(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2016년10월19일
(11) 등록번호 10-1667757
(24) 등록일자 2016년10월13일

(51) 국제특허분류(Int. Cl.)
G06F 17/30 (2006.01)
(52) CPC특허분류
G06F 17/30159 (2013.01)
G06F 17/30097 (2013.01)
(21) 출원번호 10-2015-0154589
(22) 출원일자 2015년11월04일
심사청구일자 2015년11월04일
(56) 선행기술조사문헌
정호민 외 3명, 효율적인 에너지 소모를 위한 동기화 시스템 설계, 한국정보과학회 학술발표논문집 39(2A), 102-104 pages, (2012.11.공개)*
정호민 외 1명, 앵커 기반 파일 유사도 예측 기법을 사용한 중복제거 시스템 설계 및 구현, 정보과학회논문지 : 시스템 및 이론 39(5), 298-305 pages, (2012.10.공개)*
*는 심사관에 의하여 인용된 문헌

(73) 특허권자
한림대학교 산학협력단
강원도 춘천시 한림대학길 1, 한림대학교(옥천동)
(72) 발명자
김병관
강원도 춘천시 퇴계로 168, 204동 1403호 (퇴계동,퇴계 (2)주공아파트)
연승호
경기도 용인시 기흥구 흥덕3로 20, 1205동 1001호 (영덕동, 흥덕마을신동아파밀리에아파트)
고영웅
강원도 춘천시 춘주로 174, 107동 1501호 (퇴계동,그린타운아파트)
(74) 대리인
특허법인세신

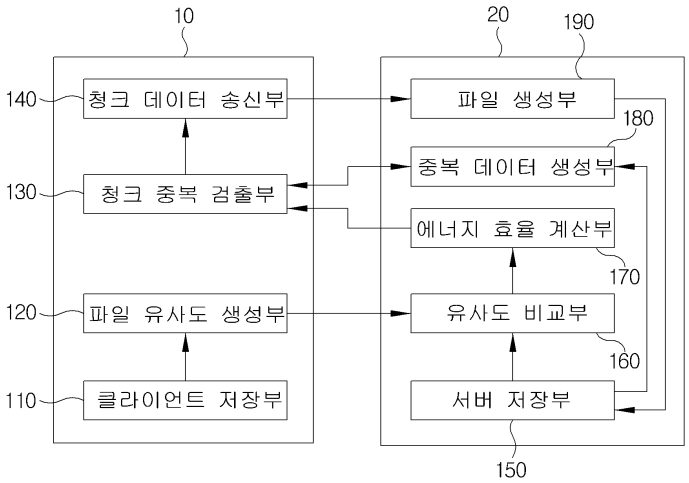
전체 청구항 수 : 총 7 항 심사관 : 고재용

(54) 발명의 명칭 **에너지 효율화를 고려한 파일 동기화 시스템 및 방법**

(57) 요 약

본 발명은 에너지 효율화를 고려한 파일 동기화 시스템 및 방법에 관한 것으로, 본 발명에 따른 에너지 효율화를 고려한 파일 동기화 시스템은, 해시 계산의 오버헤드를 줄이기 위해 앵커에 근거하여 유사도 해시 값들을 생성하는 파일 유사도 생성부를 포함하는 클라이언트 장치와, 클라이언트 장치로부터 이미 수신한 파일들을 저장하고 있는 서버 저장부와, 파일 유사도 생성부로부터 수신한 유사도 해시 값들을 서버 저장부에 저장되어 있는 유사도 해시 값들과 비교하는 유사도 비교부와, 유사도 비교부에서 일정 이상의 유사도를 갖는 파일들이 발견되면 가변 길이 청킹 및 고정 길이 청킹 관련 에너지 효율을 계산하는 에너지 효율 계산부를 포함하는 서버 장치를 포함함으로써, 중복 제거 관련 에너지 소모량을 측정하여 중복 제거 방식을 선택할 수 있다.

대 표 도 - 도1



(52) CPC특허분류
G06F 17/30174 (2013.01)

공지예외적용 : 있음

명 세 서

청구범위

청구항 1

에너지 효율화를 고려한 파일 동기화 시스템에 있어서,

해시 계산의 오버헤드를 줄이기 위해 앵커에 근거하여 유사도 해시 값들을 생성하는 파일 유사도 생성부와, 가변 길이 청킹 또는 고정 길이 청킹을 수행할 수 있는 청크 중복 검출부와, 중복이 없는 청크 데이터를 송신하는 청크 데이터 송신부를 포함하는 클라이언트 장치와,

상기 클라이언트 장치로부터 이미 수신한 파일들을 저장하고 있는 서버 저장부와, 상기 파일 유사도 생성부로부터 수신한 유사도 해시 값들을 상기 서버 저장부에 저장되어 있는 유사도 해시 값들과 비교하는 유사도 비교부와, 상기 유사도 비교부에서 일정 이상의 유사도를 갖는 파일들이 발견되면 가변 길이 청킹 및 고정 길이 청킹 관련 에너지 효율을 계산하는 에너지 효율 계산부와, 상기 청크 중복 검출부로부터 수신한 정보 및 상기 서버 저장부에 저장된 청크 데이터를 이용하여 중복된 청크 데이터를 생성하는 중복 데이터 생성부와, 상기 청크 데이터 송신부로부터 수신한 청크 데이터를 상기 중복 데이터 생성부에서 생성된 청크 데이터와 결합하여 하나의 파일을 생성하는 파일 생성부를 포함하는 서버 장치를 포함하고,

상기 에너지 효율 계산부는, 가변 길이 청킹에 따른 전체 가변 길이 청킹 에너지에, 가변 길이 청킹 관련 유사도를 이용하여 중복되지 않은 청크 파일 전송 에너지를 구한 후 더하여 가변 길이 청킹 관련 총 에너지를 계산하고, 그리고 고정 길이 청킹에 따른 전체 고정 길이 청킹 에너지에, 고정 길이 청킹 관련 유사도를 이용하여 중복되지 않은 청크 파일 전송 에너지를 구한 후 더하여 고정 길이 청킹 관련 총 에너지를 계산하는 것을 특징으로 하는 파일 동기화 시스템.

