



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2015-0132432
(43) 공개일자 2015년11월25일

(51) 국제특허분류(Int. Cl.)
G06F 15/167 (2006.01) *G06F 12/10* (2006.01)
G06F 13/16 (2006.01) *G06F 15/00* (2006.01)
G06F 9/50 (2006.01) *H04L 29/08* (2006.01)

(52) CPC특허분류
G06F 15/167 (2013.01)
G06F 12/1027 (2013.01)

(21) 출원번호 10-2015-7029330

(22) 출원일자(국제) 2014년03월13일
 심사청구일자 없음

(85) 번역문제출일자 2015년10월14일

(86) 국제출원번호 PCT/US2014/025145

(87) 국제공개번호 WO 2014/151168
 국제공개일자 2014년09월25일

(30) 우선권주장
 13/831,753 2013년03월15일 미국(US)

(71) 출원인
 마이크로소프트 테크놀로지 라이센싱, 엘엘씨
 미국 워싱턴주 (우편번호 : 98052) 레드몬드 원
 마이크로소프트 웨이

(72) 발명자
 하퍼 테이비드 티. 3세
 미국 워싱턴주 98052-6399 레드몬드 원 마이크로
 소프트 웨이 엘씨에이 - 인터내셔널 패턴즈 마이
 크로소프트 코포레이션 내
 (뒷면에 계속)

(74) 대리인
 김태홍, 김진희

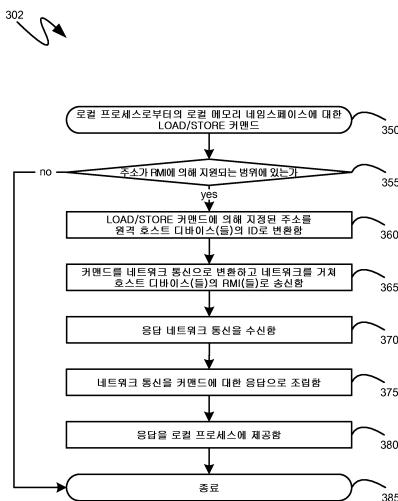
전체 청구항 수 : 총 10 항

(54) 발명의 명칭 네트워크를 통한 메모리 공유

(57) 요 약

물리적으로 구별되는 네트워크화된 컴퓨팅 디바이스들 간에 메모리가 공유된다. 각각의 컴퓨팅 디바이스는 로컬적으로 실행 중인 프로세스들로부터 커맨드를 수신하고 이러한 커맨드를 원격 컴퓨팅 디바이스로 전송 가능한 형태들로 변환하는 원격 메모리 인터페이스(Remote Memory Interface)(RMI)를 포함한다. RMI는 또한 그 RMI에 보내지는 원격 통신을 수신하고 그 통신을 로컬 메모리로 보내지는 커맨드로 변환한다. 공유되는 저장 용량의 양이 중앙 제어기(단일의 제어기, 제어기들의 계층적 집합체), 또는 피어-투-피어 협상에 의해 통보된다. 원격 고속 비휘발성 저장 매체로 보내지는 요청들이 검출되거나 플래그 설정되고, 요청을 생성하는 프로세스는 프로세스가 효율적으로 재시작될 수 있도록 일시 중단된다. 원격 메모리에 의해 제공되는 저장 용량이 로컬적으로 실행 중인 프로세스의 프로세스 공간에 매핑된다.

대 표 도 - 도3b



(52) CPC특허분류

G06F 13/1663 (2013.01)

G06F 15/00 (2013.01)

G06F 9/50 (2013.01)

H04L 67/1097 (2013.01)

(72) 발명자

버거 더글라스 크리스토퍼

미국 워싱턴주 98052-6399 레드몬드 원 마이크로소프트 웨이 엘씨에이 - 인터내셔널 패턴츠 마이크로소프트 코포레이션 내

피터슨 에릭 씨.

미국 워싱턴주 98052-6399 레드몬드 원 마이크로소프트 웨이 엘씨에이 - 인터내셔널 패턴츠 마이크로소프트 코포레이션 내

몰츠 데이비드 에이.

미국 워싱턴주 98052-6399 레드몬드 원 마이크로소프트 웨이 엘씨에이 - 인터내셔널 패턴츠 마이크로소프트 코포레이션 내

명세서

청구범위

청구항 1

다수의 컴퓨팅 디바이스들 간에 메모리 저장 용량을 공유하는 방법에 있어서,

제1 컴퓨팅 디바이스에서, 상기 제1 컴퓨팅 디바이스 상에서 실행 중인 프로세스로부터 상기 제1 컴퓨팅 디바이스의 로컬적으로 주소 지정 가능한 메모리 네임스페이스(locally addressable memory namespace)에서의 메모리 주소를 지정하는 메모리 중심 요청(memory-centric request)을 수신하는 단계;

상기 제1 컴퓨팅 디바이스에서, 상기 지정된 메모리 주소가 제2 컴퓨팅 디바이스 상에 설치된 메모리에 의해 지원되는 것으로 결정하는 단계;

상기 제1 컴퓨팅 디바이스에서, 상기 수신된 요청을 상기 제2 컴퓨팅 디바이스로 보내지는 네트워크 통신으로 변환하는 단계;

상기 제1 컴퓨팅 디바이스에서, 상기 제2 컴퓨팅 디바이스로부터의 네트워크 통신을 수신하는 단계;

상기 제1 컴퓨팅 디바이스에서, 상기 제2 컴퓨팅 디바이스로부터의 상기 수신된 네트워크 통신을 상기 요청에 대한 응답으로 변환하는 단계; 및

상기 제1 컴퓨팅 디바이스에서, 상기 응답을 상기 프로세스에 제공하는 단계를 포함하는, 메모리 저장 용량을 공유하는 방법.

청구항 2

제1항에 있어서,

상기 제2 컴퓨팅 디바이스에서, 상기 제2 컴퓨팅 디바이스로 보내진 상기 네트워크 통신을 수신하는 단계;

상기 제2 컴퓨팅 디바이스에서, 상기 제2 컴퓨팅 디바이스로 보내진 상기 네트워크 통신을 메모리 중심 요청으로 변환하는 단계;

상기 제2 컴퓨팅 디바이스에서, 상기 제2 컴퓨팅 디바이스 상에 설치된 상기 메모리의 일부분에 대해 상기 메모리 중심 요청을 수행하는 단계;

상기 제2 컴퓨팅 디바이스에서, 상기 제2 컴퓨팅 디바이스 상에 설치된 상기 메모리의 상기 일부분에 대해 상기 메모리 중심 요청을 수행한 것에 대한 응답을 수신하는 단계; 및

상기 제2 컴퓨팅 디바이스에서, 상기 메모리 중심 요청을 수행한 것에 대한 상기 응답을 상기 제2 컴퓨팅 디바이스로부터의 상기 네트워크 통신으로 변환하는 단계를 더 포함하는, 메모리 저장 용량을 공유하는 방법.

청구항 3

제1항에 있어서,

상기 다수의 컴퓨팅 디바이스들 간에 공유되어야 하는 상기 제1 컴퓨팅 디바이스의 메모리의 일부분을 식별하는 단계; 및

상기 제1 컴퓨팅 디바이스의 상기 메모리의 상기 식별된 일부분이 상기 제1 컴퓨팅 디바이스의 상기 로컬적으로 주소 지정 가능한 메모리 네임스페이스의 일부로 되는 것을 방지하는 단계를 더 포함하는, 메모리 저장 용량을 공유하는 방법.

청구항 4

제3항에 있어서, 상기 식별하는 단계는, 상기 다수의 컴퓨팅 디바이스들 간의 상기 메모리 저장 용량의 공유를 조정하는 메모리 공유 제어기로부터 수신되는 통신에 응답하여 수행되는, 메모리 저장 용량을 공유하는 방법.

청구항 5

제3항에 있어서, 상기 식별하는 단계는, 상기 다수의 컴퓨팅 디바이스들 간의 피어-투-피어 협상(peer-to-peer negotiation)에 응답하여 수행되는, 메모리 저장 용량을 공유하는 방법.

청구항 6

제1항의 단계들을 수행하는 것에 관련된 컴퓨터 실행 가능 명령어들을 포함하는, 하나 이상의 컴퓨터 판독 가능 매체.

청구항 7

다수의 컴퓨팅 디바이스들 간에 메모리 저장 용량을 공유하는 시스템에 있어서,

제1 컴퓨팅 디바이스; 및

상기 제1 컴퓨팅 디바이스와 물리적으로 구별되는 제2 컴퓨팅 디바이스를 포함하고;

상기 제1 컴퓨팅 디바이스는,

제1 로컬적으로 주소 지정 가능한 메모리 네임스페이스;

제1 메모리;

제1 원격 메모리 인터페이스; 및

상기 제1 컴퓨팅 디바이스 상에서 실행 중인 제1 프로세스를 포함하고,

상기 제2 컴퓨팅 디바이스는,

제2 운영 체제;

제2 메모리로서, 상기 제2 메모리의 일부분은 상기 시스템의 다른 컴퓨팅 디바이스들과 공유 가능하고, 상기 일부분은 상기 제2 운영 체제에 의해 구분되는 것인 상기 제2 메모리; 및

상기 메모리의 상기 일부분에 직접 액세스할 수 있는 제2 원격 메모리 인터페이스를 포함하고,

상기 제1 로컬적으로 주소 지정 가능한 메모리 네임스페이스는, 상기 제1 컴퓨팅 디바이스의 상기 제1 메모리 및 상기 제2 컴퓨팅 디바이스의 상기 제2 메모리의 상기 일부분 둘 다에 의해 지원되는, 메모리 저장 용량을 공유하는 시스템.

청구항 8

제7항에 있어서, 상기 제1 원격 메모리 인터페이스는,

상기 제1 프로세스로부터, 상기 제1 로컬적으로 주소 지정 가능한 메모리 네임스페이스에서의 메모리 주소를 지정하는 메모리 중심 요청을 수신하는 단계;

상기 지정된 메모리 주소가 상기 제2 메모리의 상기 일부분에 의해 지원되는 상기 제1 로컬적으로 주소 지정 가능한 메모리 네임스페이스의 일부분에 대응하는 것으로 결정하는 단계;

상기 수신된 요청을 상기 제2 컴퓨팅 디바이스로 보내지는 네트워크 통신으로 변환하는 단계;

상기 제2 컴퓨팅 디바이스로부터의 네트워크 통신을 수신하는 단계;

상기 제2 컴퓨팅 디바이스로부터의 상기 수신된 네트워크 통신을 상기 요청에 대한 응답으로 변환하는 단계; 및

상기 응답을 상기 제1 프로세스에 제공하는 단계를 수행하는, 메모리 저장 용량을 공유하는 시스템.

청구항 9

제7항에 있어서, 공유 가능하게 되어야 하는 상기 제1 메모리의 일부분 및 상기 제2 메모리의 상기 일부분을 결정하는 제1 메모리 공유 제어기를 더 포함하는, 메모리 저장 용량을 공유하는 시스템.

청구항 10

제1 컴퓨팅 디바이스 상에 물리적으로 설치된 원격 메모리 인터페이스 유닛에 있어서,
 상기 원격 메모리 인터페이스 유닛은,
 상기 컴퓨팅 디바이스 상에서 실행 중인 프로세스로부터 상기 제1 컴퓨팅 디바이스의 로컬적으로 주소 지정 가능한 메모리 네임스페이스에서의 메모리 주소를 지정하는 메모리 중심 요청을 수신하는 단계로서, 상기 로컬적으로 주소 지정 가능한 메모리 네임스페이스는 상기 제1 컴퓨팅 디바이스 상에 설치된 제1 메모리 및 제2 컴퓨팅 디바이스 상에 설치된 제2 메모리의 일부분 둘 다에 의해 지원되는 것인 상기 메모리 중심 요청을 수신하는 단계;
 상기 지정된 메모리 주소가 상기 제2 컴퓨팅 디바이스 상에 설치된 상기 제2 메모리의 상기 일부분에 대응하는 것으로 결정하는 단계로서, 상기 제2 컴퓨팅 디바이스는 상기 제1 컴퓨팅 디바이스로부터 멀리 떨어져 있는 것인 상기 결정하는 단계;
 상기 수신된 요청을 상기 제2 컴퓨팅 디바이스로 보내지는 네트워크 통신으로 변환하는 단계;
 상기 제2 컴퓨팅 디바이스로부터의 네트워크 통신을 수신하는 단계;
 상기 제2 컴퓨팅 디바이스로부터의 상기 수신된 네트워크 통신을 상기 요청에 대한 응답으로 변환하는 단계; 및
 상기 응답을 상기 프로세스에 제공하는 단계를 수행하도록 구성되어 있는, 원격 메모리 인터페이스 유닛.

