



등록특허 10-2059121



(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2019년12월24일
(11) 등록번호 10-2059121
(24) 등록일자 2019년12월18일

- (51) 국제특허분류(Int. Cl.)
G06F 15/173 (2006.01)
- (21) 출원번호 10-2014-7007211
- (22) 출원일자(국제) 2012년09월25일
심사청구일자 2017년09월15일
- (85) 번역문제출일자 2014년03월18일
- (65) 공개번호 10-2014-0068989
- (43) 공개일자 2014년06월09일
- (86) 국제출원번호 PCT/US2012/057121
- (87) 국제공개번호 WO 2013/049064
국제공개일자 2013년04월04일
- (30) 우선권주장
13/415,712 2012년03월08일 미국(US)
61/541,056 2011년09월29일 미국(US)
- (56) 선행기술조사문헌
KR1020060064544 A
US5133053 A

(73) 특허권자
오라클 인터내셔널 코포레이션
미국, 캘리포니아 94065, 레드우드 쇼어스 엠에스
5오피7, 오라클 파크웨이 500

(72) 발명자
쉬 폐이치
중국 베이징 시챙 디스트릭트 유수구안실리 스트
리트 빌딩 #5 룸 401
진 용순
중국 베이징 하이디안 디스트릭트 쟈오톰 유니버
시티 로드 1-2-1004

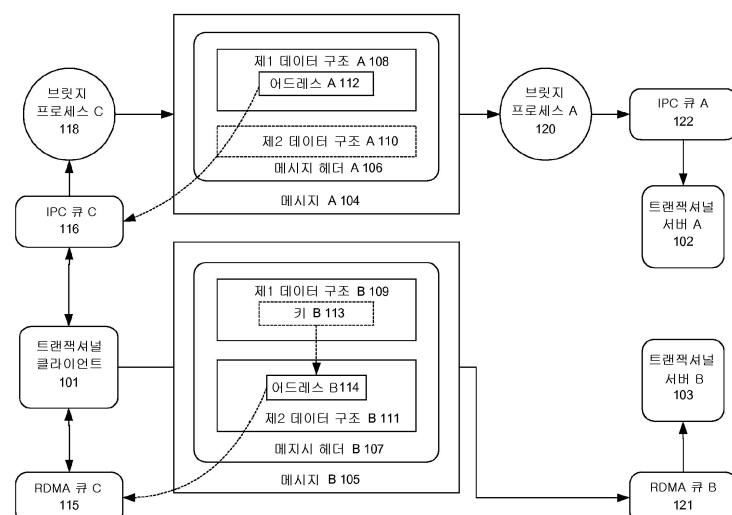
(74) 대리인
박장원

전체 청구항 수 : 총 21 항

심사관 : 홍경아

(54) 발명의 명칭 **트랜잭션 네트워크 환경에서 콤플렉스 메시지 헤더를 지원하기 위한 시스템 및 방법****(57) 요약**

플렉서블 트랜잭션 데이터 구조는 트랜잭션 네트워크 환경에 메시지 헤더를 저장하기 위해 사용될 수 있다. 상기 플렉서블 트랜잭션 네트워크 데이터 구조는 동적인 수의 필드들을 가질 수 있고, 특정 ID들을 통해 액세스가능하다. 메시지 헤더는 제1 메시지 큐를 이용하여 클라이언트에 액세스하기 위한 어드레스 정보를 저장하는 제1 데이터 구조 및 제2 메시지 큐를 이용하여 클라이언트에 액세스하기 위한 어드레스 정보를 저장하는 제2 데이터 구조를 포함할 수 있다. 제1 타입의 서버는 상기 제1 메시지 큐를 이용하여 상기 클라이언트에 액세스하기 위한 상기 어드레스 정보를 획득하는 데 상기 제1 데이터 구조만을 이용하는 동작을 한다. 제2 타입의 서버는 상기 제1 데이터 구조로부터 키를 획득하고 그 다음, 상기 제2 메시지 큐를 이용하여 상기 클라이언트에 액세스하기 위한 상기 어드레스 정보를 상기 제2 데이터 구조로부터 획득하는 데 상기 키를 이용하는 동작을 한다.

대표도

명세서

청구범위

청구항 1

트랜잭션 네트워크 미들웨어 머신 환경(transactional middleware machine environment)에서 콤플렉스 메시지 헤더를 지원하기 위한 시스템으로서,

제1 메시지 큐(queue)를 이용하여 트랜잭션 클라이언트에 액세스하기 위한 어드레스 정보를 저장하는 제1 데이터 구조와; 그리고

제2 메시지 큐를 이용하여 상기 트랜잭션 네트워크 클라이언트에 액세스하기 위한 어드레스 정보를 저장하는 제2 데이터 구조를 포함하고,

여기서, 제1 탑의 트랜잭션 네트워크 서버가 상기 제1 메시지 큐를 이용하여 상기 트랜잭션 네트워크 클라이언트와 통신하기 위한 상기 어드레스 정보를 획득하기 위해 상기 콤플렉스 메시지 헤더 내의 상기 제1 데이터 구조만을 체크하는 동작을 하고,

제2 탑의 트랜잭션 네트워크 서버가 상기 콤플렉스 메시지 헤더 내의 상기 제1 데이터 구조로부터 키(key)를 획득하고, 상기 제2 메시지 큐를 이용하여 상기 트랜잭션 네트워크 클라이언트와 통신하기 위한 상기 콤플렉스 메시지 헤더 내의 상기 제2 데이터 구조로부터의 상기 어드레스 정보를 획득하기 위해 상기 키를 이용하는 동작을 하는 것을 특징으로 하는 트랜잭션 네트워크 미들웨어 머신 환경에서 콤플렉스 메시지 헤더를 지원하기 위한 시스템.

청구항 2

제1항에 있어서,

상기 트랜잭션 네트워크 클라이언트는 상기 콤플렉스 메시지 헤더를 사용하여 트랜잭션 네트워크 서버에 메시지를 보내는 동작을 하는 것을 특징으로 하는 트랜잭션 네트워크 미들웨어 머신 환경에서 콤플렉스 메시지 헤더를 지원하기 위한 시스템.

청구항 3

제2항에 있어서,

상기 트랜잭션 네트워크 클라이언트는 상기 메시지를 보내기 전에 상기 트랜잭션 네트워크 서버 상에서 정보를 얻는 동작을 하는 것을 특징으로 하는 트랜잭션 네트워크 미들웨어 머신 환경에서 콤플렉스 메시지 헤더를 지원하기 위한 시스템.