청구항 2

제1항에 있어서,

상기 파일 유사도 생성부는 앵커가 발견되면 블록 버퍼만큼의 길이를 새로 읽어 다이제스트 함수를 통해 해시 값을 생성하는 것을 특징으로 하는 파일 동기화 시스템.

청구항 3

제2항에 있어서,

상기 파일 유사도 생성부는 해시 값을 정렬하여 소정 개수의 대표 해시 값들을 얻은 후, 상기 유사도 비교부에 대표 해시 값들과 대표 해시 값 관련 오프셋 정보를 송신하는 것을 특징으로 하는 파일 동기화 시스템.

청구항 4

삭제

청구항 5

제1항에 있어서,

상기 에너지 효율 계산부는 가변 길이 청킹 관련 전체 에너지 소모량, 고정 길이 청킹 관련 전체 에너지 소모량, 및 중복 제거 미수행 관련 전체 에너지 소모량 중 가장 적은 에너지를 소모하는 패턴을 선택하여 상기 청크 중복 검출부에 제공하는 것을 특징으로 하는 파일 동기화 시스템.

청구항 6

제5항에 있어서,

상기 청크 중복 검출부는 송신하려는 파일들을 청킹하여 각 청크들의 해시, 오프셋 및 크기를 송신하고,

상기 중복 데이터 생성부는 상기 청크 중복 검출부로부터 수신한 임시 파일의 오프셋 위치에 상기 서버 저장부

에 저장된 중복 청크들을 복사하고,

상기 파일 생성부는 상기 청크 데이터 송신부로부터 수신한 청크들을 임시 파일의 해당 영역에 복사하여 하나의 파일을 생성하는 것을 특징으로 하는 파일 동기화 시스템.

청구항 7

에너지 효율화를 고려한 파일 동기화 방법에 있어서,

서버 장치에서 클라이언트 장치로부터 이미 수신한 파일들을 서버 저장부에 저장하고 있는 단계와,

상기 클라이언트 장치에서 해시 계산의 오버헤드를 줄이기 위해 앵커에 근거하여 유사도 해시 값들을 생성하는 단계와,

상기 서버 장치에서 상기 유사도 해시 값들을 생성하는 단계로부터 제공된 유사도 해시 값들을 상기 서버 저장부에 저장되어 있는 유사도 해시 값들과 비교하는 단계와,

상기 서버 장치에서 상기 비교하는 단계에서 일정 이상의 유사도를 갖는 파일들이 발견되면 가변 길이 청킹 및 고정 길이 청킹 관련 에너지 효율을 계산하는 단계와,

상기 클라이언트 장치에서 상기 에너지 효율을 계산하는 단계로부터 수신한 패턴에 따라 가변 길이 청킹 또는 고정 길이 청킹을 수행하여 청크 중복을 검출하는 단계와,

상기 서버 장치에서 상기 청크 중복을 검출하는 단계로부터 수신한 정보 및 상기 서버 저장부에 저장된 청크 데이터를 이용하여 중복된 청크 데이터를 생성하는 단계와,

상기 클라이언트 장치에서 중복이 없는 청크 데이터를 송신하는 단계와, 및

상기 서버 장치에서 상기 청크 데이터를 송신하는 단계로부터 수신한 청크 데이터를 상기 중복된 청크 데이터를 생성하는 단계로부터 생성된 청크 데이터와 결합하여 하나의 파일을 생성하는 단계를 포함하고,

상기 에너지 효율을 계산하는 단계는, 가변 길이 청킹에 따른 전체 가변 길이 청킹 에너지에, 가변 길이 청킹 관련 유사도를 이용하여 중복되지 않은 청크 파일 전송 에너지를 구한 후 더하여 가변 길이 청킹 관련 총 에너지를 계산하고, 그리고 고정 길이 청킹에 따른 전체 고정 길이 청킹 에너지에, 고정 길이 청킹 관련 유사도를 이용하여 중복되지 않은 청크 파일 전송 에너지를 구한 후 더하여 고정 길이 청킹 관련 총 에너지를 계산하는 것을 특징으로 하는 파일 동기화 방법.

청구항 8

제7항에 있어서,

상기 유사도 해시 값들을 생성하는 단계는 앵커가 발견되면 블록 버퍼만큼의 길이를 새로 읽어 다이제스트 함수를 통해 해시 값을 생성하고, 생성된 해시 값을 정렬하여 소정 개수의 대표 해시 값들을 얻은 후, 상기 서버 장치에 대표 해시 값들과 대표 해시 값 관련 오프셋 정보를 송신하는 것을 특징으로 하는 파일 동기화 방법.

발명의 설명

기 술 분 야

[0001] 본 발명은 에너지 효율화를 고려한 파일 동기화 시스템 및 방법에 관한 것으로, 특히 중복 제거 관련 에너지 소모량을 측정하여 중복 제거 방식을 선택할 수 있는 파일 동기화 시스템 및 방법에 관한 것이다.

배 경 기 술

[0002] 파일을 백업 또는 동기화 등의 이유로 저장하려는 모바일 클라이언트와 drop box, 네이버의 N 드라이브 서버와 같이 이를 저장하려는 서버가 있는 경우, 서버 측의 에너지 효율뿐만 아니라 다수의 클라이언트의 에너지를 효율적으로 관리하는 방법이 요구되는데, 특히 배터리 소모에 민감한 모바일 클라이언트들에는 반드시 필요하다. 조그마한 용량을 송신하고 수신하는 것은 배터리 소모에 영향을 미치지 않지만, 요즘 같이 동영상, mp3, 그림을 포함한 대용량 문서 파일 등을 업로드 또는 다운로드 받을 때에는 WiFi, 3G와 같은 무선 네트워크를 사용함으로써 많은 에너지 소모가 발생하여 배터리 시간을 줄일 수 있다.