발명의 설명

배경기술

[0001] 컴퓨팅 디바이스들 간의 통신의 처리 속도(throughput)가 계속하여 증가함에 따라, 하나의 컴퓨팅 디바이스로부터 다른 컴퓨팅 디바이스로 데이터를 전송하는 데 드는 비용이 점차적으로 더 적어지게 된다. 결과적으로, 대규모 처리를 수행하기 위해 멀리 떨어져 배치되어 있는 서버 컴퓨팅 디바이스들이 점점 더 이용되고, 이러한 처리로부터 얻어진 데이터는 컴퓨터 네트워크들을 거쳐 이러한 서버 컴퓨팅 디바이스들에 통신 연결되어 있는 로컬 개인용 컴퓨팅 디바이스들을 통해 다시 사용자들에게 전달된다.

[0002] 종래의 서버 컴퓨팅 디바이스들은, 전형적으로, 대량의 이러한 서버 컴퓨팅 디바이스들이 물리적으로 같은 장소에 배치될 수 있게 최적화되어 있다. 예를 들어, 종래의 서버 컴퓨팅 디바이스들은 종종 "블레이드(blade)" 아키텍처를 이용하여 구축되고, 이 경우 서버 컴퓨팅 디바이스의 하드웨어는 물리적으로 콤팩트하고, 다수의 이러한 블레이드들이 "랙(rack)" 아키텍처에 수직으로 배열될 수 있도록 설계된 물리적 하우징 내에 배치된다. 랙 내의 각각의 서버 컴퓨팅 디바이스는 서로 네트워크로 연결될 수 있고, 다수의 이러한 랙들은, 데이터 센터 내와 같이, 물리적으로 같은 장소에 배치될 수 있다. 그러면, 계산 작업이 단일의 데이터 센터 내의 다수의 이러한 서버 컴퓨팅 디바이스들에 걸쳐 분산될 수 있고, 그로써 작업을 보다 효율적으로 완료할 수 있다.

[0003] 계산 작업을 다수의 서버 컴퓨팅 디바이스들에 걸쳐 분산시킬 때, 그 다수의 서버 컴퓨팅 디바이스들 각각은, SAN(Storage Area Network) 또는 다른 유사한 메커니즘들 등을 통해, 다수의 서버 컴퓨팅 디바이스들 중 임의의 것에 의해 똑같이 액세스될 수 있는 디스크 어레이들 또는 다른 유사한 컴퓨터 판독 가능 매체의 집합체의 형태로 구성된 컴퓨터 판독 가능 저장 매체 상에 저장될 수 있는 단일의 데이터 집합에 액세스할 수 있다. 계산 작업이 이어서, 이러한 계산 작업이 수행되는 저장된 데이터의 다수의 사본들을 꼭 만들 필요 없이, 다수의 서버 컴퓨팅 디바이스들에 의해 병렬로 수행될 수 있다.

[0004] 안타깝게도, 각각의 서버 컴퓨팅 디바이스의 처리 유닛들은 계산 작업들을 수행하기 위해 이용할 수 있는 메모리의 양이 제한되어 있다. 보다 구체적으로는, 각각의 서버 컴퓨팅 디바이스의 처리 유닛들은 물리적으로 처리 유닛들과 동일한 서버 컴퓨팅 디바이스 내에 있는 메모리에만 직접 액세스할 수 있다. 가상 메모리 기법들은 전형적으로 주어진 서버 컴퓨팅 디바이스 상에 물리적으로 설치된 것보다 더 많은 양의 메모리에의 액세스를 필요로 하는 계산 작업들의 처리를 가능하게 하기 위해 이용된다. 이러한 가상 메모리 기법들은 메모리에서 디스크로 데이터를 스왑핑(swapping)할 수 있고, 그로써 더 많은 양의 메모리가 있는 것처럼 보이게 한다. 안타깝게도, 메모리에서 디스크로 그리고 다시 거꾸로 데이터를 스왑핑하는 것은 종종 용납할 수 없는 지연들을 유입시킨다. 이러한 지연들은 디스크가 물리적으로 동일한 서버 컴퓨팅 디바이스 상에, 또는 다른 컴퓨팅 디바이스와 같은 원격지에, 또는 SAN의 일부로서 배치되어 있는지에 관계없이 똑같이 존재할 수 있다. 보다 구체적으로는, 이러한 스왑핑을 지원하기 위해 사용되는 저장 매체의 속도를 향상시키는 것은 가상 메모리 기법들의

사용에 의해 유입되는 지연들을 해결하지 않는다.

발명의 내용

[0005] 하나의 실시예에서, 하나의 컴퓨팅 디바이스의 물리적으로 일부인 메모리가 제1 컴퓨팅 디바이스에 통신 연결되어 있는 다른 상이한 컴퓨팅 디바이스 상에서 실행 중인 프로세스들의 프로세스 공간(process space)에 매핑되고 그 프로세스들에 의해 직접 액세스 가능할 수 있다. 그로써, 하나의 컴퓨팅 디바이스의 로컬적으로 주소 지정 가능한 메모리 네임스페이스(locally addressable memory namespace)가 다른 상이한 컴퓨팅 디바이스 상에 물리적으로 있을 수 있는 메모리에 의해 지원된다.

[0006] 다른 실시예에서, 원격 메모리 인터페이스(Remote Memory Interface)(RMI)는 로컬적으로 실행 중인 프로세스들에 메모리 관리 기능을 제공하여, 로컬적으로 주소 지정 가능한 메모리 네임스페이스로 보내지는 커맨드를 로컬적으로 실행 중인 프로세스들로부터 수신하고 이어서 이러한 커맨드를 통신 연결을 통해 원격 컴퓨팅 디바이스 - 이 원격 컴퓨팅 디바이스의 물리 메모리는 로컬적으로 주소 지정 가능한 메모리 네임스페이스의 일부를 지원함 - 로 전송 가능한 형태들로 변환할 수 있다. RMI는 또한 그 RMI에 보내지는 원격 통신을 수신하고 그 통신을 로컬적으로 설치된 메모리로 보내지는 커맨드로 변환할 수 있다.

[0007] 또 다른 실시예에서, 제어기는 다른 컴퓨팅 디바이스들 상에서 실행 중인 프로세스들과 얼마만큼의 메모리 저장 용량을 공유해야 하는지를 결정할 수 있다. 이러한 제어기는 다수의 컴퓨팅 디바이스들 간의 메모리의 공유를 조정할 수 있는 중앙 제어기일 수 있거나, 다수의 컴퓨팅 디바이스들 자체 간의 퍼어-투-퍼어 통신(peer-to-peer communications)의 형태로 구현될 수 있다. 또 다른 대안으로서, 이러한 제어기는 하나의 레벨의 제어기들이 컴퓨팅 디바이스 세트들 사이의 메모리의 공유를 조정하고 다른 레벨의 제어기들이 각각의 개별 컴퓨팅 디바이스 세트 내의 개별 컴퓨팅 디바이스들 사이의 공유를 조정하는 계층적 형식(hierarchical format)으로 구현될 수 있다.

[0008] 추가의 실시예에서, 로컬적으로 실행 중인 프로세스가 원격 컴퓨팅 디바이스 상의 물리 메모리에 의해 지원되는 로컬적으로 주소 지정 가능한 메모리 네임스페이스의 일부분에 액세스하려고 시도하는 경우, 이러한 액세스가 겹출되거나 플래그 설정될 수 있고, 이러한 요청을 생성하는 작업의 실행이, 데이터의 원격 액세스의 완료 때까지, 일시 중단될 수 있다. 이러한 일시 중단(suspension)은 현재의 가상 메모리 동작들보다 엄청나게 더 빠를 수 있는 이러한 원격 메모리 동작들의 효율성에 따라 조정될 수 있다.

[0009] 다른 추가의 실시예에서, 메모리를 공유하는 개별 컴퓨팅 디바이스들의 운영 체제들은 공유되는 이러한 메모리의 저장소(storage)의 양을 조절하는 기능은 물론, 이러한 컴퓨팅 디바이스들 상에서 실행 중인 프로세스들의 프로세스 공간에, 이러한 프로세스들이 실행 중인 컴퓨팅 디바이스로부터 멀리 떨어져 있는 메모리에 의해 지원되는 저장 용량을 매핑하는 기능을 포함할 수 있다.

[0010] 이 발명의 내용은 이하에서 발명을 실시하기 위한 구체적인 내용에 추가로 기술되는 일련의 개념들을 간략화된 형태로 소개하기 위해 제공된다. 이 발명의 내용은 청구된 발명 요지의 주요 특징들 또는 필수적인 특징들을 나타내기 위한 것이 아니며, 청구된 발명 요지의 범주를 제한하기 위해 사용되기 위한 것도 아니다.

[0011] 부가의 특징들 및 장점들이 첨부 도면을 참조하여 계속되는 이하의 상세한 설명으로부터 명백하게 될 것이다.

도면의 간단한 설명

[0012] 이하의 상세한 설명은 첨부 도면과 관련하여 살펴볼 때 가장 잘 이해될 수 있다.

도 1은 예시적인 메모리 공유 환경의 블록도.

도 2는 메모리 공유를 가능하게 하는 예시적인 아키텍처의 블록도.

도 3a 및 도 3b는 예시적인 메모리 공유 메커니즘들의 흐름도.

도 4는 예시적인 범용 컴퓨팅 디바이스를 나타낸 블록도.

발명을 실시하기 위한 구체적인 내용

[0013] 이하의 설명은 네트워크를 통한 메모리 공유에 관한 것이다. 메모리가 네트워크 등을 통해 서로 통신 연결되어 있는 컴퓨팅 디바이스들 간에 공유될 수 있다. 각각의 컴퓨팅 디바이스는 로컬적으로 실행 중인 프로세스들에 메모리 관리 기능을 제공하여, 로컬적으로 실행 중인 프로세스들로부터 로컬적으로 주소 지정 가능한 메모리 네

임스페이스로 보내지는 커맨드를 수신하고 이어서 이러한 커맨드를 원격 컴퓨팅 디바이스로 전송 가능한 형태들로 변환할 수 있는 원격 메모리 인터페이스(RMI)를 포함할 수 있다. RMI는 또한 그 RMI에 보내지는 원격 통신을 수신하고 그 통신을 로컬 메모리로 보내지는 커맨드로 변환할 수 있다. 공유되는 메모리의 양이 중앙 제어기(단일의 제어기 또는 제어기들의 계층적 집합체)에 의해 통보될 수 있거나, 메모리 공유를 수행하는 개별 컴퓨팅 디바이스들 사이의 피어-투-피어 협상(peer-to-peer negotiation)에 의해 통보될 수 있다. 실제로는 원격 메모리 상에 저장되어 있는 데이터에 액세스하라는 요청들이 검출되거나 플래그 설정(flag)될 수 있고, 이러한 요청을 생성하는 작업의 실행이, 원격 메모리 액세스의 효율성에 적절하게, 그 실행이 효율적으로 재시작(resume)될 수 있는 방식으로 일시 중단될 수 있다. 운영 체제는 실제로는 하나 이상의 원격 컴퓨팅 디바이스들의 물리 메모리에 의해 지원되는 용량을 포함하는 로컬적으로 주소 지정 가능한 메모리 네임스페이스를 로컬적으로 실행 중인 애플리케이션들에 제공할 수 있다. 이러한 운영 체제 메커니즘들은 또한 다수의 컴퓨팅 디바이스들 간에 공유 가능한 메모리의 양을 조절할 수 있다.

