Evaluación de Desempeño de un Sistema de Almacenamiento Distribuido

Idónea Comunicación de Resultados para obtener el grado de

Maestro en Ciencias
(Ciencias y Tecnologías de la Información)

P R E S E N T A

Ing. Moisés Quezada Naquid

Asesores:
Dr. Miguel López Guerrero
Dr. Ricardo Marcelín Jiménez

16 de Octubre de 2007

Resumen

La motivación de este proyecto de investigación es evaluar la utilización de componentes de bajo costo y amplia disponibilidad en la construcción de un sistema de almacenamiento distribuido tolerante a fallas. Los objetivos de este trabajo son: identificar, modelar y evaluar diversas decisiones de diseño de un sistema con las características descritas a continuación.

El sistema de almacenamiento distribuido considerado se encarga de almacenar los archivos enviados por clientes y recuperarlos cuando éstos los requieran. Este sistema consta de los siguientes elementos: una red de almacenamiento y un despachador. La red de almacenamiento consta de computadoras conectadas mediante una red local. Los usuarios se conectan a una computadora denominada despachador, la cual se encarga de atender las solicitudes de éstos. Los clientes pueden conectarse de forma local o remota.

Cada archivo a almacenar se divide en fragmentos a los que se les añade redundancia en información, generando archivos denominados dispersos. Cada archivo disperso se almacena en una computadora diferente. Con esto el sistema es capaz de tolerar fallas de paro en las computadoras de la red de almacenamiento. Asimismo, para la recuperación se realiza la operación inversa; a partir de los archivos dispersos se recupera el archivo original.

Para la arquitectura funcional de este sistema se plantearon dos modelos. En el primero, el despachador se encarga de procesar las peticiones recibidas y realizar las operaciones necesarias para almacenar los archivos o recuperarlos, según sea el caso. En el segundo, el despachador se encarga sólo de recibir las peticiones y elige, de forma aleatoria, una computadora que se encuentre disponible en la red de almacenamiento para realizar el procesamiento de

la solicitud.

Por otra parte se plantearon dos modelos para evaluar la confiabilidad del sistema. En estos modelos, cada una de las computadoras de la red de almacenamiento se considera como un nodo, el cual puede estar en estado activo realizando tareas de almacenamiento o recuperación o en estado de reserva para recuperar los contenidos de algún nodo en falla. En el primer modelo, la recuperación de contenidos se realiza de forma centralizada en un nodo de reserva. Este nodo se encarga de solicitar los dispersos necesarios a los demás y procesarlos para obtener los contenidos requeridos. En el segundo modelo se descentraliza la recuperación, delegando esta tarea en algunos de los nodos que participaban en el almacenamiento con el nodo en falla.

De los resultados obtenidos de las simulaciones para el primer modelo para evaluar el tiempo de respuesta, se puede observar que el despachador es el cuello de botella del sistema, debido a que el procesamiento de los archivos se lleva a cabo en un sólo punto. Este primer modelo se planteó para tener un punto de referencia sobre el cual se compara el segundo modelo planteado, donde se mejora el desempeño debido a la descentralización de las tareas de procesamiento sobre los nodos de la red de almacenamiento. Analizando los resultados obtenidos para el tiempo de respuesta se puede concluir que entre mayor sea el número de computadoras en la red de almacenamiento, mejor será el desempeño del sistema. Sin embargo, los resultados de la evaluación de la confiabilidad del Modelo de Confiabilidad Centralizado indican que al agregar computadoras al sistema el tiempo de vida media disminuye drásticamente, por lo tanto es mejor tener un número reducido de éstas para obtener un buen desempeño y un tiempo de vida media grande para los discos duros. Para evitar esta reducción, se propuso el Modelo de Confiabilidad Descentralizado, con el cual a pesar del aumento de la tasa de fallas en el sistema se obtiene un tiempo de vida del sistema igual o mejor que en el caso centralizado, debido a la reducción del tiempo de restauración de contenidos ya que se distribuye este proceso sobre los nodos activos del sistema.

Al finalizar el estudio de evaluación del sistema se identificó un compromiso entre sus parámetros. Por una parte al aumentar el número de máquinas en la red de almacenamiento, se obtiene un mejor desempeño en cuanto al tiempo de respuesta del sistema; sin embargo, esto representa un aumento de la tasa de fallas. Así, al distribuir el procesamiento de restauración de contenidos sobre las máquinas activas, provoca que el sistema aumente su tiempo de vida media y sea más estable. Además el hecho de aumentar el número de máquinas representa un incremento en el número de comités, así, cada máquina disminuye la cantidad de tiempo en la cual trabaja para almacenar archivos y por lo tanto se tiene un mayor balance de carga en el sistema.

Agradecimientos

Desde la infancia han estado a mi lado, apoyándome en cada paso que he dado. Este nuevo paso es uno de los más grandes que he dado y no lo hubiese hecho sin su ayuda. Gracias por estar ahí siempre festejando mis logros y dándome ánimo en las derrotas.

Estoy muy orgulloso de ustedes, son unos excelentes padres. Me han dado todo y lo único que han pedido a cambio es amor y comprensión. Por eso les dedico este triunfo que he logrado, que sin ustedes no hubiera logrado.

Los Amo

M. Q. N.

Contenido

Lista de acrónimos

SAN Red de Área de Almacenamiento, del inglés *Storage Area Network*

P2P Red entre pares, del inglés *Peer-to-Peer*

FCP Protocolo de Canal de Fibra, del inglés *Fibre Channel Protocol*

ISCSI Protocolo de almacenamiento para las redes de área de almacenamiento, del inglés *Internet Small Computer System Interface*

RAID Arreglo Redundante de Discos Independientes, del inglés *Redundant Array of Independent Disks*

IDA Algoritmo de Dispersión de Información, del inglés *Information Dispersal Algorithm*

WAN Red de Área de Amplia, del inglés *Wide Area Network*

LAN Red de Área Local, del inglés *Local Area Network*

SDL Lenguaje de Especificación y Descripción, del inglés *Specification and Description Language*

SAD Sistema de Almacenamiento Distribuido

PAC Política de Atención Centralizada

PAD Política de Atención Descentralizada

PA Protocolo de Almacenamiento

PR Protocolo de Recuperación

PA-PAC Protocolo de Almacenamiento para la Política de Atención Centralizada

PR-PAC Protocolo de Recuperación para la Política de Atención Centralizada

PA-PAD Protocolo de Almacenamiento para la Política de Atención Centralizada

PR-PAD Protocolo de Recuperación para la Política de Atención Descentralizada

MCC Modelo de Confiabilidad Centralizado

MCD Modelo de Confiablidad Descentralizado

N_f Nodo que presenta una falla de paro

N_r Nodo de reserva

mttf Tiempo de vida media, del inglés *Mean Time To Failure*

B Bytes

Lista de figuras

Lista de tablas

Lista de algoritmos

Capítulo 1
Introducción

Día a día las empresas generan una gran cantidad de información producto de sus operaciones, ésta debe ser almacenada ya que es de vital importancia para el funcionamiento de aquéllas. Por ejemplo, las bases de datos donde se registran los clientes y sus transacciones son muy importantes para un banco ya que son donde se almacenan todos los movimientos que realizan aquéllos y esto refleja el saldo de cada uno. Otro ejemplo es cualquier laboratorio de investigación, donde se generan una gran cantidad de datos que provienen de los resultados de experimentos asistidos por computadora. En estos ejemplos se habla de generación de datos, sin embargo el problema que conlleva esta actividad es que deben almacenarse de forma fiable y deben estar disponibles siempre que el usuario los requiera.

El problema del almacenamiento fiable mencionado anteriormente ha motivado la creación de diversas propuestas para resolverlo. Antes del surgimiento de las computadoras la forma tradicional de almacenar los datos era sobre papel y bastaba tener un espacio físico suficiente para almacenar los documentos generados. El principal problema se presentaba al realizar la búsqueda de la información para una consulta o para realizar alguna otra operación, lo cual podía tomar mucho tiempo. Por otra parte, en caso de una catástrofe, si no se contaba con un respaldo que estuviera a salvo simplemente la información era destruida e imposible de recuperar.

El surgimiento de las computadoras permitió a los usuarios aumentar la capacidad de almacenamiento de información en un menor espacio, ya que en un disco duro se puede

almacenar mayor cantidad de datos sin tener que ocupar un gran espacio físico, además el tiempo de consulta ó recuperación de información disminuyó considerablemente. Sin embargo, aún cuando se tienen las computadoras para almacenar información existe el problema del almacenamiento fiable, es decir, si toda la información se almacena en un sólo dispositivo (por ejemplo, discos duros o algún otro dispositivo de almacenamiento local) es muy probable que al presentarse una falla en éste, la información se pierda irremediablemente. Este hecho ha impulsado el desarrollo de nuevas técnicas para almacenar los datos de forma fiable, como lo son los sistemas basados en redes de computadoras [3]. Al contar con diversos dispositivos de almacenamiento en una red, la primer idea que surge en la mente es agruparlos para aumentar la capacidad de los sistemas. Esto da lugar a un nuevo problema y que es el de almacenar los datos de forma fiable. Así, surgen sistemas que se encargan de distribuir los datos sobre algunas o todas las computadoras disponibles dentro de una red y que, aún en presencia de fallas en sus componentes (dispositivos de almacenamiento), son capaces de preservar los contenidos almacenados por los usuarios y tenerlos disponibles para cuando éstos los requieran.

En el contexto del almacenamiento distribuido, este proyecto de investigación presenta la evaluación de la utilizaación componentes de bajo costo y amplia disponibilidad para construir un sistema de almacenamiento distribuido tolerante a fallas. Los objetivos del presente trabajo son: identificar, modelar y evaluar diversas decisiones de diseño de un sistema de almacenamiento distribuido.

Esta tesis se estructura como se describe a continuación. En el capítulo 2, se describe el estado del conocimiento de los sistemas de almacenamiento distribuido, además de la descripción de algunos temas necesarios para la comprensión de las propuestas planteadas. En el capítulo 3, se presentan los modelos de las operaciones del sistema de almacenamiento distribuido, así como cada uno de los protocolos para coordinar las tareas de almacenamiento y recuperación de los archivos. Además se describen los modelos para evaluar su confiabilidad.

En el capítulo 4, se describen los resultados obtenidos de la evaluación de desempeño del sistema en términos del tiempo de respuesta y confiabilidad. El estudio de evaluación de desempeño se llevó a cabo mediante una herramienta de simulación, en la cual se implementaron los modelos necesarios para el estudio. Por último, en el capítulo 5, se plantean las conclusiones obtenidas, así como las recomendaciones para el trabajo futuro.

Capítulo 2
Sistemas de almacenamiento distribuido

El problema de almacenar datos de forma eficiente ha sido objeto de estudio por varios años desde la aparición de las computadoras. Existen dos soluciones a este problema: emplear métodos basados en dispositivos locales o emplear métodos basados en redes de computadoras. Los métodos de almacenamiento basados en dispositivos locales permiten almacenar datos de forma centralizada y tienen una alta velocidad de acceso; pero la desventaja principal es que se tiene una baja capacidad de almacenamiento, acotada por los recursos locales. Por otro lado, con el auge de las redes de computadoras, se han diseñado sistemas basados en éstas que permiten almacenar los datos sobre un conjunto de computadoras logrando un aumento en la capacidad total de almacenamiento y la posibilidad de incrementar su confiabilidad. Sin embargo surgen nuevas problemáticas: la coordinación de los componentes del sistema, anchos de banda que limitan la transferencia de los archivos, posible aumento en el tiempo de respuesta del sistema y aumento en su complejidad, entre otros.

2.1. Métodos de almacenamiento

Los métodos tradicionales de almacenamiento se basan en el uso de dispositivos locales (discos duros, CD, DVD, HDVD). Si bien la capacidad de estos dispositivos se incrementa constantemente, nunca será suficiente, ya que día con día los usuarios requieren almacenar una

mayor cantidad de datos. La ventaja principal de estos métodos es la alta tasa de transferencia de los datos que reduce el tiempo de acceso para almacenarlos o recuperarlos; sin embargo existen problemas por los cuales se hace ineficiente su uso. Uno de estos problemas es la falla o daño de un dispositivo, en ese caso los datos almacenados se pierden y es casi imposible recuperarlos. Una solución a este problema es crear copias de respaldo, sin embargo en el caso de los discos duros representa la adquisición de varios de estos dispositivos lo cual no es práctico y a veces no es posible debido a que las computadoras sólo permiten añadir un número reducido de dispositivos. En el caso de los medios de almacenamiento removibles (medios magnéticos u ópticos), se reduce la capacidad de almacenamiento, pues comparados con los discos duros, sólo permiten almacenar una pequeña cantidad de datos de forma individual, así el usuario requiere un gran número de estos medios cuando necesita almacenar grandes volúmenes de datos y además sus respaldos. Otra desventaja de este tipo de medios es que tienen una tiempo de vida media relativamente corto, pues se dañan con facilidad provocando la pérdida de datos.

Actualmente han surgido nuevos dispositivos de almacenamiento extraibles tales como, las memorias tipo *flash*, entre otras. Estas memorias permiten almacenar datos de forma eficiente con una alta tasa de transferencia, además permiten agregar, actualizar o eliminar datos fácilmente. Estos dispositivos presentan una buena opción cuando la cantidad de datos no es muy grande y permiten a los usuarios guardar su información de forma confiable, sin embargo se debe tener cuidado en el manejo de los dispositivos, ya que su constante uso llega a dañarlos si no se tiene cuidado. Quizá el problema principal de los dispositivos extraibles no sea el cuidado que se debe tener, sino el hecho de que pueden perderse con facilidad debido a que son pequeños y con ello se aumenta la probabilidad de perder los datos almacenados.

Los problemas antes mencionados que presentan los métodos tradicionales y el auge de las redes, abren la posibilidad de crear nuevos métodos de almacenamiento. Estos métodos se basan en arquitecturas de redes de computadoras, dando lugar a sistemas de almacenamiento

distribuido. La principal ventaja de estos sistemas, con respecto a los métodos tradicionales es el aumento de la capacidad total del sistema debido a que reune un conjunto de computadoras compartiendo sus recursos de almacenamiento; sin embargo existe la posibilidad de un incremento en el tiempo de respuesta del sistema. Otra ventaja es que es posible aumentar su confiabilidad. Dos claros ejemplos de estos sistemas son las Redes de Área de Almacenamiento (SAN, por sus siglas en inglés *Storage Area Networks*) y las redes P2P (por sus siglas en inglés *Peer-to-Peer*).

Las SAN emplean tecnología de canal de fibra, para la cual la tasa de transferencia es muy alta y la probabilidad de error en las transmisiones es muy baja (del orden de 10^{-9}), dando así una alto grado de confiabilidad al sistema. Sin embargo, es necesario adquirir componentes diseñados para soportar este tipo de tecnología que no son tan accesibles para la mayoría de los usuarios ya que los costos se incrementan demasiado. Estas redes se comunican mediante diferentes protocolos, el más común es el protocolo de canal de fibra o FCP (de inglés *Protocol Fibre Channel*), aunque se han desarrollado nuevos protocolos tales como el protocolo ISCSI, que permite interconectar los dispositivos de almacenamiento mediante redes Ethernet.

Otra solución es el uso de las redes P2P, en las cuales diversos usuarios conectados de forma remota o local, comparten sus recursos de almacenamiento. Cada computadora conectada a una red de este tipo cumple funciones de cliente y también de servidor, así los usuarios puede acceder a los contenidos que requieran siempre y cuando el usuario servidor los comparta. El principal problema de estas redes, en cuanto a disponibilidad, es que dependen de que los usuarios se encuentren conectados o no, ya que si un usuario no está conectado es imposible acceder a los recursos que éste tenía compartidos. Además, como cada usuario es responsable de los contenidos que comparte, no existe un control sobre estos y puede suceder que un usuario con malas intenciones pueda compartir archivos infectados con virus u otro tipo de programas maliciosos, por esta razón esta solución es una forma insegura de almacenar información.

2.2. Computación fiable

Las redes de computadoras, permiten crear nuevos métodos de almacenamiento para los datos de los usuarios, sin embargo se tiene que garantizar el funcionamiento de estos nuevos métodos, es decir se deben crear sistemas fiables. La garantía de funcionamiento es la propiedad que permite a un usuario mantener la confianza en el servicio que se le ofrece. Esta garantía puede clasificarse de la siguiente manera [1]:

- **Disponibilidad (Availability)**. Es la capacidad del sistema para que los datos sean accesibles ante su petición por parte de un cliente.
- **Confiabilidad (Reliability)**. Es la capacidad de un sistema para funcionar en presencia de fallas durante un determinado tiempo.
- **Seguridad (Safety)**. Es la capacidad del sistema para seguir funcionando en presencia de una falla.
- **Capacidad de Mantenimiento (Maintainability)**. Es la capacidad del sistema para restaurarse en un determinado tiempo después de presentarse una falla.

2.2.1. Fallas y averías

Las fallas y las averías en los componentes de un sistema son un factor importante, ya que interrumpen el funcionamiento de éste evitando así el cumplimiento de algunas o todas las propiedades descritas anteriormente. Una falla es la desviación de un sistema sobre su funcionamiento normal. Por otra parte, una avería es la causa de un error en el funcionamiento interno de un sistema e indirectamente la causa de una falla. Las fallas se clasifican de acuerdo a diferentes modelos, los cuales describen sus causas. A continuación se dan a conocer los modelos de fallas más comunes [1]:

- **Paro**. Corresponde a la situación en la que un componente detiene su funcionamiento y permanece en este estado. Los demás componentes pueden detectar este tipo de fallas.

- **Caída**. Al igual que en la falla de paro, se detiene el funcionamiento, sin embargo los demás componentes no pueden detectar esta falla.

- **Caída y enlace**. Una falla de este tipo se presenta cuando un componente detiene su funcionamiento, permanece en ese estado y además un enlace falla perdiendo mensajes sin retrasar, corromper o duplicarlos.

- **Omisión de recepción**. Un componente incurre en este tipo de falla cuando recibe sólo un subconjunto de mensajes de los que se le enviaron.

- **Omisión de transmisión**. Esta falla ocurre cuando un componente envía un subconjunto de los mensajes que debería enviar.

- **Omisión general**. Este tipo de falla incluye a las dos anteriores, es decir, se detiene, permanece en ese estado o ambas.

- **Fallas Bizantinas**. Este tipo de fallas se definen cuando un componente presenta un comportamiento arbitrario.

  Un sistema tolerante a fallas debe ser capaz de detectar, corregir los errores o ambos, antes de que se produzcan las averías y el usuario detecte su existencia. El factor clave para lograr este objetivo es integrar redundancia en el sistema.

2.2.2. Redundancia

El principal objetivo de la redundancia es proveer al sistema de mecanismos para detectar y enmascarar los errores internos que se producen, evitando así que el usuario pueda observar

los efectos de las fallas. S. Mullender en [1] define la redundancia como: *recursos adicionales que no son necesarios en un sistema ideal*. Existen tres tipos de redundancia:

- **Redundancia de Recursos Físicos**. Este tipo de redundancia se refiere a la duplicación de componentes físicos en el sistema. Es decir, en el caso del almacenamiento distribuido, se deben tener duplicados de los datos, almacenados en diversos dispositivos (por ejemplo, discos duros). Así, si existe un error en alguno de los dispositivos, se puede obtener la información de alguna copia. Por ejemplo, un sistema basado en arreglos de discos duros RAID (Redundant Array of Independet Disks) en los cuales se distribuyen los datos, en otras palabras, en cada disco de un RAID se almacenan copias idénticas de los datos. La ventaja principal de este tipo de redundancia es que permite un acceso rápido para almacenar y recuperar los datos, sin embargo, representa un costo mayor debido a que se venden como una sola unidad con varios dispositivos internos.

- **Redundancia en Tiempo**. Este tipo de redundancia se refiere a la repetición de operaciones de cómputo o repetición de comunicaciones en el dominio del tiempo. Por ejemplo, cuando se transmite un archivo, si el receptor no notifica que se ha recibido en un determinado tiempo, el emisor debe enviarlo de nuevo hasta que se le notifique una recepción exitosa o hasta que se intente un número determinado de veces. La ventaja de emplear este tipo de redundancia consta en que se asegura el resultado que se obtendrá, aunque la desventaja es que aumenta el número de operaciones a ejecutar o aumentan las comunicaciones introduciendo retardos en el sistema.

- **Redundancia en Información**. Este tipo de redundancia se refiere a técnicas de codificación específicas. Por ejemplo, al agregar un bit de paridad a los datos que se envían a través de un canal se permite detectar errores. Podría pensarse que este tipo de redundancia no es viable debido al aumento en el tamaño de los datos; sin embargo

es la técnica más usada porque permite detectar y corregir errores.

Los tres tipos de redundancia mencionados se emplean en el diseño de sistemas tolerantes a fallas. Cada tipo se emplea en diferentes casos. La redundancia de recursos físicos se emplea para tolerar fallas permanentes tanto en software como en hardware. La redundancia en tiempo se emplea para tolerar fallas temporales y por último la redundancia en información se emplea para proteger la información relativa al estado de un sistema.