[0003] 모바일 클라이언트가 업로드, 다운로드 하려는 많은 수의 파일 중에 서버에 동일한 또는 일부 포함된 데이터가 있을 수 있지만 소수의 모바일 스토리지 클라우드 서비스는 파일 중복만을 감지하거나 그마저도 지원하지 않는 서비스가 대부분이며 파일 내용의 중복이 있는지 감지하는 메커니즘을 적용한 서비스는 아직까지 없다.

[0004] 중복을 감지하여 중복이 아닌 데이터만을 전송하기 위해 중복 제거 메커니즘을 사용해야 하며 중복 제거 메커니즘은 고정 길이 청킹(FLC : Fixed Length Chunking), 가변 길이 청킹(VLC : Variable Length Chunking)이 있다. 두 메커니즘은 각각의 장단점이 있어 한 가지 방법만을 사용하기에는 아쉬운 점이 있다. 고정 길이 청킹의 경우에는 CPU 사용이 적어 에너지를 효율적으로 사용하고 중복 제거 수행시간이 짧은 장점이 있지만 중복을 감지하지 못하는 일이 발생하여 중복 데이터를 전송하는 일이 발생하며 가변 길이 청킹의 경우에는 모든 경우에 대해 중복을 감지할 수 있지만 CPU 사용이 많고 중복 제거 수행시간이 길어지는 단점이 있다.

발명의 내용

해결하려는 과제

[0005] 상술한 문제점을 해결하기 위해, 본 발명은 중복 제거 관련 에너지 소모량을 측정하여 중복 제거 방식을 선택할 수 있는 파일 동기화 시스템 및 방법을 제공하는 것을 목적으로 한다.

[0006] 본 발명은 에너지 효율상의 이유로 두 개의 중복 제거 기법과 단순한 파일 전송을 동시에 사용할 수 있도록 앵커 기반 유사도 추출 기법을 이용하여 파일 간의 중복 유무 또는 어떠한 형태로 되어 있는지 간에 에너지 효율을 고려한 파일 동기화 시스템 및 방법을 제공하는 것을 목적으로 한다.

과제의 해결 수단

[0007] 상술한 목적을 달성하기 위해, 본 발명의 일 실시예에 따른 에너지 효율화를 고려한 파일 동기화 시스템은, 해시 계산의 오버헤드를 줄이기 위해 앵커에 근거하여 유사도 해시 값들을 생성하는 파일 유사도 생성부와, 가변 길이 청킹 또는 고정 길이 청킹을 수행할 수 있는 청크 중복 검출부와, 중복이 없는 청크 데이터를 송신하는 청크 데이터 송신부를 포함하는 클라이언트 장치와, 상기 클라이언트 장치로부터 이미 수신한 파일들을 저장하고 있는 서버 저장부와, 상기 파일 유사도 생성부로부터 수신한 유사도 해시 값들을 상기 서버 저장부에 저장되어 있는 유사도 해시 값들과 비교하는 유사도 비교부와, 상기 유사도 비교부에서 일정 이상의 유사도를 갖는 파일들이 발견되면 가변 길이 청킹 및 고정 길이 청킹 관련 에너지 효율을 계산하는 에너지 효율 계산부와, 상기 청크 중복 검출부로부터 수신한 정보 및 상기 서버 저장부에 저장된 청크 데이터를 이용하여 중복된 청크 데이터를 생성하는 중복 데이터 생성부와, 상기 청크 데이터 송신부로부터 수신한 청크 데이터를 상기 중복 데이터 생성부에서 생성된 청크 데이터와 결합하여 하나의 파일을 생성하는 파일 생성부를 포함하는 서버 장치를 제공할 수 있다.

[0008] 상기 파일 유사도 생성부는 앵커가 발견되면 블록 버퍼만큼의 길이를 새로 읽어 다이제스트 함수를 통해 해시 값을 생성할 수 있다.

[0009] 상기 파일 유사도 생성부는 해시 값을 정렬하여 소정 개수의 대표 해시 값들을 얻은 후, 상기 유사도 비교부에 대표 해시 값들과 대표 해시 값 관련 오프셋 정보를 송신할 수 있다.

[0010] 상기 에너지 효율 계산부는, 가변 길이 청킹에 따른 전체 가변 길이 청킹 에너지에, 가변 길이 청킹 관련 유사도를 이용하여 중복되지 않은 청크 파일 전송 에너지를 구한 후 더하여 가변 길이 청킹 관련 총 에너지를 계산하고, 그리고 고정 길이 청킹에 따른 전체 고정 길이 청킹 에너지에, 고정 길이 청킹 관련 유사도를 이용하여 중복되지 않은 청크 파일 전송 에너지를 구한 후 더하여 고정 길이 청킹 관련 총 에너지를 계산할 수 있다.

[0011] 상기 에너지 효율 계산부는 가변 길이 청킹 관련 전체 에너지 소모량, 고정 길이 청킹 관련 전체 에너지 소모량, 및 중복 제거 미수행 관련 전체 에너지 소모량 중 가장 적은 에너지를 소모하는 패턴을 선택하여 상기 청크 중복 검출부에 제공할 수 있다.

[0012] 상기 청크 중복 검출부는 송신하려는 파일들을 청킹하여 각 청크들의 해시, 오프셋 및 크기를 송신하고, 상기 중복 데이터 생성부는 상기 청크 중복 검출부로부터 수신한 임시 파일의 오프셋 위치에 상기 서버 저장부에 저장된 중복 청크들을 복사하고, 그리고 상기 파일 생성부는 상기 청크 데이터 송신부로부터 수신한 청크들을 임시 파일의 해당 영역에 복사하여 하나의 파일을 생성할 수 있다.

