodos, a partir de 15 peticiones por segundo la modalidad reconstrucción ofrece mejores tiempos de respuesta que la modalidad degradada. Lo mismo en el caso de la configuración con 33 nodos, a partir de 80 peticiones por segundo, ofrece mejores tiempos de respuesta que la modalidad degradada.

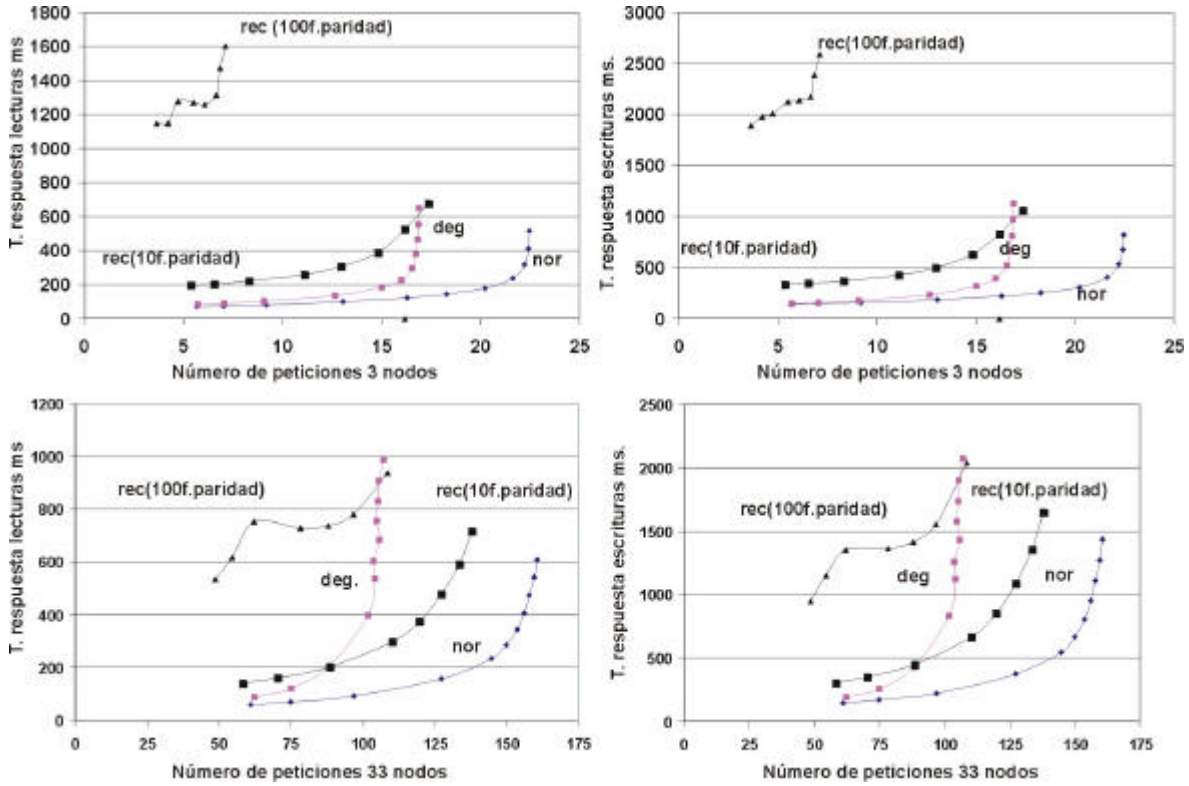

Fig. 3.5. Tiempos de Lectura/Escritura normal, degradado y en reconstrucción.

IV. Conclusiones

En este artículo se ha propuesto un modelo de subsistema con tolerancia a fallos mediante la incorporación de redundancia según los esquemas usados por las Matrices de Discos Redundantes (Raid). Se han propuesto dos tipos de dispositivos redundantes, cuya funcionalidad permite al subsistema de E/S proporcionar servicios tolerantes a fallos. El dispositivo VRAID, tiene como función la gestión de los nodos y el dispositivo Raid se encarga de la gestión de los discos.

