



(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2014년04월04일
(11) 등록번호 10-1379848
(24) 등록일자 2014년03월25일

(51) 국제특허분류(Int. Cl.)
G06F 21/78 (2013.01) G06F 15/16 (2006.01)
(21) 출원번호 10-2013-0016390
(22) 출원일자 2013년02월15일
심사청구일자 2013년02월15일
(56) 선행기술조사문헌
KR1020100122197 A
KR1020110041193 A
KR1020110109610 A

(73) 특허권자
서울대학교산학협력단
서울특별시 관악구 관악로 1 (신림동)
(72) 발명자
박영훈
서울 영등포구 당산로 214, 405동 2001호 (당산동5가, 삼성래미안4차아파트)
서승우
서울 관악구 관악로 1, 뉴미디어통신공동연구소 132동 405호 (신림동, 서울대학교)
(74) 대리인
김 순 영, 김영철

전체 청구항 수 : 총 14 항

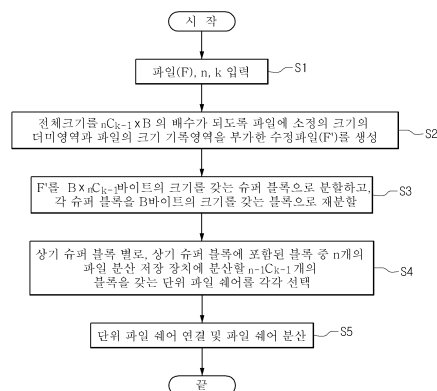
심사관 : 구본재

(54) 발명의 명칭 일정 수 이상의 파일 셰어로 복구 가능한 파일 분산 관리 장치 및 그 방법

(57) 요약

일정 수 이상의 파일 셰어로 복구 가능한 파일 분산 관리 장치 및 그 방법이 개시된다. 파일로부터 생성한 복수의 파일 셰어를 n 개의 파일 분산 저장 장치에 분산하고, 임의의 k 개($0 < k \leq n$)의 파일 분산 저장 장치에 저장된 복수의 파일 셰어를 획득하는 경우에만 파일을 복원할 수 있는 파일 분산 관리 방법에 있어서, 파일을 $B * {}_nC_{k-1}$ 바이트의 크기를 갖는 슈퍼 블록으로 분할하고, 각 슈퍼 블록을 B 바이트의 크기를 갖는 블록으로 재분할하는 단계; 상기 슈퍼 블록 별로, 슈퍼 블록에 포함된 블록 중 n 개의 파일 분산 저장 장치에 분산할 ${}_{n-1}C_{k-1}$ 개의 블록을 갖는 단위 파일 셰어를 각각 선택하는 단계; 및 n 개의 파일 분산 저장 장치에 분산되도록 선택된 단위 파일 셰어를 단위 파일 셰어가 포함되는 슈퍼 블록 순서대로 연결한 파일 셰어를 n 개의 파일 분산 저장 장치 각각에 분산하는 단계를 포함하는, 파일 분산 관리 방법이 개시된다.

대표도 - 도3



특허청구의 범위

청구항 1

파일로부터 생성한 복수의 파일 셰어를 n 개의 파일 분산 저장 장치에 분산하는 파일 분산 관리 방법에 있어서, 파일을 $B * {}_nC_{k-1}$ 바이트(단, k 은 0보다 크고 n 이하인 소정의 정수)의 크기를 갖는 슈퍼 블록들로 분할하고, 상기 슈퍼 블록들 각각을 B 바이트의 크기를 갖는 블록들로 재분할하는 단계;

상기 슈퍼 블록들 마다, 상기 슈퍼 블록들에 포함된 블록들 중 상기 n 개의 파일 분산 저장 장치에 분산할 ${}_{n-1}C_{k-1}$ 개의 블록들을 갖는 단위 파일 셰어들을 각각 선택하는 단계; 및

상기 슈퍼 블록들 마다 선택된 단위 파일 셰어들을 상기 단위 파일 셰어가 포함되는 슈퍼 블록 순서대로 연결하여 복수의 파일 셰어를 생성하고, 생성된 상기 복수의 파일 셰어를 상기 n 개의 파일 분산 저장 장치에 분산하는 단계를 포함하고,

상기 분산된 복수의 파일 셰어로부터 상기 파일을 복원하기 위해서는, 상기 분산된 복수의 파일 셰어들 중 적어도 상기 k 개 이상의 임의의 파일 셰어들을 획득하여야 하는, 파일 분산 관리 방법.

청구항 2

제1항에 있어서,

상기 슈퍼 블록들 마다, 상기 슈퍼 블록들에 포함된 블록들 중 상기 n 개의 파일 분산 저장 장치에 분산할 ${}_{n-1}C_{k-1}$ 개의 블록들을 갖는 단위 파일 셰어들을 각각 선택하는 단계는,

집합 $\{1, 2, \dots, n\}$ 의 원소 $n-k+1$ 개짜리 부분집합의 집합 T 와 집합 $W=\{1,2,3,\dots, {}_nC_{k-1}\}$ 및 일대일대응($f:T \rightarrow W$)을 생성하는 단계를 더 포함하는, 파일 분산 관리 방법.

청구항 3

제1항에 있어서,

상기 슈퍼 블록들 마다, 상기 슈퍼 블록들에 포함된 블록들 중 상기 n 개의 파일 분산 저장 장치에 분산할 ${}_{n-1}C_{k-1}$ 개의 블록들을 갖는 단위 파일 셰어들을 각각 선택하는 단계는,

상기 n 개의 파일 분산 저장 장치에 분산할 단위 파일 셰어들 중 상기 k 개의 임의의 단위 파일 셰어들은 원래 ${}_nC_{k-1}$ 개의 파일 블록을 포함하지만, 상기 k 개보다 적은 수의 단위 파일 셰어들은 원래 ${}_nC_{k-1}$ 개의 파일 블록 모두를 포함하지 않도록 ${}_{n-1}C_{k-1}$ 개의 블록을 선택하는 단계를 더 포함하는, 파일 분산 관리 방법.

청구항 4

제 3항에 있어서,

상기 슈퍼 블록들 마다 선택된 단위 파일 셰어들을 상기 단위 파일 셰어가 포함되는 슈퍼 블록 순서대로 연결하여 상기 복수의 파일 셰어를 생성하고, 생성된 상기 복수의 파일 셰어를 상기 n 개의 파일 분산 저장 장치에 분산하는 단계는,

집합 $\{1, 2, \dots, n\}$ 의 원소 $n-k+1$ 개짜리 부분집합의 집합 T 와 집합 $W=\{1,2,3,\dots, {}_nC_{k-1}\}$ 및 일대일대응($f:T \rightarrow W$)을 생성하는 단계; 및

$i=1, 2, \dots, n$ 에 대하여, T 의 원소 중 i 를 포함하는 상기 $\{1,2,\dots,n\}$ 의 ${}_{n-1}C_{k-1}$ 개의 부분집합을 선택하여, 이를

$S(1), S(2), \dots, S({}_{n-1}C_{k-1})$ 라 했을 때, i 번째 파일 분산 저장장치에 분산되도록 상기 ${}_{n-1}C_{k-1}$ 개의 블록 중 $f(S(1))$ 번째 블록, $f(S(2))$ 번째 블록, $\dots$, $f(S({}_{n-1}C_{k-1}))$ 번째 블록을 i 번째 파일 분산 저장 장치에 저장하도록 선택하는 단계를 더 포함하는, 파일 분산 관리 방법.

청구항 5

제1항에 있어서,

상기 파일을 $B * {}_{n-1}C_{k-1}$ 바이트(단, k 은 0보다 크고 n 이하인 소정의 정수)의 크기를 갖는 슈퍼 블록들로 분할하고, 상기 슈퍼 블록들 각각을 B 바이트의 크기를 갖는 블록들로 재분할하는 단계는,

상기 파일에 추가영역을 부가하여 수정 파일을 생성하는 단계; 및

상기 수정 파일을 B 바이트의 크기를 가지는 블록으로 분할하는 단계를 더 포함하는, 파일 분산 관리 방법.

