클러스터 DBMS를 위한 고가용성 클러스터 관리기의 설계 및 구현

김영창[0] 장재우
전 북 대 학 교 컴 퓨 터 공 학 과
{yckim, jwchang}@dblab.chonbuk.ac.kr

김홍연 김 준
한국전자통신연구원 한국전자통신연구원
{kimhy, jkim}@etri.re.kr

Design and Implementation of Cluster Management Tool with High-Availability for Cluster DBMS

Young-Chang Kim[0] Jae-Woo Chang
Dept. of Computer Engineering,
Chonbuk National University

Hong-Yun Kim June Kim
Electronics and Telecommunications
Research Institute

요 약

최근 인터넷 서버의 역할을 위해 24 시간 무정지 서비스가 요구되면서 여러 개의 단일 서버를 고속의 네트웍으로 연결한 클러스터 DMBS 에 관한 연구가 국내외적으로 활발히 진행중이다. 그러나, 이러한 클러스터 DBMS 를 효율적으로 관리할 수 있는 관리 도구에 대한 연구는 미흡한 실정이다. 본 논문에서는 각 서버의 상태를 모니터링한 정보를 바탕으로 서버의 오류를 감지하고 복구하므로써, 전체 클러스터 DBMS 가 정상적인 서비스를 할 수 있도록 지원할 수 있는 고가용성 클러스터 관리기를 설계하고 구현한다.

1. 서론

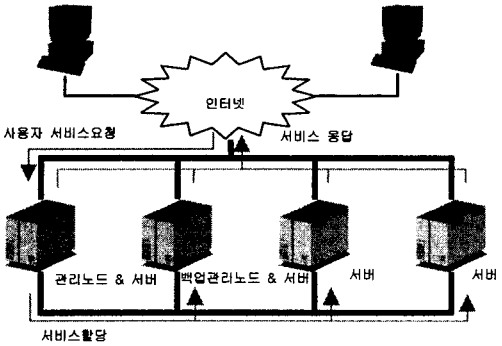
최근 인터넷 환경에서 급속히 증대되는 24 시간 무정지 서비스 요구를 효과적으로 처리하기 위하여, 저비용으로 시스템 성능 및 시스템 확장을 용이하게 하는 클러스터 컴퓨팅 시스템이 필요하게 되었다[1,2]. 이러한 인터넷 환경에서는 기존의 단일 대용량 데이터베이스 서버들의 한계 때문에 고성능과 고가용성의 서비스를 제공하는 것이 이미 한계에 도달하였다. 이러한 한계를 극복하고 더욱 강력한 컴퓨팅 파워와 시스템의 안정적 서비스를 제공하기 위한 클러스터 DBMS 에 대한 연구 및 개발이 활발히 이루어지고 있다. 이러한 클러스터 DBMS 를 효율적으로 관리하기 위해서는 현재 운영중인 컴퓨터가 고장 등의 이유로 운용할 수 없을 때에도 전체 클러스터 DBMS 가 정상적인 동작을 할 수 있도록 지원하는 클러스터 관리기가 필요하다. [3,4]

이를 위해 본 논문에서는 클러스터 DBMS 를 위한 고가용성 클러스터 관리기를 설계하고 구현한다. 이는 리눅스 가상 서버를 통해, 사용자의 서비스 요구를 적절한 서버에 전달하여 처리하고, 각 노드의 시스템 환경, 네트웍 및 데이터베이스를 모니터링하여 고장을 감지하여, 서버 노드의 고장이 발생했을 때 고장이 전체 클러스터 DBMS 의 동작에 영향을 미치지 않도록 복구절차를 수행한다.

본 논문의 구성은 다음과 같다. 제 2 장에서는 본 논문에서 설계하는 클러스터 시스템의 전반적인 구조와 고가용성 클러스터 관리기의 설계에 대해 기술하고, 제 3 장에서는 설계된 고가용성 클러스터 관리기의 구현 및 테스팅에 대해 기술한다. 마지막으로 제 4 장에서는 결론 및 향후연구를 제시한다.

