工學碩士 學位論文

HACMP 와 OPS의 구축 설명과 구축 후 장애 극복 사례

指導敎授 權 五 欽

이 論文을 [illegible] 學位論[illegible] 提出함

2003년 2월

釜慶大學校 産業大學院

컴 퓨 터 공 학 과

徐 振 碩

이 論文을 徐振碩의 工學碩士
學位論文으로 認准함

2002년 12월 18일

主　審　工學博士　鄭　木　童

委　員　工學博士　徐　庚　龍

委　員　工學博士　權　五　欽

목 차

그 림 목 차

표 목 차

A study on the troubleshooting in HACMP and OPS system

Jin-Seok Seo

Department of Computer Engineering

Graduate School of Industry

Pukyong National University

Abstract

I propose that this thesis should explain the setting files and formations needed among several kinds of building techniques in order to build the IBM's HACMP provided with the high solubility solution.

Particularly, this thesis is divided into three kinds of building techniques and definitely exposed to the characteristics and usages.

And it is represented to classify the disable examples after the building of HACMP, which a company now uses through the Mode 3 technique, into systems, adapters, networks

I hope that people who are interested in this thesis or are using this technique will be very helpful.

First of all, let me examine how to draw up, use, and form the internal scripts.

제 1 장 서 론

고 가용성(high availability)은 하드웨어, 소프트웨어, 네트워크 등에 문제가 발생하였을 때 서비스를 지속할 수 있도록 해주는 기술을 말한다. 고 가용성은 정보화 시대의 도래, 전자 상거래의 발전 및 정보 및 멀티미디어 서비스 등의 컴퓨팅 환경의 변화에 따라 점차 중요시되고 있다.

특히 24시간 일년 내내 서비스를 제공해주어야 하는 네트워크 컴퓨팅 부분에서 고 가용성은 필수 불가결한 것으로 인식되고 있다.

매일 24시간 서비스를 제공해주어야 하는 ISP, 멀티미디어 서비스 업체, 하역업체, 은행 등의 기업체들은 30분 혹은 단 1~2분이라도 서비스가 중단되어서는 안 된다. 특히 갑작스런 서비스 중단은 회사에 엄청난 피해를 가져올 수도 있다. 1995년에 Oracle과 Datamation의 보고서에 의하면 갑작스런 서비스 중단은 평균적으로 기업체에 시간당 80,000~350,000 달러의 손실을 입히는 것으로 나타났다. 또 다른 예로 1993년 미국의 세계 무역센터 폭발 사고로 빌딩에 입주해 있던 350여 개의 회사 중에서 145개회사가 자료 손실로 도산한 것으로 나타났다. 이러한 사례에서 볼 수 있듯이 고 가용성은 정보화 시대의 모든 기업들에 꼭 필요한 요소이다. 24시간 365일 작업을 수행할 수 있는 server가 되기 위해서는 무언가 고 가용성의 solution이 요구되었고, 그 결과로 IBM에서 내놓은 것이 HACMP이다.

그래서 본 논자는 A기업에서 IBM 시스템에 HACMP를 이용한 고 가용성 서버를 바탕으로 HACMP의 비정상 작동 시의 문제들과 DataBase Locking 문제를 가지고 아래와 같이 구성하였다.

제2장에서는 HACMP의 기본개념을 알아보고[1-3], 제3장에서는 신규 시스템의 도입배경을 알아보고, 제4장에서는 HACMP로 구성할 수 있는 구성방식들과 현 시스템의 Mode-3(OPS와 HACMP)를 구성한 형태를 설명하고[2], 제5장에서는 현재 구성된 Mode-3에서의 문제진단 방법과 해결 사례를 알아보고[4-5], 제6장에서 결론을 내린다.

제 2 장 HACMP의 기본개념

2.1 기본 작동 원리

HACMP의 개념은 흔히 high availability, 또는 fault resilient라고 하는 것으로서, 두 대 이상의 system을 하나의 "cluster"로 묶어서, 각 system이 감시하고 있다가 한 대에 장애가 발생하면, 다른 system이 장애가 발생한 system의 자원을 "takcover(인수)"하는 것이다. 즉, 장애 발생 시에는 자원을 분명히 system service의 중단이 일어나게 되며, 이 중단 시간을 최소화 하는 것이 HACMP의 기능인 것이다. 대개의 경우, 중단 시간은 30초에서 300초에 이르게 된다.

이와 비교되는 개념으로는 fault tolerant system이 있다. 이는 system box 단위의 중복 설치보다는 하나의 system box 내부에서 hardware 부품 단계에서 중복 설계를 함으로써 일부 부품에 장애가 발생하더라도 system down이 전혀 일어나지 않도록 되어있는 개념이다.

이 두 개념을 비교하면 다음과 같이 요약할 수 있다.

표 2.1 Fault resilient와 Fault tolerant의 비교

비교 항목	HACMP	Fault Tolerant
Failover time	30-300초	0
Concurrent 유지보수	불필요	필수
동일성능 system 대비가격	2배	10-20배
Application	거의 제한 없음	제한적
운영체제	일반 AIX	독자적 운영체제 사용
사용 하드웨어 장치	일반 RS/6000 하드웨어 장치	독자적 특수 하드웨어 사용

위의 표에서 알 수 있듯이 HACMP는 fault tolerant system과 비하여 나름대로의 장단점이 있다. 어느 system을 선택해야 하는지 결정하는데 가장 큰 요소는 무엇보다도, 5분 정도의 down-time이 용납되는 상황인지에 대한 평가일 것이다.

HACMP는 clstmgr(cluster manager), clsmuxpd(cluster SNMP agent), clinfo(cluster information service), cllockd(cluster lock manager) 등의 요소로 구성되어 있다. clstmgr은 clsmuxpd를 통해 다른 노드(system)와 keepalive 또는 heartbeat이라 불리는 패킷을 교환한다. 이 keepalive 패킷에 대한 응답을 받지 못하면, 그 응답을 주지 못하는 노드에는 무언가 장애가 발생한 것으로 인식되고, 그에 따라 미리 정해진 복구 스크립트가 수행된다. 이 keepalive path는 노드간에 구성된 TCP/IP LAN, 그리고 RS-232나 target mode SSA 등의 serial network을 이용한다.

이 복구 스크립트에 의해 복구되는 장애는 2.1.1 system 전체의 장애, 2.1.2 network adapter의 장애, 2.1.3 network 전체이다.

2.1.1 system 전체에 장애 발생시

한 노드로부터 Keepalive 패킷이 전혀 오지 않는 경우, backup 노드는 그 노드가 down되었다고 판단하여 그 노드가 가지고 있던 자원, 즉 disk volume group, filesystem, IP address, application등을 takeover 해서 자기가 대신 service를 수행한다. 이때, client는 잠시 service 중단을 겪게 되고, telnet session등을 맺고 있던 것은 끊어져서 다시 맺어야 한다.

2.1.2 network adapter 장애 발생시

대개 노드는 1장의 service adapter와 1장 이상의 standby adapter를 가진다. 이때, service adapter에 장애가 발생하면 그 IP address는 같은 노드

내의 standby adapter로 swapping이 일어난다. 이때는 client들은 30초 미만의 hang을 겪을 뿐 대부분의 service를 계속 받을 수 있고, telnet 등도 끊어지지 않는다.

2.1.3 network 전체 장애 발생시

network 장애가 발생하는 경우엔 clstrmgr은 일단 network_down event를 발생시킨다. 이때 실행되어야 할 복구 스크립트는 HACMP에서 default로 제공되지는 않는다. 이는 network 구성이 site마다 너무 다양하기 때문에 표준 스크립트를 작성하기가 곤란하기 때문이다. Dual network이 구성된 경우엔, primary network이 down되면 backup network이 takeover 하도록 HACMP를 customize할 수 있다.

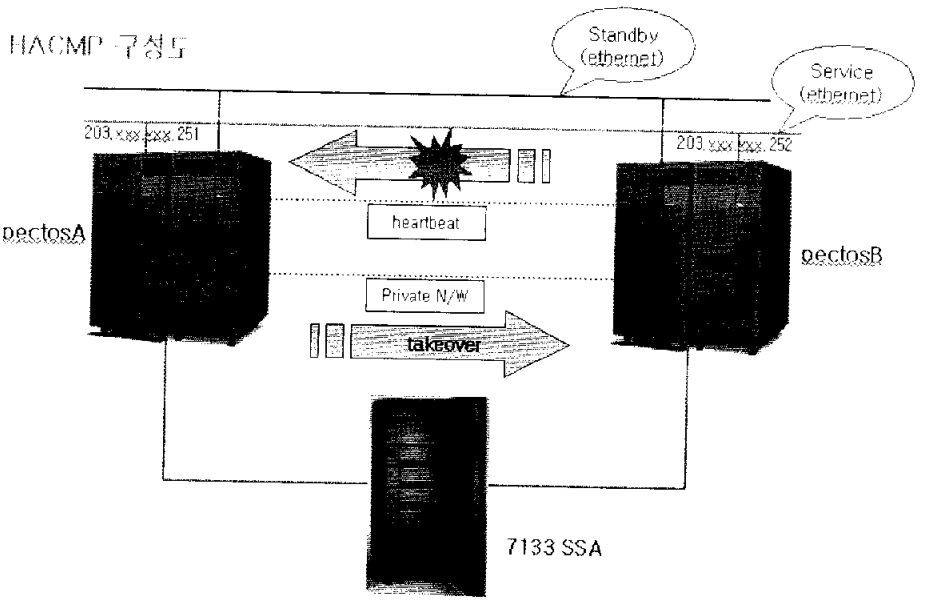

그림 2.1 HACMP가 구축된 시스템형태

2.2 HACMP를 이용하여 보호할 수 있는 자원

HACMP로 보호되는 자원은 다음과 같다.

2.2.1 **Disk volume group** - backup 노드에서 varyon되어 사용 가능해진다.

2.2.2 **Filesystem (with NFS export/mount)** - backup 노드에서 mount되고, NFS export/mount 까지 수행된다.

2.2.3 **Concurrent volume group** - Concurrent volume group을 concurrent mode로 varyon한다.

2.2.4 **IP address** - fail된 service IP address를 standby adapter에 config한다.

2.2.5 **Application** - 위의 복구가 다 수행되면 주어진 스크립트에 따라 application을 restart한다.

이러한 자원을 takeover 처리하는 방법에는 다음 3가지 종류가 있다.

Hot standby - 아무 업무를 하지 않고 대기 상태인 standby server가 active server를 감시하고 있다가 active server에 장애 발생시에 자원을 takeover하는 방식

Mutual takeover - 두 server가 각각 자신의 일을 수행하며 서로를 감시하다가 상대방에 장애가 생기면 그 자원을 takeover하여 자신이 하던 일과 인수받은 일을 같이 수행하는 방식

Concurrent access (mode 3) - Mutual takeover 방식을 그대로 수행하며, 추가로 Oracle parallel server를 사용하여 하나의 volume group에 여러 server가 동시 access하여 병렬처리를 하는 방식

2.3 HACMP cluster를 구성하기 위한 조건

HACMP cluster를 구성하기 위해서는 다음과 같은 전제 조건이 충족되어야 한다.

1) Cluster를 구성하는 노드간에 TCP/IP LAN이 구성되어야 한다.