청구항 6

파일을 $B * {}_{n-1}C_{k-1}$ 바이트의 크기를 갖는 슈퍼 블록으로 분할하고, 각 슈퍼 블록을 B 바이트의 크기를 갖는 블록으로 재분할하는 단계; 상기 슈퍼 블록 별로, 상기 슈퍼 블록에 포함된 블록 중 n 개의 파일 분산 저장 장치에 분산할 ${}_{n-1}C_{k-1}$ 개의 블록을 갖는 단위 파일 셰어를 각각 선택하는 단계; 및 n 개의 파일 분산 저장 장치에 분산되도록 선택된 상기 단위 파일 셰어를 상기 단위 파일 셰어가 포함되는 상기 슈퍼 블록 순서대로 연결한 파일 셰어를 n 개의 파일 분산 저장 장치 각각에 분산하는 단계를 포함하여, 파일로부터 생성한 복수의 파일 셰어를 n 개의 파일 분산 저장 장치에 분산하고, 임의의 k 개(단, k 은 0보다 크고 n 이하인 소정의 정수)의 파일 분산 저장 장치에 저장된 복수의 파일 셰어를 획득하는 경우에만 파일을 복원할 수 있는 파일 분산 관리 방법을 실행하기 위한 컴퓨터 프로그램용 기록매체.

청구항 7

파일을 $B * {}_{n-1}C_{k-1}$ 바이트의 크기를 갖는 슈퍼 블록으로 분할하고, 각 슈퍼 블록을 B 바이트의 크기를 갖는 블록으로 재분할하는 단계; 상기 슈퍼 블록 별로, 상기 슈퍼 블록에 포함된 블록 중 n 개의 파일 분산 저장 장치에 분산할 ${}_{n-1}C_{k-1}$ 개의 블록을 갖는 단위 파일 셰어를 각각 선택하는 단계; 및 n 개의 파일 분산 저장 장치에 분산되도록 선택된 상기 단위 파일 셰어를 상기 단위 파일 셰어가 포함되는 상기 슈퍼 블록 순서대로 연결한 파일 셰어를 n 개의 파일 분산 저장 장치 각각에 분산하는 단계를 포함하여, 파일로부터 생성한 복수의 파일 셰어를 n 개의 파일 분산 저장 장치에 분산하고, 임의의 k 개($0 < k \leq n$)의 파일 분산 저장 장치에 저장된 복수의 파일 셰어를 획득하는 경우에만 파일을 복원할 수 있는 파일 분산 관리 방법을 실행하기 위한 컴퓨터 프로그램을 기록하는 컴퓨터 판독 가능한 기록매체.

청구항 8

파일로부터 생성한 복수의 파일 셰어를 n 개의 파일 분산 저장 장치에 분산하는 파일 분산 관리 장치에 있어서,

파일을 $B * {}_{n-1}C_{k-1}$ 바이트의 크기(단, k 은 0보다 크고 n 이하인 소정의 정수)를 갖는 슈퍼 블록들로 분할하고, 상기 슈퍼 블록들 각각을 B 바이트의 크기를 갖는 블록들로 재분할하는 파일 분할부;

상기 슈퍼 블록들 마다, 상기 슈퍼 블록들에 포함된 블록들 중 상기 n 개의 파일 분산 저장 장치에 분산할 ${}_{n-1}C_{k-1}$ 개의 블록들을 갖는 단위 파일 셰어들을 각각 선택하는 파일 셰어 선택부;

상기 슈퍼 블록들 마다 선택된 상기 단위 파일 셰어들을 상기 단위 파일 셰어들이 포함되는 슈퍼 블록 순서대로 연결하여 복수의 파일 셰어들을 생성하고, 생성된 상기 복수의 파일 셰어들을 상기 n 개의 파일 분산 저장 장치

에 분산하는 파일 셰어 분산부를 포함하고,

상기 분산된 복수의 파일 셰어들로부터 상기 파일을 복원하기 위해서는, 상기 분산된 복수의 파일 셰어들 중 적어도 상기 k개 이상의 임의의 파일 셰어들을 획득하여야 하는, 파일 분산 관리 장치.

청구항 9

제8항에 있어서,

상기 파일 셰어 선택부는,

집합 $\{1, 2, \dots, n\}$ 의 원소 $n-k+1$ 개짜리 부분집합의 집합 T 와 집합 $W=\{1,2,3,\dots, {}_nC_{k-1}\}$ 및 일대일대응($f:T \rightarrow W$)을 생성하는 일대일 대응 관리부를 더 포함하는, 파일 분산 관리 장치.

청구항 10

제8항에 있어서,

상기 파일 셰어 선택부는,

상기 n개의 파일 분산 저장 장치에 분산할 단위 파일 셰어들 중 상기 k개의 임의의 단위 파일 셰어들은 원래 ${}_nC_{k-1}$ 개의 파일 블록을 포함하지만, 상기 k개보다 적은 수의 단위 파일 셰어들은 원래 ${}_nC_{k-1}$ 개의 파일 블록 모두를 포함하지 않도록 ${}_nC_{k-1}$ 개의 블록을 선택하는, 파일 분산 관리 장치.

청구항 11

제10항에 있어서,

상기 블록 선택부는,

집합 $\{1, 2, \dots, n\}$ 의 원소 $n-k+1$ 개짜리 부분집합의 집합 T 와 집합 $W=\{1,2,3,\dots, {}_nC_{k-1}\}$ 및 일대일대응($f:T \rightarrow W$)을 생성하는 일대일대응 관리부를 포함하며,

$i=1, 2, \dots, n$ 에 대하여, T 의 원소 중 i 를 포함하는 상기 $\{1,2,\dots,n\}$ 의 ${}_nC_{k-1}$ 개의 부분집합을 선택하여, 이를 $S(1), S(2), \dots, S({}_nC_{k-1})$ 라 했을 때, i 번째 파일 분산 저장장치에 분산되도록 상기 ${}_nC_{k-1}$ 개의 블록 중 $f(S(1))$ 번째 블록, $f(S(2))$ 번째 블록, $\dots, f(S({}_nC_{k-1}))$ 번째 블록을 i 번째 파일 분산 저장 장치에 저장하도록 선택하는, 파일 분산 관리 장치.

청구항 12

제8항에 있어서,

상기 파일 분할부는,

상기 파일에 추가영역을 부가하여 수정 파일을 생성하는 수정 파일 생성부를 포함하고,

상기 수정 파일을 B바이트의 크기를 가지는 블록으로 분할하는, 파일 분산 관리 장치.

청구항 13

제8항에 있어서,

상기 수정 파일 생성부는,

상기 파일에 소정의 크기의 더미영역과 파일의 크기 기록영역을 부가하되, 상기 파일의 크기, 상기 더미영역의 크기 및 상기 파일의 크기 기록 영역의 합이 $nC_{k-1} \times$ 블록의 크기(B)의 배수가 되도록 더미영역의 크기를 결정하는 파일 분산 관리 장치.

청구항 14

파일로부터 생성한 복수의 파일 셰어를 n개의 파일 분산 저장 장치에 분산하는 파일 분산 관리 시스템에 있어서,

파일을 소정의 크기를 가지는 블록들로 분할하고, 상기 블록들을 순서대로 nC_{k-1} 개(단, k은 0보다 크고 n이하인 소정의 정수)씩 포함하는 슈퍼 블록들을 생성하는 파일 분할부, 상기 슈퍼 블록들 마다, 상기 슈퍼 블록들에 포함된 블록들 중 n개의 파일 분산 저장 장치에 분산할 $n-1C_{k-1}$ 개의 블록들을 갖는 단위 파일 셰어를 각각 선택하는 파일 셰어 선택부, 상기 단위 파일 셰어를 모두 연결한 파일 셰어를 n개의 파일 분산 저장 장치에 분산하는 파일 셰어 분산부를 포함하는, 파일 분산 관리 장치; 및

n개의 파일 분산 저장 장치를 포함하고,

상기 n개의 파일 분산 저장 장치에 저장된 파일 셰어들로부터 상기 파일을 복원하기 위해서는, 상기 n개의 파일 분산 저장 장치에 저장된 파일 셰어들 중 적어도 상기 k개 이상의 파일 셰어들을 획득하여야 하는, 파일 분산 관리 시스템.

명세서

기술분야

[0001] 실시예들은 파일 분산 관리 장치 및 그 방법에 관한 것이다. 보다 구체적으로, 일정 수 이상의 파일 셰어로 복구 가능한 파일 분산 관리 장치 및 그 방법에 관한 것이다.

배경기술

[0002] 비밀 정보를 보관하는 경우, 비밀 정보가 분실되거나, 파괴될 리스크는 항상 존재한다. 이와 동시에 비밀정보가 도난될 수 있는 리스크 역시 존재한다. 분실 또는 파괴의 리스크는 비밀 정보를 복수 곳의 장소에 보관해 둬으로써 줄일 수 있지만, 이 경우 도난의 리스크가 증가하는 단점이 있다. 이들 리스크를 함께 해결하는 방법의 하나로써 비밀 분산법(Secret Sharing)이 제시되었다.

