분산컴퓨팅 환경에서의 고가용성 클러스터링 프레임워크 기본설계 연구

김점구* · 노시춘**

요 약

클러스터링은 상호 의존적 구성에 필요한 구조적인 기술이다. 클러스터링은 가변적 업무부하를 처리하거나, 서비스 연속성을 저해하는 고장 발생 시 운영이 계속되도록 여러대의 컴퓨터시스템 기능을 서로 연결하는 메커니즘이다. 고 가용성 클러스터링 기능은 가능한 오랜시간 서버 시스템이 작동하는데 중점을 둔다. 이 클러스터 는 멀티플 시스템 에서 실행되는 노드와 서비스를 중복하여 가지고 있어서 서로가 서로를 추적할 수 있다. Active-Standby 상태의 두 시스템이 있을 경우 활성 서버에 장애가 발생했을 때 모든 서비스가 대기 서버에서 구동돼 서비스가 이루어진다. 이 기능을 절체 또는 스위치오버(switchover)라 한다. 고가용성 클러스터링 기능은 가능한 오랜시간 서버 시스템이 작동하는데 중점을 둔다. 이 클러스터는 멀티플 시스템 에서 실행되는 노드와 서비스를 중복하여 가지고 있어서 서로가 서로를 추적할 수 있다. 한 노드가 장애 발생 시 둘째 노드가 몇초 이내에 고장 난 노드 임무를 수행한다. 고가용성 클러스터링 구조는 효율성 여부가 측정되어야 한다. 시스템 성능은 인프라시스템의 performance, latency, 응답시간(responseTime), CPU 부하율(CPU utilization), CPU상의 시스템 프로세스(system process)수로 대표된다.

A Study of Basic Design Method for High Availability Clustering Framework under Distributed Computing Environment

Jeom goo Kim* · SiChoon Noh**

ABSTRACT

Clustering is required to configure clustering interdependent structural technology. Clustering handles variable workloads or impede continuity of service to continue operating in the event of a failure. Long as high-availability clustering feature focuses on server operating systems.Active-standby state of two systems when the active server fails, all services are running on the standby server, it takes the service. This function switching or switchover is called failover. Long as high-availability clustering feature focuses on server operating systems. The cluster node that is running on multiple systems and services have to duplicate each other so you can keep track of. In the event of a node failure within a few seconds the second node, the node shall perform the duties broken. Structure for high-availability clustering efficiency should be measured. System performance of infrastructure systems performance, latency, response time, CPU load factor(CPU utilization), CPU processes on the system (system process) channels are represented.

Keywords:Infrastructure System, High Availability, Clustering, Load Balacing(HALB), Mechanism

접수일(2013년 6월 1일), 수정일(1차: 2013년 6월 11일), 게재확정일(2013년 6월 12일)

* 남서울대학교 컴퓨터학과

** 남서울대학교 컴퓨터학과(교신저자)

1. 서 론

클러스터링은 가변적 업무부하를 처리하거나 서비스 연속성을 저해하는 고장 발생 시 운영이 계속되도록 여러대의 컴퓨터시스템 기능을 서로 연결하는 메커니즘이다. 두대 이상 컴퓨터를 마치 하나의 컴퓨터처럼 동작하도록 연결하여 병렬 처리나 부하 배분 및 고장 대비 등 목적에 사용 할 수 있다. 클러스터링은 상호 의존적 구성에 필요한 구조적인 기술로서 active-standby 상태의 두 시스템이 있을 경우 활성 서버에 장애 발생 시 모든 서비스가 대기 서버에서 구동돼 서비스가 이루어진다. 이 기능을 스위치오버(switchover)라 한다. 고 가용성 클러스터링 기능은 가능한 오랜시간 서버 시스템 작동에 중점을 두며 멀티 시스템에서 실행되는 노드와 서비스를 중복하여 가지고 있어 서로 상대를 추적할 수 있다. 한 노드 장애 발생 시 둘째 노드가 몇 초 이내에 고장 노드 임무를 대행한다. 본 연구는 기존 분산처리 과정에서 노출되는 문제점 진단을 토대로 클러스터링 자원을 효과적으로 활용하여 다중화 부하분산 시스템을 구축하고 기존 시스템에서 사용되는 정적 부하분산 대신 동적 부하 균등화 기법을 사용하는 인프라스트럭쳐의 고가용성 클러스터링 프레임워크 설계방안을 제시한다. 기술 순서는 서론, 전통 정보시스템처리구조, 분산처리 과정의 문제점, 클러스터링 메커니즘 설계, 결론의 순서이다.