2) 노드 사이의 LAN에는 최소한 노드당 2개씩의 adapter가 있어야 한다. (service adapter와 standby adapter)

3) 각 노드의 service IP address 및 stabdby IP address끼리는 같은 subnet에 속해 있어야 하고, service IP address와 stabdby IP address는 반드시 다른 subnet에 속해 있어야 한다. 즉, service IP와 standby IP는 하나의 물리적인 network에 연결되어야 하지만, 동시에 그 둘 사이에 ping이 되어서는 안된다.

4) 노드 사이에는 TCP/IP LAN 외에도 1개 이상의 serial network(RS-232, target mode SSA 등)을 구성하는 것이 강력하게 권고된다.

5) 노드 간에 외장 disk를 공유하는 경우, SSA이거나 differential SCSI disk 이어야 한다.

6) Cluster를 구성하는 노드들의 AIX version 및 HACMP version을 일치시켜 주어야 한다.

제 3 장 고 가용성 시스템 도입배경

3.1 고 가용성 시스템 도입 및 대체 필요성

3.1.1 전산화 업무의 확장과 처리물량 증가에 따른 처리 속도가 저하되고 있다.

3.1.2 신식 증설 및 하역장비 추가에 대한 처리능력 사전 확보가 필요하다.

- C/C 및 T/C 증설에 따른 무선단말기 접속 기능 확대가 요구된다.

3.1.3 부산항 신설 터미널의 첨단 시스템 도입에 대응해야 한다.

3.1.4 Rightsizing을 통한 정보시스템의 재편성으로 기업경쟁력 강화를 필요로 한다.

3.2 현행문제점

3.2.1 현행 전산처리 기법의 후진 및 기기 성능 저하를 가져오고 있다.

3.2.2 장애 시 대처 능력이 저하되고 있다.

- 이원화 주전산기 구성으로 장애 시 전환에 복구시간 과다소요

- 물류 처리 시스템 고장 시 대체 시스템(관리업무)으로 전환에 따라 관리업무 전산처리 실시 중지(HA 기능불능)

3.2.3 시스템 장애 시 현행 File관리 체계로는 File복구에 장시간 소요

현 주전산기는 '90년에 생산된 단순 직열 처리 기종으로 현행 다중병렬 기종에 비해 단위 시간당 처리량이 적고, 처리속도가 매우 느리다.

3.3 개선방안

3.3.1 주전산기(Main Hardware System : Server)의 무 정지 운전 체제

구축 : 24시간 연속작업 가능

3.3.2 현행 주전산기 집중처리방식을 다수의 Workstation으로 특성별 업무 분산처리

3.3.3 기기 운영방식 개선

- Hot Stand-by 혹은 Fault-tolerance(Non-stop) 방식 채택

3.4 기대효과

3.4.1 무 장애 혹은 긴급 복구체계 시스템 구현으로 시스템 운영 요원의 상시 대기가 불필요하다.

3.4.2 현행 집중 처리방식을 분산처리 방식으로 시스템을 구현함에 따라 장애 대처 능력이 향상된다.

3.4.3 다중 병렬처리 시스템 도입으로 처리속도 및 시스템 신뢰도 향상

3.4.4 현행 집중처리방식을 업무 특성별 분산처리방식으로 변경 장애대처 능력이 향상된다.

제 4 장 HACMP 4.2.2 FOR AIX 4.3의 구축

4.1 하드웨어와 소프트웨어 자원

본 절에서는 HACMP를 구축하기 전 사용되어진 하드웨어와 소프트웨어 자원을 알아본다.

4.1.1 RISC SYSTEM/6000

- 호스트 이름 : pectosA, pectosB
- 모델명 : IBM System Rack S70
- 메모리 : 1024 MB
- 하드 디스크 : Internal - 4GB 2개
- Network : FDDI Adater 2개

  fddi0 20-70 SysKonnect PCI FDDI Adaper (48110040)

  fddi1 40-60 SysKonnect PCI FDDI Adaper (48110040)
- Disk Array Adapter : SSA Adapter 2개

  ssa0 20-68 IBM SSA Enhanced RAID Adapter(14104500)

  ssa1 30-70 IBM SSA Enhanced RAID Adapter(14104500)

4.1.2 External Hard Disk

- IBM 7133-SSA
- 하드 디스크 용량 : 72GB

4.1.3 소프트웨어 자원

- 운영체제 : AIX 4.3
- 고 가용성 지원 소프트웨어 : HACMP for AIX 4.2.2

· 데이터 베이스 : ORACLE Parallel Server 7.3.4

4.2 HACMP를 구축하기 전 환경설정

본 절에서는 HACMP를 구축하기 전 네트웍 구성형태와 시스템 내에 필요한 환경 파일들을 알아본다. 그리고 공유 LVM구성과 LVM관리를 정의한다. 사용되어진 클러스트의 정의된 구조와 이 클러스트에 사용되어진 응용 스크립트들을 살펴보도록 한다.

4.2.1 Network 구성

· FDDI Network Adapter 구성

표 4.1 FDDI Network Adapter 구성

구분	노드1(pectosA_node)	노드2(pectosB_node)
Service(fi0)	203.XXX.XXX.251	203.XXX.XXX.252
Boot(fi1)	203.XXX.XXX.101	203.XXX.XXX.102
Standby(fi1)	192.XXX.XXX.251	192.XXX.XXX.252
Default Gateway	203.XXX.XXX.6	203.XXX.XXX.6

· Serial Network(RS 232C) 구성

표 4.2 Serial Network(RS 232C) 구성

구분	노드1(pectosA_node)	노드2(pectosB_node)
RS 232	tty1	tty1

4.2.2 HACMP를 구축하기 위한 환경 화일 편집

: 각 노드(pectosA_node, pectosB_node)에서 작업

/etc/hosts

: Cluster 노드의 TCP/IP Adapter를 모두 기입해야 하며 노드간에 동일하게 구성이 되어야 한다. (Service, Standby, Boot, Switch IP Address)

vi /etc/hosts

203.xxx.xxx.101	pectosA_boot
203.xxx.xxx.251	pectosA
192.xxx.xxx.251	pectosA_stb
203.xxx.xxx.102	pectosB_boot
203.xxx.xxx.252	pectosB
192.xxx.xxx.252	pectosB_stb

/.rhosts

: HACMP Configuration에서 rsh을 사용하므로 노드간 동일하게 구성을 해야하며 HACMP 구성과 점검이 완료되면 보안을 위해 /.rhosts 파일을 삭제하는 것이 좋다.

vi /.rhosts

pectosA

pectosB

pectosA_boot

pectosB_boot

pectosA_stb

pectosB_stb

pectosA_tty

pectosB_tty

/etc/services

: 응용프로그램이 어느 system에서든지 정상적으로 동작할 수 있도록 /etc/services를 동일하게 구성해야 한다.

/etc/inittab

: system booting시 load 되는 file중 고 가용성을 요하는 응용프로그램에 영향을 주는 내용이 있으면 양쪽 system에서 공히 load되도록 해줘야 한다.

위의 환경설정이 끝난 후 Network Configuration Test로 아래방법을 취한다.

노드1(pectosA_node)에서

ping pectosB_boot

ping pectosB_stb

노드2(pectosB_node)에서

ping pectosA_boot

ping pectosA_stb

4.2.3 공유되어 사용되어지는 LVM 구성요소들 정의

OPS(Oracle Parallel Server)

Share Disk에 Concurrent LVM(Logical Volume Manager)을 구성 할 때 File System이 아닌 Raw logical volume을 사용하여야 한다.

Raw logical volume으로 CLVM생성시 /dev Directory에 Userid "Oracle" Group "dba"인 raw logical file을 만들어 주어야 한다.

Raw logical volume 이란?

Filesystem을 만들어 mount를 시키는 개념이 아니라 logical volume만을 만들어 그 공간을 직접 하나의 device처럼 사용하는 개념이다.

LVM 구성

일반적으로 유닉스 시스템에서 저장 장치를 쓰는 방법은 그 장치의 블록 디바이스(Block Device)에 파일 시스템 (File System)을 만들어서 (다른 표현으로는 포맷(Format)한다라고 하지만 유닉스의 세계에서는 잘 쓰지 않는다.) 디렉토리에 마운트 시키는 것이다.

LVM을 써도 마찬가지 절차를 밟는다. 단지, 실제 블록 디바이스가 아닌 가상의 블록 디바이스를 쓴다는 점이 틀리다.

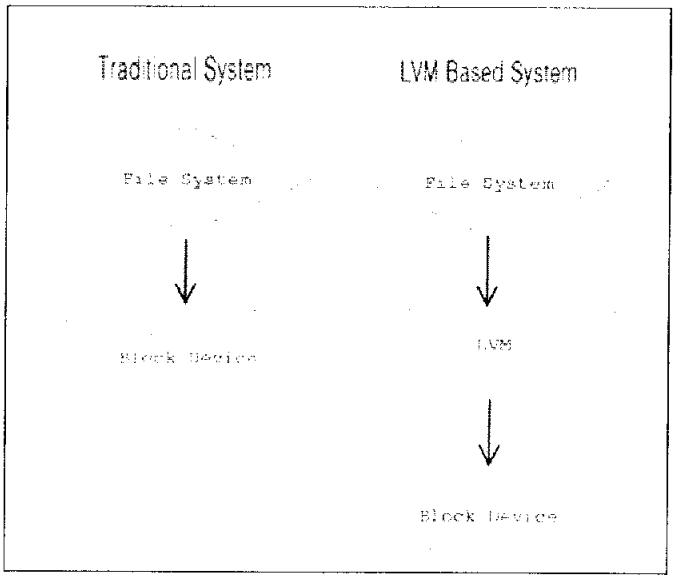

그림 4.1 전통적인 파일 구조와 LVM을 기반으로 하는 파일 구조

- Volume Group : pectvg
- Physical Volume(source | mirroring)

PV Name	PV Size	PV Name	PV Size
hdisk1	4.5GB	hdisk9	4.5GB
hdisk2	4.5GB	hdisk10	4.5GB

hdisk3	4.5GB	hdisk11	4.5GB
hdisk4	4.5GB	hdisk12	4.5GB

- Logical Volume

LV Name	PPs	LV Name	PPs
log1_a	2	data5	63
log2_a	2	data6	89
controlfile1	1	data7	125
.	.	.	.
.	.	.	.
.	.	.	.
data4	63	data25	46

LVM 관리

볼륨1 (pectosA)

1) Create Volume Group

: 볼륨 그룹을 아래형태로 만든다.

smit mkvg

- VG Name : pectvg
- PP Size : 8 MB
- PV Name : hdisk1, hdisk2, hdisk3, hdisk4

Major number : 53 (pectosA)

Active VG at system restart? : No

2) Create raw logical volume

: 1)의 볼륨그룹 내에 속하는 여러 raw logical volume을 아래의 형태로 만든다.

smit mklv

Add a Logical Volume

- Logical volume NAME : log1_a
- VOLUME GROUP name : pectvg
- Number of LOGICAL PARTITION : 2
  PHYGICAL VOLUME name : hdisk1
  Mirror Write Consistency? : no
- Enable BAD BLOCK relocation? : no

3) Varyoff VG

   # varyoffvg pectvg

4) Change ownership

   # chown oracle.dba filename

노드2(pectosB)

5) Export VG

   # exportvg pectvg

6) Import VG

   # importvg -V 'major number' -y pectvg hdisk1

   * 위의 major number는 각 노드의 값을 넣어 준다.

     pectosA(53), pectosB(51)

   # smit vg

     import a Volume Group

   - VOLUME GROUP Name : pectvg
     PHYSICAL VOLUME Name : hdisk1
     ACTIVATE Volume group after it is imported? : yes
   - Volume Group MAJOR NUMBER : 53

7) Change VG

smit chvg

- VOLUME GROUP name : pectvg
- Activate volume group AUTOMATICALLY : no
  at system restart ?
- A QUORUM of disks required to keep the volume : no
  group on-line ?