청구항 4

제2항에 있어서,

상기 트랜잭션 네트워크 서버는 상기 콤플렉스 메시지 헤더 내의 상기 어드레스 정보에 근거하여 상기 트랜잭션 네트워크 클라이언트에 응답 메시지를 보내는 동작을 하는 것을 특징으로 하는 트랜잭션 네트워크 미들웨어 머신 환경에서 콤플렉스 메시지 헤더를 지원하기 위한 시스템.

청구항 5

제1항에 있어서,

상기 제1 데이터 구조는 C 데이터 구조이고, 상기 제2 데이터 구조는 타입화된 버퍼(typed buffer) 데이터 구조인 것을 특징으로 하는 트랜잭션 네트워크 미들웨어 머신 환경에서 콤플렉스 메시지 헤더를 지원하기 위한 시스템.

청구항 6

제1항에 있어서,

상기 트랜잭션 네트워크 클라이언트는 로컬 브릿지 프로세스(local bridge process)를 통해 제1 탑의 트랜잭션 네트워크 서버와 통신하는 동작을 하는 것을 특징으로 하는 트랜잭션 네트워크 미들웨어 머신 환경에서 콤플렉스 메시지 헤더를 지원하는 시스템.

원하기 위한 시스템.

청구항 7

제1항에 있어서,

상기 트랜잭션 클라이언트는 고성능 네트워크를 통해 제2 타입의 트랜잭션 서버와 직접 통신하는 동작을 하는 것을 특징으로 하는 트랜잭션 미들웨어 머신 환경에서 캠플렉스 메시지 헤더를 지원하기 위한 시스템.

청구항 8

제7항에 있어서,

상기 고성능 네트워크는 RDMA(remote direct memory access) 프로토콜을 사용하는 인피니밴드(IB) 네트워크인 것을 특징으로 하는 트랜잭션 미들웨어 머신 환경에서 캠플렉스 메시지 헤더를 지원하기 위한 시스템.

청구항 9

제1항에 있어서,

제1 타입의 트랜잭션 서버는 캠플렉스 메시지 헤더의 제2 데이터 구조 내의 어드레스 정보를 이해하지 못하는 것을 특징으로 하는 트랜잭션 미들웨어 머신 환경에서 캠플렉스 메시지 헤더를 지원하기 위한 시스템.

청구항 10

제1항에 있어서,

상기 제2 데이터 구조는 동적인 수의 필드들을 가진 플렉서블 트랜잭션 데이터 구조이고, 상기 필드들의 각각은 특정 ID를 통해 액세스 가능한 것을 특징으로 하는 트랜잭션 미들웨어 머신 환경에서 캠플렉스 메시지 헤더를 지원하기 위한 시스템.

청구항 11

트랜잭션 미들웨어 머신 환경에서 캠플렉스 메시지 헤더를 지원하기 위한 방법으로서,

제1 메시지 큐를 이용하여 트랜잭션 클라이언트에 액세스하기 위한 어드레스 정보를 저장하는 제1 데이터 구조를 제공하는 단계와;

제2 메시지 큐를 이용하여 상기 트랜잭션 클라이언트에 액세스하기 위한 어드레스 정보를 저장하는 제2 데이터 구조를 제공하는 단계와;

제1 타입의 트랜잭션 서버로 하여금 상기 제1 메시지 큐를 이용하여 상기 트랜잭션 클라이언트와 통신하기 위한 상기 어드레스 정보를 획득하기 위해 상기 캠플렉스 메시지 헤더 내의 상기 제1 데이터 구조만을 체크하도록 하는 단계와; 그리고

제2 타입의 트랜잭션 서버로 하여금 상기 캠플렉스 메시지 헤더 내의 상기 제1 데이터 구조로부터 키를 획득하고, 상기 제2 메시지 큐를 이용하여 상기 트랜잭션 클라이언트와 통신하기 위한 상기 캠플렉스 메시지 헤더 내의 상기 제2 데이터 구조로부터의 상기 어드레스 정보를 획득하기 위해 상기 키를 사용하게끔 하는 단계를 포함하는 것을 특징으로 하는 트랜잭션 미들웨어 머신 환경에서 캠플렉스 메시지 헤더를 지원하기 위한 방법.

청구항 12

제11항에 있어서,

상기 트랜잭션 클라이언트로 하여금 상기 캠플렉스 메시지 헤더를 이용하여 트랜잭션 서버에 메시지를 보내도록 하는 것을 더 포함하는 것을 특징으로 하는 트랜잭션 미들웨어 머신 환경에서 캠플렉스 메시지 헤더를 지원하기 위한 방법.

청구항 13

제12항에 있어서,

상기 트랜잭션 클라이언트로 하여금 상기 메시지를 보내기 전에 상기 트랜잭션 서버 상에서 정보를 얻도록

하는 것을 더 포함하는 것을 특징으로 하는 트랜잭션 네트워크 미들웨어 머신 환경에서 캡플렉스 메시지 헤더를 지원하기 위한 방법.

청구항 14

제12항에 있어서,

상기 트랜잭션 네트워크 서버로 하여금 상기 캡플렉스 메시지 헤더 내의 상기 어드레스 정보에 근거하여 상기 트랜잭션 네트워크 클라이언트에 응답 메시지를 보내도록 하는 것을 더 포함하는 것을 특징으로 하는 트랜잭션 네트워크 미들웨어 머신 환경에서 캡플렉스 메시지 헤더를 지원하기 위한 방법.

청구항 15

제11항에 있어서,

상기 제1 데이터 구조는 C 데이터 구조이고, 상기 제2 데이터 구조는 타입화된 버퍼 데이터 구조인 것을 특징으로 하는 트랜잭션 네트워크 미들웨어 머신 환경에서 캡플렉스 메시지 헤더를 지원하기 위한 방법.

청구항 16

제11항에 있어서,

상기 트랜잭션 네트워크 클라이언트로 하여금 로컬 브릿지 프로세스를 통해 제1 타입 트랜잭션 네트워크 서버와 통신하도록 하는 것을 더 포함하는 것을 특징으로 하는 트랜잭션 네트워크 미들웨어 머신 환경에서 캡플렉스 메시지 헤더를 지원하기 위한 방법.

청구항 17

제11항에 있어서,

상기 트랜잭션 네트워크 클라이언트로 하여금 고성능 네트워크를 통해 바로 제2 타입 트랜잭션 네트워크 서버와 통신하도록 하는 것을 더 포함하는 것을 특징으로 하는 트랜잭션 네트워크 미들웨어 머신 환경에서 캡플렉스 메시지 헤더를 지원하기 위한 방법.

청구항 18

제17항에 있어서,

상기 고성능 네트워크는 RDMA 프로토콜을 사용하는 인피니밴드(IB) 네트워크인 것을 특징으로 하는 트랜잭션 네트워크 미들웨어 머신 환경에서 캡플렉스 메시지 헤더를 지원하기 위한 방법.

