



(19) 대한민국특허청(KR)

(12) 등록특허공보(B1)

(45) 공고일자 2015년08월03일

(11) 등록번호 10-1541633

(24) 등록일자 2015년07월28일

(51) 국제특허분류(Int. Cl.)

G06F 13/14 (2006.01) G06F 13/38 (2006.01)
G06F 3/06 (2006.01)

(21) 출원번호 10-2013-7024392

(22) 출원일자(국제) 2012년03월14일

심사청구일자 2013년09월13일

(85) 번역문제출일자 2013년09월13일

(65) 공개번호 10-2013-0132606

(43) 공개일자 2013년12월04일

(86) 국제출원번호 PCT/JP2012/057252

(87) 국제공개번호 WO 2012/157335

국제공개일자 2012년11월22일

(30) 우선권주장

JP-P-2011-110650 2011년05월17일 일본(JP)

(56) 선행기술조사문헌

WO2004077211 A2*

*는 심사관에 의하여 인용된 문헌

(73) 특허권자

가부시끼가이샤 도시바

일본국 도쿄도 미나토구 시바우라 1초메 1방 1고

(72) 발명자

다나카 신고

일본 105-8001 도쿄도 미나토구 시바우라 1초메

1방 1고 가부시끼가이샤 도시바 지적재산부 내

야마우라 다카히로

일본 105-8001 도쿄도 미나토구 시바우라 1초메

1방 1고 가부시끼가이샤 도시바 지적재산부 내

(뒷면에 계속)

(74) 대리인

장수길, 박충범, 이중희

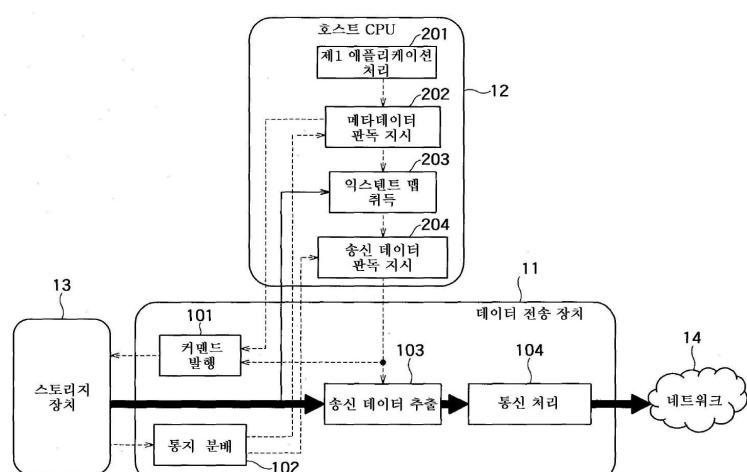
전체 청구항 수 : 총 10 항

심사관 : 김세영

(54) 발명의 명칭 데이터 전송 장치, 데이터 송신 시스템 및 데이터 송신 방법

(57) 요 약

블록 단위로 데이터를 관리하는 스토리지 장치에 기억되어 있는 파일 데이터를 전송하는 데이터 전송 장치로서, 제1 발행부는, CPU로부터의 요구에 응답하여, 상기 스토리지 장치에 대해, 제1 파일 데이터가 어느 위치에 기억되어 있는지를 나타내는 제1 맵 정보를 포함하는 메타데이터의 판독 커맨드를 생성하고, 이를 상기 스토리지 장치에 보내고, 제2 발행부는, 상기 메타데이터를 취득한 상기 CPU로부터, 상기 제1 파일 데이터의 판독 커맨드를 수취하여, 상기 스토리지 장치에 대해, 상기 제1 파일 데이터를 저장한 블록의 판독 커맨드를 생성하고, 이를 상기 스토리지 장치에 보내고, 추출부는, 상기 스토리지 장치로부터 판독되는 블록 데이터를 수취하고, 상기 CPU로부터 상기 제1 맵 정보를 수취하고, 상기 제1 맵 정보에 기초하여, 상기 블록 데이터로부터 상기 제1 파일 데이터를 추출하고, 통신 처리부는, 상기 제1 파일 데이터를 상기 네트워크에 송신한다.

대 표 도

(72) 발명자

야마구치 겐사쿠

일본 105-8001 도쿄도 미나토구 시바우라 1초메 1
방 1고 가부시끼가이샤 도시바 지적재산부 내

와다 스나오

일본 105-8001 도쿄도 미나토구 시바우라 1초메 1
방 1고 가부시끼가이샤 도시바 지적재산부 내

오야 야스오

일본 105-8001 도쿄도 미나토구 시바우라 1초메 1
방 1고 가부시끼가이샤 도시바 지적재산부 내

명세서

청구범위

청구항 1

블록 단위로 데이터를 관리하는 스토리지 장치에 기억되어 있는 파일 데이터를 판독하여 전송하는 데이터 전송 장치로서,

제1 파일 데이터가 상기 스토리지 장치의 어느 위치에 저장되어 있는지를 나타내는 제1 맵 정보에 기초하여 제1 파일 데이터의 판독 지시를 수취하고, 상기 제1 파일 데이터의 상기 판독 지시에 따라서 상기 제1 파일 데이터를 저장하고 있는 블록의 판독 커맨드를 생성하고, 상기 블록의 상기 판독 커맨드를 상기 스토리지 장치에 보내도록 구성되는 발행부 - 상기 제1 맵 정보는, 상기 제1 맵 정보를 포함하는 메타데이터를 수취하고 상기 메타데이터를 해석함으로써 상기 제1 맵 정보를 취득하는 외부의 호스트 CPU에 의해 취득됨 -

상기 블록의 상기 판독 커맨드에 따라서 상기 스토리지 장치로부터 판독되는 블록 데이터를 수취하고, 상기 외부의 호스트 CPU에 의해 취득된 상기 제1 맵 정보를 수취하고, 상기 제1 맵 정보에 기초하여, 상기 블록 데이터로부터 상기 제1 파일 데이터를 추출하도록 구성되는 송신 데이터 추출부와,

상기 송신 데이터 추출부에 의해 추출된 상기 제1 파일 데이터를 송신하도록 구성되는 통신 처리부를 포함하는, 데이터 전송 장치.

청구항 2

제1항에 있어서,

상기 외부의 호스트 CPU로부터의 지시에 응답하여, 상기 제1 맵 정보를 포함하는 메타데이터의 판독 커맨드를 생성하고, 상기 메타데이터의 상기 판독 커맨드를 상기 스토리지 장치에 보내도록 구성되는 제1 발행부를 더 포함하고,

상기 외부의 호스트 CPU는 상기 메타데이터의 상기 판독 커맨드에 따라서 상기 스토리지 장치로부터 보내진 상기 메타데이터를 수취하는, 데이터 전송 장치.

청구항 3

제1항 또는 제2항에 있어서,

통신 카드로서 구성되는, 데이터 전송 장치.

청구항 4

블록 단위로 데이터를 관리하는 스토리지 장치에 기억되어 있는 파일 데이터를 판독하여 송신하고, CPU와 데이터 전송 장치를 포함하는 데이터 송신 시스템으로서,

상기 CPU는,

상기 스토리지 장치에 저장된 제1 파일 데이터의 송신 지시를 생성하도록 구성되는 제1 애플리케이션 처리부와,

상기 송신 지시에 기초하여, 상기 제1 파일 데이터가 상기 스토리지 장치 상의 어느 위치에 기억되어 있는지를 나타내는 제1 맵 정보를 포함하는 메타데이터의 판독 지시를 생성하도록 구성되는 메타데이터 판독 지시부

를 포함하고,

상기 데이터 전송 장치는, 상기 메타데이터의 상기 판독 지시에 기초하여, 상기 메타데이터의 판독 커맨드를 생성하고, 상기 메타데이터의 상기 판독 커맨드를 상기 스토리지 장치에 보내도록 구성되는 제1 발행부를 포함하고,

상기 CPU는 상기 스토리지 장치로부터 복귀된 상기 메타데이터를 수취하고, 상기 메타데이터를 해석함으로써 상기 제1 맵 정보를 취득하고, 상기 제1 맵 정보에 기초하여, 상기 제1 파일 데이터의 판독 지시를 생성하도록 구

성되는 송신 데이터 관독 지시부를 포함하고,

상기 데이터 전송 장치는,

상기 제1 파일 데이터의 상기 관독 지시에 기초하여, 상기 제1 파일 데이터를 저장하고 있는 블록의 관독 커맨드를 생성하고, 상기 관독 커맨드를 상기 스토리지 장치에 보내도록 구성되는 제2 발행부와,

상기 관독 커맨드에 따라서 상기 스토리지 장치로부터 관독되는 블록 데이터를 수취하고, 상기 제1 맵 정보에 기초하여, 상기 블록 데이터로부터 상기 제1 파일 데이터를 추출하도록 구성되는 송신 데이터 추출부와,

상기 송신 데이터 추출부에 의해 추출된 상기 제1 파일 데이터를 송신하도록 구성되는 통신 처리부를 포함하는, 데이터 송신 시스템.

청구항 5

제4항에 있어서,

상기 CPU는,

제2 애플리케이션 처리부와,

애플리케이션 데이터 관독 지시부와,

애플리케이션 추출부

를 포함하고,

상기 제2 애플리케이션 처리부는, 제2 파일 데이터의 취득 지시를 생성하고,

상기 제1 발행부는, 상기 취득 지시에 기초하여, 상기 제2 파일 데이터가 상기 스토리지 장치 상의 어느 위치에 기억되어 있는지를 나타내는 제2 맵 정보를 포함하는 메타데이터의 관독 커맨드를 생성하여, 상기 관독 커맨드를 상기 스토리지 장치에 보내고,

상기 애플리케이션 관독 지시부는, 상기 스토리지 장치로부터 복귀된 상기 메타데이터를 수취하고, 상기 메타데이터를 해석함으로써 상기 제2 맵 정보를 취득하고, 상기 제2 맵 정보에 기초하여, 상기 제2 파일 데이터의 관독 지시를 생성하고,

상기 제2 발행부는, 상기 제2 파일 데이터의 상기 관독 지시에 기초하여, 상기 제2 파일 데이터를 저장하고 있는 블록의 관독 커맨드를 생성하여, 상기 관독 커맨드를 상기 스토리지 장치에 보내고,

상기 애플리케이션 데이터 추출부는, 상기 스토리지 장치로부터 관독되는 블록 데이터를 수취하고, 상기 제2 맵 정보에 기초하여, 상기 블록 데이터로부터 상기 제2 파일 데이터를 추출하고, 추출된 상기 데이터를 상기 제2 애플리케이션 처리부에 보내는, 데이터 송신 시스템.

