



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2019-0097843
(43) 공개일자 2019년08월21일

(51) 국제특허분류(Int. Cl.)
G06F 15/163 (2006.01) G06F 13/16 (2006.01)
(52) CPC특허분류
G06F 15/163 (2013.01)
G06F 13/1673 (2013.01)
(21) 출원번호 10-2018-0017760
(22) 출원일자 2018년02월13일
심사청구일자 2018년02월13일

(71) 출원인
서강대학교산학협력단
서울특별시 마포구 백범로 35 (신수동, 서강대학교)
(72) 발명자
김영재
경기도 고양시 일산서구 대산로 164, 202동 902호 (주엽2동, 문촌마을2단지아파트)
김태욱
서울특별시 도봉구 마들로 551, 112동 1501호 (창동, 쌍용아파트)
(74) 대리인
유미특허법인

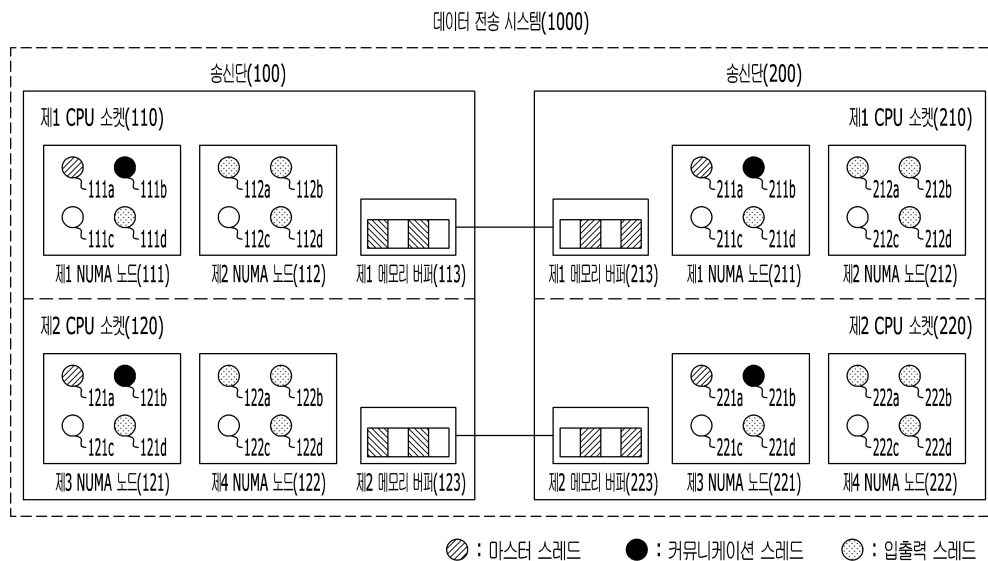
전체 청구항 수 : 총 10 항

(54) 발명의 명칭 데이터 전송 시스템 및 방법

(57) 요약

송신단의 메모리 버퍼들에 로드된 데이터를 수신단의 메모리 버퍼들로 전송하는 데이터 전송 시스템으로서, 상기 송신단 및 상기 수신단 각각은 마스터 스레드, 커뮤니케이션 스레드 또는 입출력 스레드 중 어느 하나가 할당되는 CPU 코어, 상기 CPU 코어를 적어도 하나 이상 포함하는 CPU 소켓들, 그리고 상기 CPU 소켓들의 수만큼 분할되어 상기 CPU 소켓들 각각에 배치된 메모리 버퍼들을 포함한다.

대표도



이 발명을 지원한 국가연구개발사업

과제고유번호 2015-0-00590

부처명 과학기술정보통신부

연구관리전문기관 정보통신기술진흥센터

연구사업명 정보통신방송기술개발사업

연구과제명 (총괄) 빅데이터 처리 고도화 핵심기술개발 사업 총괄 및 고성능컴퓨팅 기술을 활용한 성능 가속화 기술 개발

기 여 율 1/1

주관기관 울산과학기술원

연구기간 2017-06-01~2018-03-31

명세서

청구범위

청구항 1

데이터 전송 시스템으로서,

복수의 CPU 소켓들, 그리고

각 CPU 소켓마다 배치된 메모리 버퍼를 포함하고,

각 CPU 소켓은 마스터 스레드가 할당된 CPU 코어, 커뮤니케이션 스레드가 할당된 CPU 코어 및 입출력 스레드가 할당된 CPU 코어를 적어도 하나씩 포함하는 데이터 전송 시스템.

청구항 2

제1항에서,

각 CPU 소켓에 포함된 CPU 코어들에 할당된 스레드는 해당 CPU 소켓에 배치된 메모리 버퍼에만 접근하는 데이터 전송 시스템.

청구항 3

제1항에서,

각 CPU 소켓에 포함된 CPU 코어들에 할당된 마스터 스레드, 커뮤니케이션 스레드 및 입출력 스레드는 해당 CPU 소켓에 배치된 메모리 버퍼를 통해 전송되는 파일을 처리하고자 상호작용을 수행하는 데이터 전송 시스템.

청구항 4

제1항에서,

각 CPU 소켓이 해당 CPU 소켓에 포함된 CPU 코어들로 구성된 NUMA 노드를 다수 포함하는 경우, 마스터 스레드와 커뮤니케이션 스레드는 동일한 NUMA 노드에 포함된 CPU 코어들에 할당되는 데이터 전송 시스템.

청구항 5

제4항에서,

입출력 스레드는 마스터 스레드가 할당된 CPU 코어와 인접한 CPU 코어에 할당되는 데이터 전송 시스템.

