HACMP 安装配置，管理与诊断分析

HACMP 工作原理

HACMP 的工作原理是利用 LAN 来监控主机及网络、网卡的状态。在一个 HACMP 环境中有 TCP/IP 网络和非 TCP/IP 网络。TCP/IP 网络即应用客户端访问的公共网，该网可以是大多数 AIX 所支持的网络，如 Ethernet，T.R.，FDDI，ATM，SOCC，SLIP，等等。 非 TCP/IP 网络用来为 HACMP 对 HA 环境（Cluster）中的各节点进行监控而提供的一个替代 TCP/IP 的通讯路径，它可以是用 RS232 串口线将各节点连接起来，也可以是将各节点的 SCSI 卡或 SSA 卡设置成 Target Mode 方式。

安装 HACMP 软件

主机	屏幕显示	键盘操作
xinzhuang1 xinzhuang2	将含有 HACMP 软件的光盘插入光驱	
	root：/>	smitty install_latest
	INPUT device /directory for	/dev/cd0
	Software SOFTWARE to install	All
	PREVIEW only?	no <Enter>
	安装完成后提示 OK	
	root：/>	shutdown –Fr <Enter> 关机重启

HACMP 主要进程

Cluster Manager daemon(/usr/sbin/cluster/clstrmgr)：是 HACMP 的核心进程，运行于每个群集节点，监视群集目标，如节点、网络接口、网络等的变化，生成事件，激活相应的事件脚本程序 script 做处理。

Cluster SMUX Peer daemon(/usr/sbin/cluster/clsmuxpd)：也运行于每个群集节点，通过 clinfo 对客户端应用提供 SNMP 支持，该守护进程维护管理信息库 MIB(Management Information Base)中的群集状态，可通过标准的 SNMP 函数访问这些数据，注意同时要运行 snmpd。

Cluster Information Program daemon(/usr/sbin/cluster/clinfo)：可运行于任一群集节点或 RS6000 客户机，通过查询 SMUX Peer，来维护群集状态拓扑图，并为客户端提供应用程序接口 API 来访问这些数据。

Cluster Lock Manager daemon（/usr/sbin/cluster/cllockd）：群集配置成共同访问的方式，

需要该守护进程运行于每个群集节点，通过 API 为应用访问共享磁盘提供加锁功能，避免冲突。

HACMP 主要术语（在课堂上逐一解释，画一张图）

Cluster

Node

Network

Topology

Resource

Resource group

Application server

Reintegration

Synchronization

Event

HACMP 主要事件

见/usr/sbin/cluster/events 系统自带的事件脚本

/usr/sbin/cluster/events/custom 华为自己编写的脚本

HACMP 主要日志

```
cluster.log      - Generated by cluster scripts and daemons
cluster.mmdd     - Cluster history files generated daily
 cspoc.log        - Generated by CSPOC commands
 cm.log           - Generated by the clstrmgr daemon
 dms_loads.out    - Generated by deadman's switch activity
 emuhacmp.out     - Generated by the event emulator scripts
 hacmp.out        - Generated by event scripts and utilities
```

HACMP 资源接管的三种方式

Resource Group 的三种节点关系
(Node Relationship):
(1) Cascading
(2) Concurrent
(3) Rotating

安装配置 HACMP

安装配置 HACMP 时，应保证卷组 datavg 不是激活状态；

参数规划

主机参数

	生产机 xinzhuang1	备份机 xinzhuang2
主机名	xinzhuang1	xinzhuang2
网络掩码	255．255．255．0	
网卡 IP 地址		
en0	xinzhuang1_svc 46.147.1.100 xinzhuang1_boot 46.147.1.102	xinzhuang2_svc 46.147.1.101 xinzhuang2_boot 46.147.1.103
en1	xinzhuang1_stb 46.147.2.100	xinzhuang2_stb 46.147.2.101

集群（HACMP）参数

	生产机	备份机
Cluster ID	1	
Cluster Name	xinzhuang_cluster	
Node Name	xinzhuang1	xinzhuang2
Adapters Service Boot StandBy Serial	 xinzhuang1_svc xinzhuang1_boot xinzhuang1_stb xinzhuang1_tty	 xinzhuang2_svc xinzhuang2_boot xinzhuang2_stb xinzhuang2_tty
Application Server Start Script Stop Script	btpdb2_srv /etc/btpdb2_start /etc/btpdb2_stop	
Resource Group		
Group Name Cluster Mode Cluster Node Service Adapter Share VG	xinzhuang1_gr cascading xinzhuang1 xinzhuang2 xinzhuang1_svc datavg	xinzhuang2_gr cascading xinzhuang2 xinzhuang1 xinzhuang2_svc
Filesystems	/btp /btpdbs /btpmsg /btplog /btprpt /btpslg /db2catalog /db2log /db2tmp	
Application Server	btpdb2_srv	

各主机/etc/hosts 文件及/.rhosts 文件内容

生产机(xinzhuang1)