청구항 6

제4항에 있어서,

상기 데이터 전송 장치는,

플로우 제어부와,

데이터 버퍼부와,

버퍼 관리부를 포함하고,

상기 제1 애플리케이션 처리부는, 상기 제1 파일 데이터를 송신하는 커넥션을 지정하고,

상기 데이터 버퍼부는, 커넥션마다의 버퍼를 갖고, 상기 송신 데이터 추출부에 의해 추출된 데이터를, 상기 제1 애플리케이션 처리부에 의해 지정된 커넥션에 대응하는 버퍼에 버퍼링하고,

상기 통신 처리부는, 상기 데이터 버퍼에 있어서의 상기 버퍼로부터 상기 데이터를 관독하여 상기 데이터를 송신하고,

상기 버퍼 관리부는, 상기 지정된 커넥션에 대응하는 상기 버퍼의 가용 상태를 관리하고,

상기 플로우 제어부는, 상기 송신 데이터 판독 지시부로부터 상기 송신 데이터 판독 지시를 수취하고, 상기 제1 파일 데이터 중에서 아직 판독되지 않은, 상기 가용 상태에 대응하는 사이즈의 데이터에 대한 판독 지시를 생성하고 상기 판독 지시를 상기 제2 발행부에 송신하는, 데이터 송신 시스템.

청구항 7

제6항에 있어서,

상기 CPU는,

제3 애플리케이션 처리부와,

애플리케이션 데이터 기입 요구부와,

애플리케이션 데이터 기입부를 포함하고,

상기 제3 애플리케이션 처리부는, 상기 스토리지 장치와는 다른 임의의 장치 상에 존재하는 애플리케이션 데이터의 송신 지시를 생성하고, 상기 애플리케이션 데이터를 송신하는 커넥션을 지정하고,

상기 베퍼 관리부는, 상기 제3 애플리케이션 처리부에 의해 지정된 커넥션에 대응하는 베퍼의 가용 상태를 관리하고,

상기 플로우 제어부는, 상기 가용 상태에 대응하는 사이즈의 데이터에 대한 기입 허가 신호를 생성하고,

상기 애플리케이션 데이터 기입부는, 상기 기입 허가 신호에 기초하여, 상기 애플리케이션 데이터 중에서 상기 베퍼에 아직 기입되지 않은 상기 가용 사이즈의 데이터를 특정하여 상기 데이터를 상기 베퍼에 기입하고,

상기 통신 처리부는, 상기 커넥션에 대응하는 상기 베퍼로부터 데이터를 판독하여 상기 데이터를 송신하는, 데이터 송신 시스템.

청구항 8

제7항에 있어서,

상기 제3 애플리케이션 처리부는, 상기 제1 애플리케이션 처리부에서와 같은 커넥션을 지정하는, 데이터 송신 시스템.

청구항 9

블록 단위로 데이터를 관리하는 스토리지 장치에 기억되어 있는 파일 데이터를 판독하여 송신하는 데이터 송신 방법으로서,

CPU에 의해, 상기 스토리지 장치에 저장된 제1 파일 데이터의 송신 지시를 생성하는 단계와,

상기 CPU에 의해, 상기 송신 지시에 기초하여, 상기 제1 파일 데이터가 상기 스토리지 장치 상의 어느 위치에 기억되어 있는지를 나타내는 제1 맵 정보를 포함하는 메타데이터의 판독 지시를 생성하는 단계와,

데이터 전송 장치에 의해, 상기 메타데이터의 상기 판독 지시에 기초하여, 상기 메타데이터의 판독 커맨드를 생성하고, 상기 판독 커맨드를 상기 스토리지 장치에 송신하는 단계와,

상기 CPU에 의해, 상기 스토리지 장치로부터 복귀된, 상기 제1 맵 정보를 포함하는 메타데이터를 수취하고, 상기 메타데이터를 해석함으로써 상기 제1 맵 정보를 취득하고, 상기 제1 맵 정보에 기초하여, 상기 제1 파일 데이터의 판독 지시를 생성하는 단계와,

상기 데이터 전송 장치에 의해, 상기 제1 파일 데이터를 저장하고 있는 블록의 판독 커맨드를 생성하고, 상기 제1 파일 데이터의 상기 판독 커맨드에 기초하여, 상기 블록의 상기 판독 커맨드를 상기 스토리지 장치에 송신하는 단계와,

상기 데이터 전송 장치에 의해, 상기 블록의 상기 판독 커맨드에 따라서 상기 스토리지 장치로부터 판독되는 블록 데이터를 수취하고, 상기 제1 맵 정보에 기초하여, 상기 블록 데이터로부터 상기 제1 파일 데이터를 추출하는 단계와,

상기 데이터 전송 장치에 의해, 상기 제1 파일 데이터를 송신하는 단계

를 포함하는, 데이터 송신 방법.

청구항 10

블록 단위로 데이터를 관리하는 스토리지 장치에 기억되어 있는 파일 데이터를 전송하기 위해, 컴퓨터에, 제1 파일 데이터가 상기 스토리지 장치 상의 어느 위치에 저장되어 있는지를 나타내는 제1 맵 정보에 기초하여 제1 파일 데이터의 판독 지시를 수취하고, 상기 제1 파일 데이터의 판독 지시에 따라서 상기 제1 파일 데이터를 저장하고 있는 블록의 판독 커맨드를 생성하고, 상기 블록의 상기 판독 커맨드를 상기 스토리지 장치에 보내는 단계 - 상기 제1 맵 정보는, 상기 제1 맵 정보를 포함하는 메타데이터를 수취하고 상기 메타데이터를 해석함으로써 상기 제1 맵 정보를 취득하는 외부의 호스트 CPU에 의해 취득됨 - 와,

상기 블록의 상기 판독 커맨드에 따라 상기 스토리지 장치로부터 판독된 블록 데이터를 수취하고, 상기 외부의 호스트 CPU에 의해 취득된 상기 제1 맵 정보를 수취하고, 상기 제1 맵 정보에 기초하여, 상기 블록 데이터로부터 상기 제1 파일 데이터를 추출하는 단계와,

추출된 상기 제1 파일 데이터를 송신하는 단계

를 실행시키는 명령어들이 저장되어 있는 비일시적인 컴퓨터 판독가능한 기록 매체.

발명의 설명

기술 분야

[0001] 본 발명의 실시형태는 프로세서 대신에 통신 프로토콜 처리를 행하는 데이터 전송 장치, 데이터 송신 시스템 및 데이터 송신 방법에 관한 것이다.

배경 기술

[0002] 컴퓨터에 직접 접속된 HDD(Hard Disk Drive) 및 SSD(Solid State Drive)는 "DAS(Direct Attached Storage)"라고 부르고 있으며, 이를 스토리지를 네트워크를 경유해서 접속하는 네트워크 스토리지가 점점 더 보급되고 있다. 네트워크 스토리지는, 크게 나누어 SAN(Storage Area Network)과 NAS(Network Attached Storage)로 분류할 수 있다. 알려진 SAN은 iSCSI 및 파이버 채널(fiber channel) 등을 포함하며, 스토리지 장치는 블록 장치로서 네트워크를 경유해서 (스토리지를 이용하는) 호스트 머신(host machine)에 접속되고, 호스트 머신 측에 파일 시스템이 구비되어 있다. 한편, 알려진 NAS는 NFS(Network File System) 등을 포함하며, 스토리지 장치 측이 파일 시스템을 내장하고 있고 호스트 장치는 파일 시스템의 상위로부터 스토리지에 액세스할 수 있다. 이에 따라, 호스트 머신의 부하를 경감할 수 있을 뿐만 아니라, 복수대의 호스트 머신이 스토리지 상의 파일 시스템을 공유할 수 있는 이점이 있다.

[0003] 이를 네트워크 스토리지는, 주로 이더넷(Ethernet)에 의해 통신하며 또한 통신 프로토콜로서 인터넷 등에서 널리 사용되고 있는 TCP/IP나 UDP/IP를 사용한다. 이를 네트워크 스토리지 이용이 용이하게 되고, 네트워크 스토리지는 종래의 DAS에 비해 스토리지로서의 확장성과 보수성 측면에서 유용성이 향상되었다.

[0004] 또한, 최근에는 영상 스트리밍 분배가 점점 보급되고 있다. 예를 들면, "VOD(Video On Demand) 서버"라고 불리는 콘텐츠 분배 서버가 스트리밍 분배를 행한다. 이러한 콘텐츠 분배 서버는, 그 자신의 스토리지 장치로부터 영상 콘텐츠를 판독하고, 그 영상 콘텐츠를 네트워크 접속된 사용자 단말기에 송신하는 처리를 행한다. Web 서버도 마찬가지로 스토리지 상의 콘텐츠를 사용자 단말기에 송신한다. 통신 프로토콜로서는, 예를 들어 TCP/IP 기반의 HTTP나 리얼타임 통신을 위한 UDP/IP 기반의 RTP가 사용된다. 또한, RTP의 경우에는, 에러 정정에 FEC(Forward Error Correction) 등이 사용된다.

[0005] 이러한 네트워크 스토리지, VOD 서버 또는 Web 서버에 이용되는 스토리지와 네트워크 등의 디바이스의 속도는 현저하게 증가하였다. 이더넷은 현재 1 Gbps가 점점 주류가 되고 있고, 데이터 센터 등에서는 10 Gbps가 사용되기 시작하고 있다. 또한, 차세대의 40/100 Gbps 이더넷 시스템의 사양 책정도 완료되어, 이들도 장래에 확실히 보급되어 질 것으로 기대된다. 또한, 스토리지에 관해서는 HDD RAID 스트라이핑(striping) 처리(즉, 병렬화)의 고속화에 더하여, 최근의 SSD 전송 성능의 향상은 현저하며, 따라서 현행 제품 단독으로 2 Gbps 초과의 판독 레이트를 가질 수 있다. 스토리지 장치의 I/F 규격의 SATA에 관해서도, 벤드폭 6 Gbps의 SATA 3.0이

이미 보급되고 있다. 이상을 감안하면, 네트워크 스토리지에 관해서도, 가까운 장래에 10 Gbps 클래스의 대역이 제공될 것으로 예상된다.

[0006] 이러한 네트워크 스토리지의 성능 향상에 따라, 스토리지를 제어하는 호스트 CPU에 대한 처리 부하도 증대된다. 종래에는, "TCP/IP 오프로드(offload) 엔진"(이하, "TOE")이라고 일컫는 기술에 의해, 그 해결이 시도되었다. TOE는, 호스트 CPU 대신에, 전술한 TCP/IP 처리를 행하는 전용 프로세서나 전용 회로를 구비하여, 호스트 CPU에 대한 TCP/IP 처리 부하를 오프로드한다. 이 TOE를 이용함으로써, 종래의 소프트웨어 통신 프로토콜 처리보다 고속으로 TCP/IP 처리를 행할 수 있게 되어, 네트워크 스토리지의 성능 향상에 공헌할 수 있다.

[0007] 스토리지 장치나 TOE는 호스트 CPU에 의해 제어되고, 데이터는 메인 메모리를 통해 입출력하는 것으로 상정된다. 따라서, 스토리지 상의 데이터를 네트워크에 입출력할 때, 스토리지 장치(SSD, HDD 등)와 TOE 사이의 데이터 전송은, 반드시 메인 메모리를 통해 행해진다.

[0008] 또한, 호스트 CPU 상에서 동작하여 이를 사이를 브릿지하는 애플리케이션 소프트웨어에 있어서는, OS의 커널(kernel) 공간과 사용자 공간 사이의 데이터 교환 처리 등에 의해, 메인 메모리 상에서 수 회의 전송 데이터의 카피가 발생할 경우도 있다. 또한, 파일 시스템 처리도 필요하다. 따라서, 통상의 파일 시스템 실장에서는, 스토리지로부터 고정 바이트 길이의 섹터 단위로 기록 또는 판독된 데이터를, 임의의 바이트 길이의 파일로 변환하고, 따라서 데이터의 카피가 더 발생한다.