2.2.3. Dispersión de información

El almacenamiento y transmisión de datos en los sistemas distribuidos ha dado lugar a problemas importantes de seguridad y confiabilidad. Existen métodos que permiten resolver estos problemas, uno de ellos es la dispersión de información. El algoritmo de dispersión de información (IDA, del inglés *Information Dispersal Algorithm*) presentado por M. O. Rabin en [16], es un método eficiente que puede transformar un archivo de datos F en n partes llamadas dispersos que serán almacenados en diferentes localidades. El archivo original puede recuperarse a partir de m dispersos. Así cada disperso es de longitud igual a $\frac{|F|}{m}$ y el tamaño total de datos es $\frac{|F|}{m} * n$. Este algoritmo puede asociarse a la familia de los códigos de corrección de errores [2], en los cuales se agregan bits al mensaje creando bloques y el mensaje original puede recuperarse, en presencia de errores, a partir de un subconjunto de bloques. En la figura 2.1 se muestra un ejemplo de funcionamiento del IDA, con $n = 5$ y $m = 3$. Así un archivo se transforma en 5 dispersos y con cualesquiera 3 de ellos puede recuperarse el archivo original. Cabe señalar que el algoritmo presentado por M. O. Rabin es el algoritmo general de dispersión y se han desarrollado diversas propuestas con base en dicho algoritmo.

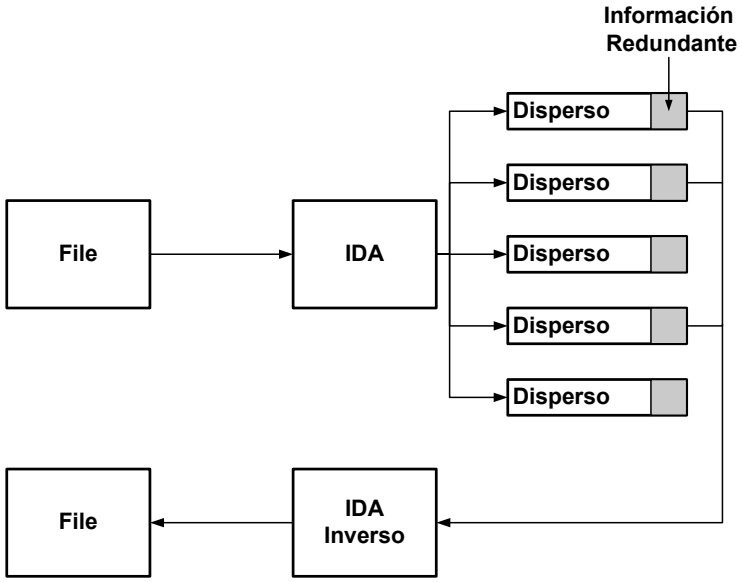

Figura 2.1: Algoritmo de Dispersión de Información (IDA).

2.3. Estrategias de selección y esquemas de almacenamiento

En un sistema de almacenamiento distribuido basado en la dispersión de información se deben tener en cuenta dos cosas para almacenar de forma fiable los datos. Lo primero que se debe considerar son los dispositivos sobre los cuales se almacenarán los datos. La forma en que se designan los lugares donde se almacenarán estos datos se denomina *estrategia de selección*. Lo segundo a considerar, una vez que se definieron los almacenes, es definir la forma en la que cada uno de los almacenes será empleado para cada tarea de almacenamiento, a lo cual se le denomina *esquema de almacenamiento*. A continuación se definen una estrategia de selección y un esquema de almacenamiento específicos para el propósito de este trabajo de tesis.

Una *estrategia de selección* B de V define la forma en cómo se crean subconjuntos sobre un

conjunto V de dispositivos de almacenamiento. Cada subconjunto b_i se denomina *comité* y es de tamaño b. En esta estrategia cada comité se genera a través de las combinaciones posibles de b elementos sobre los v elementos del conjunto V. Esta estrategia es *balanceada* ya que todos los comités tienen el mismo tamaño e *incompleta* ya que ningún elemento de V participa en todos los comités. En la tabla 2.2 se muestra un ejemplo para $|V| = 6$ y $b = 5$, es decir, se tienen 6 dispositivos de almacenamiento, cada comité es de tamaño igual a 5 y la "x" representa cuando un dispositivo pertenece al bloque.

Bloque \ Dispositivo	n_1	n_2	n_3	n_4	n_5	n_6
b_1	x	x	x	x	x	
b_2	x	x	x	x		x
b_3	x	x	x		x	x
b_4	x	x		x	x	x
b_5	x		x	x	x	x
b_6		x	x	x	x	x

Figura 2.2: Estrategia de Selección B de V.

Un esquema de almacenamiento es un proceso distribuido que se ejecuta por los sitios conectados a una red y se realiza por pasos. Para cada paso, se eligen los sitios que almacenarán datos. En otras palabras, al definirse cada bloque, mediante la estrategia B de V, el esquema de almacenamiento define cuál de los bloques participará en la siguiente tarea de almacenamiento. Al elegir una estrategia balanceada e incompleta en un esquema de almacenamiento, se garantizan dos cosas:

1. Todos los bloques almacenarán la misma cantidad de datos.

2. Ningún bloque participará en todas las tareas de almacenamiento.

El esquema de almacenamiento que se empleará en esta tesis usa la estrategia de selección "k de v", en la cual se designan como sitios de almacenamiento todos los comités posibles de

tamaño k del conjunto v. Así el número de comités que se pueden generar están dados por el coeficiente binomial $\binom{v}{k}$.

2.4. Sistemas de almacenamiento distribuido

En la actualidad han surgido nuevas propuestas de sistemas de almacenamiento distribuido, dando solución a algunas de las problemáticas mencionadas en la sección 2.1 (ver [3]). La mayoría de los sistemas existentes están diseñados para trabajar usando redes , los más relevantes para este trabajo se mencionan a continuación.

- **OceanStore**. El desarrollo de este sistema se encuentra a cargo de la universidad de Berkeley [11]. Este sistema está pensado para funcionar a escala global, es decir, que existan miles de servidores compartiendo sus recursos de almacenamiento para ofrecer a los usuarios una alta disponibilidad de los datos que deseen almacenar. Para garantizar la disponibilidad de la información, emplea códigos de corrección resistentes al borrado. Estos códigos transforman un archivo en m bloques tal que el archivo original puede ser recuperado con un subconjunto de éstos. Además cada bloque se almacena en varios servidores. La integridad de los archivos que se almacenan en el sistema se logra asignándoles identificadores mediante una función *hash* con el fin de que no se sustituyan los archivos por otros y se corrompa el archivo original. Para reconstruir el archivo de algún usuario, el sistema localiza a los servidores que tienen la información necesaria mediante *Tapestry*, que es una capa de localización y encaminamiento que trabaja sobre el protocolo TCP/IP y que se encarga de mapear un identificador de archivo a diferentes servidores. Para actualizar los datos emplea un algoritmo de administración de versiones, el cual genera una nueva versión cada vez que se realiza una actualización. La tolerancia a fallas del sistema se garantiza mediante un protocolo de acuerdo Bizantino. Actualmente sólo se encuentra implementado un prototipo de este

sistema, el cual se llama *POND* y es software libre.

- **PAST**. Este sistema, desarrollado por Microsoft Research, se diseñó como un sistema de escala global [4]. El almacenamiento de los archivos se realiza mediante duplicados de éstos sobre un gran número de sitios sobre Internet, garantizando la disponibilidad. A cada archivo se le asigna un identificador y una llave de descripción mediante los cuales un usuario puede recuperarlo. Un usuario puede compartir tanto el identificador como su llave de descripción a otro usuario. Para la localización de archivos se emplea un esquema de encaminamiento llamado *Pastry* que garantiza que la solicitud de recuperación será dirigida a los servidores correctos. La tolerancia a fallas de este sistema se basa en la cantidad de duplicados que se hayan almacenado, así para aumentar esta propiedad se necesita un mayor número de almacenes. La ventaja de este sistema es que presenta baja latencia y una alta disponibilidad. La desventaja es que no ofrece servicios de búsqueda por lo tanto si el identificador y la llave de descripción se pierden es imposible que el usuario recupere su archivo. Actualmente no se ha implementado este sistema.

- **Farsite**. Este sistema, desarrollado por Microsoft Research, basa su funcionamiento en la duplicación de archivos. Cada cliente contribuye con sus recursos de almacenamiento a cambio de un servicio de archivos confiable y de alta disponibilidad [7]. Los archivos son distribuidos entre todas las máquinas que contribuyen en el sistema, por lo tanto la administración no está centralizada. Antes de almacenar los archivos, cada uno de éstos es codificado, comprimido a tiempo de escritura y descomprimido a tiempo de lectura. La localización de cada archivo es almacenada en un directorio distribuido. A la fecha no se ha implementado totalmente este sistema.

- **LANStore**. Este sistema, desarrollado por Vilmos Bilicki, presenta una propuesta de almacenamiento altamente confiable y totalmente descentralizada. La característica

principal de este sistema es que se implementa sobre computadoras de escritorio utilizando su capacidad de almacenamiento. La característica de confiabilidad del sistema se logra mediante el uso de códigos de corrección de errores, mediante el algoritmo Reed-Solomon [8], mediante el cual los datos son divididos en bloques y se les agrega redundancia en información. Así el sistema es capaz de recuperar los datos originales a partir de los datos originales y los datos redundantes.

- **Bigtable**. Este sistema desarrollado en *Google Inc.*, permite el almacenamiento de estructuras de datos. El sistema está diseñado para manejar grandes volúmenes de datos (del orden de Petabytes) sobre miles de servidores. Bigtable se emplea para almacenar la información necesaria para los productos de Google, tales como, GoogleEarth, Google Finance, etc. los cuales requieren almacenar una gran cantidad de datos para su funcionamiento. El sistema contiene tres componentes básicos, una librería para cada cliente, un servidor maestro y varios servidores tableta [9]. Cabe señalar, que este sistema es software propietario y no puede adquirirse.

- **Celeste**. Propuesta realizada por Sun Microsystems. Su diseño se basa en el funcionamiento de las redes peer-to-peer (P2P) y en sistemas de almacenamiento orientado a objetos [10]. Este sistema provee almacenamiento de componentes de Cómputo de Utilidades Públicas. Otra característica importante es que el mantenimiento automático se lleva a cabo mediante políticas, las cuales permiten determinar el compromiso entre la capacidad, el funcionamiento, y la confiabilidad. Actualmente no se encuentra implementado este sistema, se encuentra en desarrollo.

- **RobuSTore**. Desarrollado en la universidad de California. RobuSTore emplea códigos de borrado para agregar redundancia y separa los datos codificados a través de una gran cantidad de discos. Este sistema reduce la dependencia del desempeño con respecto al comportamiento de discos individuales [12].

- **SAFE**. Sistema desarrollado en el Instituto de Tecnología de Georgia, Estados Unidos. Este sistema emplea códigos de corrección de errores para garantizar la tolerancia a fallas, y códigos de cifrado para garantizar la seguridad del sistema [13]. Actualmente sólo se ha implementado sobre la plataforma Linux a nivel biblioteca para usuario.

- **Cleversafe**. Este sistema fue desarrollado por la compañía Cleversafe Inc. [14]. Este sistema se basa en el Algoritmo de Dispersión de Información (IDA) de M. O. Rabin [16], creando su propio algoritmo llamado Cleversafe IDA, el cual divide un archivo en 11 fragmentos que son almacenados en diferentes servidores conectados a Internet. El sistema es capaz de recuperar el archivo original con 6 de los fragmentos. La idea principal de este proyecto es crear un sistema de almacenamiento global. Una ventaja de este sistema es que tiene licencia GPL, en otras palabras es software libre.

Los sistemas antes mencionados permiten a los usuarios almacenar sus datos de forma fiable. OceanStore y PAST se diseñaron para ser sistemas de escala global, así que no es óptimo para un usuario o empresa, donde se tengan pocas computadoras, almacenar su información mediante éstos. Farsite al emplear la duplicación de archivos requiere gran cantidad de espacio para almacenar los copias. LANStore presenta una buena solución de bajo costo; sin embargo la recuperación de los datos depende de los originales, este puede ser un problema si se pierden los datos originales es imposible reconstruir la información, además el espacio que emplea aumenta al tener los datos originales y los redundantes. El sistema Bigtable sólo se desarrolló para la empresa Google Inc. y no puede ser adquirido, por lo tanto no puede ser una solución para almacenamiento de los usuarios. La desventaja de CELESTE es que sólo se diseñó para el almacenamiento de componentes y no para archivos. RobuStore presenta la desventaja de que crea bloques a partir del archivo original pero genera un número mayor de bloques codificados, por lo tanto requiere mas espacio para el almacenamiento. Por otra parte el sistema SAFE presenta la desventaja de que requiere un gran número de servidores

para almacenar los datos codificados. El sistema Cleversafe presenta una alta disponibilidad y una alta confiabilidad; sin embargo se diseñó para redes WAN y por lo tanto no es una buena propuesta para usuarios que tienen necesidad de almacenar sólo dentro de una red LAN.

Una vez estudiados los sistemas más representativos en el contexto del almacenamiento distribuido se observó que la mayoría de éstos requieren un gran número de computadoras para almacenar de forma fiable los datos. En este trabajo de tesis, se evalúa el desempeño de un sistema que utiliza una cantidad reducida de almacenes y aun así permite almacenar los archivos de los usuarios de forma fiable, además de emplear tecnología de red de bajo costo.

Capítulo 3
Modelado de las operaciones de un sistema de almacenamiento distribuido

En el capítulo 2 se describió la problemática del almacenamiento de archivos, asimismo se realizó la revisión de los sistemas que se han propuesto. En este capítulo se describen los mecanismos para coordinar las tareas de almacenamiento y recuperación, así como los mecanismos para garantizar la tolerancia a fallas del sistema. Estos mecanismos se presentan mediante dos enfoques, tanto centralizado como distribuido. El caso del enfoque centralizado permite tener una referencia para comparar su desempeño con el enfoque distribuido.

El sistema que se evalúa en este trabajo se encuentra en desarrollo por otros miembros del área de investigación [15]. El sistema es capaz de almacenar archivos y recuperarlos cuando los clientes lo requieran. Los clientes pueden conectarse de forma remota o local al sistema y utilizar los servicios que éste ofrece. Para lograr este propósito la arquitectura del sistema consta de un conjunto de computadoras para almacenar los datos y un despachador de servicios. A continuación se da una descripción detallada del sistema.

El *Sistema de Almacenamiento Distribuido* (SAD) se encarga de almacenar archivos procesándolos mediante un algoritmo de dispersión (IDA, del inglés *Information Dispersal Algorithm*) [15, 16], el cual transforma un archivo en cinco partes llamadas *dispersos*. A cada disperso se le agrega redundancia en información para lograr una recuperación del archivo original aún cuando se pierdan a lo más dos de los dispersos. Así, la recuperación se lleva

a cabo ensamblando cualesquiera tres de los cinco dispersos generados (ver Fig. 3.1). Cada disperso debe almacenarse en una computadora diferente, esto con el fin de garantizar la tolerancia a fallas. La forma en que se eligen las cinco máquinas para almacenar se define mediante el esquema de almacenamiento "*k de v*", donde $k = 5$ y v es el número de máquinas en la red de almacenamiento, además la lista de comités que se obtiene a partir del esquema anterior debe generarse de forma estática al iniciar las operaciones del sistema.

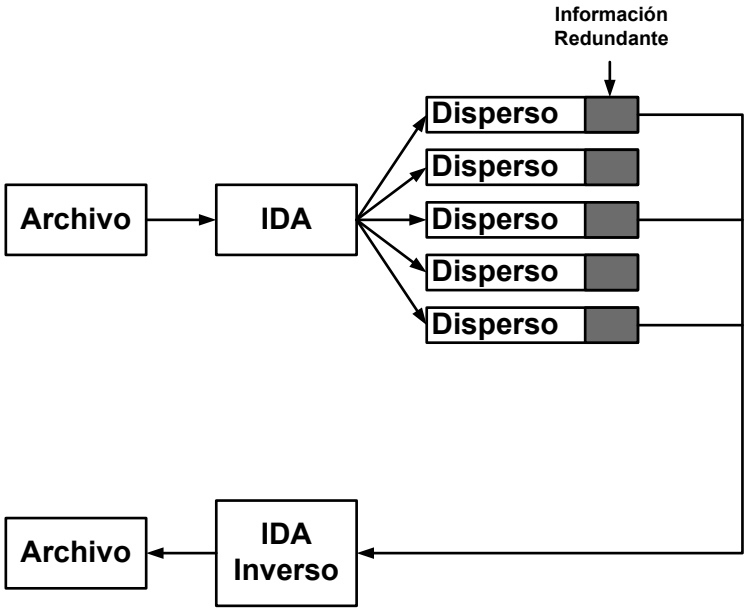

Figura 3.1: Algoritmo de dispersión con n=5 y m=3.

Para este sistema se denomina *cliente* a un usuario con necesidades de almacenar sus archivos. Un cliente envía su solicitud al sistema, la cual se recibe por un servidor que denominaremos *despachador*. Para llevar a cabo el almacenamiento se requieren dispositivos, que para el caso de este sistema serán computadoras de escritorio. Estas computadoras aportarán sus capacidades de almacenamiento (discos duros) para almacenar los dispersos generados en el procesamiento de los archivos enviados por los clientes.

En este contexto, el sistema se propuso con base en los requerimientos descritos anteriormente. Como se muestra en la figura 3.2, cada componente del Sistema de Almacenamiento

Distribuido realiza las funciones definidas a continuación:

- **Cliente**. Los usuarios se conectan de forma remota o local con el sistema de almacenamiento. Estos clientes generan solicitudes de servicio que pueden ser de dos tipos: almacenamiento o recuperación.

- **Despachador**. Este componente actúa como interfaz entre los clientes y la red de almacenamiento. Por otra parte, el despachador recibe y procesa las peticiones de servicio generadas por los clientes. Cuando recibe una petición de servicio, el despachador coordina a los demás componentes para realizar la operación solicitada. Asimismo, se encarga de realizar algunas tareas adicionales, dependiendo de la política de atención que se implemente en este componente.

- **Red de Almacenamiento**. Este componente está representado por una red local, en la cual existen v computadoras en servicio realizando tareas de almacenamiento y recuperación y s computadoras de reserva esperando a activarse en caso de presentarse una falla de alguna de las computadoras en servicio. Todas las computadoras se encuentran conectadas al despachador a través de un conmutador (switch).

El sistema de almacenamiento permite a los clientes conectarse, sin que éstos tengan acceso directo a las computadoras de la red de almacenamiento, evitando así que puedan acceder a los archivos almacenados, por lo tanto el despachador protege los datos de cada computadora fungiendo como un elemento de seguridad en el sistema, el cual cumple las funciones de un elemento que se conoce en informática con el nombre de *cortafuegos* . Cada archivo que los clientes envían o que se les envía, debe transferirse al despachador, para que éste lo reexpida a su destino final, ya sea a los clientes o a la red de almacenamiento (almacenamiento y recuperación respectivamente).

Para el modelado del SAD, se definieron dos políticas de atención para procesar las solicitudes de los clientes, así como cada uno de los protocolos para coordinar su procesamiento.

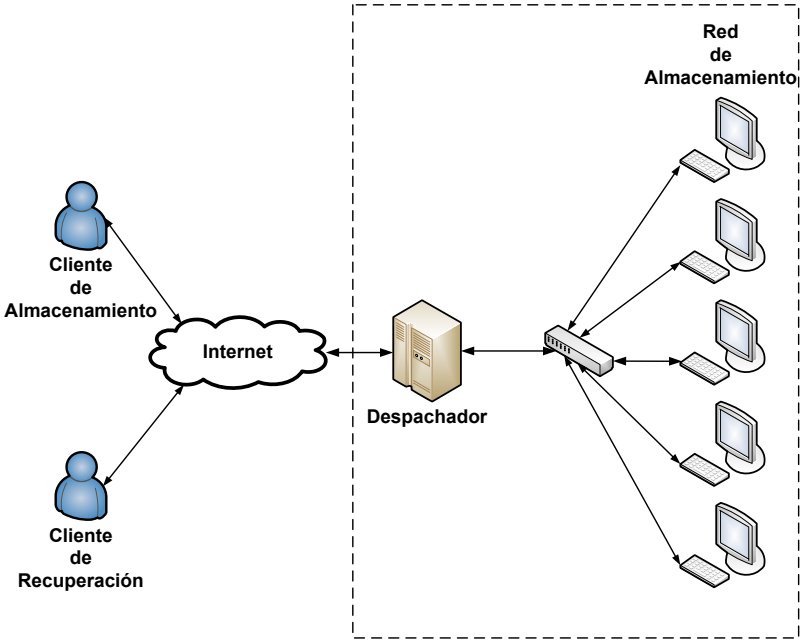

Figura 3.2: Sistema de Almacenamiento Distribuido.

