

	(19) 대한민국특허청(KR) (12) 공개특허공보(A)	(11) 공개번호 10-2014-0033571 (43) 공개일자 2014년03월19일
(51) 국제특허분류(Int. Cl.) G06F 15/16 (2006.01) G06Q 50/22 (2012.01) (21) 출원번호 10-2012-0097341 (22) 출원일자 2012년09월03일 심사청구일자 2012년09월03일	(71) 출원인 경희대학교 산학협력단 경기도 용인시 기흥구 덕영대로 1732, 국제캠퍼스 내 (서천동, 경희대학교) (72) 발명자 이승룡 경기 성남시 분당구 구미로144번길 25, 102동 102호 (구미동, 삼환빌라) 무하마드 비탈 아민 경기 용인시 기흥구 덕영대로 1732, 전자정보대학 351호 (서천동, 경희대학교) (74) 대리인 특허법인 신지	

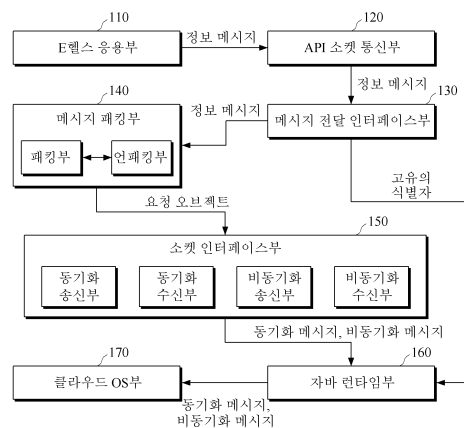
전체 청구항 수 : 총 5 항

(54) 발명의 명칭 헬스 클라우드를 위한 고성능 자바 소켓

(57) 요약

본 발명에 따른 헬스 클라우드를 위한 고성능 자바 소켓은 사용자로부터 수집된 신체 정보를 포함하는 정보 메시지를 전달하는 e헬스 응용부, 수신된 정보 메시지를 전달하고, e헬스 응용부와 운영체제 사이의 통신을 연결하는 API 소켓 통신부, 수신된 정보 메시지를 전달하고, 모든 프로세스를 초기화하며, 각각의 프로세스에 할당된 고유한 식별자를 공유하는 메시지 전달 인터페이스부, 수신된 정보 메시지를 패킹(Packing)하거나 언패킹(Unpacking)하며, 수신된 정보 메시지에 기초하여 요청 오브젝트 및 응답 오브젝트를 생성하여 전달하는 메시지 패킹부, 수신된 요청 오브젝트 및 응답 오브젝트를 동기화 또는 비동기화하여 전달하는 소켓 인터페이스부를 포함한다.

대표도 - 도1



이 발명을 지원한 국가연구개발사업

과제고유번호	1415123336
부처명	지식경제부
연구사업명	대학 IT 연구센터 육성지원
연구과제명	동서신의학 u-라이프케어연구센터
기 여 율	1/2
주관기관	경희대학교산학협력단
연구기간	2012.01.01 ~ 2012.12.31이 발명을 지원한 국가연구개발사업
과제고유번호	2011-0030823
부처명	교육과학기술부
연구사업명	선도연구센터지원사업
연구과제명	디지털로그휴먼미디어 감성인지 기술연구개발
기 여 율	1/2
주관기관	경희대학교산학협력단
연구기간	2012.09.01 ~ 2013.08.31

특허청구의 범위

청구항 1

사용자로부터 수집된 신체 정보를 포함하는 정보 메시지를 전달하는 e헬스 응용부;

수신된 상기 정보 메시지를 전달하고, 상기 e헬스 응용부와 운영체제 사이의 통신을 연결하는 API 소켓 통신부;

수신된 상기 정보 메시지를 전달하고, 모든 프로세스를 초기화하며, 각각의 프로세스에 할당된 고유한 식별자를 공유하는 메시지 전달 인터페이스부;