2. 전통 정보시스템 분산처리 구조

구조(architecture)란 원래 건축 분야에서 사용되던 용어로서 건물의 기본 구조나 골격, 틀을 의미한다. 컴퓨터나 통신 분야에서도 컴퓨터 구조나 네트워크 구조를 각각 컴퓨터와 네트워크의 기본 구조 혹은 구성을 나타낸다. 컴퓨터 시스템의 기본 구조는 물리적 구조와 논리적 구조로 분류된다. 컴퓨터시스템의 물리적 구조는 하드 웨어로서 ①프로세서(대형 컴퓨터, 워크 스테이션, PC, 수퍼 컴퓨터 등) ② 메모리(하드디스크, 자기 테이프, RAM, ROM 등) ③ 네트워크(LAN, WAN, 게이트웨이 등)이며 논리적 구조는 물리적 구조를 토대로 소프트웨어에 의해 실현되는 분산 처리 시스템의 구조로서 OS, DBMS, 통신 소프트웨어 등 제어 프로그램, 응용프로그램, 데이터베이스, 지식베이스이다. 시스템 운영 중 서비스의 연속성을 저해하는 요인은 크게 두 가지로 나누는데 계획된 다운타임과 계획되지 않은 다운 타임이다. 계획된 다운타임에는 시스템 구성, 용량 변경 , 신규시스템 도입, 데이터 백업, 소프트웨어 추가, 애플리케이션 마이그레이션이 있다. 계획되지 않은 다운타임에는 정전 및 UPS 장애, 하드웨어 장애, 소프트웨어 장애, 네트워크 장애, 내부 외부의 위해행 위에 의한 장애발생이 있다[1][2].

3. 분산컴퓨팅 처리과정의 문제점

3.1 계획되지 않은 다운타임 발생 문제

정보시스템은 하나의 부하분산 서버가 모든 기능을 처리해야 하므로 이에 따른 통신트래픽 볼륨 증가와 서버 내부 과부하에 의해 접속이 불안정 해질 수 있다. 네트워크 트래픽 증가와 서버의 과부하로 접속이 불안정해지거나 지연시간을 초래하게 되면 최악의 경우 서버 및 네트워크 장애로 인해 서비스 중지가 나타나게 된다. 이는 계획되지 않은 다운타임이며 이를 해결하기 위해 우선적으로 자원을 확충시키고 있으나 자원 확충은 고비용이 소요되며 이에 대한 합리적 가이드라인이 부재하다[3].

3.2 분산형 데이터베이스 품질확보 문제

분산처리 시스템에서는 하나의 일을 분할하여 각각에 대해 복수 프로세서에서 동시에 처리하며 이들 복수개의 프로세서에서는 데이터베이스 등 공유 데이터를 액세스하면서 처리한다. 대표적 분산형 데이터베이스는 시스템 전체의 제어 방법, 데이터 갱신에 따른 동기 제어, 교착 상태 검출과 회피 등의 많은 기술적인 과제를 가지고 있다. 데이터의 갱신을 동반하는 경우 문제가 발생하는데 공통의 데이터베이스 액세스를 처리하는 경우 처리순서를 원활하게 제어하지 않으면 자료 정확성에서 오류가 발생한다.