8) Change ownership

* Import 시 그 볼륨그룹의 속성이 변하게 되므로 반드시 그 속성을 변화시켜 주어야 한다.

chown oracle.dba filename

9) Varyoff VG

varyoffvg pectvg

LVM 변경 시 고려사항

장유 Component들은 노드들 사이에 동일하게 정의되어야 하므로 LVM 요소의 변경은 모든 노드의 ODM에 반영되어야 한다. 필요하다면 HACMP ODM definitions가 shared LVM components에 일어난 변화들을 반영할 수 있도록 update 되어야 한다. 하나의 volume group이 export되고 reimport될 때 그 volume group의 ownership과 permission은 root와 system group으로 reset 된다. 그런데 database server와 같이 raw logical volume을 사용하는 application들은 이에 영향을 받을 수 있으므로 필요하다면 ownership과 permission을 원래 대로 해 놓아야 한다.

4.3 Cluster 정의

4.3.1 Cluster ID 및 Name 구성

Cluster ID : 1

- Cluster Name : pectos

4.3.2 Cluster 노드 구성

- 노드 Name : pectosA_node, pectosB_node

4.3.3 Cluster Adapter 구성

- 노드 1 : pectosA_node

표 4.3 pectosA_node의 Cluster Adapter 구성

Adapter IP Label	Network Type	Network Name	Network Attribute	Adapter Function	Adapter Identifier
pectosA	fddi	fddia	public	service	203.XXX.XXX.251
pectosA_boot	fddi	fddia	public	boot	203.XXX.XXX.101
pectosA_stb	fddi	fddia	public	standby	192.XXX.XXX.251
pectosA_tty	rs232	rs232a	serial	service	

노드 2 : pectosB_node

표 4.4 pectosB_node의 Cluster Adapter 구성

Adapter IP Label	Network Type	Network Name	Network Attribute	Adapter Function	Adapter Identifier
pectosB	fddi	fddia	public	service	203.XXX.XXX.252
pectosB_boot	fddi	fddia	public	boot	203.XXX.XXX.102
pectosB_stb	fddi	fddia	public	standby	192.XXX.XXX.252
pectosB_tty	rs232	rs232a	serial	service	

4.3.4 Resource Group 정의

Resource Group Name : pectos_rg

Node Relation ship : concurrent

Participating Node Name : pectosA_node, pectosB_node

Resource Group Name : pectosA_rg

Node Relation ship : cascading

Participating Node Name : pectosA_node

Resource Group Name : pectosB_rg

Node Relation ship : cascading

Participating Node Name : pectosB_node

4.3.5 Application Server 정의

Application Server Name : opsA

Application Start Script : /usr/sbin/cluster/app/opsA_start

Application Stop Script : /usr/sbin/cluster/app/opsA_stop

OPS(Oracle Parallel Server)환경에서 hacmp를 작동 시 양 노드에서 database를 start시키고 stop시키는 방법을 정의한다.(스크립트는 부록 참조)

4.3.6 Resource Group에 Resource 첨가

- pectos_rg

  Concurrent Volume groups : pectvg

  pectosA_rg

  Service IP label : pectosA

  Application Servers : opsA

  Application servers : 7318

- pectosB_rg

  Service IP label : pectosB

  Application Servers : opsB

4.3.7 File 편집

/usr/sbin/cluster/etc/clhosts (Service, Boot IP, address를 기입)

pectosA

pectosB

pectosA_boot

pectosB_boot

4.3.8 노드 환경의 구성

- Configuration run time parameter

  Node name : pectosA_node

Debug Level : high

Host uses NIS or Name Server : false

- Synchronizing Time Clocks

Time server의 생성

startsrc -s timed -a "-M"

chrctcp -S -c timed -f master='yes' f trace='yes'

4.4 HACMP 구성을 위한 3가지 방식

4.4.1 Mode 1

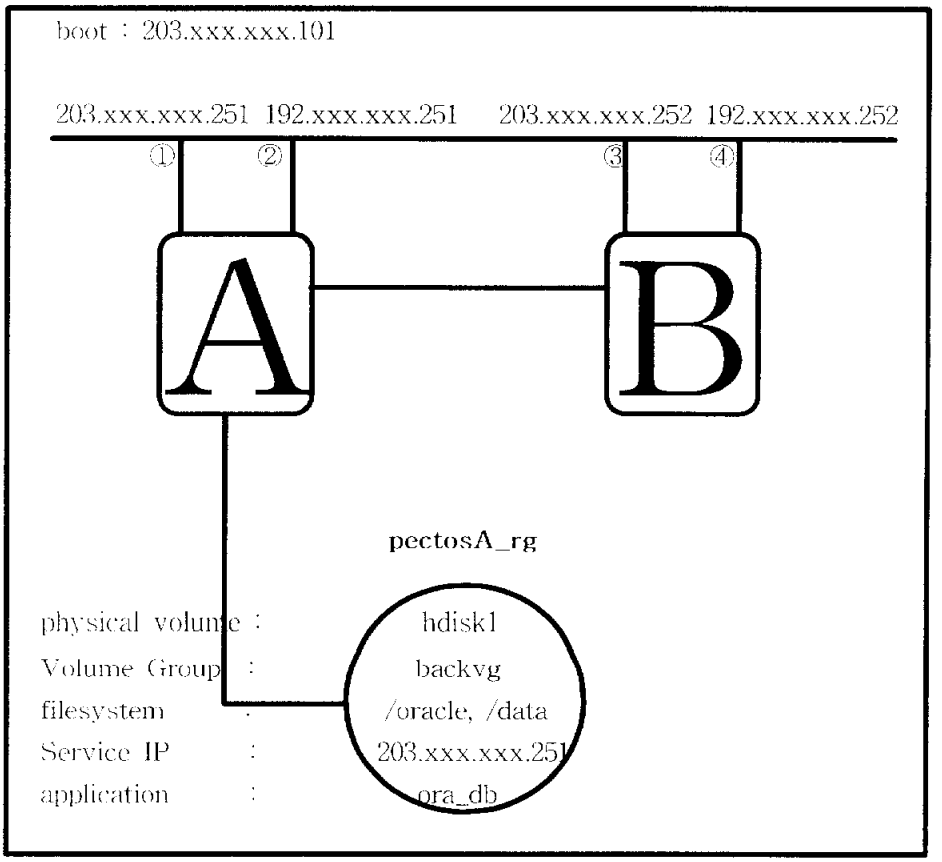

그림 4.2 HACMP start 후 정상적인 서비스 상태 & Adapter Fail

Cascading 방식(Hot standby)

B node 는 아무런 서비스도 하지 않는다.

①이 Fail 된 경우 ②가 IP address 203.xxx.xxx.251을 전달받아 서비스를 계속한다.

②가 Fail 된 경우는 아무런 변화도 일어나지 않는다.

즉 Service IP가 Fail된 경우만 adapter swapping이 일어나게 된다.

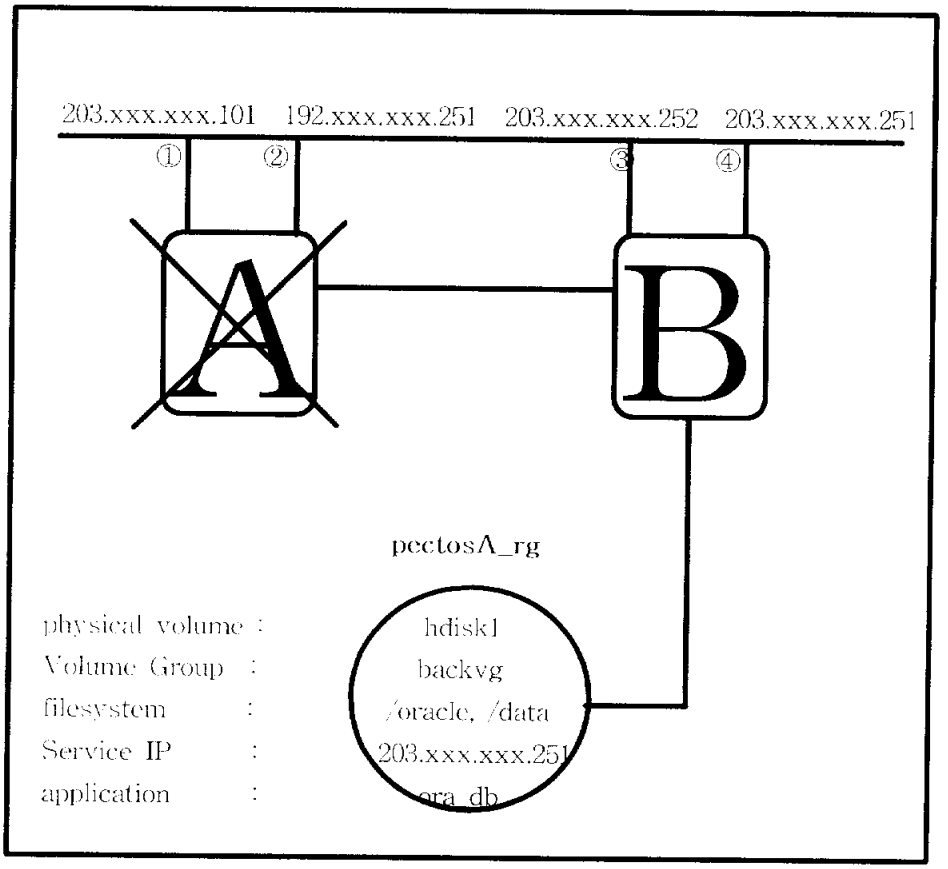

그림 4.3 노드 A 가 Fail된 경우

A 노드 Fail시 B 노드는 A 노드의 리소스의 권한을 이양 받아 서비스

를 수행하게 된다.

B 노드의 Standby Adapter인 ④는 A 노드의 서비스 IP address를 전달받아 서비스를 계속 수행한다.