/etc/hosts		/.rhosts
127.0.0.1	loopback localhost xinzhuang1	xinzhuang1_svc
46.147.1.100	xinzhuang1_svc	xinzhuang1_boot
46.147.1.102	xinzhuang1_boot	xinzhuang1_stb
46.147.2.100	xinzhuang1_stb	xinzhuang1_tty
		xinzhuang2_svc
46.147.1.101	xinzhuang2_svc	xinzhuang2_boot
46.147.1.103	xinzhuang2_boot	xinzhuang2_stb
46.147.2.101	xinzhuang2_stb	xinzhuang2_tty
备份机(xinzhuang2)		
/etc/hosts		/.rhosts
127.0.0.1	loopback localhost xinzhuang2	xinzhuang1_svc
46.147.1.100	xinzhuang1_svc	xinzhuang1_boot
46.147.1.102	xinzhuang1_boot	xinzhuang1_stb
46.147.2.100	xinzhuang1_stb	xinzhuang1_tty
		xinzhuang2_svc
46.147.1.101	xinzhuang2_svc	xinzhuang2_boot
46.147.1.103	xinzhuang2_boot	xinzhuang2_stb
46.147.2.101	xinzhuang2_stb	xinzhuang2_tty

配置网络及主机名

屏幕显示		键盘操作	
root：/>		smitty mktcpip <Enter>	
将各网卡设置成如下所示（参见“参数设定”）			
Hostname	IP Address	Network Mask	Network Interface
xinzhuang1_svc	46.147.1.100	255.255.255.0	en0
xinzhuang1_stb	46.147.2.100	255.255.255.0	en1
xinzhuang2_svc	46.147.1.101	255.255.255.0	en0
xinzhuang2_stb	46.147.2.101	255.255.255.0	en1
root：/>		smitty chinet <Enter>	
将 en0 的 IP 地址改为对应的 xinzhuang1_boot 地址（46.147.1.102） 将 en0 的 IP 地址改为对应的 xinzhuang2_boot 地址（46.147.1.103）			

编辑各主机的/etc/hosts 及/.rhosts 文件

配置心跳信号线：

不同的机型，支持心跳线的串口也是不同的：F50、H50、H70 只能使用内置串口的最后一个口（即 S3）；F80、H80、M80、M85 只能使用内置串口的最后一个口（即 S4）；S70、S7A、S80、S85 不能使用内置串口做为心跳线的接口，只能用外接的多用户卡上（一般配置 8 口多用户卡）的串口。

主机	屏幕显示	键盘操作
xinzhuang1	root：/>	smitty mktty <Enter><Enter><Enter>
	Parent Adapter	sa1
	Port number	0
	Enable LOGIN	<disable><Enter>
xinzhuang2	root：/>	smitty mktty <Enter><Enter><Enter>

	Parent Adapter	sa1
	Port number	0
	Enable LOGIN	<disable><Enter>
检查配置		
xinzhuang1	root：/>	stty < /dev/tty1
	命令进入等待状态	
xinzhuang2	root：/>	stty < /dev/tty1
两主机都显示终端设置信息，且返回到命令提示符		

配置 HACMP 集群

本节所有操作仅在集群的一台机器上执行，这里建议在生产机上操作。

1.1.1 定义集群

smitty hacmp

→Cluster Configuration

→Cluster Topology

→Configure Cluster

→Add a Cluster Definition

Cluster ID	1
Cluster Name	xinzhuang_cluster

1.1.2 定义节点

smitty hacmp

→Cluster Configuration

→Cluster Topology

→Configure Nodes

→Add Cluster Nodes

Node names	xinzhuang1 xinzhuang2

1.1.3 定义网络

smitty hacmp

→Cluster Configuration

→Cluster Topology

→Configure Networks

→Add a Network

分别选择“ IP-based Network ”和“ Non IP-based Network ”对网络进行配置。在配置“ IP-based Network ”时，按 F7 键同时选择网段“ 46.147.1.0/24 和 46.147.2.0/24 ”。

Network Type	Network Name	Network Type	Subnet(s)

IP-based Network	network1	ether	46.147.1.0/24 46.147.2.0/24
Non IP-based Network	serialnet	rs232	

1.1.4 定义网卡

smitty hacmp

→Cluster Configuration

→Cluster Topology

→Configure Adapters

→Adapters on IP-based network

Adapter IP Label	Network Type	Network Name	Network Attribute	Adapter Function	Adapter Identifier	Node Name
xinzhuang1_svc	ether	network1	public	service	46.147.1.100	xinzhuang1
xinzhuang1_stb	ether	network1	public	standby	46.147.2.100	xinzhuang1
xinzhuang1_boot	ether	network1	public	boot	46.147.1.102	xinzhuang1
xinzhuang1_tty	rs232	serialnet	serial	service	/dev/tty1	xinzhuang1
xinzhuang2_svc	ether	network1	public	service	46.147.1.101	xinzhuang2
xinzhuang2_stb	ether	network1	public	standby	46.147.2.101	xinzhuang2
xinzhuang2_boot	ether	network1	public	boot	46.147.1.103	xinzhuang2
xinzhuang2_tty	rs232	serialnet	serial	service	/dev/tty1	xinzhuang2

