(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(51) Int. Cl.[6]
G06F 15/16

(11) 공개번호 특2000-0011302
(43) 공개일자 2000년02월25일

(21) 출원번호 10-1999-0022327
(22) 출원일자 1999년06월15일

(30) 우선권주장 9/113,674 1998년07월10일 미국(US)
(71) 출원인 인터내셔널 비지네스 머신즈 코포레이션 포만 제프리 엘
미국 10504 뉴욕주 아몬크
(72) 발명자 어렌드트제임스더블유.
미합중국78681텍사스주라운드로크블랙잭드라이브1501
차오칭윤
미합중국78750텍사스주오스틴루스틱록드라이브11518
맨시사이도르로드르포아우스구스토
미합중국78728텍사스주오스틴샌드사이드드라이브14601
(74) 대리인 김성기, 송병옥

심사청구 : 있음

(54) 높은크기조정가능성을갖는고가용성클러스터시스템관리구조

요약

본 발명의 클러스터 시스템은 자원 그룹(resource group)의 집합(set)으로 취급되며, 각 자원 그룹은 가용성이 높은 애플리케이션과 이러한 가용성이 높은 애플리케이션이 종속되는 자원을 포함한다. 자원 그룹은 2 내지 M개(M은 전체 클러스터의 사이즈 N에 비하여 상대적으로 작음)의 데이터 처리 시스템을 가질 수 있다. 자원 그룹의 구성 및 상태 정보는 자원 그룹의 구성 요소인 데이터 처리 시스템에만 전체가 복제(replicate)된다. 클러스터 내에서 데이터 처리 시스템의 고장이 발생하면, 고장난 데이터 처리 시스템을 포함하는 자원 그룹만이 영향을 받는다. 사용 가능한 정족수의 데이터 처리 시스템을 보유한 각 자원 그룹은 계속하여 서비스를 제공하여 클러스터가 복구(restore)되는 도중이라도 클러스터 내의 많은 애플리케이션이 계속하여 기능을 수행하도록 한다.

대표도

도1

색인어

시스템, 서버, 클러스터, 자원 그룹, 데이터 처리 시스템, 복제

명세서

도면의 간단한 설명

도 1은 본 발명의 바람직한 실시예가 구현될 수 있는 클러스터 다중 처리 시스템의 블록도.

도 2a 내지 2h는 본 발명의 바람직한 실시예에 따른 데이터 처리 시스템과 자원 그룹들 간의 구성 정보 분산 다이어그램.

도 3은 본 발명의 바람직한 실시예에 따른 자원 그룹을 포함하는 클러스터 내의 구성 및 상태 정보를 복제하는 과정을 나타내는 고수준 흐름도.

도 4는 본 발명의 바람직한 실시예에 따른 자원 그룹을 포함하는 클러스터 내 노드 고장을 처리하는 과정을 나타내는 고수준 흐름도.

발명의 상세한 설명

발명의 목적

발명이 속하는 기술 및 그 분야의 종래기술

본 발명은 일반적으로 클러스터 시스템 관리(cluster system management)에 관한 것으로, 특히 높은 크기 조정 가능성을 갖는 클러스터들의 관리에 관한 것이다. 더욱 상세하게는, 본 발명은 높은 크기 조정 가능성을 갖는 클러스터를 관리하기 위한 부분적 분산 클러스터 구성 정보(partially distributing cluster configuration information)에 관한 것이다.

"클러스터" 또는 클러스터 다중 프로세서 시스템(cluster multiprocessor system; CMP)으로도 언급되는 클러스터 시스템은 다수의 데이터 처리 시스템 사이에 하드웨어와 소프트웨어가 공유되어 네트워크화된 데이터 처리 시스템들의 집합(set)이며, 반드시 필요한 것은 아니지만 일반적으로 매우 높은 가용성(available)과 확장성(scalable)을 갖는 애플리케이션 서비스(application service)를 제공하도록 구성된다. 항공기 제어 등과 같은 임무가 매우 중요한 애플리케이션에 사용되는 고장 허용(fault tolerance)에 대한 대안으로, 클러스터 시스템은 흔히 높은 가용성을 달성하도록 구현된다. 고장 허용 데이터 처리 시스템은 특화된(specialized) 하드웨어를 기반으로 하여, 하드웨어 구성 요소가 프로세서, 메모리 보드, 하드디스크 드라이브, 어댑터 및 전원 공급기(power supply) 등의 어느 것이거나 구별하지 않고, 고장난 하드웨어를 검출하여 여분(redundant)의 하드웨어 구성요소로 스위칭한다. 고장 허용 시스템은 작업을 중단 없이 흔적이 남지 않는 컷오버(cutover)를 제공하지만, 이 시스템은 여분의 하드웨어가 요구되기 때문에 고가이고, 데이터 처리 시스템 고장의 가장 일반적인 원인인 소프트웨어 오류를 처리할 수 없다.

고 가용성이 높아지면 표준 하드웨어를 사용하더라도 시스템 전체로 자원이 공유되도록 하는 소프트웨어를 제공한다. 노드, 구성 요소 또는 애플리케이션이 고장이 발생하는 경우, 원하는 자원에 대한 대체 경로(alternative path)가 신속하게 설정된다. 많은 경우에 있어서, 자원 가용성의 재설정에 필요한 간단한 인터럽션(brief interruption)은 수용 가능하다. 하드웨어 비용은 고장 허용 시스템에 비하여 훨씬 더 적게 들며, 정상 동작중에 백업 장치가 사용될 수 있다. 이러한 목적으로 사용되는 소프트웨어의 예로서, 뉴욕주 아몬크에 소재하는 IBM사가 상용화한 AIX™(Advanced Interactive Executive; 유닉스 운영체재의 일종)용 HACMP(High Availability Cluster Multiprocessing) 소프트웨어와 IBM사가 상용화한 RS6000 SP 소프트웨어가 있다.

클러스터 시스템 관리 문제는 일반적인 시스템 관리 문제의 특별한 유형으로, 추가적인 자원 종속성(resource dependency)과 관리 정책 제한(management policy constraint)을 갖는다. 특히, 시스템 관리를 위하여 필요한 클러스터 구성 정보의 유지는 특별한 문제를 유발한다. 시스템 관리를 위하여 필요한 클러스터 구성 정보는 일반적으로 중앙 집중된 데이터베이스 또는 고 가용성을 위하여 하나 이상의 데이터 처리 시스템에 복제된 데이터베이스에 기억된다. 중앙 집중식 클러스터 구성 데이터베이스를 관리하는 데이터 처리 시스템은 잠재적으로 병목 현상을 일으키거나 고장의 한 원인이 된다.

중앙 집중식 클러스터 구성 데이터베이스의 문제점을 회피하기 위하여, 이 데이터베이스가 클러스터 내의 다수의 처리 시스템에 복사되어 유지될 수 있다. 작은 클러스터 내의 경우, 시스템 구성 및 상태 정보가 클러스터 내의 모든 데이터 처리 시스템에 쉽게 복사될 수 있어서, 각 데이터 처리 시스템이 이 정보를 고장 회복 및 적재 균형(load balancing)과 같은 시스템 관리 기능을 수행하는데 사용한다. 클러스터 크기가 작은 경우(2 내지 8 데이터 처리 시스템)에는, 전체 복사가 고 가용성 클러스터 구성 데이터베이스를 제공하며 적절하게 동작한다. 그러나 대규모 클러스터의 경우, 전체 복사에 관련된 비용은 지나칠 정도로 높다.

분산 데이터베이스를 항상 일관된 상태로 유지하기 위하여, 2 단계 확인(two-phase commit) 프로토콜이 사용된다. 전체 복사된 데이터베이스(즉, 모든 데이터 처리 시스템은 사본을 가짐)에 있어서, 각 기록 동작에 대하여 2N(N은 클러스터 내의 데이터 처리 시스템의 수)개의 메시지가 교환되어야 한다. 따라서 클러스터 구성/상태 데이터베이스의 사이즈가 클러스터 사이즈에 대하여 선형적으로 증가하면, 데이터베이스에 대한 액세스 시간은 클러스터 사이즈에 대하여 선형적 또는 지수적으로 증가한다. 또한 클러스터를 성장시키면, 이벤트의 수(및 갱신되어야 할 상태 정보의 양)는 클러스터 사이즈에 대하여 선형적으로 증가한다. 따라서 전체 복사된 분산 클러스터 구성 데이터베이스를 갖는 클러스터를 성장시키는데 필요한 시간 또는 비용은 N^2의 차수로 증가한다. 따라서 클러스터 시스템 관리의 복잡성(complexity)의 특징은 N^2의 차수를 갖는 것이다. 대규모 클러스터 시스템(1,000 이상의 데이터 처리 시스템)의 경우, 클러스터 구성 데이터베이스를 전체 복사하는 경우 비실용적이다.