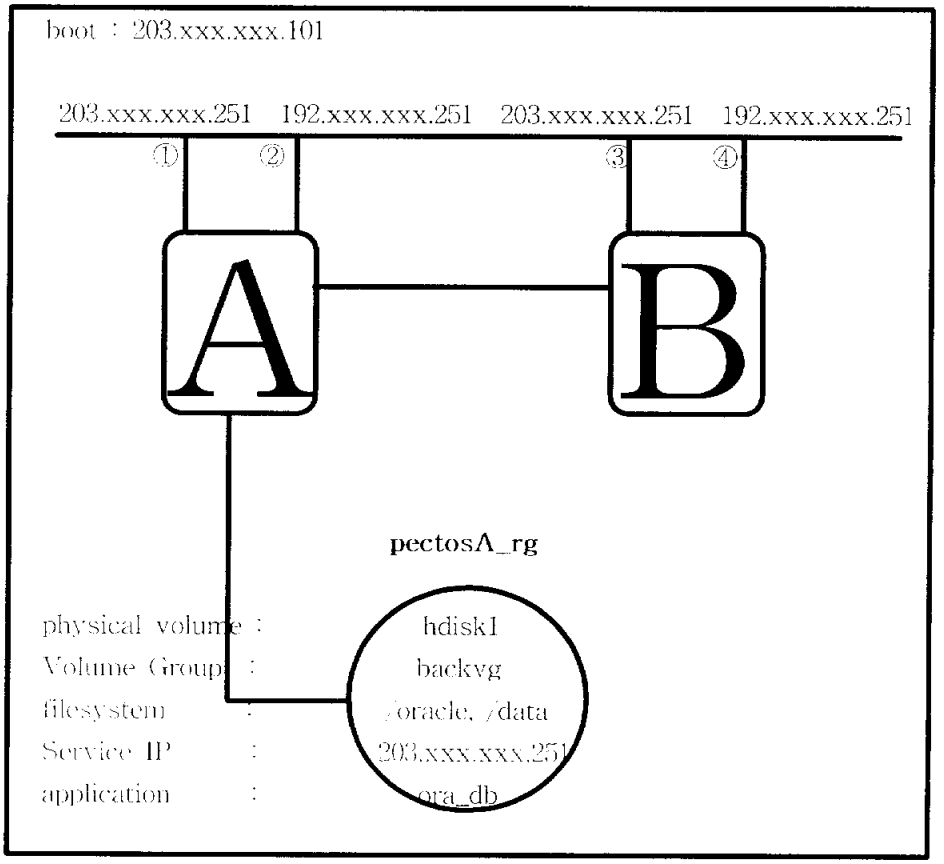

그림 4.1 Fail된 노드의 Reintegration

노드의 Reintegration시 A 노드는 일단 Boot IP Address로 결합한 후 Reintegration이 완료된 후 자신의 Service IP Address를 가져와 서비스를 수행한다. Hot Standby의 경우 노드의 Reintegration과 동시에 Fail 시 B 노드에 Takeover하였던 모든 리소스의 권한을 다시 A 노드에서 가져와 서비스를 수행한다.

Rotating 방식

Adapter Fail & 노드 Fail시 시스템의 동작은 Hot standby방식과 동일하나 노드 Reintegration 과정에서 Hot standby와는 다르게 Fail 되었던 A 노드가 모든 리소스의 권한을 다시 가져가는 것이 아니라 B 노드가 Fail 될 때까지 B 노드가 모든 리소스의 권한을 가지고 서비스를 계속한다. 즉 노드 Fail 시에만 리소스의 모든 권한을 다른 노드에 넘겨준다.

4.4.2 Mode 2

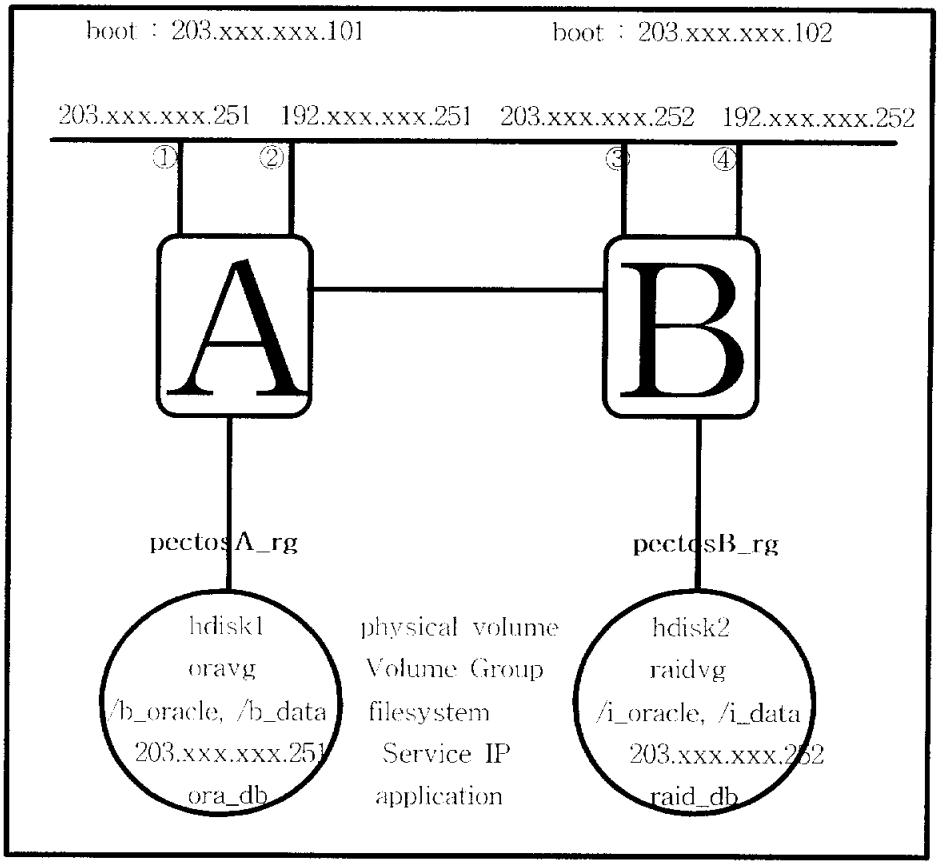

그림 4.5 HACMP start 후 정상적인 서비스 상태 & Adapter Fail

양 쪽 노드는 서로 다른 리소스 그룹을 가지고 서비스를 수행한다.

①이 Fail 된 경우 ②가 IP Address 203.xxx.xxx.252를 전달받아 서비스를 계속한다.

③이 Fail 된 경우 ④가 IP Address 203.xxx.xxx.252를 전달받아 서비스를 계속한다.

②나 ④가 Fail된 경우는 아무런 변화도 일어나지 않는다.

즉 서비스 IP가 Fail된 경우만 Adapter Swapping이 일어나게 된다.

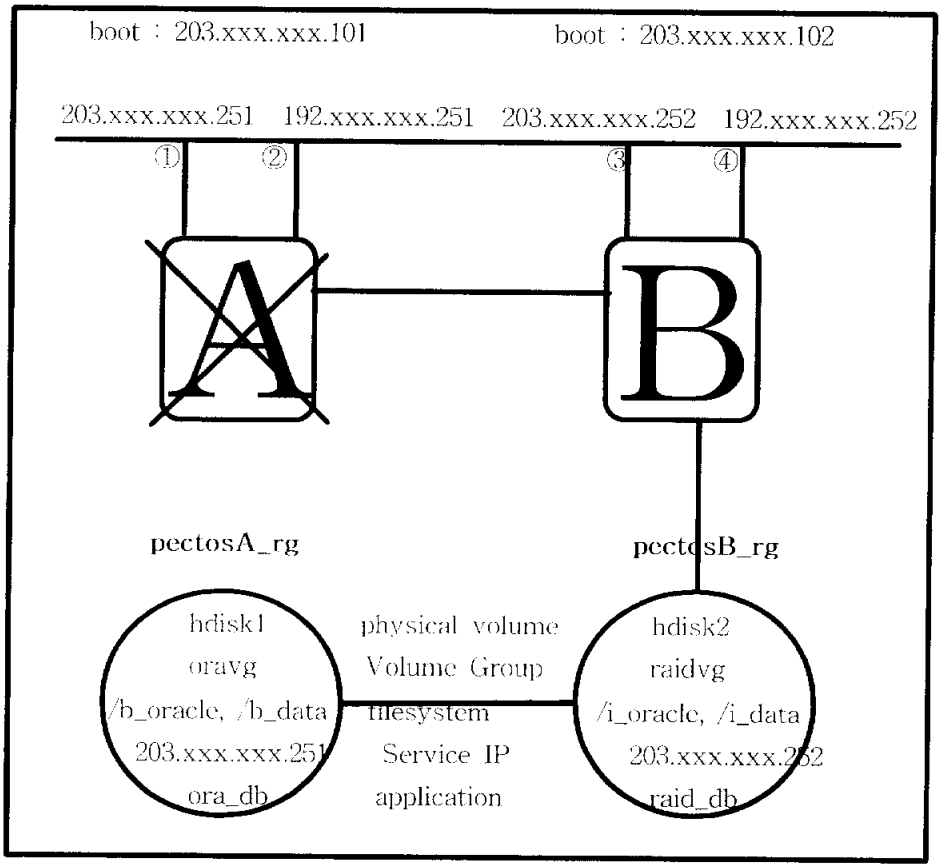

그림 4.6 노드 A 가 Fail된 경우

A 노드 Fail시 B 노드는 A 노드의 리소스의 권한을 이양 받아 기존에 서비스를 수행하던 리소스와 함께, 서로 다른 리소스를 하나의 노드에서 동시에 서비스하게 된다.

B 노드의 Standby Adapter인 ④는 A 노드의 서비스 IP Address를 전달받아 서비스를 계속 수행한다.

- Fail된 노드의 Reintegration

노드의 Reintegration시 일단 Boot IP address로 결합한 후 Reintegration이 완료된 후 자신의 서비스 IP address와 B 노드에 넘겨 주었던 리소스들을 가져와 서비스를 수행한다. 그림 4.5 참조

4.4.3 Mode 3(concurrent access)

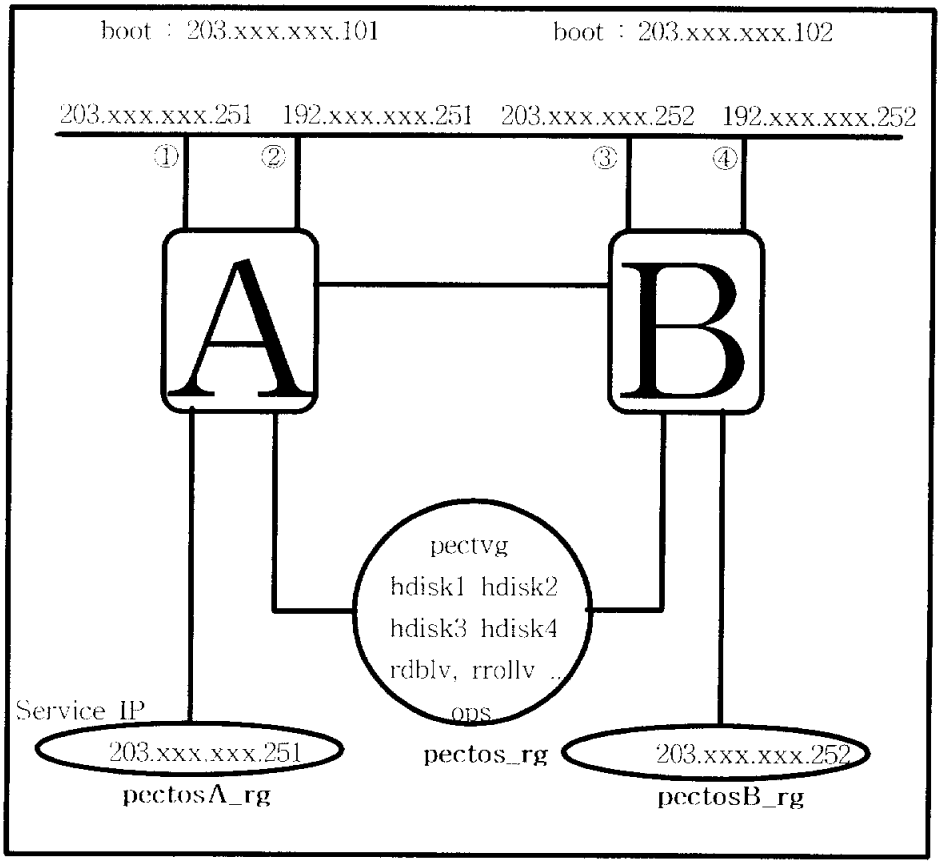

그림 4.7 HACMP start 후 정상적인 서비스 상태 & Adapter Fail

양 쪽 노드 는 Concurrent한 리소스 그룹을 가지고 서비스를 수행한다.