Por otra parte, se plantearon dos modelos de confiabilidad para evaluar su tolerancia a fallas. En la primera política de atención, el despachador no sólo realiza la gestión de las solicitudes, también tiene a su cargo operaciones de procesamiento de archivos. El despachador procesa cada archivo para almacenarse a fin de obtener los dispersos o ensamblarlos para obtener el archivo original, según sea el caso de la operación solicitada. Este modelo se llama *Política de Atención Centralizada* (PAC). En el segundo modelo, llamado *Política de Atención Descentralizada*, el despachador delega las tareas de procesamiento a algunas computadoras de la red de almacenamiento. La secuencia de mensajes que describe las acciones realizadas, desde que el cliente envía su solicitud hasta que el sistema envía una confirmación al cliente indicándole que se ha almacenado su archivo, recibe el nombre de *Protocolo de Almacenamiento*. Por otra parte, la secuencia que describe las acciones realizadas cuando una solicitud de recuperación se recibe, se llama *Protocolo de Recuperación*. Estos dos protocolos se definen

para llevar a cabo las tareas necesarias para atender las solicitudes de los clientes, además son un mecanismo de coordinación, tanto interno, como externo, es decir, la coordinación Despachador-Red de Almacenamiento y la coordinación Cliente-Sistema. Los protocolos de comunicación, se propusieron con base en los requerimientos de cada una de las políticas para coordinar el almacenamiento y la recuperación de los archivos, enviados o solicitados por los clientes, respectivamente.

En este trabajo se evalúan las dos políticas de atención implementadas en el sistema, así como el modelo de confiabilidad. Para ambas políticas sólo se consideran las v computadoras en servicio de la red de almacenamiento, mientras que en el modelo de confiabilidad se consideran, además, las s computadoras de reserva. En este capítulo se describen las políticas de atención, los protocolos de la aplicación y los modelos de confiabilidad del sistema.

3.1. Política de Atención Centralizada (PAC)

Como ya se mencionó, la PAC es la primera política que se modeló para el sistema, que servirá de referencia en el estudio de evaluación de desempeño. En esta política, el procesamiento de las solicitudes de almacenamiento y recuperación se realiza de forma centralizada en el despachador. Es decir, cada solicitud enviada por los clientes se procesa directamente en el despachador, si el despachador se encuentra ocupado atendiendo una solicitud, las nuevas solicitudes que se reciban se forman en una fila de espera. En otro caso, si la fila de espera está vacía se procesa la solicitud inmediatamente. Cabe aclarar que, mientras una petición está siendo procesada con la política PAC implementada, el despachador no puede atender alguna otra sino hasta que se desocupe. Es decir, en el intervalo de tiempo comprendido entre el tiempo en que inicia el procesamiento de una solicitud y el tiempo en que el despachador envía una respuesta de finalización al cliente, sólo existe una solicitud que está siendo procesada en el sistema.

En el despachador se encuentra implementado el IDA, que se encarga de obtener los dispersos para almacenar el archivo del cliente. Asimismo, al aplicar el algoritmo inverso a los dispersos correspondientes se logra recuperar el archivo solicitado por el cliente. Además en el despachador se encuentra almacenada una base de datos donde se describe que máquinas han almacenado los dispersos correspondientes a cada archivo, cabe señalar que se asume que esta base de datos está disponible en cualquier momento que se requiera para realizar la consulta en el proceso de recuperación. En el contexto de la política PAC, el protocolo de almacenamiento recibe el nombre PA-PAC haciendo referencia a dicha política. Asimismo, el protocolo de recuperación recibe el nombre PR-PAC. A continuación se describen ambos protocolos.

3.1.1. Protocolo de Almacenamiento de la Política de Atención Centralizada (PA-PAC)

En la PAC las tareas de procesamiento se llevan a cabo en el despachador y las tareas de almacenamiento en la red de almacenamiento propiamente; sin embargo, el sistema debe coordinarse para llevarlas a cabo. El sistema realiza dichas tareas empleando el PA-PAC, mediante el cual se coordina cada paso del proceso de almacenamiento. Este protocolo se muestra en la figura 3.3 en el caso de que la atención sea exitosa y en la figura 3.4, el caso en el que no se tenga la capacidad suficiente para almacenar el archivo del cliente. Cada paso se describe a continuación.

1. El protocolo inicia cuando el cliente envía una solicitud al sistema mediante un mensaje StoreReq. En este mensaje se incluye el tamaño del archivo a almacenar.

2. Una vez recibida la solicitud de almacenamiento, el despachador regresa al cliente su número de atención con el mensaje Ticket. Si el despachador está desocupado, procede a procesar la solicitud. En caso de que se encuentre procesando otra solicitud, la nueva

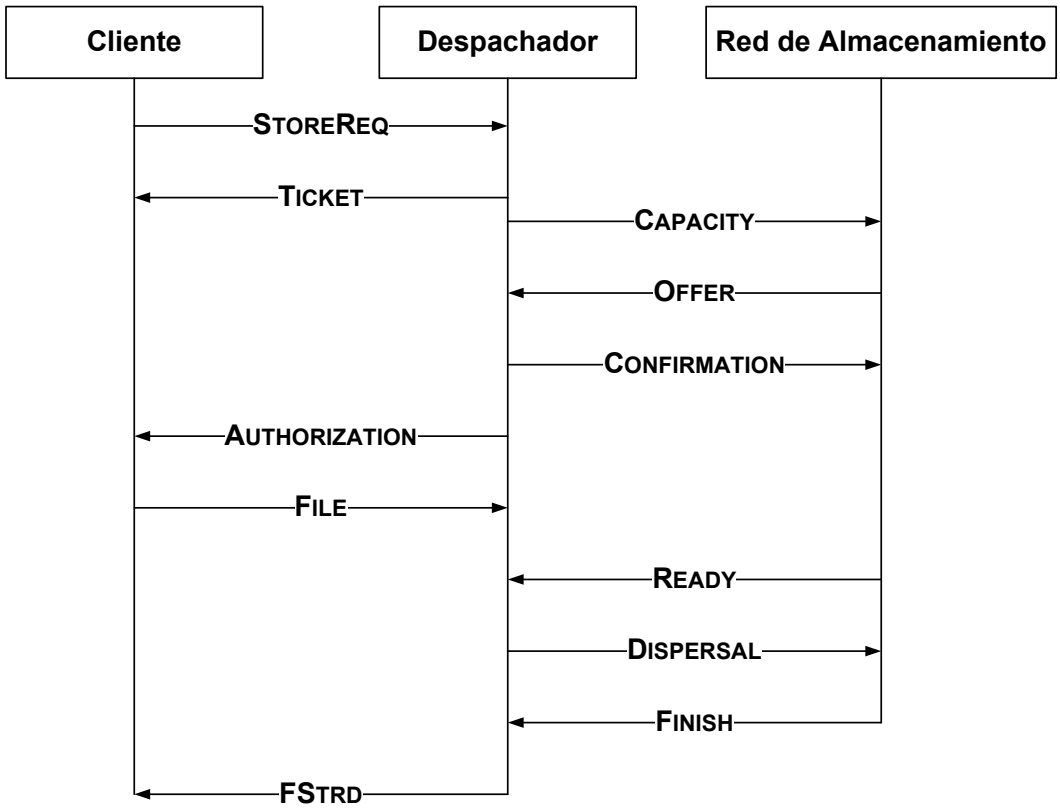

Figura 3.3: Protocolo de almacenamiento de la PAC (PA-PAC), caso con atención exitosa.

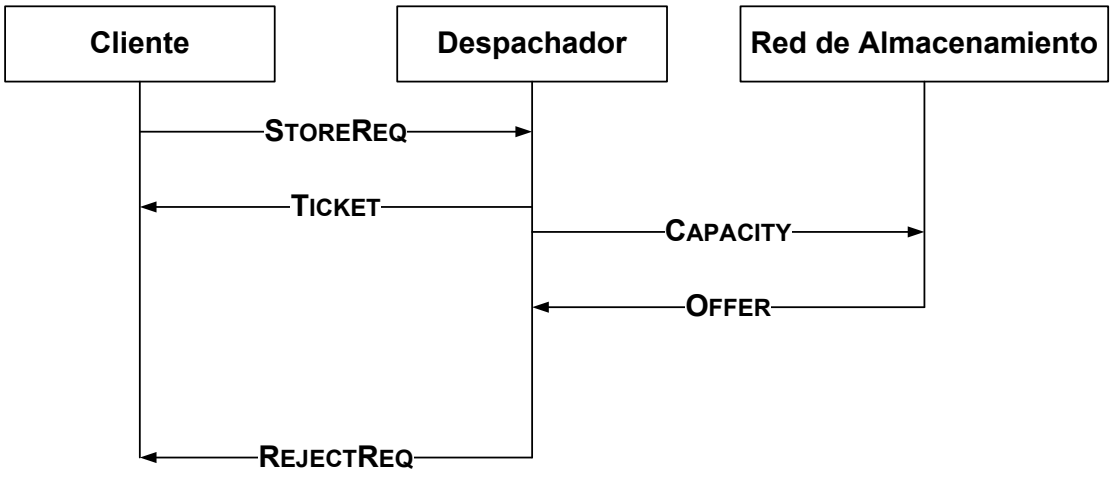

Figura 3.4: Protocolo de almacenamiento de la PAC (PA-PAC), caso con atención fallida.

solicitud se forma en una fila de espera hasta que llegue su turno para ser atendido. Este número de transacción es importante y el despachador debe gestionarlo, ya que cada mensaje que es enviado en el resto del protocolo hace referencia a este número para poder diferenciar entre cada una de las solicitudes.

3. El despachador debe asegurar que las máquinas del comité donde se almacenarán los dipersos tengan la capacidad suficiente para almacenarlos. Para este propósito se usa el mensaje CAPACITY que se envía a cada máquina que participa en el comité. En caso de no tener la capacidad suficiente, el despachador envía un mensaje de rechazo (REJECTREQ) al cliente.

4. Al recibir el mensaje de solicitud de capacidad, cada computadora envía mediante el mensaje OFFER la capacidad que tiene disponible, reservando el espacio solicitado mientras se confirma la operación. En caso de ser rechazada la oferta de una computadora, ésta libera el espacio reservado anteriormente, evitando así problemas de inconsistencia en el espacio libre del disco duro.

5. El número de máquinas necesario para almacenar un archivo es cinco, por lo tanto el despachador espera a que las cinco máquinas envíen una oferta y les confirma la operación (mensaje CONFIRMATION).

6. En el momento que una máquina elegida recibe el mensaje de confirmación, ésta envía un mensaje READY indicándole que se encuentra lista para recibir el disperso que le corresponde.

7. El despachador espera a que las máquinas del comité estén listas para recibir, esperando sus mensajes READY. Cuando se han recibido estos mensajes se envía un mensaje AUTHORIZATION al cliente, con el cual se le otorga el permiso de enviar su archivo. En este momento el despachador espera a que finalice la transferencia.

8. El cliente al recibir el mensaje de autorización envía el archivo que desea almacenar y espera a que su solicitud sea atendida.

9. Cuando el despachador recibe el archivo, lo procesa para obtener los dispersos correspondientes. Una vez que se obtienen los dispersos, el despachador envía cada disperso a su destino.

10. Al finalizar la tarea de almacenamiento de los dispersos, las computadoras envían un mensaje de finalización (FINISH).

11. El despachador, al recibir los cinco mensajes de finalización por parte de la red de almacenamiento, procede a informar al cliente que su archivo se ha almacenado, mediante el mensaje FSTRD. En este punto se libera el despachador, por lo tanto si existe alguna otra solicitud formada en la fila, procede a atenderla y realizar su correspondiente procesamiento. Si la fila de espera está vacía, el despachador espera a que una nueva solicitud sea enviada por algún cliente.

3.1.2. Protocolo de Recuperación de la Política de Atención Centralizada (PR-PAC)

En la PAC las tareas de recuperación se realizan por el despachador, sin embargo el sistema debe estar coordinado para evitar inconsistencias en la reconstrucción del archivo original. El sistema lleva a cabo dichas tareas empleando el protocolo PR-PAC, mediante el cual se coordina cada paso del proceso de recuperación de un archivo. El intercambio de mensajes para las tareas de recuperación se muestra en la figura 3.5 en el caso de una recuperación exitosa y en la figura 3.6, se muestra el caso en el que no se tengan los suficientes dispersos para recueprar el archivo original. Cada paso del protocolo se describe a continuación.

1. El cliente envía su solicitud de recuperación mediante el mensaje RETRIEVEREQ.

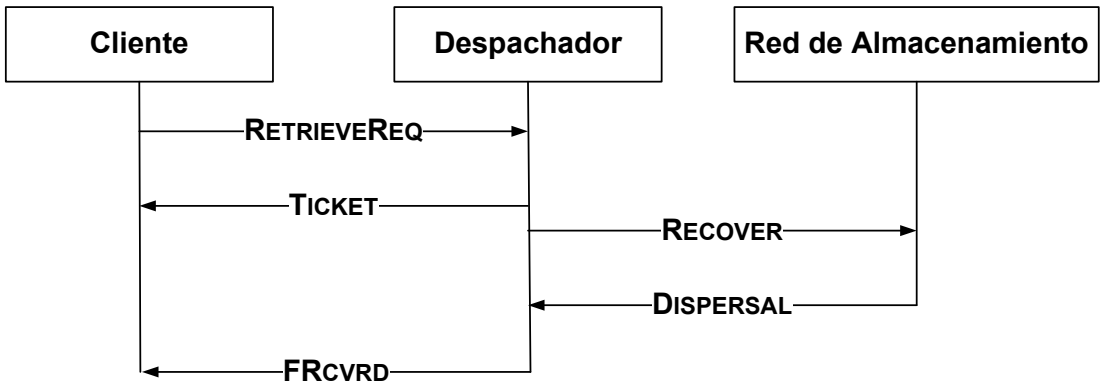

Figura 3.5: Protocolo de recuperación de la PAC (PR-PAC), con atención exitosa.

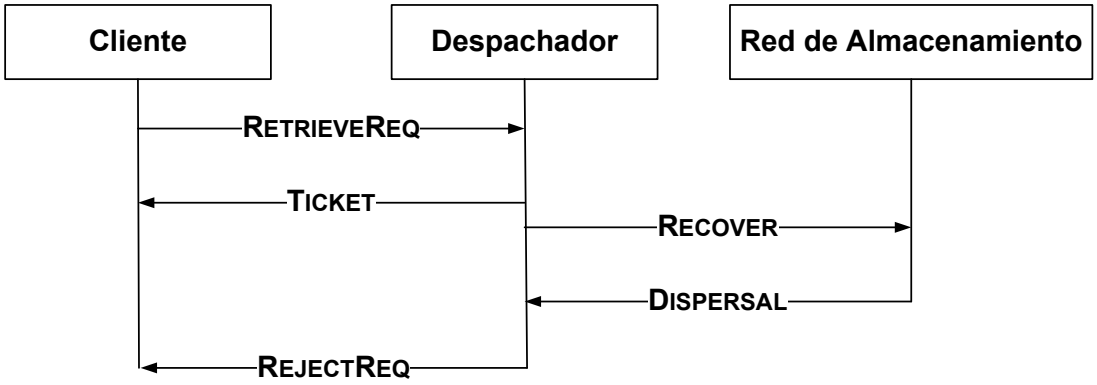

Figura 3.6: Protocolo de recuperación de la PAC (PR-PAC), con atención fallida.

2. Al recibir la solicitud de recuperación, el despachador envía un mensaje *Ticket* al cliente, indicándole el número de transacción que le fue asignado a su solicitud. Al igual que en el protocolo de almacenamiento PA-PAC, si el sistema se encuentra ocupado, la solicitud recién recibida se forma en la fila de espera. En caso contrario, se procede a atenderla de inmediato.

3. El despachador selecciona tres máquinas del comité que almacenadó los dispersos y les indica que deben enviar el disperso correspondiente al archivo solicitado. Este mensaje es de tipo RECOVER. En el caso de que no se tengan tres dispersos disponibles, el sistema debe enviar un mensaje REJECTREQ rechazando la solicitud.

4. Cuando una computadora de la red de almacenamiento recibe el mensaje *Recover*, envía al despachador el disperso en el mensaje *Dispersal*.

5. Al recibir los tres dispersos solicitados, el despachador se encarga de ensamblarlos y obtener el archivo original. Una vez realizada esta operación, el despachador envía el archivo original al cliente.

3.2. Política de Atención Descentralizada (PAD)

Para este segundo modelo se descentralizó el procesamiento de las solicitudes, haciendo que el despachador delegue las tareas a algunas de las computadoras de la red de almacenamiento. Es decir, cuando el despachador recibe una solicitud busca una computadora disponible de la red de almacenamiento y si encuentra una, reexpide la solicitud hacia esta máquina, a la cual se denomina *coordinador de procesamiento*. En caso de no encontrar una máquina disponible, la solicitud se forma en la fila de espera del despachador. El estado de cada una de las computadoras se almacena en el despachador mediante una lista, la cual indica la computadora y su estado, ya sea *libre* o bien *ocupada*, además el despachador almacena

una base de datos donde se asocia una máquina a cada disperso que se genere y al igual que en la PAC, se asume que la base de datos siempre está disponible para realizar las consultas necesarias. En la figura 3.7, se muestra el sistema con un coordinador de procesamiento.

En la PAD, en el despachador sólo existe una fila de espera. Cada una de las máquinas de la red de almacenamiento contendrá el IDA, ya que éstas son las encargadas de procesar las solicitudes de los clientes. En esta política de atención, el sistema puede atender hasta v solicitudes simultáneamente. Donde v es el número de máquinas en servicio en la red de almacenamiento. Esto se puede realizar, ya que cada computadora puede realizar las tareas de procesamiento de solicitudes cuando el despachador se lo indique.

Para llevar a cabo la PAD, se definieron nuevos protocolos de almacenamiento y de recuperación. Los nombres que reciben estos protocolos son *PA-PAD* y *PR-PAD* (almacenamiento y recuperación respectivamente). Estos protocolos se detallan a continuación.

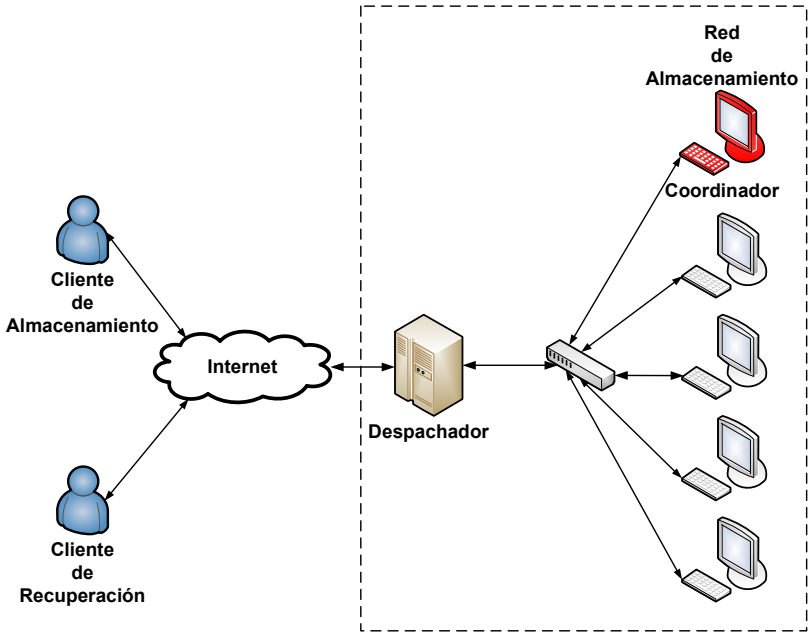

Figura 3.7: Sistema de almacenamiento distribuido con un coordinador de procesamiento.

3.2.1. Protocolo de Almacenamiento de la Política de Atención Descentralizada (PA-PAD)

Para la PAD se modificó el proceso de almacenamiento, ya que el despachador delega este procesamiento a las máquinas de la red de almacenamiento. En la figura 3.8 se muestra el intercambio de mensajes para el PA-PAD y cada paso se describe a continuación.