3.3 다양한 하드웨어, 소프트웨어 연동

분산처리 시스템에서는 해결 할 과제가 많으며 광역 네트워크를 기본으로 한 분산처리 시스템에 있어서 하드웨어는 슈퍼컴, 메인 프레임, 미니 컴, PC, 내장 초소형 컴퓨터, 스마트폰 등 다양하며 소프트웨어는 서로 다른 운영체제, 프로그래밍 언어, 상이한 하드웨어를 통합 동작 하는 것이 쉽지 않다. 소프트웨어 통합과 함께 동작은 더 어렵다. 다른 기계에서 작동하는 소프트웨어는 같은 언어가 아니며 수행환경이 다르다. 상이한 환경의 모든 소프트웨어 간 연동은 예측되지 않은 트러블을 야기시킬 수 있다[4][5].

3.4 클러스터링 방식의 문제점

분산처리 시스템에서는 통신 처리나 정보처리의 기구(mechanism)가 복잡하다. 컴퓨터시스템의 가용성 보증을 위해 가장 보편적인 클러스터 형태로써 한 대의 로드밸런스 서버가 다수 리얼서버로 요청을 분산한다. 일반적인 기존의 단일 부하 분산 서버 클러스터링 방식은 하나의 부하 분산기를 두고 모든 트래픽을 조절하는 기능을 수행한다. 하나의 분산 서버로 전체 서버 관리 시 많은 문제점이 발생된다.

4. 고가용성 클러스터(HAC, High-Availability Clusters) 클러스터링 메커니즘 설계

4.1 고가용성 클러스터 요건

시스템은 어떤 클러스터가 제공하는 서비스의 가용성을 기본적으로 개선하려는 목적으로 구현되고 시스템이 실패할 경우 서비스를 제공하기 위해 중복 노드를 운영한다. 복수의 중복노드로 구성된 시스템이 동일한 기능을 수행하도록 동일한 하드웨어와 소프트웨어 채널을 구성한다. 두 대의 시스템 구간을 고속의 채널로 연결하여 서로 상대방의 현재 상태를 상시적으로 크로스체크 한다. 두대 서버는 전체 서비스를 분할, 각각 일부 서비스를 제공한다. 시스템 가동 과정에서 하드웨어 ,소프트웨어, 네트워크, 데이타베이스 장애 발생 시 상대방 서비스를 인수하여 가용성을 지속적으로 유지시킨다[3][6].

4.2 고가용성 클러스터 기능의 품질조건

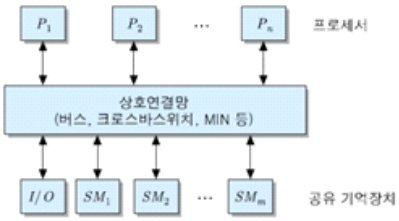
고가용성 클러스터 메커니즘에서의 품질만족도 측정은 단일 기능을 포함한 관련기능의 종합구조에 부합되는 다각적 측면의 기능을 설정해야 한다. 클러스터 단일기능 측정 만으로서는 복잡 다양한 고가용성 클러스터 시스템 성능관리에 한계가 있어 측정자체의 의미와 가치 또한 한계일 수밖에 없다. 고가용성 클러스터의 효율은 시스템 performance로서 최종적이고 종합적인 성능이 이루어진다. 통합 구조로 설계된 기능은 계량적으로 측정과 검증이 가능해야 한다. 계량화되지 못할 경우 파라메타별 품질 달성 여부 및 수준평가 어렵기 때문이다. 고가용성 클러스터를 기반으로한 정보시스템은 사용자의 중단없는 서비스 요구가 높아지므로 24시간×365일 항시 정상적 고품질 서비스 제공이 필요하다. 클러스터 구성원 간 hartbeat 통신으로 장애 발생시 3초안에 클러스터 재분배 및 자동분산을 실시한다. 고가용성 클러스터 품질조건은 다음 항목을 설정한다[6][7].

- MTBF(mean time between failure:평균고장간격) : 시스템 가용성은 시스템 고장이 발생할 때 까지의 평균시간
- MTTR(mean time to repair:평균복구간격) : 시스템에 고장이 발생하고 나서 복구가 이루어질 때 까지의 평균소요시간
- 가동률 : 양자의 합계에 의한 MTBF 비율과 MTTR을 가동율이다.