고 가용성 클러스터 시스템에 있어서 다른 중요한 문제는 네트워크 분할(network partition)을 어떻게 처리할 것인가 하는 것이다. 클러스터가 2이상의 부분으로 나누어지고, 그 중 한 부분에 속한 데이터 처리 시스템이 다른 부분의 데이터 처리 시스템과 통신할 수 없는 경우, 네트워크 분할이 발생한다. 네트워크 분할이 발생하면, 클러스터의 (일시적으로) 독립적인 부분으로부터 동일 애플리케이션(특히, 클러스터 구성 데이터베이스와 같은 데이터베이스 애플리케이션)의 다중 복사를 실행하지 않는 것이 중요하다. 이 문제를 처리하는 표준화된 방법은 클러스터가 정족수(quorum)에 도달하지 못하면 오프라인 상태를 유지하도록 하는 것이다. 정족수는 경우마다 다르다. 구현예에서는, 다수 원칙에 따른 정족수(majority quorum)가 사용되고, 클러스터의 일부는 활성 상태의 서버 수가 적어도 N/2+1이면, 정족수에 도달하였다고 한다. 다른 방안은 시스템이 정족수에 도달할 수 있는 최대 수가 클러스터의 일부분만임을 보장하는한, 정족수에 도달하기 위해 활성 상태에 있는 더 작은 수의 서버를 요구할 수 있다. 높은 크기 조정 가능성을 갖는 클러스터에 있어서, 정족수에 대한 조건이 매우 제한되는 경향이 있다. 본 발명이 다른 정족수 형태에 사용될 수 있지만, 본 명세서에서는 다수 원칙에 따른 정족수가 사용된다.

따라서 네트워크 분할이 발생하면, 클러스터 내 의 다수의 데이터 처리 시스템을 포함하고 있는 클러스터 일부분이 존재할 경우 그 일부분만이 애플리케이션을 실행(run)할 수 있다. 즉, 클러스터의 데이터 처리 시스템중 적어도 절반 이상이 온라인(online) 상태가 아니면 클러스터는 어떠한 서비스도 제공하지 않는다.

따라서 클러스터 구성 정보를 포함하는 분산된 데이터베이스를 전체 복사와 연관된 비용을 발생시키지 않고 유지하는 메커니즘을 제공하는 것이 바람직하다. 또한, 1,000개 이상의 데이터 처리 시스템의 경우에도 메커니즘이 임의 크기의 클러스터 사이즈에 대해서도 확장 및 응용 가능하도록 하는 것이 더욱 효과적이다. 네트워크가 분할된 후, 정족수에 도달하지 못한 경우에도 클러스터 여러 부분들이 계속해서 서비스를 제공하도록 하는 것이 또한 효과적이다.

발명이 이루고자하는 기술적 과제

본 발명의 목적은 개선된 클러스터 시스템 관리 방법 및 장치를 제공하는 것이다.

본 발명의 또 다른 목적은 높은 크기 조정 가능성을 갖는 클러스터를 관리하는 개선된 방법 및 장치를 제공하는 것이다.

본 발명의 또 다른 목적은 높은 크기 조정 가능성을 갖는 클러스터를 관리하기 위하여 클러스터 구성 정보를 부분적으로 분산시키는 방법 및 장치를 제공하는 것이다.

상기 목적은 이하 기술되는 바와 같이 달성된다. 클러스터 시스템은 자원 그룹(resource group)의 집합(set)으로 취급되며, 각 자원 그룹은 가용성이 높은 애플리케이션과 이러한 가용성이 높은 애플리케이션이 종속되는 자원을 포함한다. 자원 그룹은 2 내지 M개(M은 전체 클러스터의 사이즈 N에 비하여 상대적으로 작음)의 데이터 처리 시스템을 가질 수 있다. 자원 그룹의 구성 및 상태 정보는 자원 그룹의 구성 요소인 데이터 처리 시스템에만 전체가 복제(replicate)된다. 클러스터 내에서 데이터 처리 시스템의 고장이 발생하면, 고장난 데이터 처리 시스템을 포함하는 자원 그룹만이 영향을 받는다. 사용 가능한 정족수의 데이터 처리 시스템을 보유한 각 자원 그룹은 계속하여 서비스를 제공하여 클러스터가 복구(restore)되는 도중이라도 클러스터 내의 많은 애플리케이션이 계속하여 기능을 수행하도록 한다.

본 발명의 상기 목적 및 부가적인 목적, 특징 및 이점은 발명의 상세한 설명으로부터 명백히 이해될 수 있다.

발명의 구성 및 작용

본 발명은 그 목적 및 장점을 이용하는 최적 모드로서 실시되는 다음에 설명한 실시예의 상세한 기술을 참조하면서 첨부한 도면을 파악하면 최적으로 이해될 것이다.

도 1은 본 발명의 바람직한 실시예가 구현될 수 있는 클러스터 다중 처리 시스템의 블록도이다. 시스템(102)은 복수의 서버 노드(server node; 104-110)을 포함하며, 일반적으로 서버 노드 각각은 고유의 이름으로 식별된다. 각 노드(104-110)는 IBM사가 상용화한 RISC System/6000??과 같은 대칭형 다중 프로세서(symmetric multiprocessor; SMP) 데이터 처리 시스템일 수 있으며, 또는 Windows NT™서버로서 기능하는 데이터 처리 시스템일 수 있다.

시스템(102) 내의 각 노드(104-110)는 "Advanced Interactive Executive(AIX??)"와 같은 IBM사가 상용화한 운영 체제 또는 마이크로소프트사가 상용화한 Windows NT™ 운영 체제를 포함할 수 있다. 시스템(102) 내의 각 노드(104-110)는 또한 상기 운영 체제와 함께 또는 운영 체제 상에서 실행될 수 있는 고 가용성 클러스터 소프트웨어(high availability cluster software)를 포함한다. 이 고 가용성 클러스터 소프트웨어는 다음에 설명되는 특징을 갖는다.

노드(104-110)는 공중 근거리 통신망(local area network; LAN)(112-114)에 연결되어 있다. 이 공중의 근거리 통신망은 이더넷(Ethernet), 토큰링(Token-Ring), 화이버 분산 데이터 인터페이스(fiber distribution data interface; FDDI) 또는 기타 다른 네트워크일 수 있다. 클라이언트(client; 116-120)는 공중 네트워크(112-114)를 통하여 서버(104-110)에 액세스한다. 클라이언트(116-120)는 액세스가 가능한 데이터 처리 시스템으로, 각각의 클라이언트는 노드(104-110) 상에서 실행되는 서버 애플리케이션을 질의(query)하는 클라이언트 애플리케이션 또는 "프론트-앤드(front-end)" 프로그램을 실행한다.

통상적으로, 각 노드(104-110)는 디스크 버스(128-130)를 통하여 공유된 외부 디스크(122-126)의 데이터에 액세스할 수 있는 서버 또는 "백-앤드(back-end)" 프로그램 애플리케이션을 실행한다. 노드(104-110)는 또한 부가적인 네트워크(132) 또는 네트워크들에 의하여 연결될 수 있다. 예를 들어, 클라이언트(116-120)가 액세스할 수 없는 상태에서 시스템(102) 내의 노드(104-110) 사이의 두 지점을 연결할 수 있는 전용 네트워크가 설치될 수 있다. 이더넷, 토큰링, FDDI 또는 직렬 광 채널 커넥터(SOCC) 네트워크가 전용 네트워크가 될 수 있으며, 전용 네트워크가 사용 가능한 경우, 이러한 전용 네트워크는 록 트래픽(lock traffic)에 사용될 수 있다. 직렬 네트워크가 노드(104-110) 사이의 두 지점간 통신을 제공하며, 대체 서브시스템(alternative subsystem)이 고장난 경우에 제어 메시지(control message) 및 하트비트(heartbeat) 트래픽에 사용된다.

예시된 실시예에 설명된 바와 같이, 시스템(102)은 소정 수준의 중복성(redundancy)을 가져서 각 단일점 고장(single points of failure)을 제거한다. 예컨대, 각 노드(104-110)는 서비스 어댑터와 대기 어댑터로 이루어진 두 개의 네트워크 어댑터(도시되지 않음)에 의하여 공중 네트워크(112-114)에 각각 연결될 수 있다. 서비스 어댑터는 노드와 네트워크 간의 제1 활성 연결(primary active connection)을 제공하고, 대기 어댑터는 상기 서비스 어댑터가 고장난 경우에 이 서비스 어댑터를 대체한다. 따라서 시스템(102) 내의 자원이 사용될 수 없게 되면, 대체 자원이 고장난 자원을 신속하게 교체할 수 있게 된다.