[0013] 본 발명의 다른 실시예에 따른 에너지 효율화를 고려한 파일 동기화 방법은, 서버 장치에서 클라이언트 장치로

부터 이미 수신한 파일들을 서버 저장부에 저장하고 있는 단계와, 상기 클라이언트 장치에서 해시 계산의 오버헤드를 줄이기 위해 앵커에 근거하여 유사도 해시 값들을 생성하는 단계와, 상기 서버 장치에서 상기 유사도 해시 값들을 생성하는 단계로부터 제공된 유사도 해시 값들을 상기 서버 저장부에 저장되어 있는 유사도 해시 값들과 비교하는 단계와, 상기 서버 장치에서 상기 비교하는 단계에서 일정 이상의 유사도를 갖는 파일들이 발견되면 가변 길이 청킹 및 고정 길이 청킹 관련 에너지 효율을 계산하는 단계와, 상기 클라이언트 장치에서 상기 에너지 효율을 계산하는 단계로부터 수신한 패턴에 따라 가변 길이 청킹 또는 고정 길이 청킹을 수행하여 청크 중복을 검출하는 단계와, 상기 서버 장치에서 상기 청크 중복을 검출하는 단계로부터 수신한 정보 및 상기 서버 저장부에 저장된 청크 데이터를 이용하여 중복된 청크 데이터를 생성하는 단계와, 상기 클라이언트 장치에서 중복이 없는 청크 데이터를 송신하는 단계와, 및 상기 서버 장치에서 상기 청크 데이터를 송신하는 단계로부터 수신한 청크 데이터를 상기 중복된 청크 데이터를 생성하는 단계로부터 생성된 청크 데이터와 결합하여 하나의 파일을 생성하는 단계를 제공함으로써, 상술한 목적을 달성할 수 있다.

발명의 효과

[0014] 상술한 구성에 의해, 본 발명은 중복 제거 관련 에너지 소모량을 측정하여 중복 제거 방식을 선택할 수 있다.

[0015] 또한, 본 발명은 앵커 기반 유사도 추출 기법을 이용함으로써 유사도 생성에 따른 에너지 소모를 줄일 수 있다.

[0016] 또한, 본 발명은 대표 해시를 이용함으로써 청크 단위의 비교를 빠르게 처리할 수 있을 뿐만 아니라 고정 길이 청킹의 경우 오프셋 값을 고려하면 중복 제거율을 높일 수 있다.

도면의 간단한 설명

[0017] 도 1은 본 발명의 일실시예에 따른 파일 동기화 시스템의 블록도를 도시하는 도면이다.

도 2는 도 1에 도시된 파일 유사도 생성부의 동작을 설명하는 도면이다.

도 3은 앵커 기반 유사도 추출 알고리즘을 도시하는 도면이다.

도 4는 중복율 별 고정 길이 청킹 및 가변 길이 청킹 에너지 소모량을 설명하기 위한 도면이다.

도 5는 고정 길이 청킹에서의 바이트 시프트 문제를 설명하기 위한 도면이다.

도 6은 본 발명에 실시예에 따른 고정 길이 청킹에 근거한 유사도 청킹 방식을 설명하기 위한 도면이다.

도 7은 도 1에 도시된 에너지 효율 계산부에서의 동기화 방법 선택 알고리즘을 도시하는 도면이다.

도 8은 본 발명의 다른 실시예에 따른 파일 동기화 방법의 흐름도를 도시하는 도면이다.

발명을 실시하기 위한 구체적인 내용

[0018] 이하, 첨부된 도면을 참조하여 본 발명에 따른 에너지 효율화를 고려한 파일 동기화 시스템 및 방법의 바람직한 실시예를 설명한다. 참고로, 아래에서 본 발명을 설명함에 있어서, 본 발명의 구성요소를 지칭하는 용어들은 각각의 구성 요소들의 기능을 고려하여 명명된 것이므로, 본 발명의 기술적 구성요소를 한정하는 의미로 이해되어서는 안 될 것이다.

[0019] 도 1은 본 발명의 일실시예에 따른 파일 동기화 시스템의 블록도를 도시하는 도면이다.

[0020] 도 1에 도시된 바와 같이, 파일 동기화 시스템은 클라이언트 장치(10)와 서버 장치(20)를 포함한다. 클라이언트 장치(10)는 클라이언트 저장부(110), 파일 유사도 생성부(120), 청크 중복 검출부(130) 및 청크 데이터 송신부(140)를 포함하고, 서버 장치(20)는 서버 저장부(150), 유사도 비교부(160), 에너지 효율 계산부(170), 중복 데이터 생성부(180) 및 파일 생성부(190)를 포함한다.

[0021] 클라이언트 저장부(110)에는 클라이언트 장치(10)의 프로그램 등에 의해 생성된 파일이 저장되어 있으며, 파일 유사도 생성부(120)는 해시 계산의 오버헤드를 줄이기 위해 앵커에 근거하여 유사도 해시 값들을 생성하며, 청크 중복 검출부(130)는 가변 길이 청킹 또는 고정 길이 청킹을 수행하고 그 수행 결과에 따른 청크 중복 내용을 서버 장치(20)에 송신하며, 청크 데이터 송신부(140)는 청크 중복 검출부(130)에서 송신하지 아니한 청크 데이터를 송신한다.

[0022] 한편, 파일 유사도 생성부(120)는 앵커가 발견되면 블록 버퍼만큼의 길이를 새로 읽어 다이제스트 함수를 통해 해시 값을 생성하고, 생성된 해시 값을 정렬하여 소정 개수의 대표 해시 값들을 얻은 후, 유사도 비교부(160)에

대표 해시 값들과 대표 해시 값 관련 오프셋 정보를 송신할 수 있다.

[0023] 서버 저장부(150)에는 클라이언트 장치(10)로부터 이미 수신한 파일들이 저장되어 있으며, 유사도 비교부(160)는 파일 유사도 생성부(120)로부터 수신한 유사도 해시 값들을 서버 저장부(150)에 저장되어 있는 유사도 해시 값들과 비교한다.