[0014] 본 명세서에 기술된 기법들은 특정의 유형의 컴퓨팅 자원들의 공유에 대해 언급한다. 상세하게는, 기술된 메커니즘들은 "메모리"의 공유에 관한 것이다. 본 명세서에서 이용되는 바와 같이, "메모리"라는 용어는 로컬적으로 주소 지정 가능한 메모리 네임스페이스를 통해 중앙 처리 유닛 상에서 실행 중인 명령어들에 의해 직접 액세스 가능한 데이터의 저장을 지원하는 임의의 물리 저장 매체를 의미한다. 그 용어로서의 "메모리"의 예는, 본 명세서에서, RAM(Random Access Memory), DRAM(Dynamic RAM), SRAM(Static RAM), T-RAM(Thyristor RAM), Z-RAM(Zero-capacitor RAM) 및 TTRAM(Twin Transistor RAM)(이들로 제한되지 않음)을 포함하는 것으로 정의된다. 예들의 이러한 목록이 제한되지는 않지만, "메모리"라는 용어의 정의를 앞서 제공된 것을 넘어 확장하는 것으로 의도되지도 않는다. 상세하게는, "메모리"라는 용어가, 본 명세서에서 이용되는 바와 같이, 구체적으로는 저장 네임스페이스(storage namespace) 또는 파일 시스템을 통해 액세스 가능한 데이터를 저장하는 저장 매체를 제외한다.

[0015] 꼭 그럴 필요는 없지만, 이하의 설명들의 양태들이 컴퓨팅 디바이스에 의해 실행되는, 프로그램 모듈들과 같은, 컴퓨터 실행 가능 명령어들과 일반적으로 관련하여 제공될 것이다. 보다 구체적으로는, 달리 언급하지 않는 한, 이 설명들의 양태들은 하나 이상의 컴퓨팅 디바이스들 또는 주변 기기들에 의해 수행되는 기능들 및 동작들의 심볼 표현들을 참조할 것이다. 그에 따라, 때때로 컴퓨터에 의해 실행되는 것으로 언급되는 이러한 기능들 및 동작들이, 처리 유닛이 구조적 형태(structured form)로 데이터를 표현하는 전기 신호들을 조작하는 것을 포함한다는 것을 잘 알 것이다. 이 조작은 데이터를 변환하거나 데이터를 메모리 내의 위치들에 유지하고, 이는, 통상의 기술자라면 잘 아는 방식으로, 컴퓨팅 디바이스 또는 주변 기기들의 동작을 재구성하거나 다른 방식으로 변경한다. 데이터가 유지되는 데이터 구조들은 데이터의 형식(format)에 의해 규정되는 특정의 속성(property)들을 가지는 물리적 위치들이다.

[0016] 일반적으로, 프로그램 모듈들은 특정의 작업들을 수행하거나 특정의 추상 데이터 형식(abstract data type)들을 구현하는 루틴들, 프로그램들, 객체들, 구성요소들, 데이터 구조들 등을 포함한다. 게다가, 통상의 기술자라면 컴퓨팅 디바이스들이 종래의 서버 컴퓨팅 랙(server computing rack) 또는 종래의 개인용 컴퓨터로 제한될 필요가 없고, 핸드헬드 디바이스, 멀티프로세서 시스템, 마이크로프로세서 기반 또는 프로그램 가능 가전 제품, 네트워크 PC, 미니컴퓨터, 메인프레임 컴퓨터 등을 비롯한 다른 컴퓨팅 구성들을 포함한다는 것을 잘 알 것이다. 이와 유사하게, 메커니즘들이 또한 통신 네트워크를 통해 연결되어 있는 분산 컴퓨팅 환경들에서도 실시될 수 있기 때문에, 컴퓨팅 디바이스들이 독립형 컴퓨팅 디바이스(stand-alone computing device)로 제한될 필요는 없다. 분산 컴퓨팅 환경에서는, 프로그램 모듈들이 로컬 및 원격 저장 디바이스들 둘 다에 배치될 수 있다.

[0017] 도 1을 참조하면, 컴퓨팅 디바이스들의 네트워크(190)를 포함하는 예시적인 시스템(100)이 예시되어 있다. 이하의 설명을 위한 예시적인 기초를 제공하기 위해, 서버 컴퓨팅 디바이스들(110, 120 및 130)의 형태로 되어 있는 3 개의 서버 컴퓨팅 디바이스들이 네트워크(190)를 통해 서로 통신 연결되어 있는 것으로 예시되어 있다. 서버 컴퓨팅 디바이스들(110, 120 및 130) 각각은 컴퓨터 실행 가능 명령어들을 실행할 수 있는 처리 유닛들을 포함할 수 있다. 이러한 컴퓨터 실행 가능 명령어들의 실행에서, 데이터가 처리 유닛들에 의해 메모리에 저장될 수 있다. 실행 중인 컴퓨터 실행 가능 명령어들에 따라, 메모리에 저장되도록 요청되는 데이터의 양이 서버 컴퓨팅 디바이스 상에 설치된 물리 메모리의 저장 용량보다 더 클 수 있다. 이러한 경우에, 전형적으로, 어떤 데이터가 메모리로부터 보다 느린 비휘발성 저장 매체(하드 디스크 드라이브 등)로 "스왑핑"(swappping)되는 가상 메모리 메커니즘들이 이용된다. 이러한 방식으로, 더 많은 메모리 용량이 이용 가능하게 된다. 메모리로부터 디스크로 스왑핑된 데이터가 실행 중인 프로세스에 의해 메모리로부터 읽혀지려고 시도될 때, 페이지 폴트(page fault)가 발생될 수 있고, 이러한 데이터가 보다 느린 디스크로부터 다시 읽혀지고 메모리에 재차 저장되는 동안 - 데이터는 이

어서 메모리로부터 처리 유닛들에 제공될 수 있음 - 처리가 일시적으로 중단될 수 있다. 통상의 기술자라면 잘 알겠지만, 이러한 프로세스가, 특히 서버 컴퓨팅과 관련하여, 바람직하지 않을 수 있는 지연들을 유입시킬 수 있다.

[0018] 도 1의 예시적인 시스템(100)은 서버 컴퓨팅 디바이스(130)가 그 서버 컴퓨팅 디바이스(130)에 처리 기능을 수행할 것을 요구할 수 있는 작업(job)(140)을 할당받은 일 실시예를 나타낸 것이다. 보다 구체적으로는, 서버 컴퓨팅 디바이스(130)의 중앙 처리 유닛(CPU)들(132)과 같은 하나 이상의 처리 유닛들은 작업(140)과 연관된 컴퓨터 실행 가능 명령어들을 실행할 수 있다. 전형적으로, 작업(140)과 연관된 컴퓨터 실행 가능 명령어들의 실행은 데이터를 서버 컴퓨팅 디바이스(130)의 메모리(135)와 같은 메모리에 저장하는 것을 필요로 할 수 있다. 이하의 설명을 위해, 작업(140)과 연관된 컴퓨터 실행 가능 명령어들이 필요로 할 수 있는 메모리의 양이 메모리(135)의 양을 초과할 수 있거나, 보다 정확하게는, 작업(140)의 처리에 할당될 수 있는 메모리(135)의 메모리 저장 용량을 초과할 수 있다.

[0019] 도 1에 예시된 실시예에서, 작업(140)과 연관된 컴퓨터 실행 가능 명령어들을 실행하고 있을 수 있는 서버 컴퓨팅 디바이스(130)의 CPU(132)는, 메모리(135)에 데이터를 저장하고 그로부터 데이터를 검색하기 위해, 하나 이상의 메모리 관리 유닛(memory management unit)(MMU)들[MMU(133) 등]과 통신할 수 있다. 통상의 기술자라면 잘 알겠지만, 전형적으로, STORE 명령어는 데이터를 메모리(135)에 저장하기 위해 이용될 수 있고, LOAD 명령어는 메모리(135)로부터 데이터를 읽고 이 데이터를 CPU(132)의 하나 이상의 레지스터들에 로드하기 위해 이용될 수 있다. 개별적으로 예시되어 있지만, MMU(133)는 종종 CPU(132)와 일체로 되어 있는 부분이다. 앞서 언급한 바와 같이, 작업(140)과 연관된 컴퓨터 실행 가능 명령어들을 실행할 때, CPU(132)가 작업(140)에 할당된 메모리 용량을 넘어 부가의 데이터를 메모리에 저장하려고 시도할 때, 하나의 실시예에서, 이러한 부가의 메모리 용량이 원격 메모리 인터페이스(RMI)(131)의 기능을 통해 로컬적으로 주소 지정 가능한 메모리 네임스페이스의 일부로서 이용 가능하게 될 수 있다. 보다 구체적으로는, 컴퓨터 실행 가능 명령어들에 의한 로컬적으로 주소 지정 가능한 메모리 네임스페이스의 일부분에 액세스하려는 시도들은 이러한 액세스가 RMI(131)로 보내지게 할 수 있고, RMI(131)는 이러한 액세스를 네트워크 통신으로 변환할 수 있으며, 예를 들어, 서버 컴퓨팅 디바이스들(110 및 120) 중 하나와 같은 다른 상이한 컴퓨팅 디바이스와 통신할 수 있고, 그로써, 컴퓨팅 디바이스(130) 상에서 실행 중인 프로세스들을 위해서 이러한 다른 컴퓨팅 디바이스들 상에 설치된 물리 메모리를 이용할 수 있다.

[0020] 이와 같이, 하나의 실시예에서, 원격 메모리 인터페이스(131)는, CPU(132) 및 그에 의해 실행되고 있는 컴퓨터 실행 가능 명령어들의 관점에서 볼 때, MMU(133)와 같은 메모리 관리 유닛으로서 역할할 수 있다. 예를 들어, 메모리 페이지 테이블(memory page table), 또는 다른 유사한 메모리 인터페이스 메커니즘은 원격 메모리 인터페이스(131)와 연관될 수 있는 로컬적으로 주소 지정 가능한 메모리 네임스페이스의 특정의 부분들(특정의 페이지들 또는 특정의 주소 범위들 등)을 식별할 수 있다. 로컬적으로 주소 지정 가능한 메모리 네임스페이스의 그 부분들로 보내지는 LOAD 또는 STORE 명령어들 또는 다른 유사한 명령어가 원격 메모리 인터페이스(131)로 보내질 수 있다. 이와 같이, 서버 컴퓨팅 디바이스(130)에 의해 실행되고 있는 프로세스들에 의해 이용 가능한 것과 같은 로컬적으로 주소 지정 가능한 메모리 네임스페이스는 물리 메모리(135)보다 더 클 수 있는데, 그 이유는 원격 메모리 인터페이스(131)가 컴퓨팅 디바이스(130) 상에서의 증가된 메모리 네임스페이스를 지원하기 위해 원격 컴퓨팅 디바이스들의 메모리[예를 들어, 서버 컴퓨팅 디바이스(120)의 메모리(125) 또는 서버 컴퓨팅 디바이스(110)의 메모리(115) 등]를 이용할 수 있기 때문이다.

[0021] 이러한 실시예에서, 컴퓨팅 디바이스(130) 상에서, 원격 메모리 인터페이스(131)에 의해 지원되는 로컬 메모리 네임스페이스의 일부분으로 보내지는 커맨드를 수신할 때, 원격 메모리 인터페이스(131)는 커맨드를 네트워크(190)를 통해 하나 이상의 다른 컴퓨팅 디바이스들[서버 컴퓨팅 디바이스들(110 및 120) 등]로 전달될 수 있는 형식으로 변환할 수 있다. 예를 들어, 원격 메모리 인터페이스(131)는 네트워크(190)에 의해 이용되는 네트워크 프로토콜들에 의해 규정되는 패킷 구조에 따라 커맨드를 패킷화(packetize)할 수 있다. 다른 예로서, 원격 메모리 인터페이스(131)는 통신을 특정의 컴퓨팅 디바이스들의 원격 메모리 인터페이스들[예를 들어, 서버 컴퓨팅 디바이스(120)의 원격 메모리 인터페이스(121) 또는 서버 컴퓨팅 디바이스(110)의 원격 메모리 인터페이스(111) 등]로 보내기 위해, 적절한 네트워크 주소 지정 정보(network addressing information) 및 다른 유사한 라우팅 정보[네트워크(190)에 의해 사용되는 프로토콜들에 의해 결정됨]를 생성할 수 있다.