발명의 내용

해결하려는 과제

[0009] 따라서, 스토리지 장치와 TOE 사이에서 데이터 전송을 행할 경우는, 호스트 CPU에서의 소프트웨어 처리가 개재되고, 따라서 적어도 일 회는 메인 메모리에서의 기입 또는 판독이 요구되고, 또는 OS, 애플리케이션 또는 파일 시스템 처리에 의해 메인 메모리 상에서 경우에 따라 복수 회의 메모리 카피가 발생하게 되어, 메인 메모리에 대한 부하가 대폭 증대되어버린다.

[0010] 이러한 메모리 부하에 대응하기 위하여, 종래는 충분한 처리 성능을 갖는 호스트 CPU와 고속의 메인 메모리가 구비되었다. 최근의 네트워크 스토리지의 전송 레이트가 거의 10 Gbps로 향상되는 것 때문에, 이는 다소 문제가 된다. 특히, 메인 메모리의 성능을 높이기 위해서는, 일반적으로 호스트 CPU의 업그레이드가 요구되며, 또한 PC나 서버 등의 경우에는, 부수되는 칩셋의 업그레이드도 요구된다. 이에 따라, 비용이나 소비 전력의 문제 가 특히 현저하게 된다.

과제의 해결 수단

[0011] 본 발명의 일 실시형태로서의 데이터 전송 장치는, 블록 단위로 데이터를 관리하는 스토리지 장치에 기억되어 있는 파일 데이터를 네트워크에 전송하는 데이터 전송 장치로서, 제1 발행부와, 제2 발행부와, 송신 데이터 추출부와, 통신 처리부를 포함한다.

[0012] 제1 발행부는, 외부의 호스트 CPU로부터의 요구에 응답하여, 제1 파일 데이터가 상기 스토리지 장치 상의 어느 위치에 기억되어 있는지를 나타내는 제1 맵 정보를 포함하는 메타데이터의 판독 커맨드를 생성하고, 상기 판독 커맨드를 상기 스토리지 장치에 보낸다.

[0013] 제2 발행부는, 상기 스토리지 장치로부터 복귀된 상기 메타데이터를 취득한 상기 외부의 호스트 CPU로부터 상기 제1 파일 데이터의 판독 커맨드를 수취하고 상기 제1 파일 데이터를 저장한 블록의 판독 커맨드를 생성하여, 이 판독 커맨드를 상기 스토리지 장치에 보낸다.

[0014] 송신 데이터 추출부는, 상기 판독 커맨드에 따라서 상기 스토리지 장치로부터 판독되는 블록 데이터를 수취하고, 상기 외부의 호스트 CPU로부터 상기 제1 맵 정보를 수취하고, 상기 제1 맵 정보에 기초하여, 상기 블록 데이터로부터 상기 제1 파일 데이터를 추출한다.

[0015] 상기 통신 처리부는, 상기 송신 데이터 추출부에 의해 추출된 상기 제1 파일 데이터를 상기 네트워크에 송신한다.

도면의 간단한 설명

[0016] 도 1은 제1 실시형태에 따른 데이터 송신 시스템의 개략 구성을 나타내는 블록도이다.

도 2는 스토리지 장치에 저장된 데이터의 일례를 나타내는 도면이다.

도 3은 커맨드들과 송신 데이터 추출부의 어드레스 어리어들 사이의 연관을 나타내는 도면이다.

도 4는 제1 실시형태에 따른 데이터 송신 시스템의 다른 예를 나타내는 도면이다.

도 5는 제2 실시형태에 따른 데이터 송신 시스템의 개략 구성을 나타내는 블록도이다.

도 6은 스토리지 장치에 저장된 데이터의 다른 예를 나타내는 도면이다.

도 7은 제2 실시형태에 따른 데이터 송신 시스템의 다른 구성예를 나타내는 도면이다.

발명을 실시하기 위한 구체적인 내용

[0017] 이하, 도면을 참조하여 본 발명의 실시형태를 설명한다.

[0018] (제1 실시형태)

도 1은 제1 실시형태에 따른 데이터 송신 시스템의 개략 구성을 나타내는 블록도이다. 점선 화살표는 제어 흐름을 나타내고, 실선 화살표는 스토리지로부터 판독된 데이터의 흐름을 나타내고, 굵은 선 화살표는 판독된 데이터 내의 송신 데이터의 흐름을 나타낸다.

도 1의 데이터 송신 시스템은 데이터 전송 장치(11)와, 호스트 CPU(프로세서)(12)와, 스토리지 장치(13)를 포함하고 있다. 호스트 CPU(12)의 제어에 기초하여, 데이터 전송 장치(11)는 데이터를 스토리지 장치(13)로부터 판독하고, 네트워크(14)에 송신한다.

도 1의 데이터 송신 시스템의 구체적인 제품 형태로는, PC, 서버 장치, 전용 LSI, FPGA(Field Programmable Gate Array) 등의 네트워크 스토리지, VOD(Video On Demand) 서버 또는 Web 서버가 상정되지만, 이들에 한정되는 것은 아니다.

[0022] 호스트 CPU(12)는 시스템 전체의 제어를 행한다. 호스트 CPU(12)는, 소프트웨어에 의해 실행되는 메인 처리부로서, 데이터 송신 애플리케이션 처리를 행하는 제1 애플리케이션 처리부(201)와, 스토리지 장치(13)의 메타데이터의 판독을 지시하는 메타데이터 판독 지시부(202)와, 판독된 메타데이터를 해석해서 익스텐트 맵(extent map) 정보를 취득하는 익스텐트 맵 취득부(203)와, 데이터 전송 장치(11)에 데이터의 판독 및 데이터의 송신을 지시하는 송신 데이터 판독 지시부(204)를 갖는다. 물론 호스트 CPU(12)는, 그 밖의 여러 가지 처리를 행하는 것이 가능하다.

[0023] 스토리지 장치(13)는 송신될 데이터를 저장하고 있다. 스토리지 장치(13)의 구체적인 실장 형태는, 예를 들면 SSD(Solid State Drive), HDD(Hard Disk Drive) 또는 SD 메모리 카드 등이 상정되지만, 이들로 제한되는 것은 아니다.

[0024] 데이터 전송 장치(11)는, 호스트 CPU(12) 대신에, 스토리지 장치(13)로부터 데이터를 판독하고, 통신 처리를 행하고, 판독된 데이터를 네트워크(14)에 송신한다. 데이터 전송 장치(11)는, 스토리지 장치(13)에 대하여 커맨드를 발행하는 커맨드 발행부(101)(제1 발행부 및 제 2 발행부를 포함함), 발행된 커맨드의 응답 통지를 수취하여, 이를 호스트 CPU의 해당 구성요소(도 1의 구성예에서, 메타데이터 판독 지시부(202) 및 송신 데이터 판독 지시부(204))에 분배하는 통지 분배부(102), 스토리지 장치(13)로부터 판독된 데이터를 수취해서 필요한 데이터를 추출하는 송신 데이터 추출부(103) 및 추출된 데이터를 네트워크(14)에 송신하는 통신 처리부(104)를 갖는다. 통신 처리부(104)는 전술한 TOE에 상당한다. 데이터 전송 장치(11)의 구체적인 실장 형태로는, FPGA 또는 전용 LSI가 상정되고, PCI 카드 또는 PCI(Peripheral Component Interconnect)-Express 카드로서 실장될 수도 있지만, 이들로 한정되는 것은 아니다.

[0025] 스토리지 장치(13)는 RAID(Redundant Arrays of Inexpensive Disks) 기능을 갖출 수도 있고, 혹은, 데이터 전송 장치(11) 내의 스토리지 장치(13)와의 I/F가 RAID 기능을 갖출 수도 있다. 이를 경우는 본 발명의 본질과 관계가 없으므로 그 설명을 생략한다.

[0026] 네트워크(14)는 이더넷, 무선 LAN 및 임의의 다른 무선 통신 및 유선 통신을 포함한다.

[0027] 호스트 CPU(12)와 데이터 전송 장치(11)는, 본 장치가 PC이거나 서버 장치인 경우에는, 예를 들면 칩셋이나 PCI Express 등에 의해 접속될 수 있으며, SoC 등의 경우에는, 예를 들면 전용의 버스 등에 의해 접속될 수 있다. 스토리지 장치(13)와 데이터 전송 장치(11)는, 예를 들면 SATA, SAS, SCSI 또는 SD 메모리 카드 규격 등에 의해

접속된다. 이들은 단지 예일 뿐이며, 이들로 한정되는 것은 아니다.

[0028] 데이터 전송 장치(11)는, 하드웨어 또는 소프트웨어에 의해 구성될 수 있으며, 그 실장 형태는 문제되지 않지만, 처리 효율의 관점으로부터 하드웨어 기반으로 구성되는 것이 바람직하다. 또한, 통신 처리부(104)에서 수행되는 통신 처리는 어떤 것이라도 좋고, 예를 들면 TCP/IP 기반의 프로토콜 처리, UDP/IP 기반의 프로토콜 처리, 이더넷 처리 및 무선 처리 등을 포함한다. 또한, TCP와 같은 수신처 장치와 접속되는 프로토콜의 경우, 이하의 동작 설명은, 접속 상대와 접속이 확립된 후의 데이터 전송의 단계에서의 동작을 상정하고 있음을 알아야 한다.

[0029] 스토리지 장치(13)는 데이터를 예를 들면 섹터 단위로 관리한다. 여기서는 추상적으로 "블록"이라고 부른다. 사용자가 취급하는 데이터는 블록 단위 데이터가 아니므로, 스토리지 장치(13)는 보통은 파일 시스템에 의해 액세스된다. 파일 시스템은, 바이트 단위의 파일 데이터를 블록 단위의 스토리지 장치(13)의 기록 에리어(스토리지 에리어) 내로 맵핑하는 시스템이다. 파일 시스템은, 공지의 파일 시스템을 이용하므로, 그 상세한 설명은 생략하지만, 간단하게 설명하면, 스토리지 상에 있는 어떤 영역에, 맵핑 정보를 포함하는 메타데이터(다른 여러 가지 정보를 포함함)를 데이터와 함께 기억한다. 스토리지로부터 파일 데이터를 판독할 때는, 우선, 이 메타데이터를 판독하고 해석하여, 원하는 파일 데이터가 실제로 어느 블록에 기억되어 있는지를 나타내는 정보(제1 맵 정보)(즉, 익스텐트 맵)를 취득한다. 그후, 그 익스텐트 맵 정보에 따라 블록 데이터를 판독하고, 필요 부분만을 추출해서 원하는 데이터를 취득한다. 익스텐트 맵은 구체적으로는 도 2와 같이 제공된다.

[0030] 도 2는, 1 블록이 512 바이트이고, 블럭 235 내지 238에 원하는 파일 데이터의 소정 부분이 저장되어 있는 상태를 보여주고 있다. 데이터는 블록 235의 124번째 바이트부터 시작하고 있으므로, 오프셋 124 바이트 및 길이 1458 바이트에 의해 표현된다.