청구항 6

데이터 전송 시스템이 데이터를 전송하는 방법으로서,

송신단의 마스터 스레드 및 커뮤니케이션 스레드가 상호작용하여 상기 송신단의 입출력 스레드를 활성화시키는 단계,

상기 활성화된 입출력 스레드가 상기 송신단의 메모리 버퍼들 중 제1 메모리 버퍼에 데이터를 로드하는 단계,

상기 제1 메모리 버퍼에 로드된 데이터에 수신단의 커뮤니케이션 스레드가 접근하여 상기 데이터를 상기 수신단의 메모리 버퍼들 중 제2 메모리 버퍼로 로드하는 단계, 그리고,

상기 제2 메모리 버퍼에 로드된 데이터를 상기 수신단의 입출력 스레드가 스토리지에 저장하는 단계를 포함하고,

상기 제1 메모리 버퍼는 상기 송신단의 마스터 스레드, 상기 송신단의 커뮤니케이션 스레드 및 상기 송신단의 입출력 스레드가 할당된 CPU 코어들을 포함하는 CPU 소켓에 배치된 메모리 버퍼이고,

상기 제2 메모리 버퍼는 상기 수신단의 커뮤니케이션 스레드 및 상기 수신단의 입출력 스레드가 할당된 CPU 코어들을 포함하는 CPU 소켓에 배치된 메모리 버퍼인 데이터 전송 방법.

청구항 7

제6항에서,

상기 송신단의 CPU 소켓이 해당 CPU 소켓에 포함된 CPU 코어들로 구성된 NUMA 노드를 다수 포함하는 경우, 마스터 스레드와 커뮤니케이션 스레드는 동일한 NUMA 노드에 포함된 CPU 코어들에 할당되는 데이터 전송 방법.

청구항 8

제7항에서,

입출력 스레드는 마스터 스레드가 할당된 CPU 코어와 인접한 CPU 코어에 할당되는 데이터 전송 방법.

청구항 9

제7항에서,

상기 수신단의 CPU 소켓이 해당 CPU 소켓에 포함된 CPU 코어들로 구성된 NUMA 노드를 다수 포함하는 경우, 마스터 스레드와 커뮤니케이션 스레드는 동일한 NUMA 노드에 포함된 CPU 코어들에 할당되는 데이터 전송 방법.

청구항 10

제9항에서,

입출력 스레드는 마스터 스레드가 할당된 CPU 코어와 인접한 CPU 코어에 할당되는 데이터 전송 방법.

발명의 설명

기술 분야

[0001] 본 발명은 데이터 전송 기술에 관한 것이다.

배경 기술

[0002] 도래하는 빅데이터 시대에 과학자들은 대용량 데이터를 전세계적으로 흩어져 있는 연구소 및 컴퓨팅 센터와 공유하거나 이들로 전송해야 하는 경우가 많아졌다. 하지만, 이미 존재하는 데이터 전송 프레임워크는 연구소 및 컴퓨팅 센터의 병렬 파일 시스템(PFS)의 특성을 반영하지 못하는 한계가 존재하였다.

[0003] 이러한 문제를 해결하기 위해 LADS(Layout-Aware Data Scheduling) 데이터 전송 프레임워크가 디자인되었다. LADS 프레임워크는 오브젝트 단위의 데이터 전송을 하여 다수의 스레드가 오브젝트 단위로 작업할 수 있게 함으로써 종단 간 데이터 전송의 효율성을 향상시켰으며, 병렬 파일 시스템 내 데이터 레이아웃을 인지하여 데이터를 스케줄링 함으로써 테라비트 네트워크(terabit-network)에서의 데이터 통신의 성능을 최적화하였다.

[0004] 그러나, LADS 프레임워크는 불균일 기억 장치 접근(Non-Uniformed Memory Access, NUMA) 구조에서 발생할 수 있는 일부 문제점들을 해결하지 못하는 단점이 존재한다. 구체적으로, 종단 간 데이터 전송의 경우 크게 스토리지, CPU 및 메모리에서 병목현상이 일어날 수 있는데, LADS 프레임워크는 데이터 레이아웃 인지 다중 스레드 아키텍처를 이용하는바, 스토리지 및 CPU에서 발생하는 병목현상을 해결할 수 있다. 하지만, LADS 프레임워크는 하나의 RMA 버퍼를 사용하는바, 메모리 컨트롤러의 과부하 및 다른 소켓의 메모리에 접근하는 문제를 야기하여 메모리에서 발생하는 병목 현상은 해결하지 못하는 한계가 있다.

발명의 내용

해결하려는 과제

[0005] 본 발명이 해결하고자 하는 과제는 CPU 소켓 마다 메모리 버퍼를 배치하는 데이터 전송 시스템 및 이를 이용한 데이터 전송 방법을 제공하는 것이다.

[0006] 본 발명이 해결하고자 하는 다른 과제는 각각의 CPU 소켓에 할당된 스레드가 해당 CPU 소켓에 배치된 메모리 버퍼에 접근하도록 하는 메모리 인지 스레드 스케줄링 기술을 제공하는 것이다.