[0022] 그 후에, 그 다른 컴퓨팅 디바이스들 상의 원격 메모리 인터페이스들은, 그들로 보내지는 네트워크 통신을 수신할 때, 그 네트워크 통신을 적절한 메모리 중심 커맨드(memory-centric command)로 변환하고 이러한 네트워크 통신을 수신하는 원격 메모리 인터페이스와 동일한 컴퓨팅 디바이스 상에 물리적으로 존재하는 메모리에 대해

이러한 커멘드를 수행할 수 있다. 예를 들어, 원격 메모리 인터페이스(131)로부터 통신을 수신할 때, 서버 컴퓨팅 디바이스(120) 상의 원격 메모리 인터페이스(121)는 메모리(125)의 부분(126)과 관련하여 동작을 수행할 수 있다. 메모리(125)의 부분(126)은, 이하에서 더 상세히 기술될 방식 등에 의해, 다른 컴퓨팅 디바이스들과 공유되기 위해 따로 확보해 둘 수 있는 부분일 수 있다. 유사한 방식으로, 서버 컴퓨팅 디바이스(110) 상의 원격 메모리 인터페이스(111)는, 서버 컴퓨팅 디바이스(130)의 원격 메모리 인터페이스(131)로부터 네트워크(190)를 통해 통신을 수신한 것에 응답하여, 서버 컴퓨팅 디바이스(110)의 메모리(115)의 부분(116)과 관련하여 동작을 수행할 수 있다. 상이한 물리적 부분으로서 예시되어 있지만, 메모리(115)의 부분(116) 및 메모리(125)의 부분(126)은 메모리(115 및 125)의 메모리 저장 용량의 일부가 그 컴퓨팅 디바이스들의 원격 메모리 인터페이스들 [즉, 원격 메모리 인터페이스들(각각, 111 및 121)]에 의한 이용을 위해 예비되어 있다는 것을 그래픽적으로 나타내기 위한 것에 불과하다. 통상의 기술자라면 잘 알겠지만, 로컬적으로 주소 지정 가능한 메모리 네임스페이스를 지원하는 메모리(115 및 125)의 실제의 물리적 데이터 저장 유닛들(트랜지스터들 등)과 원격 메모리 인터페이스들(각각, 111 및 121)에 의한 이용을 위해 예비되어 있는 메모리 저장 용량을 제공하는 그 저장 유닛들 사이에 명확한 구분이 존재할 필요는 없다.

[0023] 추가의 설명을 제공하기 위해, 예를 들어, 작업(140)을 실행할 때, 서버 컴퓨팅 디바이스(130)의 CPU(132)가 데이터를 원격 메모리 인터페이스(131)에 의해 지원되는 로컬적으로 주소 지정 가능한 메모리 네임스페이스의 일부분에 저장하려고 시도하는 경우, 메모리 관리 유닛(133)과 달리, 이러한 요청이 원격 메모리 인터페이스(131)로 보내질 수 있고, 원격 메모리 인터페이스(131)는 이어서 요청을 서버 컴퓨팅 디바이스(120) 상의 원격 메모리 인터페이스(121) 및 서버 컴퓨팅 디바이스(110) 상의 원격 메모리 인터페이스(111)로 보내질 수 있는 네트워크 통신으로 변환할 수 있다. 이러한 네트워크 통신을 수신할 때, 원격 메모리 인터페이스(121)는 네트워크 통신을 데이터를 저장하라는 요청으로 변환할 수 있고, 이어서 서버 컴퓨팅 디바이스(120)의 메모리(125)의 부분(126)에 데이터를 저장하는 것을 수행할 수 있다. 이와 유사하게, 이러한 네트워크 통신을 수신할 때, 원격 메모리 인터페이스(111)도 네트워크 통신을 데이터를 저장하라는 요청으로 변환할 수 있고, 서버 컴퓨팅 디바이스(110)의 메모리(115)의 부분(116)에 그 데이터를 저장하는 것을 수행할 수 있다. 이러한 방식으로, 서버 컴퓨팅 디바이스(130) 상에서 실행 중인 프로세스들에 의해 주소 지정 가능한 메모리 네임스페이스가 서버 컴퓨팅 디바이스(130) 상에 존재하는 메모리(135)보다 더 클 수 있다. 보다 구체적으로는 그리고 이하에서 더 상세히 기술하는 바와 같이, 다른 컴퓨팅 디바이스들로부터의 공유 메모리[서버 컴퓨팅 디바이스(120)의 메모리(125)의 부분(126) 및 서버 컴퓨팅 디바이스(110)의 메모리(115)의 부분(116) 등]가 서버 컴퓨팅 디바이스(130) 상에서 실행 중인 프로세스들[예를 들어, 작업(140)과 연관된 프로세스들]에 의해 주소 지정 가능한 메모리 네임스페이스를 지원할 수 있고, 그로써 이러한 프로세스들은 이러한 프로세스들이 실행 중인 서버 컴퓨팅 디바이스(130) 상의 메모리(135)로부터 이용 가능한 것보다 더 많은 메모리를 이용할 수 있게 된다.

[0024] 하나의 실시예에서, 공유 가능하게 되는 메모리 저장 용량의 양은 메모리 공유 제어기(170)와 같은 중앙 메커니즘에 의해 조정될 수 있다. 예를 들어, 메모리 공유 제어기(170)는 예시적인 서버 컴퓨팅 디바이스들(110, 120 및 130)과 같은 컴퓨팅 디바이스들로부터 정보를 수신할 수 있고, 이러한 수신된 정보에 기초하여, 공유 가능하게 되어야 하는 서버 컴퓨팅 디바이스들(110, 120 및 130)의 메모리 저장 용량의 양을 결정할 수 있다. 예를 들어, 메모리 공유 제어기(170)는 메모리(125)의 부분(126)을 공유 가능하게 만들라고 서버 컴퓨팅 디바이스(120)에 지시할 수 있다. 유사한 방식으로, 메모리 공유 제어기(170)는 메모리(115)의 부분(116)을 공유 가능하게 만들라고 서버 컴퓨팅 디바이스(110)에 지시할 수 있다. 그에 응답하여, 서버 컴퓨팅 디바이스들(110 및 120)의 운영 체제 또는 다른 유사한 메모리 제어기 프로세스들은 메모리(115)의 부분(116) 및 메모리(125)의 부분(126)을 따로 확보해 두고, 그 부분들을 그 서버 컴퓨팅 디바이스들 상에서 로컬적으로 실행 중인 프로세스들을 위해 이용하지 않을 수 있다. 보다 구체적으로는, 원격 메모리 인터페이스에 의한 이용을 위해 예비되어 있고 그로써 원격 컴퓨팅 디바이스들 상에서 실행 중인 프로세스들과 공유되는 메모리 저장 용량과 로컬적으로 주소 지정 가능한 메모리 네임스페이스를 구분하기 위해, 메모리의 특정의 페이지들, 메모리의 특정의 주소들, 또는 다른 유사한 식별자들이 이용될 수 있다. 따라서, 예를 들어, 서버 컴퓨팅 디바이스(120) 상에서 실행 중인 프로세스들에 의해 이용될 수 있는 로컬적으로 주소 지정 가능한 메모리 네임스페이스는 공유를 위해 따로 확보해 둔 부분(126)을 제외한 메모리(125)의 부분에 의해 지원될 수 있다. 유사한 방식으로, 서버 컴퓨팅 디바이스(110) 상에서 실행 중인 프로세스들에 의해 이용될 수 있는 로컬적으로 주소 지정 가능한 메모리 네임스페이스는 부분(116)을 제외한 메모리(115)의 부분에 의해 지원될 수 있다.

[0025] 예시적인 서버 컴퓨팅 디바이스들(110, 120 및 130)과 같은 컴퓨팅 디바이스들로부터 메모리 공유 제어기(170)에 의해 수신되는 정보는 이러한 컴퓨팅 디바이스들 상에 물리적으로 설치된 또는 이러한 컴퓨팅 디바이스들 다른 방식으로 이용 가능한 메모리의 총량, 현재 이용되고 있는 메모리 저장 용량의 양, 원하는 메모리 저장 용

량의 양, 및 다른 유사한 정보를 명시하는 정보를 포함할 수 있다. 이러한 정보에 기초하여, 메모리 공유 제어기(170)는 서버 컴퓨팅 디바이스들(110, 120 및 130) 각각에 의해 공유 가능하게 될 메모리 저장 용량의 양을 식별할 수 있다. 하나의 실시예에서, 메모리 공유 제어기(170)는 그에 따라 서버 컴퓨팅 디바이스들(110, 120 및 130)에 지시할 수 있는 반면, 다른 실시예들에서, 메모리 공유 제어기(170)는 개별 컴퓨팅 디바이스들의 운영 체제들 또는 다른 유사한 제어 메커니즘들이 수락하거나 무시할 수 있는 요청들을 발행하기만 할 수 있다.

[0026] 하나의 실시예에서, 메모리 공유 제어기(170)는 공유되는 메모리 저장 용량의 양을 계속하여 조절할 수 있다. 이러한 실시예에서, 개별 컴퓨팅 디바이스들의 운영 체제들 또는 다른 유사한 제어 메커니즘들은 로컬적으로 주소 지정 가능한 메모리 네임스페이스의 크기가 런타임 동안 동적으로 변경될 수 있는 메커니즘들을 포함할 수 있다. 예를 들어, 서버 컴퓨팅 디바이스(130)에 의한 작업(140)의 실행의 결과, 메모리에 대한 요구가 증가할 수 있다. 그에 응답하여, 서버 컴퓨팅 디바이스(130) 상에서 실행 중인 프로세스들은 메모리 공유 제어기(170)와 통신할 수 있고, 부가의 공유 메모리를 요청할 수 있다. 메모리 공유 제어기(170)는 이어서, 일례로서, 서버 컴퓨팅 디바이스(110)가 공유 가능하게 해놓은 메모리(115)의 부분(116)을 증가시키라고 서버 컴퓨팅 디바이스(110)에 요청할 수 있다. 그에 응답하여, 하나의 실시예에서, 서버 컴퓨팅 디바이스(110) 상에서 실행 중인 운영 체제 또는 다른 유사한 제어 메커니즘들은 서버 컴퓨팅 디바이스(110) 상에서 로컬적으로 실행 중인 프로세스들에 이전에 할당되었던 메모리(115)의 그 부분들에 저장된 데이터를 스왑핑하여, 이러한 데이터를, 예를 들어, 하드 디스크 드라이브 상에 저장할 수 있다. 그 후에, 메모리(115)의 그 부분들이 공유 가능한 부분(116)에 추가됨으로써 공유 가능한 부분(116)을 증가시켜 서버 컴퓨팅 디바이스(130) 상에서의 작업(140)의 실행의 증가된 요구사항들에 대응할 수 있다. 서버 컴퓨팅 디바이스(110) 상에서 로컬적으로 실행 중인 프로세스들이 이어서 이러한 프로세스들에 이전에 할당되었다가 그 후에 공유되고 있는 부분(116)에 재할당된 메모리(115)의 그 부분들에 액세스하려고 시도하는 경우, 페이지 폴트가 발생할 수 있고, 메모리(115)의 다른 부분들로부터의 어떤 다른 데이터를 디스크로 이동시킴으로써 이전에 디스크로 스왑 아웃(swap out)된 데이터를 다시 메모리(115)로 스왑인(swap in)할 공간을 만들기 위해 가상 메모리 메커니즘들이 이용될 수 있다

[0027] 다른 실시예에서, 메모리 공유 제어기(170)는 공유에 전용되어 있는 개별 컴퓨팅 디바이스들의 메모리의 부분들의 조절에 있어서 제한되어 있을 수 있다. 예를 들어, 메모리 공유 제어기(170)는, 예를 들어, 그 컴퓨팅 디바이스가 제시작하고 있는 동안, 또는 그 컴퓨팅 디바이스가 다른 작업들의 실행을 일시 중단한 기간 동안과 같은, 규정된 기간 동안만 임의의 특정의 컴퓨팅 디바이스에 의해 공유되는 메모리의 양을 조절할 수 있다.