1.1.5 定义资源组

- # smitty hacmp

→Cluster Configuration

→Cluster Resources

→Define Application Servers

→ Add an Application Server

Server Name	btpdb2_srv
Start Script	/etc/btpdb2_start
Stop Script	/etc/btpdb2_stop

注意：请将两脚本生成，内容可为空，但要确保 root 的 x 权限。

- # smitty hacmp

→Cluster Configuration

→Cluster Resources

→Define Resource Groups

→Add a Resource Group

Resources Group Name	xinzhuang1_gr	xinzhuang2_gr
Node Relationship	cascading	cascading
Participating Node Names	xinzhuang1 xinzhuang2	xinzhuang2 xinzhuang1

- # smitty hacmp
  →Cluster Configuration
  →Cluster Resources
  →Change/Show Resources/Attributes for a Resource Group

Resources Group Name	xinzhuang1_gr	xinzhuang2_gr
Node Relationship	Cascading	Cascading
Participating Node Names	xinzhuang1 xinzhuang2	xinzhuang2 xinzhuang1
Service IP Label	xinzhuang1_svc	xinzhuang2_svc
Volume Groups	datavg	
File System	/btp /btpdbs /btpmsg /btplog /btprpt /btpslg /db2catalog /db2log /db2tmp	
Application Servers	btpdb2_srv	

1.1.6 同步 HACMP 集群定义

- 同步集群拓扑

  # smitty hacmp
  →Cluster Configuration
  →Cluster Topology
  →Synchronize Cluster Topology

最好先 Emulate，确认无问题时再 Actual

- 同步集群资源

  # smitty hacmp
  →Cluster Configuration
  →Cluster Resources
  →Synchronize Cluster Resource

最好先 Emulate，确认无问题时再 Actual

上述两个步骤，只有当运行状态为 OK 时，才对。

1.1.7 校验 HACMP 集群定义

smitty hacmp
→Cluster Configuration
→Cluster Verification

→Verify Cluster

上述步骤，只有当运行状态为 OK，且无报错时，才对。

1.1.8 HACMP 性能调整

生产机、备份机应分别确认。

- # smitty hacmp

  →Cluster Configuration

  →Advanced Performance Tuning Parameters

  →Change/Show I/O pacing

HIGH water mark for pending write I/Os per file	33
LOW water mark for pending write I/Os per file	24

- # smitty hacmp

  →Cluster Configuration

  →Advanced Performance Tuning Parameters

  →Change/Show syncd frequency

Syncd frequency (in seconds)	10

- # smitty hacmp

  →Cluster Configuration

  →Advanced Performance Tuning Parameters

  →Change/Show a Network Module

  →-rs232 和 ethernet

Failure Detection Rate	Slow

1.2 HACMP/DB2/BTP 的启动和停止脚本

在两台机器上，编辑启动和停止 HACMP/DB2/BTP 的脚本，并修改脚本权限为 700，文件均放在/etc 目录下。**脚本内容参见附录一。**

/etc/ha_start、/etc/ha_stop、/etc/ha_stat、/etc/ha_list、/etc/btpdb2_start、/etc/btpdb2_stop