[0003] 비밀 분산법은 비밀 정보 MSK로부터 복수의 셰어 정보 SH(1), ..., SH(N)를 생성하고, 이들을 복수의 분산 관리 장치 PA(1), ..., PA(N)에 분산하여 관리시키고, 이들 셰어 정보 SH(1), ..., SH(N) 중 소정 수 이상의 정보를 얻을 수 있었던 경우에만, 비밀 정보 MSK를 복원할 수 있는 방식이다.

[0004] 기존 파일 저장 시스템에서는 파일을 하나의 저장 장치에 저장하는 방식이었다. 하지만, 이 경우 전술한 비밀 정보와 같이, 저장된 파일의 기밀성, 무결성 및 가용성에 문제가 될 수 있다. 저장 장치에 바이러스 또는 웜이 감염되었거나 저장 장치를 분실했을 경우, 저장된 파일의 내용이 노출될 수 있다. 또한, 파일의 내용이 변조되거나 저장장치의 분실 혹은 실수 등으로 파일을 잃어버렸을 경우, 무결성과 가용성에 문제가 있을 수 있다.

[0005] 비밀 분산법은 다양한 형태로 제시되고 있었지만, 파일 저장 시스템의 특성에 맞게 적절한 형태로 제안되지 못하는 실정이다. 기존의 비밀 분산법은 비밀 정보를 복구하는데 복잡한 계산을 요구하는데, 파일에 적용할 경우 매우 많은 계산량을 요구한다. 이를 해결하기 위하여, 이 특허에서는 경량화된 비밀 분산법을 제안할 것이다.

선행기술문헌

특허문헌

[0006] (특허문헌 0001) 한국 공개특허 제2012-34156호

발명의 내용

해결하려는 과제

[0007] 본 발명의 일 측면에 의하면, 파일 시스템의 파일 저장시 파일 내용의 순서를 무작위로 재배열하여 기밀성을 향상시킬 수 있다.

[0008] 본 발명의 다른 측면에 의하면, 파일 시스템의 파일 저장시 파일을 분할하고, 그 중 일부를 이용하여 복구할 수 있게 함으로써, 무결성과 가용성을 모두 보장할 수 있다.

과제의 해결 수단

[0009] 본 발명의 일 측면에 의하면, 파일로부터 생성한 복수의 파일 셰어를 n 개의 파일 분산 저장 장치에 분산하고, 임의의 k 개($0 < k \leq n$)의 파일 분산 저장 장치에 저장된 복수의 파일 셰어를 획득하는 경우에만 파일을 복원할 수 있는 파일 분산 관리 방법에 있어서, 파일을 $B * {}_nC_{k-1}$ 바이트의 크기를 갖는 슈퍼 블록(super block)으로 분할하고, 각 슈퍼 블록을 B 바이트의 크기를 갖는 블록으로 재분할하는 단계; 상기 슈퍼 블록 별로, 상기 슈퍼 블록에 포함된 블록 중 n 개의 파일 분산 저장 장치에 분산할 ${}_{n-1}C_{k-1}$ 개의 블록을 갖는 단위 파일 셰어를 각각 선택하는 단계; 및 n 개의 파일 분산 저장 장치에 분산되도록 선택된 상기 단위 파일 셰어를 상기 단위 파일 셰어가 포함되는 상기 슈퍼 블록 순서대로 연결한 파일 셰어를 n 개의 파일 분산 저장 장치 각각에 분산하는 단계를 포함하는, 파일 분산 관리 방법이 제공된다.

[0010] 본 발명의 다른 측면에 의하면, 파일을 $B * {}_nC_{k-1}$ 바이트의 크기를 갖는 슈퍼 블록으로 분할하고, 각 슈퍼 블록을 B 바이트의 크기를 갖는 블록으로 재분할하는 단계; 상기 슈퍼 블록 별로, 상기 슈퍼 블록에 포함된 블록 중 n 개의 파일 분산 저장 장치에 분산할 ${}_{n-1}C_{k-1}$ 개의 블록을 갖는 단위 파일 셰어를 각각 선택하는 단계; 및 n 개의 파일 분산 저장 장치에 분산되도록 선택된 상기 단위 파일 셰어를 상기 단위 파일 셰어가 포함되는 상기 슈퍼 블록 순서대로 연결한 파일 셰어를 n 개의 파일 분산 저장 장치 각각에 분산하는 단계를 포함하여, 파일로부터 생성한 복수의 파일 셰어를 n 개의 파일 분산 저장 장치에 분산하고, 임의의 k 개($0 < k \leq n$)의 파일 분산 저장 장치에 저장된 복수의 파일 셰어를 획득하는 경우에만 파일을 복원할 수 있는 파일 분산 관리 방법을 실행하기 위한 컴퓨터 프로그램을 기록하는 컴퓨터 판독 가능한 기록매체가 제공된다.

[0011] 본 발명의 또 다른 측면에 의하면, 파일로부터 생성한 복수의 파일 셰어를 n 개의 파일 분산 저장 장치에 분산하고, 임의의 k 개($0 < k \leq n$)의 파일 분산 저장 장치에 저장된 복수의 파일 셰어를 획득하는 경우에만 파일을 복원할 수 있는 파일 분산 관리 장치에 있어서, 파일을 $B * {}_nC_{k-1}$ 바이트의 크기를 갖는 슈퍼 블록으로 분할하고, 각 슈퍼 블록을 B 바이트의 크기를 갖는 블록으로 재분할하는 파일 분할부; 상기 슈퍼 블록 별로, 상기 슈퍼 블록에 포함된 블록 중 n 개의 파일 분산 저장 장치에 분산할 ${}_{n-1}C_{k-1}$ 개의 블록을 갖는 단위 파일 셰어를 각각 선택하는 파일 셰어 선택부; n 개의 파일 분산 저장 장치에 분산되도록 선택된 상기 단위 파일 셰어를 상기 단위 파일 셰어가 포함되는 상기 슈퍼 블록 순서대로 연결한 파일 셰어를 n 개의 파일 분산 저장 장치 각각에 분산하는 파일 셰어 분산부를 포함하는, 파일 분산 관리 장치가 제공된다.

[0012] 본 발명의 또 다른 측면에 의하면, 파일로부터 생성한 복수의 파일 셰어를 n 개의 파일 분산 저장 장치에 분산하

고, 임의의 k 개($0 < k \leq n$)의 파일 분산 저장 장치에 저장된 복수의 파일 셰어를 획득하는 경우에만 파일을 복원할 수 있는 파일 분산 관리 시스템에 있어서, 파일을 소정의 크기를 가지는 블록으로 분할하고, 상기 블록을 순서대로 nC_{k-1} 개씩 포함하는 슈퍼 블록을 생성하는 파일 분할부, 상기 슈퍼 블록 별로, 상기 슈퍼 블록에 포함된 블록 중 n 개의 파일 분산 저장 장치에 분산할 $n-1C_{k-1}$ 개의 블록을 갖는 단위 파일 셰어를 각각 선택하는 파일 셰어 선택부, 상기 단위 파일 셰어를 모두 연결한 파일 셰어를 n 개의 파일 분산 저장 장치 각각에 분산하는 파일 셰어 분산부를 포함하는, 파일 분산 관리 장치; 및 n 개의 파일 분산 저장 장치를 포함하는, 파일 분산 관리 시스템이 제공된다.

발명의 효과

- [0013] 본 발명의 일 측면에 의하면, 하나의 파일이 여러 대의 기기에 나뉘어서 저장되므로 k 개 미만의 기기에서 데이터가 유출되었을 경우, 원래 파일의 복원이 불가능하게 하는 효과가 있다.
- [0014] 또한, k 개 이상의 기기가 있다고 하더라도, 파일 내용의 순서가 섞여 있으므로 원본 파일을 복구해내기가 매우 어렵게 하여 보안수준을 향상시킬 수 있다.
- [0015] 나아가, 기기의 분실, 실수로 삭제, 또는 바이러스 감염 등으로 인한 파일의 변조로 인하여 파일이 손상되었을 경우, 나머지 k 개의 기기를 통하여 원본 파일을 복구할 수 있는 효과가 있다.
- [0016] 또한, 기존의 비밀 분산 기술을 크기가 F 인 파일에 적용했을 경우, k 의 제곱과 F 에 비례하는 연산량이 발생한다. 이로 인하여 불필요한 계산량 소모가 발생할 우려가 있다. 이 특허에서는 조합적인 방법을 이용하여 연산량이 오로지 F 에만 비례하는 기술을 제안할 것이다.
- [0017] 즉, 본 발명의 일 측면에 의하면, 파일 시스템의 파일 저장시 파일의 기밀성, 무결성, 가용성을 모두 보장함과 동시에 분산된 파일의 복구를 빠르게 수행할 수 있는 효과가 있다.