①이 Fail된 경우 ②가 IP Address 203.xxx.xxx.251을 전달받아 서비스를 계속한다.

③이 Fail 된 경우 ④가 IP Address 203.xxx.xxx.252를 전달받아 서비스를 계속한다.

②나 ④가 Fail된 경우는 아무런 변화도 일어나지 않는다.

즉 서비스 IP가 Fail된 경우만 Adapter Swapping이 일어나게 된다.

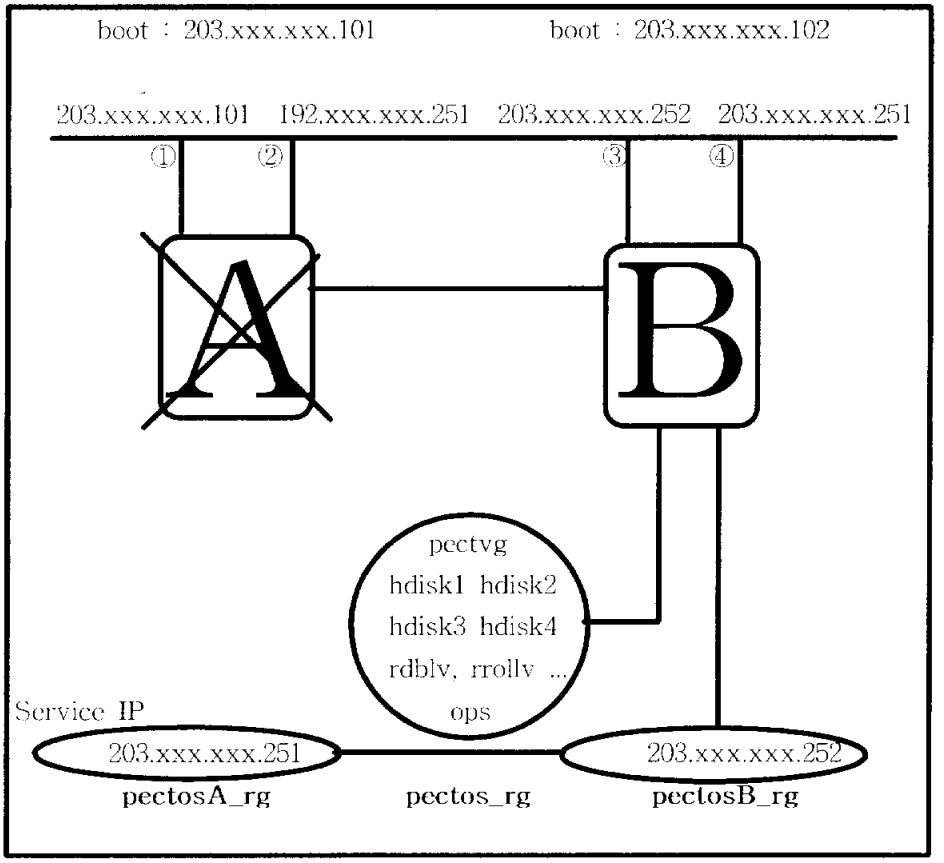

그림 4.8 노드 A 가 Fail 된 경우

B 노드 의 Standby Adapter인 ④가 A 노드의 서비스 IP Address를 전달받아 서비스를 계속 수행하고 A 노드를 통해 Concurrent하게 Access하던 리소스들을 B 노드를 통해 서비스 받게 된다.

Fail된 노드의 Reintegration

노드의 Reintegration시 일단 Boot IP address로 결합한 후 Reintegration이 완료된 후 자신의 서비스 IP address를 가져오고 B 노드를 통해 Concurrent하게 Access하던 리소스들을 자신의 노드를 통해 Access함과 동시에 서비스 하게된다. 그림 4.7 참조

제 5 장 현재 사용중인 시스템의 평가

5.1 HACMP의 장애 유형

: 현재 사용중인 HACMP의 장애유형을 알아보고 각 장애유형별로 사례를 알아본다.

5.1.1 시스템 장애유형

Disk Volume Group 장애

한 노드로부터 Keepalive 패킷이 전혀 오지 않는 경우, backup 노드는 그 노드가 down되었다고 판단하여 그 노드가 가지고 있던 자원, 즉 disk volume group, IP address, application등을 takeover해서 자기가 대신 service를 수행한다.

5.1.2 네트웍크 장애유형

네트워크 Adapter 장애

Service Adapter에 장애가 발생하면 IP address는 같은 노드 내의 standby adapter로 swapping이 일어난다. 이때는 client들은 30초미만의 hang을 겪을 뿐 대부분의 service를 계속 받을 수 있고, telnet등도 끊어지지 않는다.

네트워크 전체 장애

Network 장애가 발생하는 경우에는 clstrmgr은 일단 network_down event를 발생시킨다. Dual network이 구성된 경우에는, primary network이 down되면 backup network이 takeover하도록 HACMP를 customize해야 한다.

5.2 HACMP상의 장애 유형별 극복사례

5.2.1 시스템 장애 극복사례

비정상적 종료(치명적 오류)로 인해 rebooting되었을 때 booting시 rs232c 라인 점검 부분에서 E075 error code를 나타내고 시스템이 멈출 시 현 HACMP 7.3.4버전에서는 사람이 수동으로 케이블 분리작업을 하여 이 부분을 해결한다. 이후 버전에서는 케이블 분리작업 없이 booting shell에서 해결되어진다

Disk Volume Group 장애 극복사례

사용자(Client)가 데이터베이스 테이블 작업이나 Stored Procedure 프로그램작업 시 데이터베이스의 Background Daemon 중의 하나인 Lock Daemon과 HACMP의 클러스터링 Lock Daemon과의 동기화가 늦어지거나 이루어지지 않을 시 발생한다. 즉, HACMP가 작동하여 A 노드에서 B 노드로 takeover작업이 일어났는데도 불구하고 current 볼륨그룹이 제대로 넘어가지 않고 Data lock이 발생한 것처럼 보이는 현상이다. 이때 각 회사에서 운영되어지는 메뉴얼(수 작업) 방식으로 하나하나 해결해 나가는 방법이 있으며, 본 부분에서는 간단한 shell 프로그램으로 메뉴얼 방식에서 자동화 방식으로 해결하였다.(shell 프로그램은 부록 참조)[4,5]

각 Disk Volume Group의 ODM이 일치하지 않아 일어나는 현상 즉, 양 노드의 Volume Group내의 row device가 일치하지 않아 HACMP의 event가 발생 시 takeover가 일어나지 않게 하려면 평상시 양 노드의 Volume Group내의 변경된 환경을 clverify 유틸리티를 통해 점검하고 변경이 있을 시 ODM에 변경된 내용을 꼭 update해 두어야 한다.[9]

명령어 : cfgmgr -v

그리고 원인불명으로 시스템 hang현상 시 HACMP의 stop mode를 가

본 defalt mode인 graceful에서 forced로 전환 후 다시 한번 stop시킴으로서 고 가용성 소프트웨어에 의한 문제성을 해결 할 수 있다.

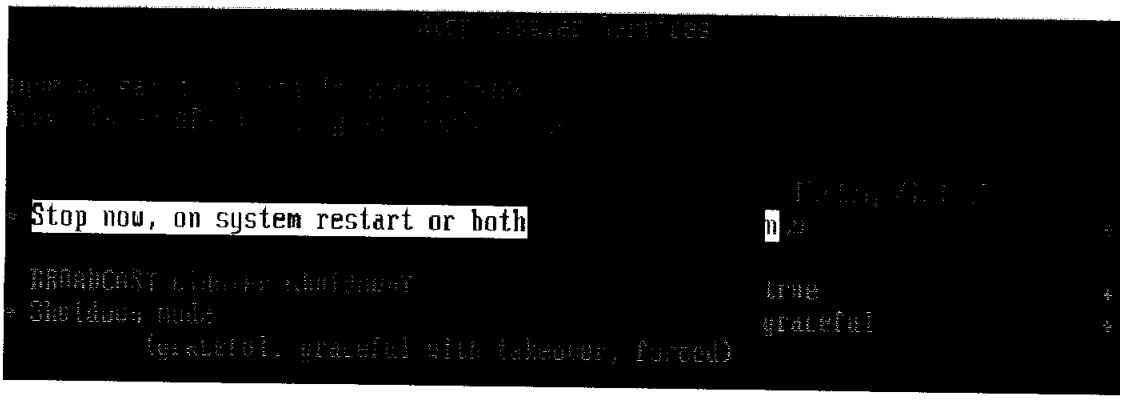

그림 5.1 고 가용성 소프트웨어 STOP 시 mode변경

5.2.2 네트워크 Adapter 장애 극복사례

양 노드간의 service ip address 와 standby ip address간 네트웍 통신에 장애가 발생 시, 대부분의 경우는 네트웍 swapping fail이 발생한 후 얼마 지나지 않아 네트웍 swapping completed가 나타난다. 그러나 사고 발생 시(hacmp event가 작동 시) 정상적인 네트웍 통신이 이루어지지 않아 문제가 발생하는 경우가 종종 있다. 이 경우 네트워크 장비 추가 작업이나 원거리 통신에 필요한 네트워크 작업 후 완전한 점검이 이루어지지 않은 상태에서 line 인결 작업을 했을 시 발생한다. 그리고 아직 정확한 원인과 해결책이 미흡 하지만, 현재의 해결책은 양 노드의 service ip address 끼리 혹은 standby ip address 끼리 cross ping test로 점검하고 있으며, 사고 발생 시는 smitty chinet로 강제 ip address변경을 통해 처리한다.[8]