[0028] 메모리 공유 제어기(170)가 단일의 디바이스로서 나타내어져 있지만, 계층적 접근법이 또한 이용될 수 있다. 예를 들어, 메모리 공유 제어기(170)는 데이터 센터에서 흔히 발견되는 것과 같이, 단일의 서버 컴퓨팅 디바이스 랙(rack of server computing devices) 내와 같은 물리적으로 같은 장소에 배치되어 있는 서버 컴퓨팅 디바이스들[예시적인 서버 컴퓨팅 디바이스들(110, 120 및 130) 등]에 앞서 기술된 공유 메모리 제어를 제공하는 데 전용될 수 있다. 다른 상이한 메모리 공유 제어기는 이어서, 예를 들어, 데이터 센터에서의 다른 랙의 서버 컴퓨팅 디바이스들 사이와 같이, 다른 세트의 컴퓨팅 디바이스들 사이의 공유 메모리 제어를 제공하는 데 전용될 수 있다. 상위 레벨 메모리 공유 제어기는 이어서 특정의 서버 컴퓨팅 디바이스 랙들에 할당된 개별 메모리 공유 제어기들을 제어할 수 있다. 예를 들어, 랙 레벨 메모리 공유 제어기들은 단일의 랙 내의 서버 컴퓨팅 디바이스들 사이의 메모리 공유를 제어할 수 있는 반면, 데이터 센터 레벨 메모리 공유 제어기는 서버 컴퓨팅 디바이스 랙들 사이의 메모리 공유를 제어할 수 있고, 개별 서버 컴퓨팅 레벨에서의 이러한 공유의 구현은 랙 레벨 메모리 공유 제어기들에 맡긴다.

[0029] 또 다른 실시예에서, 메모리 공유 제어기(170)는 별개의 프로세스 또는 디바이스일 필요가 없고, 오히려 자신의 메모리를 공유하는 컴퓨팅 디바이스들[예를 들어, 서버 컴퓨팅 디바이스들(110, 120 및 130) 등] 사이의 피어-투-피어 통신을 통해 구현될 수 있다. 보다 구체적으로는, 서버 컴퓨팅 디바이스들(110, 120 및 130) 각각에서 개별적으로 실행 중인 프로세스들이 서로 통신할 수 있고, 공유되어야 하는 서버 컴퓨팅 디바이스들(110, 120 및 130) 각각의 메모리(각각, 115, 125 및 135)의 양을 협상할 수 있다. 운영 체제의 메모리 관리 기능과 연관된 프로세스들과 같은, 이러한 로컬적으로 실행 중인 프로세스들은 이어서 다른 프로세스들에 합의되고 협상된 공유를 구현하라고 지시할 수 있다.

[0030] 도 2를 참조하면, 도 2에 도시된 시스템(200)은 원격 메모리 인터페이스들의 예시적인 동작을 더 상세히 설명하는 예시적인 일련의 통신들을 나타내고 있다. 설명을 위해, 서버 컴퓨팅 디바이스(130)의 처리 유닛(132)이 작업(140)과 연관된 컴퓨터 실행 가능 커맨드를 실행하고 있는 것으로 도시되어 있다. 이러한 컴퓨터 실행 가능 커맨드의 실행의 일부로서, CPU(132)는 서버 컴퓨팅 디바이스(130)에 물리적으로 설치되어 있는 메모리(135)로 나타낸 것과 같은 메모리에 데이터를 저장하거나 그 메모리로부터 데이터를 검색하려고 시도할 수 있다. 도 2의

시스템(200)에서, 통상의 기술자라면 잘 알겠지만, 컴퓨팅 디바이스(130)의 CPU(132) 상에서 실행 중인 프로세스들에 의해 직접 액세스될 수 있는 메모리를 포함하는, 로컬적으로 주소 지정 가능한 메모리 네임스페이스(231)가 도시되어 있다. 하나의 실시예에서, 그리고 이하에서 더 상세히 기술하는 바와 같이, 로컬적으로 주소 지정 가능한 메모리 네임스페이스(231)는 로컬적으로 설치된 메모리(135)에 의해 지원될 수 있는 부분(234)과 원격 메모리 인터페이스(131)에 의해 지원될 수 있는 부분, 그리고 또한 원격 메모리를 포함할 수 있다. 예를 들어, 컴퓨팅 디바이스(130)가 16 GB의 로컬적으로 설치된 메모리를 포함하는 경우, 로컬적으로 주소 지정 가능한 메모리 네임스페이스(231)의 부분(234)도 약 16 GB일 수 있다. 이와 유사하게, 로컬적으로 주소 지정 가능한 메모리 네임스페이스(231)의 부분(235)이 4 GB인 경우, 본 명세서에 기술된 메커니즘들을 통해 그것을 지원할 수 있는 4 GB의 공유 메모리가 원격 컴퓨팅 디바이스 상에 있을 수 있다.

[0031] 데이터를 메모리에 저장하기 위해, CPU(132)는 공지된 STORE 커맨드와 같은 적절한 커맨드를 발행할 수 있고, 이 커맨드는 하나 이상의 메모리 관리 유닛들(133)에 의해 수신될 수 있으며, 메모리 관리 유닛들(133)은 차례로 메모리(135)와 인터페이스하여 CPU(132)에 의해 제공된 데이터를 물리 메모리(135)에서의 적절한 저장 위치들, 주소들, 페이지들, 또는 다른 유사한 저장 단위(storage unit)들에 저장할 수 있다. 이와 유사하게, 고속 회발성 저장 매체로부터 데이터를 검색하기 위해, CPU(132)는 공지된 LOAD 커맨드와 같은 다른 적절한 커맨드를 발행할 수 있고, 이 커맨드는 MMU(133)에 의해 수신될 수 있으며, MMU(133)는 차례로 도 1에 도시된 물리 메모리(135)와 인터페이스하여, CPU에 의해 요청된 데이터를 적절한 저장 위치들로부터 검색하고 이러한 데이터를, 작업(140)과 연관된 컴퓨터 실행 가능 명령어들의 실행의 일부로서 CPU(132)에 의한 추가적인 사용(consumption)을 위해, CPU(132)의 레지스터들에 로드할 수 있다. CPU(132)에 의해 발행된 STORE 및 LOAD 커맨드가 로컬적으로 설치된 메모리(135)에 의해 지원되는 로컬적으로 주소 지정 가능한 메모리 네임스페이스(231)의 부분(234)으로 보내지는 경우, 도 2의 시스템(200)에 통신(221 및 222)으로 그래픽으로 나타낸 바와 같이, 이러한 STORE 및 LOAD 커맨드, 그리고 그 결과 메모리(135)에 대해 행해지는 동작들이 메모리 관리 유닛(233)에 의해 관리될 수 있다.

[0032] 하나의 실시예에서, 로컬적으로 주소 지정 가능한 메모리 네임스페이스(231)가 컴퓨팅 디바이스(130) 상에 설치된 메모리보다 더 클 수 있다. 이러한 실시예에서, 예시적인 작업(140)과 같은 컴퓨팅 디바이스(130) 상에서 실행 중인 프로세스들은 더 큰 로컬적으로 주소 지정 가능한 메모리 네임스페이스(231)를 직접 주소 지정할 수 있고, 그 네임스페이스의 부분들을 그 프로세스들의 프로세스 공간에 매핑시킬 수 있다. 이러한 더 큰 로컬적으로 주소 지정 가능한 메모리 네임스페이스(231)가 서버 컴퓨팅 디바이스(130) 상에 설치되어 있는 메모리(135)에 의해서뿐만 아니라, 다른 상이한 컴퓨팅 디바이스 상에 물리적으로 설치되어 있는 원격 메모리[메모리(125) 등]에 의해서도 지원될 수 있다. 그렇지만, 컴퓨팅 디바이스(130) 상에서 실행 중인 프로세스들은 로컬적으로 주소 지정 가능한 메모리 네임스페이스(231)가 실제로 어느 물리 메모리를 나타내고 있는지에 관해서는 모를 수 있다.

[0033] 예를 들어, 작업(140)과 연관된 컴퓨터 실행 가능 명령어들은, CPU(132) 상에서 실행 중인 동안, 데이터를 로컬적으로 주소 지정 가능한 메모리 네임스페이스(231)의 부분(235) - 언급된 바와 같이, 이러한 컴퓨터 실행 가능 명령어들은 부분(235)을 로컬적으로 주소 지정 가능한 메모리 네임스페이스(231)의 임의의 다른 부분과 동일한 방식으로 인식하게 될 것임 - 에 저장하려고 시도할 수 있다. CPU(132)는 이어서 부분(235)의 일부인 로컬적으로 주소 지정 가능한 메모리 네임스페이스(231)의 어떤 부분을 식별하는 주소, 페이지, 또는 다른 유사한 위치 식별자를 지정하는 적절한 커맨드(앞서 기술된 STORE 커맨드 등)를 발행할 수 있을 것이다. 이러한 커맨드는, MMU(133)로 보내지기보다는, 그 대신에, 원격 메모리 인터페이스(131)로 보내질 수 있다. 예를 들어, CPU(132)에 의해 발행된 메모리 중심 커맨드에 의해 지정되는 위치 식별자가, 부분(234) 대신에, 부분(235)의 일부인 것으로 결정하기 위해 TLB(Translation Lookaside Buffer) 또는 다른 유사한 테이블 또는 데이터베이스가 참조될 수 있을 것이고, 결과적으로, 이러한 커맨드가 원격 메모리 인터페이스(131)로 보내질 수 있다. 도 2의 예시적인 시스템(200)에서, 이러한 커맨드가 CPU(132)로부터 원격 메모리 인터페이스(131)로의 통신(223)에 의해 예시되어 있다.

[0034] 이러한 커맨드를 수신할 때, 원격 메모리 인터페이스(131)는 이러한 커맨드를 네트워크 통신(241)과 같은 네트워크 통신으로 변환하고, 그 네트워크 통신을 하나 이상의 다른 컴퓨팅 디바이스들 상의 원격 메모리 인터페이스[예를 들어, 서버 컴퓨팅 디바이스(120)의 원격 메모리 인터페이스(121) 등]로 보낼 수 있다. 커맨드(223)를 네트워크 통신(241)으로 변환할 때, 원격 메모리 인터페이스(131)는 커맨드를 패킷화할 수 있거나, 네트워크(190)를 구현하는 데 이용되는 프로토콜에 적절한 네트워크 통신을 다른 방식으로 생성할 수 있다. 예를 들어, 네트워크(190)가 이더넷 하드웨어를 이용하여 구현되는 경우, 원격 메모리 인터페이스(131)는 이더넷에 대한 최

대 전송 단위를 초과하지 않는 단위들을 갖는 네트워크 통신을 생성할 수 있다. 이와 유사하게, 네트워크(190)가 TCP/IP(Transmission Control Protocol/Internet Protocol)를 이용하여 구현되는 경우, 원격 메모리 인터페이스(131)는 원격 메모리 인터페이스(121)의 IP 주소를 패킷의 목적지로서 지정할 수 있는, TCP/IP 헤더들을 가지는 패킷들을 생성할 수 있다. 네트워크(190)를 구현하는 데 이용되는 프로토콜들에 따라 다른 유사한 변환들이 수행될 수 있다.

[0035] 네트워크 통신(241)이 그 보내지는 서버 컴퓨팅 디바이스(120)의 원격 메모리 인터페이스(121)에 의해 수신되면, 원격 메모리 인터페이스(121)는 이러한 네트워크 통신(241)을, 서버 컴퓨팅 디바이스(120) 상에 설치된 메모리(125)로 보내질 수 있는 적절한 메모리 중심 커맨드(251)로 변환할 수 있다. 보다 구체적으로는, 원격 메모리 인터페이스(121)는 네트워크 통신(241)을 역패킷화(unpacketize)할 수 있고, 공유 가능 메모리로서 따로 확보해 두었고, 그 결과, 예를 들어, 컴퓨팅 디바이스(120)의 메모리 관리 유닛들이 아니라, 원격 메모리 인터페이스(121)의 제어 하에 있을 수 있으며, 그에 따라, 컴퓨팅 디바이스(120) 상에서 실행 중인 프로세스들이 이용할 수 있게 되어 있는 로컬적으로 주소 지정 가능한 메모리 네임스페이스로부터 제외될 수 있는 메모리(125)의 부분(126)에서의 하나 이상의 주소들에 적절한 메모리 중심 커맨드(251)를 생성할 수 있다.