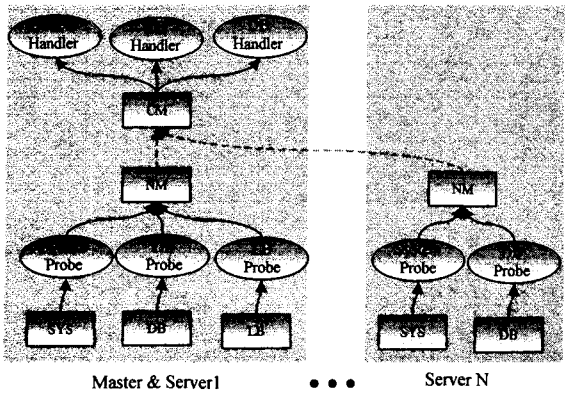
2. 고가용성 클러스터 관리기의 설계

클러스터 DBMS 는 여러 개의 서버노드로 구성되어 있으며, 하나의 노드는 서버역할과 관리노드의 역할을 동시에 수행한다. 사용자의 서비스 요청은 관리노드를 통해 리눅스 가상 서버의 스케줄링 알고리즘에 따라 실제 서버 노드에 전달된다 본 논문에서 설계한 클러스터 DBMS 의 시스템 구성은 [그림 1]과 같다.



[그림 1] 클러스터 DBMS 의 시스템 구성

클러스터 관리기는 CM(Cluster Manager), NM(Node Manager), Probe, Handler 의 4 개 모듈로 나뉘어진다. Probe 는 각 노드에서 시스템 자원을 모니터링하는 데몬(Daemon)과 데이터베이스를 모니터링하여 이벤트를 발생시키며, 발생된 이벤트는 CM 과 통신을 담당하는 NM 을 통해 CM 에 전달된다. CM 은 Probe 로부터 NM 을 통해 전달받은 각 노드의 모니터링 정보와 이벤트를 바탕으로 전체 클러스터 시스템을 통합관리하고, 각 노드의 상태 테이블을 유지한다. 또한, CM 은 이벤트를 해석하여 적절한 서비스 Handler 에게 전달하고, Handler 는 이벤트에 따라 정해진 절차를 수행한다. [그림 2]는 클러스터 관리기를 구성하는 모듈의 전체적인 구조를 나타낸다.



[그림 2] 클러스터 관리기의 모듈 구조

서버노드중 하나의 노드는 데이터 베이스 서버역할과 관리노드의 역할을 동시에 수행하며, 일반 서버노드중의 하나가 백업 노드 역할을 수행하며 관리노드를 모니터링한다. 관리노드의 고장 발생시 백업노드는 CM 과 각 Handler 를 기동시켜 관리노드로의 역할을 수행하고, 다른 정상상태 노드중의 하나가 백업노드의 역할을 수행하여, 전체 클러스터 DBMS 가 24 시간 무정지 서비스를 수행할 수 있도록 한다.

아울러 클러스터 관리기는 클러스터 시스템을 구성하는 노드들간의 통신을 수행하는 클러스터 네트웍과, 사용자의 서비스 요청과 처리된 결과를 사용자에게 전달하는 서비스 네트웍을 사용한다. 클러스터 네트웍의 고장 발생시 해당 노드는 더 이상 클러스터 관리기로부터 사용자의 서비스 요청을 전달 받을 수 없는 노드 격리 상태가 된다. 아울러 서비스 네트웍의 고장 발생시 해당 노드는 관리노드로부터 전달받은 사용자 서비스 요청에 대한 결과를 사용자에게 전달 할 수 없는 상태가 된다. 마지막으로 두 네트웍에서 모두 고장이 발생 했을 때 노드는 완전고장 상태가 된다. [표 1]은 네트웍의 상태에 따른 고장의 분류를 나타낸다.

