(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

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

(45) 공고일자 1999년12월01일
(11) 등록번호 10-0232247
(24) 등록일자 1999년09월03일

(21) 출원번호 10-1995-0036939
(22) 출원일자 1995년10월25일
(65) 공개번호 특1996-0015278
(43) 공개일자 1996년05월22일
(30) 우선권주장 332,157 1994년10월31일 미국(US)

(73) 특허권자 인터내셔널 비지네스 머신즈 코포레이션 포만 제프리 엘
미국 10504 뉴욕주 아몬크
(72) 발명자 클레멘트 리차드 아타나시오
미국 뉴욕 10566 피크스킬 디 루보 드라이브 5
마리아 안젤라 부트리코
미국 뉴욕 10913 블러벨트 반 위크 로드 54
제임스 라일 패터슨
미국 텍사스 78759-5108 오스틴 바커 릿지 코브 10601
크리스토스 알키비아디스 폴리조이스
미국 뉴욕 10606-1935 화이트 플레인스 마킨 애비뉴 25 아파트 PH-105
스테판 에드윈 스미스
미국 뉴욕 10541 마오팩 해트필드 로드 19
(74) 대리인 이병호

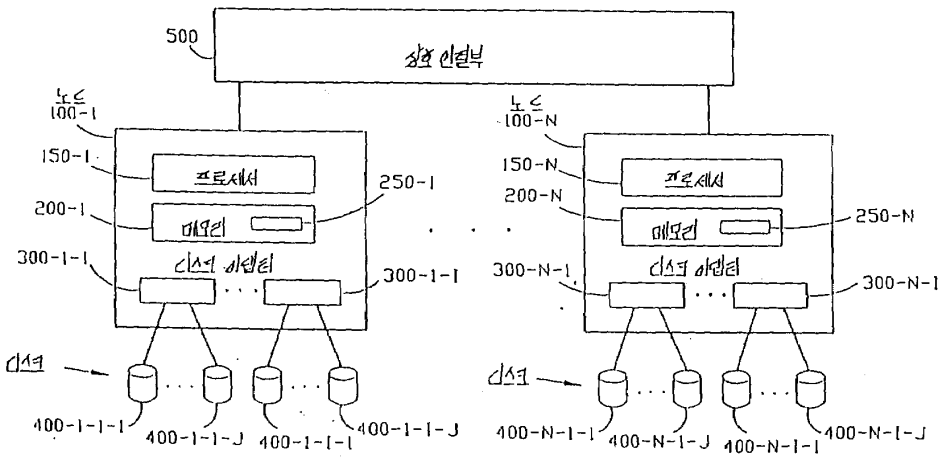
심사관 : 안철흥

(54) 클러스터화된 다중처리 시스템 및 시스템내 디스크 액세스 경로의 고장 회복 방법

요약

클러스터화된 계산기 시스템의 디스크 액세스 경로 고장으로부터 회복하기 위한 시스템 및 방법. 상기 클러스터화 된 계산기 시스템의 각 노드에는 그 노드에서 실행되는 응용들로부터의 물리적 디스크 액세스 요구들을 처리하고 또한 상기 디스크 액세스 요구들을 그 디스크가 물리적으로 연결되어 있는 적절한 서버로 향하게 하기 위한 대리 소프트웨어가 제공된다. 각 노드상의 상기 대리 소프트웨어는 그 노드로부터 나온 모든 펜딩된 요구들을 위한 상태 정보를 유지한다. 디스크 액세스 경로상의 고장 검출에 응답하여, 모든 노드들에 있는 대리 소프트 웨어는 디스크 액세스에 대한 계속적인 모든 요구들을 상기 디스크에 물리적으로 연결 제2노드로 향하게 한다.

대표도



명세서

[발명의 명칭]

클러스터화된 다중처리 시스템 및 시스템내 디스크 액세스 경로의 고장 회복 방법

[도면의 간단한 설명]

제1도는 본 발명의 바람직한 일실시예에 대한 전체적 블럭도.

제2도는 트윈-테일 디스크의 바람직한 구조를 보이는 도면.

제3도는 클라이언트 노드에서 요구를 처리하는데 관련되는 단계들을 보이는 흐름도.

제4도는 서버 노드에서 요구를 처리하는데 관련되는 단계들을 보이는 흐름도.

제5도는 조정자 노드에서 복구에 관련되는 단계들을 보이는 흐름도.

제6도는 참가자 노드에서 복구에 관련되는 단계들을 보이는 도면.

제7도는 가상 공유 디스크들에 관련된 데이터 및 메모리 상주 논리에 대한 상세한 블럭도.

* 도면의 주요부분에 대한 부호의 설명

100 : 노드	150 : 프로세서
200 : 메모리	300 : 디스크 어댑터
400 : 디스크	500 : 상호 연결부

[발명의 상세한 설명]

[발명의 배경]

[발명의 분야]

본 발명은 일반적으로 분산 컴퓨터 환경(distributed computing environment)에 관한 것이다. 좀더 구체적으로는, 클러스터내의 각 프로세서가 그 클러스터내의 어떠한 디스크도 액세스할 수 있는 프로세서들의 클러스터에서의 사용 방법에 관한 것이다.

[관련기술]

고성능 마이크로프로세서들에 대한 가용성(availability)은 클러스터(cluster)를 모놀리식 시스템(monolithic system)에 대한 매력적인 대안이 되게 만들었다. 자신의 계산(computation)을 여러 노드들로 분할시킬 수 있는 응용들은 이러한 구조를 이용할 수 있는데, 이러한 구조는 일반적으로 모놀리식 시스템보다 더 우수한 가격 대 성능비를 제공한다. 그러한 응용들은 대량의 과학적 계산, 데이터 베이스 및 트랙잭션 처리(transaction processing) 시스템, 판단 지원 시스템 등을 포함한다.