4.3 고가용성 인프라 구성

고가용성 클러스터링 서비스의 도입은 하드웨어 이중화로부터 시작되며 하드웨어, 소프트웨어가 클러스터링 기능을 구현하므로서 이루어진다. 고가용성이 관리하는 포인트는 크게 전원, CPU, 메모리, HDD 등 하드웨어, 네트워크 카드, LAN 케이블 등 네트워크, DB 및 각종 애플리케이션 등 프로세스이다. 고가용성 구축 시 먼저 하드웨어적으로 완벽한 이중화를 구성하고 고가용성 클러스터링 구성은 단순히 시스템 두 대를 이용한 구성이 아닌 다중노드 클러스터링 구조

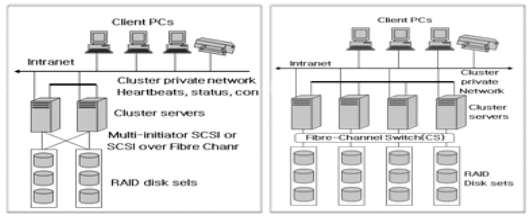
를 적용한다. 고가용성 클러스터링 시스템에서 다중처리 기능 설계 시 복수 프로세서들과 입출력 프로세서들을 메모리에 연결시키는 방법을 구현한다. 다중처리 시스템은 거의 비슷한 능력을 갖는 두개 이상 프로세서를 포함 모든 프로세서는 같은 메모리를 공동으로 사용하고 모든 프로세서는 입출력 채널과 제어장치 및 그 외의 장치들을 공동 사용한다. 균일 기억장치 액세스(uniform memory access: UMA)모델을 확보하여 모든 프로세서 들이 상호 연결망에 의해 접속된 주기억 장치를 공유한다. 프로세서들은 주 기억장치의 어느영역 이든 액세스할 수 있으며 소요시간을 동일하게 한다[2][3].



[그림1] 병렬처리 프로세서 구조

4.4 클러스터링 구조 설계

고가용성 클러스터링 시스템의 대표적 클러스터링 구조는 active-active active-passive, 4-node- active -active-active-active 이다. Active - passive 방식은 두 노드에서 클러스터링이 수행되며 한 시스템만 서비스를 제공하고 두번째 노드 시스템은 첫 번째 시스템 문제 발생 시 서비스를 제공한다. Active-active 방식은 두 시스템이 동시 서비스를 제공하며 양쪽서버에서 SQL 서버가 서비스 된다. 두 시스템은 독립적이며 데이터 공유는 불가하다. 클러스터링은 계속 상대 시스템 상태를 모니터링 하다가 장애 발생 시 자동 조치하게 된다. 트리거 이벤트에 의해 하나의 이벤트가 생성되면 그 이벤트에 의해 데이터 의 교환이 일어나게 된다. 두개 이상의 시스템이 네트워크를 통해 경로가 설정되어진다. 시스템 (A)에서 다른 시스템(B)으로 메시지의 형태로 데이터의 전송이 이루어진다. 메시지를 수신한 B 시스템은 수신여부 확인 메시지를 다시 시스템 A로 보내 하나의 사이클이 종료된다.