네트웍종류 / 노드상태	클러스터 네트웍	서비스 네트웍
정상	0	0
서비스네트웍단절	0	×
노드격리	×	0
노드완전고장	×	×

[표 1] 네트웍의 상태에 따른 고장의 분류

클러스터 관리기는 각 노드의 시스템, 데이터베이스, 리눅스 가상서버를 모니터링하는 Probe 가 발생한 이벤트 정보를 바탕으로 전체 클러스터 시스템을 관리한다. 각 Probe 들은 해당 서비스가 기동되었는지, 종료되었는지, 정상적으로 서비스를 제공하는지, 오류가 발생했는지에 따른 이벤트를 발생시킨다. NM 은 Probe 가 발생시킨 이벤트 정보를 CM 에 전달하고 응답을 기다린다. CM 으로부터의 응답을 정해진 시간동안에 받지 못하면, CM 에 오류가 발생한 것으로 간주하고, 유지하고 있는 백업관리노드의 IP(Internet Protocol)로 백업 CM 과의 연결을 한다. CM 은 NM 을 통해 Probe 가 발생시킨 이벤트 정보를 바탕으로 각 노드의 서비스 프로세스, Probe, NM, 그리고 노드의 오류여부에 관한 테이블을 유지 관리하며, 또한 클러스터 네트웍과 서비스 네트웍을 통해 각 노드로 ping 테스트를 통해 [표 1]의 네트웍 고장에 따른 이벤트를 발생시킨다. 시스템 모니터링 데몬과 데이터베이스, 리눅스 가상 서버에 관한 이벤트는 해당 Handler 에게 전달을 하고, 서비스 상태 테이블의 정보를 갱신하여 이를 백업관리노드의 백업 CM 에 전달한다. 이때 백업 CM 으로부터 응답이 없으면 백업 CM 에 오류가 발생한 것으로 간주하고, 서비스 가능한 다른 서버노드 중의 하나에 백업 CM 을 기동시킨다. NM 의 오류가 발생했을 때는 해당 노드가 서비스 불능임을 서비스 상태 테이블에 등록하고, NM 을 재기동 시킨다. 시스템 Handler 는 시스템 모니터링 데몬의 오류발생시 해당 데몬을 재기동 시킨다. 데이터베이스 Handler 는 데이터베이스의 오류, 클러스터 네트웍 및 서비스 네트웍의 오류발생시 데이터베이스의 오류복구 절차를 수행한다. 리눅스 가상 서버 Handler 는 시스템, 데이터베이스, 각 네트웍의 오류발생시 해당 노드를 부하분산 대상에서 제거하여 더 이상의 오류가 전체 클러스터 시스템에 영향을 미치지 않도록 한다. [표 2]는 각 모듈이 발생시키는 이벤트의 종류와 그에 따른 수행 절차를 나타낸다.

모듈	상태	이벤트	수행절차
시스템 Probe	기동	SYS_START	서비스상태테이블에등록
	정상	SYS_LOAD	모니터링된 정보 저장
	오류	SYS_ERROR	모니터링데몬재기동
	종료	SYS_STOP	서비스상태테이블에등록
DB Probe	기동	DB_START	서비스상태테이블에등록
	정상	DB_ALIVE	Time Stamp
	종료	DB_STOP	서비스상태테이블에등록
	오류	DB_ERROR	DB 복구절차수행 부하분산대상에서 제거
LB Probe	기동	LB_START	서비스상태테이블에등록
	정상	LB_ALIVE	Time Stamp
	종료	LB_STOP	서비스상태테이블에등록
	오류	LB_ERROR	LVS 재기동
NM	기동	NM_START	서비스상태테이블에등록
	정상	NM_ALIVE	Time Stamp
	종료	NM_STOP	서비스상태테이블에등록
	오류	NM_ERROR_CM	백업노드로 연결 재설정
CM	오류	CM_ERROR_SYS	시스템 Probe 재기동
	오류	CM_ERROR_DB	DB Probe 재기동
	오류	CM_ERROR_LB	LB Probe 재기동
	오류	CM_ERROR_NM	NM 재기동
	오류	CM_ERROR_SVC	DB 복구절차수행 부하분산대상에서 제거
	오류	CM_ERROR_CLS	DB 복구절차수행 부하분산대상에서 제거
	오류	CM_ERROR_NODE	DB 복구절차수행 부하분산대상에서 제거

[표 2] 이벤트의 종류 및 그에 따른 수행절차

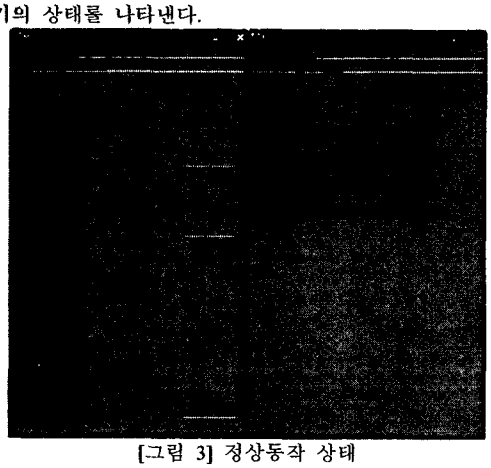
3. 클러스터 관리기의 구현 및 테스팅

본 절에서는 고가용성 클러스터 관리기를 구현하여 테스트를 수행한다. 구현된 시스템 환경은 [표 3]과 같고 동일한 사양의 4 대의 서버에서 테스트를 수행하였다. 데이터베이스는 바다 IV 의 Midas 를 사용하였고 리눅스 가상서버는 버전 0.81 버전을 사용하였다.