과제의 해결 수단

- [0007] 본 발명의 일 실시예에 따른 데이터 전송 시스템은 복수의 CPU 소켓들, 그리고 각 CPU 소켓마다 배치된 메모리 버퍼를 포함하고, 각 CPU 소켓은 마스터 스레드가 할당된 CPU 코어, 커뮤니케이션 스레드가 할당된 CPU 코어 및 입출력 스레드가 할당된 CPU 코어를 적어도 하나씩 포함한다.
- [0008] 각 CPU 소켓에 포함된 CPU 코어들에 할당된 스레드는 해당 CPU 소켓에 배치된 메모리 버퍼에만 접근한다.
- [0009] 각 CPU 소켓에 포함된 CPU 코어들에 할당된 마스터 스레드, 커뮤니케이션 스레드 및 입출력 스레드는 해당 CPU 소켓에 배치된 메모리 버퍼를 통해 전송되는 파일을 처리하고자 상호작용을 수행한다.
- [0010] 각 CPU 소켓이 해당 CPU 소켓에 포함된 CPU 코어들로 구성된 NUMA 노드를 다수 포함하는 경우, 마스터 스레드와 커뮤니케이션 스레드는 동일한 NUMA 노드에 포함된 CPU 코어들에 할당된다.
- [0011] 입출력 스레드는 마스터 스레드가 할당된 CPU 코어와 인접한 CPU 코어에 할당된다.
- [0012] 본 발명의 일 실시예에 따른 데이터 전송 시스템이 데이터를 전송하는 방법은 송신단의 마스터 스레드 및 커뮤니케이션 스레드가 상호작용하여 상기 송신단의 입출력 스레드를 활성화시키는 단계, 상기 활성화된 입출력 스레드가 상기 송신단의 메모리 버퍼들 중 제1 메모리 버퍼에 데이터를 로드하는 단계, 상기 제1 메모리 버퍼에 로드된 데이터에 수신단의 커뮤니케이션 스레드가 접근하여 상기 데이터를 상기 수신단의 메모리 버퍼들 중 제2 메모리 버퍼로 로드하는 단계, 그리고, 상기 제2 메모리 버퍼에 로드된 데이터를 상기 수신단의 입출력 스레드가 스토리지에 저장하는 단계를 포함하고, 상기 제1 메모리 버퍼는 상기 송신단의 마스터 스레드, 상기 송신단의 커뮤니케이션 스레드 및 상기 송신단의 입출력 스레드가 할당된 CPU 코어들을 포함하는 CPU 소켓에 배치된 메모리 버퍼이고, 상기 제2 메모리 버퍼는 상기 수신단의 커뮤니케이션 스레드 및 상기 수신단의 입출력 스레드가 할당된 CPU 코어들을 포함하는 CPU 소켓에 배치된 메모리 버퍼이다.
- [0013] 상기 송신단의 CPU 소켓이 해당 CPU 소켓에 포함된 CPU 코어들로 구성된 NUMA 노드를 다수 포함하는 경우, 마스터 스레드와 커뮤니케이션 스레드는 동일한 NUMA 노드에 포함된 CPU 코어들에 할당된다.
- [0014] 입출력 스레드는 마스터 스레드가 할당된 CPU 코어와 인접한 CPU 코어에 할당된다.
- [0015] 상기 수신단의 CPU 소켓이 해당 CPU 소켓에 포함된 CPU 코어들로 구성된 NUMA 노드를 다수 포함하는 경우, 마스터 스레드와 커뮤니케이션 스레드는 동일한 NUMA 노드에 포함된 CPU 코어들에 할당된다.
- [0016] 입출력 스레드는 마스터 스레드가 할당된 CPU 코어와 인접한 CPU 코어에 할당된다.

발명의 효과

- [0017] 본 발명에 따르면, 각각의 CPU 소켓에 메모리 버퍼가 배치되는바, 다수의 CPU 소켓이 단일 메모리 버퍼를 사용하는 경우에 비해 메모리 컨트롤러에 발생하는 부하를 줄일 수 있다.
- [0018] 또한, 본 발명에 따르면, 각각의 CPU 소켓은 자신에게 배치된 메모리 버퍼에 접근하면 되므로, 다른 CPU 소켓에 배치된 메모리 버퍼에 접근함으로써 발생하는 메모리 접근 지연 시간을 감소시킬 수 있다.

도면의 간단한 설명

- [0019] 도 1은 한 실시예에 따른 데이터 전송 시스템의 구조도이다.
- 도 2는 한 실시예에 따른 데이터 전송 시스템이 데이터를 전송하는 프로세스를 설명한 도면이다.
- 도 3은 한 실시예에 따른 데이터 전송 시스템에 구현된 다중 메모리 버퍼와 기존의 단일 메모리 버퍼를 비교하여 설명한 도면이다.
- 도 4는 한 실시예에 따른 데이터 전송 시스템에서 소켓 기반의 스레드 스케줄링이 구현되는 방법을 설명한 도면이다.
- 도 5는 한 실시예에 따른 데이터 전송 시스템에서 NUMA 노드 기반의 스레드 스케줄링이 구현되는 방법을 설명한 도면이다.