도면의 간단한 설명

- [0018] 도 1은 본 발명의 일 실시예에 따른, 파일 분산 관리 시스템(1000) 및 그 내부 구성을 나타낸 도면이다.
- 도 2는 본 발명의 일 실시예에 따른, 파일 분산 관리 장치(100)의 내부 구성을 나타낸 도면이다.
- 도 3은 본 발명의 일 실시예에 따른, 파일 분산 관리 방법의 일 흐름을 나타낸 도면이다.
- 도 4a는 본 발명의 일 실시예에 따른, 파일(F)의 구성을 나타낸 도면이다.
- 도 4b는 본 발명의 일 실시예에 따른, 파일(F')의 구성을 나타낸 도면이다.
- 도4c는 본 발명의 일 실시예에 따른, K , SK , RK 를 나타낸 도면이다.
- 도4d는 본 발명의 일 실시예에 따른, 단위 파일 셰어를 나타낸 도면이다.
- 도4e는 본 발명의 일 실시예에 따른, 파일 셰어를 나타낸 도면이다.
- 도4f는 본 발명의 일 실시예에 따른, 복구된 단위 파일 셰어를 나타낸 도면이다.

발명을 실시하기 위한 구체적인 내용

- [0019] 후술하는 본 발명에 대한 상세한 설명은, 본 발명이 실시될 수 있는 특정 실시예를 예시로서 도시하는 첨부 도면을 참조한다. 이들 실시예는 당업자가 본 발명을 실시할 수 있기에 충분하도록 상세히 설명된다. 본 발명의 다양한 실시예는 서로 다르지만 상호 배타적일 필요는 없음이 이해되어야 한다. 예를 들어, 여기에 기재되어 있는 특정 형상, 구조 및 특성은 일 실시예에 관련하여 본 발명의 정신 및 범위를 벗어나지 않으면서 다른 실시예로 구현될 수 있다. 또한, 각각의 개시된 실시예 내의 개별 구성요소의 위치 또는 배치는 본 발명의 정신 및 범위를 벗어나지 않으면서 변경될 수 있음이 이해되어야 한다. 따라서, 후술하는 상세한 설명은 한정적인 의미로서 취하려는 것이 아니며, 본 발명의 범위는, 적절하게 설명된다면, 그 청구항들이 주장하는 것과 균등한 모든 범위와 더불어 첨부된 청구항에 의해서만 한정된다. 도면에서 유사한 참조부호는 여러 측면에 걸쳐서 동일하게

나 유사한 기능을 지칭한다.

- [0020] 도 1은 본 발명의 일 실시예에 따른, 파일 분산 관리 시스템(1000) 및 그 내부 구성을 나타낸 도면이다. 파일 분산 관리 시스템(100)은 파일 분산 관리 장치(100) 및 n 개의 파일 분산 저장 장치(10)로 구성된다.
- [0021] 파일 분산 관리 시스템(1000)은 파일로부터 생성한 복수의 파일 셰어를 n 개로 분산하여 저장함으로써, 임의의 k 개($0 < k \leq n$)의 파일 분산 저장 장치에 저장된 복수의 파일 셰어를 획득하는 경우에만 파일을 복원할 수 있도록 관리하는 시스템이다. 일 실시예에서, 파일 분산 관리 시스템(1000)은 파일 분산 관리 장치(100) 및 n 개의 파일 분산 저장 장치(10)를 포함할 수 있다.
- [0022] 파일 분산 관리 장치(100)는 임의의 k 개($0 < k \leq n$)의 파일 분산 저장 장치에 저장된 복수의 파일 셰어를 획득하는 경우에만 파일을 복원할 수 있도록, 파일로부터 생성한 복수의 파일 셰어를 n 개의 파일 분산 저장 장치(100)에 분산하는 역할을 한다. 파일 분산 관리 장치(100)에 대한 상세한 내용은 도 2를 참조하여 후술하기로 한다.
- [0023] 파일 분산 저장 장치(10)는 파일 분산 관리 장치(100)로부터 수신한 파일 셰어를 저장하는 역할을 한다. 임의의 k 개의 파일 분산 저장 장치(10)에 저장된 파일 셰어가 모이는 경우에만, 원본 파일을 복원해낼 수 있다. 파일 분산 저장 장치(10)는 특별한 기능이 없이도 각각의 파일 셰어의 저장이 분리될 수 있는 장치들로 구성되면 충분하다. 파일 분산 저장 장치(10)가 도 1상에서는 물리적으로 분리된 형태로 표현되어 있지만, 1개의 장치에서 논리적으로 분리된 형태도 가능할 것이다.
- [0024] 일 실시예에서, 파일 분산 저장 장치(10)는 파일 셰어의 수신을 위하여 파일 분산 관리 장치(100)와 통신이 가능할 수 있다.
- [0025] 도 2는 본 발명의 일 실시예에 따른, 파일 분산 관리 장치(100)의 내부 구성을 나타낸 도면이다. 파일 분산 관리 장치(100)는 파일 분할부(110), 블록 선택부(130), 파일 셰어 분산부(140), 통신부(150), 저장부(160)를 포함할 수 있으나, 이에 제한되는 것은 아니다.
- [0026] 파일 분할부(110)는 파일을 $B * {}_nC_{k-1}$ 바이트의 크기를 갖는 슈퍼 블록으로 분할하고, 각 슈퍼 블록을 B 바이트의 크기를 갖는 블록으로 재분할하는 역할을 한다. 파일 블록은 파일을 나누는 기본 단위로 그 크기는 일정하다. 일 실시예에서, 파일 블록의 단위는 2바이트(Byte)가 될 수 있다. 분할된 블록의 총 수는 ${}_nC_{k-1} * \text{블록 단위}(B) * \text{슈퍼 블록의 수}(L)$ 가 된다.
- [0027] 파일(F)의 크기는 일 실시예에서, 2바이트(Byte) 단위로 측정하며, 2바이트 단위의 파일(F)이 ${}_nC_{k-1}$ 개 * 블록의 크기(B)의 배수 형태가 아닌 경우에 대비하여, 파일 분할부(110)는 수정파일 생성부(111)를 더 포함할 수 있다.
- [0028] 수정파일 생성부(111)는 파일(F)에 추가영역을 부가하여 수정 파일(F')을 생성하는 역할을 한다. 일 실시예에서, 수정 파일 생성부(111)는 소정의 크기의 더미영역과 원래 파일의 크기 기록영역을 포함하는 추가영역을 부가할 수 있다. 일 실시예에서, 수정 파일(F')의 크기(파일의 크기, 상기 더미영역의 크기 및 상기 원래 파일의 크기 기록 영역의 합)가 ${}_nC_{k-1}$ 개 * 블록의 크기(B)의 배수 형태(슈퍼 블록의 수, L)가 되도록 더미영역의 크기를 결정할 수 있다. 더미영역은 널문자(null character)가 될 수도 있지만, 더미값이라면 충분하며 특별히 그 형태를 제한하지 아니한다. 일 실시예에서, 원래 파일의 크기 기록영역은 4바이트의 영역일 수 있다.
- [0029] 수정파일 생성부(111)는 수정 파일(F')의 크기(파일의 크기, 상기 더미영역의 크기 및 상기 원래 파일의 크기 기록 영역의 합)가 $N({}_nC_{k-1}) * \text{블록의 크기}(B)$ 의 배수(슈퍼 블록의 수, L)형태가 되도록 더미영역의 크기를 결정

할 수 있다. 일 실시예에서, 원래 파일의 크기 기록영역의 크기는 4바이트이며, 더미영역의 크기는 $N * B * L$ - size(F)-4바이트로 계산될 수 있다. 여기서 L 은 $\left\lceil \frac{\text{size}(F) + 4}{N * B} \right\rceil$ 이다. 일 실시예에서, $n=5$, $k=3$, F의 크기=38바이트, 블록의 크기=2바이트이고, F는 도4a와 같다고 가정하자. $L = \left\lceil \frac{38 + 4}{2 * 10} \right\rceil = 3$ 이므로, F'의 처음 38바이트는 F의 내용이 그대로 들어가며, 그 다음 18바이트는 널문자(더미영역)가, 그리고 마지막 4바이트는 파일 크기 정보가 들어가서 총 60바이트의 파일이 된다. F'는 도4b와 같다(단, #0은 널문자).