수신된 정보 메시지를 패킹(Packing)하거나 언패킹(Unpacking)하며, 수신된 상기 정보 메시지에 기초하여 요청 오브젝트 및 응답 오브젝트를 생성하여 전달하는 메시지 패킹부;

수신된 상기 요청 오브젝트 및 상기 응답 오브젝트를 동기화 또는 비동기화하여 전달하는 소켓 인터페이스부;

수신된 상기 동기화 메시지 또는 상기 비동기화 메시지를 스트리밍(Streaming)으로 전달하는 자라 런타임부; 및

수신된 상기 동기화 메시지 또는 상기 비동기화 메시지를 다른 컴퓨터 단말과 공유하며, 헬스 클라우드 서버를 작동시키는 운영체제를 포함하는 클라우드 OS부;

를 포함하는 것을 특징으로 하는 헬스 클라우드를 위한 고성능 자바 소켓.

청구항 2

제 1항에 있어서,

상기 API 소켓 통신부는,

연산 과정 및 데이터 복사 과정에서 메모리 오버헤드(Memory Overhead)를 줄이기 위해 버퍼(Buffer) 및 원시 자료형(Primitive Data Type)을 지원하지 않는 것을 특징으로 하는 헬스 클라우드를 위한 고성능 자바 소켓.

청구항 3

제 1항에 있어서,

상기 요청 오브젝트 및 상기 응답 오브젝트는 헤더(Header)와 페이로드(Payload)의 두 파트로 나뉘어진 메시지를 포함하고, 상기 헤더는 고정된 용량을 가지며, 메시지에 의존하여 동적인 용량 변화를 가지는 상기 페이로드에 대한 정보를 포함하는 것을 특징으로 하는 헬스 클라우드를 위한 고성능 자바 소켓.

청구항 4

제 3항에 있어서,

상기 소켓 인터페이스부는 상기 메시지 정보를 한번에 읽어 수신하며, 먼저 상기 헤더의 용량 정보를 읽고, 다음으로 상기 헤더의 바이트로부터 상기 페이로드의 용량 및 종류를 식별하는 것을 특징으로 하는 헬스 클라우드를 위한 고성능 자바 소켓.

청구항 5

제 1항에 있어서,

상기 소켓 인터페이스부는,

동기화에 의해 유지되는 다수의 작업 스레드(Worker Thread)로 구성된 스레드풀(Thread Pool)을 포함하고, 상기 작업 스레드는 상기 정보 메시지의 용량 및 종류를 판단하며, 작업 처리 후 캐시(Cache)에 저장되는 것을 특징으로 하는 헬스 클라우드를 위한 고성능 자바 소켓.

명세서

기술 분야

[0001] 본 발명은 헬스 클라우드(Health Cloud)에서 고성능 연산을 분산 처리 가능하도록 하기 위한 통신 방식을 개선하는 장치 및 방법에 관한 것이다.

배경 기술

[0002] 네트워크의 발달에 따라 IT 자원의 효율적 분배를 목적으로 클라우드 컴퓨팅(Cloud Computing) 기술이 빠르게 발전하고 있다. 클라우드 컴퓨팅은 IT와 관련된 기능들이 서비스 형태로 제공되는 컴퓨팅 스타일이다. 각 단말의 독립적인 하드웨어 성능에 의존하던 기존의 컴퓨팅 환경과 달리 서로 다른 물리적인 위치에 존재하는 컴퓨터들의 리소스를 가상화 기술로 통합해 제공하는 기술을 말한다. 최근 통신 인프라의 급속한 발달에 힘입어 컴퓨팅 환경의 자원을 효율적으로 분배하고자 하는 요구가 증대함에 따라서 발전 및 개발 속도가 증가하고 있다. 클라우드 컴퓨팅은 네트워크를 이용해 클라우드 망의 컴퓨터 자원을 공유하여 사용할 수 있기 때문에, 상대적으로 저성능의 컴퓨터로 고성능의 처리를 가능하게 해준다.