[0031] 따라서, 익스텐트 맵은, 데이터가 저장되어 있는 블록 위치의 정보를 나타내고, 구체적으로는, 그 데이터의 유효 데이터가 저장되어 있는 유효 데이터 위치의 정보를 나타낸다. 즉, 익스텐트 맵 정보는 스토리지의 어디에 원하는 데이터가 저장되어 있는지를 특정하기 위한 정보를 나타낸다.

[0032] 이 예에서는, 데이터가 저장되어 있는 바이트 위치를 오프셋 및 길이에 의해 나타내고 있지만, 이들은 예를 들어 시작 바이트 위치와 종료 바이트 위치 등에 의해 나타내도 되며, 유효 데이터의 위치를 특정할 수 있는 한, 표현 형식은 문제되지 않는다.

[0033] 또한, 이 예에서는 1458 바이트의 하나의 데이터 시퀀스가 예로서 제공되었지만, 익스텐트 맵 정보는 이산적인 복수의 데이터 시퀀스로 나타내고 있어도 좋다. 즉, 하나의 익스텐트 맵 정보가, 2개의 데이터 시퀀스, 즉, 블록 235 내지 238 가운데서 1458 바이트를 나타내는 정보와, 블록 346 내지 250 가운데서 2000 바이트를 나타내는 정보를 포함할 수도 있다.

[0034] 또한, 이상의 익스텐트 기반의 파일 시스템을 적용할 수 있지만, 본 발명은 블록(또는 클러스터) 기반의 파일 시스템도 포함한다. 블록(또는 클러스터) 기반의 파일 시스템의 경우에는, 익스텐트 대신에 블록(클러스터) 리스트가 제공되지만, 이것도 익스텐트가 특수할 경우(하나의 익스텐트의 길이를 512 바이트로 고정한 경우)로 간주하면 마찬가지로 생각할 수 있다. 이후에는 "익스텐트"라는 용어를 이용하여 설명하며, 이를 양쪽을 포함하는 것으로 한다.

[0035] 공지 기술에서는, 통상, 호스트 CPU(12)는 애플리케이션에 의한 파일 데이터 판독 요구에 기초하여, 디바이스 드라이버 및 파일 시스템 소프트웨어 등의 제어 소프트웨어를 실행함으로써, 이 익스텐트 맵을 취득하고, 이 익스텐트 맵에 따라 원하는 데이터 스토리지로부터 판독을 행하고, 애플리케이션에 판독된 데이터를 돌려준다. 한편, 본 실시형태에 따르면, 이 처리의 일부를 데이터 전송 장치(11)가 대신하고, 스토리지로부터 판독된 데이터를 호스트 CPU(12)에 복귀시키지 않고서 네트워크(14)에 송신한다. 구체적인 처리의 흐름은 다음과 같다.

[0036] 우선, 제1 애플리케이션 처리부(201)는, 애플리케이션 소프트웨어의 처리에 의해, 송신 파일(제1 파일) 및 필요에 따라서 그 파일 내의 데이터 위치를 특정한다. 전형적인 예로서는, 네트워크(14)에 접속된 다른 장치로부터의 요구에 기초하여 파일이 특정된다. 그후, 메타데이터 판독 지시부(202)에, 해당 파일의 송신 지시와 함께, 파일 정보(파일 식별자 및 필요에 따라 파일 내의 데이터 위치)가 보내진다.

[0037] 메타데이터 판독 지시부(202)는, 제공된 정보에 기초하여, 스토리지 장치(13)로부터 파일 시스템의 메타데이터를 판독하기 위해, 데이터 전송 장치(11)의 커맨드 발행부(101)에 커맨드 발행을 지시한다. 커맨드 발행 지시는, 파일 식별자 및 필요에 따라 데이터 위치를 포함한다. 커맨드 발행부(101)는, 메타데이터 판독 커맨드(즉, 제1 커맨드)를 발행하고, 이를 스토리지 장치(13)에 보낸다. 커맨드 발행부(101)가 구비하는 기능 중에서, 제1

커맨드를 발행하는 기능은 제1 발행부에 상당한다.

[0038] 스토리지 장치(13)는 이 커맨드에 기초하여, 대응하는 메타데이터를 판독하고, 이를 익스텐트 맵 취득부(203)에 제공한다.

[0039] 익스텐트 맵 취득부(203)는, 이 메타데이터 정보를 해석하고, 판독하려는 파일 데이터의 익스텐트 맵을 취득한다. 또한, 익스텐트 맵 정보의 취득은, 미리 캐쉬되어 있는 것을 참조하여 행할 수도 있지만, 마찬가지 수순으로 미리 판독을 행할 필요가 있음을 알아야 한다.

[0040] 그후, 취득한 익스텐트 맵 정보를, 송신 데이터 판독 지시부(204)가 데이터 전송 장치(11)에 제공하고, 데이터 전송 장치(11)에 대해 그 데이터의 판독과 송신의 지시를 행한다.

[0041] 데이터 전송 장치(11)에서는, 커맨드 발행부(101)가 그 익스텐트 맵 정보를 수취하고, 그 정보에 기초하여, 송신하려는 데이터를 저장한 블록을 판독하기 위해, 블록 판독 커맨드(즉, 제2 커맨드)를 발행한다. 커맨드 발행부(101)가 구비하는 기능 중, 제2 커맨드를 발행하는 기능은 제2 발행부에 상당한다. 커맨드는 스토리지 장치(13)측의 I/F 시스템에 의존하지만, 통상, 한 번에 판독될 수 있는 블록수가 한정되거나, 이산적인 블록은 동시에 지정할 수 없거나 하는 경우가 있고, 결과적으로는, 하나의 익스텐트 맵 정보가 복수의 커맨드 군으로 분할될 경우도 있어서, 그 분할 처리도 행한다.

[0042] 그후, 커맨드 발행부(101)가 그 커맨드를 순차적으로 스토리지 장치(13)에 대해 발행하면, 스토리지 장치(13)는 응답으로서 데이터가 포함된 블록(들)을 신속히 돌려주고, 송신 데이터 추출부(103)는 그 블록 데이터를 수취한다.

[0043] 한편, 송신 데이터 판독 지시부(204)는, 커맨드 발행부(101)에 지시를 줄 때, 송신 데이터 추출부(103)에, 개개의 커맨드에 대응하는 블록의 어느 부분에 송신하려는 데이터가 포함되어 있는지를 나타내는 정보를 제공한다. 즉, 전술한 도 2의 예에서, 송신 데이터 판독 지시부(204)는 블록 235 내지 238의 블록 위치 정보를 커맨드 발행부(101)에 제공하고, 오프셋 124 바이트 및 길이 1458 바이트를 포함하는 유효 데이터 위치 정보를 송신 데이터 추출부(103)에 제공한다.

[0044] 송신 데이터 추출부(103)는, 수취한 블록 데이터로부터 이 정보를 기초로 원하는 1458 바이트의 데이터 시퀀스를 추출하고, 이를 통신 처리부(104)에 제공한다.

[0045] 그후, 통신 처리부(104)는 필요에 따라 수신처 장치와 제어 정보를 통신하면서, 네트워크(14) 상에 그 데이터를 송신한다.

[0046] 스토리지 장치(13)와의 I/F 처리에 대해 보다 상세하게 설명한다. 우선, 스토리지 장치(13)가 전형적인 SSD나 HDD 등의 SATA I/F를 구비하고, 또한 버스 마스터(bus master)로서의 기능(즉, DMA 기능)을 구비하는 경우, 스토리지 장치(13)에의 커맨드와 관련하여, "디스크립터(descriptor)"라고 불리는 커맨드를 기술한 데이터를, 소정의 메모리 상에 준비해 두고, 스토리지 장치(13)가 이를 판독하여 커맨드가 스토리지 장치(13)에 제공된다. 전술한 커맨드 발행부(101)는 이러한 처리를 행한다.

[0047] 디스크립터는 보통 링 버퍼 형태이며, 전형적인 공지 기술에서는 호스트 CPU(12)가 사용하는 메인 메모리(주기 억 장치, 도시되지 않음) 상에 놓인다. 본 실시형태에서도 위와 같은 형태를 적용할 수 있지만, 데이터 전송 장치(11) 내의 메모리, 혹은 데이터 전송 장치(11)에 직접적으로 접속되는 메모리를 이용하는 것이, 메인 메모리가 호스트 CPU(12)에 부하를 부여하지 않는 점에서 바람직하다.

[0048] 또한, 스토리지 장치(13)가 판독한 블록 데이터가 DMA에 의해 기입되는 수신처 어드레스도, 각 디스크립터에 있어서 미리 지정되지만, 판독된 메타데이터를 호스트 CPU(12)에 제공하고, 송신 데이터를 통신 처리부(104)에 제공하는 분배 처리는, 이 어드레스 지정에 의해 실현된다. 즉, 메타데이터를 호스트 CPU(12)에 제공할 경우는, 통상은 메인 메모리를 통해서 행하므로, 디스크립터에 기재하는 수신처 어드레스로서 메인 메모리 어드레스를 기재한다. 송신 데이터를 통신 처리부(104)에 제공할 경우는, 수신처 어드레스로서 통신 처리부(104)의 어드레스를 기재한다. 이에 따라, 스토리지 장치(13)에 의해 디스크립터에 지정된 수신처 어드레스에 판독된 데이터를 기입하는 것만으로, 판독된 데이터를 개별적으로 분배할 수 있다.

[0049] 또한, 스토리지 장치(13)는 커맨드를 실행하여 데이터를 판독한 후에 커맨드 완료 통지(즉, 응답 통지)를 행하지만, 이 통지는 인터럽트 등에 의해 행해지고, 어디까지 커맨드를 처리했는지, 즉, 링 버퍼 디스크립터 군의 어느 디스크립터까지 처리했는지를, 예를 들어 각 디스크립터의 처리 완료를 나타내는 "DONE" 비트를 설정하는 등에 의해 통지한다. 즉, 호스트 CPU(12)는 인터럽트 통지를 받으면, 그 "DONE" 비트가 기입되어 있는 디스크

팁터를 순차적으로 판독(그후, 판독한 비트를 리셋함)하고, 따라서 그때까지의 처리가 완료하고 있는 것을 판정할 수 있다. 여기서, 한번에 복수의 디스크립터의 처리가 완료하고 있을 경우도 있을 수 있다.

[0050] 통지 분배부(102)는 이상과 같이 완료하였다고 판정된 디스크립터가 메타데이터 판독이었던 경우는 메타데이터 판독 지시부(202)에 통지하고, 통지 분배부(102)는 상기 디스크립터가 송신 데이터 판독이었을 경우는 송신 데이터 판독 지시부(204)에 통지하여, 완료 통지를 분배한다.

[0051] 메타데이터 판독 지시부(202)는 이 통지를 받으면, 메타데이터의 판독이 완료했다고 판정하고, 익스텐트 맵 취득부(203)에 익스텐트 맵의 취득을 지시한다.