[그림2] 고가용성 클러스터링 구조

4.5 병렬처리 소프트웨어 설계

소프트웨어 구조측면의 고가용성은 맨 하단에 OS가 있고, 그 위에 애플리케이션이 있다. 고 가용성은 맨 상단에서 시스템과 애플리케이션을 모니터링 한다. 소프트웨어 구성요소로서 클러스터는 운영 체제, 클러스터 서비스, 클러스터 리소스, 클러스터 된 서비스나 응용 프로그램 자체 등 소프트웨어 구성요소로 구성한다. 클러스터에 참여하려면 클러스터는 쿼럼 장치에 액세스 할 수 있는 노드와 통신할 수 있어야 한다. 쿼럼 장치를 가지고 있는 노드가 실패할 경우 다른 노드에서 클러스터 구성 데이터를 사용할 수 있게 하려면 모든 노드가 연결되어 있는 실제 저장 장치에 쿼럼 리소스를 저장해야 한다. 데이터베이스 미러링은 데이터베이스의 가용성을 높여주는 주요 소프트웨어 솔루션이다. 고가용성 클러스터링 시스템은 하나의 운영체제에 의해 운영되며 이 운영체제는 프로세스와 각 작업의 상호작용을 여러 단계에서 도와준다. 프로그램, 데이터 집합, 데이터 단위들의 단계에서 상호작용 을 도운다. 요구되는 기능은 작업-단위 병렬성, 태스크-단위 병렬성, 스레드-단위 병렬성, 명령어-단위 병렬성이다.

- 작업-단위 병렬성(job-level parallelism) :
  독립적인 작업 프로그램(job program) 단위로 병렬처리[예] 성적관리 프로그램 // 실험 데이터 처리 프로그램
- 태스크-단위 병렬성(task-level parallelism) :
  하나의 큰 작업을 기능에 따라 분할한 작은 프로그램(태스크) 단위로 병렬처리
- 스레드-단위 병렬성(thread-level parallelism):
  동시에 처리될 수 있는 가장 작은 크기의 독립적인 단위 프로그램인 스레드 단위의 병렬처리
- 명령어-단위 병렬성(instruction-level parallelism):

데이터 의존성이 존재하지 않는 여러 개의
명령어들을 동시에 수행하는 병렬처리

4.6. 제어 알고리즘 설계

◉ 프로세스간 통신(Interprocessor communication)

고가용성 클러스터링 시스템에서 프로세서간 통신은 분할된 부분을 나누어 처리하는 프로세서 간의 데이터 교환을 위한 메카니즘이다. 프로세서 간 통신은 ①동기 ②전송기능 ③메시지 표현형식 3가지 속성으로 규정된다. 밀결합형 시스템에서의 프로세스간 통신은 공유 메모리 사용방식이며 소 결합형 시스템에서의 프로세스간 통신은 message passing 방식이다. 소결합형 시스템의 프로세스 간 통신에서는 논리적 통신 선로를 설정하여 서로 동기를 취하면서 데이터를 전송하는 경우이다. 동기 전송에서 긴 데이터를 보내는 경우에는 몇 개 블럭으로 분할하여 블럭마다 확인을 취하면서 차례로 송신한다. 만일 에러가 발생한 경우에도 미리 정해진 동기점을 기준으로 재전송 한다.

◉ 커미트먼트(Commitment) 제어

고가용성 클러스터링 시스템에서 프로세스나 자원이 공간적으로 분산되어 있는 환경에서는 하나의 트랜잭션)이 복수의 부분적 일(서브트랜잭션)로 분할되어, 각각을 개별적으로 다른 프로세서에서 병행 처리한다. 이들이 모두 정확하게 종료했을 때에 한하여 원래의 일이 정상 종료로 하는 방식이다. 서브트랜젝션 처리가 성공하지 못하면 원래 트랜젝션 처리를 실패한 것으로 간주한다. 이 처리를 원활히 진행하는 방법을 커미트먼트제어라 하며 서브 트랜잭션은 일의 단위를 나타내어 데이터베이스로의 조회나 갱신 작업(데이터 삽입 ,삭제,변경) 등 단위이다. Commitment 제어는 분산 데이터베이스에 있어 다수 복제(copy)를 동시에 갱신하는 경우나 분산 환경에서 장애가 발생한 때 회복 방법으로 사용된다.

◉ 하트비트(Heartbeat) 기능

고가용성 클러스터링 시스템에서 heartbeat는 상호의존적 고가용성 구성에 필요한 구조적 기술로서 diagnostics configuration,서비스를 보호 하고 복구할 수 있는 동작인 recovery actions, 다양한 시스템 구성요소의 상태를 감시하고 관리하는 상태관리가 필요하다. Active- standby 상태의 두 시스템이 있을 때 활성 서버에 장애가 발생했을때 모든 서비스가 대기 서버에서 구동돼 서비스가 이루어지며 이하 동작은 다음같은 프로시듀어를 거친다.