[0003] 생물의학(Biomedical), 건강관리(Healthcare) 데이터 수집, 유비쿼터스 건강 산업 등과 같은 건강 관련 산업들에 있어서, 클라우드 컴퓨팅 기반의 e-헬스 시스템 또는 헬스 클라우드(Health Cloud)가 개발되고 있다. 헬스 클라우드는 사용자의 건강 정보를 수집하고 이러한 정보를 클라우드 서버에서 저장 및 분석하여 사용자에게 건강관리 서비스를 제공하는 시스템이다. 건강 관련 산업들은 사용자에게 대한 다양한 정보를 수집하고, 사용자의 건강 상태 또는 신체 상태를 파악할 수 있는 CT나 MRI 이미지와 같은 이미지 데이터를 수집한다. 또한 이러한 수집된 데이터를 분석함으로써 사용자의 건강 상태를 추론하고 적절한 권고안을 제공한다. 메디컬 이미지나 유전 및 단백질 주석과 같은 서비스를 제공하고 추론된 결과를 제공하는 임상적 의사결정지원 시스템(Clinical Decision Support System) 및 의학 치료 지원 장치 등은 고성능의 컴퓨터를 필요로 한다. 예를 들어, CT나 MRI 촬영 이미지 등을 정확하게 분석하기 위해서는 고성능의 컴퓨터를 필요로 한다. 하지만 이러한 고성능의 컴퓨터는 매우 고가이기 때문에, 학교나 작은 규모의 연구실 및 개인 연구가는 충분한 성능의 컴퓨터를 사용하기 어렵다.

[0004] 그러나 클라우드 컴퓨팅 기반의 모델은 이러한 문제점을 해결할 수 있다. 클라우드 컴퓨팅을 이용하면, 개인 연구가도 발전된 과학 어플리케이션 및 시뮬레이션을 처리할 수 있다. 현재 마이크로 소프트의 아저 플랫폼(Azure Platform)과 같은 공공 클라우드 제공업체들은 과학적 어플리케이션을 필요에 의해 그들의 시설을 제공한다. 심지어 오픈 누블라(Open-Nubula)와 같은 오픈 소스 클라우드 플랫폼에서 작동하는 소규모의 개인 클라우드를 이용하여 제한된 규모의 고성능 연산(High Performance Computing)을 제공할 수 있다.

[0005] 하지만 현재 일반적으로 클라우드 컴퓨팅에서 클라우드 서버를 구성하는 다수의 컴퓨터 단말 사이에 연산을 분산하여 처리하기 위한 통신은 막대한 클라우드 컴퓨팅 연산을 미처 따라가지 못하고 있다. 현재 클라우드 서버 내의 단말 사이의 내부 통신언어는 C나 포트란과 같은 과거에 개발되어 사용되는 언어가 사용되고 있다. 하지만 이러한 언어들의 경우 고성능 연산 기반의 병렬 연산에 있어서, 사용 효율이나 네트워크의 오버헤드(Overhead) 등에 문제를 드러내고 있다.

발명의 내용

해결하려는 과제

[0006] 본 발명에서 해결하고자 하는 클라우드 컴퓨팅을 이용하여 건강관리 서비스를 제공하는 헬스 클라우드에 있어서, 다수의 컴퓨터 단말 사이의 데이터를 효과적으로 공유하여 분산 처리하기 위한 고성능 자바 소켓을 제공하는 것이다.