본 발명 분야의 당업자는 도 1의 예시적 실시예에 설명된 하드웨어를 변경할 수 있다. 예컨대, 시스템은 더 많거나 더 적은 수의 노드, 추가적인 클라이언트, 및/또는 도시되지 않은 기타 다른 연결을 포함할 수 있다. 추가적으로, 본 발명에 따른 시스템(102)은 데이터 처리 시스템(104-110) 간의 신뢰성 있는 통신과 동기화(synchronizations) 및 이하 상세히 설명되는 바와 같은 집적된 클러스터 시스템 관리

장치를 포함한다.

도 2a 내지 도 2h는 본 발명의 바람직한 실시예에 따른 클러스터 시스템에 사용되는 데이터 처리 시스템 및 자원 그룹 사이의 구성 정보 분산(configuration information distribution)을 도시하고 있다. 본 발명에 있어서, 클러스터는 자원 그룹의 시스템으로 취급되고, 자원 그룹 내의 각의 데이터 처리 시스템에 대한 클러스터 구성 정보는 해당 자원 그룹 내의 자신을 제외한 모든 다른 데이터 처리 시스템에만 복제된다.

클러스터 구성 정보의 주된 용도 중의 하나는 클러스터 자원의 가용성을 높이기 위한 것이다. 예를 들어, 클러스터 내의 데이터 처리 시스템이 고장나면, 해당 데이터 처리 시스템 상의 애플리케이션은 다른 데이터 처리 시스템으로 이동한다. 따라서 고장난 데이터 처리 시스템이 제공하던 서비스는 잠시 중단(interruption)된 후에 계속 제공된다. 일반적으로 많아야 하나의 데이터 처리 시스템만이 시간상 임의의 시점에서 고 가용성 애플리케이션을 관리하지만, 애플리케이션 또는 다른 자원이 높은 가용성을 갖기 위해서는 클러스터 내의 다수의 데이터 처리 시스템이 구성되어 해당 애플리케이션 또는 자원을 실행하여야 한다.

본 발명에 따라, 고 가용성 애플리케이션 및 이 애플리케이션이 종속되는 모든 자원은 자원 그룹을 형성한다. 각각의 자원 그룹은 그룹을 관리할 수 있는 데이터 처리 시스템의 순서 목록(ordered list)을 갖는다. 자원 그룹 내의 데이터 처리 시스템 수는 2 내지 M(M은 대규모 클러스터의 경우 클러스터 사이즈 N에 비하여 매우 작음)까지 변한다. 구성 및 상태 정보는 데이터베이스 내에 구성 객체(object)로 조직되며(organize), 높은 가용성의 자원 그룹은 각각 구성 및 상태 객체를 갖는다. 자원 그룹에 대한 구성 및 상태 객체는 관련된 오너(owner)의 목록을 가지며, 이 목록은 대응하는 자원 그룹을 관리할 수 있는 데이터 처리 시스템의 목록과 동일하다. 구성 및 상태 정보 객체는 오너 목록 내의 데이터 처리 시스템에만 복제된다.

도 2a의 예시적 실시예는 4개의 자원 그룹(220-226)으로 조직된 9개의 데이터 처리 시스템(202-218)을 예시하고 있다. 각각의 자원 그룹(220-226) 내에 일반적으로 오직 하나의 데이터 처리 시스템만이 임의의 주어진 시점에서 해당 자원 그룹에 대한 주어진 애플리케이션을 관리한다. 그러나, 제 1 데이터 처리 시스템이 고장인 경우, 애플리케이션의 관리를 맡을 다른 데이터 처리 시스템이 지정된다. 각각의 자원 그룹(220-226)에 대한 구성 객체는 그 자원 그룹 내의 각각의 데이터 처리 시스템에 복제된다. 자원 그룹 내의 각각의 데이터 처리 시스템은 그 자원 그룹에 대한 구성 객체의 오너로서 나열된다. 구성 객체는 각 자원 그룹이나 자원에 관련된 클러스터 구성 정보 및 상태 정보를 포함하는데, 이러한 구성 정보 및 상태 정보는 데이터 처리 시스템, 네트워크, 네트워크 인터페이스 카드(어댑터), 및 네트워크 연결 정보와 같은 접속 형태 정보(topology information); 자원의 애플리케이션 타입용 애플리케이션 패키지, 자원의 공유 디스크 타입용 공유 디스크(shared disk), 데이터 처리 시스템 및 디스크 연결 정보, 자원의 서비스 IP 어드레스 타입용 서비스에 대한 서비스 IP 어드레스, 애플리케이션이 설치되고 구성되는 데이터 처리 시스템, 관리 정책(management policy), 관리 규칙(management rule), 및 자원 종속성 관계와 같은 자원 그룹 정보; 및 데이터 처리 시스템의 상태, 네트워크의 상태, 네트워크 인터페이스 카드의 상태, 공유 디스크의 상태, 애플리케이션의 상태, 및 이벤트 처리의 상태와 같은 클러스터 시스템 상태 정보를 포함한다. 구성 객체는 또한 자원 종속성을 평가하는 규칙뿐만 아니라 데이터 처리 시스템, 네트워크, 네트워크 인터페이스 카드, 공유 디스크, 자원 그룹 및 자원들을 추가/수정/삭제하는 규칙을 포함한다.

도 2b 내지 2h는 자원 그룹 내의 주어진 데이터 처리 시스템에 대한 상태 및 성능 정보의 복제를 도시하고 있다. 도 2b에 도시된 바와 같이, 데이터 처리 시스템(202)은 자원 그룹(220)의 구성원(member)이다. 따라서, 데이터 처리 시스템(202)에 대한 구성 및 상태 정보는 이 자원 그룹 내의 데이터 처리 시스템(데이터 처리 시스템(208 및 214)을 포함함)에 복제된다. 데이터 처리 시스템(202)이 관리하는 임의의 애플리케이션에 대한 구성 객체는 자신과 관련된 오너 목록 내에 데이터 처리 시스템(202, 208 및 214)을 오너로서 나열(list)한다. 데이터 처리 시스템(214) 또한 자원 그룹(220)의 구성원이며, 따라서 데이터 처리 시스템(214)의 구성 및 상태 정보도 데이터 처리 시스템(202 및 208)에 복제되며, 데이터 처리 시스템(214)이 관리하는 애플리케이션에 대한 구성 객체는 데이터 처리 시스템(202, 208 및 214)을 오너로서 나열한다.

도 2c 및 도 2d는 자원 그룹(222)의 데이터 처리 시스템(204 및 216)에 대한 구성 정보의 복제와 자원 그룹(224)의 데이터 처리 시스템(206 및 218)에 대한 구성 정보의 복제를 설명한다. 데이터 처리 시스템(204 및 216)에 대한 구성 및 상태 정보는 데이터 처리 시스템(204, 210 및 216) 각각에 복제되며, 데이터 처리 시스템(206 및 218)에 대한 구성 및 상태 정보는 데이터 처리 시스템(206, 212 및 218) 각각에 복제된다. 데이터 처리 시스템(204 또는 216)이 관리하는 애플리케이션은 데이터 처리 시스템(204, 210 및 216)을 포함하는 구성 객체 오너 목록을 가지며, 이 구성 객체 자신은 데이터 처리 시스템(204, 210 및 216) 각각으로 복제된다. 마찬가지로, 데이터 처리 시스템(206 또는 218)이 관리하는 애플리케이션은 데이터 처리 시스템(206, 212 및 218)을 오너로 지정한 구성 객체 오너 목록을 가지며, 이 구성 객체는 데이터 처리 시스템(206, 212 및 218) 각각에 복제된다.