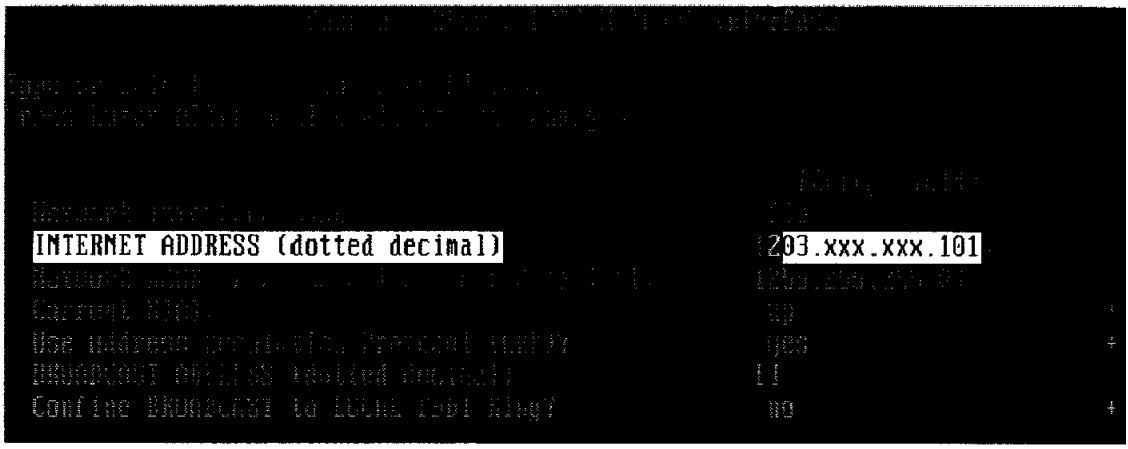

그림 5.2 네트워크 인터페이스 변경화면

한 노드에서 네트웍 adapter swapping이 정상작동 하지 않는 현상 즉, Service port의 ip address가 Standby port로 swapping event가 일어나지 않을 경우(Standby network port에 ping이 되지 않는다.) 양쪽 노드의 HACMP를 stop시킨 후 Service와 Standby port를 재 설정하고, Main 라우터 장비를 clear 시킨 후 HACMP를 up시킨다.[8]

"pdisable"에 의한 rs232 serial line(tty1) 장애 시 복구 절차

시간적 여유가 있을 경우

i. 양 노드에 hacmp 정지

ii. tty1 port setup

노드1 : # stty < /dev/tty1 (약간의 시간이 경과된다)

노드2 : # stty < /dev/tty1

양 노드간 port설정 값이 동일하게 표시되는지 확인한다.

iii. 만약, 동일하지 않을 시 포트 삭제 후 재구성을 필요로 한다.

- smitty tty
- remove tty

- port 선택 후 삭제

iv. port재 설정 작업을 한다.

smitty tty

- add tty

- port 설정(부록참조)

v. ii.항을 재 실시하여 정상인 경우 hacmp 기동 후 종료하고 비정상일 경우는 장애노드를 재부팅 후 일련의 과정을 재 작업한다.

시간적 여유가 없을 시

: standby 노드를 정지시키고 service 노드로만 현장 업무 지원을 한다.

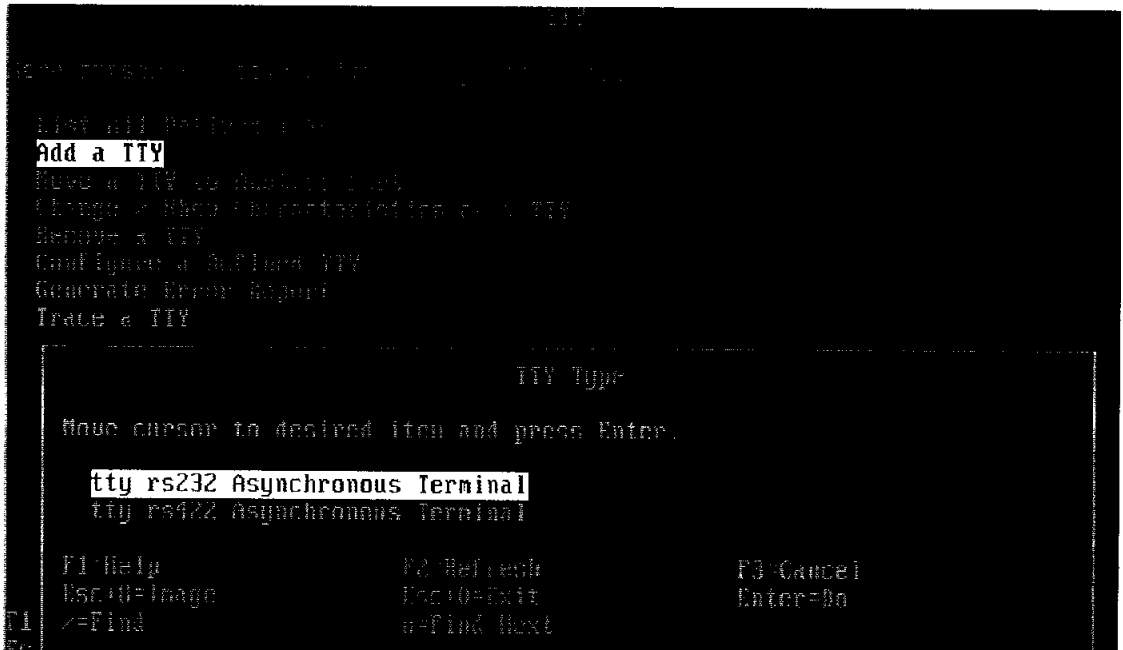

그림 5.3 rs232(tty) 추가 설정 및 삭제 화면

5.2.3 네트워크 전체에 영향을 주는 장애 극복사례

Main network control을 하는 라우터에서 비정상적으로 led가 표시되며

(정상 : 초록색, 비정상 : 주홍색, 빨간색) 네트웍 전체가 hang이 발생하는 현상일 때, 먼저 물리적인 하드웨어 문제인지(하드웨어 관련 led인지) 확인 후 하드웨어 문제가 아니라면, 방화벽에 있는 장비나 라우터의 라우팅 테이블을 확인하여야 한다. 라우팅 테이블에서 만약 비정상적인 형태를 취하고 있을 시 라우팅 테이블을 삭제하고 정상 라우팅 테이블을 만들어 준다.

참고 : 일반적인(동적) 라우팅 테이블은 문제가 발생하지 않으나, 위의 현상은 정적 라우팅 테이블의 경우를 나타낸 것이다.

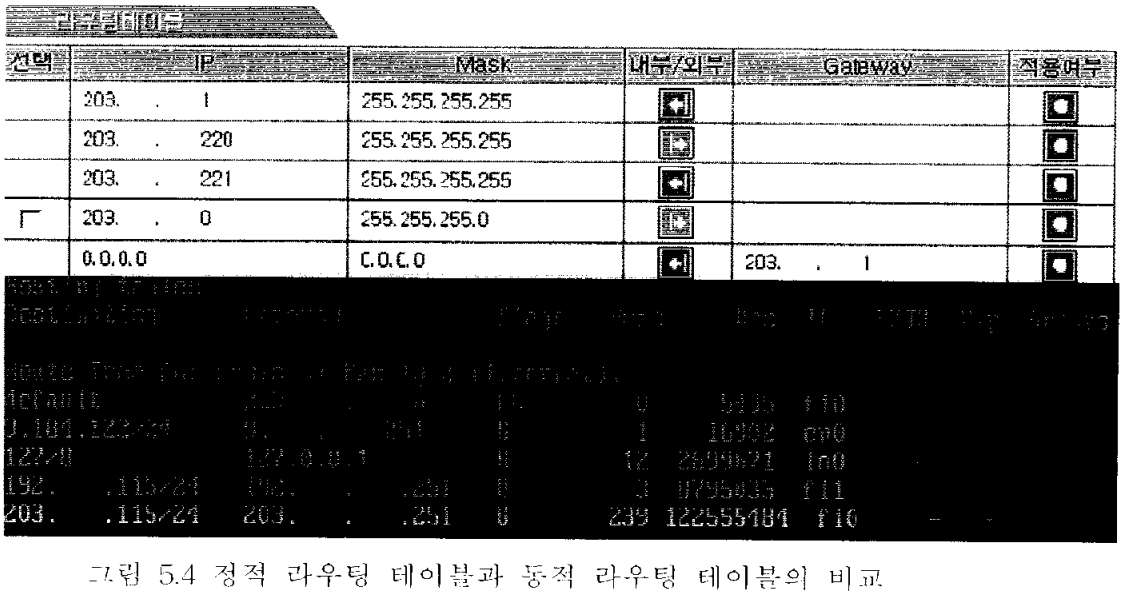

그림 5.4 정적 라우팅 테이블과 동적 라우팅 테이블의 비교

5.2.4. Well-known problem의 장애 극복을 해결하는 방법

DMS(DeadMan Switch)가 작동하는 경우

Clean takeover를 수행하기 위해, 주어진 시간 내에 DMS가 reset되지 않으면 busy 노드를 crash하고 takeover를 수행한다[4]. HACMP 클러스터 구성에서 가장 느린 네트워크에서 발생한다.

DMS Problem을 피하는 방법

: 장애발생 복구 시 시스템의 원상회복 속도 향상을 기하기 위한 조치 방법으로 동기화 file을 60초에서 10초로 조정

- enable I/O Pacing (#smitty chgsys)

  . High watermark : 30 중간값 (33)

  . Low watermark : 20 중간값 (24)

- increase the frequency of the syncd daemon (/sbin/rc.boot)

  (More frequency하게 조정할 경우, overall system performance를 고려)

  . syncd 10

- change the heartbeat rate of a NIM(Network interface module)

  (Not make it faster)

  . Heartbeat rate : slow

5.3 HACMP 비정상적인 종료에 관한 진단

5.3.1 비정상적인 종료에 관한 진단

ps ef | grep oracle

Lock이 걸려 종료되지 못한 Oracle process가 남아있다.

lsvg -l pectvg

Oracle을 Stop하였는데도 Open되어 있는 Raw logical volume이 존재한다.

tail -f /tmp/hacmp.out

HACMP를 Stop or software적으로 takeover후 위의 명령을 사용하여 스크립트를 관찰하면 Event error가 발생한다.

ps ef | grep cluster

HACMP를 Stop하여도 cluster daemon들이 종료되지 않았을 경우이다.

5.3.2 비정상적인 종료에 관한 진단log file

- Examming Message and Log File Check

  /var/adm/cluster.log

- HACMP/6000 스크립트 와 daemon에 의해 생성된 Message를 저장.

  /tmp/cm.log

- clstrmgr의 동작에 의해 생성되는 message를 저장.

  /tmp/hacmp.out

- HACMP의 스크립트에 의해 생성된 message를 저장하며 스크립트에 의해 수행된 각 명령어의 기록을 저장한다.

  /usr/sbin/cluster/history/cluster.mmdd

- HACMP의 스크립트에 의해 생성된 message를 날짜별로 관리한다.

  system error log : errpt

HACMP/6000의 스크립트와 daemon을 포함한 모든 AIX subsystem으로 부터 생성된 message를 보관한다. HACMP 구동시 위의 사항을 점검함으로써 Event error, 또는 다른 사항의 문제를 확인 할 수 있다.