[0030] 블록 선택부(130)는 상기 슈퍼 블록 별로, 상기 슈퍼 블록에 포함된 블록 중 n 개의 파일 분산 저장 장치에 분산할 ${}_{n-1}C_{k-1}$ 개의 블록을 갖는 단위 파일 셰어를 각각 선택하는 역할을 한다.

[0031] 일 실시예에서, 블록 선택부(130)는 ${}_nC_{k-1}$ 개의 연속한 파일 블록을 블록 단위로 나누어서 n 개의 단위 파일 셰어를 생성하기 위하여, 나뉜 n 개의 단위 파일 셰어 중 임의의 k 개의 단위 파일 셰어가 원래 N 개의 파일 블록을 포함하고 있지만, 임의의 $k-1$ 개의 단위 파일 셰어는 원래 N 개의 파일 블록 모두를 포함하고 있지는 않도록 ${}_{n-1}C_{k-1}$ 개의 블록을 선택할 수 있다. 이를 위해, ${}_nC_{k-1}$ 개의 블록 각각이 서로 다른 임의의 조합의 k 개의 파일 분산 저장 장치에 분산되도록, n 개의 파일 분산 저장 장치에 분산할 ${}_{n-1}C_{k-1}$ 개의 블록을 각각 선택한다.

[0032] 일 실시예에서, 블록 선택부(130)는, 집합 $\{1, 2, \dots, n\}$ 의 원소 $n-k+1$ 개짜리 부분집합의 집합 T 와 집합 $W=\{1, 2, 3, \dots, {}_nC_{k-1}\}$ 및 일대일대응($f: T \rightarrow W$)을 생성하는 일대일대응 관리부(131)를 포함하며, $i=1, 2, \dots, n$ 에 대하여, T 의 원소 중 i 를 포함하는 상기 $\{1, 2, \dots, n\}$ 의 ${}_{n-1}C_{k-1}$ 개의 부분집합을 선택하여, 이를 $S(1), S(2), \dots, S({}_{n-1}C_{k-1})$ 라 했을 때, i 번째 파일 분산 저장장치(10)에 분산되도록 상기 ${}_nC_{k-1}$ 개의 블록 중 $f(S(1))$ 번째 블록, $f(S(2))$ 번째 블록, $\dots, f(S({}_{n-1}C_{k-1}))$ 번째 블록을 선택할 수 있다.

[0033] 일 실시예에서, $n=5$, $k=3$, F(단위 파일 셰어)={1, 2, 3, 4, 5, 6, 7, 8, 9, 10}일 때, 집합 $\{1, 2, 3, 4, 5\}$ 의 원소 $5-3+1=3$ 개짜리 부분집합의 집합 T 는 수학적 1의 형태, W 는 수학적 2의 형태를 갖고, T 와 W 는 10!의 가 지수를 갖는 일대일 대응 관계($f: T \rightarrow W$)를 갖는다.

[0034] [수학적 1]

[0035] $T = \{\{1, 2, 3\}, \{1, 2, 4\}, \{1, 2, 5\}, \{1, 3, 4\}, \{1, 3, 5\}, \{1, 4, 5\}, \{2, 3, 4\}, \{2, 3, 5\}, \{2, 4, 5\}, \{3, 4, 5\}\}$

[0036] [수학적 2]

[0037] $W = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$

[0038] $i=1, 2, 3, 4, 5$ 에 대하여 T 의 원소 중 i 를 포함하는 상기 $\{1, 2, 3, 4, 5\}$ 의 ${}_5C_{3-1}=3$ 개의 부분집합을 선택하여, 이를 $S(1), S(2), S(3)$ 라 했을 때, i 번째 파일 분산 저장장치(10)에 분산되도록 상기 ${}_5C_{3-1}=10$ 개의 블록 중 $f(S(1))$ 번째 블록, $f(S(2))$ 번째 블록, $f(S(3))$ 번째 블록을 i 번째 파일 분산 저장 장치에 저장하도록 선택하면, 단위 파일 셰어는 수학적 3과 같다.

[0039] [수학적 3]

[0040] $f_1=\{1, 2, 3, 4, 5, 6\}$

- [0041] $f_2=\{1, 2, 3, 7, 8, 9\}$
- [0042] $f_3=\{1, 4, 5, 7, 8, 10\}$
- [0043] $f_4=\{2, 4, 6, 7, 9, 10\}$
- [0044] $f_5=\{3, 5, 6, 8, 9, 10\}$
- [0045] 파일 셰어 분산부(140)는 n 개의 파일 분산 저장 장치에 분산되도록 선택된 상기 단위 파일 셰어를 상기 단위 파일 셰어가 포함되는 상기 슈퍼 블록 순서대로 연결한 파일 셰어를 n 개의 파일 분산 저장 장치 각각에 분산하는 역할을 한다. 실시예에 따라, 통신부(150)를 통해서 파일 셰어를 파일 분산 저장 장치(10) 각각에 분산할 수도 있다.
- [0046] 통신부(150)는 파일 분산 관리 장치(100)와 외부 객체와의 통신을 수행하는 역할을 한다. 파일 셰어 분산부(140)는 외부 객체와 통신할 수도 있지만, 통신부(150)의 지원을 받아 외부 객체와의 통신을 수행할 수도 있다. 통신 방법은 객체와 객체가 네트워킹할 수 있는 모든 통신 방법을 포함할 수 있을 것이며, 통신기능이 있다면 유선/무선 통신, 3G, 4G, 혹은 그 이후의 방법의 종류에 제한되지 아니한다.
- [0047] 저장부(160)는 파일 분산 관리 장치(100) 내부의 데이터 처리물을 저장하는 역할을 한다. 구체적으로, 파일 분할부(110), 블록 선택부(130)의 기능 수행 과정 중 또는 기능 수행 과정을 통해 얻어진 제반 정보를 저장할 수 있다.
- [0048] 도 3은 본 발명의 일 실시예에 따른, 파일 분산 관리 방법의 일 흐름을 나타낸 도면이다. 먼저 파일(F)의 정보, 파일 분산 저장 장치의 개수(n), 원래 파일을 복구할 수 있게 하기 위한 최소 파일 셰어의 수(k)을 입력한다(S1). 바이트 단위의 파일(F)이 nC_{k-1} 개 * B(블록의 크기)의 배수(슈퍼 블록 수) 형태가 아닌 경우에 대비하여, 전체크기를 $nC_{k-1} * B$ 의 배수가 되도록 파일에 소정의 크기의 더미영역과 원래 파일의 크기 기록영역을 부가한 수정파일(F')를 생성한다(S2). 이후, F'를 $B * nC_{k-1}$ 바이트의 크기를 갖는 슈퍼 블록으로 분할하고, 각 슈퍼 블록을 B바이트의 크기를 갖는 블록으로 재분할한다.(S3). 상기 슈퍼 블록 별로, 상기 슈퍼 블록에 포함된 블록 중 n 개의 파일 분산 저장 장치에 분산할 $n-1C_{k-1}$ 개의 블록을 갖는 단위 파일 셰어를 각각 선택한다(S4). S4과정에서, 집합 $\{1, 2, \dots, n\}$ 의 원소 $n-k+1$ 개짜리 부분집합의 집합 T와 집합 $W=\{1,2,3,\dots, nC_{k-1}\}$ 및 일대일대응($f:T \rightarrow W$)을 생성하고, $i=1, 2, \dots, n$ 에 대하여, T의 원소 중 i 를 포함하는 상기 $\{1,2,\dots,n\}$ 의 $n-1C_{k-1}$ 개의 부분집합을 선택하여, 이를 $S(1), S(2), \dots, S(n-1C_{k-1})$ 라 했을 때, i 번째 파일 분산 저장장치(10)에 분산되도록 상기 nC_{k-1} 개의 블록 중 $f(S(1))$ 번째 블록, $f(S(2))$ 번째 블록, $\dots, f(S(n-1C_{k-1}))$ 번째 블록을 선택할 수 있다. S4과정에서 선택된 단위 파일 셰어를 단위 파일 셰어가 포함되는 슈퍼 블록 순서대로 연결한 파일 셰어를 n 개의 파일 분산 저장 장치(10) 각각에 분산한다(S5). 분산이 완료된 이후에, k 개의 파일 분산 관리 장치(10)에서 획득한 파일 셰어를 조합하면, 원래의 파일 블록 모두를 포함하고 있으므로, 원본 파일을 복원할 수 있다.
- [0049] 본 발명의 서술한 실시예들은 서플링 키와, 리커버리 키를 이용하여 구현될 수도 있다. 이하에는, 실제 구현 사례를 설명한다.
- [0050] **가. 구현 사례 1**
- [0051] (1) 개념 정의 단계
- [0052] n : 파일 셰어의 수