#1. Master-server 두 대가 동일하게 작동하므로 서버 중 한대가 fail 시 나머지 서버 가계속해서 임무를 수행한다. 각 서버에 동일한 데이터가 저장되므로 어플리케이션 접근에 의한 부하를 분산시킬 수 있다. Master-server 중 한대는 primary server로서 기능하다. fail 발생 시 standby server가 임무를 수행한다.

#2. 시스템에서는 master 서버와 standby 서버 간의 컨넥션 동기화가 가능하다. Master서버가 장애를 일으키면 클라이언트와 리얼서버 간의 세션유지를 위한 연결상태 정보를 standby 서버가 그대로 유지하면서 로드밸런스를 떠맡게 된다. 두 대의 리얼 서버가 failover 기능 로드 밸런스 서버의 역할을 겸한다.

#3. 리얼서버1/마스터 서버 장애시 리얼서버2가 로드밸런싱하며 리얼서버1은 클러스터 군에서 제외된다. 로드 밸런스 서버는 모니터 시스템을 이용하여 장애를 감지하고 SMS를 통해 관리자에게 보고한다. 리얼서버의 장애는 서버 모니터 시스템과 리얼서버 모니터를 이용하여 이중으로 감지하고 장애상황을 관리자에게 더욱 신속하고 정확하게 보고 하도록 감시시스템을 운영한다.

#4. HA 메커니즘이 활성서버 장애를 자동으로 감지해 대기 서버로 페일오버(failover)한다.

5. 시스템 성능 측정기준

제안하는 방법론의 시스템 성능 측정기준을 제시하여 고가용성 클러스터링 구조의 효율성 여부를 측정한다. 시스템 performance, latency, 응답시간(response time), CPU 부하율(CPU utilization), CPU 상의 시스템 프로세스(system proces s)수로 대표 된다. Performance는 인프라구조에 의해 부작용으로 발생하는 시스템의 부하 수준이다. 응답시간은 호스트 접속 요구 패킷을 송신 개시하고 호스트로부터 응답 시간 패킷을 수신 완료할 때까지의 시간을 의미하는

것으로 응답시간은 네트워크 전송시간 + 서버 처리 시간 + 클라이언트 처리 시간의 합계시간으로 산출된다. CPU 부하율은 CPU 사용율 의미한다.

◉ Performance

인프라 구조에 의해 발휘되는 대표적인 시스템의 성능 수준이다. Performance는 latency, 응답시간(response time), CPU부하율, CPU상의 시스템 Process 수로 나타난다. Performance는 부(-)가 지향 목표 수준이다.

▪ 측정단위 : 응답시간 지연율(%) 분석 지수

◉ 캐쉬 적중(cache hit)율

프로세서가 원하는 데이터가 캐쉬 내에 있을 때 적중률(hit ratio)이다. 프로세서가 원하는 데이터 가 캐쉬에 없을 때 캐쉬 미스(cache miss)이다.

▪ 측정단위 : 미스율 = (1-h)

h =프로그램과 데이터의 지역성(locality)

◉ CPU 부하율(CPU Utilization)

차단 시스템 상에서 CPU 사용율을 의미한다. CPU 사용량의 증감은 트래픽량(traffic volume)과 악성코드 등 유해 트래픽의 발생에 의해서 좌우된다.

▪ 측정단위 : 백분율(%)

◉ 시스템 프로세스(System Process)

차단 시스템 상에서 수행되는 process수를 의미, 시스템 상에서 생성된 process가 memory상에 loading되면 process가 가동된다. 시스템 부하의 수준에 따라 process가 증가하거나 감소.

▪ 측정단위 : Process수

◉ 응답시간(Response Time)

호스트 접속 요구 패킷을 송신 개시하고 호스트로부터 응답시간 패킷을 수신 완료할 때까지의 시간을 의미하는 것으로 응답시간은 네트워크 전송 시간 + 서버 처리 시간 + 클라이언트 처리 시간의 합계시간으로 산출된다.