마이크로프로세서 클러스터(microprocessor cluster)는 네트워크나 통신 스위치와 같은 프로세서간 통신 메커니즘에 결합된 다수의 개별적인 계산기 시스템(computing system)으로 구성된다. 각 계산기 시스템은 자신의 프로세서, 메모리, I/O 서브시스템을 가지며 별도의 운영 체제(operating system)를 실행한다.

그러나 최대의 효과를 얻기 위해서는 응용이 특정 계산기 시스템으로부터 분리되어 클러스터내의 모든 노드들을 동등하게 다룰수 있는 것이 바람직하다. 이러한 능력은 때로 “단일 시스템 이미지(single system image)”라 불린다.

단일 시스템 이미지의 유용한 점으로는 동일한 I/O 디바이스 자원들이 클러스터내 모든 프로세서들에게 동등하게 이용가능해야 한다는 요건이다. 이는 처리 작업들이 프로세서들 사이에서 자유롭게 이동될 수 있게 한다. 게다가 이는 자신들의 계산을 위해 데이터 공유 모델(data sharing model)을 채택하는 병렬 응용들(parallel applications)의 개발을 용이하게 한다.

바람직하게는 가용성을 높이기 위해, 모든 프로세서들에게 동일한 I/O 자원들을 제공하기 위한 많은 다양한 접근법들이 있을 수 있다. 데이터 복제(data replication)는 가장 간단한 형태(특히 판독 전용 데이터인 경우)이지만, 이 방법은 비용(공유되지 않는 자원들)을 증가시키며 시간의 경과에 따라 정보가 변경되는 경우는 곤란한 점들을 드러낸다.

대안적인 방법의 하나로는 여러 프로세서들에 물리적으로 부착될 수 있는 디바이스들을 구비하는 것이다. 예를 들면 트윈-테일(twin-tailed)(이중 포트형)디스크들이 보편적인 것이다. 4-테일 디스크 및 8-테일 디스크까지 만들 수 있기는 하지만 이들은 비용도 증가시키고 동작도 난해하다.

상기 두 경우 모두에 있어, 각 프로세서는 자원들에 대해 독립적인 액세스를 가지므로, 프로세서 및/또는 어댑터의 고장이 발생한 경우에 데이터의 연속적인 액세스를 위해 아무런 조처도 제공할 필요가 없다.

NFS, AFS, DFS와 같은 분산 화일 시스템(distributed file system)은 특정 I/O 디바이스를 그 디바이스에 의도된 서비스로부터 분리시키고, 이러한 서비스들을 클러스터내 프로세서들에게 제공한다. 이는 상기 디바이스의 사용을 상기 서비스들로 제한하여, 메모리 계층내에서 데이터의 위치를 명시적으로 알고 있는 응용들에는 적합하지 않게 한다. 예를 들어, 데이터 베이스 시스템은 데이터 버퍼링이나 데이터 배열등의 서비스를 제공하는 화일 시스템에 의존하기 보다는 자기 자신의 버퍼링에 의존할 수도 있고 디스크상에 데이터를 자신만의 방식으로 배열하기를 원할 수도 있다. 이러한 경우, 응용은 I/O 디바이스에 대한 직접적인 액세스를 선호할 수도 있다.

높은 가용성면에서 보자면, HA-NFS는 NFS 고객들에게 가용성이 높은 NFS서버를 제공하지만 높은 가용성을 가능하게 하는 중요한 기능들을 제공하기 위해서는 그 기저에 깔린 네트워크 기술(IP 어드레스 인수

(takeover))에 지나치게 의존한다.

[발명의 개요]

본 발명의 목적은, 모든 I/O 디바이스가 모든 노드에 부착됨을 요구하지 않고도 클러스터내 프로세서들로 하여금 I/O 디바이스들을 공유할 수 있게 하는 시스템에서, 노드 및/또는 어댑터 고장시에 투명한 복구(transparent recovery)를 제공하며 응용들로 하여금 생존 노드들(surviving nodes)에서 실행되게 하여 상기 고장에도 불구하고 처리를 계속할 수 있게 하는 것이다. 상기 투명성은 고장이 나기 전에 발송한 어떠한 요구들도 응용이 재발송할 필요가 없음을 의미한다.

따라서 본 발명은 클러스터화된 계산기 시스템의 디스크 액세스 경로내 고장들로부터 회복시키기 위한 시스템 및 방법을 제공한다. 클러스터화된 계산기 시스템의 각 노드에는, 그 노드에서 실행되는 응용들로부터의 물리적 디스크 액세스 요구들을 처리하고 상기 디스크 액세스 요구들을 그 디스크가 물리적으로 연결된 적절한 서버로 향하게 하기 위한 대리 소프트웨어(proxy software)가 제공된다.

상기 각 노드상의 대리 소프트웨어는 그 노드로부터 개시되는 모든 펜딩된 요구들을 위한 상태 정보를 유지한다. 디스크 액세스 경로(예를 들면 노드 내부 또는 디스크 어댑터)의 고장 검출에 응답하여, 모든 노드들상에 있는 대리 소프트웨어는 또다른 모든 디스크 액세스 요구들을 동일한 디스크에 물리적으로 연결된 제2노드로 향하게 한다.