测试 HACMP

检查心跳线是否已经配置好：

1、在两台上分别输入如下命令：

```
S85_1# cat /etc/hosts >/dev/tty0
S85_2# cat </dev/tty0
```

如果在 S85_2 机能接收到信息，则表明心跳线已经配置好

2、测试心跳线是否可以正常工作：

在 A 机上：

```
# stty </dev/tty1
```

在 B 机上：

```
# stty </dev/tty1
```

如果在两台主机上均出现了该 TTY 的配置信息，则说明心跳线设置正确。

3、如果用 7133 做心跳，先安装软件：device.ssa.*,具体是那个背不出了，好像是 target 或者 tmssa 之类，在系统盘上有。

2，chdev -l ssar -a node_number=1

3，cfgmgr -v

4，lsdec -Cc tmssa 看有没有 tmssa1.tm 和 tmssa1.im

5，测试心跳 stty < /dev/tty0 只是测试，和实际配置无关 3124 3125

6，在定义心跳时，选 ssa target 方式 ，用/dev/tmssa1 做心跳设备。

HACMP 的测试：

HACMP 将诊测并响应于三种类型的故障：1 网卡故障，2 网络工作，3 节点故障。下面就这三种故障分别进行介绍。

1、网卡故障

前面讲到，HACMP 的群集结构中，除了 TCP/IP 网络以外，还有一个非 TCP/IP 网络，它实际上是一根“ 心跳 ”线，专门用来诊测是节点死机还是仅仅网络发生故障。如下图所示，一旦节点加入了 Cluster(即该节点上的 HACMP 已正常启动)，该节点的各个网卡、非 TCP/IP 网络就会不断地接收并送 Keep-Alive 信号，K-A 的参数是可调的，HA 在连续发送一定数量个包都丢失后就可确认对方网卡，或网络，或节点发生故障。因此，有了 K-A 后，HACMP 可以很轻易地发现网卡故障，因为一旦某块网卡发生故障发往该块网卡的 K-A 就会丢失。此时 node 1 上的 cluster manager(HACMP 的“ 大脑 ”)会产生一个 swap-adapter 的事件，并执行该事件的 script(HACMP 中提供了大部分通用环境下的事件 scripts,它们是用标准 AIX 命令和 HACMP 工具来写的)。每个节点上都有至少两块网卡，一块是 service adapter,提供对外服务，另一

块是 standby adapter，它的存在只有 cluster manager 知道，应用和 client 并不知道。一旦发生 swap-adapter 事件后，cluster manager 将原来 service adapter 的 IP 地址转移到 standby adapter 上，而 standby 地址转移到故障网卡上，同时网络上其他节点进行 ARP 的刷新。网卡互换(swap-adapter)在几秒内就可完成，以太网为 3 秒，并且这种转换对应用和 client 来说是透明的，只发生延迟但连接并不中断。

1、网卡故障：

网络接口故障：

用命令：# ps –ef | grep cluster，确认所有节点上的 HACMP 已启动。

用命令：# errclear 0，清空系统错误日志。

用命令：# tail –f /tmp/hacmp.out，监控 HACMP 的运行状态。

用命令：# ifconfig en0 down，宕掉 Service 网卡。

用命令：# netstat –in，查看 Standby 网卡是否接管了宕掉的 Service 网卡的 IP 地址和 MAC 地址。

用命令：# ifconfig en1 down，宕掉接管了 Service 网卡 IP 地址和 MAC 地址后的 Standby 网卡。

用命令：# netstat –in，查看 Service 网卡是否将 IP 地址和 MAC 地址接管回来。

2、网络故障

如果发往 node1 上的 service 和 standby 网卡上的 K-A 包全都丢失，而非 TCP/IP 网络上的 K-A 仍然存在，那么 HACMP 判断 node1 仍然正常而网络发生故障。此时 HACMP 执行一个 network_down 事件。

网卡连接电缆故障：

用命令：# ps –ef | grep cluster，确认所有节点上的 HACMP 已启动。

用命令：# errclear 0，清空系统错误日志。

用命令：# tail –f /tmp/hacmp.out，监控 HACMP 的运行状态。

断开与 Service 网卡连接的网线。

用命令：# netstat –in，查看 Standby 网卡是否接管了 Service 网卡的 IP 地址和

MAC 地址。

重新连接上与原 Service 网卡连接的网线。

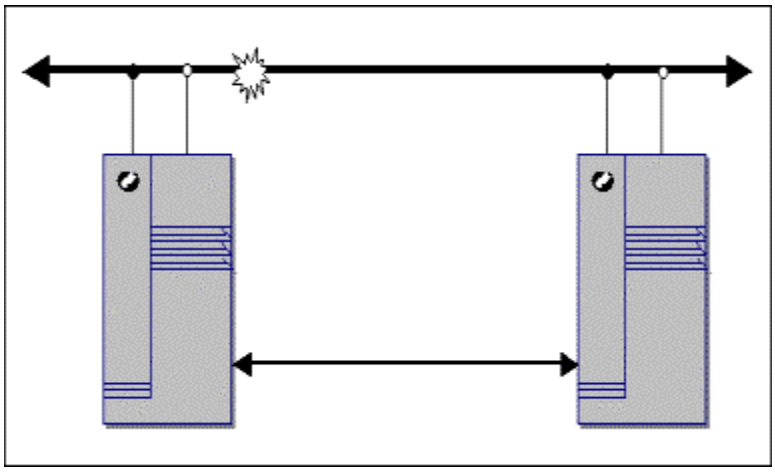
用命令：# netstat –in，查看此时原 Service 网卡的 IP 地址和 MAC 地址是否为原 Standby 网卡的 IP 地址和 Service 地址。

断开与原 Standby 网卡连接的网线。

用命令：# netstat –in，查看 Service 网卡的 IP 地址和 MAC 地址是否恢复为原来的 Service 网卡的 IP 地址和 MAC 地址。

重新连接上与 Standby 网卡连接的网线。

用命令：# netstat –in，查看 Standby 网卡的 IP 地址和 MAC 地址是否恢复为原来的 Standby 网卡的 IP 地址和 MAC 地址。