[0052] 또한, 스토리지 장치(13)에 의한 블록 데이터의 전송은, 통상은 일정한 지역을 수반하므로, 호스트 CPU(12)가 스토리지 장치(13)로부터의 실행 완료를 매회 기다리지 않고, 복수의 블록의 판독 커맨드를 스토리지 장치(13)에 대하여 통합해서 발행할 경우가 있다. 이 경우, 송신 데이터 판독 지시부(204)(및 메타데이터 판독 지시부(202))와 스토리지 장치(13)와 송신 데이터 추출부(103) 사이에서 동기가 취해지지 않게 되므로, 예를 들면 이하의 방법을 채택하여 동기를 확립한다.

[0053] 첫 번째 방법은, 스토리지 장치(13)가 커맨드를 받은 순서와 반드시 동일 순서로 판독을 실행할 경우에 유효한 방법으로, 송신 데이터 판독 지시부(204)가, 스토리지 장치(13)에 복수의 커맨드의 판독 지시를 제공할 때, 그 지시를 행하는 것과 같은 순서로, 대응하는 블록 내의 유효 데이터의 위치 정보를 송신 데이터 추출부(103)에 통지하는 방법이다. 송신 데이터 추출부(103)는, 수취한 유효 데이터 위치 정보를 예를 들면 큐잉하여 두고, 스토리지 장치(13)로부터 블록 데이터를 수취했을 때, 큐잉된 유효 데이터 위치 정보를 큐잉의 순서로 하나씩 추출하여, 블록 데이터와의 대응을 실현할 수 있다.

[0054] 두 번째 방법은, 스토리지 장치(13)가, 내부 처리의 효율 등을 우선할 경우 등에, 커맨드를 받은 순서와는 역순으로 판독을 실행할 수 있는 방법이다. 이 경우에, 예를 들면 각 블록의 수신처 어드레스로서 상이한 어드레스를 지정하는 것이 생각된다. 즉, 예를 들면 스토리지 장치(13)에 4개의 판독 커맨드를 통합해서 지시할 경우, 그 4개의 커맨드에 대해 상이한 전송처 어드레스를 지정해 둔다. 본 실시형태에서는, 전송처 어드레스로서 송신 데이터 추출부(103)의 어드레스를 지정하지만, 이 시스템을 이용할 경우는, 예를 들면 1개의 커맨드로 최대 4개 블록의 판독 지시를 제공할 경우, 개별 커맨드에 대응하는 전송처 어드레스에, 송신 데이터 추출부(103)의 어드레스 영역의 상이한 어드레스 영역을 지정해 둔다.

[0055] 예를 들면, 송신 데이터 추출부(103)의 어드레스 영역이 0×20200000 내지 $0 \times 202fffff$ 인 경우, 도 3과 같이, 0×20200000 내지 $0 \times 202007ff$ 의 512×4 바이트 영역이 커맨드 0에 대응하고, 0×20200800 내지 $0 \times 20200fff$ 의 512×4 바이트 영역이 커맨드 1에 대응하고, 0×20201000 내지 $0 \times 202017ff$ 의 512×4 바이트 영역이 커맨드 2에 대응하고 등으로 대응이 확립되고, 송신 데이터 판독 지시부(204)는 각 커맨드에 이들 어드레스 영역의 헤드를 지정하도록 커맨드 발행부(101)에 지시한다. 이어서, 전송된 블록 데이터는 지속적으로 송신 데이터 추출부(103)에 전송되지만, 각 데이터는 송신 데이터 추출부(103) 내의 상이한 어드레스 영역에 전송되고, 데이터를 받은 송신 데이터 추출부(103)는, 받은 블록 데이터가 예를 들면 0×20201200 으로부터 시작되는 어드레스에 기입된 데이터이면, 커맨드 2의 헤드로부터 2번째의 블록 데이터로 특정할 수 있다. 이 커맨드 2의 유효 데이터 위치 정보가 도 2와 같은 경우, 이 블록 데이터는 유효 데이터의 389번째 바이트로부터 900번째 바이트의 데이터인 것을 안다.

[0056] 한편, 송신 데이터 판독 지시부(204)는, 송신 데이터 추출부(103)에 대해 유효 데이터 위치 정보의 8개의 아이템을 통지하고, 이때, 커맨드 0에 대응하는 유효 데이터 위치 정보, 커맨드 1에 대응하는 유효 데이터 위치 정보, 커맨드 2에 대응하는 유효 데이터 위치 정보 등으로 알 수 있도록 송신 데이터 추출부(103)에 통지한다. 이에 따라, 송신 데이터 추출부(103)는 수취한 데이터 및 유효 데이터 위치 정보가 어느 커맨드에 속한 것인가를 판정할 수 있으므로, 송신 데이터 추출부(103)는 이들을 대응시킬 수 있다. 어드레스를 사용하는 것은 본 발명의 일례일 뿐이며, 스토리지 장치(13)로부터 송신 데이터 추출부(103)에 통지할 수 있는 정보라면 그 밖의 다른 정보를 이용할 수도 있다.

[0057] 이와 같이하여, 블록 데이터 수신처 어드레스 영역을 나눔에 의해, 송신 데이터 추출부(103)에서 수취된 블록 데이터의 순서가 원래의 데이터 시퀀스 순서(통상적으로는 네트워크(14)에 송신하는 순서)와 다른 경우에도, 블록 데이터의 수신처 어드레스를 확인함으로써 그 데이터가 어느 커맨드에 기초한 것인지를 판정할 수 있고, 또한 이를 기초로 하여 데이터 시퀀스를 원래의 순서대로 재배열할 수 있다.

[0058] 이상의 스토리지 I/F의 상세는 일례일 뿐이며, 본 발명은 이에 한정되는 것은 아니다.

[0059] 또한, 스토리지 장치(13)가 내부 에러 등 어떠한 이유에 의해 요구된 커맨드를 완료할 수 없을 경우가 있다. 이 경우, 스토리지 장치(13)는 데이터 전송 장치(11)에 에러 통지를 행하지만, 이러한 상태에서는, 호스트 CPU(12)는 이 정보를 취득할 수 없으므로, 데이터 전송 장치(11)가 호스트 CPU(12)에 에러 통지를 중계해서 일상적으로 호스트 CPU(12)가 이상을 알 수 있도록 한다.

[0060] 이상, 본 발명의 제1 실시형태를 설명하였다. 이상과 같은 실시형태를 이용함으로써, 애플리케이션으로부터의 요구에 기초하여, 호스트 CPU(12)와 메인 메모리(도시되지 않음)에 데이터 전송 처리의 부하를 부여하지 않고서 스토리지 장치(13) 상의 데이터를 네트워크(14)에 송신할 수 있으므로, 처리 효율 및 데이터 전송 속도를 향상 시킬 수 있다. 또한, 호스트 CPU(12), 또는 이에 부수되는 칩셋, 메모리, 마더보드 등의 그레이드를 낮출 수 있으므로, 장치의 저비용화를 도모할 수 있다. 또한, 데이터 전송의 대부분의 처리를, 호스트 CPU(12)의 소프트웨어 처리 대신에, 데이터 전송 장치(11)의 하드웨어 처리에 의해 실행할 수 있게 되므로, 장치의 저소비 전력화를 도모할 수 있다.

[0061] 또한, 위에 있어서, 통지 분배부(102)는 반드시 데이터 전송 장치(11) 상에 형성할 필요는 없으며, 호스트 CPU(12) 상에 형성해도 된다. 또한, 커맨드 발행부(101)도 그 일부 처리를 호스트 CPU(12)측에서 행해도 되지만, 처리 효율의 측면에 있어서, 하나의 송신 데이터 판독 지시를 복수의 커맨드로 분할하는 처리는 데이터 전송 장치(11) 상에서 행하는 것이 바람직하다. 또한 반대로, 메타데이터 판독 지시부, 익스텐트 맵 취득부 및 송신 데이터 판독 지시부를 데이터 전송 장치(11) 상에 형성해도 된다.

[0062] 또한, 제1 애플리케이션 처리부(201)에 의해 특정된 파일 정보를 후속하는 처리부에 전달하는 처리는, Linux OS 등에 실장되어 있는 send file 함수 등과 같은 표준 시스템 콜에 의해 인터페이스되고, 이 함수 내의 처리를 전술의 동작을 실현하도록 제공하는 실시형태가 바람직하다. 이러한 표준 함수에 의해 애플리케이션과 인터페이스를 행함으로써, 기존의 애플리케이션에 변경을 가할 필요가 없으며, 따라서 본 발명의 실시를 용이하게 할 수 있다.

[0063] 이상의 구성에 의해, 스토리지 장치(13)로부터 네트워크(14)에의 호스트 CPU(12)를 수반하지 않는 데이터 전송이 가능하게 되지만, 이 형태가 바람직하지 않을 경우가 있다. 예를 들면, 다른 애플리케이션 처리 등에 의해, 데이터 전송없이, 호스트 CPU(12) 자신이 스토리지 장치(13)의 파일(즉, 제2 파일) 데이터를 참조할 것이 요구되는 경우가 있다. 이 경우의 구성은 도 4와 같다. 도 4의 제2 애플리케이션부(301)는, 데이터 송신없이 단순하게 스토리지 상의 데이터를 판독할 것이 요구되는 애플리케이션 처리부이다. 또한, 도 4는, 도 1에 더하여, 해당 데이터의 판독을 데이터 전송 장치(11)에 지시하는 애플리케이션 데이터 판독 지시부(303)와, 판독된 당해 데이터를 취득해서 유효 데이터를 추출하는 애플리케이션 데이터 추출부(302)를 더 제공한다.

[0064] 도 4에 도시된 구성은 다음의 동작을 행한다. 우선, 제1 애플리케이션 처리부(201)에 의한 데이터 송신 요구에 기초한 처리는 도 1에서와 같다. 한편, 제2 애플리케이션 처리부(301)는, 파일(즉, 제2 파일)의 데이터 판독 요구(즉, 취득 지시)를 행하면, 제1 애플리케이션 처리부에서와 마찬가지로 메타데이터를 판독하고, 판독하려는 데이터의 익스텐트 맵 정보(즉, 제2 맵 정보)를 취득하지만, 이에 기초한 데이터 판독 지시를 행할 경우, 애플리케이션 데이터 판독 지시부(303)는, 판독된 데이터의 수신처 어드레스를 메인 메모리(도시되지 않음)에 설정하도록 판독 지시를 제공한다. 이어서, 스토리지 장치(13)로부터 판독된 데이터는 메인 메모리에 기입됨으로써, 애플리케이션 추출부(302)가 그 데이터를 판독한다. 여기서, 송신 데이터 추출과 같이, 애플리케이션 데이터의 익스텐트 맵 정보 중에서, 유효 데이터 위치 정보를 애플리케이션 데이터 판독 지시부(303)가 애플리케이션 데이터 추출부(302)에 전달하고, 따라서 애플리케이션 데이터 추출부(302)는 그 정보에 기초하여 유효 데이터를 추출하고, 이를 애플리케이션에 전달한다. 통지 분배부(102)는, 애플리케이션 데이터 판독 커맨드에 대한 응답 통지를 애플리케이션 데이터 판독 지시부(303)에 통지한다.

[0065] 이상과 같이 동작함으로써, 데이터 전송 장치(11)에 의한 효율적인 데이터 송신과, 종래의 애플리케이션에 의한 데이터 판독을 행할 수 있게 되고, 따라서 본 발명의 각 실시를 용이하게 할 수 있다.