청구항 19

제11항에 있어서,

제1 타입 트랜잭션 네트워크 서버로 하여금 상기 캡플렉스 메시지 헤더의 제2 데이터 구조 내의 어드레스 정보를 이해하지 못하도록 하는 것을 더 포함하는 것을 특징으로 하는 트랜잭션 네트워크 미들웨어 머신 환경에서 캡플렉스 메시지 헤더를 지원하기 위한 방법.

청구항 20

제11항에 있어서,

상기 제2 데이터 구조는 동적인 수의 필드들을 가진 플렉서블 트랜잭션 네트워크 데이터 구조이며, 상기 필드들의 각각은 특정 ID를 통해 액세스 가능한 것을 특징으로 하는 트랜잭션 네트워크 미들웨어 머신 환경에서 캡플렉스 메시지 헤더를 지원하기 위한 방법.

청구항 21

삭제

청구항 22

명령어들을 포함하는 컴퓨터 프로그램을 저장하는 비-휘발성 저장 매체로서, 상기 명령어들은 실행시 컴퓨터로 하여금 청구항 제11항 내지 제20항 중 어느 한 항의 방법을 수행하게 하는 비-휘발성 저장 매체.

발명의 설명

기술 분야

[0001] 저작권 공지

[0002] 본 명세서에서 개시된 부분은 저작권 보호를 받는 내용을 포함한다. 저작권자는 미국특허상표청의 특허 파일 또는 기록에 나타난 대로 본 특허 문서 또는 특허 개시내용을 어느 누군가가 팩스밀리 재생하는 것은 반대하지 않지만, 그 밖의 모든 것은 저작권으로 보호된다.

[0003] 기술분야

[0004] 본 발명은 일반적으로, 미들웨어와 같은 컴퓨터 시스템들 및 소프트웨어에 관한 것이며, 특히 트랜잭션 네트워크 미들웨어 환경(transactional middleware machine environment)을 지원하는 것에 관한 것이다.

배경 기술

[0005] 트랜잭션 네트워크 미들웨어 시스템 또는 트랜잭션 지향식 미들웨어(transaction oriented middleware)는 조직 내에서 다양한 트랜잭션들을 프로세스할 수 있는 기업 어플리케이션 서버들을 포함한다. 고성능 네트워크 및 멀티프로세서 컴퓨터와 같은 새로운 기술들의 발달에 따라, 트랜잭션 네트워크 미들웨어의 성능을 더 향상시킬 필요가 있다. 이러한 것들은 대체로 본 발명의 실시예들이 해결하도록 의도된 영역들이다.

발명의 내용

[0006] 트랜잭션 네트워크 미들웨어 환경에서 캡슐화된 메시지 헤더를 지원하기 위한 시스템 및 방법이 본 명세서에 개시된다. 상기 캡슐화된 메시지 헤더는 제1 메시지 큐(queue)를 이용하여 트랜잭션 네트워크 클라이언트에 액세스하기 위한 어드레스 정보를 저장하는 제1 데이터 구조 및 제2 메시지 큐를 이용하여 상기 트랜잭션 네트워크 클라이언트를 어드레싱하기 위한 어드레스 정보를 저장하는 제2 데이터 구조를 포함한다. 제1 탑재된 트랜잭션 네트워크 서버는 상기 제1 메시지 큐를 이용하여 상기 트랜잭션 네트워크 클라이언트와 통신하기 위한 상기 어드레스 정보를 획득하기 위해 상기 캡슐화된 메시지 헤더 내의 상기 제1 데이터 구조를 체크하는 동작을 한다. 제2 탑재된 트랜잭션 네트워크 서버는 상기 캡슐화된 메시지 헤더 내의 상기 제1 데이터 구조로부터 키(key)를 획득하고 상기 제2 메시지 큐를 이용하여 상기 트랜잭션 네트워크 클라이언트와 통신하기 위해 상기 캡슐화된 메시지 헤더 내의 상기 제2 데이터 구조로부터 상기 어드레스 정보를 획득하기 위해 상기 키를 이용하는 동작을 한다.

도면의 간단한 설명

[0007] 도 1은 본 발명의 실시예에 따른 캡슐화된 메시지 헤더를 지원하는 트랜잭션 네트워크 미들웨어 환경의 예를 도시한다.

도 2는 본 발명의 실시예에 따른 트랜잭션 네트워크 미들웨어 환경에서 캡슐화된 메시지 헤더를 지원하기 위한 예시적인 흐름도를 도시한다.

도 3은 본 발명의 실시예에 따른 트랜잭션 네트워크 미들웨어 환경에서 캡슐화된 메시지 헤더의 예를 도시한다.

발명을 실시하기 위한 구체적인 내용

[0008] 복수의 프로세서들을 가진 고속 머신들 및 고성능 네트워크 연결의 장점을 취할 수 있는 Tuxedo와 같은 트랜잭션 네트워크 미들웨어 시스템을 지원하기 위한 시스템 및 방법이 본 명세서에 기술된다. 플렉서블 트랜잭션 데이터 구조(flexible transactional data structure)가 트랜잭션 네트워크 미들웨어 환경에서 메시지 헤더를 저장하는데 사용될 수 있다. 상기 플렉서블 트랜잭션 데이터 구조는 동적인 수의 필드들을 가질 수 있고, 특정 ID들을 통해 액세스 가능하다. 메시지 헤더는 제1 메시지 큐를 이용하여 클라이언트에 액세스하기 위한 어드레스 정보를 저장하는 제1 데이터 구조 및 제2 메시지 큐를 이용하여 클라이언트에 액세스하기 위한 어드레스 정보를 저장하는 제2 데이터 구조를 포함할 수 있다. 제1 탑재된 서버는 상기 제1 메시지 큐를 이용하여 상기 클라이언트에 액세스하기 위한 상기 어드레스 정보를 획득하기 위해 상기 제1 데이터 구조만을 이용하는 동작을 한다. 제2 탑재된 서버는 상기 제1 데이터 구조로부터 키를 획득하고 그 다음, 상기 제2 메시지 큐를 이용하여 상기 클라이언트에

언트에 액세스하기 위한 상기 어드레스 정보를 상기 제2 데이터 구조로부터 획득하기 위해 상기 키를 이용하는 동작을 한다.