본 발명의 바람직한 일 실시예에 있어서, 상기 대리 소프트웨어(proxy software)는 하나의 소프트웨어 계층으로 구현되는데, 이는 가상 디바이스들을 정의하고 이 디바이스들로의 I/O 요구들을 인터셉트하고 이 요구들(기록을 위해서는, 데이터도 포함하여)을 실제디바이스가 물리적으로 연결되어 있는 적절한 서버 프로세서에게 경로설정함으로써 프로세서들로 하여금 원격 프로세서들에 물리적으로 연결된 I/O 디바이스들을 액세스할 수 있게 한다. 서버 프로세서(server processor)는 실제의 I/O 동작을 수행하며 종료 메시지(판독을 위해서는, 데이터도 포함하여)를 개시 프로세서(originating processor)에게 되돌려 준다. 상기 종료 메세지를 받자 마자, 개시 프로세서는 그에 따라 상기 요구를 발송한 프로세스에게 알려준다.

트윈-테일 디스크를 사용하면, 다음과 같이 높은 가용성을 얻을 수 있다.

특정한 한 디스크에 대하여 그 디스크에 연결된 프로세서들 중 하나가 1차 서버(primlary server)로 지정된다. 정상적인 동작 동안에는 클러스터내 임의의 장소에서 개시된 상기 디스크에 대한 I/O 요구가 1차 서버로 보내진다. 만약 1차 서버나 그의 디스크 어댑터가 고장나면 그 디스크에 연결된 나머지 프로세서들 중 하나가 상기 디스크를 위한 1차 서버가 되며, 각 프로세서에 대한 요구 경로설정 정보(request routing information)는 변경되어 새로운 요구들은 새로운 1차 서버로 전송되도록 한다.

본 실시예에 있어서, 상기 서버는 완전히 상태가 없다(stateless). 즉, 펜딩된 원격 요구들의 모든 상태는 클라이언트상에서 유지된다. 그러므로 서버 또는 어댑터가 고장나는 경우, 고장전에 발송되었던 펜딩된 요구들은 클라이언트에 의해 새로운 서버로 재발송될 수 있으며 응용들은 상기 고장을 절대 알지 못한다.

[바람직한 실시예에 대한 상세한 설명]

제1도는 본 발명의 바람직한 일실시예에 대한 블럭도로서, 회복가능한 가상 공유 디스크들의 서브시스템을 포함한다. 이는 독립적인 계산 노드(computing node)(이하 노드로 칭함)(100-1 내지 100-N)들의 집합(클러스터)을 포함한다.

각 노드는 노드(100-1로 표시)를 위한 프로세서(150-1로 표시)와 메모리(200-1로 표시)를 구비한다(따라서, 노드(100-N)는 프로세서(150-N)와 메모리(200-N)를 구비한다). 본 기술분야의 당업자라면 각 노드가 별도의 메모리를 가질 수도 있고 노드들이 메모리를 공유할 수도 있음을 쉽게 인식할 것이다.

노드들은 상호 연결부(interconnection)(500)을 통해 통신할 수 있다. 이 상호 연결부는 스위치, 근거리 통신망, 공유 메모리 또는 클러스터내 노드들로 하여금 데이터를 교환할 수 있게 하는 어떠한 종류의 매체도 가능하다.

각 노드는 노드(100-1)를 위해 다수의 디스크 어댑터(300-1-1 내지 300-1-I로 표시)를 구비한다(따라서 노드(100-N)를 위해서는 (300-N-1 내지 300-N-I)로 표시되는 디스크 어댑터들을 구비). (400-1-1-1 내지 400-1-1-J)로 표시된 디스크들은 어댑터(300-1-1)에 부착된다(따라서, (400-N-I-1 내지 400-N-I-J)로 표시된 디스크들은 어댑터(300-N-I)에 부착된다). 노드당 디스크 어댑터들의 갯수는 모든 노드들이 동일할 필요가 없다. 또한 어댑터당 디스크들의 갯수도 모든 디스크 어댑터들이 동일할 필요가 없다. 어떤 노드들은 심지어 디스크 어댑터를 전혀 가지지 않을 수도 있다.

다수의 노드들에 의해 공유되는 디스크들은 직접적으로 연결된 물리적 디스크들을 어드레스하기 위해 노드에 의해 사용되는 동일한 프로그래밍 인터페이스를 이용함으로써 클러스터 전체를 통해 하나의 공통 명칭으로 어드레스된다. 이는 디스크들이 클러스터내 각 노드에 물리적으로 연결되었다는 착각(illusion)을 갖게 한다. 상기와 같은 액세스를 가능하게 하는 소프트웨어 및 프로그래밍 인터페이스를 가상 공유 디스크(Virtual shared disk)라 부른다.

각 프로세서의 메모리는 가상 공유 디스크에 관한 대리 논리(proxy logic) 및 상태 데이터(state data)를 포함한다. 상기 상태 데이터는 물리적으로 연결된 디스크들을 위해 일반적으로 운영 체제에 의해 유지되는 유형의 데이터(예를 들면, 디바이스 상태, 디바이스 명칭, 펜딩된 요구 정보)뿐 아니라 후술하는 몇가지 부가적인 정보들도 포함한다. 상기 논리 및 관련 데이터는 노드(100-1)에 대해 블럭(250-1)로 보여진다(따라서, 노드(100-N)에 대해서는 블럭(250-N)), 노드(100-K)를 위한 위와 같은 블럭이 제7도에 상세히 보여진다. 상기 대리 논리는 제7도의 블럭(250-K-A)으로서 보여진다.