[0024] 에너지 효율 계산부(170)는 유사도 비교부(160)에서 구한 유사도가 소정 미만이면, 즉 유사도가 높은 파일이 없다면 중복 제거 단계를 수행하지 않고 파일 그대로 송신하게 하는 전체 파일 송신 패턴을 송신한다.

[0025] 하지만, 에너지 효율 계산부(170)는 유사도 비교부(160)에서 일정 이상의 유사도를 갖는 파일이 발견되면 가변 길이 청킹 및 고정 길이 청킹 이용에 따른 전체 에너지 효율을 계산하고, 중복 데이터 생성부(180)는 청크 중복 검출부(130)로부터 수신한 정보를 이용하여 중복된 청크 데이터를 생성하고 중복이 아닌 청크에 대해서는 청크 중복 검출부(130)로 관련 정보를 송신하고, 파일 생성부(190)는 청크 데이터 송신부(140)로부터 수신한 청크 데이터를 중복 데이터 생성부(180)에서 생성된 청크 데이터와 결합하여 하나의 파일을 생성하여 동기화를 마친다.

[0026] 한편, 구현 예로서는 청크 중복 검출부(130)는 송신하려는 파일들을 청킹하여 각 청크들의 해시, 오프셋 및 크기를 송신하고, 중복 데이터 생성부(180)는 청크 중복 검출부(130)로부터 수신한 임시 파일의 오프셋 위치에 서버 저장부(150)에 저장된 중복 청크들을 복사하고, 그리고 파일 생성부(190)는 청크 데이터 송신부(140)로부터 수신한 청크들을 임시 파일의 해당 영역에 복사하여 하나의 파일을 생성할 수 있다. 파일 생성부(190)는 생성된 파일을 서버 저장부(150)에 저장할 수 있다.

[0027] 한편, 에너지 효율 계산부(170)는, 가변 길이 청킹에 따른 전체 가변 길이 청킹 에너지에, 가변 길이 청킹 관련 유사도를 이용하여 중복되지 않은 청크 파일 전송 에너지를 구한 후 더하여 가변 길이 청킹 관련 총 에너지를 계산하고, 그리고 고정 길이 청킹에 따른 전체 고정 길이 청킹 에너지에, 고정 길이 청킹 관련 유사도를 이용하여 중복되지 않은 청크 파일 전송 에너지를 구한 후 더하여 고정 길이 청킹 관련 총 에너지를 계산할 수 있다.

[0028] 에너지 효율 계산부(170)는 가변 길이 청킹 관련 전체 에너지 소모량, 고정 길이 청킹 관련 전체 에너지 소모량, 및 중복 제거 미수행 관련 전체 에너지 소모량 중 가장 적은 에너지를 소모하는 패턴을 선택하여 청크 중복 검출부(130)에 제공할 수 있다.

[0029] 도 2는 도 1에 도시된 파일 유사도 생성부의 동작을 설명하는 도면이고, 도 3은 앵커 기반 유사도 추출 알고리즘을 도시하는 도면이다.

[0030] 유사도 해시 예측 기법으로, 연속되는 블록을 해시하기 위해서 라빈 해시를 사용할 수 있지만, 본 발명의 파일 유사도 생성부(120)는 해시 계산의 오버헤드를 줄이기 위해 앵커 위치에서만 유사도 해시를 구하는 방법을 사용한다.

[0031] 앵커는 일반적으로 특정한 표시를 나타내기 위해서 사용하는 몇 개의 바이트를 의미하며 값이 나올 확률을 조합하여 샘플링 확률을 표현하는 방법이다. 예를 들어 한 개의 바이트가 0일 확률은 1/256이며, 두 개의 연속된 0일 확률은 1/65536이다. 만약 1/16384의 확률로 나타나는 바이트 조합을 앵커로 사용하는 경우는 평균적으로 16384 바이트 크기의 파일 블록마다 앵커가 발견될 수 있다. 일반적으로 가변 길이 청크(variable length chunking)는 앵커를 기반으로 바이트 스트림을 나누고 중복 제거를 처리하는 방법을 사용하는데, 본 발명에서는 유사도 판단에 이런 앵커 방식을 사용한다.

[0032] 앵커가 나올 확률을 작게 만든다면 전체 파일에서 해시 개수가 줄어들고 해시를 계산하는 양도 줄어들어 수행시간을 줄일 수 있다. 반대로 앵커가 나올 확률이 높아지는 경우는 해시 개수가 많아지고 해시 계산을 하는 수행시간도 늘어나게 된다. 본 논문에서는 특정한 앵커가 발견되는 위치에서만 해시 값을 추출하여 대표 해시로 사용하기 때문에 앵커가 나올 확률을 작게 만들수록 해시 수행 시간이 급격히 줄어들 수 있다.

[0033] 그리고 이와 같은 앵커에 근거한 파일 스트림 청킹(File stream chunking)에 관한 도해가 도 2의 상단에 도시되어 있다. 그리고 이들 청킹 데이터는 해시 함수(Hash function), 예를 들어 MD5 또는 SHA1를 이용하여 해시 값을 생성하는데, 도 2에서는 해시 세트(Hash set)에 이들 생성된 5개의 해시 값, 9183, 5813, 1324, 8155, 3324가 저장되고, 대표 해시 알고리즘(Representative hash algorithm)을 통해 대표 해시 값들로 2개의 해시 값, 9183 및 8155를 선정한다. 이와 같이 대표 해시 값을 이용하면, 유사도 비교를 빠르게 수행할 수 있고, 또한 오프셋 값을 이용하면 고정 길이 청킹을 수행하는 경우 중복 제거율을 높일 수 있다.