[0009] 본 발명의 실시예에 따르면, 시스템은, 신속하게 준비될 수 있고 수요에 따라 스케일링될 수 있는 대량의 병렬 인-메모리 그리드(massively parallel in-memory grid)를 포함하는 완전한 Java EE 어플리케이션 서버 커플렉스를 제공하기 위해 WebLogic Suite와 같은 어플리케이션 서버 또는 미들웨어 환경과 함께 고성능 하드웨어, 예컨대 64-비트 프로세서 기술, 고성능 대용량 메모리 및 리던던트 인피니밴드(redundant InfiniBand) 및 이더넷 네트워킹의 조합을 포함한다. 실시예에 따르면, 시스템은 전체, 절반 또는 1/4 랙(rack) 또는 다른 구성으로 배치될 수 있고, 상기 시스템은 어플리케이션 서버 그리드, 스토리지 영역 네트워크 및 인피니밴드(IB) 네트워크를 제공한다. 미들웨어 머신 소프트웨어는 예컨대, WebLogic Server, JRockit 또는 Hotspot JVM, Oracle Linux 또는 Solaris, 및 Oracle VM과 같은 어플리케이션 서버, 미들웨어 및 다른 기능을 제공할 수 있다. 실시예에 따르면, 시스템은 IB 네트워크를 통해 서로와 통신하는 복수의 컴퓨팅 노드들, IB 스위치 게이트웨이 및 스토리지 노드들 또는 유닛들을 포함할 수 있다. 랙 구성으로 구현될 때, 랙의 사용되지 않은 부분들은 비어있는 채로 있거나 필러(filler)들에 의해 채워질 수 있다.

[0010] "Sun Oracle Exalogic" 또는 "Exalogic"으로 본 명세서에서 지칭되는 본 발명의 실시예에 따르면, 시스템은 Oracle Middleware SW suite 또는 Weblogic과 같은 미들웨어 또는 어플리케이션 서버 소프트웨어를 호스팅하기 위한 배치가 용이한 솔루션(easy-to-deploy solution)이다. 본 명세서에 기술된 바와 같이, 실시예에 따르면, 시스템은 하나 이상의 서버들, 스토리지 유닛들, 스토리지 네트워킹을 위한 IB 페브릭 및 미들웨어 어플리케이션을 호스팅하는 데 요구되는 모든 다른 컴포넌트들을 포함하는 "박스 내의 그리드"이다. 상당한 성능이 예컨대, 리얼 어플리케이션 클러스터(Real Application Cluster)들 및 액사로직 오픈 스토리지(Exalogic Open storage)를 이용하여 대량의 병렬 그리드 구조를 활용함으로써 모든 타입의 미들웨어 어플리케이션들에 제공될 수 있다. 시스템은 선형적 I/O 확장성(scalability)을 포함하는 개선된 성능을 제공하고, 사용 및 관리하기에 간단하며, 작업상 중대한(mission-critical) 가용성 및 신뢰성을 제공한다.

[0011] 본 발명의 실시예에 따르면, Tuxedo는 고성능, 분산형 비즈니스 어플리케이션들의 구성, 실행 및 관리(administration)를 할 수 있는 소프트웨어 모듈들의 세트이며, 다수의 단단(multi-tier) 어플리케이션 개발 툴들에 의해 트랜잭션 미들웨어로서 사용되었다. 추가적으로, Tuxedo 시스템과 같은 트랜잭션 미들웨어 시스템은 액사로직 미들웨어 머신과 같은 복수의 프로세서들을 가진 고속 머신들 및 인피니밴드(IB) 네트워크와 같은 고성능 네트워크 연결의 장점을 취할 수 있다.

[0012] 본 발명의 실시예에 따르면, 트랜잭션 미들웨어 시스템은 예컨대, 브릿지 프로세스를 바이패싱하고 단일 포인트 병목(bottleneck)을 방지하는 로컬 메시지 전달과 같은 방식으로 짧은 레이턴시를 달성하도록 RDMA(Remote Direct Memory Access) 프로토콜을 이용하여 로컬 머신과 원격 머신 사이에서 메시지들을 교환할 수 있다. RDMA 프로토콜을 이용하여 로컬 머신과 원격 머신 사이에서 메시지들을 교환하는 것은 2012년 3월 8일자로 출원된 발명의 명칭이 "SYSTEM AND METHOD FOR PREVENTING SINGLE-POINT BOTTLENECK IN A TRANSACTIONAL MIDDLEWARE MACHINE ENVIRONMENT"인 미국 특허 출원 제13/415,760호에서 개시되며, 상기 출원은 본 명세서에 그 전체가 참조로서 포함된다.

커플렉스 메시지 헤더를 지원하는 것

[0014] 본 발명의 실시예에 따르면, 트랜잭션 미들웨어 머신 환경은 머신들 간에 메시지들을 전달할 시 브릿지 프로세스들을 바이패싱하는 것을 지원하고 단일 포인트 병목을 방지하기 위해 커플렉스 메시지 헤더를 사용할 수 있다. 커플렉스 메시지 헤더는 RDMA 메시지 큐 및 시스템 V IPC(Inter-process Communication) 메시지 큐와 같은 트랜잭션 미들웨어 머신 환경에서의 여러 가지 메시지 큐들을 지원하기 위한 플렉서블 데이터 구조를 사용할 수 있다.

[0015] 도 1은 본 발명의 실시예에 따른 커플렉스 메시지 헤더를 지원하는 트랜잭션 미들웨어 머신 환경의 예를 도시한다. 도 1에 도시된 바와 같이, 트랜잭션 클라이언트(101)는 여러 가지 서비스 요청 메시지들, 메시지 A(104) 및 메시지 B(105)를 여러 가지 타입의 트랜잭션 서버들, 서버 A(104) 및 서버 B(105)에 보낼 수 있다. 서버 A는 IPC 큐를 이용하는 제1 타입의 서버일 수 있고, 서버 B는 RDMA 큐를 이용하는 제2 타입의 서버일 수 있다. 추가적으로, 서버는 동시에 IPC 큐 및 RDMA 큐 모두를 청취(listen)할 수 있다. 서버 A(102) 및 서버 B(103)는 하드웨어 컴퓨팅 노드들일 수 있다. 서버 A(102) 및 서버 B(103)는 미들웨어 머신의 랙에 제공될 수 있다.