Para gestionar las peticiones de lectura y escritura de los dispositivos VRAID y Raid, se propone una estrategia basada en el concepto de acción. Cada acción al menos contiene una de las siguientes operaciones: LOCK, LEER, XOR, ESCRIBIR, y UNLOCK. De esta forma, cuando un dispositivo realiza una petición de lectura o escritura, la descompone en un conjunto de acciones dirigidas a sus dispositivos subyacentes.

Mediante la evaluación del subsistema de E/S se concluye que, éste bajo un esquema de redundancia nivel 5, tanto en el dispositivo VRAID como Raid, logra el mejor rendimiento en comparación con el resto de los esquemas de redundancia. Se observo que el tiempo de respuesta de los clientes bajo el modo reconstrucción, depende fuertemente del tamaño elegido de la zona de reconstrucción. Concluimos que mientras más pequeña la zona mejor es el tiempo de respuesta en los clientes. Por otra parte, en trabajos anexos [Vega99] demostramos que los tiempos de reconstrucción de un dispositivo, decrecen en la medida que aumenta el tamaño de la zona de reconstrucción. Por tal razón, es necesario un

compromiso coste beneficio al decidir sobre el tamaño de la zona de reconstrucción ya que un menor tiempo de reconstrucción disminuye la probabilidad que el sistema este expuesto a un segundo y definitivo fallo.

Bibliografía

[Cabrera91] Cabrera L.F., Long D.E., Swift: Using Distributed Disk striping to Provide High I/O Data Rate. Computing Systems, 405-436, 1991.

[Carretero96] Carretero J., Pérez F, de Miguel P, García F, Alonso L, ParFiSys: A Parrallel File Systems for MPP. ACM SIGOPS 30(2):74-80, Abril 1996.

[Carretero95a] Carretero J., Un Sistema de Ficheros Paralelo con Coherencia de Cache para Multiprocesadores de Propósito General. Tesis Doctoral, Facultad de Informática, Universidad Politécnica de Madrid, Mayo 1995.

[Carretero95b] Carretero J, Pérez, de Miguel P, García F, Alonso L., Prototype Posix-Style Parallel File Server for the GPMIMD: Final Report. Technical Report ESPRIT Project P5404, D1.7/2, CEE, UPM, 1995.

[Chen94] Chen P., Lee E., Gibson G., Katz R., Patterson D. RAID: High-Performance, reliable secondary storage. ACM Computer Survey, 26(2), junio 1994.

[Hartman95] Hartman, J, Ousterhout J.K., The Zebra Stripe Network File System. ACM Transactions on Computer Systems, 13(3):274-310, agosto 1995.

[Holland94] Holland M., On line Data Reconstruction in Redundante Array. PhD. Thesis. Informe Técnico, Carnegie Mellon University, School of Computer Science.

[Patterson88] Patterson D., Gibson G., Katz R., A case for Redundant Arrays of Inexpensive Disks (Raid). Proceeding of ACM SIGMOD, pag. 109-116, ACM, junio 1988.

[Rosales98] Rosales F, Vega R., Achieving Data Availabitity on Parallel and Distributed File Systems 3rd International Meeting on Vector and Parallel Processing, 21-23 june 1998, Porto, Portugal.

[Vega98] Vega R., Rosales F., Disponibilidad de Datos en Sistemas de Ficheros Paralelos. IX Jornada de Paralelismo, Donostia España, pag 422-431, Septiembre 1998.

[Vega99] Vega R., Aspectos de Tolerancia a Fallos en un Sistema de Ficheros Distribuido y Paralelo mediante Esquemas de Redundancia. Tesis Doctoral, Facultad de Informática, Universidad Politécnica de Madrid. Octubre 1999.