과제의 해결 수단

[0007] 본 발명에 따른 헬스 클라우드를 위한 고성능 자바 소켓은 사용자로부터 수집된 신체 정보를 포함하는 정보 메시지를 전달하는 e헬스 응용부, 수신된 정보 메시지를 전달하고, e헬스 응용부와 운영체제 사이의 통신을 연결하는 API 소켓 통신부, 수신된 정보 메시지를 전달하고, 모든 프로세스를 초기화하며, 각각의 프로세스에 할당된 고유한 식별자를 공유하는 메시지 전달 인터페이스부, 수신된 정보 메시지를 패킹(Packing)하거나 언패킹(Unpacking)하며, 수신된 정보 메시지에 기초하여 요청 오브젝트 및 응답 오브젝트를 생성하여 전달하는 메시지 패킹부, 수신된 요청 오브젝트 및 응답 오브젝트를 동기화 또는 비동기화하여 전달하는 소켓 인터페이스부, 수

신된 동기화 메시지 또는 비동기화 메시지를 스트리밍(Streaming)으로 전달하는 자바 런타임부 및 수신된 동기화 메시지 또는 비동기화 메시지를 다른 컴퓨터 단말과 공유하며, 헬스 클라우드 서버를 작동시키는 운영체제를 포함하는 클라우드 OS부를 포함한다.

[0008] API 소켓 통신부는 연산 과정 및 데이터 복사 과정에서 메모리 오버헤드(Memory Overhead)를 줄이기 위해 버퍼(Buffer) 및 원시 자료형(Primitive Data Type)을 지원하지 않는다. 그리고 요청 오브젝트 및 상기 응답 오브젝트는 헤더(Header)와 페이로드(Payload)의 두 파트로 나뉘어진 메시지를 포함하고, 헤더는 고정된 용량을 가지며, 메시지에 의존하여 동적인 용량 변화를 가지는 페이로드에 대한 정보를 포함한다.

[0009] 소켓 인터페이스부는 메시지 정보를 한번에 읽어 수신하며, 먼저 헤더의 용량 정보를 읽고, 다음으로 헤더의 바이트로부터 페이로드의 용량 및 종류를 식별한다. 그리고 동기화에 의해 유지되는 다수의 작업 스레드(Worker Thread)로 구성된 스레드풀(Thread Pool)을 포함하고, 작업 스레드는 상기 정보 메시지의 용량 및 종류를 판단하며, 작업 처리 후 캐시(Cache)에 저장된다.

발명의 효과

[0010] 본 발명에 따른 헬스 클라우드를 위한 고성능 자바 소켓을 적용하면, 고가의 단말을 이용하지 않고도, 많은 양의 연산을 효과적으로 처리할 수 있으며, 클라우드 컴퓨팅의 분산 처리 과정에서 각 단말 사이의 고속 통신이 가능하기 때문에 더욱 빠른 연산을 할 수 있다.

도면의 간단한 설명

[0011] 도 1은 본 발명에 따른 헬스 클라우드를 위한 고성능 자바 소켓을 나타내는 구성도이다.

발명을 실시하기 위한 구체적인 내용

[0012] 이하, 첨부된 도면들을 참조하여 본 발명의 실시예를 상세하게 설명한다. 본 명세서에서 사용되는 용어는 실시예에서의 기능 및 효과를 고려하여 선택된 용어들로서, 그 용어의 의미는 사용자 또는 운용자의 의도 또는 업계의 관례 등에 따라 달라질 수 있다. 따라서 후술하는 실시예들에서 사용된 용어의 의미는, 본 명세서에 구체적으로 명시된 경우에는 명시된 정의에 따르며, 구체적으로 명시하지 않는 경우, 당업자들이 일반적으로 인식하는 의미로 해석되어야 할 것이다.

[0013] 자바(Java) 기반의 고성능 메시지 전송 방법은 MPJ-Express라 불리는 라이브러리에 의해 제공된다. MPJ-Express는 메시지 전달을 위한 도구 기반의 자바 NIO(Non Blocking Input Output)를 제공한다. 자바 NIO는 블록화를 하지 않고 입출력 작업을 수행하는 방법이다. 그러나 버퍼링 층(Buffer Layer)에 의존하는 NIO는 메시지 송수신을 위한 백버퍼(Back Buffer) 및 포스버퍼(Forth Buffer)에 대응하는 메시지의 바이트 복사에 의한 계산이 부하로 작용한다. 본 발명에 따른 헬스 클라우드를 위한 고성능 자바 소켓(High Performance JAVA Socket, 이하 HPJS라 칭함)은 더 나은 연산 성능과 메모리 활용을 제공하는 싱글 카피 프로토콜을 따르는 컴퍼넌트(Component)와 소켓 사이의 직접적인 바이트 조절에 의한 부하에 직면한다.