- [0053] k : 원래 파일을 복구할 수 있게 하기 위한 최소 파일 셰어의 수 ($0 < k \leq n$)
- [0054] N : ${}_nC_{k-1}$
- [0055] M : ${}_{n-1}C_{k-1}$
- [0056] $K=\{K_i\}$: 시드 키. 1부터 N 까지 정수의 무작위 재배열로 구성되어 있음.
- [0057] K_i : i 번째 시드 키 값
- [0058] SK: Shuffling Key. $n * {}_{n-1}C_{k-1}$ 짜리 행렬로 구성되어 있음
- [0059] $SK(i, j)$: 행렬 SK의 i 번째 행 j 번째 열의 값
- [0060] RK: Recovery Key. $(n-k+1) * N$ 짜리 행렬로 구성되어 있음.
- [0061] $RK(i, j)$: 행렬 RK의 i 번째 행 j 번째 열의 값. 각각의 값은 두 값의 순서쌍으로 되어 있음.
- [0062] (2) 키 생성 단계
- [0063] 다음과 같은 방법으로 K , SK, RK를 생성한다.
- [0064] 1: $k_1, k_2, \dots, k_N$ 에 $1, 2, \dots, N$ 의 재배열을 대입한다;
- [0065] 2: $y_1, y_2, \dots, y_N$ 에 모두 1을 대입한다;
- [0066] 3: i 에 1을 대입한다;
- [0067] 4: (반복문 시작) 모든 $\{1, 2, \dots, N\}$ 의 원소가 $n-k+1$ 개 짜리인 부분집합 $\{c_1, c_2, \dots, c_{n-k+1}\}$ 에 대하여 다음을 반복한다. (단, $c_1 < c_2 < \dots < c_{n-k+1}$)
- [0068] 5: (반복문 시작) $j=1$ 부터 $n-k+1$ 까지 반복
- [0069] 6: $SK(c_j, y_{c_j})$ 에 k_i 를 대입;
- [0070] 7: $RK(j, k_j)$ 에 (c_j, y_{c_j}) 을 대입;
- [0071] 8: y_{c_j} 에 $y_{c_j}+1$ 를 대입;
- [0072] 9: (반복문 끝)
- [0073] 10: (반복문 끝)
- [0074] 11: SK, RK를 모든 분산 저장에 참여하는 파일 분산 저장 장치에 공유;
- [0075] (3) 파일 분산 단계
- [0076]
- [0077] 이 과정의 목표는 파일을 n 개의 파일 분산 저장 장치에 분산 저장하기 위하여 파일의 내용을 쪼개 나누어 n 개의 파일 셰어를 생성하는 과정이다. 이를 위하여 파일 분할 과정을 사용한다. 먼저 파일 분할 과정을 살펴본다.
- [0078] *파일 분할 과정*
- [0079] 이 과정의 목표는 N 개의 연속한 파일 블록을 블록 단위로 나누어서 n 개의 단위 파일 셰어를 생성한다. 이 때, 나뉜 n 개의 단위 파일 셰어 중 임의의 k 개의 단위 파일 셰어는 원래 N 개의 파일 블록을 포함하고 있지만, 임의

의 $k-1$ 개의 단위 파일 셰어는 원래 N 개의 파일 블록 모두를 포함하고 있지는 않다. 분할 과정은 다음과 같다.

[0080] 입력: N 개의 연속한 파일 블록 $b[1], b[2], \dots, b[n]$

[0081] 1: n 개의 단위 파일 셰어 $f_1, f_2, \dots, f_n$ 을 초기화한다;

[0082] 2: (반복문 시작) $i=1$ 부터 n 까지 반복

[0083] 3: (반복문 시작) $j=1$ 부터 M 까지 반복

[0084] 4: f_1 에 블록 $b[SK(i, j)]$ 를 추가한다;

[0085] 5: (반복문 끝)

[0086] 6: (반복문 끝)

[0087] 7: n 개의 단위 파일 셰어 $f_1, f_2, \dots, f_n$ 을 반환;

[0088] 파일 분할 과정을 이용한 파일 분산 단계는 다음과 같다.

[0089] 입력: 원본 파일 F

[0090] 1: n 개의 파일 셰어 $f_1, f_2, \dots, f_n$ 를 초기화;

[0091] 2: L 에 $\left\lceil \frac{\text{size}(F) + 4}{N \times B} \right\rceil$ 의 값을 대입;

[0092] 3: F' 에 원본 파일 F , $N * B * L - \text{size}(F) - 4$ 바이트의 널문자, 그리고 4 바이트의 파일 크기 정보를 연결한 값을 대입;

[0093] 4: (반복문 시작) $i=1$ 부터 N 까지 다음을 반복

[0094] 5: F_{tmp} 에 F' 의 $(i-1) * N+1$ 번째 블록부터 $i * N$ 번째 블록까지 복사;

[0095] 6: F_{tmp} 에 (n,k) 분할 과정을 적용하고, 그 결과물을 $(f_1', f_2', \dots, f_n')$ 에 대입;

[0096] 7: (반복문 시작) $j=1$ 부터 n 까지 다음을 반복

[0097] 8: f_j 에 f_j 와 f_j' 를 연결한 것을 대입;

[0098] 9: (반복문 끝)

[0099] 10: (반복문 끝)

[0100] 11: n 개의 파일 셰어 $f_1, f_2, \dots, f_n$ 을 반환;

[0101] (3) 파일 복구 단계

[0102]

[0103] 이 과정의 목표는 k 개의 파일 셰어를 이용하여 원래 파일로 복구하는 것이다. 복구 과정은 다음과 같다.

[0104] 입력: k 개의 단위 파일 셰어 $f_{S_1'}, f_{S_2'}, \dots, f_{S_k'}$

[0105] 1: F' 를 초기화한다;

- [0106] 2: L 에 $\frac{size(f_{i_2})}{M \times B}$ 의 값을 대입한다;
- [0107] 3: R 을 $\{s_1, s_2, \dots, s_k\}$ 로 지정;
- [0108] 4: (반복문 시작) $j=1$ 부터 N 까지 반복
- [0109] 5: (반복문 시작) $i=1$ 부터 $n-k+1$ 까지 반복
- [0110] 6: $RK(I, j)$ 의 값을 (z, t) 에 대입;
- [0111] 7: (조건문) 만일 $z \in R$ 이라면
- [0112] 8: 반복문 하나를 빠져나간다;
- [0113] 9: (조건문 끝)
- [0114] 10: (반복문 끝)
- [0115] 11: (x_j, y_j) 에 (z, t) 의 값을 대입;
- [0116] 12: (반복문 끝)
- [0117] 13: (반복문 시작) $l=1$ 부터 L 까지 다음을 반복
- [0118] 14: (반복문 시작) $j=1$ 부터 N 까지 반복
- [0119] 15: F' 에 블록 $f_{x_j}[M(1-1)+y_j]$ 를 추가한다;
- [0120] 16: (반복문 끝)
- [0121] 17: (반복문 끝)
- [0122] 18: S 에 F' 의 마지막 4바이트의 값을 대입한다;
- [0123] 19: $SIZE$ 의 값에 S 의 값을 정수로 변환한다;
- [0124] 20: F 에 F' 의 처음 $SIZE$ 바이트를 대입한다;
- [0125] 21: F 를 반환한다;
- [0126] 이하에는, 실제 구현 사례1을 적용한 실제 구현 사례2를 설명한다.

[0127] 2. 구현 사례 2

[0128] (1) 개념 정의 단계

[0129] $n=5, k=3, N={}_5C_{3-1}=10, M={}_{5-1}C_{3-1}=6$

[0130] (2)키 생성 단계

[0131] $K = \{3, 1, 4, 10, 5, 9, 2, 6, 8, 7\}$ 인 경우 도 4c와 같은 결과 값을 얻는다.