노드 또는 어댑터가 고장난 경우에도 계속 액세스 가능하도록 유지되어야 하는 디스크들은 서로 다른 노드들상에서 하나 이상의 어댑터에 연결된다. 제2도는 트윈-테일 디스크의 구조를 보이는데, 여기서 디스크(400-L-P-X)는 노드(100-L)상의 어댑터(300-L-P) 및 노드(100-M)상의 어댑터 (300-M-Q)에 연결된다.

정상적인 동작 동안에는, 모든 디스크마다 그 테일들중 하나가 1차 테일(primary tail)로서 선택된다. 각 노드는 시스템내 모든 가상 디스크를 현재 1차 테일을 보유하는 노드로 매핑시키는 테이블(제7도의 블럭 250-K-B)을 가진다.

상기 1차 테일은 사용되는 유일한 테일이다. 디스크의 나머지 테일은 대기(stand-by)중이다.

임의의 노드에서 실행되는 응용들은, 마치 모든 디스크들이 국부적으로 연결되어 있는 것처럼 임의의 디스크에 대한 I/O 요구도 발송할 수 있다. 개시 노드(node of origin)에서 요구를 처리하는 논리가 제3도에 보여진다. 요구가 발송되면(블럭 700), 어떤 노드가 1차 테일을 가졌는가를 판별하기 위해서 전술한 맵(테이블)(250-K-B)이 검사된다. 만약 개시 노드가 또한 서버 노드(즉 1차 테일을 보유하는 노드)라면, 상기 요구는 국부적으로 서비스된다(블럭 715). 만약 서버 노드가 개시 노드와 다르다면 서버 노드에 요구 설명자(request descriptor)가 전송된다. 만약 상기 요구가 기록 요구라면(블럭 730에서 판별됨), 기록될 데이터도 서버에 전송된다(블럭 740).

블럭(750)에서 상기 요구(판독이건 기록이건)는 원격 서버로부터의 응답을 기다린다. 응답이 도착할 때, 만약 상기 요구가 판독(블럭 760에서 판별됨)이었다면 네트워크에 도착한 데이터는 개시 요구에 주어진다(블럭 770). 만약 상기 요구가 판독이 아니었다면 요구는 종료된다(블럭 780).

요구 설명자는 운영 체제가 통상적으로 물리적 디스크 디바이스 드라이버에 보내는 데이터와 동일한 유형의 데이터(예를 들면, 디바이스 명칭, 오프셋, 요구의 크기, 옵션 플래그, 요구의 유형)를 포함할 뿐 아니라 개시 노드, 특정한 요구 식별자, 1차 노드의 어드레스와 같은 부가적인 데이터도 포함한다.

서버 노드에서 요구를 처리하는 논리가 제4도에 보여진다. 요구(블럭 800)는 국부적으로 실행되는 프로세스에 의해 발송된 것일 수도 있고, 원격 노드로 부터 네트워크에 도착한 것일 수도 있다. 블럭(810)에서 액세스 요구는 디바이스에 발송된다. 블럭(820)에서 상기 논리는 요구의 소스(source)를 판별한다. 만약 상기 요구가 국부적으로 생긴 것이라면 I/O가 종료되자 마자 동작이 종료되고 개시 프로세스(originating process)가 이를 통보받는다. 만약 요구가 다른 노드에서 개시된 것이라면, 블럭(830)에서 응답이 그 개시 노드로 되돌려 보내진다. 만약 상기 요구가 판독이라면 판독된 데이터도 전송된다. 응답이 개시 노드에 도착하면 동작은 종료되며 개시 프로세스가 이를 통보받는다.

프로세서나 어댑터의 고장이 발생하면, 이는 종래의 방식으로 검출되며 그에 따라 모든 노드들이 통보를 받는다. 본 기술분야의 당업자라면 고장을 검출하는데 여러가지 메커니즘들(예를 들면, 주기적인 건강 진단에 근거하여)이 사용될 수 있음을 인식할 것이다.

노드가 고장인 경우에 영향을 받는 가상 공유 디스크들은 그 기초를 이루는 물리적 디바이스를 위해 상기 고장난 노드가 1차 테일 역할을 하는 디스크들이다.

어댑터가 고장난 경우에 영향을 받는 가상 공유 디스크들은 그 기초를 이루는 물리적 디바이스들의 1차 테일이 상기 고장난 어댑터에 연결된 디스크들이다. 상기 영향을 받은 모든 가상 공유 디스크들을 위해, 그 기초를 이루는 물리적 디바이스의 나머지 테일이 새로운 1차 테일로 선택된다. 이 선택은 소정의 우선순위에 기초하여 통계적으로 이루어지거나 또는 런-타임 정보를 사용하는 정책 모듈(policy module)에 의해 동적으로 이루어 질 수 있다. 상기 동적인 선택 논리(dynamic selection logic)는 남아 있는 활성 테일들(active tails)간에 부하 균형(load balancing)을 이루도록 의도될 수 있다.

클러스터내 노드들중 하나가 조정자(coordinator)로 지정된다. 이 조정자는 클러스터내에 있는 모든 노드들에게 고장에 대해 알려줄 책임이 있다. 조정자에 의해 실행되는 논리가 제5도에 보여진다. 각 참가자 노드(participant node)를 위한 논리가 제6도에 보여진다. 상기 조정자도 또한 참가자이다.