3 、节点故障

如果不仅 TCP/IP 网络上的 K-A 全部丢失，而且非 TCP/IP 网络上的 K-A 也丢失，那么 HACMP 断定该节点发生故障，并产生 node-down 事件。此时将有资源接管，即放在共享磁盘陈列上的资源将由备份节点接管，接管包括一系列操作：Acquire disks,Varyon VG, Mount file systems,Export NFS file systems, Assume IP network Address, Restart highly available applications,其中 IP 地址接管和重新启动应用由 HACMP 来实现，而其他是由 AIX 来完成。

当整个节点发生故障时，HACMP 将故障节点的 service IP address 转移到备份节点上，使网络上的 client 仍然使用这个 IP 地址，这个过程称为 IP 地址接管(IPAT)。

当一个节点 down 掉后，如果设置了 IP 地址接管，网络上的 clients 会自动连接到接管节点上；同样，如果设置了应用接管，该应用会在接管节点上自动重启，从而使系统能继续对外服务。对于要实现接管的应用，只需在 HACMP 中把它们设置成 application server,并告诉 HACMP 启动这个应用的 start script 的全路径名和停止该应用的 stop script 的全路径名。由此可见，应用接管的配置在 HACMP 中十分简单，重要的是 start script 和 stop script 的写作，这需要用户对自己应用的了解。

节点故障：
模拟*作系统崩溃：
用命令：# ps –ef | grep cluster，确认所有节点上的 HACMP 已启动。
用命令：# errclear 0，清空系统错误日志。
用命令：# tail –f /tmp/hacmp.out，监控 HACMP 的运行状态。
用命令：# cat /etc/hosts > /dev/kmem，模拟*作系统崩溃状态。
用命令：# netstat –in、# lsvg –o、# ps –ef APP_PID，查看备份节点是否接管了故障节点的 Service 地址、共享卷组和应用程序。
重新启动故障节点，并启动 HACMP。
用命令：# netstat –in、# lsvg –o、# ps –ef APP_PID，查看该节点是否将原属于他的 Service 地址、共享卷组和应用程序接管回来了。
模拟 CPU 故障：
用命令：# ps –ef | grep cluster，确认所有节点上的 HACMP 已启动。
用命令：# errclear 0，清空系统错误日志。
用命令：# tail –f /tmp/hacmp.out，监控 HACMP 的运行状态。
直接断开某个节点的电源，模拟 CPU 故障。
用命令：# netstat –in、# lsvg –o、# ps –ef APP_PID，查看备份节点是否接管了故障节点的 Service 地址、共享卷组和应用程序。
重新启动故障节点，并启动 HACMP。
用命令：# netstat –in、# lsvg –o、# ps –ef APP_PID，查看该节点是否将原属于他的 Service 地址、共享卷组和应用程序接管回来了。

4、其他故障

HACMP 只去检测网卡、网络和节点是否发生故障，并作出相应的转移、接管行为。对于其他故障，那么 HACMP 缺省不作任何动作。

a.硬盘故障

一般我们都将硬盘设置成 RAID-5 方式或 mirror 方式，从而提供硬盘的高可用性。RAID-5 将奇偶较验位分散在硬盘组中，因此当一组内的一个硬盘坏掉，组内的其他硬盘可以通过奇偶较验位将该硬盘上的数据恢复出来。RAID-5 方式一般是由硬件实现的，如下 7133 的 SSA 适配器,而且如果同一组内的两个硬盘坏掉，该组硬盘的数据很可能就会全部丢失。mirror 方式是将同一个数据写到至少两个物理外置上，因此它的效率没有 RAID-5 好，而且用盘量大，但安全性比 RAID-5 高，而且它易于实现，通过 AIX 中的(Logic Volume Management)可以很方便地设置。

b.硬盘控制卡

存储设备连接到主机上都必须通过一块控制卡，SCSI 设备是 SCSI Adapter,SSA 设备是 SSA Adapter,如果这块卡坏掉，与之连接的外设就无法利用。有几种办法可以解决这个问题。

一种办法是用多个 adapter。每个主机上都有两块或两块以上 adapter,分别连接 mirror 的数据，因此无论是硬盘坏掉，还是 Adapter 坏掉，所有好数据还是可以被主机利用，不会出现单点故障。这种方法实现起来并不难，但必须配置多块 adapter,而且必须采用数据 mirror 方式。这种方法也不用通过 HACMP 来实现。

另一种方法仍只用一块 adapter,利用 HACMP 中的 Error Notification Facility(错误通告机制)来解决。

Error Notification Facility 是 HACMP 提供的对其他设备的监控工具，任何报告给 AIX 的错误(error)都能被捕获被采取相应措施。HACMP 提供了 smit 界面，使配置简单化。