[0036] 메모리 중심 커맨드(251)에 응답하여, 원격 메모리 인터페이스(121)는, 커맨드(251)가 STORE 커맨드인 경우, 확인 응답(acknowledgment)을 수신할 수 있거나, 커맨드가 LOAD 커맨드인 경우, 요청된 데이터를 수신할 수 있다. 이러한 응답이 도 2의 시스템(200)에서 메모리(125)의 부분(126)으로부터 원격 메모리 인터페이스(121)로의 통신(252)으로 예시되어 있다. 응답 통신(252)을 수신할 때, 원격 메모리 인터페이스(121)는 통신(252)을 네트워크 통신(242)으로 변환할 수 있고, 네트워크 통신(242)을, 통신(241)을 송신했던 원격 메모리 인터페이스(131)로 보낼 수 있다. 원격 메모리 인터페이스(131)를 참조하여 앞서 상세히 기술된 바와 같이, 원격 메모리 인터페이스(121)는, 통신(252)을 네트워크 통신(242)으로 변환할 때, 네트워크(190)를 구현하는 데 이용되는 프로토콜들에 따라 통신(252)을 네트워크 통신(242)으로 패킷화, 패키지화(package), 형식 설정(format), 또는 다른 방식으로 변환할 수 있다.

[0037] 원격 메모리 인터페이스(131)는, 네트워크 통신(242)을 수신할 때, 그 통신을 역패킷화할 수 있고 CPU(132)에 대해 적절한 응답[통신(225)으로 예시되어 있음]을 생성할 수 있다. 보다 구체적으로는, CPU(132)로부터의 통신(223)이 STORE 커맨드인 경우, 통신(225)이 데이터가 적절히 저장되었다는 확인 응답일 수 있지만, 본 예에서, 이러한 확인 응답은 데이터가 실제로는 서버 컴퓨팅 디바이스(120)의 메모리(125)의 부분(126)에 적절히 저장되었다는 것이다. 이와 유사하게, CPU(132)로부터의 통신(223)이 LOAD 커맨드인 경우, 통신(225)은 CPU(132)가 자신의 레지스터들 중 하나 이상에 로드되도록 요청한 데이터 - 즉, 본 예에서, 메모리(125)의 부분(126)으로부터 읽혀진 데이터 - 일 수 있다.

[0038] 이러한 방식으로, 서버 컴퓨팅 디바이스(130) 상에서 실행 중인 프로세스들이, 자신들은 알지 못한 채로 그리고 이러한 프로세스들 자체에 어떤 수정도 하지 않고, 예를 들어, 컴퓨팅 디바이스(120)의 메모리(125)와 같은 다른 컴퓨팅 디바이스들 상에 설치된 메모리를 이용할 수 있다. 보다 구체적으로는, 원격 메모리 인터페이스(131)는 서버 컴퓨팅 디바이스(130) 상에서 실행 중인 프로세스들에 의해 직접 주소 지정될 수 있는 로컬적으로 주소 지정 가능한 메모리 네임스페이스(231)의 일부인 것처럼 보이는 메모리와 통신하는 메모리 관리 유닛으로서 역할한다.

[0039] 커맨드(223)의 수신과 응답(225)의 제공 사이의 지연 시간을 감소시키기 위해, 하나의 실시예에서, 원격 메모리 인터페이스(131)는 서버 컴퓨팅 디바이스(130)의 일부인 네트워킹 하드웨어와 직접 통신할 수 있다. 예를 들어, 원격 메모리 인터페이스(131)는 서버 컴퓨팅 디바이스(130)의 네트워크 인터페이스에의 직접 연결(direct connection)을 포함하는 전용 프로세서일 수 있다. 이러한 전용 프로세서는 독립형 프로세서일 수 있거나 하나 이상의 CPU들과 같은 다른 프로세스들과 통합되어 있을 수 있는, 공지된 메모리 관리 유닛 프로세서(MMU)들과 유사할 수 있다. 원격 메모리 인터페이스(121)는 또한 서버 컴퓨팅 디바이스(120)의 네트워크 인터페이스에의 직접 연결을 포함하는 전용 프로세서일 수 있으며, 그로써 통신의 다른 쪽 단부 상에서의 지연 시간을 감소시킨다.

[0040] 다른 실시예에서, 원격 메모리 인터페이스(131)에 의해 제공되는 것으로 앞서 기술된 기능이 서버 컴퓨팅 디바이스(130) 상에서 실행 중인 운영 체제 또는 유트리티에서 구현될 수 있다. 이와 유사하게, 원격 메모리 인터페이스(121)에 의해 제공되는 것으로 앞서 기술된 기능이 마찬가지로 서버 컴퓨팅 디바이스(120) 상에서 실행 중인 운영 체제 또는 유트리티에서 구현될 수 있다. 이러한 실시예에서, 이러한 원격 메모리 인터페이스에 의해 생성되어 보내지는 통신은 감소된 지연 시간을 제공하기 위해 감소된 네트워크 스택(network stack)을 통과할

수 있다. 예를 들어, 이러한 원격 메모리 인터페이스를 나타내는 컴퓨터 실행 가능 명령어들은, 기본 제공 드라이버(built-in driver)들 또는 적절한 기능을 가지는 것 등에 의해, 네트워킹 하드웨어에 직접 액세스할 수 있다.

[0041] 통상의 기술자라면 잘 알겠지만, 앞서 기술된 메커니즘들은 종래의 가상 메모리 메커니즘들과 상이하고, 데이터가 메모리로부터 스왑되는 저장 매체의 대체물이 결코 아니다. 결과적으로, 응답(225)이 종래의 가상 메모리와 관련하여 가능한 것보다 실질적으로 더 빠르게 제공될 수 있기 때문에, 간단한 일시 중단 및 재시작(resume)이 커맨드(223)와 같은 커맨드를 생성하는 실행 중인 프로세스들에 적용될 수 있다. 보다 구체적으로는 그리고 통상의 기술자라면 잘 알겠지만, 가상 메모리와 관련하여, 메모리에 더 이상 저장되어 있지 않고, 그 대신에, 디스크로 스왑되어 있는 데이터가 이러한 메모리에 요청될 때, 이러한 데이터가 보다 느린 디스크로부터 메모리로 다시 스왑될 때까지 요청측 프로세스의 실행이 일시 중단될 수 있다. 이러한 스왑이 완료될 때, 요청측 프로세스가 재시작될 수 있고, 요청된 데이터가, 예를 들어, 하나 이상의 처리 유닛들의 적절한 레지스터들에 로드되는 것 등에 의해, 요청측 프로세스에 제공될 수 있다. 그러나 앞서 기술된 메커니즘들에서, 데이터가 보다 느린 디스크 매체, 심지어 물리적으로 이러한 프로세스들이 실행 중인 컴퓨팅 디바이스의 일부인 디스크로부터 스왑될 수 있는 것보다 실질적으로 더 빠르게 원격 물리 메모리로부터 획득될 수 있다. 결과적으로, 보다 완전히 일시 중단된 실행 스크립트 또는 다른 유사한 처리를 재시작함에 있어서 불필요한 지연을 피하기 위해 간단한 일시 중단 및 재시작이 이용될 수 있다.

[0042] 예를 들어, 하나의 실시예에서, 커맨드(223)가, 그 커맨드(223)에 의해 지정되는 또는 그 커맨드(223)가 보내지는 메모리 주소, 페이지 또는 다른 유사한 위치 식별 정보에 기초하여, 주소 지정 가능한 원격 메모리(235)로 보내지는 것으로 결정될 수 있다. 이러한 실시예에서, 커맨드(223)가 원격 메모리에 의해 지원되는 로컬적으로 주소 지정 가능한 메모리 네임스페이스(231)의 부분(235)으로 보내지는 것으로 결정되는 경우, CPU(132)에 의해 실행되고 있는 프로세스는 일시 중단된 상태에 놓일 수 있고, 이 일시 중단된 상태로부터 종래의 일시 중단 상태로부터 보다 더 신속하게 재시작될 수 있다. 보다 구체적으로는, 실행되고 있는 프로세스가 자체적으로, 실행 중인 프로세스에 의해 지정된 메모리 위치에 기초하여, 커맨드(223)가 부분(235)으로 보내지는 것으로 결정할 수 있다. 결과적으로, 실행 중인 프로세스는 자동으로 그 자신을 일시 중단된 상태에 둘 수 있고, 이 일시 중단된 상태로부터 종래의 일시 중단 상태로부터 보다 더 신속하게 재시작될 수 있다. 대안적으로, 이러한 결정이 CPU(132) 또는 실행 중인 프로세스를 자동으로 일시 중단된 상태에 둘 수 있는 다른 유사한 구성요소에 의해 행해질 수 있다.

[0043] 다른 실시예에서, 원격 메모리 인터페이스(131) 또는 다른 메모리 관리 구성요소는 커맨드(223)가 부분(235)으로 보내지는 것을 검출할 수 있고, 실행 중인 프로세스에 그에 따라 통지할 수 있다. 보다 구체적으로는 그리고 앞서 언급한 바와 같이, 커맨드(223)가 보내지는 메모리 위치들이 검출될 수 있고, 그 메모리 위치들을 바탕으로, 커맨드(223)가 부분(235)으로 보내지는지에 관한 결정이 행해질 수 있다. 커맨드(223)가 부분(235)으로 보내지는 것으로 결정되는 경우, 예컨대, 실행 중인 프로세스로 또는 프로세스 관리 구성요소들로, 통지가 생성될 수 있다. 이러한 통지에 응답하여, 통지가 실행 중인 프로세스에 직접 또는 프로세스 관리 구성요소들에 제공되었는지에 따라, 실행 중인 프로세스는 그 자신을 일시 중단된 상태에 둘 수 있고, 이 일시 중단된 상태로부터 종래의 일시 중단 상태로부터 보다 더 신속하게 재시작될 수 있거나, 일시 중단된 상태에 놓여 있을 수 있다.

[0044] 도 3a 및 도 3b를 참조하면, 이 도면들에 도시된 흐름도들(301 및 302)은, 각각, 원격 컴퓨팅 디바이스 상에 물리적으로 설치된 메모리가 로컬적으로 실행 중인 프로세스에 의해 이용될 수 있는 예시적인 일련의 단계들을 나타내고 있다. 먼저 도 3a를 참조하면, 처음에, 단계(310)에서, 원격 메모리 인터페이스로 보내지는 네트워크 통신이 수신될 수 있다. 수신되면, 그 네트워크 통신은 앞서 언급된 LOAD 커맨드 또는 STORE 커맨드와 같은 적절한 네트워크 중심 커맨드(network-centric command)로 조립될 수 있다. 이러한 조립은 단계(315)의 일부로서 일어날 수 있고, 앞서 언급한 바와 같이, 네트워크 프로토콜들을 통해 다양한 컴퓨팅 디바이스들 간의 통신이 설정된 경우 어느 형식이든 적절한 형식으로부터 네트워크 통신을 역패키지화(unpackage)하는 것을 수반할 수 있다. 단계(320)에서, 적절한 커맨드가 로컬 메모리에서 수행될 수 있다. 예를 들어, 수신된 커맨드가 특정의 메모리 주소에서 시작하여 저장될 데이터를 지정하는 STORE 커맨드인 경우, 단계(320)에서, 이러한 데이터가 수신된 커맨드에 의해 지정된 주소에 상응하는 주소 또는 다른 유사한 메모리 위치에서 시작하는 로컬 메모리에 저장될 수 있을 것이다. 이와 유사하게, 수신된 커맨드가 지정된 주소 또는 다른 유사한 메모리 위치에서 시작하는 로컬 메모리에 저장된 데이터를 요청하는 LOAD 커맨드인 경우, 단계(320)에서, 로컬 메모리의 상응하는 위치들에 저장된 데이터가 획득될 수 있다. 하나의 실시예에서, 수신된 커맨드가 로컬 메모리와 관련하여 이용되어야 하는 주소를 지정할 수 있는 반면, 다른 실시예에서, 수신된 커맨드에 의해 지정된 주소가 공유될 메모리로

서 규정된 주소들, 페이지들, 또는 다른 위치들의 범위에 따라 변환될 수 있다.