[0014] 본 발명에 따른 HPJS는 자바 소켓 도구를 이용한 고성능 퍼포먼스를 지원하는 프로세스 라이브러리(Library)이다. HPJS는 클라우드 플랫폼을 위한 더 나은 리소스 사용 효율, 최적화된 연산 및 네트워크 오버헤드(Overhead)의 감소를 제공한다.

[0015] 도 1은 본 발명에 따른 헬스 클라우드를 위한 고성능 자바 소켓을 나타내는 구성도이다.

[0016] 도 1을 참조하면 본 발명에 따른 헬스 클라우드를 위한 고성능 자바 소켓은 e헬스 응용부(110), API 소켓 통신부(120), 메시지 전달 인터페이스부(130), 메시지 패킹부(140), 소켓 인터페이스부(150), 자바 런타임부(160) 및 클라우드 OS부(170)을 포함한다.

[0017] e헬스 응용부(110)는 사용자로부터 사용자의 개인 정보 및 신체 정보를 수집하고, 수집된 정보 메시지를 API 소켓 통신부(120)로 전달한다. e헬스 응용부(110)로부터 전달되는 메시지나 e헬스 응용부(110)로 수신되는 메시지는 바이트 형식으로 메시지를 송수신한다.

[0018] API 소켓 통신부(120)는 e헬스 응용부(110)로부터 수신된 정보 메시지가 메시지 인터페이스부(130)와 e헬스 응용부(110) 사이를 왕복하도록 조절한다. API(Application Programming Interface)는 운영체제와 응용프로그램

사이의 통신에 사용되는 메시지 형식이다. API 소켓 통신부(120)는 e헬스 응용부(110)와 클라우드 OS부(170)사이의 통신에 사용된다.