[0034] 이런 유사도 해시 추출 알고리즘이 도 3에 도시되어 있다. 오프셋(offset)을 0으로 설정하고 입력된 파일의 길이를 length에 초기화한다. while문을 통해 FileStream이 EndofFileStream이 아닐 때까지 알고리즘을 수행하게

한다. 매 루프마다 chnew에 FileStream에 읽어온 캐릭터를 저장하고 chnew와 chold를 128로 나누어 1이 됐을 경우 앵커로 정한다. 앵커가 발견되면 block 버퍼만큼의 길이를 새로 읽어 다이제스트(Digest) 함수를 통해 해시하고 hash 값에 저장한다. hash 값의 크기가 MinHash보다 작은 경우에는 Hasharry에 저장하지 않으며 만약 MinHash보다 크면 상위의 해시이기 때문에 Hasharray 배열에 hash를 저장한다. 저장하는 내용은 offset값과 hash 값을 저장하고 hasharray 배열을 hash 값을 기준으로 quicksort하여 Hasharray[0]에 제일 작은 해시 값이 들어가게 정렬한다. 그리고 Minhash 값은 Hasharray[0]의 값을 다시 넣어준다. 또한, chold를 1로 설정하고 앵커가 아닐 경우에는 chnew값을 chold에 넣어준다.

[0035] 도 4는 중복율 별 고정 길이 청킹 및 가변 길이 청킹 에너지 소모량을 설명하기 위한 도면이며, 도 5는 고정 길이 청킹에서의 바이트 시프트 문제를 설명하기 위한 도면이며, 도 6은 본 발명에 실시예에 따른 고정 길이 청킹에 근거한 유사도 청킹 방식을 설명하기 위한 도면이고, 도 7은 도 1에 도시된 에너지 효율 계산부에서의 동기화 방법 선택 알고리즘을 도시하는 도면이다.

[0036] 도 4는 100MB 파일의 0%, 20%, 40%, 60%, 80% 중복이 있는 파일을 생성한 다음 각 중복율에 따라 고정 길이 청킹, 가변 길이 청킹 방식을 사용했을 때의 에너지 소모량을 나타낸다. 중복율이 변화가 있더라도 고정 길이 청킹, 가변 길이 청킹에서 수행하는 중복 제거의 소모량은 변하지 않는 것을 확인 할 수 있으며 전송되지 않는 중복 데이터 양만큼 파일 전송 에너지가 비례적으로 줄어드는 것을 확인할 수 있는데, 이를 수치를 표현하면 파일 전송 에너지는 중복율이 0%일 때 50000mJ에서 80%일 때 10000mJ로 줄어든다.

[0037] 그리고 도 4에서 확인할 수 있듯이, 고정 길이 청킹의 중복 제거 방법의 에너지 소모량이 가변 길이 청킹보다 적은 장점이 있다. 하지만, 고정 길이 청킹은 경계 이동 문제(byte shift problem)가 있어 파일 유사도보다 찾을 수 있는 중복 데이터의 크기가 작아질 수 있다. 버저닝(Versioning)된 파일에서 새로운 데이터가 앞쪽에 삽입된다면 기존의 데이터들이 뒤로 밀리게 되며 결국 앞 부분의 데이터만을 중복으로 인식하는 문제가 생길 수 있다.

[0038] 도 5는 고정 길이 청킹에서의 이러한 바이트 시프트 문제를 설명하기 위한 도면으로, 소스파일(Old file)에 박스 모양의 새로운 데이터(New data)가 삽입되어 수정된 파일(New file)이 만들어진다. 청크 경계(Chunk Boundary)를 보면 소스파일에서는 굵은 수직선 부분과 블록의 경계가 일치했지만 수정된 파일에서는 굵은 수직선과 블록의 경계가 일치하지 않고 화살표만큼 경계가 이동된 것을 확인할 수 있다. 따라서 새로운 데이터가 삽입된 이후에는 삽입된 데이터 이외의 기존 데이터는 중복 데이터이지만, 중복 데이터로 인식하지 못하게 되고, 따라서 중복 제거율이 낮아질 수 있다.

[0039] 그러므로 고정 길이 청킹에서 실제 중복을 얼마나 찾을 수 있는지를 계산할 수 있어야 고정 길이 청킹, 가변 길이 청킹, 일반적인 파일 전송으로 동기화할지를 결정할 수 있다. 고정 길이 청킹이 얼마만큼의 중복을 찾을 수 있는지를 알아낼 수 있는 것은 유사도 해시와 그 해시가 어디에 위치에 있는지를 가지고 찾을 수 있다.

[0040] 도 6에서 검은 점은 소스파일과 수정된 파일의 동일 대표 유사도 해시를 나타낸 것이다. 그러나 두 개의 대표 해시는 각각의 파일에서 위치하는 부분이 다르므로 서로의 오프셋(offset)을 비교하고 다른 오프셋을 가진 해시의 첫 번째의 위치가 고정 길이 청킹이 중복을 제거할 수 있는 최대량이라는 것을 알 수 있다.

[0041] 따라서 다음의 수식을 이용하여 실질적인 가변 길이 청킹 또는 고정 길이 청킹 관련 총 에너지를 계산할 수 있다.

[0042] 가변 길이 청킹 관련 총 에너지 = 전체 가변 길이 청킹 에너지 + (전체 파일 전송 에너지 * (1 - 유사도))

[0043] 고정 길이 청킹 관련 총 에너지 = 전체 고정 길이 청킹 에너지 + (전체 파일 전송 에너지 * (시프트 위치(MB)/파일 크기(MB)))

[0044] 위의 수식을 이용해 파일을 동기화하기 위한 에너지 효율 계산부(170)에서의 동기화 방법 선택 알고리즘이 도 7에 도시되어 있다. 모든 파일에 대해 수행할 필요 없이 일정 크기 이상의 유사도를 가진 파일만을 대상으로 한다.