- [0016] 본 발명의 실시예에 따르면, 컴플렉스 메시지 헤더는 브릿지 프로세스를 바이패싱하는 것을 지원하고 단일 포인트 병목을 방지하기 위해서 플렉서블 데이터 구조를 이용할 수 있다. 도 1에 도시된 바와 같이, 메시지 헤더 A(106)는 제1 데이터 구조 A(108) 및 제2 데이터 구조 A(110)를 포함하고, 메시지 헤더 B(107)는 제1 데이터 구조 B(109) 및 제2 데이터 구조 B(111)를 포함한다. 제1 데이터 구조 A(108) 및 제1 데이터 구조 B(109)의 각각은 IPC 큐를 이용하여 클라이언트에 액세스하기 위한 어드레스 정보를 저장하는 단순한 데이터 구조일 수 있다. 추가적으로, 메시지 헤더 B(107)의 제2 데이터 구조 B(111)는 제2 메시지 큐를 이용하여 클라이언트에 액세스하기 위한 어드레스 정보를 저장하는 데이터 버퍼일 수 있고, 메시지 헤더 A(106)의 제2 데이터 구조 A(110)는 비어있는 채로 있거나, 존재하지 않거나 또는 초기에 생성되지 않을 수 있다.
- [0017] 도 1에 도시된 바와 같은 예에서, 시스템 V IPC 큐 A(122)를 사용하는 것을 선호하는 서버인 서버 A(102)는 브릿지 프로세스 A(120) 및 브릿지 프로세스 C(118)를 통해 IPC 메시지 큐 C(116)를 이용하여 클라이언트에 액세스하기 위한 어드레스 정보를 획득하기 위해 단순한 데이터 구조(108)만 사용할 수 있다. 한편, RDMA 큐를 사용하는 것을 선호하는 서버인 서버 B(103)는 키(113) 예컨대, 부정의 긴 값(negative long value)을 가진 가짜(faked) IPC 큐 어드레스를 획득하기 위해 단순한 데이터 구조(109)를 사용할 수 있다. 더욱이, 서버 B(103)는 제2 메시지 큐를 이용하여 클라이언트에 액세스하기 위한 어드레스 B(114)에 관한 정보를 획득하기 위해 데이터 버퍼를 찾아보기 위해 상기 키(113) 또는 특정 필드 이름을 사용할 수 있다.
- [0018] 본 발명의 실시예에 따르면, 클라이언트는 동시에 IPC 큐 및 RDMA 큐를 유지할 수 있다. 클라이언트가 타겟 서버에 서비스 요청 메시지를 보내기 전에, 클라이언트는 상기 클라이언트가 이전에 수신한 정보에 근거하여 타겟 서버의 탑입을 먼저 결정할 수 있다. 타겟 서버가 IPC 큐를 사용하는 것을 선호한다고 결정되면, 클라이언트는 도 1에 도시된 바와 같은 메시지 헤더 A(106)와 유사한 포맷의 메시지 헤더를 가진 서비스 요청 메시지를 보낼 수 있다. 이와는 달리, 타겟 서버가 RDMA 큐를 사용하는 것을 선호한다고 결정되면, 클라이언트는 도 1에 도시된 바와 같은 메시지 헤더 B(107)와 유사한 포맷의 메시지 헤더를 가진 서비스 요청 메시지를 보낼 수 있다.
- [0019] 도 2는 본 발명의 실시예에 따른 트랜잭션 미들웨어 머신 환경에서 컴플렉스 메시지 헤더를 지원하기 위한 예시적인 흐름도를 도시한다. 도 2에 도시된 바와 같이, 단계(201)에서, 제1 데이터 구조가 제1 메시지 큐를 이용하여 트랜잭션 클라이언트에 액세스하기 위한 어드레스 정보를 저장하도록 제공된다. 또한, 단계(202)에서, 제2 데이터 구조가 제2 메시지 큐를 이용하여 트랜잭션 클라이언트에 액세스하기 위한 어드레스 정보를 저장하도록 제공된다. 그 다음, 단계(203)에서, 제1 탑입의 트랜잭션 클라이언트가 제1 메시지 큐를 이용하여 트랜잭션 클라이언트와 통신하기 위한 어드레스 정보를 획득하기 위해 컴플렉스 메시지 헤더 내의 제1 데이터 구조를 체크할 수 있다. 또한, 단계(204)에서, 제2 탑입의 트랜잭션 클라이언트가 컴플렉스 메시지 헤더 내의 제1 데이터 구조로부터 키를 획득하고, 제2 메시지 큐를 이용하여 트랜잭션 클라이언트와 통신하기 위해 컴플렉스 메시지 헤더 내의 제2 데이터 구조로부터 어드레스 정보를 획득하기 위해 상기 키를 사용할 수 있다.
- [0020] Tuxedo로 메시지 큐 정보를 운반(carrying)
- [0021] 본 발명의 실시예에 따르면, Tuxedo 어플리케이션은 RDMA 프로토콜의 장점을 취하며, 시스템 V IPC 큐들과 관련된 브릿지 프로세스들을 바이패싱하기 위해 RDMA 큐들을 이용할 수 있다. 시스템 V IPC 큐들과 관련된 브릿지 프로세스들은 Tuxedo 환경에서의 단일 포인트 병목이 될 수 있다. RDMA 큐들을 이용하여, 서로 다른 Tuxedo 머신들 간의 원격 메시지 전달이 로컬 메시지 전달과 유사한 방식으로 짧은 레이턴시를 달성할 수 있다.
- [0022] Tuxedo 클라이언트 및 서버는 여러 가지 버전들로 컴파일될 수 있고, 여러 가지 머신들에 배치될 수 있다. 모든 머신이 바이패스 브릿지 프로세스 기능을 지원하지 않을 가능성이 있다. 시스템은 이 머신들 간의 메시지 호환성을 유지할 수 있다. 예를 들어, 일 버전의 Tuxedo 서버는 상기 Tuxedo 서버가 바이패스 브릿지 특징을 지원하지 않더라도, Tuxedo 메시지 헤더의 모든 필드들의 정확한 오프셋(offset)들에 액세스할 수 있다. 따라서, 여러 가지 버전들로부터의 Tuxedo 상에서 실행되는 어플리케이션들은 서로로부터 보내진 메시지들을 인지할 수 있다.
- [0023] 도 3은 본 발명의 실시예에 따른 트랜잭션 미들웨어 머신 환경에서의 컴플렉스 메시지 헤더의 예를 도시한다. 도 3에 도시된 바와 같이, 컴플렉스 메시지 헤더는 C-구조들(301 및 302) 내의 여러 Tuxedo 메시지 헤더들 및 FML32 탑입의 버퍼(FML32 typed buffer)(303) 내의 Tuxedo 메시지 헤더를 포함한다.
- [0024] 클라이언트는 C-구조들에 시스템 V IPC 큐 어드레스 정보를 저장할 수 있다. 상호 운용성 문제를 해결하기 위해서, C-구조들은 바이패스 브릿지 특징이 활성화될 때에도 메시지 헤더 내의 동일한 위치에서 변경되지 않은 채로 유지될 수 있다. 추가적으로, 클라이언트는 자신의 전체 RDMA 메시지 큐 정보를 시스템 V IPC 큐 어드레스에 대해 사용되지 않은 ID들을 가진 플렉서블 트랜잭션 헤더 예컨대, FML32 탑입의 버퍼 헤더(303)에 저장할 수

있다.