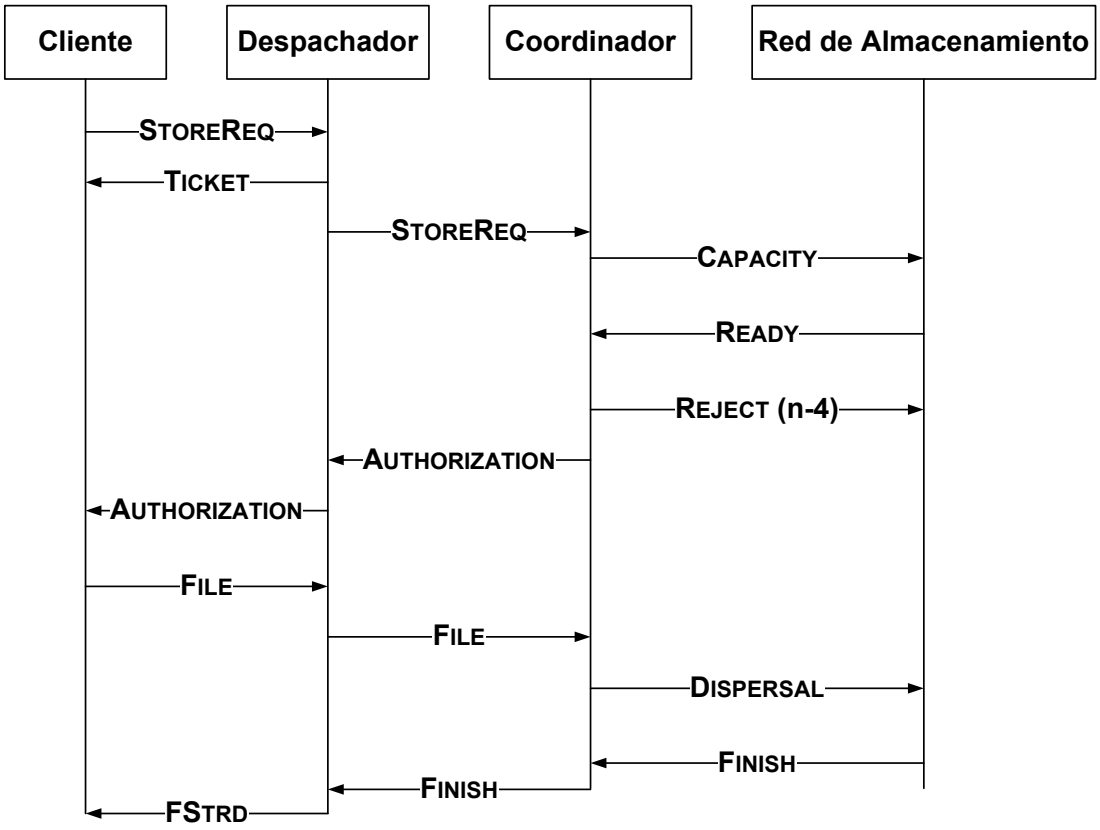

Figura 3.8: Protocolo de almacenamiento de la PAD (PA-PAD).

1. El protocolo se inicia cuando una solicitud de almacenamiento se envía por el cliente al despachador. Esta solicitud se envía mediante el mensaje StoreReq, en el cual se incluye el tamaño del archivo a almacenar.

2. Una vez recibida la solicitud, el despachador regresa al cliente su número de atención mediante el mensaje TICKET. Si alguna de las máquinas de la red está disponible para ser coordinador de procesamiento, el despachador reexpide la solicitud a ésta computadora. En caso contrario, la solicitud es formada en la fila de espera.

3. El coordinador de procesamiento, para procesar la solicitud de almacenamiento, debe preguntar a las máquinas del comité correspondiente si tienen la capacidad suficiente para almacenar un disperso (es decir, un tercio del tamaño del archivo original), enviándoles un mensaje CAPACITY.

4. Cada computadora que recibe la solicitud de capacidad, verifica su capacidad disponible en disco duro, reservando el espacio solicitado. Si la computadora tiene espacio suficiente, indica al despachador que está lista para recibir los dispersos con el mensaje READY. En caso contrario, envía el mensaje NORDY al despachador, indicando espacio en disco insuficiente.

5. El número de máquinas necesario para almacenar un archivo usualmente es cuatro (a diferencia de la política PAC, ya que el coordinador de procesamiento también almacena un disperso), por lo tanto, el coordinador de procesamiento espera a que a las cuatro máquinas restantes del comité envíen el mensaje READY para almacenar los dispersos. En caso de que el coordinador de procesamiento no tenga capacidad para almacenar un disperso, se deben elegir a cinco máquinas.

6. Una vez confirmada la capacidad de la red para almacenar los dispersos, el coordinador de procesamiento envía el mensaje AUTHORIZATION al despachador. El despachador se encarga de reexpedir este mensaje al cliente indicándole que inicie la transferencia del archivo. En este punto el despachador espera a que se realice dicha transferencia.

7. Cuando el cliente recibe la autorización envía su archivo al despachador y espera a que

se procese su solicitud.

8. Una vez recibido el archivo, el despachador lo retransmite a la máquina encargada de procesar la solicitud. Cuando el coordinador de procesamiento recibe el archivo, realiza el procesamiento correspondiente para obtener los dispersos. Una vez obtenidos los dispersos, se envían a sus destinos correspondientes.

9. Al terminar de almacenar un disperso, cada computadora de la red de almacenamiento que haya realizado esta operación, envía un mensaje FINISH al coordinador de procesamiento, indicando que el almacenamiento fue satisfactorio. Asimismo, dicha computadora, reexpide el mensaje de finalización al despachador.

10. El despachador, al recibir el mensaje de finalización, procede a informar al cliente que su archivo se ha almacenado, mediante el mensaje FSTRD. Con este mensaje se da por terminado el protocolo y el procesamiento de la solicitud de almacenamiento.

3.2.2. Protocolo de Recuperación de la Política de Atención Descentralizada (PR-PAD)

Para la PAD se modificó también el proceso de recuperación, ya que en ésta, el despachador delega este procesamiento a algunas máquinas de la red de almacenamiento. En la figura 3.9 se muestra el intercambio de mensajes del PR-PAD y cada paso se describe a continuación.

1. El protocolo inicia cuando el cliente envía una solicitud de recuperación al despachador. Esta solicitud se envía mediante el mensaje RETRIEVEREQ.

2. Una vez que recibe la solicitud, el despachador regresa al cliente su número de atención mediante el mensaje TICKET. Si alguna de las máquinas de la red está disponible

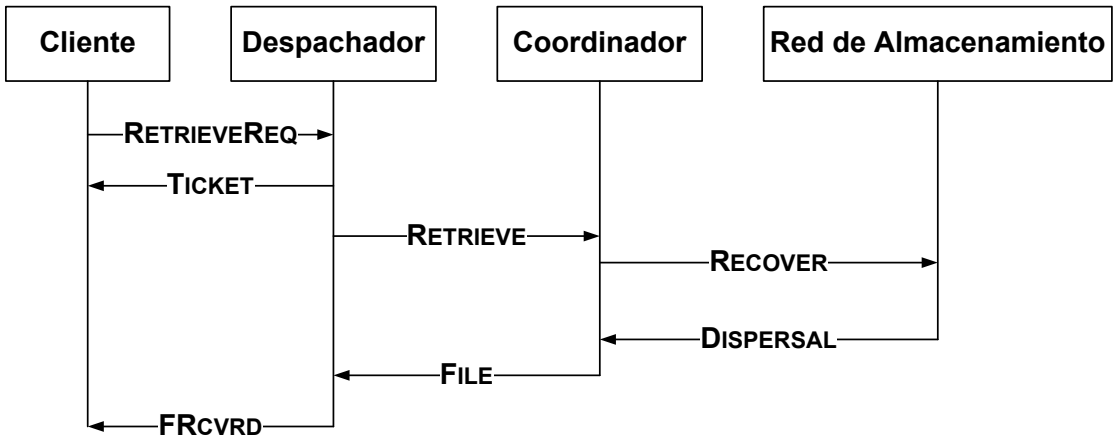

Figura 3.9: Protocolo de recuperación de la PAD (PR-PAD).

para ser coordinador de procesamiento, el despachador reexpide la solicitud a esta computadora. En caso contrario, la solicitud se forma en la fila de espera.

3. El coordinador de procesamiento al recibir el mensaje de recuperación debe seleccionar tres máquinas que tengan almacenado los dispersos del archivo. Una vez realizada esta operación, el coordinador de procesamiento envía el mensaje RECOVER a éstas, incluyendo el *Ticket* y espera la transferencia de los dispersos.

4. Cuando una computadora recibe el mensaje de solicitud de un disperso, transfiere al coordinador de procesamiento el disperso solicitado en el mensaje DISPERSAL.

5. Cuando el coordinador de procesamiento cuenta con los tres dispersos del archivo del cliente, realiza el ensamble de dichos dispersos, recuperando así el archivo original.

6. Cuando el despachador recibe el archivo, lo retransmite al cliente acompañado de un mensaje de finalización (mensaje FRCVRD), dando por terminada la transacción.

Es así como quedan descritas cada una de las políticas de servicio y sus protocolos correspondientes para almacenar y recuperar archivos. En la siguiente sección se describen los modelos de confiabilidad de planteados para el sistema.

3.3. Tolerancia a fallas

Una vez propuesto el sistema, se planteó un modelo de confiabilidad para evaluar su tolerancia a fallas. Este modelo permite estudiar la capacidad del sistema para recuperarse cuando existen fallas de paro en las computadoras de la red de almacenamiento. El modelo de confiabilidad del Sistema de Almacenamiento Distribuido se constituye por los siguientes elementos:

- **Nodos**. Cada una de las computadoras de la red de almacenamiento se considera como un nodo. Cada uno de estos componentes se considera como activo si se encuentra realizando tareas de almacenamiento o de recuperación, así como también si se encuentra realizando el procesamiento de una solicitud (cada caso depende de la política de atención que se implemente). Un nodo en estado de reserva se encuentra en estado ocioso, es decir, no participa en las tareas antes mencionadas y se encuentra en espera para restaurar los contenidos de algún otro nodo que presente una falla.

- **SuperNodo**. Este componente representa un elemento teórico para realizar las funciones de monitor para el sistema, es decir, este elemento se encarga de la detección de fallas o también para designar a un nodo de reserva para realizar la restauración de contenidos.

En el modelo para la evaluación de la confiabilidad del sistema se toma en cuenta el balance de carga entre sus nodos. Dicho balance se lleva a cabo mediante la creación de comités basándose en la estrategia "Selección k de v" descrito en el capítulo 2.

Para evaluar la confiabilidad del sistema se plantearon dos modelos, el *Modelo de Confiabilidad Centralizado* y el *Modelo de Confiabilidad Descentralizado*. En el primer modelo el nodo de reserva (N_r), encargado de restaurar los contenidos del nodo en falla (N_f), realiza esta tarea por completo. En la segunda, N_r elige un nodo coordinador de restauración de

cada comité en los que participaba N_f y delega la tarea de recuperación a estos coordinadores de restauración. Para ambos modelos se tienen las siguientes consideraciones:

- Un *SuperNodo* encargado de la supervisión del sistema.
- v nodos activos.
- s nodos de reserva.
- k nodos por comité.
- Cada nodo participa en $\binom{v-1}{k-1}$ comités (generado de los posibles subconjuntos donde aparece un nodo, ver Apéndice A).
- Con esta consideración se garantiza que todos los nodos almacenarán la misma cantidad de dispersos.
- El mínimo número de nodos activos que garantiza la operación de un comité es $m = 3$, ya que en el ensamble de dispersos sólo se requieren tres.
- El sistema interrumpe sus operaciones si un comité deja de operar o si no existe algún nodo de reserva para restaurar contenidos.
- Cada nodo tiene asociado un tiempo de vida media. Este tiempo se describe mediante una variable aleatoria por nodo, siendo estas variables independientes e idénticamente distribuidas.
- Cada nodo tiene asociado un tiempo de reparación. Este tiempo se describe mediante una variable aleatoria. Estas variables son independientes e idénticamente distribuidas.
- Las fallas en los nodos se consideran "*fallas de paro*".

- Cuando ocurre una falla en el sistema, éste debe detener las operaciones de almacenamiento y recuperación de archivos para proceder a restaurar los contenidos de los nodos que presenten una falla.

En las siguientes subsecciones se describen los dos modelos planteados para el sistema así como las máquinas de estados que describen el funcionamiento para cada uno de éstos.

3.3.1. Modelo de Confiabilidad Centralizado (MCC)

Como ya se mencionó, el primer modelo para evaluar la confiabilidad del sistema es el Modelo de Confiabilidad Centralizado. En este modelo las tareas de restauración de contenidos se llevan a cabo en los nodos de reserva designados por el SuperNodo. Tres nodos de cada comité en los que participaba un nodo en falla envían al nodo elegido los dispersos correspondientes y éste se encarga de procesarlos para obtener los contenidos del nodo al cual restaura.

Al presentarse una falla de paro en algún nodo activo del sistema se realizan una serie de pasos para llevar a cabo la restauración. Esta serie de pasos se describe a continuación.

1. Cuando un nodo falla el SuperNodo lo detecta. Esto se realiza, para asegurar que los $v-1$ nodos activos restantes detecten también que el nodo falló. Inmediatamente, el nodo en falla (N_f) debe ser enviado a reparación y cuando finalice este proceso el nodo se integra al sistema como nodo de reserva. A su vez, el SuperNodo debe elegir uno de los s nodos de reserva para iniciar la restauración e indicarle en qué comités participó N_f.

2. En caso de no existir algún nodo de reserva para realizar la restauración, se considera un colapso en el sistema, por lo tanto, el sistema detiene sus operaciones.

3. Cuando un nodo en estado de reserva recibe la indicación para restauración, para cada comité donde participó N_f debe solicitar a tres nodos activos los dispersos relacionados

a N_f. Cada nodo sabe en qué comité participó con N_f, por lo tanto sabe qué dispersos debe enviar. En caso de que en algún comité no existan al menos 3 nodos activos, el nodo en restauración (N_r) debe indicar al SuperNodo que hubo una FALLA FATAL en el sistema y éste a su vez debe indicar el colapso a cada uno de los nodos activos, por lo tanto, se detiene el funcionamiento del sistema.

4. Cuando un nodo activo recibe la solicitud de dispersos por parte de N_r debe enviar los dispersos correspondientes a dicho nodo.

5. Una vez restaurados los dispersos que había almacenado N_f, N_r debe indicarle al SuperNodo que está listo para integrarse como un nodo en servicio.

6. En el caso de que N_r falle durante la restauración de contenidos, el SuperNodo debe elegir otro nodo de reserva para reiniciar la restauración.

Para describir los pasos anteriores, las máquinas de estados 1 y 2 representan el funcionamiento del SuperNodo y de un nodo, respectivamente. Para dar una mejor idea acerca del funcionamiento de ambas máquinas, en la tabla 3.1 se describe la lista de variables, funciones y mensajes que emplean.

Tabla 3.1: Variables, Funciones y Mensajes del MCC.

Variables
tiempo_de_vida, es el tiempo de falla de un nodo.
regreso_a_trabajar, es el tiempo para reparar un nodo.
respuesta, es el tiempo para transmitir un disperso.
restauracion, es el tiempo para reconstruir un disperso
respuestas, cuenta el número de nodos de un comité que han sido interrogados.
Continúa ...

Tabla 3.1: Variables, Funciones y Mensajes del MCC (continuación).

activos, cuenta el número de nodos en servicio en un comité.
disponible, cuenta el número de dispersos recibidos.
Sobrevivientes, el conjunto de nodos activos de un comité determinado.
Activos, el conjunto de todos los nodos activos.
Reservas, el conjunto de todos los nodos de reserva.
$Nodos$, $Activos \cup Reservas$.
$Esquema(j)$, los comités donde j participa.
Comite, un subconjunto de *Activos*.
Funciones
$f()$, número exponencial pseudoaleatorio, modela el tiempo de falla.
$r()$, número exponencial pseudoaleatorio, modela el tiempo de reparación.
contenidos(), tiempo para transmitir una cantidad de información.
ida_inverso(), tiempo que toma la transformación del IDA inverso.
cliente(), el nodo caído que un nodo de reserva está restaurando.
Mensajes
INIT, el nodo calcula su *tiempo_de_vida*.
STATE, pregunta si un nodo está activo o caído.
CRASH, el nodo presenta una falla.
ACTIVE, el nodo está activo.
INFO, solicitud de los dispersos necesarios.
DISPERSAL, envío de los dispersos que se solicitan.
Continúa ...

Tabla 3.1: Variables, Funciones y Mensajes del MCC (continuación).

SUCCESS, restauración exitosa de un nodo caído.
TEAM(T), el nodo caído trabajaba en el comité T.
NEXT_TEAM, solicitud del siguiente comité en el que el nodo caído trabajaba.
FATAL, mensaje de falta de *activos* para completar la restauración.
COLLAPSE, detiene el funcionamiento, el sistema se colapsa.
REPAIR, inicia la restauración del nodo caído.
BACK, busca un nodo de reserva

Algoritmo 1: *Máquina de estados del SuperNodo del MCC*

```
1  estado S0 /*el usuario inicia el funcionamiento del sistema
2  Al recibir STARTS del usuario
3     ∀ k ∈ Activos, envia(INIT) a k, ahora
4     ∀ j ∈ Reservas, envia(INIT_SPARE) a j, ahora /* Mensaje para poner en reserva s nodos
5     estado ← S1

6  estado S1 /* operación cotidiana
7  Al recibir BACK de j
8     envia(INIT_SPARE) a j, ahora
9     Reservas = Reservas ∪ {j}

10 Al recibir SUCCESS de j
11    Activos = Activos ∪ {j}

12 Al recibir FATAL de j
13    ∀ k ∈ Nodos, envia(COLLAPSE) a k, ahora
14    deten_funcionamiento
15    estado ← S2

16 Al recibir CRASH de j
17    si j ∈ Activos entonces
18          Activos = Activos \ {j}
19    otro
20          j = cliente(j)
21    fin
22    Sea Esquema(j) una copia del conjunto de comités donde j aparece
23    si  ∃ s ∈ Reservas entonces
24          si ∃ T ∈ Esquema(j) entonces
25                envia(TEAM(T)) a s, ahora
26                Esquema(j) = Esquema(j) \ {T}
27          otro
28                ∀ k ∈ Nodos, envia(COLLAPSE) a k, ahora
29                detener funcionamiento
30                estado ← S2
31          fin
32    fin

33 Al recibir NEXT_TEAM de j
34    k = cliente(j)
35    si ∃ T ∈ Esquema(k) entonces
36          envia(TEAM(T)) to j, ahora
37          Esquema(k) = Esquema(k) \ {T}
38    otro
39          envia(TEAM(∅)) a j, ahora
40    fin

41 estado S2 /* nada se puede hacer ahora
```

Algoritmo 2: *Máquina de estados para un nodo del MCC.*

```
1  estado N0/N2 /* Activo/Reserva aún no se ha calculado su tiempo_de_vida
2  Al recibir INIT de SuperNodo
3    tiempo_de_vida = calcula tiempo aleatorio de acuerdo a f()
4    envia(CRASH) a mí, al tiempo tiempo_de_vida
5    estado ← N1/N3

6  estado N1 /* un nodo activo coopera para restaurar a un comité
7  Al recibir STATE de s
8    respuesta = calcula de acuerdo a contenidos()
9    si respuesta > tiempo_de_vida entonces
10       envia(CRASH) a s, ahora
11   otro
12       envia(ACTIVE) a s, ahora
13   fin

14 Al recibir INFO de s
15   envia(DISPERSAL) a s, al tiempo respuesta

16 estado N3 /* un nodo de reserva consulta un comité
17 Al recibir TEAM(T) de SuperNodo
18   si T == ∅ entonces
19       tiempo_de_vida = calcula tiempo aleatorio de acuerdo a f()
20       envia(CRASH) a mí, al tiempo tiempo_de_vida
21       envia(SUCCSESS) a SuperNodo, ahora
22       estado ← N1
23   otro
24       ∀ k ∈ T, envia(STATE) a k, ahora
25       respuestas = 0
26       activos = 0
27       Sobrevivientes = ∅
28       estado ← N4
29   fin

30 estado N4 /* esperando por el estado de los participantes de un comité
31 Al recibir ACTIVE /CRASH de j
32   respuestas++
33   si msg==ACTIVE entonces
34       activos++
35       Sobrevivientes = Sobrevivientes ∪ {j}
36   fin
37   si respuestas==5 entonces
38       si activos ≥ 3 entonces
39           ∀ k ∈ Sobrevivientes, envia(INFO) a k, ahora
40           disponible = 0
41           estado ← N5
42       otro
43           envia(FATAL) a SuperNodo, ahora
44       fin
45   fin

46 estado N5 /* esperando por los dispersos
47 Al recibir DISPERSAL de j
48   disponible++
49   si (disponible ≥ 3) entonces
50       tiempo_de_restauración = calcula de acuerdo a ida_inverso()
51       envia(NEXT_TEAM) a SuperNodo, al tiempo_de_restauración
52       estado ← N3
53   fin

54 estado N6 /* abandonar el centro de reparación
55 Al recibir REPAIR de mí
56   envia(BACK) a SuperNodo, ahora
57   estado ← N2

58 para cualquier estado /* puede ocurrir en cualquier momento
59 Al recibir CRASH de mí
60   regreso_a_trabajar = calcula de acuerdo a r()
61   envia(CRASH) a SuperNodo, ahora
62   envia(REPAIR) a mí, al instante regreso_a_trabajar
63   estado ← N6

64 Al recibir COLLAPSE de SuperNodo
65   detener_funcionamiento
66   estado ← N7

67 estado N7 /* nada se puede hacer ahora
```

3.3.2. Modelo de Confiabilidad Descentralizado (MCD)

En este segundo modelo, el sistema acelera el proceso de restauración al distribuir el trabajo entre algunos de los nodos que lo conforman. La idea del Modelo de Confiabilidad Descentralizado es que el SuperNodo elija un coordinador de restauración para cada comité. Cada coordinador es elegido del conjunto de nodos activos.