고장을 검출하면(블럭 900), 조정자는 모든 참가자들에게 고장의 영향을 받는 가상 공유 디스크들을 중단(suspend)시킬 것을 알리는 메세지를 방송한다(블럭 910). 이 메세지를 수신하면(블럭 1000), 각 참가자는 상기 영향받은 가상 공유디스크들을 중단시킨다(블럭 1010). 가상 공유 디스크의 중단(suspension)이란 가상 디바이스가 일시적으로 아무런 1차 테일도 갖지 않는 것으로 표시된다는 것을 뜻한다. 고장난 서버에 전송된 펜딩된 요구(pending request)들은 개시 클라이언트(client of origin)에 의해 상기 요구의 저장 목적으로 제공된 큐에 저장된다. 가상 디바이스가 중단된 동안 도착하는 요구들도 개시 클라이언트에 의해 동일한 큐에 저장된다.

고장의 영향을 받은 디바이스들이 중단된 다음에 각 참가자는 조정자에게 애크놀리지(acknowledgement)를 보낸 후(블럭 1020), 조정자로부터 재개 메세지(resume message)를 수신할 때까지 대기한다(즉, 영향받은 VSD 에 대해 더 이상의 동작을 취하지 않는다)(블럭 1030). 다른 처리는 영향을 받지 않는다. 조정자는 모든 참가자로부터의 응답을 기다리며(블럭 920), 그 후 상기 영향받은 가상 공유 디스크를 재개시키는 메세지를 모든 참가자들에게 방송한다(블럭 930). 이 메세지를 받으면, 각 참가자들은 영향받은 가상 디바이스들을 재개시킨다(블럭 1040).

가상 디바이스의 재개(resumption)란, 선택된 새로운 1차 테일을 보유하고 있는 노드가 그 가상 디바이스를 위한 목적지 맵(destination map)에 기록된다는 것을 의미한다. 재개후에 참가자는 조정자에게 애크놀리지를 보내고(블럭 1050), 모든 펜딩된 요구들(중단전에 펜딩된 것뿐 아니라 중단 기간동안 시작된 요구도)을 그 디바이스를 위한 새로운 서버로 재발송한다(블럭 1060). 조정자는 모든 노드들로 부터 2차 애크놀리지를 받는다(블럭 940).

고장의 영향을 받은 가상 공유 디스크들의 중단 및 재개를 달성하기 위해, 전술한 두 단계 수행(two-phase commit)의 변형 대신에 협약 프로토콜(agreement protocol)들이 사용될 수 있다. 더 나아가 조정자 고장의 경우는 회복의 조정을 수행하기 위한 다른 조정자가 선택될 수 있다.

빈번하게 또는 최근에 액세스된 데이터를 가진 버퍼는 디스크의 1차 테일을 보유한 노드에 있는 메모리에 유지될 수 있다. 만약 요구된 데이터가 메모리에서 이용가능하다면 버퍼링된 메모리 사본이 사용된다. 만약 이용가능하지 않다면 물리적인 디스크 액세스가 일어나야만 한다.

트윈-테일 또는 일반적인 다중-테일 디스크(multi- tailed disk)에 대해, 디스크에 물리적으로 연결된 어떠한 노드도 서버의 역할을 할 수 있다. 더 나아가, 그 디스크에 물리적으로 연결된 프로세서들의 임의의 부분 집합이 동시에 서버 역할을 할 수 있다. 즉, 하나 이상의 1차 테일들이 동시에 활성화(active)될 수 있다. 특정 디스크에 연결되지 않은 노드들은 상기 요구를 자신의 활성 서버중 하나에 보냄으로써 디스크를 액세스할 수 있다. 서버의 선택은 통계적으로 또는 동적으로 이루어질 수 있다.

고장이 발생한 경우에 처리되고 재 경로설정되는 디스크 액세스들은 화일 시스템 동작이 아닌 물리적 액세스 명령들임을 이해해야 한다. 다시 말하면 본 발명의 시스템에서는 가상 공유 디스크를 구비한 각 노드가 디스크 디바이스 드라이버에 명령들(물리적 디스크의 특정 위치로의 판독 및 기록과 같은)을 발송함에 있어 , 마치 상기 물리적 디스크가 디스크 어댑터를 통해 상기 노드에 직접 연결되어 있는 것처럼 명령을 발송한다. 이 명령들은 가상 공유 디스크 소프트웨어로부터 디스크의 1차 테일(포트)에 직접 연결된 노드상의 디스크 드라이버 소프트웨어로 전달되고, 이 디스크 드라이버 소프트웨어는 연결된 포트를 통해서 디스크 제어기로 명령을 발송한다.

본 발명이 바람직한 실시예를 통하여 설명되기는 하였으나, 본 기술분야의 당업자라면 여러가지 변형 및 개선을 행할 수 있다. 그러므로 전술한 바람직한 실시예는 예시로서 제공된 것이지 본 발명에 대한 제한이 아니다. 본 발명의 범위는 첨부된 특허청구범위에 의해 정해진다.

(57) 청구의 범위

청구항 1

적어도 세 개의 상호연결된 노드들 - 상기 노드들중의 일부는 서버 노드이고, 각 노드는 메모리와 멀티 포트 디스크를 포함하고, 상기 멀티 포트 디스크는 1차 서버 노드에 물리적으로 연결된 적어도 하나의 1차 테일과 2차 서버 노드에 물리적으로 연결된 2차 테일을 가진다 - 을 포함하는 클러스터화된 계산기 시스템의 디스크 액세스 경로 고장으로부터 회복하는 방법에 있어서, ① 상기 클러스터화된 계산기 시스템의 디스크 액세스 경로의 고장을 검출하는 단계 - 고장난 액세스 경로는 상기 1차 테일과 관련된다 - 와, ② 상기 고장을 검출하면, 조정자 노드는 상기 디스크로의 액세스를 가지는 상기 시스템의 모든 노드들로 제1메시지를 방송하는 단계와, ③ 각 노드가 상기 디스크로의 액세스를 중단시키는 상기 제1메시지를 수신하고, 상기 디스크로의 액세스 중단을 상기 조정자 노드에 애크놀리지하는 단계와, ④ 상기 디스크로의 액세스 중단을 애크놀리지하는 상기 단계에 응답해서, 상기 조정자 노드가 상기 디스크로의 액세스를 재개시키라는 제2메시지를 상기 노드들로 방송하는 단계와, ⑤ 각 노드는 상기 제2메시지를 수신하고, 상기 2차 테일에 의해 상기 디스크로의 액세스를 재개하는 단계를 포함하는, 디스크 액세스 경로 고장 회복 방법.