발명을 실시하기 위한 구체적인 내용

- [0020] 아래에서는 첨부한 도면을 참고로 하여 본 발명의 실시예에 대하여 본 발명이 속하는 기술 분야에서 통상의 지식을 가진 자가 용이하게 실시할 수 있도록 상세히 설명한다. 그러나 본 발명은 여러 가지 상이한 형태로 구현될 수 있으며 여기에서 설명하는 실시예에 한정되지 않는다. 그리고 도면에서 본 발명을 명확하게 설명하기 위해서 설명과 관계없는 부분은 생략하였으며, 명세서 전체를 통하여 유사한 부분에 대해서는 유사한 도면 부호를 붙였다.
- [0021] 명세서 전체에서, 어떤 부분이 어떤 구성요소를 "포함"한다고 할 때, 이는 특별히 반대되는 기재가 없는 한 다른 구성요소를 제외하는 것이 아니라 다른 구성요소를 더 포함할 수 있는 것을 의미한다.
- [0022] 이하, 도면을 참조로 하여 본 발명의 실시예에 따른 데이터 전송 시스템 및 방법에 대해 설명한다.
- [0023] 도 1은 한 실시예에 따른 데이터 전송 시스템의 구조도이다.
- [0024] 도 1을 참고하면, 데이터 전송 시스템(100)은 송신단(100) 및 수신단(200)을 포함하고, 송신단(100) 및 수신단(200) 각각은 제1 CPU 소켓(110 및 210) 및 제2 CPU 소켓(120 및 220)을 포함한다.
- [0025] 비록 도 1에서는 송신단(100) 및 수신단(200)이 2개의 CPU 소켓만을 포함하는 것으로 도시하였으나, 다양한 실시예에 따라 2 이상의 CPU 소켓을 각각 포함할 수도 있음을 물론이다.
- [0026] 또한, 송신단(100)에 포함된 CPU 소켓(110 및 120)과 수신단(200)에 포함된 CPU 소켓(210 및 220)의 각 구성 및 기능은 동일한바, 이하 송신단(100)의 제1 CPU 소켓(110)과 수신단(200)의 제1 CPU 소켓(210)을 기준으로 설명한다.
- [0027] 송신단(100)의 제1 CPU 소켓(110)은 제1 NUMA 노드(111), 제2 NUMA 노드(112) 및 제1 메모리 버퍼(113)를 포함하고, 수신단(200)의 제1 CPU 소켓(210) 또한 제1 NUMA 노드(211), 제2 NUMA 노드(212) 및 제1 메모리 버퍼(213)를 포함한다. 도 1에 도시된 각각의 CPU 소켓에 포함되는 NUMA 노드들의 수는 단지 예시이며, 다양한 실시예에 따라 각각의 CPU 소켓은 2 이상의 NUMA 노드들을 포함할 수 있다.
- [0028] 송신단(100)의 제1 NUMA 노드(111) 및 제2 NUMA 노드(112)는 CPU 코어들(111a 내지 111d 및 112a 내지 112d)을 포함하며, 각각의 CPU 코어에는 마스터 스레드, 커뮤니케이션 스레드 또는 입출력 스레드 중 어느 하나가 할당된다.
- [0029] 예를 들면, 도 1에 도시된 바와 같이, 송신단(100)의 제1 NUMA 노드(111)에 포함된 CPU 코어들(111a 내지 111d) 중 3개의 CPU 코어에 각각 마스터 스레드, 커뮤니케이션 스레드 및 입출력 스레드가 할당될 수 있으며, 송신단(100)의 제2 NUMA 노드(112)에 포함된 CPU 코어들(112a 내지 112d) 중 3개의 CPU 코어에 입출력 스레드가 각각 할당될 수 있다.
- [0030] 송신단(100)의 제1 메모리 버퍼(113)는 송신단(100)에 포함된 각 CPU 소켓마다 배치된다. 예를 들면, 도 1에 도시된 바와 같이, 송신단(100)에는 2개의 CPU 소켓들(110 및 120)이 존재하는바, 동일한 사이즈의 2개의 메모리 버퍼들(113 및 123)이 CPU 소켓들(110 및 120) 각각에 배치된다.
- [0031] 본 발명에 따르면, 각각의 CPU 소켓에 메모리 버퍼가 배치되는바, 다수의 CPU 소켓이 단일 메모리 버퍼를 사용하는 경우에 비해 메모리 컨트롤러에 발생하는 부하를 줄일 수 있다.
- [0032] 이 경우, CPU 소켓들(110 및 120) 각각에 포함된 CPU 코어에 할당된 스레드는 해당 소켓에 배치된 메모리 버퍼에만 접근한다.
- [0033] 예를 들면, 도 1에서, 제1 CPU 소켓(110)에 포함된 CPU 노드들(111a 내지 111d 및 112a 내지 112d)에 각각 할당된 스레드들은 제1 메모리 버퍼(113)로 접근하며, 제2 CPU 소켓(120)에 포함된 CPU 노드들(121a 내지 121d 및 122a 내지 122d)에 각각 할당된 스레드들은 동일하게 제2 메모리 버퍼(123)로 접근한다.
- [0034] 즉, 각 CPU 소켓의 모든 스레드들과 커넥션들은 다른 소켓과 독립적으로 동작하며, 각 CPU 소켓에 포함된 CPU 코어들에 할당된 마스터 스레드, 커뮤니케이션 스레드 및 입출력 스레드는 해당 CPU 소켓에 배치된 메모리 버퍼를 통해 전송되는 파일을 처리하고자 상호작용을 수행한다.
- [0035] 본 발명에 따르면, 각각의 CPU 소켓에 포함된 스레드들은 해당 CPU 소켓에 배치된 메모리 버퍼에 접근하면 되므로, 다른 CPU 소켓에 배치된 메모리 버퍼에 접근함으로써 발생하는 메모리 접근 지연 시간을 감소시킬 수 있다.
- [0036] 수신단(200)은 송신단(100)과 동일한 구조를 가질 수 있다. 즉, 수신단(200)에 포함된 CPU 소켓의 수는 송신단(100)에 포함된 CPU 소켓의 수와 동일하며, 수신단(200)에 포함된 CPU 소켓의 구조는 송신단(100)에 포함된 대

응하는 CPU 소켓의 구조와 동일하게 구성될 수 있다.