我们已知道，用 LVM 可实现硬盘镜像，当一个盘坏掉，仍有一份数据在镜像盘里，数据仍可进行读写，但此时数据不再有可用性，若镜像盘也坏掉则数据全部丢失。所以在此例中，PV 丢失(LVM_PVMISS)的信息会大幅显示在控制台面上，从而提醒用户去仔细查看 error log 找出故障并修复它。同样，此例中 HACMP 提供了界面，结合 AIX 的功能，从而监控故障的发生。

c.应用故障

如果用户的应用有 kernel call 调用，或以 root 身份来启动等，一旦应用发生故障,很容易导致操作系统 down 掉,发生死机,这时实际上等于节点故障,HACMP

会采取相应接管措施。如果只是应用自身死掉,AIX 仍正常运行,HACMP 最多利用 Error Notification Facility 来提供监控功能，对应用本身不采取任何动作。但如果应用中调用了 AIX 的 SRC (System Resource Controller)机制所提供的 API 接口，就可以使应用在 down 掉后自动重新启动。除了 SRC 提供 API 接口外，HACMP 中的 clinfo 也提供这样的 API。

clinfo 是 cluster Information daemon,它负责维护整个 cluster 的状态的信息，clinfo API 允许应用程序利用这些状态信息来采取相应行动。

d. HACMP 故障

如果 cluster 中节点的 HACMP 进程 down 掉，HACMP 将其升级为节点故障，从而发生资源接管。

如上所述，HACMP 只全权负责诊断网卡故障、网络故障和节点故障这三类故障，并负责实现 IP 地址转换或接管，以及整个系统资源(硬件、文件、系统、应用程序，等等)的接管。对于这三类故障外的其他故障，可以结合 AIX 基本功能和 HACMP 提供的一些机制，如 Error Notification Facility, clinfo API 等，同样可以实现对故障的监控并采取相应措施。

HACMP 常见故障解决：

1、导致集群中节点失效的无反映开关 （Deadman Switch）

问题现象：
集群中的节点经历着极端的性能问题，如：大量的 I/O 传输、过多的错误记录、内存不足等，导致集群管理器（clstrmgr）没有得到足够的 CPU 处理时间，而引起无反映开关在分配的时间被重置。某个应用程序运行权限高过集群管理器时,会导致此问题。

解决方法：
术语“ Deadman Switch ”指的是在特定集群条件下，未能及时重置该开关，引起系统宕机和转储的内核扩展部分。无反映开关在超过了特定的时间限制后会宕掉处于挂起状态的节点。此过程导致集群中的其它节点接管处于挂起状态节点的资源。要解决此问题需要解决与之相关的几个性能问题：
1、调整系统 I/O pacing
2、增加信息同步（syncd）的频率
3、增加通信子系统使用的内存量
4、更改错误探测速率

1、调整系统使用 I/O 的步调：

使用 I/O pacing 调整系统，使得在大量写*作时，系统资源的分配更合理。为 HACMP 集群激活 I/O Pacing 是必要的，尤其是在集群中可能会有大量磁盘数据块写*作的时侯。
按下述步骤修改 I/O Pacing 设置：
smitty hacmp
Cluster Configuration
Advanced Performance Tuning Parameters
Change/Show I/O Pacing
修改 HIGH water mark for pending write I/Os per file 域，推荐值为 33，可用值在 0－32767 之间 。
修改 LOW watermark for pending write I/Os per file 域，推荐值为 24，可用值在 0－32767 之间。
不同的系统，以上两个值也不同。修改上两个值只能稍微减少写次数，通常能够解决上述问题。

2、增大 syncd 的运行频率：
增加 syncd 的运行频率，使缺省 60 秒运行一次变为 30 秒、20 秒或 10 秒运行一次。这样可以强迫增加 I/O 刷新速率，并减少由于沉重的 I/O 流量触发无反映开关的可能性。
按下述步骤修改 syncd 运行频率设置：
smitty hacmp
Cluster Configuration
Advanced Performance Tuning Parameters
Change/Show syncd frequency
修改 syncd frequency in seconds 域，推荐值为 10 秒，可用值在 0－32767 之间。

3、增加通信子系统可用的内存量：
如果运行命令：# netstat －m，发现请求 mbuf 被拒绝，或运行命令# errpt 发现 LOW_MBUFS 错误，则应增加网络参数“ thewall ”的值。Thewall 的缺省值为 25%的系统实内存。可以将其增加为 50%的系统实内存。
按下述步骤修改 thewall 值的设置：
vi /etc/rc.net
在此文件的末尾加入：
no -o thewall= xxxxx
xxxxx 是指你希望设置的供通信子系统使用的实内存值。如：,
no -o thewall=10240

4、修改错误探测速率：
如果激活 I/O Pacing 或增加 Syncd 运行频率不能解决无反映开关不能重置的问题时，则修改错误探测速率，将其值该为 Slow。这样可以延长一个挂起节点调用无反映开关之前，以及接管节点探测到节点故障并获得挂起节点资源之前所需的时间。
注意：
在完成上述步骤之前，I/O Pacing 必须先激活。这是因为修改此设置会调整 I/O 数据的传输量。