청구항 2

제1항에 있어서, 모든 서버 노드들은 디스크 어댑터를 포함하고, 상기 디스크 액세스 경로는 상기 디스크 어댑터와 상기 서버 노드들을 포함하고, 고장을 검출하는 상기 단계는 상기 노드들중의 적어도 하나 또는 상기 디스크 어댑터들의 상기 고장을 검출하는 단계를 포함하는, 디스크 액세스 경로 고장 회복 방법.

청구항 3

제1항에 있어서, 상기 조정자 노드가 고장난 경우에는, 회복의 조정을 수행하기 위해서 백업 조정자 노드를 지정하는 단계를 포함하는, 디스크 액세스 경로 고장 회복 방법.

청구항 4

클러스터화된 다중 처리 시스템에 있어서, ① 적어도 세 개의 상호연결된 노드들 - 상기 노드들중의 일부는 서버 노드들이고, 각 노드는 메모리를 포함한다 - 과, ② 1차 서버 노드에 물리적으로 연결된 적어도 하나의 1차 테일과 2차 서버 노드에 물리적으로 연결된 2차 테일을 가지는 멀티 포트 디스크와, ③ 개시 노드로부터의 디스크 액세스 요구를 상기 개시 노드, 상기 서버 노드들과 상기 디스크 사이에 정의된 1차 디스크 액세스 경로와 2차 디스크 액세스 경로중의 적어도 하나를 따라서 상기 디스크에 물리적으로 연결된 서버 노드로 전달하도록 상기 노드들에 결합된 디스크 액세스 요구 메커니즘과, ④ 상기 1차 디스크 액세스 경로와 상기 2차 디스크 액세스 경로중의 하나에서의 고장을 검출하도록 상기 노드들에 결합된 고장 검출 메커니즘과, ⑤ 고장이 검출될 때, 상기 디스크로의 비고장 디스크 액세스 경로를 따라서 계속적인 디스크 액세스 요구들을 재방향설정하도록 상기 각 노드들의 메모리에 저장되고 상기 고장 검출 메커니즘에 결합된 대리 논리(proxy logic)를 포함하고, 상기 대리 논리는, ⑥ 고장난 디스크 액세스 경로로의 액세스를 중단시키는 중단 메시지를 참가 노드들에 방송하고, 모든 참가 노드들로부터의 애크놀리지 메시지를 기다리도록 적응된 조정자 노드를 포함하고, ⑦ 각 참가 노드는 상기 중단 메시지를 수신하여, 상기 고장난 디스크 액세스 경로로의 액세스를 중단하고, 상기 고장난 디스크 액세스 경로로의 액세스의 중단을 확인시키는 상기 애크놀리지 메시지를 상기 조정자 노드로 보내고, 상기 조정자 노드로부터의 재개 메시지를 기다리도록 적응되고, ⑧ 상기 조정자 노드는 상기 모든 참가 노드들로부터 상기 애크놀리지 메시지를 수신하면 상기 재개 메시지를 보내도록 더 적응되고, ⑨ 상기 각 참가 노드는 상기 재개 메시지를 수신하면 상기 비고장 디스크 액세스 경로를 따라서 상기 계속적인 디스크 액세스 요구들을 재 방향설정하도록 더 적응되는, 2단계 수행 프로토콜을 포함하는, 클러스터화된 다중 처리 시스템.

청구항 5

제4항에 있어서, 상기 디스크로 들어오는 액세스 요구들을 저장하기 위한 상기 각 노드들의 큐와, 상기 큐에 있는 상기 요구들을 상기 비고장 디스크 액세스 경로를 통해 상기 디스크로 재 경로설정하는 수단을 포함하는, 클러스터화된 다중 처리 시스템.

청구항 6

제4항에 있어서, 모든 서버 노드들은 상기 디스크에 연결된 디스크 어댑터를 포함하고, 상기 고장 검출

메커니즘은 상기 서버 노드들과 상기 디스크 어댑터에서의 고장을 검출하는 수단을 포함하는. 클러스터화된 다중 처리 시스템.

청구항 7

제1항에 있어서, ① 상기 제1메시지에 응답하여, 상기 고장난 액세스 경로를 따라 상기 디스크로 전송되었던 펜딩된 요구들을 저장하고, 상기 디스크로의 액세스가 중단된 동안 도착한 요구들을 저장하는 단계와, ② 각 노드에서 상기 디스크로의 액세스를 재개하는 상기 단계에 응답하여, 상기 모든 요구들을 상기 2차 서버로 재발송하는 단계를 포함하는, 디스크 액세스 경로 고장 회복 방법.