- [0037] 도 2는 한 실시예에 따른 데이터 전송 시스템이 데이터를 전송하는 프로세스를 설명한 도면이다.
- [0038] 도 2에서, 송신단(100) 및 수신단(200)에 포함된 CPU 소켓들 각각은 마스터 스레드 및 커뮤니케이션 스레드를 각각 하나씩 포함하고, 다수의 입출력 스레드를 포함할 수 있으며, 각각의 스레드는 아래에 설명되는 것과 같은 기능을 발휘하여 송신단(100)의 메모리 버퍼들(113 및 123)에 로드된 데이터를 수신단(200)으로 전송한다.
- [0039] 이때, 송신단(100)에 포함된 CPU 소켓(110 및 120)과 수신단(200)에 포함된 CPU 소켓(210 및 220)의 각 구성 및 기능은 동일하나, 이하 송신단(100)의 제1 CPU 소켓(110)과 수신단(200)의 제1 CPU 소켓(210) 사이의 데이터 통신을 중심으로 설명한다.
- [0040] 도 2를 참고하면, 송신단(100)의 입출력 스레드(111b, 112a, 112b 및 112d)를 활성화하기 위해, 송신단(100)의 마스터 스레드(111a) 및 커뮤니케이션 스레드(111d)가 상호작용한다.
- [0041] 구체적으로, 송신단(100) 측의 마스터 스레드(111a)는 전송할 파일의 데이터 청크(data chunk) 레이아웃을 파악하고(S100), 파악한 레이아웃을 커뮤니케이션 스레드(111d)의 작업 큐에 전송한다(S101).
- [0042] 레이아웃을 수신한 커뮤니케이션 스레드(111d)는 새로운 파일 요청을 수신단(200)의 커뮤니케이션 스레드(211d)로 전송하며(S103), 커뮤니케이션 스레드(211d)는 수신한 요청을 마스터 스레드(211a)의 작업 큐에 전송한다(S105).
- [0043] 마스터 스레드(211a)는 새로운 파일 및 새로운 파일에 대한 식별자를 생성하고(S107), 커뮤니케이션 스레드(211d)의 작업 큐에 새로운 파일에 대한 식별자 요청을 전송한다(S109).
- [0044] 커뮤니케이션 스레드(211d)는 상기 요청을 송신단(100)의 커뮤니케이션 스레드(111d)로 전송하고(S111), 커뮤니케이션 스레드(111d)는 상기 요청을 마스터 스레드(111a)의 작업 큐로 전송하며(S113), 마스터 스레드(111a)는 해당 파일의 청크 정보들을 OST 큐에 전송한다(S115).
- [0045] 마스터 스레드(111a)는 파일이 스트라이핑(striping)되어 있는 OST의 개수만큼 입출력 스레드들(111b, 112a, 112b 및 112d)을 활성화시킨다(S117).
- [0046] 활성화된 입출력 스레드들(111b, 112a, 112b 및 112d)은 OST 큐들을 순회하며 청크 정보를 받아서 데이터 청크를 제1 메모리 버퍼(113)에 로드하면(S119), 제1 메모리 버퍼(113)에 데이터 청크가 저장된다(S121). 즉, 활성화된 입출력 스레드들(111b, 112a, 112b 및 112d)은 제1 CPU 소켓(110)에 배치된 제1 메모리 버퍼(113)에 데이터를 로드하며, 제2 CPU 소켓(120)에 배치된 제2 메모리 버퍼(123)에 데이터를 로드하지 않는다.
- [0047] 또한, 활성화된 입출력 스레드들(111b, 112a, 112b 및 112d)은 데이터 청크 정보를 수신단(200)으로 전송하라는 요청을 커뮤니케이션 스레드(111d)의 작업 큐에 전송한다(S123).
- [0048] 커뮤니케이션 스레드(111d)는 수신단(200)의 커뮤니케이션 스레드(211d)로 상기 요청을 전달하며(S125), 수신단(200)의 커뮤니케이션 스레드(211d)가 송신단(100)의 제1 메모리 버퍼(113)에 있는 데이터 청크에 접근하여(S127), 송신단(100)의 제1 메모리 버퍼(113)로부터 수신단(200)의 제1 메모리 버퍼(213)로 데이터 청크를 로드한다(S129). 즉, 송신단(100)과 마찬가지로 커뮤니케이션 스레드(211d)는 제1 메모리 버퍼(213)에 로드된 데이터에만 접근한다. 이후, 제1 메모리 버퍼(113)에 데이터 청크가 저장된다(S131).
- [0049] 이후, 커뮤니케이션 스레드(211d)가 입출력 스레드들(211b, 212a, 212b 및 212d)을 각 OST 큐에 배치하면(S133), 입출력 스레드들(211b, 212a, 212b 및 212d)은 제1 메모리 버퍼(213)의 데이터 청크를 OST 스토리지에 기록한다(S135). 이로써, 하나의 데이터 청크에 대한 전송이 완료되며 모든 데이터 청크들에 대한 전송이 완료될 때까지 상기 단계들이 반복된다.
- [0050] 도 3은 한 실시예에 따른 데이터 전송 시스템에 구현된 다중 메모리 버퍼와 기존의 단일 메모리 버퍼를 비교하여 설명한 도면이다.
- [0051] 도 3을 참고하면, 기존의 단일 메모리 버퍼 시스템(300)은 다수의 CPU 소켓(310 및 320)에 대해 단일 메모리 버퍼(325)가 배치된다.
- [0052] 이 경우, 제2 CPU 소켓(320)에 포함된 CPU 코어들에 할당된 스레드들(321 내지 324)은 제2 CPU 소켓(320)에 할당된 메모리 버퍼(325)에 접근하면 되는바 문제가 되지 않으나, 제1 CPU 소켓(310)에 포함된 CPU 코어들에 할당된 스레드들(311 내지 314)은 멀리 배치된 메모리 버퍼(325)에 접근해야하므로 이로 인해 메모리 접근 지연 시