二、双机用户和组不一致

HACMP 的常用命令和 SMIT 工具：

/usr/es/sbin/cluster/utilities/cllscf Show Cluster Topology

2、/usr/es/sbin/cluster/utilities/cllsclstr Show Cluster Definitions

3、/usr/es/sbin/cluster/utilities/cllsnode Show Topology Information by Node

4、/usr/es/sbin/cluster/utilities/cllsnw Show Topology Information by Network Name

5、/usr/es/sbin/cluster/utilities/cllsif Show Topology Information by Network Adapter

6、/usr/es/sbin/cluster/utilities/clshowres -g'sxptt_con_rg'

Show Resource Information by Resource Group

7、/usr/es/sbin/cluster/utilities/clshowres -n's85a'

Show Resource Information by Node

1、查看 Cluster 的运行情况：

/usr/sbin/cluster/clinfo –a

/usr/sbin/cluster/clstat

/usr/sbin/cluster/clstat 可以帮助你查看当前 HACMP 的节点状态。

屏幕会如下显示：

clstat - HACMP for AIX Cluster Status Monitor

Cluster: cluster1 (1) Tue Jul 20 09:52:03 CDT 1999

State: UP Nodes: 2

SubState: STABLE

Node: j50a State: UP <--节点 A 状态

Interface: j50_a_srv (0) Address: 192.9.200.1 <-- 服务 IP 地址

State: UP <--服务 IP 状态

Interface: j50_a_tty1 (1) Address: 0.0.0.0

State: UP <--心跳线状态

Node: j50b State: UP <--节点 B 状态

Interface: j50_b_srv (0) Address: 192.9.200.2 <-- 服务 IP 地址

State: UP <--服务 IP 状态

Interface: j50_b_tty1 (1) Address: 0.0.0.0

State: UP <--心跳线状态

2、启动 HACMP：

smitty clstart

注：有两种启动 HACMP 的方式：

now：手工启动 HACMP

restart、both：在系统启动时自动启动 HACMP

3、停止 HACMP：

smitty clstop

注：有三种停止方式：

graceful：只停止本节点上 HACMP 的运行，并释放由本节点管理的资源，但允许其它节点接管。

graceful with takeover：停止本节点上 HACMP 的运行，释放资源，让其他节点接管。

forced：停止本节点上 HACMP 的运行，但不释放资源。

4、查看 Cluster 的进程状态：

ps －ef | grep cluster

注：应有三个 HACMP 进程：clstrmgr、clinfo、clsnuxpd

5、查看 Cluster 的日志及错误信息：

more /tmp/hacmp.out

more /var/adm/cluster.log

注：可以在启动 HACMP 时使用 # tail －f /tmp/hacmp.out 命令，以查看 HACMP 的

启动是否正常或跟踪启动时的错误信息。

6、查看 Cluster 运行的历史记录：

cd /usr/sbin/cluster/history

注：此目录下存放着每天的 Cluster 运行记录

7、查看 Cluster 运行时的网络情况及资源组的使用情况：

HACMP 启动之前：

netstat –i

此时应可以看到 boot 和 standby 地址

lsvg –o

此时只能看到本地的 VG

HACMP 启动之后：

netstat –i

此时应可以看到 service 和 standby 地址

lsvg –o

此时应可以看到本地的 VG 及共享 VG

8、存储和恢复 Cluster 的配置：

存储 cluster 的配置：

#smitty hacmp

选择：Cluster Configuration→Cluster Snapshots→Add a Cluster Snapshot

键入 Snapshot 文件名（Cluster Snapshot Name）

恢复 cluster 的配置：

smitty hacmp

选择：Cluster Configuration→Cluster Snapshots→Apply a Cluster Snapshot

键入 Snapshot 文件名（Cluster Snapshot Name）

注：菜单中各项的意义：

Cluster Snapshot Name：指定 Cluster Snapshot 的文件名

Cluster Snapshot Des cription：对 Cluster Snapshot 地描述

Un/Configure Cluster Resource：是否重新配置 Cluster 资源

Force Apply if verify fails：当 Cluster 校验失败时，是否强制恢复配置

9、注意：向由 HACMP 管理的共享 VG 中增加 FS 时的具体步骤为：

1)、# smitty

选择：System storage management

File systems

Add/change/show/delete file systems

Journaled file systems

Add a journaled file system

Add a standard journaled file system

选择共享 VG

指定 FS 的大小，Mount point

2)、# smitty clstop

停止 HACMP 的运行

3)、在增加 FS 的节点上作：

varyoffvg SHARE_VG_NAME

exportvg SHARE_VG_NAME

在另一个节点上作：

importvg –y SHARE_VG_NAME –f hdiskX

varyonvg SHARE_VG_NAME

作此步的目的是将新加入的 FS 的定义取过来

4)、# smitty hacmp

选择：cluster configuration

cluster resources

define resource group

change/show resources for a resource group

将增加的 FS 加入到 Cluster 资源组的定义中

5)、# smitty hacmp

选择：cluster configuration

cluster resources

synchronize cluster resources

同步 Cluster 的资源组的配置

6)、# smitty hacmp

选择：cluster configuration

cluster verification

进行 Cluster 配置的校验