Al presentarse una falla de paro en algún nodo activo del sistema se realizan una serie de pasos para llevar a cabo la recuperación. Esta serie de pasos se describe a continuación.

1. Cuando un nodo falla el SuperNodo lo detecta. Esto se realiza, para asegurar que los $v - 1$ nodos activos restantes detecten también que el nodo falló. Inmediatamente, el nodo en falla (N_f) debe ser enviado a reparación y cuando finalice este proceso el nodo se integra al sistema como nodo de reserva.

2. Para iniciar la recuperación el SuperNodo elige un coordinador de restauración, del conjunto de nodos activos para que se encargue de restaurar los contenidos de un comité donde participaba N_f. En esta selección se debe tener en cuenta que un coordinador de restauración sólo puede restaurar un comité, por lo tanto debe elegirse uno diferente para cada uno de ellos. Así, todos los nodos activos participarán en la restauración del nodo en falla. En caso de que el número de comités sea mayor que el número de nodos activos, se deben encolar los comités pendientes y cuando un nodo activo esté disponible se le enviará el siguiente comité; así sucesivamente hasta finalizar con los comités pendientes.

3. En caso de no existir algún nodo activo para fungir como coordinador de restauración en algún comité, se considera que ocurre el colapso del sistema, por lo tanto se detiene su operación.

4. Un nodo coordinador de restauración debe solicitar a dos o tres nodos activos, depen-

diendo si él forma parte del comité que recupera o no, los dispersos relacionados con N_f. Cada nodo sabe en qué comité participó con N_f, por lo tanto sabe qué dispersos debe enviar. En caso de que en algún comité no existan al menos 2 nodos activos, el coordinador de restauración debe indicar a N_r que hubo una falla fatal y éste a su vez debe indicarlo al SuperNodo. Cuando recibe esta indicación, el SuperNodo debe comunicar el colapso a cada uno de los nodos activos y se detiene el funcionamiento del sistema.

5. El SuperNodo debe preguntar a los nodos si sobrevivirán el tiempo suficiente para recuperar el comité que se le asigne. En caso de disponer del tiempo suficiente, el nodo debe enviar una confirmación y en caso contrario un rechazo. Esto se plantea para asegurar que existan tres nodos disponibles para recuperar los contenidos y a modo de simplificar el modelo de simulación.

6. Cuando un nodo activo recibe una solicitud de envío dispersos por parte de su coordinador de restauración debe enviarle los dispersos correspondientes para que éste los procese y obtenga los contenidos. Al finalizar esta operación el coordinador de restauración debe enviar al SuperNodo una solicitud, preguntándole a qué nodo de reserva enviará los datos. Aquel debe elegir uno si existe, en caso contrario el sistema se colapsa.

7. Una vez restaurados los contenidos de N_f, N_r debe indicarle al SuperNodo que está listo para integrarse como un nodo en servicio.

Para describir los pasos anteriores, las máquinas de estados 3, 4 y 5, representan el funcionamiento del SuperNodo y de un nodo ordinario, respectivamente. Para dar una mejor idea acerca del funcionamiento de ambas máquinas, en la tabla 3.2 se describe la lista de variables, funciones y mensajes que emplean.

Tabla 3.2: Variables, funciones y mensajes del MCD.

Variables
tiempo_de_vida, es el tiempo de falla de un nodo.
tiempo_restauracion, es el tiempo de restauración de un comité.
respuestas, es el número de respuestas que se han recibido.
activos, es el número de nodos activos que han respondido la solicitud de estado.
dispersos, es el número de dispersos que se han recibido.
max_dipsersos, es el número de activos que se espera para recuperar un comité.
max_respuestas, es el número de respuestas que se espera para recuperar un comité.
$Nodo_f$, es el nodo que falló.
respuesta, es el tiempo para transmitir un disperso.
t_transmision, es el tiempo que toma transmitir un disperso.
tiempo_de_restauracion, es el tiempo que le toma al IDA inverso recuperar un disperso.
disponible, cuenta el número de dispersos recibidos.
L, es una fila de espera para los comités que se recuperarán.
t_envio_contenidos, es el tiempo que le toma a un nodo enviar los contenidos a una reserva.
regreso_a_trabajar, es el tiempo para reparar un nodo.
Sobrevivientes, el conjunto de nodos activos de un comité determinado.
Activos, el conjunto de todos los nodos activos.
Reservas, el conjunto de todos los nodos de reserva.
Nodos, $Activos \cup Reservas$.
$Esquema(j)$, los comités donde j participa.
Coordinadores, conjunto de nodos designados para ser coordinadores.
Continúa ...

Tabla 3.2: Variables, funciones y mensajes del MCD (continuación).

Funciones
sig_comite, siguiente comité para atender.
$f()$, número exponencial pseudoaleatorio, modela el tiempo de falla.
tiempo_de_restauracion, tiempo para restaurar un comité.
es_mi_comite, indica si un nodo pertenece al comité al que recupera.
transmision, es el tiempo para enviar los dispersos o contenidos referentes a $Nodo_f$.
$r()$, número exponencial pseudoaleatorio, modela el tiempo de reparación.
contenidos(), tiempo para transmitir una cantidad de información.
ida_inverso(), tiempo que toma la transformación del IDA inverso.
cliente(), el nodo caído que un nodo de reserva está restaurando.
coordinador(T), elige a un coordinador del comité T.
longitud(L), indica el tamaño del conjunto L.
Mensajes
STARTS, el usuario inicia la operaciones del sistema.
INIT, el nodo activo calcula su *tiempo_de_vida*.
INIT_SPARE, el nodo activo calcula su *tiempo_de_vida*.
STATE, pregunta si un nodo está activo o caído.
CRASH, el nodo presenta una falla.
ACTIVE, el nodo está activo.
INFO, solicitud de los dispersos necesarios.
Continúa ...

Tabla 3.2: Variables, funciones y mensajes del MCD (continuación).

DISPERSAL, envío de los dispersos que se solicitan.
SUCCESS, restauración exitosa de un nodo caído.
TEAM(T), el nodo caído trabajaba en el comité T.
FATAL, mensaje de falta de *activos* para completar la recuperación.
COLLAPSE, detiene el funcionamiento, el sistema se colapsa.
REPAIR, inicia la recuperación del nodo caído.
BACK, busca un nodo de reserva
SUPERVISOR, elige a un nodo dentro de un comité para ser coordinador de restauración.
ACCEPT_SUPERVISOR, un nodo acepta ser supervisor.
REJECT_SUPERVISOR, un nodo rechaza ser supervisor.
RECOVERED, indica al SuperNodo que se ha restaurado un comité
SPARE(S), indica el nodo de reserva asignado para almacenar los contenidos restaurados
CONTENTS, envía los contenidos restaurados al nodo de reserva designado.
FINISH, indica al SuperNodo que envío los contenidos recueprados a la reserva.

Algoritmo 3: *Máquina de estados del SuperNodo del MCD*

```
1  estado S0 /*el usuario inicia el funcionamiento del sistema
2  Al recibir STARTS del usuario
3     ∀ k ∈ Activos, envia(INIT) a k, ahora /* Mensaje para activar v nodos
4     ∀ j ∈ Reservas, envia(INIT_SPARE) a j, ahora /* Mensaje para poner en reserva s nodos
5     L=∅
6     estado ← S1

7  estado S1 /* operación cotidiana
8  Al recibir CRASH de j
9     si j ∈ Activos entonces
10          Activos = Activos \ {j}
11    otro
12          j = cliente(j)
13    fin
14    Sea L=L ∪ Esquema(j) una copia del conjunto de comités donde j aparece
15    si  ∃ s ∈ Reservas entonces
16          ∀ c ∈ Activos y c ∉ Coordinadores
17          envia(SUPERVISOR) a c, ahora
18          comités_restaurados=0;
19    otro
20          ∀ k ∈ Nodos, envia(COLLAPSE) a k, ahora
21          detener_funcionamiento
22          estado ← S2
23    fin

24 Al recibir ACCEPT_SUPERVISOR de j
25    Coordinadores= Coordinadores ∪ {j}
26    si L ≠ ∅ entonces
27          T=sig_comite(L)
28          envia(TEAM(T)) a j, ahora
29    fin

30 Al recibir BACK de j
31    envia(INIT_SPARE) a j, ahora
32    Reservas = Reservas ∪ {j}

33 Al recibir SUCCESS de j
34    Activos = Activos ∪ {j}

35 Al recibir FATAL de j
36    ∀ k ∈ Nodos, envia(COLLAPSE) a k, ahora
37    deten_funcionamiento
38    estado ← S2

39 Al recibir RECOVERED de j
40    si ∃ s ∈ Reservas entonces
41          envia(SPARE(s)) a j, ahora
42    otro
43          ∀ k ∈ Nodos, envia(COLLAPSE) a k, ahora
44          detener_funcionamiento
45          estado ← S2
46    fin

47 Al recibir FINISH de j
48    si L ≠ ∅ entonces
49          T=sig_comite(L)
50          envia(TEAM(T)) a j, ahora
51    fin

52 estado S2 /* nada se puede hacer ahora
```

Algoritmo 4: *Máquina de estados para un nodo del MCD.*

```
1  estado N0/N2 /* Activo/Reserva aún no se ha calculado su tiempo_de_vida
2  Al recibir INIT/INIT_SPARE de SuperNodo
3     tiempo_de_vida = calcula tiempo aleatorio de acuerdo a f()
4     envia(CRASH) a mí, al tiempo tiempo_de_vida
5     estado ← N1/N5

6  estado N1 /* un nodo activo coopera para restaurar a un comité
7  Al recibir SUPERVISOR(T) de SuperNodo
8     t_restauracion=tiempo_de_restauracion(T)
9     si t_restauracion < tiempo_de_vida entonces
10          envia(ACCEPT_SUPERVISOR (T))
11    otro
12          envia(REJECT_SUPERVISOR (T))
13    fin

14 Al recibir TEAM(T) de SuperNodo
15 si es_mi_comite(T) entonces
16       respuestas=1
17       activos=1
18       dispersos=2
19       max_activos=2
20 otro
21       respuestas=0
22       activos=0
23       dispersos=3
24       max_activos=3
25 fin
26 max_respuestas=0
27 ∀ i ∈ T
28    envia(STATE) a i, ahora
29    si i ≠ Nodo_f entonces
30          max_respuestas++
31    fin

32 Al recibir STATE de s
33    respuesta = calcula de acuerdo a contenidos()
34    si  respuesta < tiempo_de_vida entonces
35          envia(ACTIVE) a s, ahora
36    otro
37          envia(CRASH) a s, ahora
38    fin

39 Al recibir INFO de s
40    t_transmision=transmision(INFO)
41    envia(DISPERSAL) a s, al tiempo t_transmision

42 estado N3 /* esperando por el estado de los participantes del comité
43 Al recibir ACTIVE /CRASH de j
44    respuestas++
45    si msg==ACTIVE entonces
46          activos++
47          Sobrevivientes = Sobrevivientes ∪ {j}
48    fin
49    si respuestas==max_respuestas entonces
50          si activos ≥ max_activos entonces
51                si Sobrevivientes ≠ ∅ entonces
52                      ∀ k ∈ Sobrevivientes, envia(INFO) a k, ahora
53                      disponible = 0
54                      estado ← N4
55                otro
56                      envia(FATAL) a SuperNodo, ahora
57                fin
58          otro
59                envia(FATAL) a SuperNodo, ahora
60          fin
61    fin
```

Algoritmo 5: *Máquina de estados para un nodo del MCD (continuación...).*

```
estado N4 /* esperando por los dispersos
Al recibir DISPERSAL de j
  disponible++
  si disponible == dispersos entonces
      tiempo_de_restauración = calcula de acuerdo a ida_inverso()
      envia(RECOVERED) a SuperNodo, al tiempo_de_restauración
  fin

Al recibir SPARE(s) de SuperNodo
  t_envio_contenidos=transmision(contenidos)
  envia(CONTENTS) a s, al t_envio_contenidos
  envia(FINISH) a SuperNodo, ahora
  estado ← N1

estado N5 /* un nodo de reserva recibe los contenidos restaurados
Al recibir CONTENTS de s
  contenidos++
  si contenidos== Num_Comites entonces
      tiempo_de_vida = calcula tiempo aleatorio de acuerdo a f()
      envia(CRASH) a mí, al tiempo tiempo_de_vida
      envia(SUCCESS) a SuperNodo, ahora
      estado ← N1
  fin

estado N6 /* abandonar el centro de reparación
Al recibir REPAIR de mí
  envia(BACK) a SuperNodo, ahora
  estado ← N2

para cualquier estado /* puede ocurrir en cualquier momento
Al recibir CRASH de mí
  regreso_a_trabajar = calcula de acuerdo a r()
  envia(CRASH) a SuperNodo, ahora
  envia(REPAIR) a mí, al instante regreso_a_trabajar
  estado ← N6

Al recibir COLLAPSE de SuperNodo
  detener_funcionamiento
  estado ← N7

estado N7 /* nada se puede hacer ahora
```

Hasta este punto se definieron los modelos necesarios para evaluar el desempeño del Sistema de Almacenamiento Distribuido. Se definieron dos políticas de atención la PAC y la PAD, para las cuales se propusieron los protocolos de la aplicación. Por otra parte, se definieron los modelos MCC y MCD para evaluar la tolerancia a fallas del sistema. En el siguiente capítulo se evaluará el desempeño del SAD en términos de su tiempo de respuesta para cada una de las políticas, así como la evaluación del modelo de confiabilidad en términos del tiempo de vida del sistema para cada modelo de confiabilidad.

Capítulo 4
Evaluación de desempeño

En la actualidad han surgido diversas propuestas para dar solución a las necesidades de almacenamiento de los usuarios o de las empresas. Es importante evaluar estos sistemas para conocer su funcionamiento bajo determinadas circunstancias y describir la calidad de los servicios que ofrece. En este capítulo se realiza un estudio de evaluación de desempeño del sistema de almacenamiento descrito en el capítulo anterior, en términos de dos parámetros: tiempo de respuesta y confiabilidad.

Podemos identificar tres métodos para realizar un estudio de evaluación de desempeño de un sistema: evaluación por métodos analíticos, por mediciones directas o por simulación. El método analítico se basa en la creación de un modelo mediante el cual el sistema puede estudiarse, pudiendo así, observar su desempeño; sin embargo los modelos pueden resultar muy complejos; sin embargo representan el comportamiento de los sistemas de forma muy fiable. Por otra parte se pueden realizar mediciones directas sobre el sistema implementado, sin embargo existe el problema de escalabilidad, ya que representa la adquisición de componentes y por lo tanto un alto costo, además no es posible estudiar a detalle el funcionamiento del sistema y por último, con este método no es posible realizar repeticiones en los experimentos debido a que no siempre se tienen las mismas condiciones.

Una alternativa a los métodos mencionados es la implementación de un modelo de simulación. En una simulación se puede describir el sistema con el nivel de detalle que se requiera, además pueden agregarse o eliminarse componentes a éste sin representar algún tipo de cos-

to. Por otro lado emplear simulaciones como método de evaluación permite controlar cada escenario sobre el cual se lleva a cabo un experimento. El poder repetir experimentos es un factor importante, debido a que permite analizar las causas de determinados comportamientos de los sistemas y es deseable conocer el escenario bajo el cual se presentan dichos comportamientos. Por las ventajas de un estudio dirigido por simulación, en este capítulo se implementan, mediante este método, los modelos necesarios para evaluar el desempeño del del Sistema de Almacenamiento Distribuido en términos del tiempo de respuesta, así como la evaluación de su confiabilidad. Además en algunos casos se presenta el modelo analítico del sistema para corroborar los datos obtenidos de las simulaciones.

Cabe señalar que para los resultados obtenidos de cualquier método de evaluación es necesario calcular los intervalos de confianza, es decir, los resultados que se obtienen son aproximaciones ya que en un sistema real intervienen diversos factores que no pueden ser medidos mediante tales métodos. Así, un intervalo de confianza describe que tan certeros son los resultados obtenidos. En la ecuación 4.1 se describe el intervalo de confianza.

$$\bar{x} \pm Z_{\alpha/2} \left(\frac{S}{\sqrt{n}} \right) \tag{4.1}$$

Donde:

$\bar{x}$= Valor esperado del parámetro sobre el cual se realizaron las mediciones.

$Z_{\alpha/2}$=Área bajo la función Normal dentro del intervalo $[-\alpha/2,\alpha/2]$.

S= La desviación estándar de los datos obtenidos.

n= El número de muestras obtenidas.

El siguiente paso después de elegir el método de evaluación fue elegir la herramienta para implementar los modelos del sistema. La herramienta elegida fue el simulador de eventos discretos OMNet++ [18]. Esta herramienta se eligió porque permite simular al nivel de

detalle que se requiere, además contiene los módulos necesarios para los protocolos UDP y TCP. Mediante esta herramienta se implementaron los modelos descritos en el capítulo 3.

En la sección 4.1 se describen los modelos implementados para evaluar el SAD en términos del tiempo de respuesta para cada una de las políticas de atención, tanto la PAC como la PAD, así como los resultados obtenidos de las simulaciones realizadas. Asimismo en la sección 4.2, se describen los modelos de simulación implementados, tanto el Modelo Centralizado como el Modelo Descentralizado, para evaluar la confiabilidad del SAD en términos del tiempo de vida media del sistema.

4.1. Tiempo de respuesta

Cada una de las políticas de atención del Sistema de Almacenamiento Distribuido se evalúa según su modelo implementado. Para ambos modelos de simulación, el modelo para la PAC y la PAD, se tienen las siguientes consideraciones:

- Los tiempos entre arribos, tanto el de las solicitudes de almacenamiento como de recuperación, se modelan con una distribución exponencial con parámetros λ_{Store} y $\lambda_{Retrieve}$, respectivamente.

- El tiempo que toma procesar un archivo es proporcional al tamaño de éste, es decir, la complejidad del IDA y su algoritmo inverso es *O(l)*, donde l es el tamaño del archivo. Para obtener los dispersos del archivo enviado por el cliente, el IDA toma 20 s/MB y 15 s/MB, para ensamblar los dispersos a fin de obtener el archivo original. Estas velocidades de procesamiento, fueron obtenidas de mediciones realizadas sobre una computadora con procesador Athlon® 64 y 2 GB de memoria RAM, en la cual se implementó el IDA.

- Cada archivo para almacenar o recuperar es de tamaño fijo igual a 1 MB.

- Una política de servicio FIFO (First In First Out, el primero en llegar es el primero que se atiende), para el servicio de las solicitudes.

- La red de almacenamiento se implementa con tecnología Fast Ethermet (100 Mbps).

- Los protocolos de la aplicación se implementan sobre UDP y la transferencia de datos sobre TCP.

- La capacidad del disco duro de cada computadora de 100 GB. Para cada modelo de simulación se considera que no existe saturación, es decir, no se toma en cuenta el efecto de cuando los discos duros llegan al límite de su capacidad.

- El generador de tráfico, encargado de realizar peticiones de servicio al sistema, se encuentra divido en dos módulos. Uno de los módulos se encarga de generar las peticiones de almacenamiento y el segundo, las peticiones de recuperación. Cada uno de estos módulos representa a un conjunto de usuarios que envían sus solicitudes al sistema.

- El tiempo de respuesta se mide a partir del instante en que las solicitudes se reciben en el despachador.

- Para realizar las comunicaciones vía UDP y las transferencias vía TCP, se generan las conexiones necesarias al inicio de cada simulación y se asume que no existen errores en el canal de comunicaciones durante el funcionamiento del sistema.

En la figura 4.1 se muestra la estructura del modelo de simulación para ambas políticas de servicio. Esta estructura se definió con base en el funcionamiento del algoritmo de dispersión, para el cual se necesita un componente para procesar los archivos y un conjunto de computadoras para realizar el almacenamiento de los archivos dispersos generados.

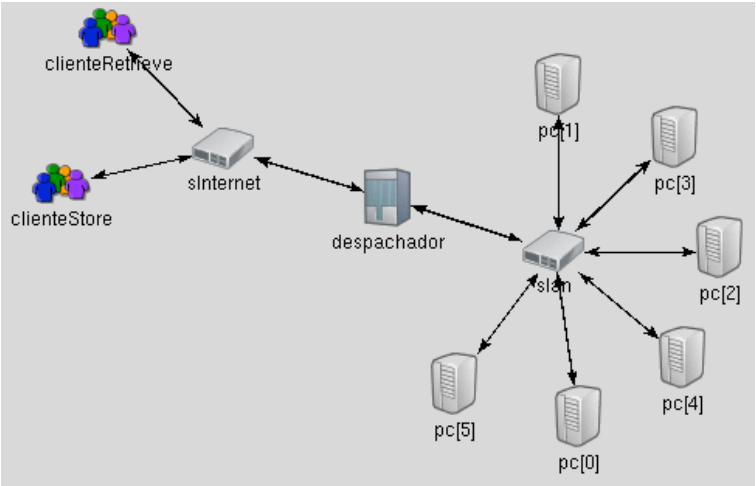

Figura 4.1: Modelo de simulación para el SAD.