- [0025] FML32 타입의 버퍼 내의 RDMA 메시지 큐 정보는 일반적으로, C-구조들 내의 시스템 V IPC 큐 어드레스 정보보다 많은 바이트들을 포함한다. 예를 들어, Tuxedo 요청 메시지 헤더의 C-구조들은 8-바이트의 "긴" 변수(variable)를 포함하지만, RDMA 메시지 큐 어드레스는 128 바이트의 어레이이다.
- [0026] FML32 타입의 버퍼는 필드-색인(field-indexed) 플렉서블 트랜잭션 네이터 구조이다. FML32 타입의 버퍼 내의 모든 입력은 특정 필드 이름에 의해 검색될 수 있다. 프로세스가 FML32 타입의 버퍼를 얻으면, 상기 프로세스는 상기 프로세스가 모르는 필드 이름을 통해 값을 얻지 못하는 경우의 어떤 문제를 야기하지 않는다. 더욱이, 어떤 프로세스가 자신이 모르는 필드를 통해 값을 얻도록 시도할 수 있는 가능성은 낫다.
- [0027] 본 발명의 실시예에 따르면, 동적인 수의 필드들(304 및 305)은 특정 ID들을 통해 액세스 가능한 동일한 버퍼 내로 투입될 수 있다. 바이패스 브릿지 특징이 활성화되지 않을 때, C-구조 헤더가 변경되지 않기 때문에, 프로세스는 FML32 헤더를 조사하지 않고, 상기 프로세스는 시스템 V IPC 큐로의 액세스만을 가진다.
- [0028] 바이패스 브릿지 특징이 활성화될 때, 클라이언트는 서버에 메시지들을 보내기 위해 시스템 V IPC 큐 대신 RDMA 메시지 큐를 사용할 수 있고, 요청 메시지에 자신의 딥신 RDMA 메시지 큐 어드레스를 저장할 수 있어서, 서버는 다시 응답을 보낼 수 있다.
- [0029] 도 3에 도시된 바와 같이, Tuxedo 서버 프로세스가 Tuxedo 클라이언트로부터 메시지를 수신할 때, Tuxedo 서버 프로세스는 먼저, C-구조들 내의 IPC 큐 어드레스 예컨대, Tuxedo 메시지 헤더 X(302)를 얻을 수 있다. 만일 Tuxedo 서버 프로세스가 입력이 부정의 긴 값을 가진다는 것을 발견하면, Tuxedo 서버 프로세스는 FML32 타입의 버퍼에 있는 Tuxedo 메시지 헤더 Y(303)로부터 특정 필드 이름을 통해 RDMA 메시지 큐 어드레스를 얻을 수 있다.
- [0030] 더욱이, Tuxedo 클라이언트는 요청을 보내기 전에 서버로부터 버전 정보를 얻을 수 있다. 따라서, 바이패스 브릿지 특징 없이 컴파일된 서버들의 경우, 오직 시스템 V IPC 큐 어드레스들만이 요청 메시지 헤더 내로 투입될 수 있고, FML32 타입의 버퍼는 터치되지 않는다.
- [0031] 바이패스 브릿지 특징이 동작불가능할 때, FML32 버퍼(303)에 어떤 RDMA 메시지 큐 어드레스 필드도 존재하지 않는다. 메시지 버퍼(C-구조 파트) 내의 큐 어드레스는 긍정(positive)의 IPC 큐 어드레스일 수 있고, 상기 긍정의 IPC 큐 어드레스는 시스템 V IPC 큐 어드레스를 나타낸다. 따라서, Tuxedo가 메시지 버퍼로부터 이 큐 어드레스를 얻을 때, 시스템은 FML32 버퍼를 조사하지 않는다.
- [0032] 본 발명의 다른 실시예들은 트랜잭션 네이터 미들웨어 머신 환경에서 컴플렉스 메시지 헤더를 지원하기 위한 시스템을 포함하고, 상기 시스템은 제1 메시지 큐를 이용하여 트랜잭션 네이터 클라이언트에 액세스하기 위한 어드레스 정보를 저장하는 제1 데이터 저장 유닛과, 제2 메시지 큐를 이용하여 상기 트랜잭션 네이터 클라이언트에 액세스하기 위한 어드레스 정보를 저장하는 제2 데이터 저장 유닛과, 상기 제1 메시지 큐를 이용하여 상기 트랜잭션 네이터 클라이언트와 통신하기 위한 상기 어드레스 정보를 획득하기 위해 상기 컴플렉스 메시지 헤더 내의 제1 데이터 구조만을 체크하는 동작을 하는 제1 타입의 트랜잭션 네이터 서버와 그리고 상기 컴플렉스 메시지 헤더 내의 상기 제1 데이터 구조로부터 키를 획득하고, 상기 제2 메시지 큐를 이용하여 상기 트랜잭션 네이터 클라이언트와 통신하기 위해 상기 컴플렉스 메시지 헤더 내의 제2 데이터 구조로부터 어드레스 정보를 획득하기 위해 상기 키를 이용하는 동작을 하는 제2 타입의 트랜잭션 네이터 서버를 포함한다.
- [0033] 다른 실시예는 시스템을 포함하고, 여기서 상기 트랜잭션 네이터 클라이언트는 상기 컴플렉스 메시지 헤더를 이용하여 트랜잭션 네이터 서버에 메시지를 보내는 동작을 한다.
- [0034] 다른 실시예는 시스템을 포함하고, 여기서 상기 트랜잭션 네이터 클라이언트는 상기 메시지를 보내기 전에 상기 트랜잭션 네이터 서버 상에서 정보를 획득하는 동작을 한다.
- [0035] 다른 실시예는 시스템을 포함하고, 여기서 상기 트랜잭션 네이터 서버는 상기 컴플렉스 메시지 헤더 내의 상기 어드레스 정보에 근거하여 상기 트랜잭션 네이터 클라이언트에 응답 메시지를 보내는 동작을 한다.
- [0036] 다른 실시예는 시스템을 포함하고, 여기서 상기 제1 데이터 구조는 C 데이터 구조이고, 상기 제2 데이터 구조는 타입화된 버퍼 데이터 구조(typed buffer data structure)이다.
- [0037] 다른 실시예는 시스템을 포함하고, 여기서 상기 트랜잭션 네이터 클라이언트는 로컬 브릿지 프로세스를 통해 제1 타입의 트랜잭션 네이터 서버와 통신하는 동작을 한다.