- [0019] API 소켓 통신부(120)는 연산과정과 데이터 복사 과정에서 메모리 오버헤드를 줄이기 위해 버퍼(buffer) 및 원시 자료형(Primitive Data Type)은 지원하지 않는다. 원시 자료형은 자료 대상 등을 생성하고 조작하는데 필요한 연산들의 집합을 가진 자료 대상의 종류를 자료형이라고 하는데, 이들 중 가장 기본이 되는 자료형을 말한다.
- [0020] 메시지 전달 인터페이스부(130)는 P2P(Peer To peer) 기반의 메시지 전달 인터페이스(Message Passing Interface, MPI)를 포함하며, 클라우드 서버에 포함된 다수의 컴퓨터 단말 사이의 P2P 통신을 위한 MPI이다. 메시지 전달 인터페이스는 분산 및 병렬 처리에서 정보의 교환에 대해 기술하는 표준이다. 그리고 메시지 전달 인터페이스부(130)는 API 소켓 통신부(120)로부터 수신된 정보 메시지를 메시지 패킹부(140)로 전달한다.
- [0021] 메시지 전달 인터페이스부(130)는 본 발명에 따른 HPJS의 프로세스 전체에서 읽기 초기화(Initialization Read)를 수행한다. 이러한 초기화 과정에서 모든 관련된 프로세스들은 그들의 고유한 식별자(Unique Identifier)를 공유한다. 본 발명에 따른 HPJS의 프로세스는 고성능 연산(High Performance Computing) 클라우드 노드의 초기화 정보를 제공하는 머신 배치 파일을 탑재한 HPJS-데몬(Daemon)에 의해 시작된다. 이 정보는 클라우드 노드의 IP, HPJS 프로세스 포트 넘버 및 그들의 처리 랭크를 포함한다. 모든 프로세스들에 배치 파일을 사용하는 것은 그들의 소켓 오브젝트 생성의 결과에 따른 고유한 식별자를 공유한다. 각각의 프로세스는 이 정보를 저장하는 소켓 테이블을 유지한다. 메시지 전달 인터페이스부(130)는 수신된 메시지를 메시지 패킹부(140)로 전달하고 고유한 식별자를 자바 런타임부(160)으로 전달한다.
- [0022] 모든 관련된 HPJS의 프로세스는 해시 테이블(Hash Table)을 생성하고 관리한다. 해시 테이블(Hash Table)은 해싱 함수의 연산에 의해 구해진 위치에 각 레코드를 한 개 이상 보관할 수 있는 버킷(bucket)들로 구성된 기억 공간이다. 상술한 고유의 식별자의 소켓 테이블은 소켓 오브젝트에 맵핑(Mapping)된다. 이 고유의 식별자들은 모든 다른 사용가능한 HPJS 프로세스와 함께 1대1 맵핑을 제공한다. 소켓 테이블은 요청 생성 및 응답 생성을 위한 메시지 패킹부(140)에 의해 재사용된다.
- [0023] 메시지 패킹부(140)는 메시지를 팩(Pack)하는 패킹부와 메시지를 언팩(Unpack)하는 언팩킹부를 포함한다. 메시지 패킹부는 API 소켓 통신부(120)의 내부 모듈 통신을 위한 응답 오브젝트와 요청 오브젝트를 생성한다. 요청 오브젝트 및 응답 오브젝트는 통신 층에 의해 송신 또는 수신되는 메시지를 포함한다. 메시지 패킹부(140)는 소켓 인터페이스부(150)로부터 전달되는 동기화 또는 비동기화 메시지를 요청 오브젝트 또는 응답 오브젝트 각각의 요청에 따라 바이트를 패(Pack)하거나 언팩(Unpack)한다.
- [0024] 메시지 패킹부(140)에 의해 생성된 요청 오브젝트 및 응답 오브젝트는 헤더(Header)와 페이로드(Payload)의 두 파트로 나뉘어진 메시지를 포함한다. 페이로드는 e헬스 응용부(110)에 의해 API 소켓 통신부(120)를 거쳐 메시지 패킹부(140)로 제공된 바이트 파트이다. 일정한 용량을 가지는 헤더 파트는 메시지에 의존하여 동적인 용량 변화를 가지는 페이로드에 대한 정보를 제공한다. 전송은 한번이 이루어지는 프로세스이지만, 수신된 헤더를 먼저 읽고 페이로드의 종류 및 용량을 결정한 후에 메모리를 할당하고 읽기를 계속한다.
- [0025] 그리고 메시지 패킹부(140)는 API 소켓 통신부(120) 내부의 들어오고 나가는 트래픽(Traffic)을 제어한다. 메시지들은 메시지 전송을 위한 동기화 전송 모듈 또는 비동기화 전송 모듈로 전송된다. 그리고 메시지의 바이트들은 동기화 또는 비동기화 수신 모듈로부터 메시지를 수신하여 할당된다. 또한 메시지 전달 인터페이스부(130)로부터 수신된 메시지를 패킹하여 소켓 인터페이스부(150)로 전달하고, 소켓 인터페이스부(150)로부터 수신된 메시지를 언팩하여 메시지 전달 인터페이스부(130)로 전달한다.
- [0026] 소켓 인터페이스부(150) 동기화 송신부, 동기화 수신부, 비동기화 송신부 및 비동기화 송신부를 포함한다. 동기화 송신부 및 비동기화 송신부는 헤더 및 페이로드와 함께 메시지를 전송한다. 그리고 네트워크 통신의 양을 줄이기 위해 리시버에 의한 리소스 유용성 승인을 기다리지 않는다.
- [0027] 소켓 인터페이스부(150)의 동기화 수신부는 한번에 메시지를 읽어 수신한다. 하지만 동기화 수신부는 한번에 읽은 메시지를 두 단계로 평가한다. 첫 번째 단계로 헤더의 용량 정보를 읽는다. 그리고 두 번째 단계로 헤더 바이트로부터 페이로드의 용량 및 종류를 식별한다. 그리고 나서 페이로드는 수신된 메시지 바이트를 두 번째 단계와 같이 읽는다. 동기화 수신부는 통신의 차단 없이 메시지를 수신한다. 동기화에 의해 유지되는 작업 스레드(Worker Thread)들로 구성된 스레드풀(Thread Pool)은 수신된 메시지를 읽는다. 수신 스레드(Receive Thread)에 의해 거론된 작업 스레드는 메시지를 읽고 동기화 수신부와 거의 유사한 두 단계의 평가 메커니즘을 처리한