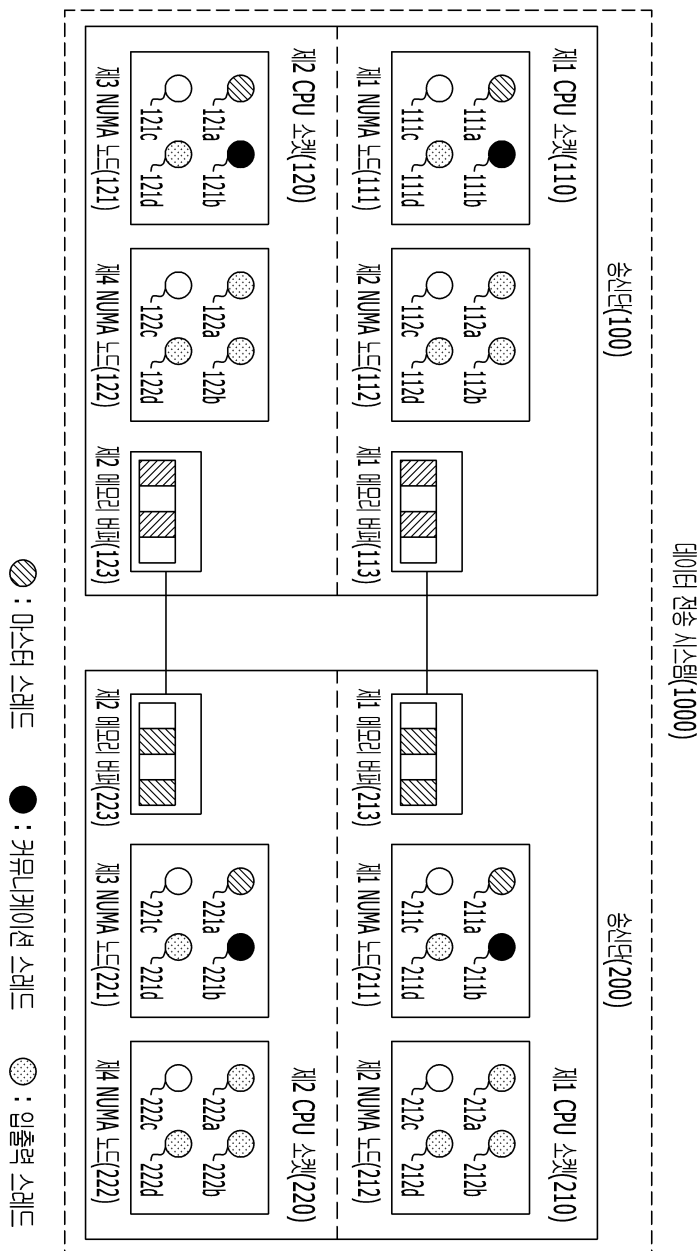
간이 발생하게 된다.