[0066] 또한, 제2 애플리케이션 처리부(301)에 의한 데이터 판독 요구는, Linux OS 등에 실장되어 있는 read 함수 등의 표준 시스템 콜에 의해 인터페이스되고, 이 함수 내의 처리를 제공하여 전술의 동작을 실현하도록 하는 실시형태가 바람직하다. 이러한 표준 함수에 의해 애플리케이션과 인터페이스를 행함으로써, 기존의 애플리케이션에 변경을 가할 필요가 없어지므로, 본 발명의 실시를 용이하게 할 수 있다.

[0067] (제2 실시형태)

[0068]

제2 실시형태는, 데이터 전송용으로 복수의 통신 커넥션이 설정되며, 커넥션마다 독립적으로 플로우 제어를 행하는 예를 나타내고 있다. 예를 들면, NAS와 같은 네트워크 스토리지 또는 VOD 서버에서 복수의 클라이언트 단말기, 혹은 클라이언트 단말기 상의 복수의 애플리케이션으로부터 동시에 판독 요구나 컨텐츠 분배 요구를 수취할 경우, 복수의 독립적인 커넥션에 의해 스토리지 장치(13)로부터 데이터를 판독해서 이를 네트워크(14)에 송신하는 동작을 실현 가능하다.

[0069]

도 5는 제2 실시형태에 따른 데이터 송신 시스템의 개략 구성을 나타내는 블록도이다. 도 5의 데이터 전송 장치(11)는, 제1 실시형태에 더하여, 송신 데이터 추출부(103)와 통신 처리부(104) 사이의 데이터 교환을 위해 커넥션마다 데이터를 버퍼링하는 데이터 버퍼부(106)와, 데이터 버퍼부(106)에 있어서의 커넥션마다의 버퍼 상태를 관리하는 버퍼 관리부(107)와, 버퍼 관리부(107)로부터의 통지에 기초하여 커넥션마다의 데이터의 플로우 제어를 행하는 플로우 제어부(105)를 더 포함한다.

[0070]

도 5에서, 도 1에서와 동일한 참조 부호를 갖는 각 부분은 마찬가지의 동작을 행하고, 또한 플로우 제어를 위한 동작을 더 행한다. 플로우 제어 정보의 전달은 굵은 점선으로 나타내고 있다.

[0071]

도 5의 데이터 송신 시스템의 처리 수순은 다음과 같다. 우선, 도 1에서와 같이, 제1 애플리케이션 처리부(201)가, 어느 한 애플리케이션 소프트웨어의 처리에 의해, 송신 파일, 및 필요에 따라 그 파일 내의 데이터 위치를 특정한다. 전형적인 예로서, 네트워크(14)에 접속된 다른 장치로부터의 요구에 기초하여, 파일을 특정한다. 그후, 메타데이터 판독 지시부(202)에 그 정보를 전달한다. 여기서, 금회는, 애플리케이션이 복수의 커넥션을 사용하므로, 제1 애플리케이션 처리부(201)는 커넥션을 식별하는 커넥션 식별자와 함께 특정한 파일 데이터에 대한 정보를 준다.

[0072]

메타데이터 판독 지시부(202)는, 받은 파일 정보에 기초하여, 스토리지 장치(13)로부터 파일 시스템의 메타데이터를 판독하기 위해, 데이터 전송 장치(11)의 커맨드 발행부(101)에게 커맨드 발행을 지시한다. 스토리지 장치(13)는, 이 커맨드에 기초하여, 메타데이터를 판독해서 이를 익스텐트 맵 취득부(203)에 제공한다. 익스텐트 맵 취득부(203)는, 이 메타데이터 정보를 해석하고, 판독하려는 파일 데이터의 익스텐트 맵을 취득한다. 그후, 취득한 익스텐트 맵 정보를, 송신 데이터 판독 지시부(204)가 데이터 전송 장치(11)에 제공하고, 데이터 전송 장치(11)에 대하여 전술한 애플리케이션으로부터 받은 커넥션 식별자를 제공하고, 데이터 전송 장치(11)에게 데이터를 판독하고 송신할 것을 지시한다.

[0073]

데이터 전송 장치(11)에서는, 우선, 플로우 제어부(105)가 그 지시를 받는다. 플로우 제어부(105)는, 커넥션마다 익스텐트 맵 정보를 유지한다. 즉, 플로우 제어부는 복수의 커넥션의 데이터 판독 요구를 일시적으로 유지한다. 그후, 버퍼 관리부(107)로부터 가용 버퍼를 갖는 커넥션의 통지를 받으면, 그 커넥션에 대한 송신 데이터의 판독 지시를 커맨드 발행부(101)에 대해 행한다. 초기 상태에서는, 모든 커넥션의 버퍼가 가용이므로, 버퍼 관리부(107)로부터 통지를 빼지 않고 커맨드 발행부(101)에게 송신 데이터 판독 지시를 주지만, 동일 커넥션에 대한 데이터 송신 요구를 연속해서 받으면, 버퍼 관리부(107)로부터의 사용가능 통지를 잠시 기다리는 경우가 있다.

[0074]

사용가능한 버퍼 사이즈 이하인 경우에만, 커맨드 발행부(101)에게 한 번에 커맨드 발행 요구를 하는 것이 필수적이다. 일례로서, 데이터 버퍼부(106)의 커넥션마다 128 KB의 버퍼가 유지되어 있는 경우에, 커넥션들 중 어느 하나에 64 KB 이상의 사용가능 버퍼가 제공되어 있으면, 버퍼 관리부(107)는 그 커넥션의 버퍼 가용성을 플로우 제어부(105)에 통지한다. 호스트 CPU(12)로부터 받은 익스텐트 맵은 이 사이즈보다 큰 데이터를 보여주는 것은 당연히 있을 수 있지만, 플로우 제어부(105)가 한 번에 커맨드 발행부(101)에게 사용가능한 64 KB 이하의 부분 데이터만을 요구할 수 있으므로, 이 부분 데이터(즉, 송신하려는 데이터 가운데 판독되지 않고 있으며, 상기 사이즈 이하의 데이터)의 판독을 커맨드 발행부(101)에게 요구한다. 동시에, 이 부분 데이터의 유효 데이터 위치를 송신 데이터 추출부(103)에게 전달한다. 그후, 처리를 끝낸 부분 데이터 위치를 진척 정보로서 커넥션마다 유지하고, 일시적으로 커맨드 발행 요구를 마친다.

[0075]

이상의 처리를 알기 쉽게 설명하기 위해, 데이터의 일례를 도 6에 나타낸다. 도 6은, 블록 235 내지 1000과, 유효 위치 데이터 정보로서 오프셋 124 바이트 및 길이 391602 바이트를 포함하는 익스텐트 맵이 호스트 CPU(12)로부터 제공된 예를 나타내고 있다. 이 경우, 최초의 부분 데이터는, 예를 들어 64 KB 이하의 어립수로서 블록 235 내지 362의 65412 바이트이다. 그리고, 이 부분 데이터의 유효 데이터 위치는 오프셋 124 바이트 및 길이 65412 바이트이다. 플로우 제어부(105)는 블록 235 내지 362의 판독 커맨드를 커맨드 발행부(101)에게 요구하고, 동시에 송신 데이터 추출부(103)에 대해 오프셋 124 바이트 및 길이 65412 바이트의 정보를 제공하고, 진척 정보로서 다음회 처리되는 블록은 블록 363 혹은 65413번째 바이트라고 하는 정보를 유지하고,

일시적으로 처리를 끝내고 버퍼 관리부(107)로부터의 다음의 버퍼 사용가능성 통지를 기다린다. 또한, 이상 서술한 64 KB 등의 각종 데이터 사이즈는 단지 일례이며, 필수적인 것은 아니다. 또한, 데이터 사이즈는 일정할 필요가 없으며, 본 발명은 사이즈가 매번 변할 경우도 포함한다.

[0076] 부분 데이터 처리를 일시적으로 끝내고, 다른 커넥션의 데이터 버퍼가 사용가능하다라고 하는 통지를 버퍼 관리부(107)로부터 받으면, 플로우 제어부(105)는, 그 커넥션에 대하여 부분 데이터의 판독 요구 처리를 행한다. 또한, 어떤 커넥션에 관해 제공된 익스텐트 맵이 나타내는 데이터 모두가 송신되면(도 6의 경우는, 6번째 부분 데이터에 대한 판독 요구가 완료되면), 그 커넥션에 대해 제공된 익스텐트 맵이 나타내는 모든 데이터의 송신이 완료되고, 이것이 호스트 CPU(12)에 통지된다. 호스트 CPU(12)에서는, 예를 들면 메타데이터 판독 지시부(202)가 이 통지를 수취하고, 후속하는 애플리케이션에 의한 데이터 송신 처리, 즉 메타데이터 판독 지시부(202)로부터 시작되는 일련의 처리를 실행가능하게 한다. 이에 반해서, 이 통지를 받을 때까지 다음 메타데이터 판독을 일시적으로 중단함으로써, 제1 애플리케이션 처리부(201)와의 플로우 제어를 실현한다.

[0077] 또한, 호스트 CPU(12)가 반드시 메타데이터 판독 지시부(202)에서 이러한 통지를 받을 필요는 없으며, 익스텐트 맵 취득부(203), 또는 송신 데이터 판독 지시부(204)에서도 즉, 최종적으로 애플리케이션에 대하여 백 프레셔(back pressure)가 통신되는 한, 임의의 부분이 통지를 받을 수 있다.

[0078] 또한, 이상의 예에서는 데이터 전송 장치(11)가 커넥션마다 한번에 단 하나의 익스텐트 맵을 수리(accept)하는 예를 나타냈지만, 두 개 이상을 수리하여도 된다. 수리되는 개수는 반드시 유한개이므로, 최대 개수의 익스텐트 맵을 수리한 후에는, 추가로 익스텐트 맵을 수리할 수 없는 시스템으로서 전술의 방법이 적용된다.

[0079] 커맨드 발행부(101)에서의 처리와 송신 데이터 추출부(103)에서의 처리는 도 1에서와 기본적으로 마찬가지이지만, 송신 데이터 추출부(103)는, 전술한 바와 같이 데이터 유효 위치 정보를 플로우 제어부(105)로부터 수취하고, 그 정보에 기초하여, 추출한 데이터를 데이터 버퍼부(106)에 기입한다. 데이터 버퍼부(106)에 기입할 경우의 수신처 어드레스는, 플로우 제어부(105)에서 생성되어, 송신 데이터 추출부(103)에 전달한다.