[0045] 요청된 커맨드를 단계(320)에서 수행한 결과, 예를 들어, 데이터가 로컬 메모리에 저장된 경우 확인 응답(acknowledgement)과 같은 응답, 또는 로컬 메모리로부터 읽혀지도록 요청된 데이터를 포함하는 응답이 얻어질 수 있다. 단계(325)에서, 이러한 응답이 수신될 수 있다. 단계(330)에서, 이러한 응답이 네트워크 통신으로 변환될 수 있고, 이 네트워크 통신이 이어서 단계(310)에서 수신된 네트워크 통신을 송신했던 컴퓨팅 디바이스의 원격 메모리 인터페이스로 보내질 수 있다. 이전에 살펴본 바와 같이, 단계(330)에서 응답을 네트워크 통신으로 변환하는 것은, 예를 들어, 적절한 패킷 헤더들을 적용하는 것, 데이터를 최대 전송 단위에 상응하는 크기들로 분할하는 것, 적절한 주소 지정 정보를 제공하는 것, 및 다른 유사한 동작들을 비롯하여, 다른 컴퓨팅 디바이스들과의 통신이 설정되어 있는 네트워크에 의해 구현되는 네트워크 프로토콜들에 따라 응답을 패킷화하는 것을 포함할 수 있다. 네트워크 통신은, 일단 생성되면, 단계(335)에서, 단계(310)에서 네트워크 통신을 송신했던 컴퓨팅 디바이스의 원격 메모리 인터페이스로 전송될 수 있다. 관련 처리가 이어서 단계(340)에서 종료될 수 있다.

[0046] 도 3b를 참조하면, 유사한 일련의 단계들이, 단계(350)에서, 로컬적으로 주소 지정 가능한 메모리 네임스페이스로 보내지는 LOAD 또는 STORE 커맨드와 같은 메모리 중심 커맨드를 로컬 프로세스로부터 수신하는 것으로 시작할 수 있다. 이 요청은 로컬적으로 주소 지정 가능한 메모리 네임스페이스에서의 하나 이상의 주소들 또는 다른 유사한 위치 식별자들을 지정할 수 있다. 초기 문제로서, 따라서, 하나의 실시예에서, 단계(355)에서, 단계(350)의 메모리 중심 커맨드에 의해 지정된 주소들이 앞서 상세히 기술된 것과 같은 원격 메모리 인터페이스에 의해 지원되는 주소 범위에 있는지에 관한 검사가 행해질 수 있다. 단계(355)에서, 메모리 중심 커맨드가 로컬적으로 설치된 메모리에 의해 지원되는 로컬적으로 주소 지정 가능한 메모리 네임스페이스의 주소들의 범위에 있지 않은 주소들로 보내지는 것으로 결정되는 경우, 흐름도(302)에 도시된 바와 같이, 원격 메모리 공유에 관련된 처리가 단계(385)에서 종료될 수 있다. 이와 달리, 단계(355)에서, 메모리 중심 커맨드가 원격 메모리 인터페이스에 의해 지원되는 로컬적으로 주소 지정 가능한 메모리 네임스페이스의 주소들의 범위에 있는 주소들로 보내지는 것으로 결정되는 경우, 처리는 단계(360)를 계속할 수 있다.

[0047] 단계(360)에서, 단계(350)에서 수신된 요청이 보내지는 주소가, 요청이 보내지는, 데이터를 저장하는 데 사용되는 메모리를 갖는 하나 이상의 원격 컴퓨팅 디바이스들의 ID(identification)로 변환될 수 있다. 보다 구체적으로는, 하나의 실시예에서, 앞서 상세히 기술된 방식 등으로, 원격 메모리 인터페이스가 STORE 커맨드를 수신하고 데이터를 원격 컴퓨팅 디바이스의 메모리에 저장할 때마다, 원격 메모리 인터페이스는 STORE 커맨드가 보내지는 로컬적으로 주소 지정 가능한 메모리 네임스페이스의 주소와 원격 컴퓨팅 디바이스의 식별자(네트워크 주소 등) - 이 원격 컴퓨팅 디바이스의 메모리에 이러한 데이터가 궁극적으로 저장됨 - 간의 연관 관계(association)를 기록할 수 있다. 그 후에, LOAD 커맨드가 로컬적으로 실행 중인 프로세스로부터 로컬적으로 주소 지정 가능한 메모리 네임스페이스에서의 동일한 주소에 대해 발행될 때, 원격 메모리 인터페이스는 이전에 기록된 연관 관계를 참조하고, 그 데이터를 얻기 위해 어느 원격 컴퓨팅 디바이스와 통신해야 하는지를 결정할 수 있다. 그에 부가하여, 하나의 실시예에서, 로컬적으로 주소 지정 가능한 메모리 네임스페이스에서의 특정의 주소에 대한 STORE 커맨드를 처음으로 수신할 때, 원격 메모리 인터페이스는 이러한 데이터를, 메모리 공유 제어기에 의해 식별되어 원격 메모리 인터페이스에 통지될 수 있는 또는 메모리 공유 제어기에 의해 식별된 컴퓨팅 디바이스들 중에서 원격 메모리 인터페이스에 의해 선택될 수 있는 원격 컴퓨팅 디바이스의 공유 메모리에 저장하려고 시도할 수 있다. 단계(360)에서, 원격 컴퓨팅 디바이스가 식별되면, 처리는 단계(365)를 계속할 수 있다. 단계(365)에서, 단계(350)에서 수신된 요청이 단계(360)에서 식별된 컴퓨팅 디바이스의 식별된 원격 메모리 인터페이스로 보내질 수 있는 네트워크 통신으로 변환될 수 있다. 이전에 살펴본 바와 같이, 이러한 변환은 로컬 컴퓨팅 디바이스와 메모리를 포함하는 원격 컴퓨팅 디바이스 사이에서 통신이 전달되는 네트워크에 의해 이용되는 프로토콜들에 따라 요청을 패킷화하고 다른 방식으로 데이터 스트림을 생성하는 것을 포함할 수 있다.

[0048] 단계(365)에서의 전송에 응답하여, 단계(370)에서, 로컬 컴퓨팅 디바이스 상의 원격 메모리 인터페이스로 보내지는 응답 네트워크 통신이 수신될 수 있다. 단계(375)에서, 앞서 상세히 기술된 방식 등으로, 그 네트워크 통신이 단계(350)에서 수신된 요청에 대한 응답으로 조립될 수 있다. 단계(380)에서, 이러한 응답이 단계(350)에서 수신된 요청을 생성한 실행 중인 프로세스에 제공될 수 있다. 관련 처리가 이어서 단계(385)에서 종료될 수 있다.

[0049] 도 4를 참조하면, 앞서 상세히 기술된 메커니즘들 중 일부를 실행할 수 있는 것과 같은 범용 컴퓨팅 디바이스들 및 앞서 기술된 스위치들과 같은 특수 목적 컴퓨팅 디바이스들 둘 다를 포함할 수 있는 예시적인 컴퓨팅 디바이스가 예시되어 있다. 예시적인 컴퓨팅 디바이스(400)는 하나 이상의 중앙 처리 유닛(CPU)들(420), 시스템 메모

리(430), 및 시스템 메모리를 비롯한 다양한 시스템 구성요소들을 처리 유닛(420)에 연결시키는 시스템 버스(421)(이들로 제한되지 않음)를 포함할 수 있다. 시스템 버스(421)는 각종의 버스 아키텍처들 중 임의의 것을 사용하는, 메모리 버스 또는 메모리 제어기, 주변 기기 버스, 및 로컬 버스를 비롯한 몇가지 유형의 버스 구조들 중 임의의 것일 수 있다. 구체적인 물리적 구현에 따라, 컴퓨팅 디바이스(400)의 CPU들(420), 시스템 메모리(430) 및 다른 구성요소들 중 하나 이상이 물리적으로 같은 장소에(예컨대, 단일 칩 상에) 배치될 수 있다. 이러한 경우에, 시스템 버스(421)의 일부 또는 전부는 단일 칩 구조 내의 통신 경로에 불과할 수 있고, 도 4에서 그에 대한 예시는 설명을 위한 표기법상 편의에 불과할 수 있다.

[0050] 컴퓨팅 디바이스(400)는 또한 전형적으로 컴퓨팅 디바이스(400)에 의해 액세스될 수 있는 임의의 이용 가능한 매체를 포함할 수 있는 컴퓨터 판독 가능 매체를 포함한다. 제한이 아닌 예로서, 컴퓨터 판독 가능 매체는 컴퓨터 저장 매체 및 통신 매체를 포함할 수 있다. 컴퓨터 저장 매체는 컴퓨터 판독 가능 명령어들, 데이터 구조들, 프로그램 모듈들 또는 다른 데이터와 같은 정보를 저장하기 위해 임의의 방법 또는 기술로 구현된 매체를 포함한다. 컴퓨터 저장 매체는 RAM, ROM, EEPROM, 플래시 메모리 또는 다른 메모리 기술, CD-ROM, DVD(digital versatile disk) 또는 다른 광 디스크 저장소, 자기 카세트, 자기 테이프, 자기 디스크 저장소 또는 다른 자기 저장 디바이스, 또는 원하는 정보를 저장하는 데 사용될 수 있고 컴퓨팅 디바이스(400)에 의해 액세스될 수 있는 임의의 다른 매체를 포함하지만, 이들로 제한되지 않는다. 그렇지만, 컴퓨터 저장 매체는 통신 매체를 포함하지 않는다. 통신 매체는 전형적으로 컴퓨터 판독 가능 명령어들, 데이터 구조들, 프로그램 모듈들 또는 다른 데이터를 반송파 또는 다른 전송 메커니즘과 같은 피변조 데이터 신호(modulated data signal)로 구현하고 임의의 정보 전달 매체를 포함한다. 제한이 아닌 예로서, 통신 매체는 유선 네트워크 또는 직접 유선 연결(direct-wired connection)과 같은 유선 매체와, 음향, RF, 적외선 및 다른 무선 매체와 같은 무선 매체를 포함한다. 상기한 것들 중 임의의 것의 조합들도 컴퓨터 판독 가능 매체의 범주 내에 포함되어야 한다.

[0051] 시스템 메모리(430)는 ROM(read only memory)(431) 및 RAM(random access memory)(432)와 같은 휘발성 및/또는 비휘발성 메모리 형태의 컴퓨터 저장 매체를 포함한다. 시작 중과 같은 때에, 컴퓨팅 디바이스(400) 내의 요소들 간에 정보를 전송하는 데 도움을 주는 기본 루틴들이 들어 있는 기본 입출력 시스템(basic input/output system)(BIOS)(433)은 전형적으로 ROM(431)에 저장되어 있다. RAM(432)에는 전형적으로 처리 유닛(420)에 의해 즉각 액세스될 수 있고/있거나 현재 처리되고 있는 데이터 및/또는 프로그램 모듈들이 들어 있다. 제한이 아닌 예로서, 도 4는 운영 체제(434), 다른 프로그램 모듈들(435), 및 프로그램 데이터(436)를 나타내고 있다.

[0052] 통신 매체를 사용할 때, 컴퓨팅 디바이스(400)는 하나 이상의 원격 컴퓨터들로의 논리적 연결들을 통해 네트워크화된 환경에서 동작할 수 있다. 도 4에 도시된 논리적 연결은 LAN(local area network), 인터넷과 같은 WAN(wide area network), 또는 다른 네트워크들일 수 있는, 네트워크(190)에의 일반 네트워크 접속(471)이다. 컴퓨팅 디바이스(400)는 네트워크 인터페이스 또는 어댑터(470)를 통해 일반 네트워크 접속(471)에 연결되고, 네트워크 인터페이스 또는 어댑터(470)는 차례로 시스템 버스(421)에 연결되어 있다. 네트워크화된 환경에서, 컴퓨팅 디바이스(400) 또는 그것의 일부분들 또는 주변 기기들과 관련하여 나타내어진 프로그램 모듈들이 일반 네트워크 접속(471)을 통해 컴퓨팅 디바이스(400)에 통신 연결되어 있는 하나 이상의 다른 컴퓨팅 디바이스들의 메모리에 저장될 수 있다. 도시된 네트워크 접속들이 예시적인 것이고 컴퓨팅 디바이스들 간에 통신 링크를 설정하는 다른 수단이 사용될 수 있다는 것을 잘 알 것이다.

[0053] 컴퓨팅 디바이스(400)는 다른 분리식/비분리식, 휘발성/비휘발성 컴퓨터 저장 매체도 포함할 수 있다. 단지 예로서, 도 4는 비분리식 비휘발성 매체로부터 판독하거나 그 매체에 기입하는 하드 디스크 드라이브(441)를 나타내고 있다. 예시적인 컴퓨팅 디바이스에서 사용될 수 있는 다른 분리식/비분리식, 휘발성/비휘발성 컴퓨터 저장 매체는 자기 테이프 카세트, 플래시 메모리 카드, DVD(digital versatile disk), 디지털 비디오 테이프, 고상 RAM(solid state RAM), 고상 ROM(solid state ROM) 등을 포함하지만, 이들로 제한되지 않는다. 하드 디스크 드라이브(441)는 전형적으로 인터페이스(440)와 같은 비분리식 메모리 인터페이스를 통해 시스템 버스(421)에 연결된다.