청구항 8

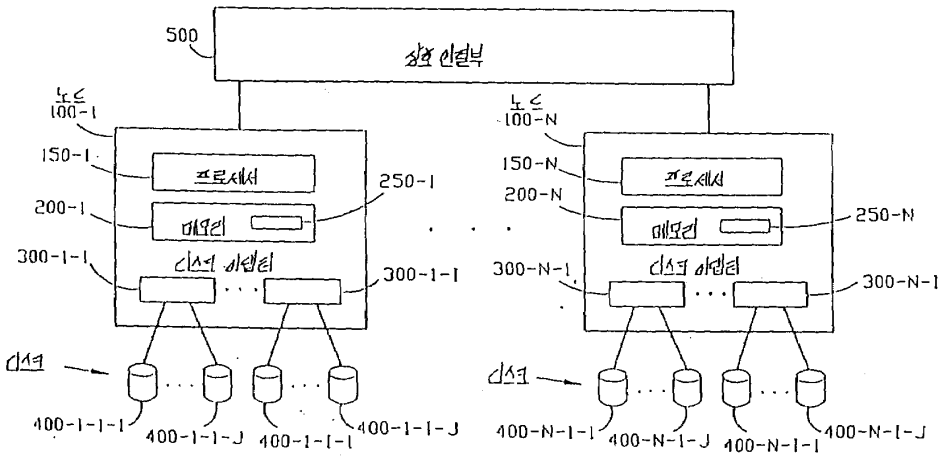
제7항에 있어서, 상기 모든 요구들은 큐에 저장되고, 재발송하는 상기 단계는 상기 2차 서버 노드에 의해 상기 큐에 저장된 상기 요구들을 상기 2차 서버 노드에 의한 디스크로 재 경로설정하는 단계를 더 포함하는, 디스크 액세스 경로 고장 회복 방법.

청구항 9

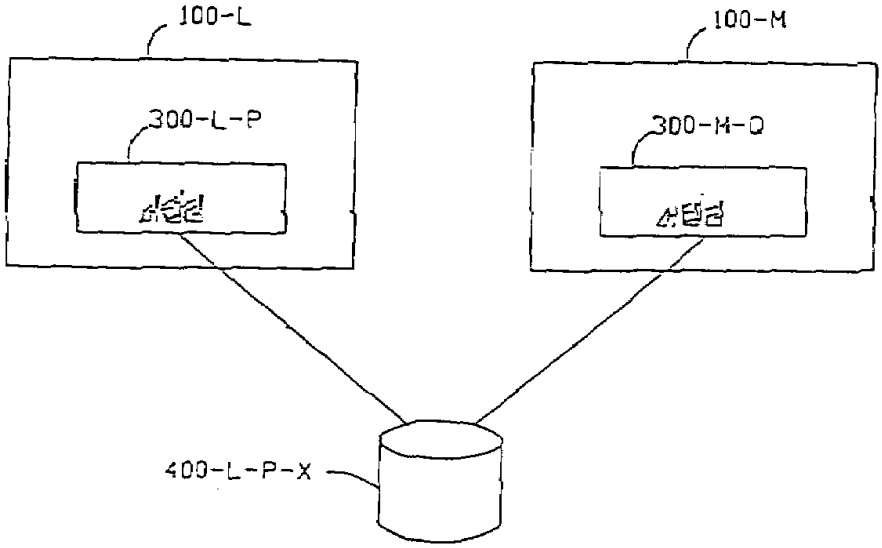
클러스터화된 다중 처리 시스템에 있어서, ① 적어도 세 개의 상호연결된 노드들 - 상기 노드들중의 일부는 서버 노드들이고, 각 노드는 메모리를 포함한다 - 과, ② 1차 서버 노드에 물리적으로 연결된 적어도 하나의 1차 테일과 2차 서버 노드에 물리적으로 연결된 2차 테일을 가지는 멀티 포트 디스크와, ③ 개시 노드로부터의 디스크 액세스 요구를 상기 개시 노드, 상기 서버 노드들과 상기 디스크 사이에 정의된 1차 디스크 액세스 경로와 2차 디스크 액세스 경로중의 적어도 하나를 따라서 상기 디스크에 물리적으로 연결된 서버 노드로 전달하도록 상기 노드들에 결합된 디스크 액세스 요구 메커니즘과, ④ 상기 1차 디스크 액세스 경로와 상기 2차 디스크 액세스 경로중의 하나에서의 고장을 검출하도록 상기 노드들에 결합된 고장 검출 메커니즘과, ⑤ 고장이 검출될 때, 상기 디스크로의 비고장 디스크 액세스 경로를 따라서 2단계 수행 프로토콜을 통해 계속적인 디스크 액세스 요구들을 재방향설정하도록 상기 고장 검출 메커니즘에 결합된 대리 논리 수단를 포함하는, 클러스터화된 다중 처리 시스템.