5.4 OPS 구성과 문제진단 및 해결

5.4.1 OPS 구성 형태(Loosely Coupled System - Clustering)

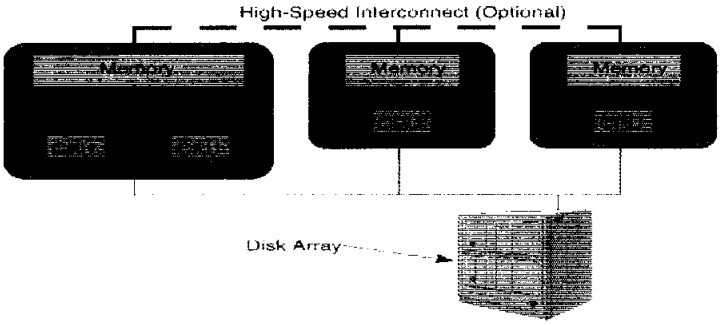

그림 5.5 Losely Coupled System - Clustering

. 각 노드는 하나 또는 그 이상의 CPU를 보유

. Memory는 노드간 비 공유

. 각 노드는 같은 Shared Disk와 Resource를 Access UNIX & VAX Cluster가 대표적임

. Oracle Parallel Server는 Loosely Coupled System을 지원하며 각 노드는 자신의 Oracle Instance를 가진다.

이러한 Instance간의 Cache Coherency를 보장하기 위한 Lock관리는 필수적이다.

5.4.2 OPS에서의 Lock 관리

OPS 환경에서의 Lock관리는 DLM이 관장하며 이는 Cluster System에서 Resource Sharing을 조정 및 관리하는 소프트웨어로써 Application들이 같은 Resource에 대해 동시에 Access를 요구할 때 서로간의 Synchronization을 유지하며 충돌이 일어나지 않도록 조정하는 기능을 담당한다.

5.4.2.1 DLM(Distributed Lock Manager)

DLM(Distributed Lock Manager)이란, OPS를 구성하는 자원(resource) 공유를 조정/관리하는 소프트웨어로써, 노드와 노드간에 발생하는 사용자들의 접근(access)를 조정하고 동기화를 유지하여 conflict가 발생하지 않도록 하는 기능을 담당한다.

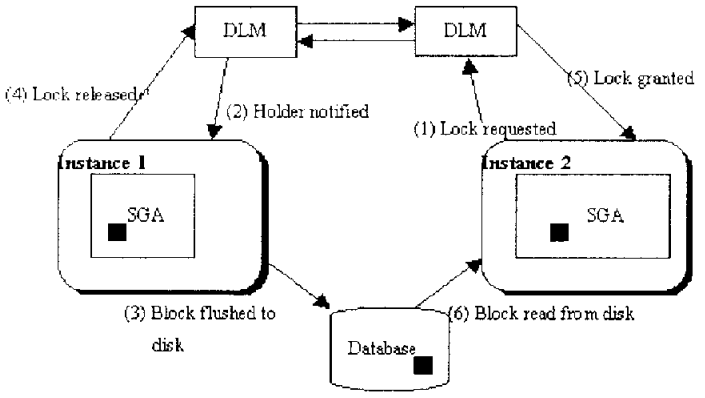

그림 5.6 DLM 구성도

DLM 의 주요 기능

· lock의 요구와 인지, Deadlock감지 기능 (lock daemon)

· 모든 노드에 걸쳐 global하게 상시적인 membership을 유지/관리

· 모든 노드의 lock processing의 이상 유무를 monitoring

· lock에 대한 memory를 할당하고, DLM client의 비정상 종료 감지 및 recovery실시 (lock monitor)

· resource 및 lock structure를 유지하여 lock mode 및 convert status관리, ownership 획득

(1) 8.0 version 이전의 DLM

Oracle 제공 : IBM SP2, Sun v 7.3, HP v 7.3

OS/vendor 제공 : Pyramid Nile cluster, DEC Tru cluster, Reliant RM1000, Sequent cluster, NCR , IBM HACMP

(2) 8.0 version 이후의 DLM

Oracle에서 각 platform에 Unix DLM 을 제공

Lock Management

OPS 에서 lock 관리의 목적은 병목 현상을 제거하고, 각 instance 의 buffer cache내에 있는dirty block의 coherency를 유지해야 한다.

(1) PCM Lock

PCM (Parallel Cache Management) Lock은 Buffer cache안에 있는 Oracle data block(data 와 index block)을 cover하는 instance lock을 의미하는데, hashed locking방식과 fine grain locking방식이 있다.

(2) Hashed Locking 방식

Version7.3 이전까지 사용된default locking방식으로 DB기동시 gc_* parameter 에 의해 할당된 lock과 Oracle datafile의 block이 static하게 대응되는 방식으로, 다른 instance에서 요구가 없으면, lock의 ownership을 계속 유지한다.

(3) Fine grain Locking 방식

block에 할당된 DLM lock이 사용 후에는 자동으로 release되는 dynamic운영 방식으로, 한정된 lock 의 효율적인 관리가 고려된 Oracle 7.3 의 새로운 방법이다.

DLM 과 Oracle Recovery 순서

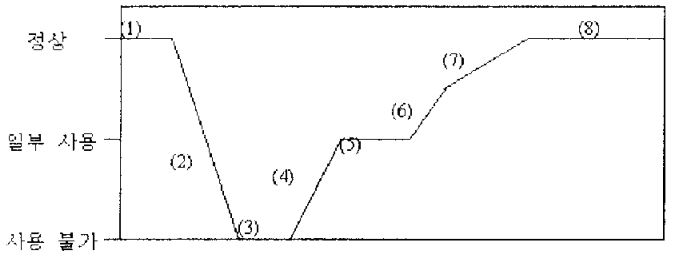

그림 5.7 DLM 과 Oracle Recovery 순서

한 instance down시는 다른 instance의 alert_SID.log를 확인한다

i. OPS 정상 운영

ii. connect manager 에 의한 한 노드 fail 인식 및 cluster member 재구성

iii. DLM Database 재구성

iv. SMON에 의한 DLM의 모든 resource check 및 instance recovery 완료까지 invalid DLM 에 대한 access 금지

v. dead thread의 redo log 내용을 DB 에 적용 (roll forward)

vi. SMON에 의한 각 block 에 대한 SCN 적용및 lock recovery 완료

vii. Rollback segment의 undo entry에 적용(rollback) 및 완료 후 각 block의 access 가능

viii. Instance Recovery 완료 및 모든 data access 가능

(4) OPS구성 시 고려 사항

· system tablespace

권장 크기 : 100 - 200M byte (단, stored procedure 와 db trigger를 많이 사용하는 경우에는 300 - 700M 정도가 적당함)

* 주의 사항 : temporary tablespace 영역으로 할당하지 말 것

· Rollback segment

갯수 : 최소한 1개 이상. 각 instance 당 1 개 또는 2 개 이상

storage : initial 과 next value 는 동일하게 줄 것. Pctincrease = 0

권장 사항 : 가능한 maxextent 를 이용하여 on-line 중 dynamic increase /decrease 를 최소화. Batch job 을 위한 전용 tablespace 및 rollback segment 유지 관리

· temporary tablespace

사용자를 업무별, instance 별로 분산 할당, Instance 별로 하나 이상 할당한다.

storage : pctincrease=0, initial 및 next value 는 sort_area_size 의 배수로 동일하게 부여

권장 : 최대 indexed data 의 2 배 이상 할당

최대 sort size 보다 크게 할당

· On-line redo log

Redo thread : instance 당 1 개

Redo log group : instance 당 3-5 개

Members/group : 최소한 각 group 당 2 개

Redo log member size : log switch가 보통 15분 정도에 한 번씩 발생되도록 부여(5 ~ 10 M byte)

group location : index 및 data file로부터 분리 저장

member location : 서로 다른 controller 및 disk 에 분리 저장

· Archive Redo log

destination : 3일 이상의 archived log file 이 저장될 수 있도록 충분히 큰 space 확보하고, free space 가 일정 수준 이하로 떨어지면 dba에게 경고

message 를 주도록 환경을 setup 한다. 가능한 1 일 이상의 archive log file 을 destination 에 보관

· Control File은 최소한 3 개 정도가 필요

MAXDATAFILES : 향후를 고려해 충분한 값을 부여 (default = 32)

MAXLOGFILES : 향후를 고려해 충분한 값을 부여 (default = 32)

MAXLOGMEMBERS : 최소한 3 이상

MAXINSTANCE : OPS구성 시 향후 추가할 노드를 고려해 충분한 값을 부여

5.4.3 OPS 관련 문제진단 및 해결

양 노드가 동시에 같은 table을 handling할 때 일어나는 현상 즉, 서비스 시스템의 처리속도가 현저히 저하된다. 즉 먼저 잡은 process가 처리를 끝내고 resource를 반납할 때까지 양 노드가 공히 성능이 떨어지는 현상이 발생한다. 현재까지 ops 환경에서 이런 문제를 해결하기가 곤란하여 양 노드를 별개의 업무를 할 수 있도록 권장하고 있다.

A사 현황

노드 1 : 회계, 기술지원, 총무, 인사, 공통, 그룹웨어, 시스템개발 Test

노드 2 : 고객, 검용, 투자개발, 기획, 품질, 상거래지원, 상거래사업지원, EDI, Project Management

B사 현황

노드 1 : 하역계획, 운영지원, 외부운영서비스, MIS통계, 시스템개발 Test

노드 2 : 인사, 총무, 급여, 회계, 장비, 정비, Web Service 조회전용, 구매 및 자산지원

노드 3 : EDI

OPS 환경에서 Data Base관련 일반적 장애 극복 조치 방법은 OPS 환경에서 어느 한쪽 노드(B 노드)에서 문제가 발생하면 문제 발생 후 다시 HACMP를 UP시킬 때는 반드시 다른 한쪽 노드(A 노드)을 clear 시킨 후 문제가 발생한 노드(B 노드)부터 UP시키도록 하여야만 합니다.

clear작업이 정확히 이루어지지 않고 A 노드를 먼저 UP시킬 때 HACMP에서 B 노드에서 A 노드로부터 문제가 있었다고 판단하여(즉, heatbet 신호가 없었다) UP 도중에 B 노드로 takeover가 발생하는 현상을 일으킨다.

그래서 정확히 clear작업이 이루어졌더라도 정확한 판단이 되지 않을 시는 항상 사고 당시 노드(B 노드)로부터 HACMP를 먼저 UP시켜야만 한다.

OPS환경에서의 장애 극복을 위한 Shell 프로그램

지금 까지 고 가용성 소프트웨어 상에서의 장애와 OPS 추가 시 장애를 분야별로 극복 사례를 정리하였다. 현재의 분산 환경구조에서 완전한 자동화 극복은 힘들겠지만, 사용자에게 보다 빠르고 신속한 서비스를 위해 각 단계별 스크립트를 연결하는 Shell 프로그램을 제시한다.