[0080] 예를 들면, 전술한 설명과 마찬가지로, 데이터 버퍼부(106)에 있어서 커넥션마다 128 KB가 확보되어 있고, 커넥션 0에 대해 버퍼 헤드 어드레스가 0×00000000, 커넥션 1에 대해 버퍼 헤드 어드레스가 0×00020000, 커넥션 2에 대해 버퍼 헤드 어드레스가 0×00040000 등과 같이 할당되어 있고, 또한, 현재 처리하고 있는 것은 커넥션 1이고 처음으로 도 6의 데이터 판독이 요구되었다고 하면, 최초의 부분 데이터의 수신처 어드레스는 0×00020000이고, 다음의 부분 데이터의 수신처 어드레스는 65413 바이트를 더한 0×0002ff85가 된다. 이와 같이하여, 수신처 어드레스를 부분 데이터 사이즈마다 증가시키면, 데이터 버퍼부(106)에는 하나의 통합적인 데이터가 제공된다. 또한, 버퍼가 예를 들면 바이트 단위의 링 버퍼라면, 0×0003ffff의 다음 바이트는 0×00020000에 기입하게 된다. 플로우 제어부(105)는, 송신 데이터 추출부(103)에 대해, 유효 데이터 위치 정보와 함께, 이러한 수신처 어드레스 정보를 전달하고, 송신 데이터 추출부(103)는 이에 기초하여 어드레스로부터 추출한 데이터를 기입함으로써, 데이터 버퍼부(106)에 대해 커넥션마다 적절한 송신 데이터를 제공할 수 있다.

[0081] 버퍼 관리부(107)는, 데이터 버퍼부(106)의 커넥션마다, 데이터 버퍼의 데이터 공급측의 기입 위치와, 데이터의 소비측의 판독 위치를 항상 파악하고, 커넥션마다 버퍼의 사용가능 상태를 파악한다.

[0082] 구체적으로는, 플로우 제어부(105)가 커맨드 발행부(101)에 대해 부분 데이터의 판독을 요구하고, 스토리지 장치(103)로부터 그 응답 통지가 돌아오면, 플로우 제어부(105)는 해당 커넥션에 대응하는 버퍼의 어느 위치까지 데이터가 기입되었는지를 알아내어, 그 위치를 진척으로서 버퍼 관리부(107)에 통지한다. 예를 들면, 도 6의 최초의 부분 데이터를 기입했을 때, 65413 바이트의 정보를, 커넥션 식별자와 함께 통지한다. 이 통지에 의해, 버퍼 관리부(107)는 커넥션 1의 버퍼의 데이터의 공급측 위치는 65413 바이트라고 알 수 있다.

[0083] 한편, 통신 처리부(104)가 데이터 버퍼부(106)로부터 송신 데이터를 판독하면, 그 판독 위치가 이동하므로, 통신 처리부(104)는 마찬가지로 판독이 완료된 바이트 위치(정확하게는, 다음 판독이 시작되는 바이트 위치)를 버퍼 관리부(107)에 통지한다.

[0084] 초기 상태에서는, 버퍼에 128 KB(정확하게는, 2의 17승인 131072 바이트)가 가용이지만, 최초에 데이터가 기입되면, 65413 바이트만큼 빈 사이즈가 감소하여 65569 바이트가 되고, 이것은 64 KB(정확하게는, 2의 16승인 65536 바이트)보다 크므로, 버퍼 가용 통지를 플로우 제어부(105)에 대해 행한다. 이어서, 플로우 제어부(105)는 다음 부분 데이터(도 6의 예에서는 65536 바이트)의 판독 요구를 발행하고, 그 데이터가 기입되면, 버퍼 사이즈가 더 감소해서 23 바이트가 되고, 이것은 64 KB보다 작으므로, 버퍼 관리부(107)는 가용 통지를 행하지 않고, 플로우 제어부(105)는 다음 부분 데이터의 판독을 보류한다. 그 후, 통신 처리부(104)가 예를 들면 1

KB(1024 B)의 데이터를 판독하면, 가용 버퍼 사이즈는 1047 바이트로 증가한다. 계속해서, 통신 처리부(104)가 판독을 행하고, 가용 사이즈가 65536 바이트 이상이 되면, 그때 버퍼 관리부(107)는 플로우 제어부(105)에 대해 가용 통지를 행한다.

[0085] 통신 처리부(104)의 송신 레이트는, 이용하고 있는 통신 프로토콜 사양이나 네트워크(14)의 대역 등에 의해 제한된다. 예를 들면, TCP 프로토콜과 같이 플로우 제어나 폭주 제어를 행하는 경우에는, 대향 장치(facing apparatus)나 네트워크(14)의 정체 상태에 따라 송신 레이트가 저하되고, 데이터 버퍼부(106)로부터의 데이터의 판독의 진행이 지연된다. 그러나, 이 경우에도, 이러한 구성에 의해 적절하게 최상위의 애플리케이션까지의 플로우 제어를 실현할 수 있다.

[0086] 이상, 본 발명의 제2 실시형태를 설명하였다. 이러한 실시형태를 이용함으로써, 제1 실시형태의 효과에 더하여, 통신 속도가 느려져도 적절히 플로우 제어를 행할 수 있게 된다. 또한, 통신 커넥션마다 플로우 제어를 행할 수 있으므로, 어떤 커넥션의 데이터 플로우가 멈추어도, 그외의 커넥션의 데이터 플로우에는 영향을 주지 않도록 동작을 행할 수 있다.

[0087] 이상의 실시형태에서는 스토리지 장치(13)에 있는 데이터를 효율적으로 송신하는 예를 설명했지만, 소정의 애플리케이션에 대해서는 스토리지 장치(13)에 없는 데이터를, 스토리지 장치(13)의 데이터를 송신하는 커넥션과 같은 커넥션에 의해 송신하는 경우도 있다. 예를 들면, 통신 처리부(104)가 TCP/IP 프로토콜보다 하위층의 통신 처리를 행할 때, 애플리케이션은 그보다 상위층의 HTTP 프로토콜에 기초한 통신을 행하는 경우에, 스토리지 장치(13)에 있는 데이터와는 다른 데이터(제어 정보 등)가 처리되는 경우나, 혹은 스토리지 장치(13)에 있는 데이터를 송신할 때 데이터의 HTTP 헤더를 송신할 필요가 있는 경우가 있다. 이 경우, 우선, HTTP 헤더만을 송신하고나서, 보디(body)로서 스토리지 장치(13)의 데이터를 송신하도록 동작할 필요가 있다. 이 동작을 실현하는 구성을 도 7에 나타낸다.

[0088] 도 7은, 도 6에 더하여, HTTP 헤더 등 스토리지 장치(13)에 없는 데이터(이하, "애플리케이션 데이터")의 송신을 요구하는 제3 애플리케이션 처리부(401)와, 그 요구를 접수하고 애플리케이션 데이터의 기입을 데이터 전송 장치(11)에 대해 요구하는 애플리케이션 데이터 기입 요구부(402)와, 애플리케이션 데이터의 기입을 행하는 애플리케이션 데이터 기입부(403)를 포함한다. 애플리케이션 데이터의 흐름은 실선으로 나타내고 있다.

[0089] 구체적으로는, 다음과 같이 동작한다. 우선, 제3 애플리케이션 처리부(401)가 소정의 방법으로 애플리케이션 데이터를 준비한다. 이것은, 예를 들면 HTTP 통신을 행하고 있을 경우에, 통신 수순, 애플리케이션 상태 또는 기타 각종 조건에 의해 정해지고나서 생성된다. 그후, 그 애플리케이션 데이터를 데이터 전송 장치(11)에 송신하기 위해, 애플리케이션 데이터와 송신 커넥션 식별자를 함께 애플리케이션 데이터 기입 요구부(402)에 전달하도록 송신을 요구한다.

[0090] 애플리케이션 데이터 기입 요구부(402)는 이것을 받으면, 플로우 제어부(105)에 그 애플리케이션 데이터 사이즈 및 데이터 기입 요구를 제공한다.

[0091] 플로우 제어부(105)는 이것을 받으면, 그 기입 요구를 일단 보류하고, 버퍼 관리부(107)로부터의 가용 통지를 기다리고, 또한, 버퍼 관리부(107)로부터 버퍼의 가용 통지가 오자마자, 기입 허가를 의미하는 응답을 애플리케이션 데이터 기입 요구부(402)로 돌려준다.

[0092] 그러면, 애플리케이션 데이터 기입 요구부(402)는 애플리케이션 데이터 기입부(403)에 기입 지시를 제공하고, 애플리케이션 데이터 기입부(403)는 실제로 데이터 버퍼에 애플리케이션 데이터를 기입한다.

[0093] 기입이 종료되면, 플로우 제어부(105)는, 스토리지 장치(13)의 데이터를 판독해서 이를 데이터 버퍼부(106)에 기입하는 경우와 마찬가지로, 기입 위치를 진척으로서 버퍼 관리부(107)에 대해 통지한다.

[0094] 이상의 일련의 처리를 완료하면, 제3 애플리케이션 처리부(401)에 대해 송신 완료를 통지한다. 통신 처리부(104)는 스토리지로부터의 데이터와 같이 이 데이터를 판독하여 송신한다.

[0095] 애플리케이션 데이터 사이즈가 버퍼 관리부(107)의 버퍼 가용 통지를 행하는 사이즈(위의 예에서는, 64 KB)보다 클 경우는, 64 KB만큼만 기입이 가능하며, 이 경우에, 애플리케이션 데이터 기입부(403)는 이 사이즈까지의 기입을 행한다. 또한, 아직 기입하지 않고 있는 데이터가 있을 경우는, 다시 애플리케이션 데이터 기입 요구부(402)가 플로우 제어부(105)에 대해 기입 요구를 제공하고, 기입하려는 데이터가 처리될 때까지 마찬가지의 처리를 반복한다.

[0096] 또한, 통신 헤더 등과 같이 애플리케이션 데이터의 기입 사이즈가 극히 작을 경우는, 버퍼 관리부(107)로부터

통지받는 가용 사이즈를, 예를 들면 애플리케이션 데이터 사이즈로 설정하기 위해, 플로우 제어부(105)는 버퍼 관리부(107)에 대해 사이즈 변경의 지시를 행할 수도 있다. 예를 들면, 20 바이트의 애플리케이션 데이터를 송신하는 경우, 플로우 제어부(105)는 버퍼 관리부(107)에 대해 다음에 20 바이트가 가용이면 통지하도록 지시하고, 버퍼 관리부(107)는 이를 따른다. 이에 따라, 버퍼가 비워질 때까지 첫되어 기다릴 필요가 없게 된다.

[0097] 이상과 같이 동작함으로써, 스토리지 장치(13)의 데이터를 송신하는 통신 커넥션과 같은 통신 커넥션에 의해 임의의 애플리케이션 데이터를 송신할 수 있다. 또한, 제3 애플리케이션 처리부(401)에 의해 행해지는 애플리케이션 데이터 송신 요구 처리는, Linux OS 등에서 실장되어 있는 send 함수와 같은 표준 시스템 콜에 의해 인터페이스되고, 이 함수 내의 처리를 전술한 동작을 실현하도록 제공하는 실시형태가 바람직하다. 이러한 표준 함수에 의해 애플리케이션과 인터페이스를 행함으로써, 기존의 애플리케이션에 변경을 가할 필요가 없어져서, 본 발명의 실시를 용이하게 할 수 있다.