[0132] (3)파일 분할 단계

[0133] F 가 도 4a와 같다고 가정하면, $L = \left\lceil \frac{38+4}{2 \times 10} \right\rceil = 3$ 이므로, F' 의 처음 38바이트는 F 의 내용이 그대로 들어가며, 그 다음 18바이트는 널문자(더미영역)가, 그리고 마지막 4바이트(수정파일의 크기 기록 영역)는 파일 크기 정보가

들어가서 총 60바이트의 파일이 된다. F'는 도 4b와 같다. (단, #0은 널문자)

- [0134] 블록의 크기가 2바이트이므로 한 번의 반복문(1슈퍼 블록)에 20바이트씩 파일이 처리되므로 3번의 반복문이 실행된다. 첫 번째 20바이트는 도 4d와 같이 5개의 단위 파일 쉼어로 분할된다.
- [0135] 예를 들어 첫 번째 단위 파일 쉼어는 SK의 첫 번째 행이 3, 1, 4, 1, 5, 9이므로, 파일의 3번째, 1번째, 4번째, 10번째, 5번째, 9번째 블록이 차례로 연결되어 만들어진다. 이와 같은 방법으로 다섯 개의 단위 파일 쉼어가 만들어진다. 이 전체의 과정을 세 번 반복하고, 그 단위 파일 쉼어들을 연결하면 도 4e과 같은 F'에 대한 다섯 개의 파일 쉼어가 만들어진다.
- [0136] (3) 파일 복구 단계
- [0137] 여기에서는 (2)에서 생성한 다섯 개의 파일 쉼어 중 임의의 세 개를 이용하여 원래 파일을 복구할 것이다. 그 세 개의 파일 쉼어를 두 번째, 네 번째, 다섯 번째 파일 쉼어인 f_2 , f_4 , f_5 라고 하자. 우선, f_2 , f_4 , f_5 를 $M=6$ 개의 블록짜리 단위 파일 쉼어로 나눈다. 여기서 첫 번째 단위 파일 쉼어를 각각 f_2' , f_4' , f_5' 라 하자. 그러면 이들은 도 4f와 같이 10개짜리 블록으로 복구된다.
- [0138] 복구 원리는 다음과 같다. 행렬 RK의 첫 번째 열이 $[(1,2), (2,2), (4,1)]^T$ 인데, 그 순서쌍에서 앞의 숫자들로 이루어진 집합 $\{1,2,4\}$ 와 입력 파일 쉼어의 첨자들의 집합인 $\{2,4,5\}$ 의 교집합의 원소를 생각한다. 이 원소들 중 하나를 고르면 4이므로 순서쌍 (4,1)을 선택한다. 이는 네 번째 파일 쉼어의 첫 번째 블록에 원본 파일의 첫 번째 블록이 있다는 뜻이므로 f_4' 의 첫 번째 파일 블록 $[1,2]$ 를 선택한다.
- [0139] 원본 파일의 두 번째 블록도 마찬가지로 다음과 같이 찾아낸다. 행렬 RK의 두 번째 열이 $[(2,4), (3,4), (4,4)]^T$ 인데, 그 순서쌍에서 앞의 숫자들로 이루어진 집합 $\{2,3,4\}$ 와 입력 파일 쉼어의 첨자들의 집합인 $\{2,4,5\}$ 의 교집합의 원소를 생각한다. 이 원소들 중 하나 2를 선택하면, 이에 대응되는 순서쌍은 (2,4)가 된다. 이는 두 번째 파일 쉼어의 네 번째 블록에 원본 파일의 두 번째 블록이 있다는 뜻이므로 f_2' 의 두 번째 파일 블록 $[3,4]$ 를 선택한다.
- [0140] 위의 과정을 반복하면 10개의 블록이 완성된다. 이런 과정을 세 번 거치고, 그 결과들을 차례로 연결하면 도 4b와 같은 결과가 나온다.
- [0141] 이 때, 마지막 네 바이트에 저장된 값이 38이므로 처음 38바이트만 선택하고 나머지는 버린다. 이렇게 해서 최종적으로 도 4a와 같은 파일을 얻어낸다.
- [0142] 이상 설명된 본 발명에 따른 실시예들은 다양한 컴퓨터 구성요소를 통하여 수행될 수 있는 프로그램 명령어의 형태로 구현되어 컴퓨터 판독 가능한 기록 매체에 기록될 수 있다. 상기 컴퓨터 판독 가능한 기록 매체는 프로그램 명령어, 데이터 파일, 데이터 구조 등을 단독으로 또는 조합하여 포함할 수 있다. 상기 컴퓨터 판독 가능한 기록 매체에 기록되는 프로그램 명령어는 본 발명을 위하여 특별히 설계되고 구성된 것들이거나와 컴퓨터 소프트웨어 분야의 당업자에게 공지되어 사용 가능한 것일 수도 있다. 컴퓨터 판독 가능한 기록 매체의 예에는, 하드 디스크, 플로피 디스크 및 자기 테이프와 같은 자기 매체, CD-ROM, DVD와 같은 광기록 매체, 플롭티컬 디스크 (floptical disk)와 같은 자기-광 매체(magneto-optical media), 및 ROM, RAM< 플래시 메모리 등과 같은 프로그램 명령어를 저장하고 수행하도록 특별히 구성된 하드웨어 장치가 포함된다. 프로그램 명령어의 예에는, 컴파일러에 의해 만들어지는 것과 같은 기계어 코드뿐만 아니라 인터프리터 등을 사용해서 컴퓨터에 의해서 실행될 수 있는 고급 언어 코드도 포함된다. 상기 하드웨어 장치는 본 발명에 따른 처리를 수행하기 위해 하나 이상의 소프트웨어 모듈로서 작동하도록 구성될 수 있으며, 그 역도 마찬가지이다.

[0143] 이상에서 본 발명이 구체적인 구성요소 등과 같은 특정 사항들과 한정된 실시예 및 도면에 의해 설명되었으나, 이는 본 발명의 보다 전반적인 이해를 돕기 위해서 제공된 것일 뿐, 본 발명이 상기 실시예들에 한정되는 것은 아니며, 본 발명이 속하는 기술분야에서 통상적인 지식을 가진 자라면 이러한 기재로부터 다양한 수정 및 변형을 꾀할 수 있다.

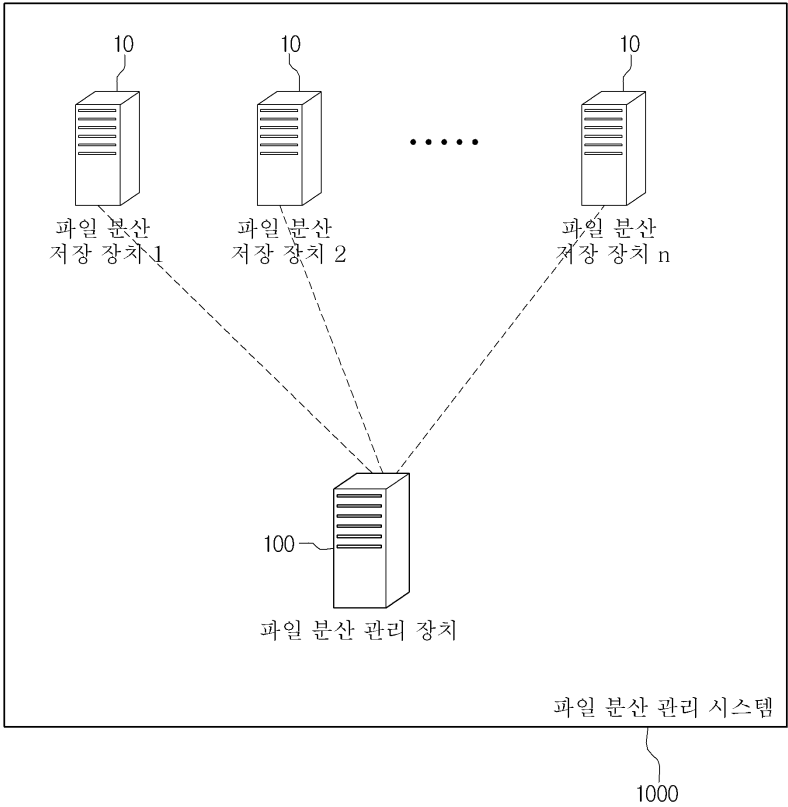
[0144] 따라서, 본 발명의 사상은 상기 설명된 실시예에 국한되어 정해져서는 아니되며, 후술하는 특허청구범위뿐만 아니라 이 특허청구범위와 균등하게 또는 등가적으로 변형된 모든 것들은 본 발명의 사상의 범주에 속한다고 할 것이다.