OS	RedHat Linux (커널 2.4.5)
Compilier	gcc 2.96-76
Database	바다 IV
LVS	version 0.81

[표 3] 구현된 시스템 환경

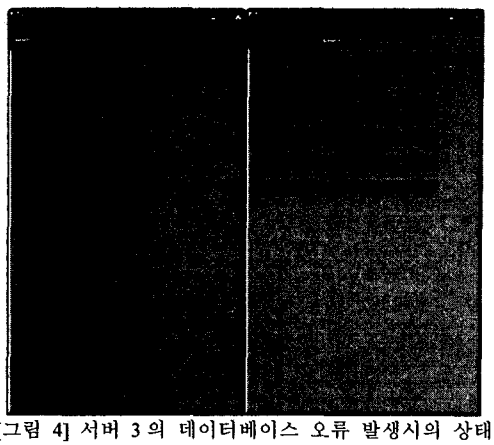
4 대의 서버중 첫번째 서버는 관리노드의 역할을 동시에 수행하며 두번째 서버가 백업관리노드의 역할을 수행한다. 정상 동작 상태일때는 사용자의 서비스 요청이 관리노드를 통해 리눅스 가상서버의 Round-Robin 알고리즘에 의해 각 4 대의 서버로 전달된다. [그림 3]은 정상 동작 상태일 때의 클러스터 관리기의 상태를 나타낸다.



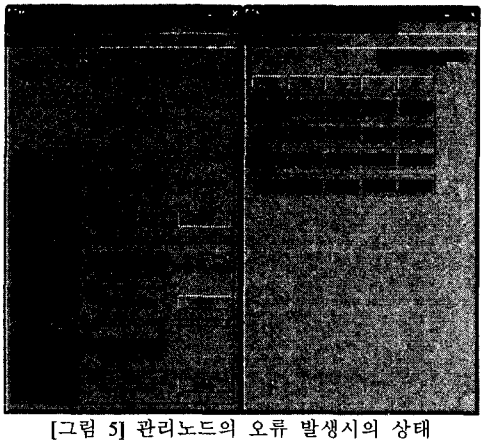
[그림 3] 정상동작 상태

첫째, 한 노드의 데이터베이스에 오류가 발생할 때를 테스팅 한다. 데이터베이스의 오류가 발생하면 DB Probe 가 이를 감지하여 NM 을 통해 CM 에 전달하고 CM 은 이를 서비스 상태 테이블에 등록한 후 DB Handler 에 전달한다. DB Handler 는 해당 노드가 더 이상 서비스를 할 수 없기 때문에 부하분산 대상에서 제거해 사용자의 서비스 요청이 잘못된 노드에 할당되는 것을 방지한다. [그림 4]은 서버 3 의 데이터베이스의 오류발생시의 클러스터 관리기의 상태를 나타낸다. 그림에 나타난바와 같이 서버 3 의 데이터베이스가 오류상태로 표시됨을 알수 있다.

둘째, 관리노드에서 오류가 발생할 때를 테스팅한다. 관리노드는 다른 하나의 서버가 동시에 그 역할을 수행하게 되는데 관리노드에 오류가 발생하면 이를 모니터링하고 있는 백업관리노드가 관리노드의 역할을 수행하게 되고, 서비스 가능 노드중의 하나의 노드가 백업관리노드로 동작하게 된다. [그림 5]는 관리노드의 오류 발생시의 상태를 나타낸다. 그림에 나타난 바와 같이 서버 1 이 관리노드의 역할을 서버역할과 함께 동시에 수행하다가 오류가 발생하면, 백업역할을 수행하던 서버 2 가 관리노드의 역할을 수행하게 된다. 또한, 서버 1 에서 실행되던 리눅스 가상서버가 관리노드로 바뀐 서버 2 에서 정상적으로 동작함을 알 수 있다.



[그림 4] 서버 3 의 데이터베이스 오류 발생시의 상태



[그림 5] 관리노드의 오류 발생시의 상태

4. 결론

본 논문에서는 클러스터 DBMS 의 효율적인 관리를 위한 고가용성 클러스터 관리기를 설계 및 구현하였다. 구현된 클러스터 관리기는 24 시간 무정지 서비스를 제공할 수 있도록 시스템, 데이터베이스를 모니터링하여 오류발생시 복구절차를 수행하며, 전체 클러스터 DBMS 가 오류에 상관없이 계속적인 서비스를 수행할 수 있도록 클러스터 시스템을 관리한다.

향후 연구로는 각 노드에서 모니터링한 시스템의 부하를 바탕으로 사용자의 요구를 적절히 분산시킬 수 있는 효과적인 부하분산알고리즘을 개발하는 것이다.

5. 참고 문헌

[1] 김진미, 온기원, 김학영, 지동해 , " 클러스터링 컴퓨팅 기술" , 1999

[2] Rajkumar Buyya , High Performance Cluster Computing Vol 1,2

[3] "Oracle 8i Administrator's Reference Release3(8.1.7) for Linux Intel", chapter 7 Oracle Cluster Management Software

[4] 최재영, 황석찬, " 클러스터를 위한 소프트웨어 도구" , 정보과학회지, 제 18 권 3 호, pp40~47.