도 2e는 데이터 처리 시스템(208)이 2개 이상의 겹치는(overlapping) 자원 그룹(220 및 226)에 속하는 정보의 복제를 도시한다. 데이터 처리 시스템(208)에 대한 구성 및 상태 정보는 자원 그룹(220 및 226)을 포함하는 데이터 처리 시스템(208)과, 그에 따른 데이터 처리 시스템(202, 210, 212 및 214)을 포함하는 적어도 하나의 자원 그룹에 속하는 각각의 데이터 처리 시스템으로 복제된다. 데이터 처리 시스템(208)에 의해 관리되는 애플리케이션들에 대한 구성 객체는 대응하는 자원 그룹에 속하는 각각의 데이터 처리 시스템들을 포함하는 오너 목록을 구비하며, 이들 데이터 처리 시스템들 각각으로 복제된다. 따라서, 예를 들어 자원 그룹(220)의 일부인 데이터 처리 시스템(208)에 의해 관리되는 애플리케이션은 데이터 처리 시스템(202, 208 및 214)을 오너로서 식별하는 구성 객체 오너 목록을 포함한다. 이 애플리케이션에 대한 구성 객체는 데이터 처리 시스템(202, 208 및 214) 상에 복제된다. 자원 그룹(226)의 대체부인 데이터 처리 시스템(208)에 의해 관리되는 애플리케이션은 데이터 처리

시스템(208, 210 및 212)을 오너로서 식별하는 구성 객체 오너 목록을 포함하며, 이 애플리케이션에 대한 구성 객체는 데이터 처리 시스템(208, 210 및 212) 상에 복제된다.

마찬가지로, 도 2f 및 도 2g는 각각 그룹(222 및 226) 및 그룹(224 및 226)과 같은 2 이상의 자원 그룹에 속하는 데이터 처리 시스템(210 및 212)에 대한 구성 정보의 복제를 설명한다. 데이터 처리 시스템(210)에 대한 구성 및 상태 정보는 데이터 처리 시스템(204, 208, 212 및 216) 전체에 걸쳐 복제되며, 데이터 처리 시스템(212)에 대한 구성 및 상태 정보는 데이터 처리 시스템(206, 208 및 218) 상에 복제된다. 자원 그룹(222)의 일부인 데이터 처리 시스템(210)에 의해 관리되는 애플리케이션에 대한 구성 객체는 데이터 처리 시스템(204, 210 및 216)을 포함하는 관련된 오너들을 포함하며, 이들 데이터 처리 시스템들로 복제되고; 자원 그룹(226)의 일부를 형성하는 데이터 처리 시스템(210)에 의해 관리되는 애플리케이션들에 대한 구성 객체는 데이터 처리 시스템(208, 210 및 212)을 포함하는 관련 오너 목록을 구비하며, 이들 데이터 처리 시스템들로 복제된다. 자원 그룹(224)의 일부인 데이터 처리 시스템(212)에 의해 관리되는 애플리케이션들에 대한 구성 객체는 데이터 처리 시스템(206, 212 및 218)을 포함하는 관련 오너 목록을 포함하며, 이들 데이터 처리 시스템들로 복제되고; 자원 그룹(226)의 일부를 형성하는 데이터 처리 시스템(212)에 의해 관리되는 애플리케이션들에 대한 구성 객체는 데이터 처리 시스템(208, 210 및 212)을 포함하는 관련 오너 목록을 구비하며, 이들 데이터 처리 시스템들로 복제된다.

자원 그룹에 대한 각각의 구성 객체/데이터베이스 기록은 구성 객체에 대한 오너들의 관련 목록 내의 데이터 처리 시스템들로만 복제되는데, 그 이유는 각각의 해당 정보가 가장 자주 사용되는 장소이기 때문이다. 애플리케이션이 클러스터 내의 모든 데이터 처리 시스템을 활용하는 가능성이 거의 없는 상황을 제외하고는 어느 대규모 클러스터 내에서 어느 데이터 처리 시스템도 전체 클러스터 내의 모든 데이터 처리 시스템에 대한 구성 및 상태 정보를 포함하지 않는다. 이 경우, 애플리케이션에 대한 구성 객체는 클러스터 내의 모든 데이터 처리 시스템을 포함하는 오너 목록을 구비할 수 있다.

자원 그룹에 대한 구성 객체/데이터베이스 기록과는 달리, 주어진 데이터 처리 시스템에 대한 구성 및 상태 정보는 그 영향의 범위 내에 모든 데이터 처리 시스템으로 복제된다(즉, 이들 데이터 처리 시스템은 객체 데이터 처리 시스템을 갖는 적어도 하나의 자원 그룹의 일부를 형성하며, 따라서 객체 데이터 처리 시스템의 고장에 의해 영향을 받을 수 있다). 따라서, 예컨대 데이터 처리 시스템(202 및 214)은 각각 데이터 처리 시스템(202, 208 및 214)을 포함하는 도 2b에 도시된 영향의 범위(228)를 포함하고; 데이터 처리 시스템(204 및 216)은 각가 데이터 처리 시스템(204, 210 및 216)을 포함하는 도 2c에 도시된 영향의 범위(230)를 각각 포함하며; 데이터 처리 시스템(206 및 218)은 각각 데이터 처리 시스템(206, 212 및 218)을 포함하는 도 2d에 도시된 영향의 범위(232)를 포함하고; 데이터 처리 시스템(208)은 데이터 처리 시스템(202, 208, 210 및 214)을 포함하는 도 2e에 도시된 영향의 범위(234)를 포함하며; 데이터 처리 시스템(210)은 데이터 처리 시스템(204, 208, 210 및 216)을 포함하는 도 2f에 도시된 영향의 범위(236)를 포함하고; 데이터 처리 시스템(212)은 데이터 처리 시스템(206, 208, 210, 212 및 218)을 포함하는 도 2g에 도시된 영향의 범위(238)를 포함한다.

본 발명에 따른 구성 및 상태 정보의 일부 복제를 위해 구성되는 클러스터 내에서 데이터 처리 시스템 고장과 같은 이벤트 발생하는 경우, 오너로서 고장난 데이터 처리 시스템을 포함하는 자원 그룹만이 영향을 받는다. 필요한 복구 동작이모든 오너들 중에서 그룹 단위로 조정된다. 관련된 오너들의 지정 목록을 이들 오너들만이 구성 객체/데이터베이스 기록을 관리하도록 함으로써, 대규모 클러스터가 병렬로 실행되는 자율 그룹의 집단으로 효과적으로 관리된다.

M개의 데이터 처리 시스템을 포함하는 자원 그룹을 관리하는 복잡성(complexity)은 M^2인데, M은 통상적으로 대규모 클러스터의 크기(N)보다 훨신 더 작으므로, 구성 및 상태 데이터베이스를 복제하는 경우와, M개의 데이터 처리 시스템들 중에서 분할된 데이터베이스에서의 정보를 액세스하는 경우 모두에서 상당한 성능 개선이 달성될 수 있다. 클러스터 시스템 관리의 복잡성은 $(M/N)^2$의 인자에 의해서 감소되기 때문에, 시스템 이벤트를 관리하기 위한 응답 시간은 상당히 빠르다. 본 발명의 접근 방법의 경우, 구성 및 상태 데이터베이스를 갱신하는 2상 코미트 프로토콜(two-phase commit protocol) 형태로 전송된 메시지의 수와 데이터베이스 액세스 시간은 모두 클러스터 내의 데이터 처리 시스템들의 부분 집합만을 포함함으로써 상당히 감소된다.

개별 클러스터 구성 데이터베이스는 자원 그룹 구성 데이터베이스의 최상위에서 실행될 수 있다. 클러스터 구성 데이터베이스는 클러스터 내의 모든 데이터 처리 시스템으로 복제될 수 있으며, 네트워크, 데이터 처리 시스템, 및 클러스터 시스템 이벤트 등에 관한 클러스터 구성 및 상태 정보를 포함한다.

본 발명에 따른 도 2a 내지 도 2h에 도시된 9-노드 예의 분할은 7개의 서로 다른 구성 데이터베이스를 발생시킨다. 노드 그룹(228)에 의해 관리되는 구성 데이터베이스의 간단한 예는 다음과 같다.

```
ha_resource_groups{
        ha_resource_group=ha_resource_group_220
        current_computer_id=202;
}
computer{
        computer_id=202
        recovery_status="up";
        computer_id=214
        recovery_status="up";
```

```
}
```

노드 그룹(230)에 의해 관리되는 구성 데이터베이스의 간단한 예는 다음과 같다.

```
ha_resource_groups{
        ha_resource_group=ha_resource_group_222
        current_computer_id=204;
}
computer{
        computer_id=204
        recovery_status="up";
        computer_id=216
        recovery_status="up";
}
```

노드 그룹(232)에 의해 관리되는 구성 데이터-베이스의 간단한 예는 다음과 같다.

```
ha_resource_groups{
        ha_resource_group=ha_resource_group_224
        current_computer_id=206;
}
computer{
        computer_id=206
        recovery_status="up";
        computer_id=218
        recovery_status="up";
}
```

노드 그룹(240)에 의해 관리되는 구성 데이터-베이스의 간단한 예는 다음과 같다.

```
ha_resource_groups{
        ha_resource_group=ha_resource_group_226
        current_computer_id=208;
}
computer{
        computer_id=208
        recovery_status="up";
}
```