- [0038] 다른 실시예는 시스템을 포함하고, 여기서 상기 트랜잭션 클라이언트는 고성능 네트워크를 통해 바로 제2 타입의 트랜잭션 서버와 통신하는 동작을 한다.
- [0039] 다른 실시예는 시스템을 포함하고, 여기서 상기 고성능 네트워크는 DRMA(remote direct memory access) 프로토콜을 사용하는 인피니밴드(IB) 네트워크이다.
- [0040] 다른 실시예는 청구항 1항의 시스템을 포함하고, 여기서 제1 타입의 트랜잭션 서버는 컴플렉스 메시지 헤더의 제2 데이터 구조 내의 어드레스 정보를 이해하지 못한다.
- [0041] 다른 실시예는 시스템을 포함하고, 여기서 상기 제2 데이터 구조는 동적인 수의 필드들을 가진 플렉서블 트랜잭션 데이터 구조이고, 상기 필드들 각각은 특정 ID를 통해 액세스가능하다.
- [0042] 다른 실시예는 트랜잭션 미들웨어 머신 환경에서 컴플렉스 메시지 헤더를 지원하기 위한 장치를 포함하고, 상기 장치는 제1 메시지 큐를 이용하여 트랜잭션 클라이언트에 액세스하기 위한 어드레스 정보를 저장하는 제1 데이터 구조를 제공하기 위한 수단과, 제2 메시지 큐를 이용하여 상기 트랜잭션 클라이언트에 액세스하기 위한 어드레스 정보를 저장하는 제2 데이터 저장 구조를 제공하기 위한 수단과, 제1 타입의 트랜잭션 서버로 하여금 상기 제1 메시지 큐를 이용하여 상기 트랜잭션 클라이언트와 통신하기 위한 상기 어드레스 정보를 획득하기 위해 상기 컴플렉스 메시지 헤더 내의 상기 제1 데이터 구조만을 체크하도록 하기 위한 수단과, 그리고 제2 타입의 트랜잭션 서버로 하여금 상기 컴플렉스 메시지 헤더 내의 상기 제1 데이터 구조로부터 키를 획득하고, 상기 제2 메시지 큐를 이용하여 상기 트랜잭션 클라이언트와 통신하기 위해 상기 컴플렉스 메시지 헤더 내의 상기 제2 데이터 구조로부터 어드레스 정보를 획득하기 위해 상기 키를 이용하도록 하기 위한 수단을 포함한다.
- [0043] 다른 실시예는 상기 트랜잭션 클라이언트로 하여금 상기 컴플렉스 메시지 헤더를 이용하여 트랜잭션 서버에 메시지를 보내도록 하기 위한 수단을 더 포함하는 장치를 포함한다.
- [0044] 다른 실시예는 상기 트랜잭션 클라이언트로 하여금 상기 메시지를 보내기 전에 상기 트랜잭션 서버 상에서 정보를 얻도록 하기 위한 수단을 더 포함하는 장치를 포함한다.
- [0045] 다른 실시예는 상기 트랜잭션 서버로 하여금 상기 컴플렉스 메시지 헤더 내의 상기 어드레스 정보에 근거하여 상기 트랜잭션 클라이언트에 응답 메시지를 보내도록 하기 위한 수단을 더 포함하는 장치를 포함한다.
- [0046] 다른 실시예는 장치를 포함하고, 여기서 상기 제1 데이터 구조는 C 데이터 구조이고, 상기 제2 데이터 구조는 타입화된 베퍼 데이터 구조이다.
- [0047] 다른 실시예는 상기 트랜잭션 클라이언트로 하여금 로컬 브릿지 프로세스를 통해 제1 타입의 트랜잭션 서버와 통신하도록 하기 위한 수단을 더 포함하는 장치를 포함한다.
- [0048] 다른 실시예는 상기 트랜잭션 클라이언트로 하여금 고성능 네트워크를 통해 바로 제2 타입의 트랜잭션 서버와 통신하도록 하기 위한 수단을 더 포함하는 장치를 포함한다.
- [0049] 다른 실시예는 장치를 포함하고, 여기서 상기 고성능 네트워크는 DRMA 프로토콜을 사용하는 인피니밴드(IB) 네트워크이다.
- [0050] 다른 실시예는 제1 타입의 트랜잭션 서버로 하여금 상기 컴플렉스 메시지 헤더의 제2 데이터 구조 내의 어드레스 정보를 이해하지 못하게끔 하는 수단을 더 포함하는 장치를 포함한다.
- [0051] 다른 실시예는 장치를 포함하고, 여기서 상기 제2 데이터 구조는 동적인 수의 필드들을 가진 플렉서블 트랜잭션 데이터 구조이고, 상기 필드들의 각각은 특정 ID를 통해 액세스가능하다.
- [0052] 본 발명은 본 발명의 교시들에 따라 프로그램된 하나 이상의 프로세서들, 메모리 및/또는 컴퓨터 판독가능 스토리지 매체를 포함하는 하나 이상의 종래의 범용 또는 특수 디지털 컴퓨터, 컴퓨팅 디바이스, 머신 또는 마이크로프로세서를 이용하여 통상적으로 구현될 수 있다. 적절한 소프트웨어 코딩은 소프트웨어 기술 분야의 숙련자들에게 분명할 바와 같이, 본 발명의 교시들에 근거하여 숙련된 프로그래머들에 의해 쉽게 준비될 수 있다.
- [0053] 일부 실시예들에서, 본 발명은 본 발명의 프로세스들 중 어느 것을 수행하도록 컴퓨터를 프로그램하는 데 사용될 수 있는 명령어들이 저장된 스토리지 매체 또는 컴퓨터 판독가능 매체(들)인 컴퓨터 프로그램 물을 포함한다. 스토리지 매체는 (이들로만 한정되는 것은 아니지만) 플로피 디스크(disk)들, 광학 디스크(disc)들, DVD, CD-ROM들, 마이크로드라이브 및 자기-광학 디스크(disk)들을 포함하는 어떤 타입의 디스크, ROM들, RAM들,