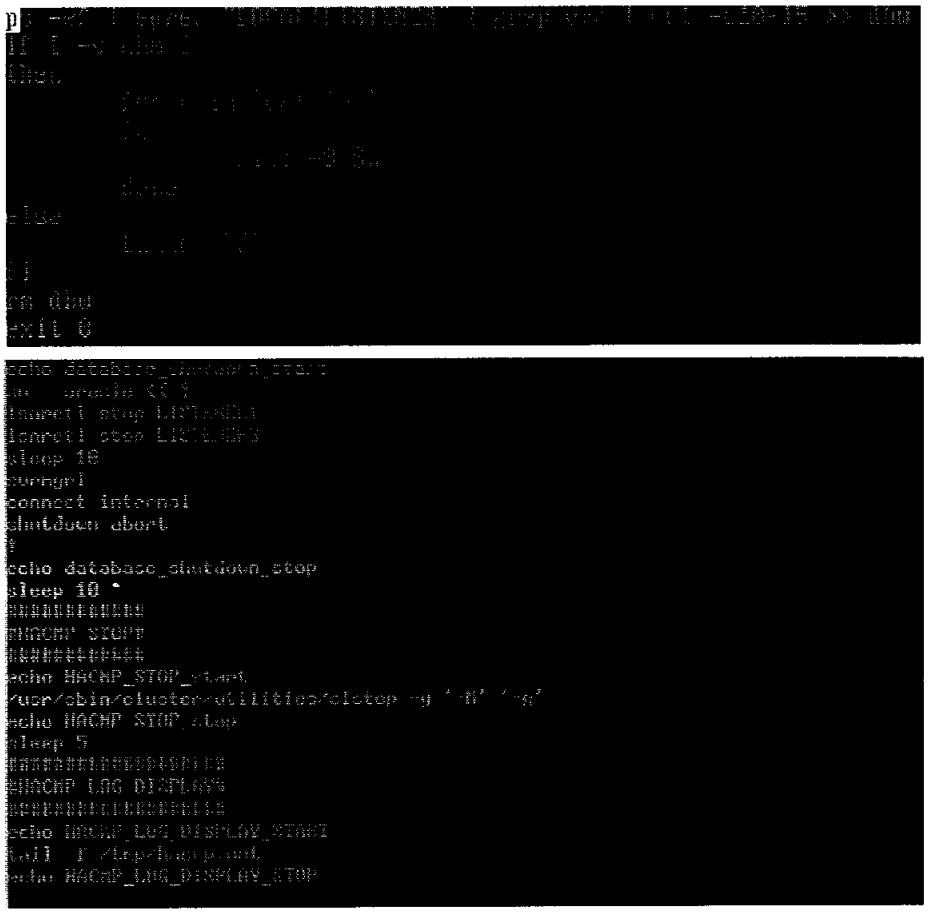

그림 5.8 OPS 환경에서의 장애극복 프로그램

제 6 장 결 론

전 세계적으로 하역 물 동량이 늘어나면서 전산 시스템의 확장이 필수화되고 무 정지 운영이 보편화되면서 클러스터링을 이용한 고 가용성 시스템을 사용하게 되었다.

이러한 필요성에 따라 본 논문에서는 P사의 신 시스템 구축 시 사용되었던 구축형태와 현 구조, 그리고 문제점을 살펴보았다. 개방된 운영체제와 확장성이 강한 시스템을 통해 일반적으로 널리 알려진 RDBMS를 적용 Real Service를 원활히 하도록 구성하였다. 그러나 본 논문에서 살펴보았듯이 아직 OPS 환경에서의 고 가용성의 문제점이 남아있다. 그래서 사용자(Client)에게 조금 더 신속하고 빠른 서비스를 위해 기존 매뉴얼 방식에서 자동화 방식을 추구한 스크립트 언결 Shell 프로그램을 제시하였다.

그리고 계속적으로 늘어나는 분산 시스템을 현재의 고 가용성 기술로 처리해 나아 가야할지 계속적인 연구가 필요하며, 최악의 순간 즉, 양 노드가 fail이 났을 때 분산된 한 시스템으로 현(fail) 시점에서 당장 필요한 작업을 가능하도록 처리할 방법을 생각하는 것도 중요한 과제이다.

참고문헌

[1] HACMP for AIX, Version 4.3.1: Concepts and Facilities, order number SC23-4276-01

[2] HACMP for AIX, Version 4.3.1: Planning Guide, order number SC23-4277-01

[3] HACMP for AIX, Version 4.3.1: Installation Guide, order number SC23-4278-01

[4] HACMP for AIX, Version 4.3.1: Troubleshooting Guide, order number SC23-4280-01

[5] HACMP for AIX, Version 4.3.1: Programming Locking Applications, order number SC23-4281-01

[6] HACMP for AIX, Version 4.3.1: Programming Client Applications, order number SC23-4282-01

[7] HACMP for AIX, Version 4.3.1: Master Index and Glossary, order number SC23-4285-01

[8] HACMP for AIX, Version 4.3.1: HANFS for AIX Installation and Administration Guide, order number SC23-4283-01

[9] HACMP for AIX, Version 4.3.1: Enhanced Scalability Installation and Administration Guide,Vol.1, order number SC23-4284-01

[10] HACMP for AIX, Version 4.3.1: Enhanced Scalability Installation and Administration Guide,Vol.2, order number SC23-4306-00.

[11] IBM International Program License Agreement, order number S29H-1286.

[12] http://www.rs6000.ibm.com/aix/library

감사의 글

오늘 이 글을 남길 수 있게 도와주신 모든 분들께 깊은 감사의 뜻을 전합니다. 지난 2년의 대학원 생활동안 지속적인 보살핌과 배려 그리고 성심성의껏 지도 편달해 주신 지도교수이신 권오흠 교수님께 진심으로 감사를 드립니다. 교수님께서 보여주신 공학인으로서의 연구자세와 열성을 다하여 가르치시는 모습은 앞으로 저의 인생에 큰 가르침이 되리라고 믿습니다.

부족한 저의 논문을 심사하시면서 세심한 지도와 많은 관심을 보여주신 정목동 교수님, 서경룡 교수님께도 깊은 고마움을 표합니다.

대학원 과정동안 가르침을 주신 우종호 교수님, 정목동 교수님, 조우현 교수님, 서경룡 교수님, 권오흠 교수님, 신봉기 교수님들에게서 저는 많은 도움을 받았습니다.

대학원 생활을 순탄히 끝낼 수 있도록 도와주신 직장동료 여러분에게 깊은 감사를 드리며, 지난 1년 동안 한국 IBM의 이주열 차장님과 한국 오라클의 배종진 과장님 그리고 알고리즘 연구실의 원형권씨는 부족한 저에게 이 논문을 일일이 읽고 내용 보충과 격려를 해 주었으며, 김종욱씨와 김은희씨는 뒤에서 많은 지원을 하며 도움을 주었습니다. 그리고 박창수 선배님의 사려 깊은 격려에 고개 숙여 감사드립니다. 연구실의 후배 백승찬, 오윤선의 도움도 컸습니다. 저를 지금까지 키워주시고 보살펴주신 부모님께 감사의 말을 전합니다. 마지막으로 지금까지 믿음과 사랑으로 힘이 되어준 사랑하는 아내 박선희와 항상 바쁘다는 이유로 많은 시간을 같이 있어주지 못한 아들 준원이 에게 미안하다는 말을 전함과 동시에 깊은 고마움을 전합니다.

2002 년 12 월

서 진 석 올림

부 록

Event 발생시 추가 응용 스크립트 file

1) /usr/sbin/cluster/app/opsA_start

```
#! /bin/ksh
su - oracle << !
lsnrctl start LISTENER1
svrmgrl
connect internal
startup parallel
!
exit 0
```

2) /usr/sbin/cluster/app/opsA_stop

```
#! /bin/ksh
su - oracle << !
lsnrctl stop LISTENER1
sleep 10
svrmgrl
connect internal
shutdown abort
!
ps -ef | egrep "LOCAL | LISTENER" | grep ora
| cut -c10-15 >> sjs
```

```
if [ -s sjs ]
then
        for n in `cat sjs`
        do
                kill -9 $n
        done
else
        banner  "OK"
fi
rm sjs
exit 0
```

Application Server Name : opsB

Application Start Script : /usr/sbin/cluster/app/opsB_start

Application Stop Script : /usr/sbin/cluster/app/opsB_stop

3) usr/sbin/cluster/app/opsB_start

```
#! /bin/ksh
su   oracle << !
lsnrctl start LISTENER2
svrmgrl
connect internal
startup parallel
!
exit 0
```

4) /usr/sbin/cluster/app/opsB_stop

```
#! /bin/ksh
su - oracle << !
lsnrctl stop LISTENER2
sleep 10
svrmgrl
connect internal
shutdown abort
!
ps -ef | egrep "LOCAL | LISTENER" | grep ora
| cut -c10-15 >> sjs
if [ -s sjs ]
then
        for n in `cat sjs`
        do
                kill -9 $n
        done
else
        banner "OK"
fi
rm sjs
exit 0
```

5) /usr/sbin/cluster/app/p10_start

```
rc.netware
```

```
cnsview -c "daemon start"
sleep 10
cnsview -c "explore"
in=`cnsview -c "explore" | wc -l | cut -c8`
echo $in
while [[ $in -le 2 ]]
do
    in=`cnsview -c "explore" | wc -l | cut -c8`
    echo $in
done
    lsdev -Cc tty
    cnsview -c "explore"
    print "current line is $in"
    /usr/sbin/cluster/app/ttyenable
    banner OK
```

6) /usr/sbin/cluster/app/p10_stop

```
/usr/sbin/cluster/app/ttydisa
cnsview -c "daemon stop"
/usr/lpp/netware/bin/stopnps
lsdev -Cc tty
```

7) /usr/sbin/cluster/app/ttyenable

```
penable 'tty2'
penable 'tty3'
penable 'tty4'
```

.

.

.

penable 'tty33'

8) /usr/sbin/cluster/app/ttydisa

pdisable 'tty2'

pdisable 'tty3'

pdisable 'tty4'

.

.

.

.

pdisable 'tty33'

9) /usr/sbin/cluster/app/rc.p10

```
ha=`ps -ef | grep cluster | wc -l`
if [[ $ha -eq 2]]
then
    rc=`ps -ef | grep rc.ping | wc -l`
    if [[ $rc -eq 1 ]]
    then
        print "7318 ethernet adapter checking start!" >
/dev/console
        nohub /usr/sbin/cluster/app/rc.ping &
    else
```

```
                d1=`ps  ef | grep rc.ping | grep sh | wc -l`
                    if [[ $rc -eq 2 ]]
                    then
                        d2=`ps  ef | grep rc.ping | grep sh | cut
c10 15`
                        kill -9 $d2
                     else
                         exit 0
                     fi
               fi
        else
               k=`ps -ef | grep rc.ping | wc -l`
               if [[ $k -ge 2 ]]
               then
                   kil=`ps -ef | grep rc.ping | grep sh | cut -c10-15
                   kill -9 $kil
                else
                   exit 0
                fi
          fi
   10) /usr/sbin/cluster/app/rc.ping
       a=0
       while [[ $a -eq 0 ]]
       do
          ping -c1 9.xxx.xxx.147 > /dev/null 2 > &1
```

```
    a=`echo $?`
done
echo "7318 Ethernet adapter Fail !!" > /dev/console
usr/sbin/cluster/utilities/clstop -y '-N' '-s' ' gr'
sleep 300
exit 0
```

11) /usr/sbin/cluster/app/cl_route (pectosA_node)

```
route add 203.xxx.xxx.6 -interface 203.xxx.xxx.252
arp -d 203.xxx.xxx.6
/etc/ping -c1 203.xxx.xxx.6
route delete 203.xxx.xxx.6 203.xxx.xxx.252
```

12) /usr/sbin/cluster/app/cl_route (pectosB_node)

```
route add 203.xxx.xxx.6 -interface 203.xxx.xxx.251
arp -d 203.xxx.xxx.6
/etc/ping -c1 203.xxx.xxx.6
route delete 203.xxx.xxx.6 203.xxx.xxx.251
```