노드 그룹(234)에 의해 관리되는 구성 데이터-베이스의 간단한 예는 다음과 같다.

```
computer{
        computer_id=208
        recovery_status="up";
}
```

노드 그룹(236)에 의해 관리되는 구성 데이터-베이스의 간단한 예는 다음과 같다.

```
computer{
        computer_id=210
        recovery_status="up";
}
```

마지막으로, 노드 그룹(238)에 의해 관리되는 구성 데이터-베이스의 간단한 예는 다음과 같다.

```
computer{
```

```
        computer_id=212
        recovery_status="up";
}
```

복구의 예로서, 이러한 분할된 시스템에 있어서 노드(208)가 고장인 것으로 가정한다. 노드의 recovery_status는 노드(202, 214, 210 및 212)를 포함하는 그룹(234)의 남아 있는 그룹 멤버에 의해 "down"으로 변경된다. 노드 그룹(234)에 대한 결과적인 구성 데이터베이스는 다음과 같다.

```
computer{
        computer_id=208
        recovery_status="down";
}
```

노드(208) 상에서 동작하는 애플리케이션인 ha_resource_group_226은 소정의 다른 노드상에서 재시작되어야 한다. 이 애플리케이션은 자원 그룹(240)에 의해 관리되며, 따라서 노드(210) 또는 노드(212) 상에서 재시작될 수 있다. 노드(21)가 자원 그룹(240) 내의 2개의 남아 있는 노드에 의해 선택되어 ha_resource_group_226을 실행하면, 노드 그룹(240)에 대한 구성 데이터베이스는 다음과 같다.

```
ha_resource_groups{
        ha_resource_group=ha_resource_group_226
        current_computer_id=210;
}
```

자원 그룹 내의 정족수 조건의 예로서, 전체 9-노드 클러스터가 재시작되고 처음에는 노드(202 및 208)만이 업(up) 상태로 실행(running) 중인 것으로 가정한다. 그룹(228)에 의해 관리되는 애플리케이션(ha_resource_group_220)은 정족수 조건에 도달하게 된다. 노드(202 및 208)는 어느 노드가 애플리케이션(ha_resource_groups_220)을 실행해야 하는지를 노드를 노드(202 및 208) 사이에서 결정할 수 있다. 이러한 접근 방법은 클러스터 전체가 정족수를 갖지 않더라도 데이터 보전성을 절충함이 없이 애플리케이션(ha_resource_groups_220)을 실행하도록 한다. 즉, 총 9개의 노드 중에서 2개의 노드만이 업(up) 상태에 있다. 반면에, 그룹(240)에 의해 관리되는 애플리케이션(ha_resource_ group_226)은 그 그룹 내에 하나의 노드[노드(208)]만을 가지며, 따라서 정족수 조건을 갖지 않는다.

본 발명에 따른 부분 복제 관리 방법은 또한 네트워크 분할(network partition)과 같은 이변을 중앙 집중식 또는 완전 복제 형태보다도 양호하게 처리한다. 구성 및 상태 정보를 자원 그룹 오너들중에서만 부분 복제하는 경우, 대응하는 오너 목록 내에 데이터 처리 시스템들의 절반 이상이 온라인이면, 클러스터 내의 각각의 자원 그룹이 서비스들을 제공할 수 있다. 따라서, 구성 정보의 부분 복제를 갖는 클러스터는 각각이 클러스터 내의 모든 데이터 처리 시스템의 정족수보다 훨신 더 작은 조각으로 분할되더라도 신뢰성 있는 서비스를 지속적으로 제공할 수 있다.

구성 데이터베이스를 분할하고, 서버들의 각각의 서브-클러스터가 그 구성을 관리하도록 함으로써, 서버들의 서브-클러스터가 "정족수"에 도달하는 경우에 서비스 제공을 개시할 수 있으며, 이러한 상황은 클러스터 전체가 정족수에 도달하기 전에 발생할 수 있다. 자원 그룹에 의해 신뢰성 있게 제공될 수 있는 적어도 하나의 서비스를 제공이 가능한 경우에 온라인 상태에 있어야 하는 자원 그룹 노드들의 "정족수"가 반드시 자원 그룹 내의 다수의 노드일 필요는 없다. 또한, 예를 들어 다중 오류가 발생하면, 클러스터가 정족수에 도달할 수 없게 되는 경우가 발생할 수 있다. 이러한 경우에, 서브-클러스터는 자신들이 정족수를 갖는 한 자신들의 서비스를 계속 제공할 수 있다. 이것은 기존의 구조가 정족수를 클러스터 전체와 연관시키면서도 정족수 조건을 각각의 자원 그룹과 연관시키는 장점이다.

복구 동작 및 로드(load) 균형은 각각의 자원 그룹 내의 서버들에 의해 기본 그룹 단위로 수행된다. 다시 말해서, 자원 할당 결정은 자원 그룹 내의 서버들에 의해서 이루어진다. 다중 자원 그룹들이 하나 이상의 서버를 공통으로 공유하는 경우, 자원 할당 결정이 조정되지 않으면 분류 조건(race condition)들이 나타날 수 있다. 예컨대, 도 2a는 공통 서버(208)를 공유하는 자원 그룹(220 및 226), 공통 서버(210)를 공유하는 자원 그룹(222 및 226), 및 공통 서버(212)를 공유하는 자원 그룹(224 및 226)을 갖는 3개의 자원 그룹을 포함하는 클러스터를 나타낸다. 이들 자원 그룹들에 대해 자원 그룹 관리자들에 의한 로드 할당의 일부 조정이 제공되어야 한다.

하나 이상의 서버를 공통으로 공유하는 자원 그룹들은 또한 구조 및 상태 정보도 공유하여야 하고, 그 자원 할당 결정도 조정하여야 한다. 이는 양쪽 자원 그룹들에 공통인 모든 서버들이 양쪽 자원 그룹들의 자원 할당 결정의 순차화시키도록함으로써 이루어진다. 예컨대, 도 2e에 도시한 바와 같이, 영향의 영역(234)을 갖는 서버(208)는 자원 그룹(220 및 226)의 구성 및 상태 정보를 서로 복제하는 기능을 한다. 서버(208)는 또한 2개의 자원 그룹의 자원 할당 결정을 순차화하는 기능도 한다.

도 3을 참조하면, 본 발명의 바람직한 실시예에 따른 자원 그룹들을 포함하는 클러스터 내에 구성 및 상태 정보를 복제하는 과정에 대한 고수준 흐름도를 도시한다. 상기 과정은 단계(302)에서 시작되어, 클러스터 시스템 내의 자원에 대한 구성 또는 상태 데이터의 변화를 예시하고 있다. 그 후, 과정은 단계(304)로 진행하여 상기 변화가 "클러스터-레벨" 변화인지, 또는 클러스터 시스템을 통해서 복제되어야 하는 변화인지를 결정한다. 구성 및 상태 정보에서의 일부 변화, 예컨대 노드의 고장 또는 재통합은 전체 클러스터 시스템을 통해서 복제되어야 한다. 예컨대 클러스터 시스템에 노드가 추가되면, 어느 자원 그룹들이 노드를 포함하는지에 관계없이 모든 기존의 노드는 해당 추가를 반영하도록 업데이트되야

한다. 구성 및 상태 정보 변화가 클러스터 레벨 변화인 경우, 그 과정은 단계(306)로 진행하여 클러스터 시스템을 통하여 그 변화를 복제한다.

구성 및 상태 정보 변화가 클러스터 레벨 변화가 아닌 경우, 과정은 단계(308)로 진행하여 변화에 의해 영향을 받는 자원 그룹 내의 노드 사이에서의 변화를 복제한다. 관련된 자원 그룹 또는 애플리케이션에만 영향을 주는 구성 및 상태 정보 변화는 자원 그룹을 통해서만 복제되어야 한다. 자원 그룹 관리자는 단순히 상위 순위를 갖는 현재의 자원 그룹 내에서 노드가 될 수 있으며, 구성 및 상태 정보 변화의 적절한 복제를 보장하는데 사용된다.

그 후, 과정은 단계(310)로 진행하여, 자원 그룹 내의 노드가 다른 자원 그룹과 공유되는지의 여부를 결정한다. 만일, 자원 그룹 내의 노드가 다른 자원 그룹과 공유되면, 과정은 단계(312)로 진행하여, 나머지 다른 자원 그룹 또는 그룹들 내의 모든 노드들의 구성 및 상태 변화를 복제한다. 상이한 자원 그룹들에 의해 공유되는 노드 또는 노드들은 적절한 복제를 보장하는 책임을 진다. 이 점에 있어서, 클러스터 시스템 내의 인터로킹 자원 그룹들은 구성 및 상태 정보의 추가적 복제를 요구하기 때문에 바람직하지 않다. 그러나, 추가 복제가 필요하지 않기 때문에, 변화에 의해 영향을 받는 자원 그룹과 공통인 노드를 갖지 않은 클러스터 시스템 내에서 변화가 자원 그룹들에 복제될 필요는 없k.