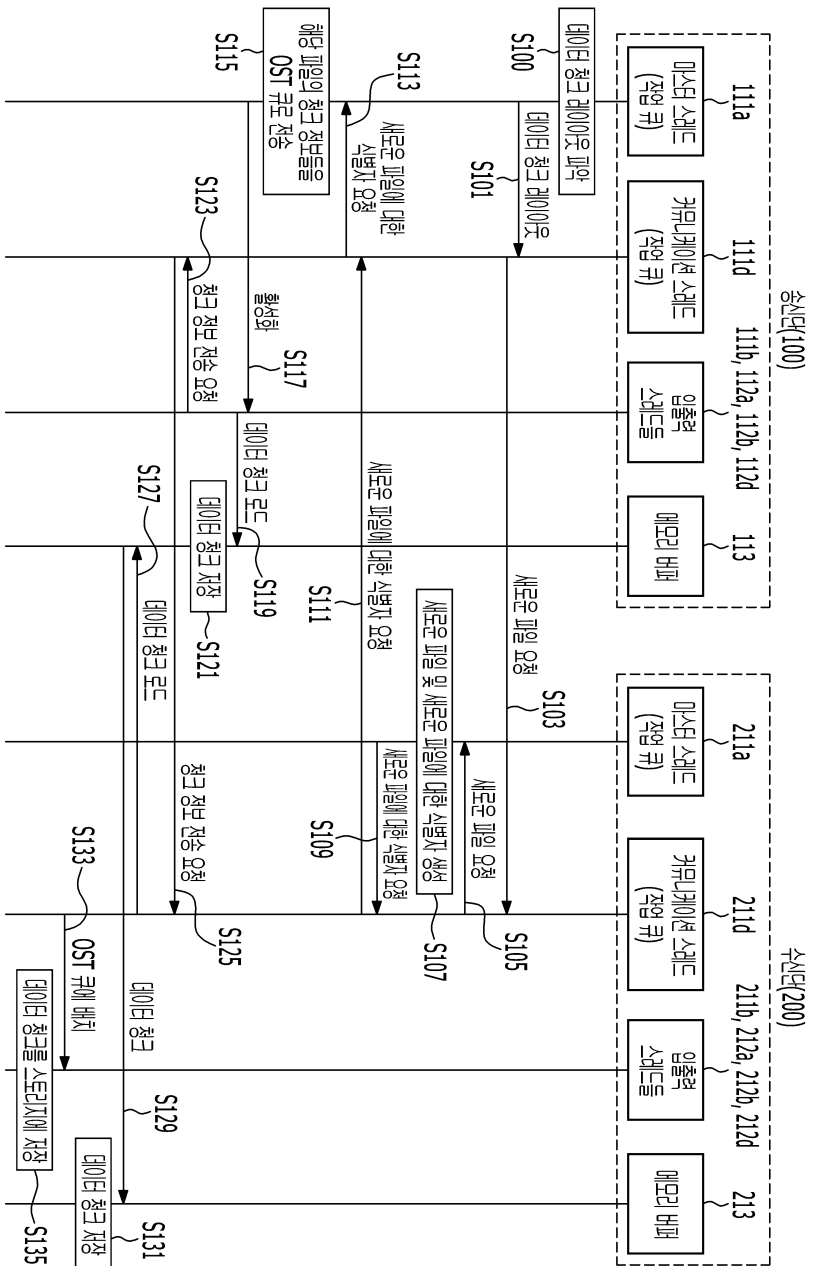
- [0053] 또한, 스레드들(311 내지 314 및 321 내지 324)이 하나의 메모리 버퍼(325)에 접근하는 경우, 메모리 버퍼(325)를 호스팅하는 제2 CPU 소켓(320)에 과부하가 일어나게 되는 문제가 발생한다.
- [0054] 이와 달리, 본 발명의 다중 메모리 버퍼 시스템(400)은 제1 CPU 소켓(410)에 대해 제1 메모리 버퍼(415)가 배치되고, 제2 CPU 소켓(420)에 대해 제2 메모리 버퍼(425)가 배치된다.
- [0055] 이 경우, 제1 CPU 소켓(410)에 포함된 CPU 코어들에 할당된 스레드들(411 내지 414)은 제1 메모리 버퍼(415)에 접근하면 되고, 제2 CPU 소켓(420)에 포함된 CPU 코어들에 할당된 스레드들(421 내지 424)은 제1 메모리 버퍼(425)에 접근하면 되는바 기존의 단말 메모리 버퍼 시스템(300)과 달리 메모리 접근 지연 시간이 발생하지 않게 된다.
- [0056] 특히, 입출력 스레드들(413, 414, 423 및 424)이 증가하는 경우에도 각 입출력 스레드는 자신이 배치된 CPU 소켓이 존재하는 메모리 버퍼에 접근하면 되는바 단말 메모리 버퍼 시스템(300)보다 메모리 컨트롤러(미도시)의 부하가 크게 줄어 들 수 있다.
- [0057] 도 4는 한 실시예에 따른 데이터 전송 시스템에서 소켓 기반의 스레드 스케줄링이 구현되는 방법을 설명한 도면이다.
- [0058] 도 4를 참고하면, 다중 메모리 버퍼 시스템(500)이 다수의 CPU 소켓들(510 및 520) 각각에 메모리 버퍼(515 및 525)를 포함하고 있더라도, 경우에 따라 제1 CPU 소켓(510)에 포함된 CPU 코어들에 할당된 스레드들(511 내지 514)이 제2 메모리 버퍼(525)에 접근할 수도 있고, 반대로 제2 CPU 소켓(520)에 포함된 CPU 코어들에 할당된 스레드들(521 내지 524)이 제1 메모리 버퍼(515)에 접근할 수도 있다.
- [0059] 이를 방지하기 위해, 각각의 CPU 소켓들(510 및 520)은 마스터 스레드, 커뮤니케이션 스레드 및 입출력 스레드를 적어도 하나씩 포함한다.
- [0060] 예를 들면, 도 4에 도시된 바와 같이, 제1 CPU 소켓(510)은 제1 마스터 스레드(511), 제1 커뮤니케이션 스레드(512) 및 제1 입출력 스레드(513 및 514)를 포함하고, 제2 CPU 소켓(520)은 제2 마스터 스레드(521), 제2 커뮤니케이션 스레드(522) 및 제2 입출력 스레드(523 및 524)를 포함한다.
- [0061] 또한, 각각의 CPU 소켓에 포함된 스레드들은 해당 CPU 소켓을 통해 전송되는 파일을 처리하고자 상호작용을 수행하고, 다른 CPU 소켓을 통해 전송되는 파일을 처리하지 않는다. 즉, 모든 스레드들은 각각의 CPU 소켓에 고정되어 작업한다.
- [0062] 구체적으로, 제1 CPU 소켓(510)에 포함된 제1 마스터 스레드(511) 제1 메모리 버퍼(515)를 통해 전송되는 파일을 처리하고자 제1 CPU 소켓(510)에 포함된 제1 커뮤니케이션 스레드(512)와 상호작용하고, 제1 커뮤니케이션 스레드(512)는 제1 메모리 버퍼(515)만을 운영한다. 또한, 제1 CPU 소켓(510)에 포함된 제1 입출력 스레드(513 및 514)는 제1 CPU 소켓(510)을 통해 할당된 오브젝트들에 대해서만 작업한다. 이는 제2 CPU 소켓(520)에서도 동일하게 적용된다.
- [0063] 도 5는 한 실시예에 따른 데이터 전송 시스템에서 NUMA 노드 기반의 스레드 스케줄링이 구현되는 방법을 설명한 도면이다.
- [0064] 도 5를 참고하면, 각 CPU 소켓이 해당 CPU 소켓에 포함된 CPU 코어들로 구성된 복수의 불균일 기억 장치 접근 노드들(이하, NUMA 노드들)을 포함하는 경우, CPU 소켓에 포함된 스레드들의 상호작용 및 공평한 코어 사용량이 중요하다. 만일 마스터 스레드와 커뮤니케이션 스레드가 서로 다른 NUMA 노드들에 할당되면, 스레드들의 상호작용 시 서로 다른 NUMA 노드 메모리 접근으로 인해 성능이 저하된다.
- [0065] 따라서, CPU 소켓에 포함된 마스터 스레드와 커뮤니케이션 스레드는 동일한 NUMA 노드들에 포함된 CPU 코어들에 각각 할당된다. 예를 들면, 도 5에 도시된 바와 같이, CPU 소켓(610)에 할당된 마스터 스레드(611a) 및 커뮤니케이션 스레드(611b)는 제1 NUMA 노드(611)에 포함된 CPU 코어들에 각각 할당될 수 있다.
- [0066] 또한, CPU 소켓의 성능의 최적화를 위해 마스터 스레드와 입출력 스레드를 최대한 가깝게 배치하고자, 입출력 스레드는 마스터 스레드가 할당된 CPU 코어와 인접한 CPU 코어에 할당된다. 예를 들면, 도 5에 도시된 바와 같이, 제2 NUMA 노드(612)에 포함된 CPU 코어들 중에서 마스터 스레드(611a)와 더 가까운 CPU 코어(612a 및 612c)에 입출력 스레드가 각각 할당될 수 있다.