[0045] 알고리즘은 클라이언트 장치(10)에서 보낼 파일의 대표 해시 값들의 집합인 ModiHasharr, 원본 파일의 참조 파일의 대표 해시 값들의 집합인 TargetHasharr 그리고 클라이언트 장치(10)에서 보낼 파일의 크기 ModifyFileLength 값들을 입력으로 한다.

[0046] ChangePosition을 ModifyFileLength 값으로 초기화하고 ModiHasharr와 TargetHasharr를 인자로 하여 이중 루프를 수행한다. ModiHasharr 값 중에 TargetHasharr 값의 해시 값과 offset값이 일치하지 않는다면

ChangePosition을 ModiHasharr[i].offset으로 설정하고 루프를 빠져나온다. 고정 길이 청킹 에너지는 A로 치환, 가변 길이 청킹 에너지는 B로 치환, 일반적인 파일 전송 에너지는 C로 치환하여 셋 중에서 제일 작은 값에 대응되는 고정 길이 청킹, 가변 길이 청킹, 일반적인 파일 전송 패턴과 소모되는 에너지 값을 알고리즘의 반환 값으로 한다.

[0047] 도 8은 본 발명의 다른 실시예에 따른 파일 동기화 방법의 흐름도를 도시하는 도면이다.

[0048] 서버 장치(20)는 클라이언트 장치(10)로부터 수신한 파일들을 서버 저장부(150)에 저장한다(S802). 파일 유사도 생성부(120)는 해시 계산의 오버헤드를 줄이기 위해 앵커에 근거하여 유사도 해시 값들을 생성한다(S804). 파일 유사도 생성부(120)는 앵커가 발견되면 블록 버퍼만큼의 길이를 새로 읽어 다이제스트 함수를 통해 해시 값을 생성하고, 생성된 해시 값을 정렬하여 소정 개수의 대표 해시 값들을 얻은 후, 상기 유사도 비교부에 대표 해시 값들과 대표 해시 값 관련 오프셋 정보를 송신할 수 있다.

[0049] 유사도 비교부(160)는 파일 유사도 생성부(120)에서 제공된 유사도 해시 값들을 서버 저장부(150)에 저장되어 있는 유사도 해시 값들과 비교한다(S806). 에너지 효율 계산부(170)는 유사도 비교부(160)에서 일정 이상의 유사도를 갖는 파일들이 발견되면 가변 길이 청킹 및 고정 길이 청킹 관련 에너지 효율을 계산한다(S808). 청크 중복 검출부(130)는 에너지 효율 계산부(170)로부터 수신한 패턴에 따라 가변 길이 청킹 또는 고정 길이 청킹을 수행하여 청크 중복을 검출한다(S810).

[0050] 중복 데이터 생성부(180)는 청크 중복 검출부(130)로부터 수신한 정보 및 서버 저장부(150)에 저장된 청크 데이터를 이용하여 중복된 청크 데이터를 생성한다(S812). 청크 데이터 송신부(140)는 서버 장치(20)에 중복이 없는 청크 데이터를 송신한다(S814). 파일 생성부(190)는 청크 데이터 송신부(140)로부터 수신한 청크 데이터를 중복 데이터 생성부(180)로부터 생성된 청크 데이터와 결합하여 하나의 파일을 생성한다(S816).

[0051] 이상에서 설명된 본 발명의 실시예들은 본 발명의 기술 사상을 예시적으로 보여준 것에 불과하며, 본 발명의 보호 범위는 이하 특허청구범위에 의하여 해석되어야 마땅할 것이다. 또한, 본 발명이 속하는 기술분야에서 통상의 지식을 가진 자라면 본 발명의 본질적인 특성에서 벗어나지 않는 범위에서 다양한 수정 및 변형이 가능할 것인 바, 본 발명과 동등한 범위 내에 있는 모든 기술 사상은 본 발명의 권리범위에 포함되는 것으로 해석되어야 할 것이다.

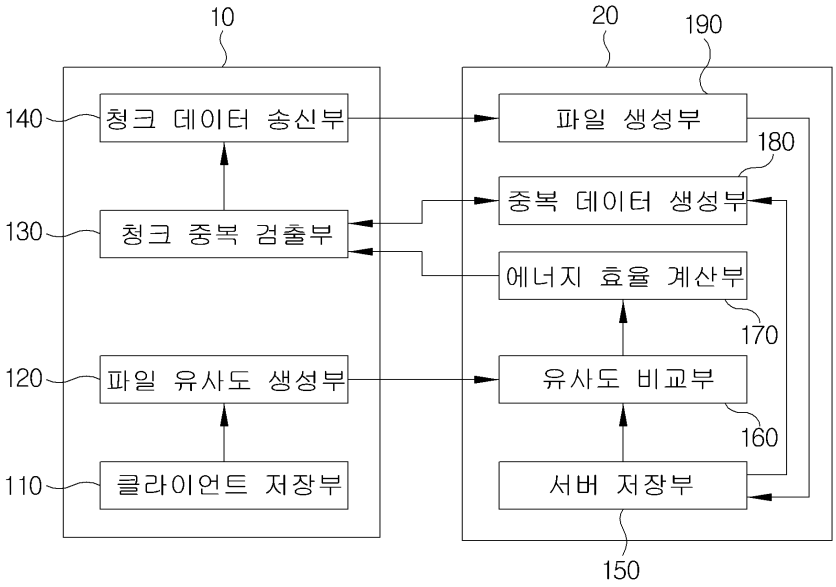
부호의 설명

[0052]