4.1.1. Evaluación de desempeño de la PAC

Para el modelo de simulación de la PAC se toma en cuenta el número mínimo de computadoras para el funcionamiento del sistema, el cual es cinco, ya que al procesar un archivo se requiere este número de computadoras para almacenar cada uno de los dispersos generados. En el modelo, la fila de espera se encuentra implementada en el despachador. Además este componente se encarga del procesamiento de los archivos o dispersos para realizar las tareas de almacenamiento y recuperación, respectivamente. Cada una de las computadoras de la red de almacenamiento se conecta al despachador mediante un conmutador. Para almacenar los dispersos el despachador elige el siguiente comité, tomando en cuenta que cada máquina únicamente almacenará un disperso por solicitud. Por otro lado, para la recuperación de un archivo se eligen tres máquinas, del comité correspondiente, para solicitar los dispersos y ensamblarlos.

Considerando los valores de λ_{Store} en el intervalo [0.005,0.290] peticiones/s y los de $\lambda_{Retrieve}$ en el intervalo [0.005,0.050] peticiones/s y un tiempo de simulación de 36000 segundos, se

obtienen los resultados descritos a continuación. Los límites se eligieron con base en el hecho de que lo importante para el sistema, es analizarlo cuando llega al límite de su capacidad y esto se logra aumentando la intensidad de llegada de las solicitudes.

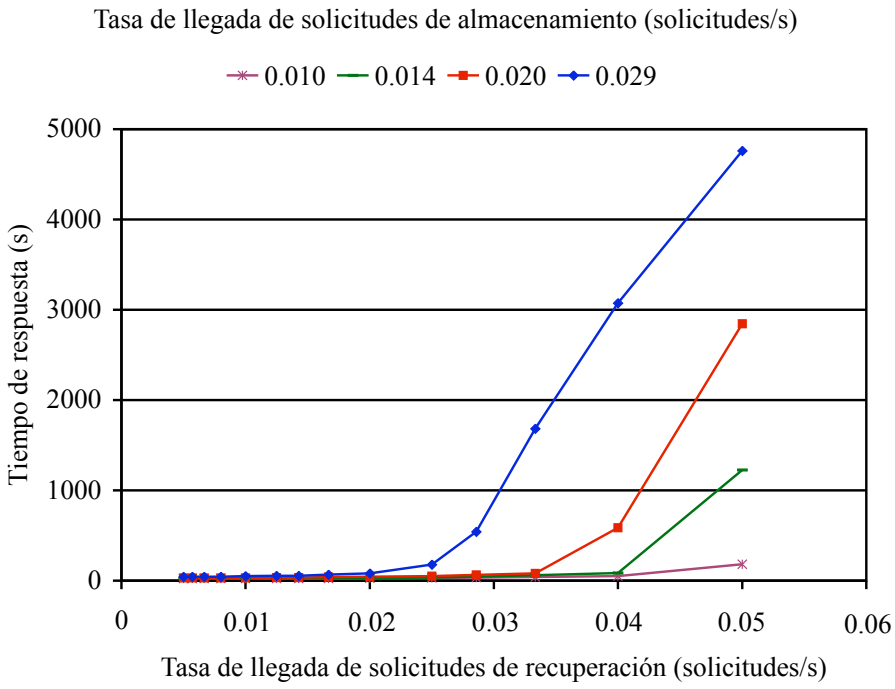

Figura 4.2: Tiempo de respuesta del sistema con la PAC, $v = 5$.

En la gráfica de la figura 4.2, se muestran los resultados más representativos de las simulaciones. Cada una de las curvas corresponde a un valor fijo de λ_{Store} y cada punto de éstas corresponde a un valor de $\lambda_{Retrieve}$. Como se puede observar, las "rodillas" de las curvas corresponden a los puntos críticos del sistema, en los cuales las intensidades de llegada de las solicitudes hacen que éste llegue al límite de su capacidad de servicio. A partir de estos puntos el sistema se vuelve inestable y el tiempo de respuesta crece rápidamente, por lo tanto, a mayor tiempo de funcionamiento del sistema, mayor es el tiempo de respuesta.

Para corroborar los resultados de las simulaciones se planteó un modelo analítico sim-

plificado del sistema, basado en una fila de espera M/G/1, en el cual sólo se consideran los tiempos de servicio asociados a las tareas de procesamiento del IDA sin considerar retardos de transmisión o retardos en los protocolos de comunicación.

El sistema puede recibir ya sea un petición de almacenamiento o una petición de recuperación. Debido a que las solicitudes tienen asociadas distribuciones exponenciales para el tiempo interarribos, se define una tasa total de llegadas de peticiones como la suma de ambas intensidades de llegada:

$$\lambda = \lambda_{Store} + \lambda_{Retrieve} \tag{4.2}$$

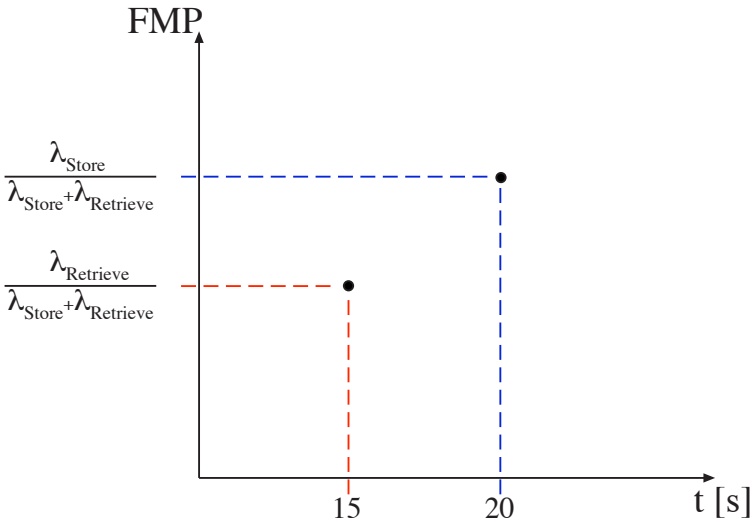

Figura 4.3: Función de masa de probabilidad del tiempo de servicio.

En la figura 4.3 se muestra la función de masa de probabilidad asociada al tiempo de servicio del sistema, en la cual se tienen dos probabilidades:

$$P\{Solicitud = Almacenamiento\} = \frac{\lambda_{Store}}{\lambda_{Store} + \lambda_{Retrieve}} \tag{4.3}$$

$$P\{Solicitud = Recuperacion\} = \frac{\lambda_{Retrieve}}{\lambda_{Store} + \lambda_{Retrieve}} \tag{4.4}$$

De acuerdo a las consideraciones antes mencionadas se puede calcular el tiempo de respuesta del sistema y comparar el modelo con los resultados de las simulaciones. Para esto se realizó el desarrollo del modelo teórico descrito a continuación.

Sea S la variable aleatoria que representa el tipo de solicitud con un espacio de eventos $E = \{s, r\}$ donde s se asocia a las solicitudes de almacenamiento y r a las de recuperación. Además, se define t_s como el tiempo que toma el IDA en generar os dispersos y t_r como el tiempo para ensamblarlos. Entonces,

$$E[T_{Servicio}] = P[S = s] * t_s + P[S = r] * t_r \tag{4.5}$$

Sustituyendo 4.3 y 4.4 en 4.5

$$E[T_{Servicio}] = \frac{\lambda_{Store}}{\lambda_{Store} + \lambda_{Retrieve}} * t_s + \frac{\lambda_{Retrieve}}{\lambda_{Store} + \lambda_{Retrieve}} * t_r \tag{4.6}$$

Por otra parte la tasa de servicio se define como:

$$\mu = \frac{1}{E[T_{Servicio}]} \tag{4.7}$$

Sustituyendo 4.6 en 4.7

$$\begin{aligned} \mu &= \frac{1}{\frac{\lambda_{Store}}{\lambda_{Store} + \lambda_{Retrieve}} * t_s + \frac{\lambda_{Retrieve}}{\lambda_{Store} + \lambda_{Retrieve}} * t_r} \\ \mu &= \frac{\lambda_{Store} + \lambda_{Retrieve}}{\lambda_{Store} * t_s + \lambda_{Retrieve} * t_r} \end{aligned} \tag{4.8}$$

El tiempo de respuesta del sistema se puede obtener calculando el valor esperado del tiempo de espera en el sistema (E[T]). Para una fila M/G/1 se tiene lo siguiente [19]:

$$\rho = \frac{\lambda}{\mu} \tag{4.9}$$

$$L = \rho + \frac{\rho^2 + \lambda^2\sigma}{2(1-\rho)} \tag{4.10}$$

$$\tag{4.11}$$

Sustituyendo los valores de λ (4.2) y μ (4.8) en 4.9, se obtiene el valor de ρ

$$\begin{aligned} \rho &= \frac{\lambda_{Store} + \lambda_{Retrieve}}{\frac{\lambda_{Store} + \lambda_{Retrieve}}{\lambda_{Store} * t_s + \lambda_{Retrieve} * t_r}} \\ \rho &= \lambda_{Store} * t_s + \lambda_{Retrieve} * t_r \end{aligned} \tag{4.12}$$

Además la varianza (σ^2) esta dada por la siguiente ecuación:

$$\begin{aligned} \sigma^2 &= P\{S = s\} * (20 - E[T_{Servicio}])^2 + P\{S = r\} * (15 - E[T_{Servicio}])^2 \\ \sigma^2 &= \frac{\lambda_{Store}}{\lambda_{Store} + \lambda_{Retrieve}} \left(20 - \left(\frac{\lambda_{Store}}{\lambda_{Store} + \lambda_{Retrieve}} t_s + \frac{\lambda_{Retrieve}}{\lambda_{Store} + \lambda_{Retrieve}} t_r\right)\right)^2 + \\ &\quad + \frac{\lambda_{Retrieve}}{\lambda_{Store} + \lambda_{Retrieve}} \left(15 - \left(\frac{\lambda_{Store}}{\lambda_{Store} + \lambda_{Retrieve}} t_s + \frac{\lambda_{Retrieve}}{\lambda_{Store} + \lambda_{Retrieve}} t_r\right)\right)^2 \end{aligned} \tag{4.13}$$

Por último tenemos que:

$$E[T] = \frac{L}{\lambda} \tag{4.14}$$

En la gráfica de la figura 4.4 se muestra la comparación del modelo analítico con el modelo de simulación, donde los extremos derechos de las curvas corresponden a los valores de las tasas de llegadas de las solicitudes, para los cuales el sistema llega a su máxima capacidad y a partir de éstos el sistema es inestable. Por otra parte, se muestra que para los valores de λ_{Store} en el intervalo [0.005,0.0290] peticiones/s y los de $\lambda_{Retrieve}$ en el intervalo [0.005,0.050]

peticiones/s, el desempeño del sistema es independiente de la tecnología de red que se emplee; sin embargo cuando la intensidad de llegada de ambas solicitudes se incrementa, los retardos de transmisión y los retardos en los protocolos de comunicación comienzan a tener un impacto importante en el desempeño del sistema, por lo tanto, es preferible emplear una tecnología de red de mayor velocidad.

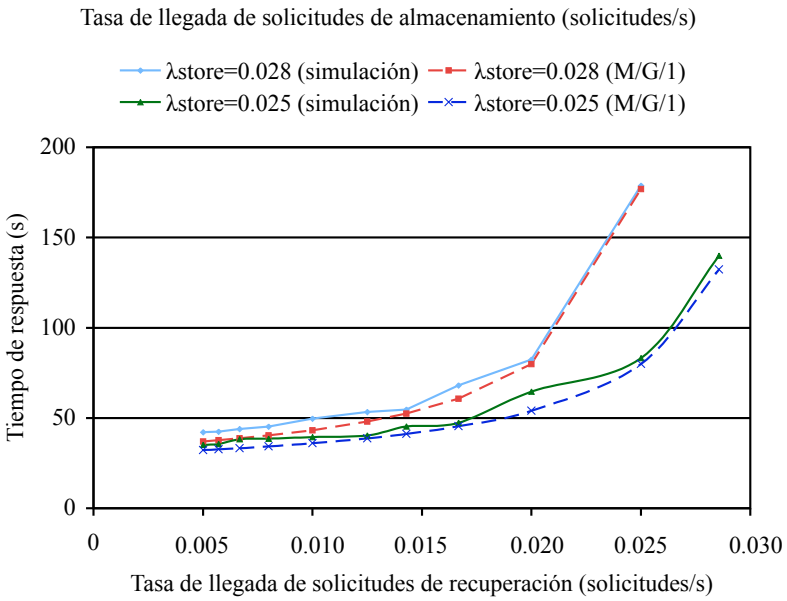

Figura 4.4: Tiempo de respuesta del sistema con el modelo matemático.

4.1.2. Evaluación de desempeño de la PAD

Para la evaluación de la Política de Atención Descentralizada, el despachador sólo se encarga de la gestión de las solicitudes; el procesamiento de los archivos y el ensamble de los dispersos se lleva a cabo en las computadoras de la red de almacenamiento. La fila de espera para las solicitudes se implementa en el despachador. La elección del coordinador la realiza el despachador eligiendo una computadora de la red de almacenamiento de forma aleatoria,

cada máquina tiene asociada una probabilidad $p = \frac{1}{v}$ de ser elegida, donde v es el número de computadoras activas en la red de almacenamiento. Si el despachador no encuentra alguna máquina disponible para ser coordinador, la solicitud recibida se forma en la fila de espera y cuando una computadora se liberé la solicitud que se encuentre al frente de la fila de espera se atiende..

En la gráfica 4.5 se muestran los resultados más representativos de las simulaciones realizadas con cinco máquinas en la red de almacenamiento, para los valores de λ_{Store} en el rango [0.005,0.200] peticiones/s y para $\lambda_{Retrieve}$ en el rango [0.005,0.200] peticiones/s. Asimismo en la gráfica 4.6 se muestran los resultados de las simulaciones con seis máquinas. Las "rodillas" de las curvas muestran los puntos para los cuales el sistema llega a su capacidad y a partir de éstos el sistema se vuelve inestable, implicando un aumento drástico en el tiempo de respuesta del sistema.

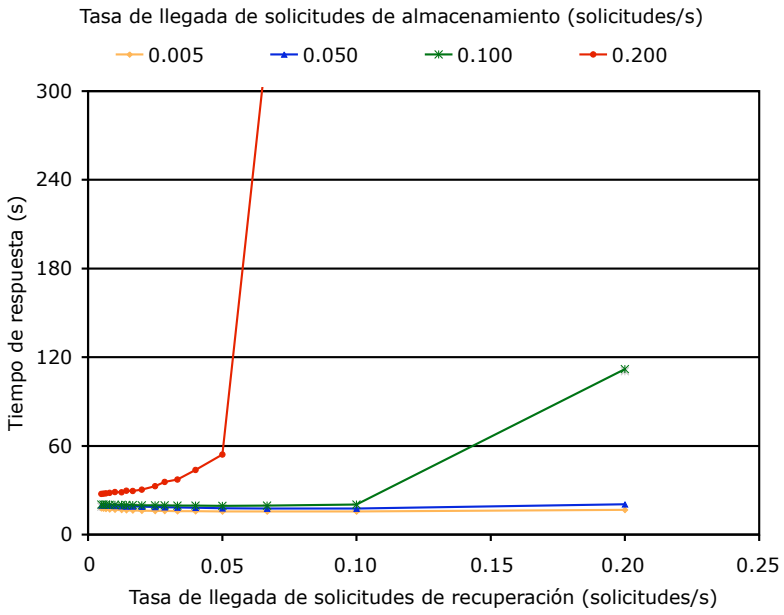

Figura 4.5: Tiempo de respuesta del sistema con la PAD, $v = 5$.

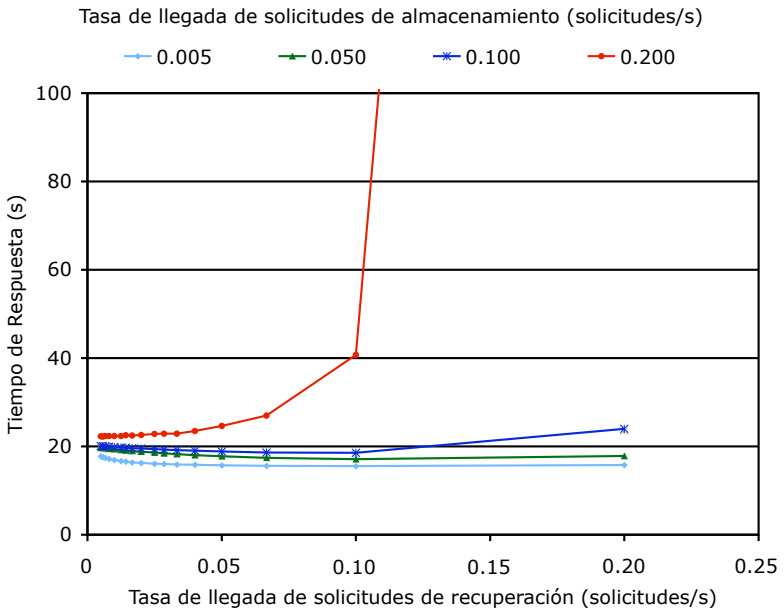

Figura 4.6: Tiempo de respuesta del sistema con la PAD, $v = 6$.

Para corroborar los resultados se planteó un modelo de simulación simplificado, en el cual no se toman en cuenta los tiempos de transmisión ni los tiempos de retardo en los protocolos de comunicación. En las gráficas de las figuras 4.7 y 4.8, se muestra la comparación entre el modelo completo y el modelo simplificado para la PAD, donde se observa que ambos modelos tienen valores muy parecidos. Así para los valores de λ_{Store} y $\lambda_{Retrieve}$ en el rango [0.005,0.200] peticiones/s la tecnología de red no tiene mayor impacto en el desempeño del sistema; sin embargo al incrementarse ambas intensidades de llegada, el sistema requiere de una mayor velocidad para disminuir los retardos de transmisión y retardos en los protocolos de comunicación, por lo tanto es recomendable emplear tecnologías de red con altas velocidades de transmisión.

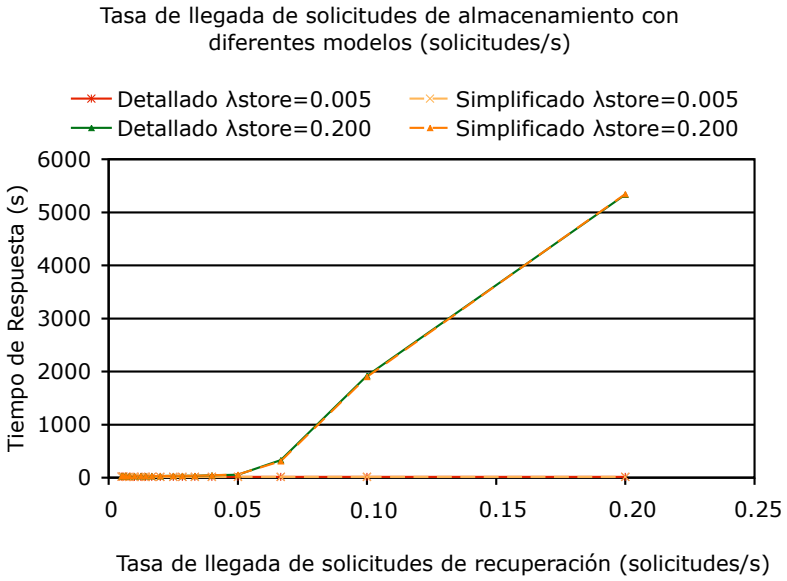

Figura 4.7: Comparación Modelo Detallado vs. Modelo Simplificado, $v = 5$.

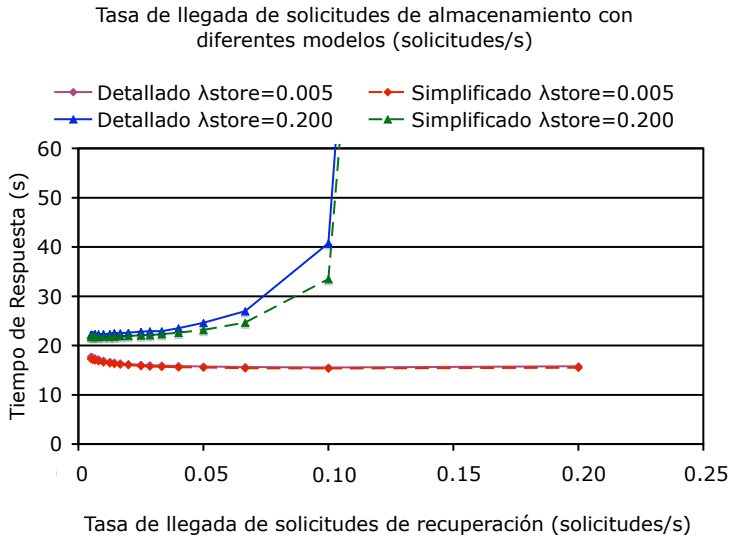

Figura 4.8: Comparación Modelo Detallado vs. Modelo Simplificado, $v = 6$.