일단 정보가 영향을 받은 자원 그룹과 공통인 적어도 하나의 노드를 갖는 영향을 받은 자원 그룹 또는 자원 그룹 내의 모든 노드들 전부에 완전히 복제되거나, 또는 영향을 받은 자원 그룹이 다른 자원 그룹과 공유되는 임의의 노드를 포함하지 않으면, 과정은 단계(314)로 진행하여, 차후의 구성 및 상태 정보 변화가 검출될 때까지 과정이 아이들 상태가 된다.

도 4를 참조하면, 본 발명의 바람직한 실시예에 따라 자원 그룹을 포함하는 클러스터 시스템 내의 노드 고장을 처리하는 과정에 대한 고수준 흐름도가 도시되어 있다. 과정은 단계(402)에서 시작되며, 자원 그룹 내에 노드의 고장을 나타낸다. 그 후 과정은 단계(404)로 진행하여, 자원 그룹(또는, 고장난 노드가 공유인 경우에는 자원 그룹들)의 "정족수"가 사용 가능한지의 여부를 결정한다. 상기한 바와 같이, 자원 그룹이 정의되는 서비스 또는 서비스들을 신뢰성 있게 제공하기 위해, 자원 그룹 내에서 충분한 자원이 사용 가능한한 정족수가 다수일 필요는 없다.

자원 그룹 내에 노드의 정족수가 사용 가능하면, 과정은 단계(406)로 진행하여 사용 가능한 노드를 활용해서 서비스를 계속 제공한다. 일부 자원의 재할당이 필요할 수도 있다. 그 후, 과정은 단계(408)로 진행하여 고장난 노드가 재저장되어 있는지의 여부를 결정한다. 고장난 노드가 재저장되어 있지 않으면, 단순히 단계(408)로 복귀된다. 그러나, 고장난 노드가 재저장되어 있으면, 과정은 단계(410)로 진행하여 필요에 따라 노드를 재통합하고 자원을 재할당한다.

다시 단계(404)를 참조하면, 노드의 정족수가 사용 가능한 상태가 아닌 경우, 과정은 단계(412)로 진행하여 영향받은 자원 그룹으로부터의 서비스를 일시 중지시킨다. 그 후, 단계(414)로 진행하여, 고장난 노드가 재저장되어 있는지의 여부를 결정한다. 상기한 바와 같이, 고장난 노드가 아직 재저장되어 있지 않은 경우, 과정은 단순히 단계(414)로 복귀된다. 그러나, 일단 고장난 노드가 재저장되면, 과정은 단계(416)으로 진행하여, 노드를 재통합하고 영향을 받은 자원 그룹으로부터의 서비스를 재개한다. 과정은 단계(410) 또는 단계(416) 중 어느 하나의 단계에서 단계(418)로 진행하여, 다른 노드 고장이 발생할 때까지 과정이 아이들 상태가 된다.

발명의 효과

본 발명은 대규모 완전 복제 문제를 완전히 복제된 서브-클러스터 시스템의 집합으로 분해하기 위해 대규모 클러스터 시스템의 국부화(localization) 특징을 사용한 것이다. 기록들은 구성 정보의 해당되는 일부분을 필요로 하는 모든 데이터 처리 시스템들에만 복제된다. 이러한 부분 복제는 복제 및 데이터 조작 비용을 상당히 감소시킨다. 비용은 데이터 처리 시스템의 총 수의 함수가 아니라, 자원 그룹 내에 데이터 처리 시스템 수의 함수만으로 증가한다. 따라서, 본 발명의 관리 형태는 고도의 조정이 가능하며, 1,000개 이상의 데이터 처리 시스템을 갖는 대규모 클러스터 시스템에 적용 가능하다.

본 발명은 완전한 기능적 클러스터 다중-처리 시스템과 관련하여 기술되었지만, 본 발명의 메카니즘은 당업자에 의해 여러 형태의 인스트럭션을 포함하는 컴퓨터 판독 가능 매체의 형태로 배포될 수 있으며, 이러한 배포를 실제로 수행하는데 사용되는 신호 포함 매체의 종류에 관계 없이 동일하게 적용될 수 있는 것을 이해할 수 있다. 컴퓨터 판독 가능 매체의 예로는 판독 전용 메모리들(ROM) 또는 소거 가능형 레모리와 같은 불휘발성, 하드-코드형 매체, 전기적으로 프로그램 가능한 판독 전용 메모리들(EEPROM), 플로피 디스크와 같은 기록 가능형 매체, 하드 디스크 드라이브 및 CD-ROM, 및 디지털 및 아날로그 통신 링크와 같은 전송형 매체를 포함한다.

본 발명은 바람직한 특정 실시예에 대해서 기술 및 도시되어 있지만, 당업자라면 본 발명의 정신 및 범위를 벗어남이 없이 그 형태 및 상세한 내용에 있어서 다양한 변경이 이루어질 수 있다는 것을 이해할 수 있다.

(57) 청구의 범위

청구항 1

클러스터 시스템에 고 가용성 컴퓨팅 서비스를 제공하기 위한 방법에 있어서,

a) 상기 클러스터 시스템 내의 데이터 처리 시스템들을 적어도 하나의 자원 그룹－여기서 각각의 자원 그룹은 적어도 2개의 데이터 처리 시스템과, 각각의 컴퓨팅 서비스를 제공하기 위한 관련 자원들을 포함함－으로 분리하는 단계;

b) 원하는 컴퓨팅 서비스의 실행에 앞서, 원하는 컴퓨팅 서비스를 담당하는 자원 그룹이 정족수 상태에 있는지의 여부를 결정하는 단계; 및

c) 상기 원하는 서비스의 제공을 담당하는 자원 그룹이 정족수 상태에 있다는 결정에 응답하여, 원하는 컴퓨팅 서비스를 제공하는 단계

를 포함하는 클러스터 시스템에 고 가용성 컴퓨팅 서비스를 제공하기 위한 방법.

청구항 2

제 1 항에 있어서,

상기 원하는 컴퓨팅 서비스를 담당하는 자원 그룹이 정족수 상태에 있는지의 여부를 결정하는 단계는 상기 자원 그룹 내에 다수의 데이터 처리 시스템들이 온라인 상태인지의 여부를 결정하는 단계를 추가로 포함하는 클러스터 시스템에 고 가용성 컴퓨팅 서비스를 제공하기 위한 방법.

청구항 3

제 1 항에 있어서,

상기 원하는 컴퓨팅 서비스를 담당하는 자원 그룹이 정족수 상태에 있는지의 여부를 결정하는 단계는 상기 자원 그룹 내에 적어도 하나의 데이터 처리 시스템이 온라인 상태인지의 여부를 결정하는 단계를 추가로 포함하는 클러스터 시스템에 고 가용성 컴퓨팅 서비스를 제공하기 위한 방법.

청구항 4

제 1 항에 있어서,

상기 클러스터 시스템 내에 데이터 처리 시스템의 고장을 검출하는 단계를 추가로 포함하는 클러스터 시스템에 고 가용성 컴퓨팅 서비스를 제공하기 위한 방법.

청구항 5

제 4 항에 있어서,

상기 자원 그룹이 고장난 데이터 처리 시스템을 포함하는지의 여부를 결정하는 단계를 추가로 포함하는 클러스터 시스템에 고 가용성 컴퓨팅 서비스를 제공하기 위한 방법.

청구항 6

제 5 항에 있어서,

상기 고장난 데이터 처리 시스템을 포함하는 상기 클러스터 시스템 내의 모든 자원 그룹들을 식별하는 단계를 추가로 포함하는 클러스터 시스템에 고 가용성 컴퓨팅 서비스를 제공하기 위한 방법.

청구항 7

클러스터 시스템에 있어서,

a) 복수의 자원 그룹－여기서 각각의 자원 그룹은 적어도 2개의 데이터 처리 시스템을 포함하여 각각의 컴퓨팅 서비스를 제공함－로 분리되는 복수의 데이터 처리 시스템들;

b) 상기 클러스터 시스템 내에 상기 데이터 처리 시스템들을 연결하는 적어도 하나의 네트워크; 및

c) 상기 클러스터 시스템 내의 데이터 처리 시스템의 고장에 응답하여, 고장난 데이터 처리 시스템을 포함하는 모든 자원 그룹을 식별하고, 고장난 데이터 처리 시스템을 포함하는 각각의 식별된 자원 그룹에 대해서 상기 각각의 식별된 자원 그룹에 정족수가 존재하는지의 여부를 결정하는 고장 메카니즘

을 포함하고,

각각의 상기 각각의 식별된 자원 그룹은 정족수가 가용성이면 컴퓨팅 서비스를 제공하는

클러스터 시스템.