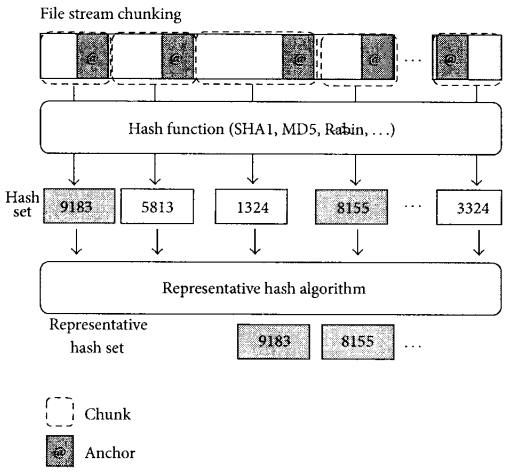
10: 클라이언트 장치	20: 서버 장치
110: 클라이언트 저장부	120: 파일 유사도 생성부
130: 청크 중복 검출부	140: 청크 데이터 송신부
150: 서버 저장부	160: 유사도 비교부
170: 에너지 효율 계산부	180: 중복 데이터 생성부
190: 파일 생성부	

도면

도면1


10
20
190
140
청크 데이터 송신부
파일 생성부
180
중복 데이터 생성부
130
청크 중복 검출부
에너지 효율 계산부
170
120
파일 유사도 생성부
유사도 비교부
160
110
클라이언트 저장부
서버 저장부
150

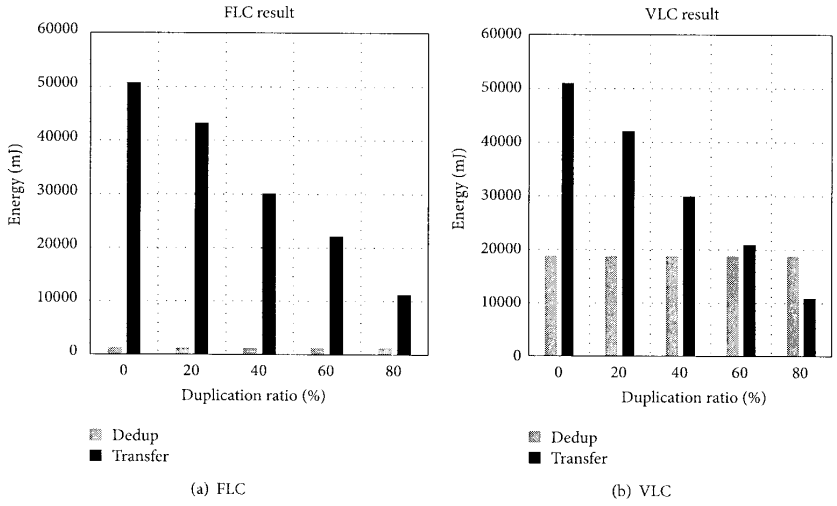
도면2


File stream chunking
Hash function (SHA1, MD5, Rabin, ...)
Hash set
9183
5813
1324
8155
3324
Representative hash algorithm
Representative hash set
9183
8155
Chunk
Anchor

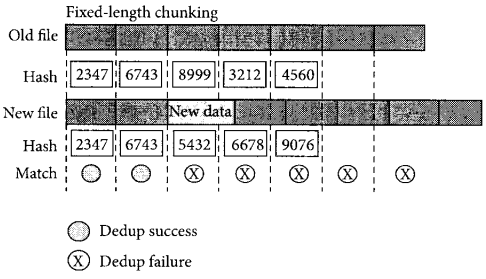
도면3

```
AnchorBaseExtractSimInfo()
Input: FileStream
Output: Hasharray
begin
   offset ← 0
   length ← Length(FileStream)
   chold ← −1
   while Read(chnew, FileStream) != EndofFileStream do
      if (chnew MOD 128 == 1) AND (chold MOD 128 == 1) then
         block ← ReadBlock(blocksize, FileStream)
         hash ← Digest(block)
         if MinHash < hash then
            Hasharray[0].value ← hash
            Hasharray[0].offset ← offset
            quicksort(Hasharray)
            MinHash ← Hasharray[0].value
         chold ← −1
      else
         chold ← chnew
   return Hasharray
end
```

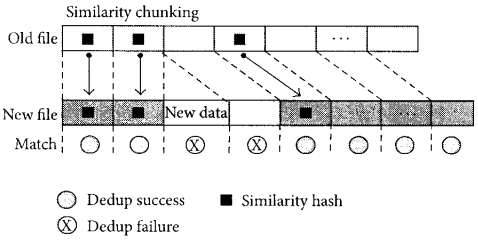
도면4



도면5



도면6



도면7

```
CalcMinEnergy()
Input: ModiHasharr, TargetHasharr, ModifyFileLength
Output: Pattern
begin
  ChangePosition ← ModifyFileLength
  Similarity ← CompareSim(ModiHasharr, TargetHasharr)
  for i ← 0 to ModiHasharr.Length do
    isEqual ← false
    for j ← 0 to TargetHasharr.Length do
      if ModiHasharr[i].hash = TargetHasharr[j].hash then
        is Equal ← true
        shift ← ModiHasharr[i].offset – Target Hasharr[j].offset
        break
    if is Equal ←false OR shift ! = 0 then
      ChangePosition ← OriHasharr[i].offset
      break
  FLCEnergy ← ModifyFileLength * BytePerFLCEnergy
  RemainPortion ← ModifyFileLength – ChangePosition
  A ← FLCEnergy + RemainPortion * BytePerTransferEnergy
  VLCEnergy ← ModifyFileLength * BytePerVLCEnergy
  RemainPortion ← ModifyFileLength * Similarity
  B ← VLCEnergy + RemainPortion * BytePerTransferEnergy
  C ← ModifyFileLength * BytePerTransferEnergy
  Pattern = < B ? (A < C ? FLC: FTP): (B < C ? VLC: FTP)
  MinEnergy = < B ? (A < C ? A : C): (B < C ? B : C)
  Return Pattern, MinEnergy
end
```

도면8


시작
수신한 파일들을 저장한다
S802
유사도 해시 값을 생성한다
S804
유사도 해시 값들을 비교한다
S806
에너지 효율을 계산한다
S808
청크 중복을 검출한다
S810
중복된 청크 데이터를 생성한다
S812
청크 데이터를 송신한다
S814
하나의 파일을 생성한다
S816
종료