附录

2 附录一：HACMP 脚本

2.1 ha_start 文件（HACMP 启动脚本）

```
/usr/sbin/cluster/etc/rc.cluster -boot -Nbi
```

2.2 ha_list 文件（查看 HACMP 状态脚本）

```
echo '============ Cluster Processes ============' > /tmp/ha_list.out
ps -ef | grep cluster | egrep -v grep >> /tmp/ha_list.out

echo >> /tmp/ha_list.out
echo '============ Network Status ============' >> /tmp/ha_list.out
netstat -i >> /tmp/ha_list.out

echo >> /tmp/ha_list.out
echo '============ VG Status ============' >> /tmp/ha_list.out
lsvg -o >> /tmp/ha_list.out

echo >> /tmp/ha_list.out
echo '============ Filesystems Mounted ============' >> /tmp/ha_list.out
mount >> /tmp/ha_list.out

echo >> /tmp/ha_list.out
echo '============ Applications ============' >> /tmp/ha_list.out
ps -ef | grep db2 | egrep -v grep >> /tmp/ha_list.out
ps -ef | grep btp | egrep -v grep >> /tmp/ha_list.out

pg /tmp/ha_list.out
rm /tmp/ha_list.out
```

2.3 ha_stat 文件（查看集群状态脚本）

```
/usr/sbin/cluster/clstat –a
```

2.4 ha_stop 文件（HACMP 关闭脚本）

```
/usr/sbin/cluster/utilities/clstop -y –Ngr
```

2.5 btpdb2_start（DB2、BTP 启动脚本）

注：在备份机上，应将“ cp -f /btp/etc/btppaswd.rza /btp/etc/btppaswd.dat ”语句中的“ btppaswd.rza ”改为“ btppaswd.rzb ”。

- 未安装“ 管理服务器（DAS）”的启动脚本

```
#!/bin/ksh
#
```

```
# Start Db2 Instance
#
su - db2inst1 -c "db2start"
#
chown db2inst1:db2iadm1 /dev/rcontainer0[0-3]lv
#
# Start BTP
#
su - btp -c "
cominit -s                  # Initial COMM module
sleep 1
comctrl -s                  # Start COMM processes
sleep 1
cp -f /btp/etc/btppaswd.rza /btp/etc/btppaswd.dat   # Get the BTP password file
btpinit -r `head -1 /btp/etc/btpdates.dat | cut -f4 -d' '`   # Re-initialize the BTP system
sleep 3
btpctrl -i"                   # Start BTP processes

exit 0
```

- 安装了“管理服务器（DAS）”的启动脚本

```
#!/bin/ksh
#
# Start Db2 Instance and DB2 Administrative Server
#
su - db2inst1 -c "db2start
db2admin start"
#
chown db2inst1:db2iadm1 /dev/rcontainer0[0-3]lv
#
# Start BTP
#
su - btp -c "
cominit -s                  # Initial COMM module
sleep 1
comctrl -s                  # Start COMM processes
sleep 1
cp -f /btp/etc/btppaswd.rza /btp/etc/btppaswd.dat   # Get the BTP password file
btpinit -r `head -1 /btp/etc/btpdates.dat | cut -f4 -d' '`   # Re-initialize the BTP system
sleep 3
btpctrl -i"                   # Start BTP processes

exit 0
```

2.6 btpdb2_stop（DB2、BTP 关闭脚本）

- 未安装“管理服务器（DAS）”的关闭脚本

```
#!/bin/ksh
#
# Stop BTP system
#
su - btp -c "
btpctrl -e          # Stop all the BTP processes
sleep 3
echo y | btpinit -c `head -1 /btp/etc/btpdates.dat | cut -f4 -d' '`          #Clear the
BTP environment
sleep 1
comctrl -c          # Stop all the COM processes
sleep 1
cominit -c"         # Clear COMM environment

#
# Stop DB2 Instance
#
su - db2inst1 -c "
db2 force application all
db2stop"

echo '********** Applications Shutdown Completed! **********'

exit 0
```

- 安装了“管理服务器（DAS）”的关闭脚本

```
#!/bin/ksh
#
# Stop BTP system
#
su - btp -c "
btpctrl -e          # Stop all the BTP processes
sleep 3
echo y | btpinit -c `head -1 /btp/etc/btpdates.dat | cut -f4 -d' '`          #Clear the
BTP environment
sleep 1
comctrl -c          # Stop all the COM processes
sleep 1
cominit -c"         # Clear COMM environment
#
# Stop DB2 Instance and DB2 Administrative Server
#
su - db2inst1 -c "db2admin stop
db2 force application all
db2stop"
echo '********** Applications Shutdown Completed! **********'

exit 0
```