Al comparar ambas políticas de atención para el caso en el que se tienen cinco máquinas en la red de almacenamiento con el caso de seis máquinas, se obtiene la gráfica de la figura 4.9.

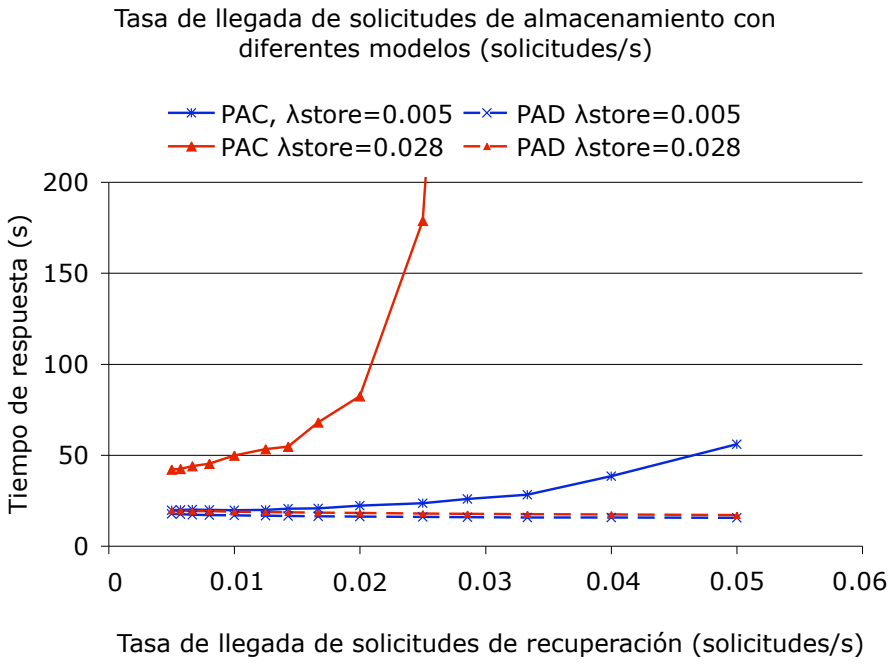

Figura 4.9: Comparación PAC vs. PAD, $v = 5$.

El impacto que tiene el hecho de agregar máquinas a la red de almacenamiento se muestra en la gráfica de la figura 4.10, en la cual se tiene el tiempo de respuesta fijando el valor de λ_{Store} =0.2 peticiones/s y $\lambda_{Retrieve}$ =[0.005,0.2] peticiones/s. Cada una de las curvas corresponde a los valores de $v = \{5, 6, 7\}$, donde v, corresponde al número de máquinas en la red de almacenamiento. En la gráfica se puede observar el decremento del tiempo de respuesta del sistema conforme se aumenta el número de máquinas en la red de almacenamiento.

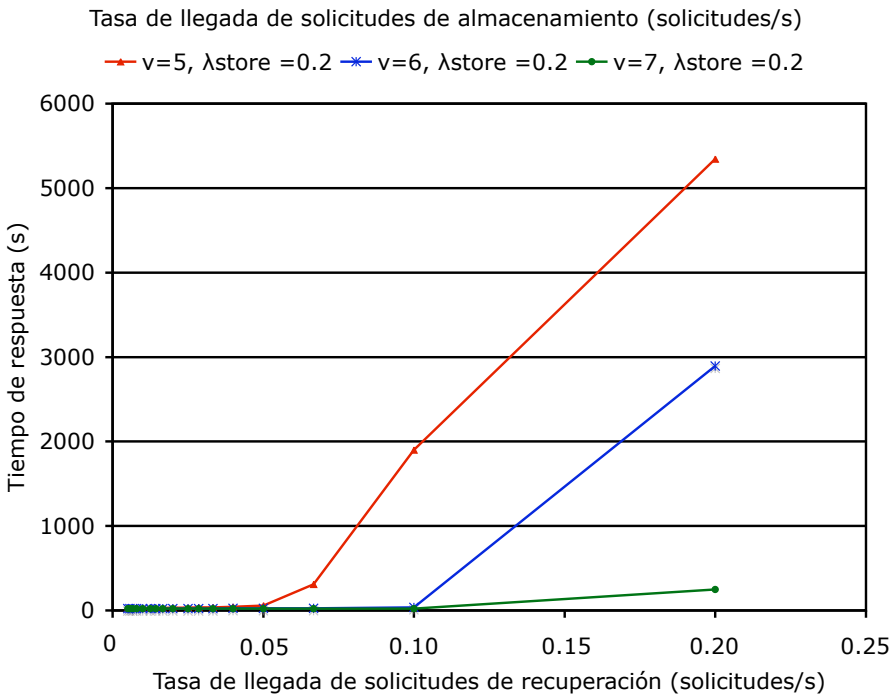

Figura 4.10: Comparación de la PAD con $v = \{5, 6, 7\}$.

La Política de Atención Descentralizada muestra un mejor desempeño en términos del tiempo de respuesta, admitiendo un mayor rango de valores de las tasas de llegada de las solicitudes a comparación de la Política de Atención Centralizada.

4.2. Tolerancia a fallas

Para la evaluar la confiabilidad del sistema se plantearon dos modelos de simulación, uno para el Modelo de Confiabilidad Centralizado y otro para el Modelo de Confiabilidad Descentralizado. Para ambos modelos se tienen las siguientes consideraciones:

- Cada disco duro tiene una capacidad de 100 GB y el análisis contempla que los discos están ocupados a su capacidad total.
- Una velocidad de transmisión de datos de 100 Mbps, para cada enlace entre los componentes del sistema.
- El tiempo que toma recuperar un disperso es de 5 MB/s. Este tiempo fue medido de la implementación del IDA sobre una computadora de escritorio con las características descritas en la sección 4.1.
- El tiempo entre fallas de cada nodo se describe mediante una distribución exponencial. Así las variables aleatorias que representan los tiempos de vida de los discos son independientes e idénticamente distribuidas.
- El tiempo de reparación de cada nodo se describe por una distribución exponencial. Las variables aleatorias que representan los tiempos de reparación son independientes e idénticamente distribuidas.
- Los parámetros de los modelos son: el número de nodos activos, el número de nodos de reserva, el tiempo de vida media de los discos duros (MTTF, del inglés *Mean Time To Failure*) y por último el tiempo de reparación de cada nodo.
- El instante de falla de cada nodo se calcula al iniciar la operación del sistema y cuando finaliza la reparación de un nodo.

4.2.1. Evaluación de la confiabilidad con el MCC

Para el estudio de evaluación de la confiabilidad del sistema con el Modelo de Confiabilidad Centralizado se obtuvieron los resultados que se muestran en la tabla 4.4, asimismo se muestra el impacto de cada uno de los parámetros sobre el tiempo de vida media del sistema. Para los resultados los intervalos de confianza se calcularon al 99.99 %.

El tiempo de vida media más grande obtenido fue de 1552.38±60.90 años que se obtuvo con los siguientes valores para cada parámetros: 5 nodos activos, 2 nodos de reserva, 20000 horas de vida media de cada disco y 10 horas de tiempo medio de reparación para cada nodo. Sin embargo para las combinaciones que se muestran en la tabla 4.1, se puede observar que se tienen tiempos de vida media del sistema dentro del intervalo de confianza, por lo tanto, al tener reservas suficientes, el tiempo de reparación no influye en el tiempo de vida media del sistema. Para el modelo centralizado el número suficiente de reservas es $s = 2$, ya que al tener este número o uno mayor , el aumento en el parámetro observado es marginal. Por otra parte, en la tabla 4.2, se puede observar que al tener reservas suficientes en el sistema, el tiempo de vida media del sistema depende solamente del tiempo de vida media de los discos duros (es decir, del tiempo de vida media de los nodos).

Tabla 4.1: Mayores tiempos de vida media del sistema con el MCC.

v	s	MTTF del disco (horas)	Tiempo de reparación (horas)	MTTF del sistema (años)
5	2	20000	5	1549.27
			10	1552.38
			20	1482.21
5	3	20000	5	1550.24
			10	1548.48
			20	1541.26

El menor valor obtenido fue de 2.05±0.08 años para los siguientes valores de los parametros: 7 nodos activos, 1 nodo de reserva, una vida media de 5000 horas por disco y 20 horas

Tabla 4.2: Incremento del tiempo de vida media del sistema con el MCC.

v	s	MTTF del disco (horas)	Tiempo de reparación (horas)	MTTF del sistema (años)	Incremento* (%)
5	3	5000	20	27.16	-
		10000		200.34	737.62
		20000		1541.26	5674.74

*Los porcentajes se calcularon en relación al valor menor de MTTF del sistema mostrado en el primer renglón.

de tiempo de reparación de un nodo. Para este caso, el tiempo de reparación es un factor importante ya que sólo se tiene una reserva para recuperar contenidos. En otras palabras, es necesario tener un tiempo de reparación corto para tener una reserva disponible en caso de que ocurra una segunda falla, mientras se recupera la primera.

De los resultados descritos en la tabla 4.4 se puede observar que los parámetros que tienen mayor impacto en el tiempo de vida media del sistema son el número de nodos activos y el tiempo de vida media de los discos duros; así, se tiene el decremento que se muestra en la tabla 4.3, donde cada decremento se toma con relación al tiempo de vida media más alto obtenido. El decremento que se presenta, se debe al aumento de la tasa de fallas en el sistema.

Tabla 4.3: Porcentajes de decremento del tiempo de vida media del sistema con el MCC.

v	s	MTTF del disco (horas)	Tiempo de reparación (horas)	MTTF del sistema (años)	Decremento* (%)
5	2	20000	10	1552.38	-
6				518.13	66.62
7				279.47	81.99

*Los porcentajes se calcularon en relación al valor mayor de MTTF del sistema mostrado en el primer renglón

En la figura 4.11 se muestra el histograma obtenido a través de los resultados obtenidos en las simulaciones, para el tiempo de vida media del sistema más alto; asimismo en la figura 4.12, se muestra histograma para el tiempo de vida menor. Además se realizó una comparación, para ambos casos, con un modelo de probabilidad exponencial de valor medio igual a los tiempos correspondientes al mayor y menor tiempo de vida media del sistema.

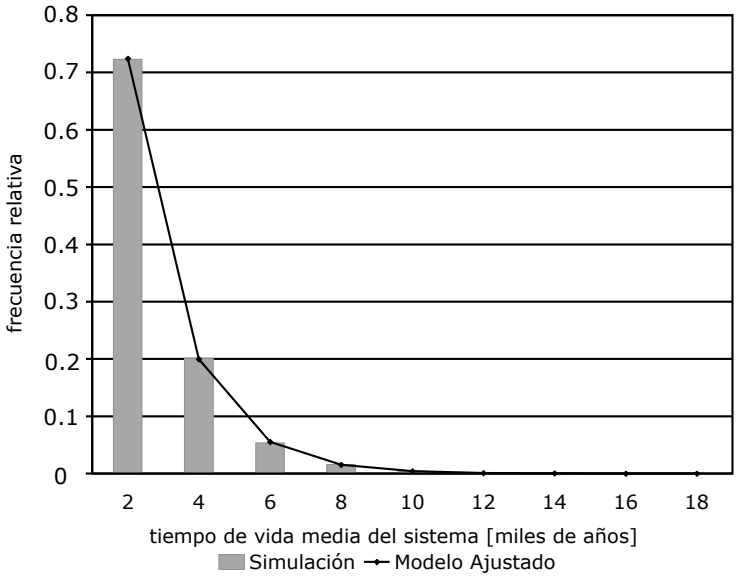

Figura 4.11: Mayor tiempo de vida media del sistema con el MCC.

Como se puede observar en las gráficas, se tiene una muy buena aproximación entre los resultados del modelo de simulación y el modelo ajustado. Por lo tanto, se puede decir que el tiempo de vida del sistema tiene un comportamiento exponencial decreciente.

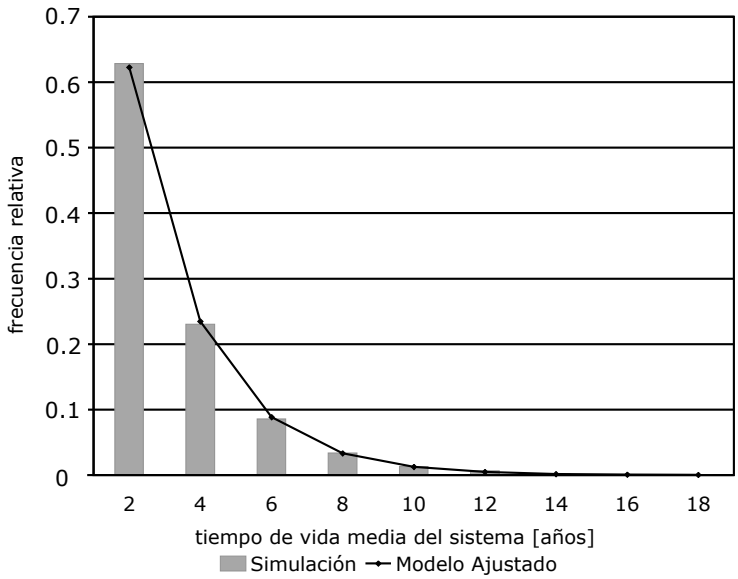

Figura 4.12: Menor tiempo de vida media del sistema con el MCC.

Tabla 4.4: Resultados del Modelo de Confiabilidad Centralizado.

v												
5				6				7			
s	MTTF del disco (horas)	Tiempo de Reparación (horas)	MTTF del Sistema (años)	s	MTTF del disco (horas)	Tiempo de Reparación (horas)	MTTF del Sistema (años)	s	MTTF del disco (horas)	Tiempo de Reparación (horas)	MTTF del Sistema (años)
1	5000	5	11.91	1	5000	5	6.08	1	5000	5	3.82
		10	7.56			10	4.44			10	2.94
		20	4.58			20	2.89			20	**2.05**
	10000	5	58.07		10000	5	31.85		10000	5	20.70
		10	33.40			10	20.64			10	14.32
		20	18.16			20	12.19			20	8.87
	20000	5	259.62		20000	5	158.31		20000	5	107.40
		10	142.26			10	93.14			10	65.96
		20	74.03			20	51.49			20	37.91
2	5000	5	26.94	2	5000	5	9.75	2	5000	5	5.43
		10	27.10			10	9.68			10	5.44
		20	25.66			20	9.47			20	5.34
	10000	5	198.79		10000	5	69.80		10000	5	37.43
		10	199.51			10	69.26			10	37.78
		20	191.56			20	68.24			20	36.56
	20000	5	1549.27		20000	5	519.82		20000	5	280.42
		10	**1552.38**			10	518.13			10	279.47
		20	1482.21			20	506.24			20	272.42
3	5000	5	26.87	3	5000	5	9.65	3	5000	5	5.47
		10	27.07			10	9.72			10	5.49
		20	27.16			20	9.59			20	5.50
	10000	5	197.57		10000	5	67.62		10000	5	37.53
		10	199.09			10	68.36			10	37.87
		20	200.34			20	68.56			20	37.49
	20000	5	1550.24		20000	5	520.94		20000	5	277.58
		10	1548.48			10	519.00			10	280.81
		20	1541.26			20	522.16			20	277.75

4.2.2. Evaluación de la confiabilidad con el MCD

Para el estudio de evaluación de la confiabilidad del sistema con el Modelo de Confiabilidad Descentralizado se obtuvieron los resultados mostrados en la tabla 4.4, asimismo se muestra el impacto de cada uno de los parámetros sobre el tiempo de vida media del sistema. Para los resultados los intervalos de confianza se calcularon al 99.99 %.

El tiempo de vida media más grande obtenido fue de 4981.60±192.06 años que se obtuvo con los siguientes valores para cada parámetro: 6 nodos activos, 3 nodos de reserva, 20000 horas de vida media de cada disco y 10 horas de tiempo medio de reparación para cada nodo. Sin embargo, al tener las reservas suficientes ($s = 2$ o mayor), se tiene un incremento marginal en el tiempo de vida media del sistema como se muestra en la tabla 4.5. Este incremento marginal se debe a que el sistema tiene las suficientes reservas para recuperarse de una siguiente falla mientras se restauran los contenidos de la falla anterior.

En la tabla 4.6, se muestra el incremento del tiempo de vida media del sistema al incrementarse el tiempo de vida media de los discos. Esto sucede cuando se tienen reservas suficientes, por lo tanto, en la tabla 4.7 se puede observar que el tiempo de reparación no influye en los resultados.

Tabla 4.5: Mayores tiempos de vida media del sistema con el MCD.

v	s	MTTF del disco (horas)	Tiempo de reparación (horas)	MTTF del sistema (años)
6	2	20000	5	4889.68
			10	4929.65
			20	4973.56
6	3	20000	5	4924.65
			10	4981.60
			20	4841.71

El menor valor obtenido fue de 8.86±0.34 años para los siguientes valores de los parámetros: 7 nodos activos, 1 nodo de reserva, una vida media de 5000 horas por disco y 20 horas

Tabla 4.6: Incremento del tiempo de vida media del sistema con el MCD.

v	s	MTTF del disco (horas)	Tiempo de reparación (horas)	MTTF del sistema (años)	Incremento (%)
6	3	5000	20	91.23	-
		10000		695.05	761.86
		20000		4841.71	5307.15

*Los porcentajes se calcularon en relación al valor menor de MTTF del sistema mostrado en el primer renglón.

Tabla 4.7: Impacto del tiempo de reparación con el MCD.

v	s	MTTF del disco (horas)	Tiempo de reparación (horas)	MTTF del sistema (años)
6	3	20000	5	4924.65
			10	4981.60
			20	4841.71

de tiempo de reparación de un nodo.

De los resultados descritos en la tabla 4.9 se puede observar que los parámetros que tienen mayor impacto en el tiempo de vida media del sistema son el número de nodos activos y el tiempo de vida media de los discos duros, y se tiene el decremento que se muestra en la tabla 4.8, donde cada decremento se toma con relación al tiempo de vida media más alto obtenido. El decremento que se presenta, se debe al aumento de la tasa de fallas en el sistema y a la formación de filas de espera. Las filas de espera se deben a que el SuperNodo distribuye la carga de trabajo sobre cada nodo activo para restaurar los contenidos de una primera falla, pero al ocurrir la segunda falla mientras se restaura la anterior, no existen nodos activos disponibles para restaurar contenidos, por lo tanto se inserta en la fila de espera la restauración de los contenidos de la segunda falla.

En la figura 4.13 se muestra el histograma obtenido a través de los resultados obtenidos en las simulaciones, para el tiempo de vida media del sistema más alto; asimismo en la figura 4.14, se muestra el histograma para el tiempo de vida menor que se obtuvo de las simulaciones. Además se realizó una comparación, para ambos casos, con un modelo de

Tabla 4.8: Porcentajes de decremento del tiempo de vida media del sistema con el MCD.

v	s	MTTF del disco (horas)	Tiempo de reparación (horas)	MTTF del sistema (años)	Decremento (%)
5				1279.99	26.43
6	3	20000	20	4841.71	-
7				3787.54	78.22

*Los porcentajes se calcularon en relación al valor mayor de MTTF del sistema mostrado en el segundo renglón.

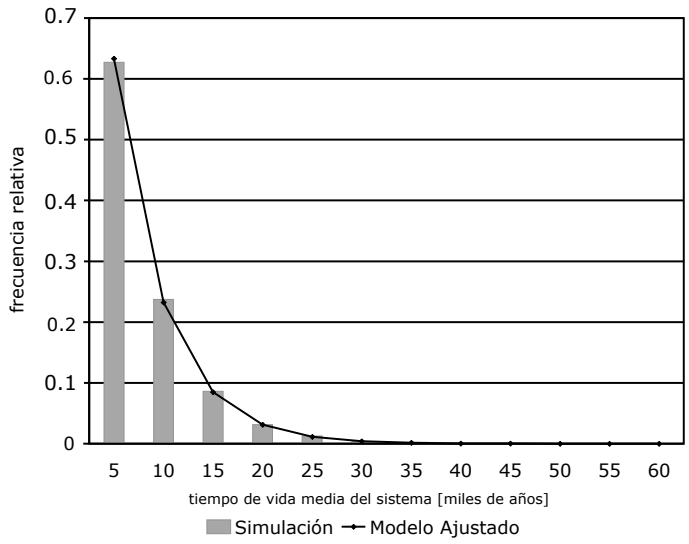

Figura 4.13: Mayor tiempo de vida media del sistema con el MCD.

probabilidad exponencial de valor medio igual a los tiempos correspondientes al mayor y menor tiempo de vida media del sistema. Como se puede observar en las gráficas, se tiene una muy buena aproximación entre los resultados del modelo de simulación y el modelo ajustado. Por lo tanto, se puede decir que el tiempo de vida del sistema tiene un comportamiento exponencial decreciente.