[0054] 이상에서 논의되고 도 4에 예시되어 있는 드라이브들 및 이들과 연관된 컴퓨터 저장 매체는 컴퓨팅 디바이스(400)에 대한 컴퓨터 판독 가능 명령어들, 데이터 구조들, 프로그램 모듈들 및 다른 데이터의 저장을 제공한다. 도 4에서, 예를 들어, 하드 디스크 드라이브(441)는 운영 체제(444), 다른 프로그램 모듈들(445), 및 프로그램 데이터(446)를 저장하는 것으로 예시되어 있다. 유의할 점은 이 구성요소들이 운영 체제(434), 다른 프로그램 모듈들(435), 및 프로그램 데이터(436)와 동일하거나 상이할 수 있다는 것이다. 운영 체제(444), 다른 프로그램 모듈들(445), 및 프로그램 데이터(446)에 여기서 상이한 참조 번호가 부여되어 있는 것은 최소한 이들이 상이한

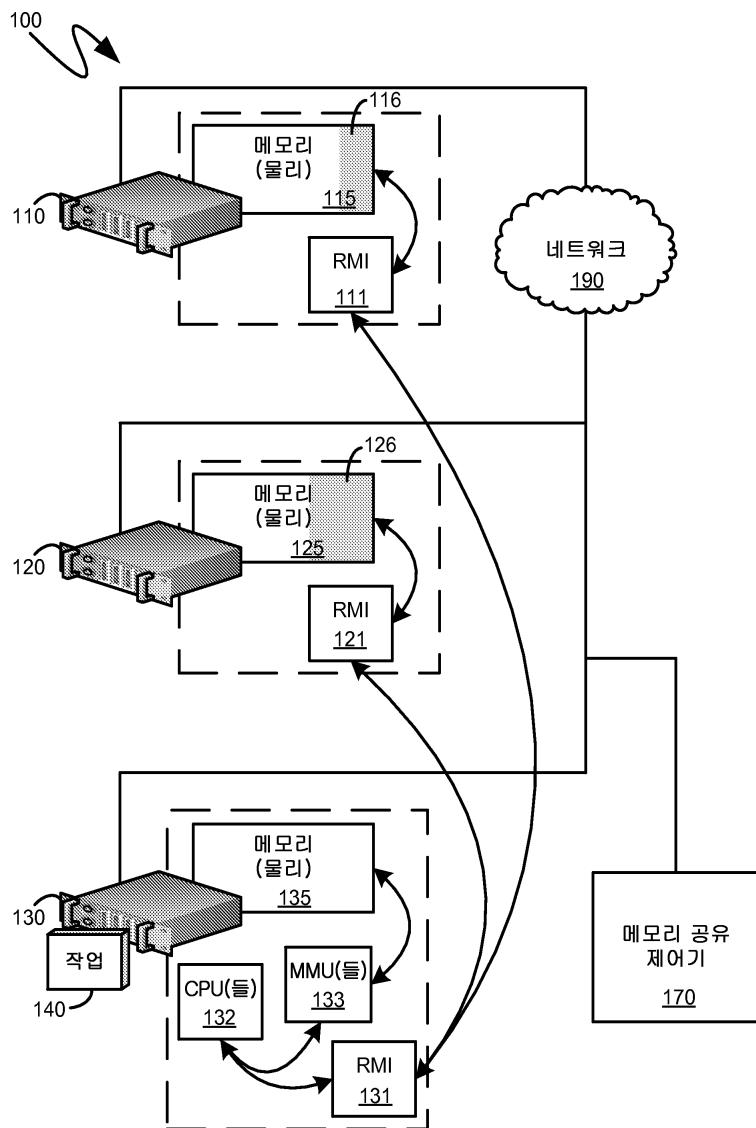
사본들임을 나타내기 위한 것이다.

[0055]

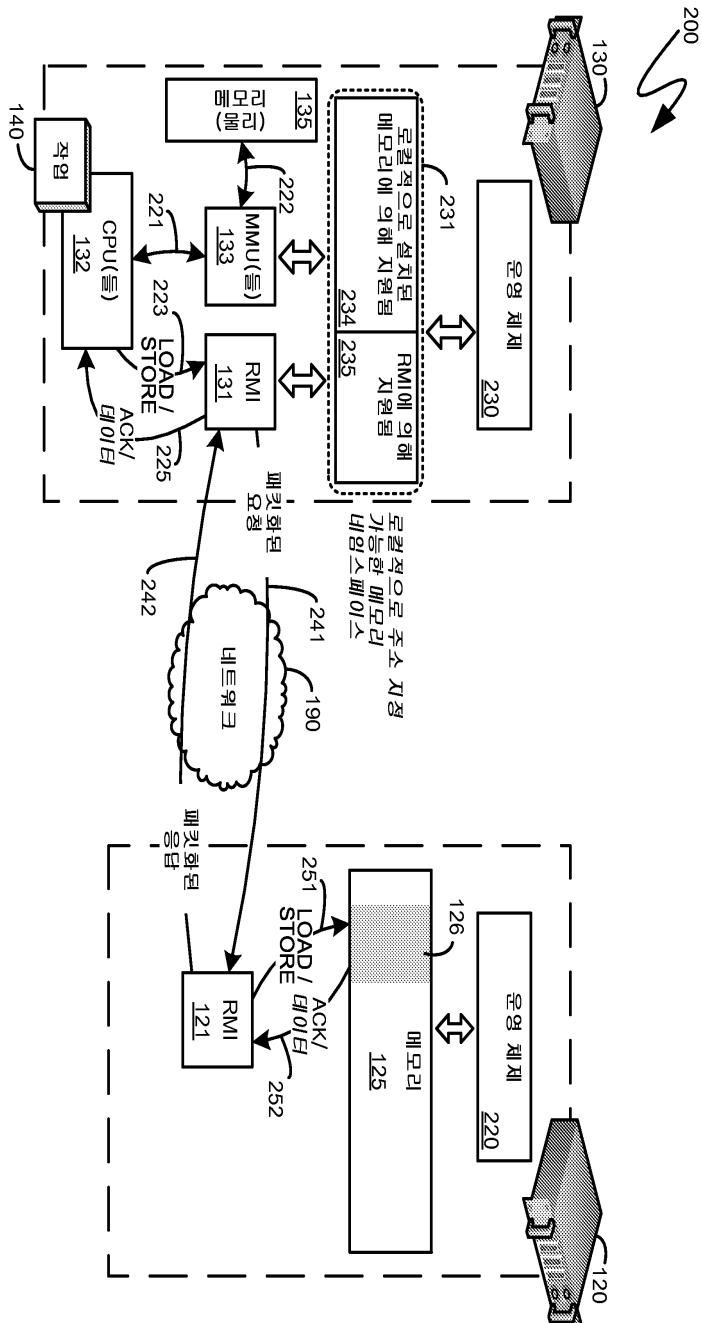
이상의 설명으로부터 알 수 있는 바와 같이, 다수의 물리적으로 상이한 컴퓨팅 디바이스들 간에 메모리를 공유하는 메커니즘이 제시되어 있다. 본 명세서에 기술된 발명 요지의 많은 가능한 변형들을 고려하여, 이하의 청구범위 및 그 규들물들의 범주 내에 속할 수 있는 모든 이러한 실시예들을 우리의 발명으로서 청구한다.

도면

도면1

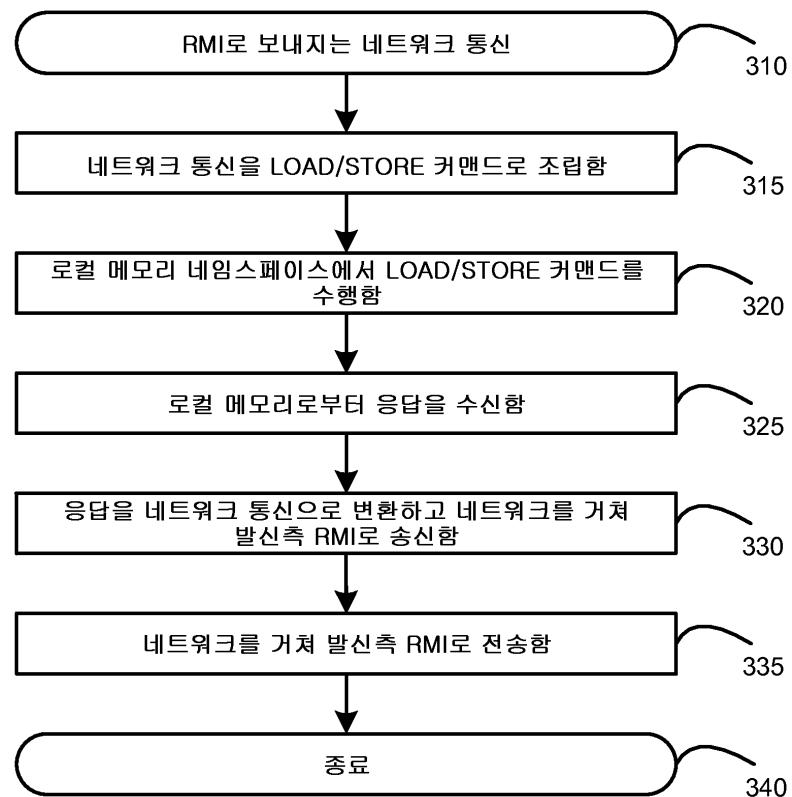


도면2



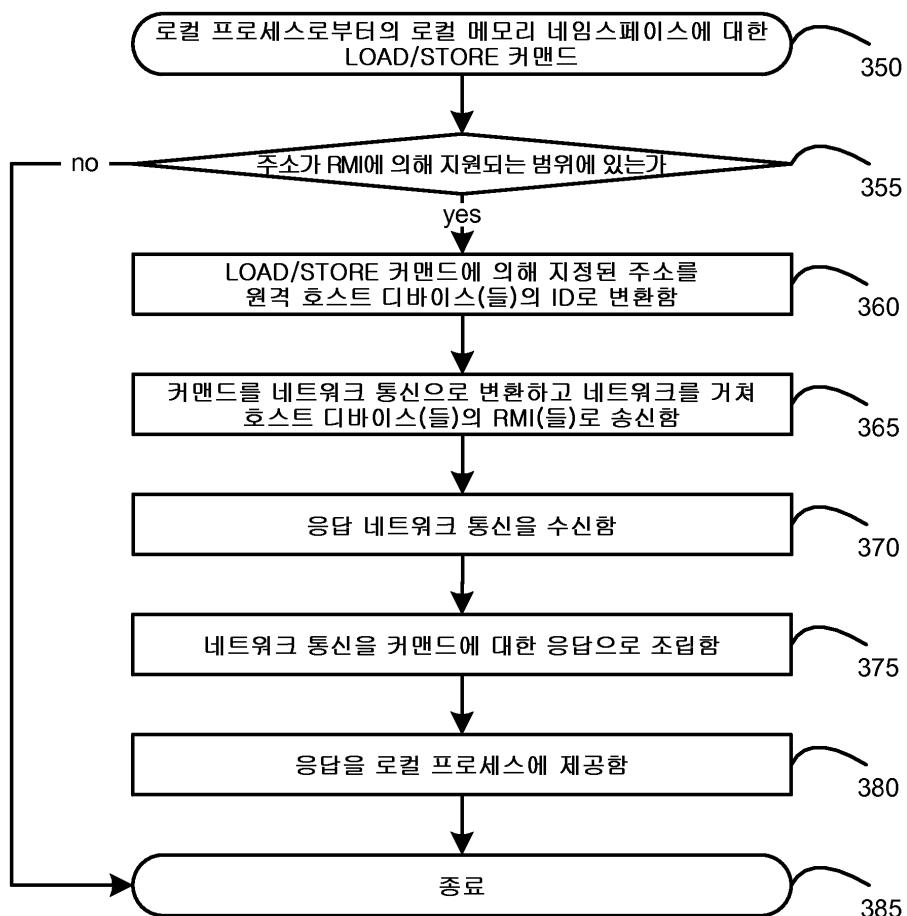
도면3a

301



도면3b

302



도면4