청구항 8

제 7 항에 있어서,

상기 자원 그룹의 정족수가 비가용성이면, 고장난 데이터 처리 시스템을 포함하는 자원 그룹에 의해 제공되는 컴퓨팅 서비스를 일시 중지시키는 수단을 추가로 포함하는 클러스터 시스템.

청구항 9

제 7 항에 있어서,

복구시 상기 고장난 데이터 처리 시스템을 재통합시키는 수단을 추가로 포함하는 클러스터 시스템.

청구항 10

제 7 항에 있어서,

데이터 처리 시스템들 중에 분산된 구성 데이터베이스를 추가로 포함하고, 상기 클러스터 시스템 내의

각각의 데이터 처리 시스템은 클러스터-레벨 구성 및 상태 정보와 데이터 처리 시스템을 포함하는 모든 자원 그룹에 대한 자원 그룹 구성 및 상태 정보를 포함하지만, 상기 데이터 처리 시스템을 포함하지 않는 임의의 자원 그룹에 대한 자원 그룹 구성 및 상태 정보는 포함하지 않는 클러스터 시스템.

청구항 11

데이터 처리 시스템에 있어서,

a) 컴퓨팅 서비스를 제공하는 인스트럭션을 실행하는 프로세서;

b) 상기 데이터 처리 시스템이 복수의 자원 그룹으로부터 분리되는 클러스터 시스템에 연결되도록 하는 네트워크 연결;

c) 상기 데이터 처리 시스템을 포함하는 클러스터 시스템 내에 각각의 자원 그룹을 식별하는 구성 정보를 포함하는 메모리; 및

d) 상기 데이터 처리 시스템을 포함하는 자원 그룹 내의 임의의 다른 데이터 처리 시스템의 고장을 검출하는 고장 메카니즘－여기서 고장 메카니즘은 고장난 데이터 처리 시스템을 포함하는 자원 그룹이 정족수 상태에 있는지의 여부를 결정하고, 고장난 데이터 처리 시스템을 포함하는 자원 그룹이 정족수 상태에 있으면 데이터 처리 시스템이 컴퓨팅 서비스를 계속 제공하도록 함 －

을 추가로 포함하는 데이터 처리 시스템.

청구항 12

제 11 항에 있어서,

상기 고장난 데이터 처리 시스템을 포함하는 자원 그룹이 정족수 상태에 있지 않으면, 컴퓨팅 서비스를 일시 중지시키는 수단을 추가로 포함하는 데이터 처리 시스템.

청구항 13

제 11 항에 있어서,

상기 고장난 데이터 처리 시스템을 포함하는 자원 그룹이 정족수 상태에 있으면, 데이터 처리 시스템으로부터 컴퓨팅 서비스 요구를 제공하는 수단을 추가로 포함하는 데이터 처리 시스템.

청구항 14

제 11 항에 있어서,

상기 컴퓨팅 서비스가 고 가용성 애플리케이션을 포함하는 데이터 처리 시스템.

청구항 15

각각이 적어도 2개의 데이터 처리 시스템과 애플리케이션 서버를 포함하는 복수의 자원 그룹으로 분리되는 클러스터 시스템 내의 데이터 처리 시스템 고장에 응답하는 방법에 있어서,

a) 상기 고장난 데이터 처리 시스템을 포함하는 모든 자원 그룹을 식별하는 단계;

b) 상기 고장난 데이터 처리 시스템을 포함하는 각각의 자원 그룹에 대해, 자원 그룹이 정족수 상태에 있는지의 여부를 결정하는 단계;

c) 상기 고장난 데이터 처리 시스템을 포함하는 각각의 자원 그룹이 정족수 상태에 있으면, 애플리케이션 서버로 요구를 제공하는 단계; 및

d) 상기 고장난 데이터 처리 시스템을 포함하는 각각의 자원 그룹이 정족수 상태에 있지 않으면, 애플리케이션 서버를 일시 중지시키는 단계

를 포함하는 데이터 처리 시스템 고장에 응답하는 방법.

청구항 16

제 15 항에 있어서,

상기 자원 그룹이 정족수 상태에 있는지의 여부를 결정하는 단계는 상기 자원 그룹 내의 복수의 데이터 처리 시스템들이 온라인 상태인지의 여부를 결정하는 단계를 추가로 포함하는 데이터 처리 시스템 고장에 응답하는 방법.

청구항 17

제 15 항에 있어서,

상기 자원 그룹이 정족수 상태에 있는지의 여부를 결정하는 단계는 상기 자원 그룹 내에 적어도 하나의 데이터 처리 시스템이 온라인 상태에 있는지의 여부를 결정하는 단계를 추가로 포함하는 데이터 처리 시스템 고장에 응답하는 방법.

청구항 18

제 15 항에 있어서,

상기 자원 그룹이 정족수 상태에 있는지의 여부를 결정하는 단계는 상기 자원 그룹에 연관된 애플리케이

션 서버로 요구를 제공하기 위해, 상기 자원 그룹이 충분한 기능 자원을 포함하는지의 여부를 결정하는 단계를 추가로 포함하는 데이터 처리 시스템 고장에 응답하는 방법.

청구항 19

컴퓨터 사용 가능 매체 내의 컴퓨터 프로그램 제품에 있어서,

a) 네트워크 내의 데이터 처리 시스템들을 적어도 하나의 자원 그룹－각각의 자원 그룹은 적어도 2개의 데이터 처리 시스템과 각각의 컴퓨팅 서비스를 제공하는 관련 자원들을 포함함－으로 분리시키는 인스트럭션;

b) 원하는 컴퓨팅 서비스의 실행에 앞서 원하는 컴퓨팅 서비스를 담당하는 자원 그룹이 정족수 상태에 있는지의 여부를 결정하는 인스트럭션; 및

c) 원하는 서비스의 제공을 담당하는 자원 그룹이 정족수 상태에 있다는 결정에 응답하여 원하는 컴퓨팅 서비스를 제공하는, 컴퓨터 사용 가능 매체 내의 인스트럭션

을 포함하는 컴퓨터 프로그램 제품.

청구항 20

제 19 항에 있어서,

상기 원하는 컴퓨팅 서비스를 담당하는 자원 그룹이 정족수 상태에 있는지의 여부를 결정하는 인스트럭션은 상기 자원 그룹 내의 다수의 데이터 처리 시스템들이 온라인 상태인지의 여부를 결정하는 인스트럭션을 추가로 포함하는 컴퓨터 프로그램 제품.

청구항 21

제 19 항에 있어서,

상기 원하는 컴퓨팅 서비스를 담당하는 자원 그룹이 정족수 상태에 있는지의 여부를 결정하는 인스트럭션은 상기 자원 그룹 내에 적어도 하나의 데이터 처리 시스템이 온라인 상태인지의 여부를 결정하는 명령을 추가로 포함하는 컴퓨터 프로그램 제품.

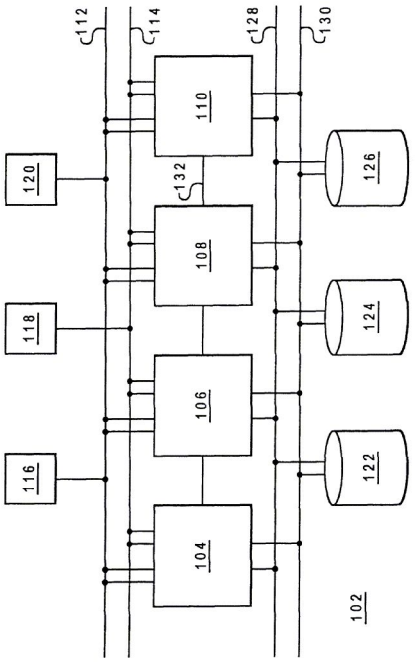
청구항 22

제 19 항에 있어서,

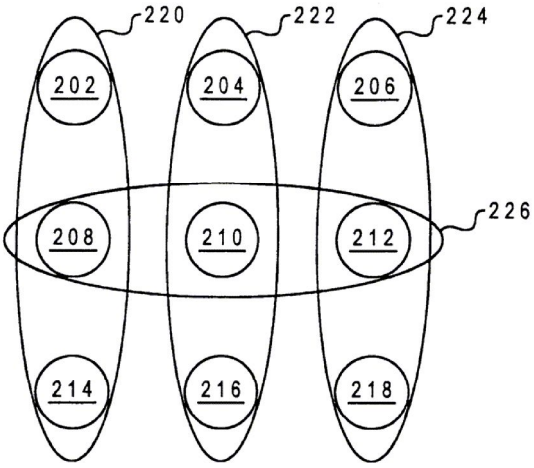
상기 원하는 컴퓨팅 서비스를 담당하는 자원 그룹이 정족수 상태에 있는지의 여부를 결정하는 인스트럭션은 상기 원하는 컴퓨팅 자원들에 대한 요구를 제공하기 위해서 충분한 기능 자원을 포함하며, 원하는 컴퓨팅 서비스의 제공을 담당하는 자원 그룹을 결정하는 인스트럭션을 추가로 포함하는 컴퓨터 프로그램 제품.