▪ 측정단위 : 초(second)

◉ 평균 기억장치 액세스 시간

캐쉬 액세스 시간, 주기억장치 액세스 시간을 감안 한 평균 기억장치 액세스 시간

▪ 측정단위 : 평균 액세스 시간(MS)

6. 결 론

정보시스템 고가용성 보증을 위해 클러스터링은 가변적 업무부하를 처리하거나, 서비스 연속성을 저해하는 고장 발생 시 운영이 계속되도록 여러 대의 컴퓨터시스템 기능을 연결하는 메커니즘이다. 고 가용성 클러스터링 기능은 가능한 오랜 시간 서버 시스템이 작동하는데 중점을 둔다. 이 클러스터는 멀티플 시스템에서 실행되는 노드와 서비스를 중복하여 가지고 있어서 서로가 서로를 추적할 수 있다. 한 노드 장애 발생 시 둘째노드가 몇 초 이내에 고장난 노드 임무를 대행 한다. 상대방 컴퓨터에 고장 발생 시 상대 서비스를 인수한다. 일반적 단일 부하분산 서버 클러스터링은 하나의 부하 분산기를 두고 모든 트래픽을 조절하는 기능을 수행 한다. 데이터베이스는 클러스터가 노드 장애 시 작동을 시작할 수 있도록 실제 저장소에 위치한다. 단일구조 상태의 취약점은 master 서버의 장애 시 마스터 서버를 재구성하는 작업이 수동으로 진행 된다. 본 연구를 통해 제안된 방법은 기업업무 운영 현장에서의 서비스 연속성을 보증하기 위한 것으로 향후 현장의 참고로 활용되기를 기대한다.

참고문헌

[1] 시스코코리아, 시스코레포트, "네트워크의 가용성을 높여라", 2004.

[2] 구자만, "고가용성으로 보안 장비 한계를 극복하라", 네트워크타임스, 2003.

[3] Sichoon Noh, Dong Chun Lee, and Kuimam J.Kim, "Improved Structure Management of Gateway Firewall Systems for Effective Networks Security", Springer, 2003.

[4] D. Dias, W. Kish, R. Mukherjee and R. Tewari, “A Scalable and Highly Available Server”,

COMPCON 1996, pp. 85-92, 1996.

[5] Xuehong gan, Trevor Schroeder, Steve Goddard and Byrav Ramamurthy", in submission to The Eighth IEEEinternational Conference on Computer Communications and Networks , 1999

[6] G. Hunt, G. Goldszmid, R. King, and R. Mukherjee,"Network Dispatcher: A Connection Router for Scalable Internet Service, Computer Networks and ISDN Systems, Vol.30, pp.347 357, 1998.

[7] D. Dias, W. Kish, R. Mukherjee and R. Tewari ," A scaleable and highly available Web server" IEEE international Conference on Data Engineering. New Orleans, February 1996.

[8] G. Trent and M. Sake, WebStone, "the First Generation in HTTP Benchmarking", MTS Silicon Graphics, February 1995.

[저 자 소 개]

김 점 구 (Jeom-goo Kim)

1990년 2월 광운대학교
전자계산학과 이학사
1997년 8월 광운대학교
전자계산학과 석사
2000년 8월 한남대학교
컴퓨터공학 박사
1999년 3월~ 현재 남서울대학교
컴퓨터학과 정교수
IT융합연구소장

email : jgoo@nsu.ac.kr

노 시 춘 (Si-Choon Noh)

1987년 : 고려대학교
경영정보학(석사)
2005년 : 경기대학교
정보보호기술(박사)
2002년 : KT 시스템보안부장
2004년 : KT 충청전산국장
2005년~현 재 : 남서울대학교
컴퓨터학과 교수
2011년~현 재 : 남서울대학교
IT융합연구소 연구위원

email : nsc321@nsu.ac.kr