부호의 설명

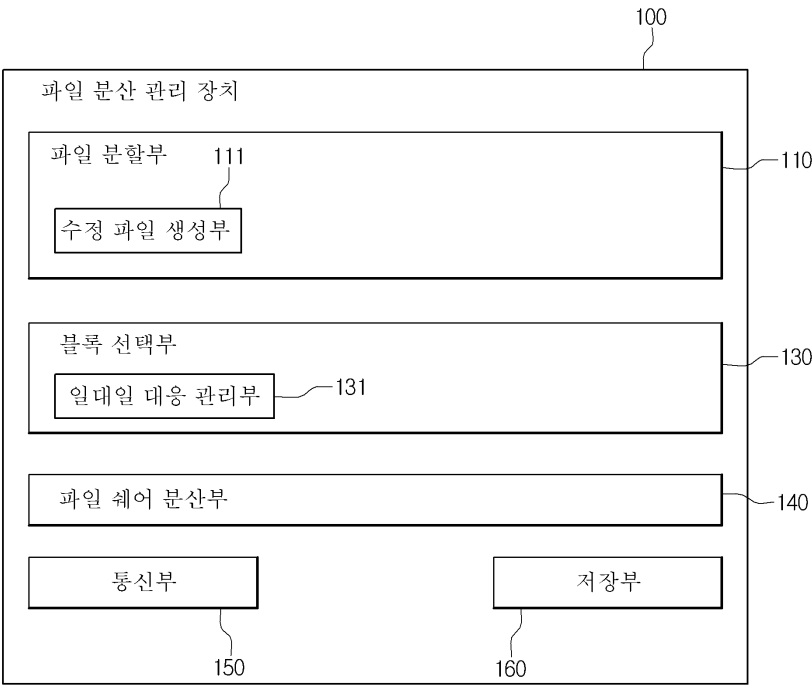
[0145] 10: 파일 분산 저장 장치
 100: 파일 분산 관리 장치
 110: 파일 분할부
 111: 수정 파일 생성부
 120: 블록 재배열부
 130: 블록 선택부
 131: 일대일 대응 관리부
 140: 파일 웨어 분산부
 150: 통신부
 160: 저장부
 1000: 파일 분산 관리 시스템

도면

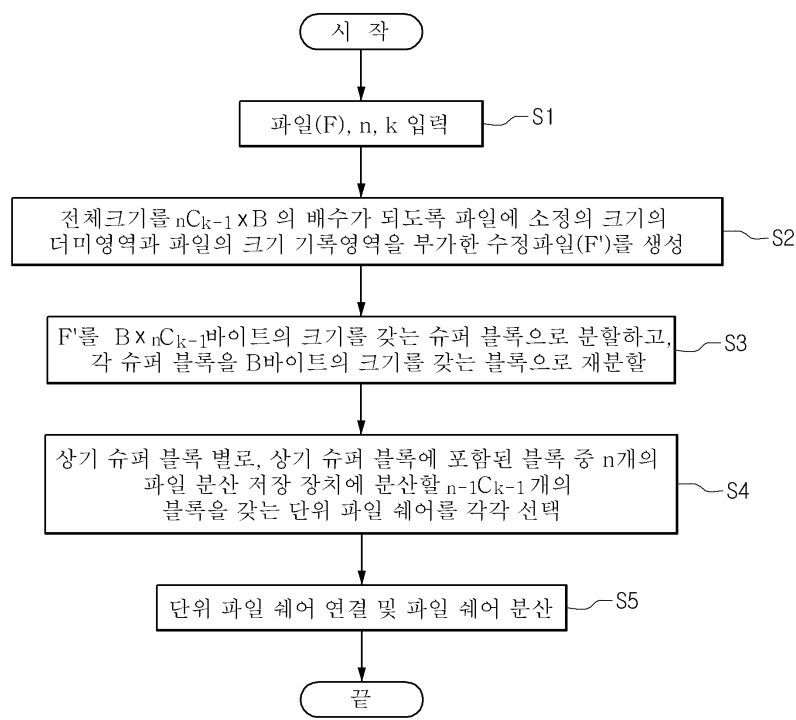
도면1



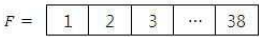
도면2



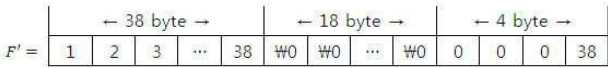
도면3



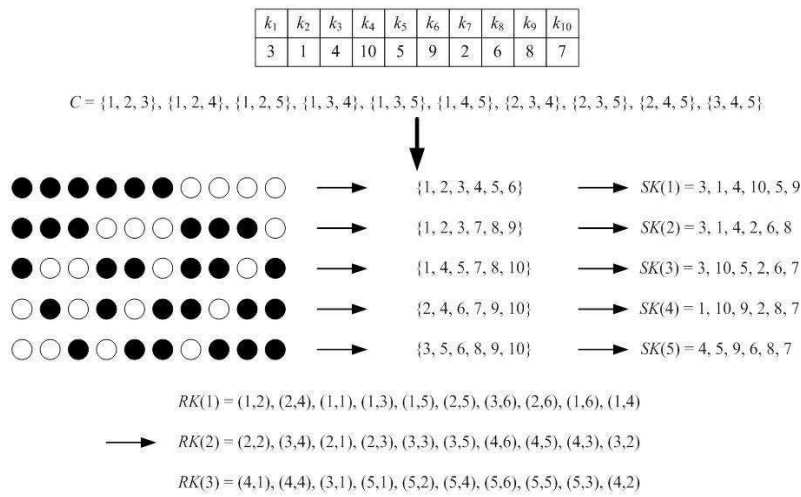
도면4a



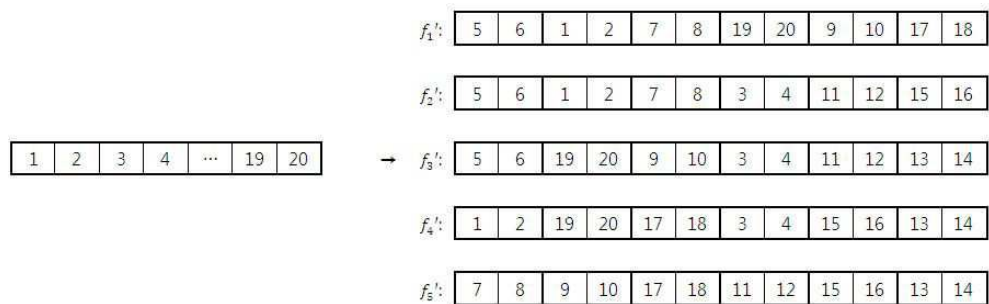
도면4b



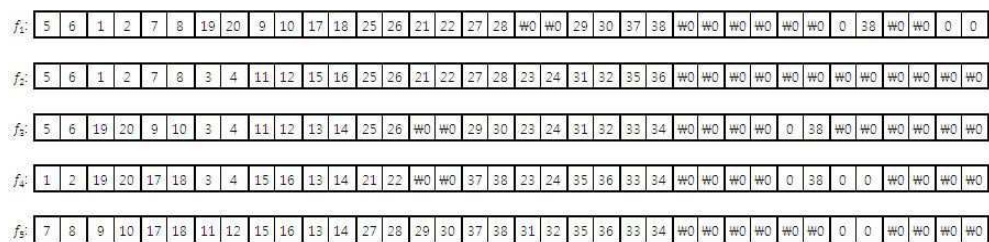
도면4c



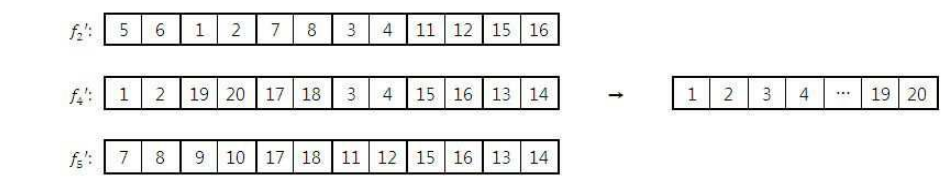
도면4d



도면4e



도면4f



【심사관 직권보정사항】

【직권보정 1】

【보정항목】 청구범위

【보정세부항목】 제4항

【변경전】

상기 상기 슈퍼 블록들

【변경후】

상기 슈퍼 블록들