도면

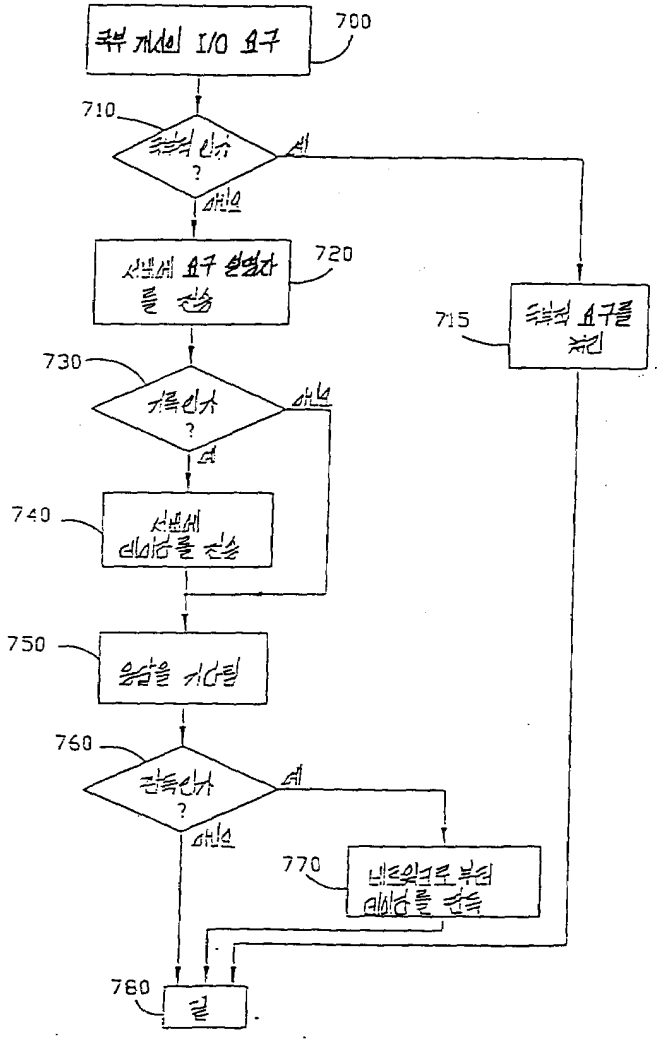
도면1



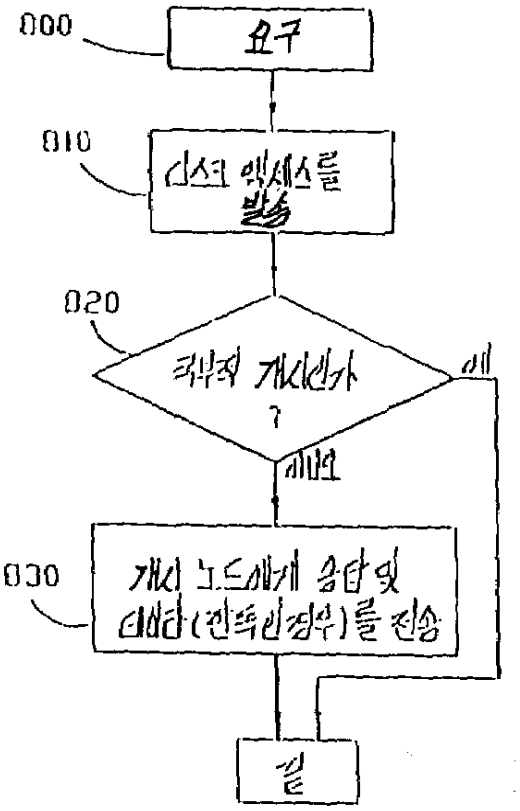
도면2


100-L
100-M
300-L-P
300-M-Q
400-L-P-X

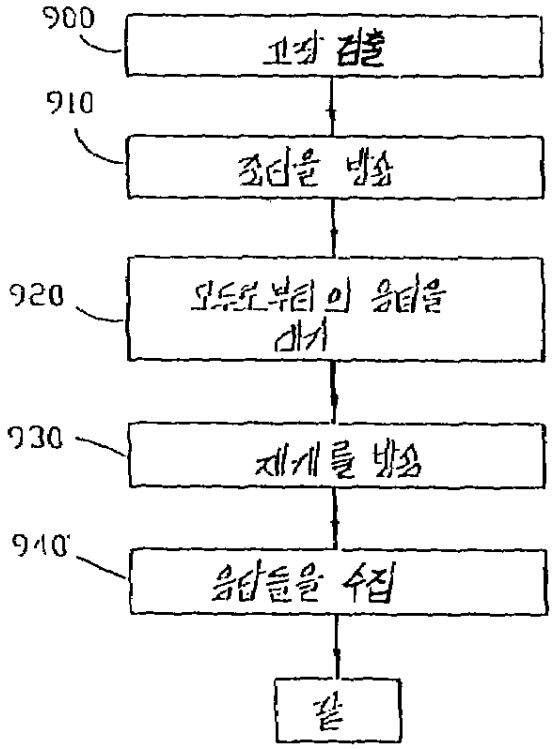
도면3


700
710
720
715
730
740
750
760
770
780

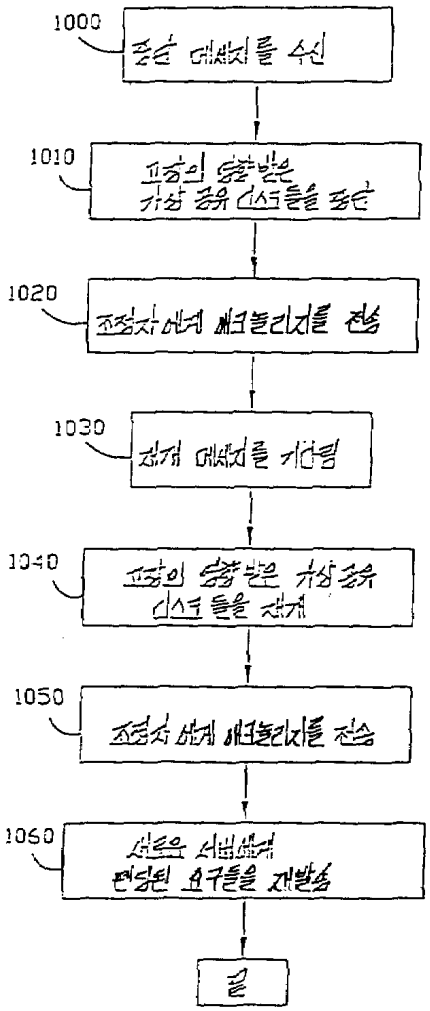
도면4



도면5



도면6



도면7