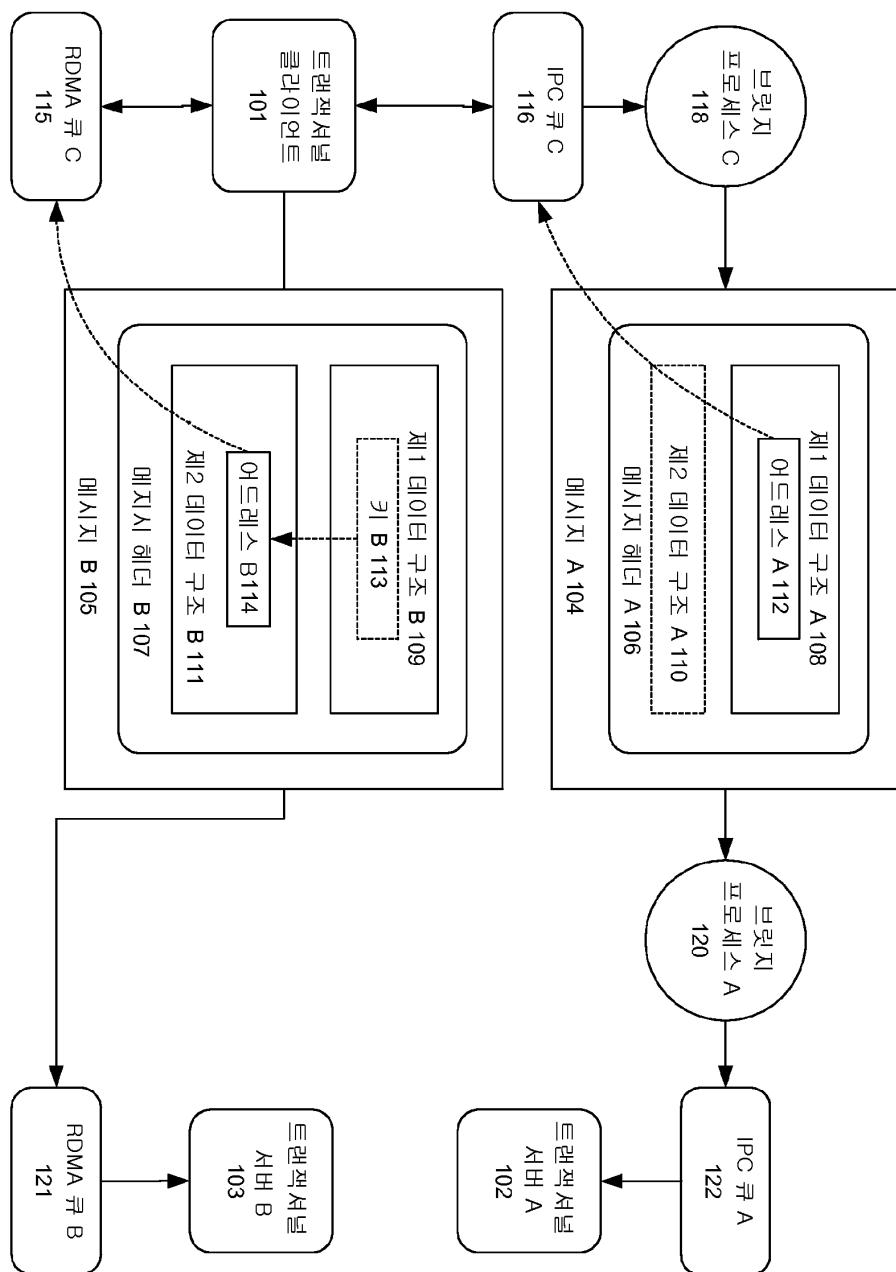
EPROM들, EEPROM들, DRAM들, VRAM들, 플래시 메모리 디바이스들, 자기 또는 광학 카드들, (분자 메모리 IC들을 포함하는)나노시스템들 또는, 명령어들 및/또는 데이터를 저장하기에 적절한 어떤 타입의 매체 또는 디바이스를 포함할 수 있다.

[0054]

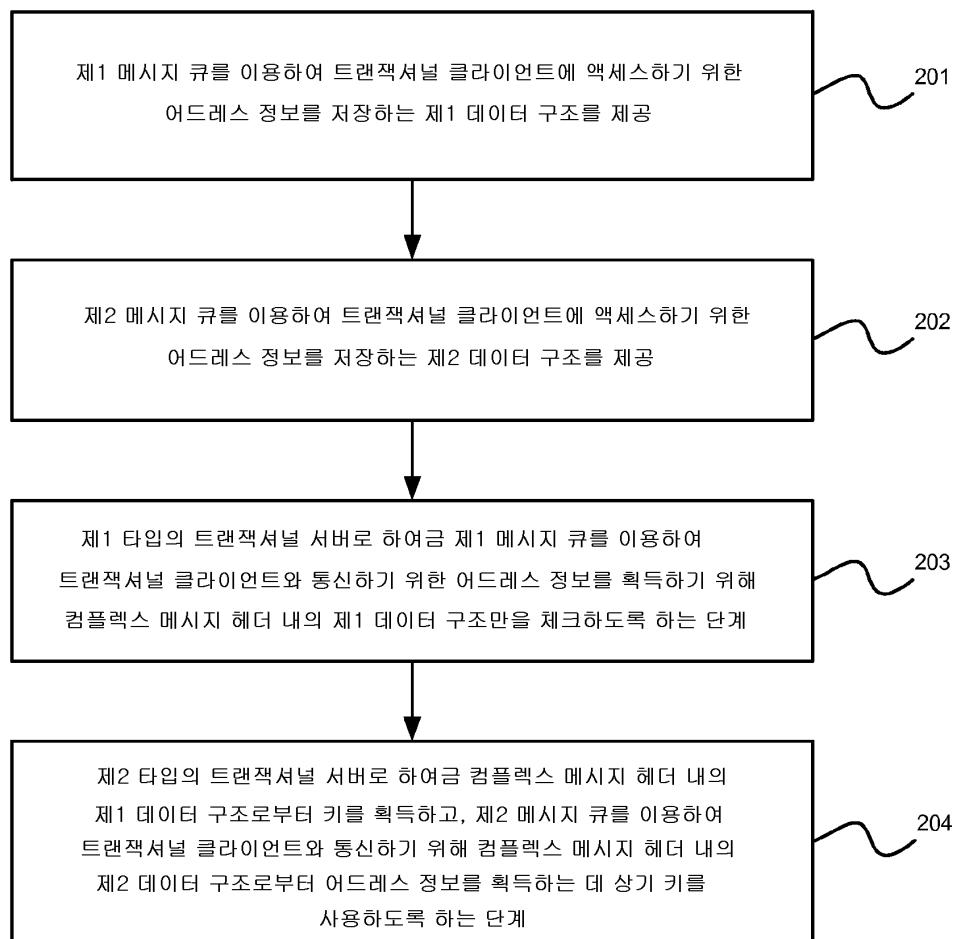
본 발명의 상기 상세한 설명은 예시 및 설명을 위해 제공되었다. 본 설명은 완전한 것이거나 또는 정확히 개시된 형태들로만 본 발명을 제한하고자 의도된 것이 아니다. 많은 수정들 및 변형들이 이 기술분야의 숙련자에게 분명할 것이다. 위 실시예들은 본 발명의 원리 및 이의 실용적 응용을 가장 잘 설명하기 위해 선택 및 기술되었으며, 그럼으로써 이 기술분야의 숙련자들은 본 발명에 대한 다양한 실시예들 및 고려되는 특별한 사용에 적합한 다양한 수정들을 이해할 수 있다. 본 발명의 범위는 다음의 특히 청구 범위 및 이의 균등물에 의해 한정되어야 함이 의도된다.

도면

도면1



도면2



도면3