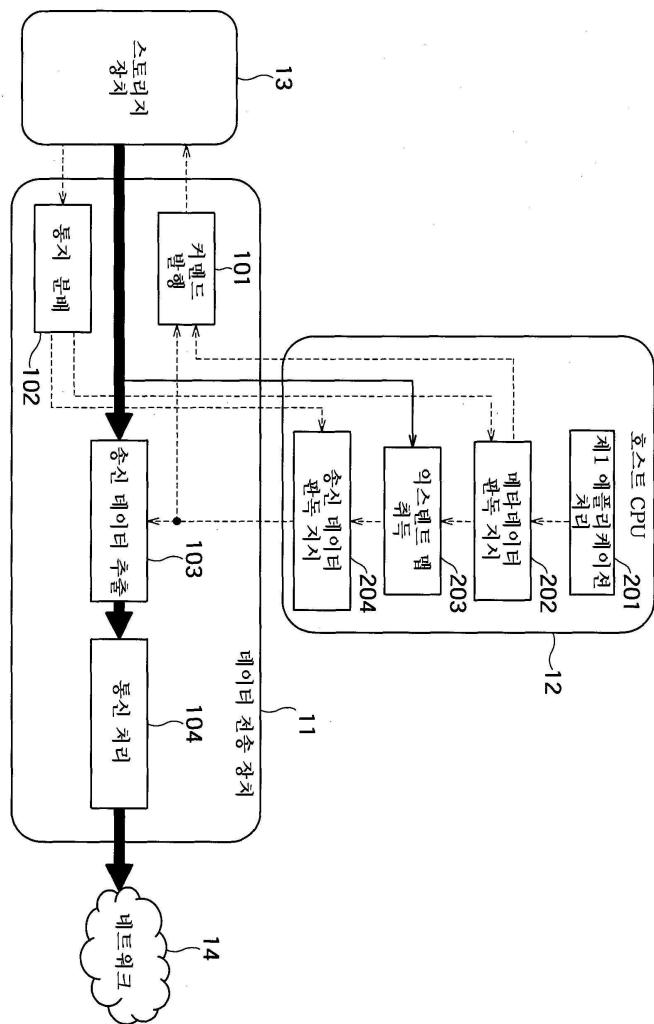
[0098] 본 발명은 상술한 개개의 실시형태에 한정되나 것은 아니며, 본 발명의 범위를 일탈하지 않고서 실시 단계에서 그 구성요소에 변경을 가하여 실현될 수도 있다. 또한, 상술한 실시형태에 개시된 구성요소들의 임의의 조합은 여러가지 발명을 구성할 수 있다. 예를 들어, 실시형태들에 나타낸 모두 구성요소들 중 일부를 생략할 수도 있다. 또한, 다른 실시형태로부터의 구성요소가 적절히 조합될 수도 있다.

[0099] 상술한 실시형태에서 설명한 데이터 송신 시스템 중 적어도 일부는 하드웨어 또는 소프트웨어로 구성될 수 있다. 소프트웨어로 구성할 경우에는, 데이터 송신 시스템의 적어도 일부의 기능을 실현하는 프로그램을 플렉서블 디스크나 CD-ROM 등의 기록 매체에 저장시키고, 컴퓨터에 의해 판독시켜서 실현할 수 있다. 기록 매체는, 자기 디스크나 광 디스크 등의 착탈가능한 것에 한정되지 않고, 하드 디스크 장치나 메모리 등의 고정형 기록 매체일 수도 있다.

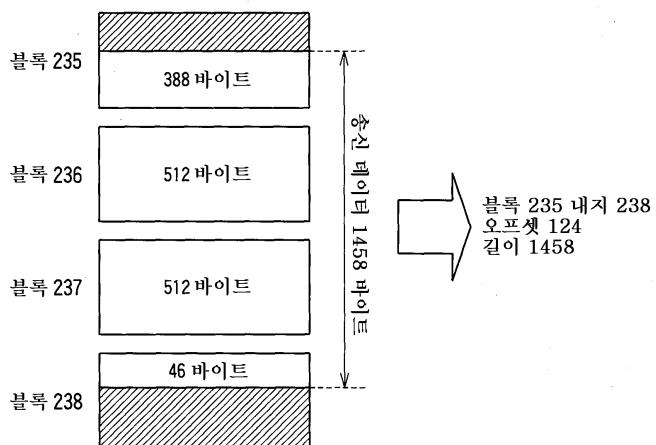
[0100] 또한, 데이터 송신 시스템의 적어도 일부의 기능을 실현하는 프로그램을, 인터넷 등의 통신 회선(무선 통신도 포함함)을 통해 배포할 수도 있다. 또한, 상기 프로그램을 암호화하거나, 변조하거나, 압축하여서, 인터넷 등의 유선 회선이나 무선 회선을 통하거나, 혹은 배포를 위한 기록 매체에 저장시켜서 배포할 수도 있다.

도면

도면1



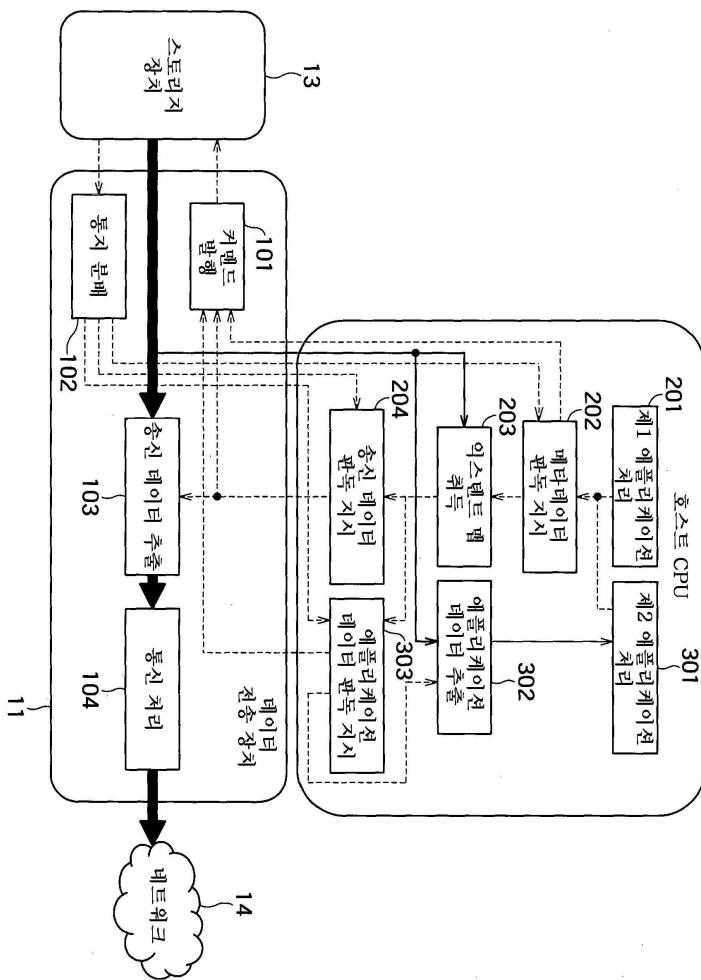
도면2



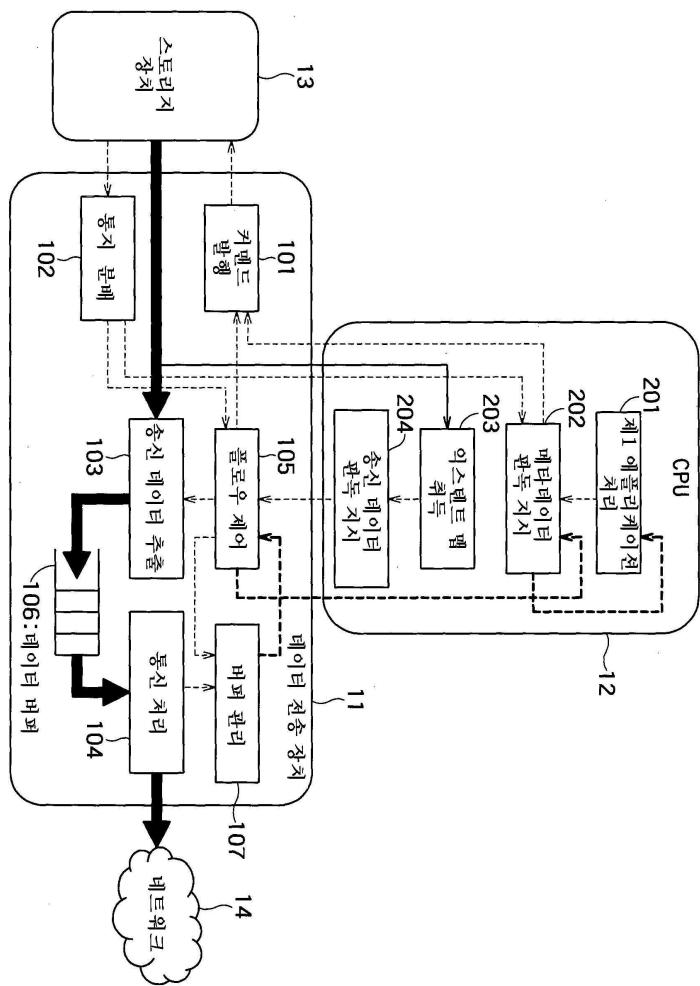
도면3

커맨드 번호	데이터 추출부의 어드레스
커맨드 0	0x20200000 ~ 0x202007ff
커맨드 1	0x20200800 ~ 0x20200fff
커맨드 2	0x20201000 ~ 0x202017ff
...	...

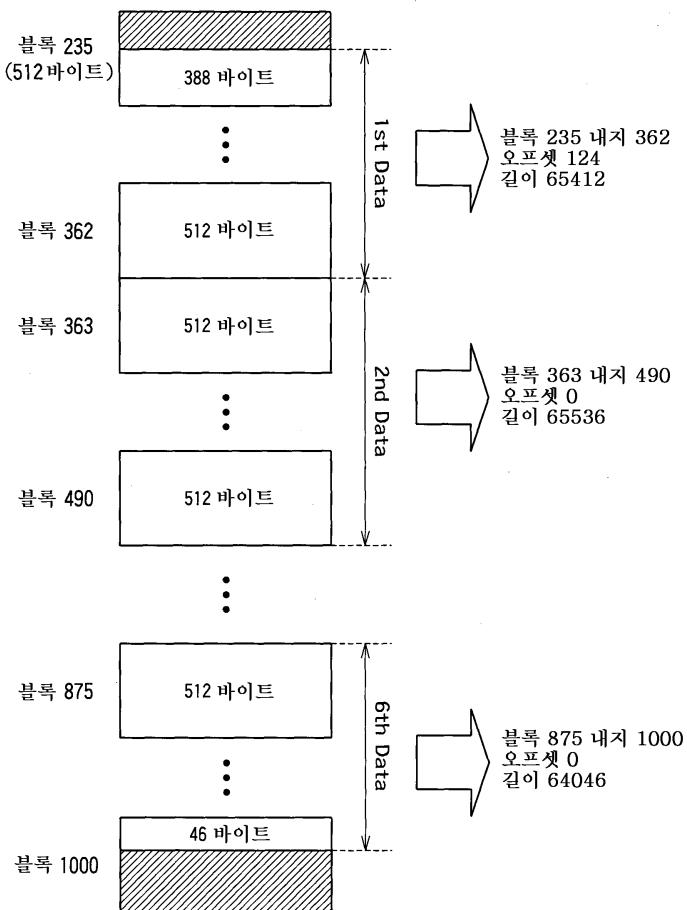
도면4



도면5



도면6



도면7