도면

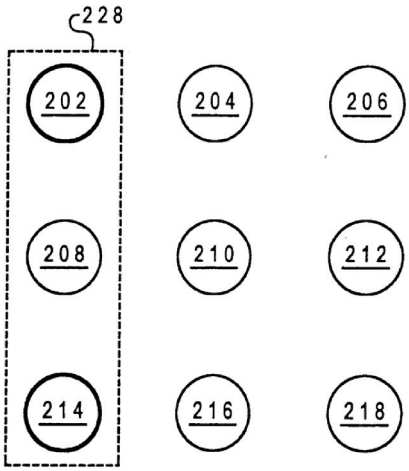
도면1



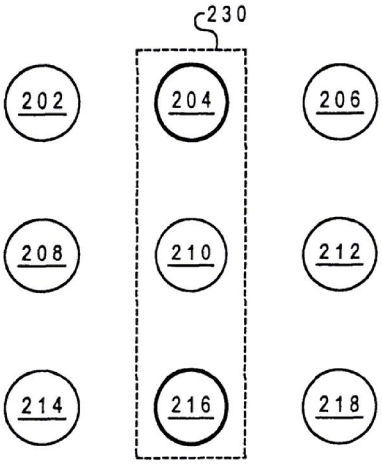
도면2a



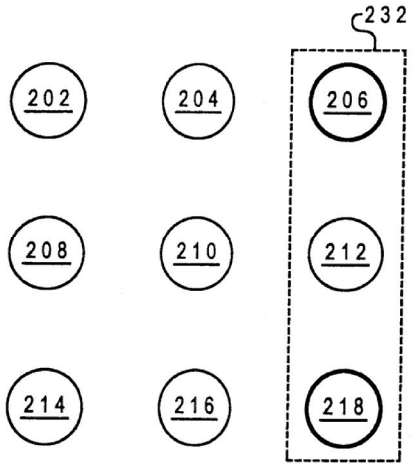
도면2b



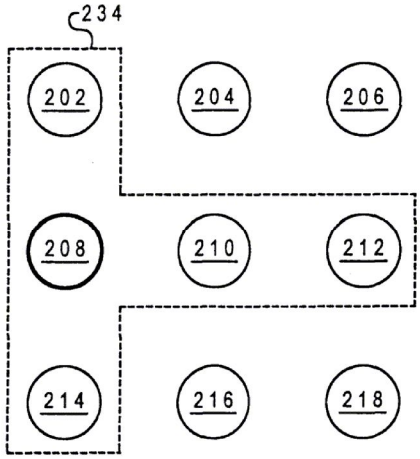
도면2c



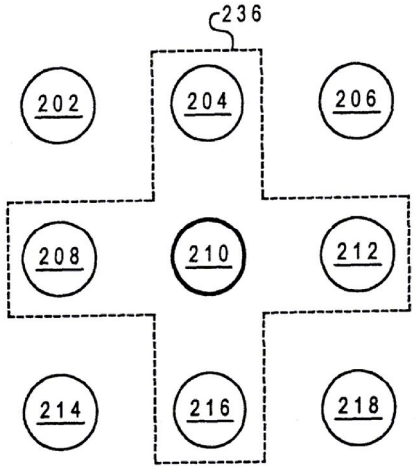
도면2d



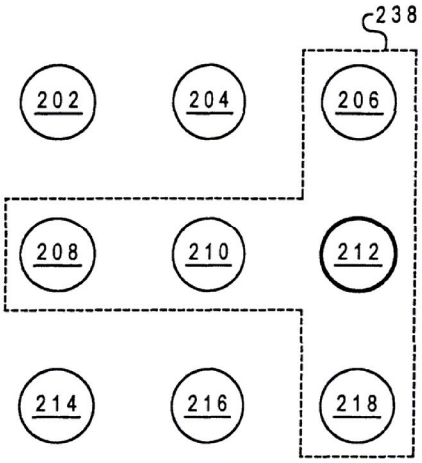
도면2e



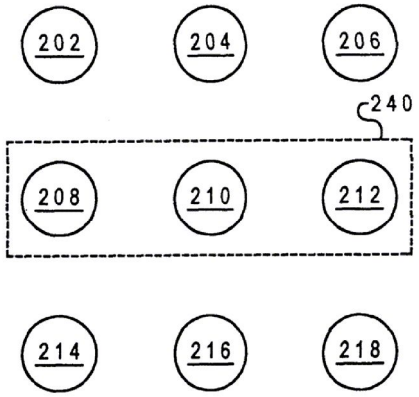
도면2f



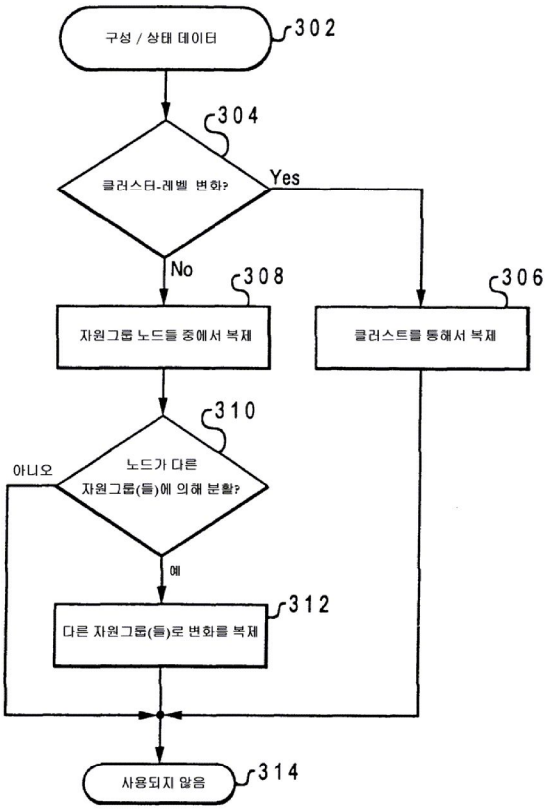
도면2g


238
202
204
206
208
210
212
214
216
218

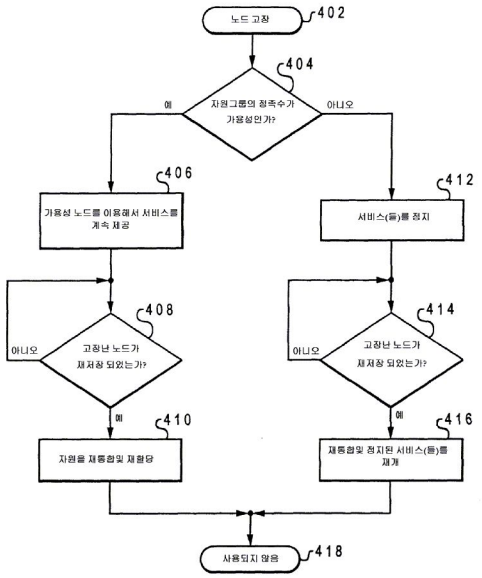
도면2h


202
204
206
240
208
210
212
214
216
218

도면3


구성 / 상태 데이터
302
304
클러스터-레벨 변화?
Yes
No
308
306
자원그룹 노드들 중에서 복제
클러스트를 통해서 복제
310
노드가 다른
자원그룹(들)에 의해 분할?
아니오
예
312
다른 자원그룹(들)로 변화를 복제
사용되지 않음
314

도면4


노드 고장
402
404
자원그룹의 정족수가
가용성인가?
예
아니오
406
412
가용성 노드를 이용해서 서비스를
계속 제공
서비스(들)를 정지
408
414
고장난 노드가
재저장 되었는가?
아니오
예
410
416
자원을 재통합및 재할당
재통합및 정지된 서비스(들)를
재개
사용되지 않음
418