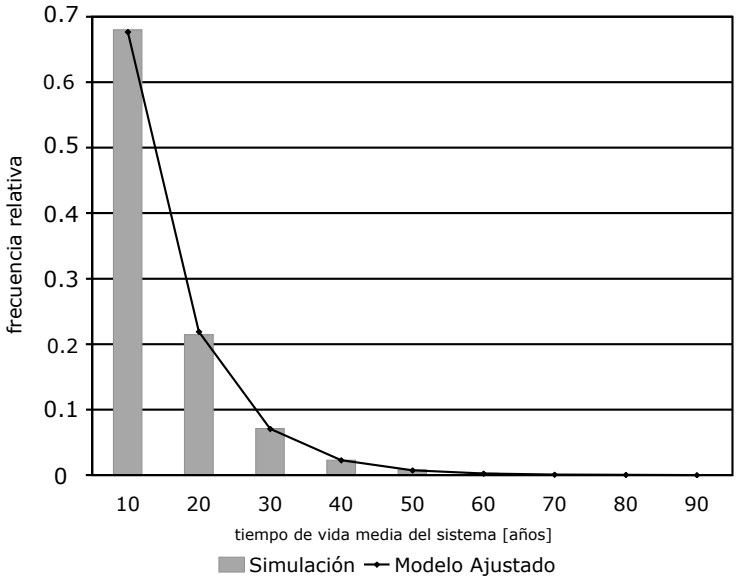

Figura 4.14: Menor tiempo de vida media del sistema con el MCD.

Tabla 4.9: Resultados del Modelo de Confiabilidad Descentralizado.

v												
5				6				7				
s	mttf del disco (horas)	Tiempo de Reparación (horas)	mttf del Sistema (años)	s	mttf del disco (horas)	Tiempo de Reparación (horas)	mttf del Sistema (años)	s	mttf del disco (horas)	Tiempo de Reparación (horas)	mttf del Sistema (años)
1	5000	5	25.89	1	5000	5	91.68	1	5000	5	59.58
		10	25.92			10	82.28			10	32.92
		20	25.21			20	38.53			20	**8.85**
	10000	5	185.50		10000	5	672.22		10000	5	400.75
		10	183.13			10	575.06			10	170.33
		20	181.95			20	192.70			20	36.47
	20000	5	1277.92		20000	5	4875.28		20000	5	2493.63
		10	1291.20			10	3652.66			10	782.19
		20	1289.37			20	864.89			20	148.31
2	5000	5	25.67	2	5000	5	90.70	2	5000	5	70.56
		10	25.80			10	91.99			10	70.62
		20	26.00			20	90.08			20	66.17
	10000	5	179.27		10000	5	676.45		10000	5	529.61
		10	182.36			10	679.46			10	502.92
		20	181.68			20	670.28			20	492.14
	20000	5	1295.09		20000	5	4889.68		20000	5	3697.95
		10	1284.74			10	4929.65			10	3689.86
		20	1268.90			20	4973.56			20	3479.42
3	5000	5	25.48	3	5000	5	93.12	3	5000	5	71.40
		10	25.29			10	91.26			10	69.98
		20	25.93			20	91.23			20	69.86
	10000	5	181.20		10000	5	685.69		10000	5	523.27
		10	179.05			10	675.29			10	531.73
		20	180.39			20	695.05			20	515.81
	20000	5	1291.61		20000	5	4924.65		20000	5	3805.19
		10	1318.80			10	**4981.60**			10	3758.70
		20	1279.99			20	4841.71			20	3787.54

Para ambos modelos de confiabilidad se puede observar (en cada una de las gráficas de resultados) una comparación con una función exponencial negativa con un valor medio igual los obtenidos experimentalmente, de la cual se obtuvo un buen grado de aproximación.

En los resultados obtenidos de las simulaciones para el MCC se puede observar que el parámetro que tiene mayor impacto en el tiempo de vida del sistema, es el número de nodos activos, pues al incrementar éste, se incrementa la tasa de fallas en el sistema haciendo que el tiempo de vida del sistema se decremente. Esto se debe a que las fallas en los nodos se representan por un proceso de Poisson y por lo tanto al observar el sistema completo, las tasas de fallas se suman, es decir, si se tienen x nodos y una tasa de fallas λ_{falla} para cada nodo, la tasa de fallas total será de $x * \lambda_{falla}$. Así si ocurre una primer falla en cualquier nodo, con dos fallas más que ocurran, en cualesquiera dos nodos, mientras se recuperan los contenidos del primero, el sistema colapsa. Esta situación se ilustra en la figura 4.15, donde se observa que al ocurrir la primera falla inicia el proceso de restauración de contenidos y si antes de finalizar éste ocurren dos fallas más, el sistema colapsa. Por otra parte, el tiempo de restauración de contenidos para el MCC es de 512000 s, que corresponde a restaurar 100 GB de datos de un nodo, sin considerar los tiempos de transmisión ya que éstos son despreciables en comparación del tiempo de restauración.

Para el caso del MCD, se puede observar que los mejores tiempos de vida media del sistema se obtiene con seis nodos activos. Esto se debe a que la restauración de los contenidos de un nodo se realiza en tan sólo un paso, donde un paso de restauración corresponde al tiempo en el que se recuperan los contenidos de un solo comité. En la tabla 4.10 se muestran los tiempos de restauración de los contenidos de un nodo, los cuales se calcularon tomando en cuenta los 100 GB de almacenamiento de cada nodo y el tiempo que le toma al IDA inverso recuperar los dispersos correspondientes.

Como se puede observar de la tabla 4.10, para el caso de 5 nodos activos, el MCD y el MCC son iguales, en cuanto a tiempo de recuperación y esto explica el por qué los tiempos

Tabla 4.10: Tiempos de restauración de contenidos con el MCD.

v	Tiempo de Restauración (s)
5	512000
6	102400
7	102400

de vida media del sistema son muy parecidos en este caso. Sin embargo, al aumentar el número de nodos activos en el MCD, el tiempo de restauración es de 102400 y no varía. Se podría pensar que al tener el mismo tiempo de restauración para los casos de $v = 6$ y $v = 7$ el tiempo de vida media del sistema debería ser el mismo; sin embargo, la tasa de fallas aumenta, provocando que el sistema disminuya dicho tiempo.

Al comparar ambos modelos, tanto el MCC como el MCD, en términos del tiempo de vida media del sistema, se obtienen los resultados mostrados en la tabla 4.11. En esta tabla se observa un mejor desempeño del sistema para el modelo descentralizado y en general se debe a que el tiempo de restauración de contenidos disminuye a una quinta parte respecto al modelo centralizado, para el caso de seis y siete nodos activos.

Tabla 4.11: Comparación entre el MCC y el MCD.

Modelo	v	s	MTTF del disco (horas)	Tiempo de reparación (horas)	MTTF del sistema (años)
MCC	5	3	20000	20	1541.26
MCD	5	3	20000	20	1279.99
MCC	6	3	20000	20	522.16
MCD	6	3	20000	20	4841.71
MCC	7	3	20000	20	277.75
MCD	7	3	20000	20	3787.54

Otro resultado obtenido de la evaluación de la confiabilidad del sistema es que, agregar más de dos nodos de reserva representa un aumento marginal en el tiempo de vida media del sistema, por lo tanto este número es el suficiente para obtener un buen grado de confiabilidad. Por otra parte, si se tienen las reservas suficientes el tiempo de reparación no tiene mayor

impacto.

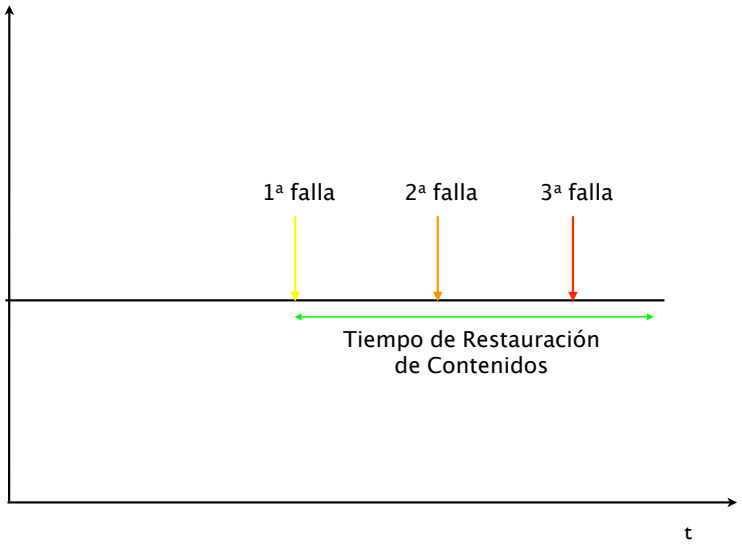

Figura 4.15: Colapso del Sistema.

Capítulo 5
Conclusiones y recomendaciones para trabajo futuro

Los servicios de almacenamiento ofrecidos de forma rápida y confiable se han convertido en el factor clave de las operaciones de las empresas. Esta situación motiva diversas ofertas para crear sistemas de almacenamiento eficientes. En este contexto, las redes de almacenamiento son una alternativa interesante. No sólo ofrecen una alta capacidad, comparadas con el almacenamiento local, sino también su naturaleza distribuida puede usarse para crear sistemas tolerantes a fallas. A partir de este trabajo se puede observar que es factible emplear tecnologías de red usadas comúnmente, tales como Fast Ethernet, para implementar una red de almacenamiento y aún proveer un nivel de rendimiento satisfactorio. Aunque el uso de tecnologías de red dedicadas de alta velocidad (por ejemplo, Canal de Fibra) es deseable, podemos usar la naturaleza distribuida de una red de almacenamiento y un balance de carga apropiado para lograr un rendimiento aceptable. Al hacer esto podemos tener un sistema capaz de hacer frente, de forma significativa, a cargas de trabajo altas.

Los resultados obtenidos de la evaluación de desempeño del sistema en términos de su tiempo de respuesta, muestran que para las condiciones simuladas y bajas intensidades de llegadas de las solicitudes, menores a 0.29 peticiones/s de almacenamiento y 0.05 peticiones/s para recuperación (cada petición es para almacenar o recuperar archivos de 1 MB), se puede implementar la Política de Atención Centralizada. Sin embargo, cuando las intensidades se incrementan es preferible emplear la Política de Atención Descentralizada, para obtener así un

buen desempeño del sistema. Además se puede observar que el sistema es independiente de la tecnología de red sobre la cual se implemente, con intensidades de llegadas de las solicitudes menores a 0.2 peticiones/s para almacenamiento y recuperación. Esto se debe a que el tiempo de procesamiento de los archivos tiene el mayor impacto en el tiempo de respuesta.

Por otro lado, del estudio de confiabilidad muestra que el tiempo de vida media del sistema depende principalmente del número de máquinas que se agreguen a la red de almacenamiento, ya que cuando aumenta este número se forma un mayor número de comités y disminuye la vida media del sistema. Otro resultado interesante es que a partir de 2 máquinas en el sistema, se obtienen resultados marginales, es decir, no se obtiene una gran diferencia agregando más reservas por lo tanto resultaría un gasto innecesario.

El resultado más importante observando los resultados del estudio de evaluación es que, mientras los resultados para la evaluación del tiempo de respuesta indican que al agregar más computadoras se obtiene un mejor desempeño, por otro lado el estudio de su confiabilidad indica lo contrario, que al aumentar el número de computadoras, disminuye el tiempo de vida media del sistema. En la Política de Atención Centralizada no afecta el número de computadoras en la red de almacenamiento y esto es porque el despachador realiza las tareas de procesamiento y gestión de las solicitudes, mientras que la red de almacenamiento cumple con el único propósito de almacenamiento de archivos. Por otra parte en la Política de Atención Descentralizada, el despachador sólo gestiona las solicitudes y las computadoras de la red de almacenamiento son las que se encargan de procesar las solicitudes, por esta razón cuanto sea más grande el número de computadoras en este bloque del sistema, mayor capacidad se tiene para atender solicitudes. Sin embargo, con el Modelo de Confiabilidad Centralizado, al aumentar el número de computadoras activas se reduce el tiempo de vida del sistema. Así, con el Modelo de Confiabilidad Descentralizado se logra evitar este efecto, haciendo que el tiempo de vida aumente considerablemente hasta llegar un punto óptimo, que en el caso de este modelo es para seis nodos activos , para el cual se tiene el mayor tiempo de vida media.

Sin embargo al aumentar a siete computadoras el tiempo de vida media del sistema disminuye debido al incremento de la tasa de fallas en el sistema, aunque se tiene un resultado mejor que en el caso del modelo descentralizado y que en el mismo modelo descentralizado con 5 nodos activos. Además, como se puede observar en la tabla 5.1, al aumentar el número de comités se tiene un mayor balance de carga en el sistema, ya que cada computadora disminuye el tiempo en el cual trabaja realizando tareas de almacenamiento, es decir, cada nodo no participa en todos los comités.

Tabla 5.1: Balance de carga del sistema.

v	Número de comités en los que participa un nodo	Número total de comités
5	1	1
6	5	6
7	15	21

En conclusión de este trabajo se puede decir que se lograron los objetivos de la tesis, es decir, se logró evaluar un sistema de almacenamiento distribuido en términos de dos prámetros: tiempo de respuesta y confiabilidad. Además se concluye que es factible tener un buen desempeño del sistema, empleando la dispersión de información sobre redes con tecnología de uso común, como lo es Fast Ethernet. Además, un resultado que no es tan evidente, es el que se obtuvo al realizar la evaluación de la confiabilidad, pues en principio se creería que aumentando el número de nodos en la red de almacenamiento, aparte de obtener un buen desempeño en cuanto tiempo de respuesta, se obtendría un mayor tiempo de vida del sistema al tener un mayor número de computadoras para procesar y recuperar los contenidos de los nodos que fallen; sin embargo al aumentar este número de computadoras se incrementa la tasa de fallas.

Durante el desarrollo de este proyecto se identificaron algunas líneas de investigación que pueden servir para futuros proyectos. Una de ellas es realizar la verificación formal de cada uno de los protocolos planteados para los modelos de tiempo de respuesta, así como de los modelos de confiabilidad. Otra línea a investigar es el estudio del impacto, de otros

esquemas de almacenamiento, sobre el tiempo de vida sistema. Por último, se puede realizar una investigación sobre el impacto de variar los parámetros del IDA, es decir, variar la cantidad de dispersos que se generan y con cuántos se recupera el archivo original.

Con base en los resultados obtenidos en este trabajo de tesis se publicó el artículo "Service Policies for a Storage Services Dispatcher in a Distributed Fault-Tolerant Storage Network and their Performance Evaluation" [20].

Apéndice A
Fórmula de M. Stifel

En este apéndice se explica la razón por la cual el numero de comités en los que participa un nodo es igual al coeficiente binomial $\binom{v-1}{k-1}$.

Partiendo de la idea de que el total de subconjuntos es la unión de aquellos en los que aparece un elemento y aquellos donde éste no aparece. M. Stifel desarrolló la siguiente fórmula:

$$\binom{n}{k} = \binom{n-1}{k-1} + \binom{n-1}{k} \tag{A.1}$$

De la ecuación A.1, se puede observar que $\binom{n-1}{k-1}$ representa el número de subconjuntos de tamaño k, en los que aparece un elemento y $\binom{n-1}{k}$, en los que no aparece. En el caso de esta tesis el número de elementos en el conjunto total V es v, donde v es el número de máquinas en la red de almacenamiento (nodo) y cada subconjunto de k elementos generado a partir de V, es un comité. Por lo tanto sustituyendo $n = v$ en A.1 se tiene la siguiente ecuación:

$$\binom{v}{k} = \binom{v-1}{k-1} + \binom{v-1}{k} \tag{A.2}$$

Así el número de comités en los que participa un nodo es igual a $\binom{v-1}{k-1}$.

Referencias

[1] S. Mullender, *Distributed Systems (2nd Ed.)*, ACM Press/Addison-Wesley Publishing Co., Ed. 1993, pp 420-423.

[2] Y. Lyuu, *Information Dispersal and Parallel Computation*, Cambridge University Press, 1992, pp. 9-10.

[3] P. N. Yianilos and S. Sobti, *The Envolving Field of Distributed Storage*, IEEE Computing, September-October 2001, pp. 35-39.

[4] A. Rowstron and P. Druschel, *Storage management and caching in PAST, a large-scale, persistent peer-to-peer storage utility*, SOSP 2001, ACM.

[5] A.V. Goldberg and P.N. Yianilos, *Towards an Archival Intermemory*, Proc. IEEE Int'l Forum on Research and Technology Advances in Digital Libraries (ADL 98), IEEE Computer Soc. Press, 1998, pp. 147-156.

[6] Y. Chen et al., *A Prototype Implementation of Archival Intermemory*, Proc. Fourth ACM Conf. Digital Libraries (DL 99), ACM Press, 1999.

[7] A. Adya, et al, *FARSITE: Federated, Available, and Reliable Storage for an Incompletely Trusted Environment*, 5th OSDI, Dec 2002.

[8] V. Bilicki, *LanStore: a highly distributed reliable filestorage system*, .NET technologies conference proceedings, 2005.

[9] F. Chang, et al, *Bigtable: A Distributed Storage System for Structured Data*, OSDI'06, pp. 205-218.

[10] G. Caronni, R. Rom, G. Scott, *Celeste: An Automatic Storage System*, Sun Microsystems Laboratories, www.sun.com/products-n-solutions/ edu/whitepapers/pdf/, 2004.

[11] J. Kubiatowicz, et al, *OceanStore: An Architecture for Global-Scale Persistent Storage*, ASPLOS, 2000.

[12] H. Xia, A. A. Chien, *RobuSTore: Robust Performance for Distributed Storage Systems*, In 14th NASA Goddard - 23nd IEEE Conference on Mass Storage Systems and Technologies (MSST2006), 2006.

[13] A. Paul, S. Adhikari, U. Ramachandran, *Design of a Secure and Fault Tolerant Environment for Distributed Storage*, Georgia Institute of Technology, 2004.

[14] C. Gladwin. http://www.cleversafe.com, 2006.

[15] R. Marcelín-Jiménez, S. Rajsbaum and B. Stevens, *Ciclyc Storage for Fault-Tolerant Distributed Executions*, IEEE Journal of Parallel and Distributed Systems, vol. 17, no. 9, 2006, pp. 1028-1036.

[16] M.O. Rabin, *Efficient Dispersal of Information for Security, Load, Balancing and Fault-Tolerance*, ACM , vol. 36, no. 2, 1989,pp. 335-348.

[17] R. Marcelín, *Almacenamiento Distribuido Tolerante a Fallas*, Tesis de Doctorado, UNAM, 2004, pp. 31-38.

[18] A. Varga, *OMNeT++: Discrete Event Simulation System, User Manual*, March 2005.

[19] D. Gross and C. M. Harris, *Fundamentals of Queueing Theory*, 3a edición, Wiley-Interscience, 1998. pp. 209-212.

[20] M. Quezada-Naquid, R. Marcelín-Jiménez, M. López-Guerrero, *Service Policies for a Storage Services Dispatcher in a Distributed Fault-Tolerant Storage Network and their Performance Evaluation*, In the IEEE 20th Canadian Conference on Electrical and Computer Engineering (CCECE 2007), pp. 231-234. Vancouver, Canada. April 2007. ISBN 1-4244-1021-5.

Evaluación de Desempeño de un Sistema de Almacenamiento Distribuido

Idónea Comunicación de Resultados para obtener el grado de

MAESTRO EN CIENCIAS
(CIENCIAS Y TECNOLOGÍAS DE LA INFORMACIÓN)

P R E S E N T A

Ing. Moisés Quezada Naquid

Asesores:
Dr. Miguel López Guerrero
Dr. Ricardo Marcelín Jiménez

16 de octubre de 2007

UNIVERSIDAD AUTÓNOMA METROPOLITANA

Casa abierta al tiempo

Evaluación de Desempeño de un Sistema de Almacenamiento Distribuido

Para obtener el grado de
Maestro en Ciencias
(Ciencias y Tecnologías de la Información)

P R E S E N T A

Ing. Moisés Quezada Naquid

Asesores

Dr. Miguel López Guerrero
Dr. Ricardo Marcelín Jiménez

Sinodales

Presidente: Dr. Héctor Benítez Pérez
Secretario: Dr. Ricardo Marcelín Jiménez
Vocal: Dra. Elizabeth Pérez Cortés

16 de Octubre de 2007