- [0067] 본 발명에 따르면, 각각의 CPU 소켓에 메모리 버퍼가 배치되는바, 다수의 CPU 소켓이 단일 메모리 버퍼를 사용하는 경우에 비해 메모리 컨트롤러에 발생하는 부하를 줄일 수 있다.
- [0068] 또한, 본 발명에 따르면, 각각의 CPU 소켓은 자신에게 배치된 메모리 버퍼에 접근하면 되므로, 다른 CPU 소켓에 배치된 메모리 버퍼에 접근함으로써 발생하는 메모리 접근 지연 시간을 감소시킬 수 있다.
- [0069] 이상에서 본 발명의 실시예에 대하여 상세하게 설명하였지만 본 발명의 권리범위는 이에 한정되는 것은 아니고 다음의 청구범위에서 정의하고 있는 본 발명의 기본 개념을 이용한 당업자의 여러 변형 및 개량 형태 또한 본 발명의 권리범위에 속하는 것이다.

도면

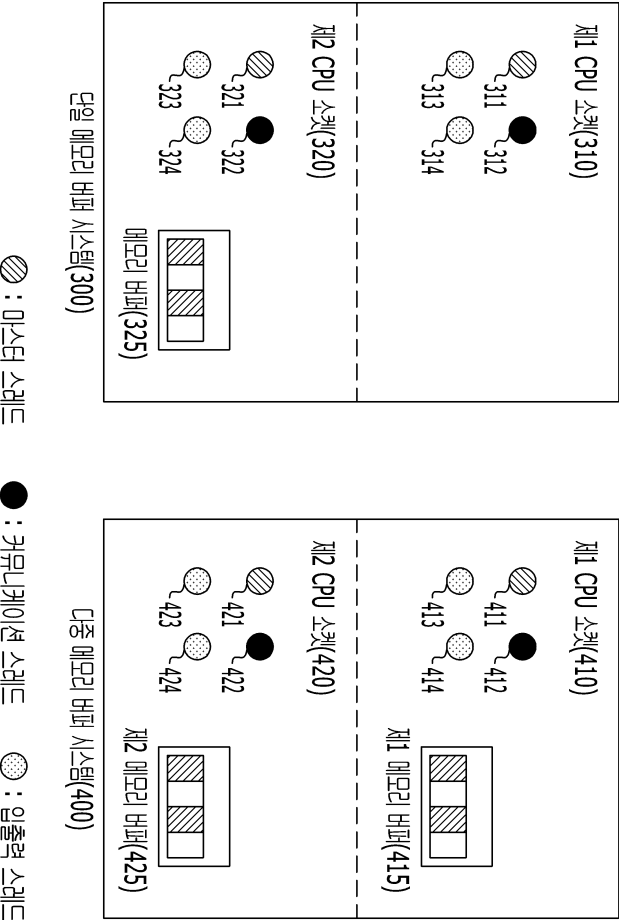
도면1



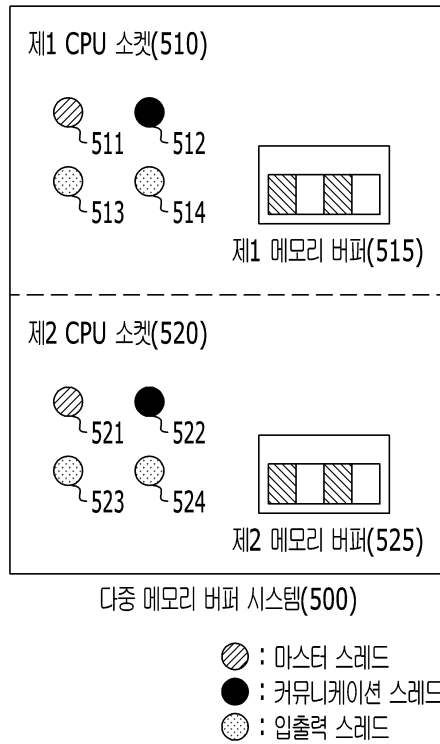


도면2

도면3



도면4



도면5