다. 동기화 수신부의 처리 후에 작업 스레드는 스레드풀로 되돌아오고 스레드 재사용을 위해 캐시(Cache)에 저장된다. 스레드풀은 항상 스레드 형성량 및 스레드 마다 연결된 리소스를 줄이기 위해 캐시된 스레드를 보수한다.

[0028] 소켓 인터페이스부(150)에서 동기화 전송 및 동기화 수신은 차단 통신 메커니즘(Blocking Communication Mechanisms)이다. 전송 과정은 차단된 연결을 유지하는 것의 결과에 따른 메시지를 전부 수신할 때까지 종료되지 않는다. 그러나 비동기화 전송 및 수신은 비차단 패션(Non-blocking Fashion)과 통신한다. 비동기화 수신은 대기 상태에서 수신을 홀딩하는 것 없이 응답하는 과정을 위한 캐시된 스레드풀로부터 작업 스레드를 야기한다. 캐시된 스레드풀의 도구와 함께, HPJS는 스레드를 재사용하고, 주어진 처리 시간에서 스레드풀 안의 스레드의 최적 숫자를 유지한다. 이 모델은 최적 메모리 작동 프로세스를 촉진한다.

[0029] 클라우드 컴퓨팅 모델의 요구에 따른 리소스와 같이 HPJS 프로세스는 한계 없는 리소스를 가진다고 추정한다. 페이로드의 용량에 관한 수신 프로세스로부터 승인을 위한 응답 대신에 전송 프로세스는 모두 한꺼번에 메시지를 전송한다. 프로세스간 통신의 오버헤드를 줄이기 위해 메시지 당 단지 하나만 전송한다.

[0030] 자바 런타임부(160)는 소켓 인터페이스부(150)로부터 수신된 메시지를 클라우드 OS부(170)로 전달하고, 메시지 전달 인터페이스부(130)로부터 수신된 고유한 식별자로 각각의 메시지를 구별한다. 자바 런타임부(160)은 메시지를 스트리밍하여 수신하거나 수신된 메시지를 스트리밍하여 전달하는 입출력 스트리밍부와 자바 소켓으로 구성된 소켓부를 포함한다.

[0031] 클라우드 OS부(170)는 헬스 클라우드 서버를 작동시키는 운영체제로서 서버를 전반적으로 관리하고 자바 런타임부(160)로부터 수신된 메시지를 헬스 클라우드에 포함된 다른 컴퓨터 단말과 공유한다.

[0032] 이상 바람직한 실시 예를 들어 본 발명을 상세하게 설명하였으나, 본 발명은 전술한 실시 예에 한정되지 않고, 본 발명의 기술적 사상의 범위 내에서 당 분야에서 통상의 지식을 가진 자에 의하여 여러 가지 변형이 가능하다.

부호의 설명

[0033] 110: e헬스 응용부

120: API 소켓 통신부

130: 메시지 전달 인터페이스부

140: 메시지 패킹부

150: 소켓 인터페이스부

160: 자바 런타임부

170: 클라우드 OS부

도면

도면1

