EtherChannel & Highly Available Cluster Multiprocessing (HACMP) in AIX V5.2

How-to and Test Experiences

Abstract: This document gives tips and a working example of how to a Highly Available Cluster Multiprocessing (HACMP) user could implement EtherChannel with HACMP. Support for this combination was announced in May, 2004.

Authors: Shawn Bodily (HACMP) and Cindy Young (EtherChannel) of IBM pSeries Advanced Technical Support and Michael Herrera (HACMP) of IBM pSeries AIX Support

Introduction

IBM AIX pSeries administrators have expressed interest in combining these components for several reasons. Those accustomed to other software availability solutions object to HACMP's additional "standby" adapter. With EtherChannel, HACMP setups could mask the standby adapter giving an outward appearance familiar to these users. Other users like the aggregated bandwidth, load balancing, or high availability benefits of EtherChannel. The result is a lower cost, high performance network that is also popular as a high speed private (non-switch) interconnect between machines.

In this test, we successfully implemented a "single adapter network" HACMP IP Address Takeover (IPAT) with the EtherChannel function included in AIX V 5.2. The EtherChannel was responsible for providing local adapter swapping – outside of HACMP. HACMP has no knowledge of EtherChannel and is completely independent. While a single adapter network is normally not ideal, EtherChannel makes this okay because there are multiple physical adapters within the single EtherChannel pseudo device. Thus, we could safely ignore the insufficient adapter warning messages posted during cluster synchronization.

Our configuration consisted of a rotating resource group with a single adapter network using IP aliasing. Our testing proved to be beneficial in simplifying the HACMP setup. We implemented the EtherChannel connection without a network switch, cabling the two test systems directly with crossover cables.

Although PCI adapter hot plug option and Hardware Address Takeover were excluded from the HACMP support announcement, our tests proved that the PCI hot plug feature will work due to the new EtherChannel Dynamic Adapter Membership (DAM) feature introduced in the May 2004 software update. This means that a failed adapter could be removed from a running EtherChannel in SMIT, the user could physically remove and replace it using the hot swap options, and the new adapter could be returned to the EtherChannel via SMIT with no disruption to the service to that IP address.

AIX EtherChannel Overview for HACMP Users

EtherChannel (EC) is a port aggregation method whereby up to eight Ethernet adapters are defined as one EtherChannel. Remote systems view the EC as one IP and MAC address so up to eight times network bandwidth is available in one network presence. Traffic is distributed across the adapters in the standard way (address algorithm) or on a round robin basis. If an adapter fails, traffic is automatically sent to the next available adapter in the EC without disrupting user connections. When only one link in the main EtherChannel is active, a failure test triggers a rapid detection / failover (in 2-4 seconds) to optional backup adapter with no disruption to user connections. Two failure tests are offered – the physical adapter link to network and the optional TCP/IP path to the user-specified node. When failure is detected, the MAC and IP addresses are activated on the backup adapter. When at least one adapter in the main channel is restored, the addresses are reactivated on the main channel.

The AIX V5.1 Network Interface Backup (NIB) configuration mode was replaced and enhanced in AIX V5.2. The new method is a single adapter EtherChannel with backup adapter, providing a priority (failback upon link repair) between the primary and backup links which the previous implementation lacked. The Dynamic Adapter Membership (DAM) enhancement in the latest version of AIX V 5.2 allows dynamic reconfiguration of adapters within the EtherChannel without disruption to the running connection. Although not tested for the May, 2004 HACMP Support announcement, our tests show that this dynamic reconfiguration enables PCI adapter hot plug on those HACMP and EC systems with the appropriate hot plug hardware.

Why implement EtherChannel?

Users choose EtherChannel for various reasons. With HACMP, it simplifies the topology, increases bandwidth, and reduces the number of IP subnets required.

- **Higher bandwidth and load balancing options**
  - multi-adapter channels utilize aggregate bandwidth
  - several user configurable alternatives for directing traffic across the channel adapters
- **Built in availability features**
  - automatically handles adapter, link and network failures
  - optional backup adapter to avoid SPOF (single point of failure) at network switch
  - design techniques to avoid SPOFs
- **A simple, flexible solution and growth path**
  - one Ethernet MAC and IP address for entire aggregation (including backup adapter)
  - accommodates future bandwidth requirements easily
  - user can add, delete, and reconfigure adapters on the fly (no service disruption)
- **Various options for interoperability with network switch**
  - multi-adapter channels for both EtherChannel and 802.3ad capable switches
  - single adapter channels and backup adapter links are transparent to the network switch
  - channel backup adapter option (connect to a different network switch to avoid SPOF)
  - channel operates without switch when two systems cabled directly (back-to-back)
- **It's free!**
  - included in AIX and regularly enhanced since AIX v4.3.3

All multi-adapter channels require special EtherChannel or IEEE 802.3ad port configuration in the network switch. In most cases, the switch will be configured for EtherChannel mode. However, if the switch doesn't support EC or if the corporation has standardized on IEEE 802.3ad, then configure 802.3ad at both the switch and in AIX. Single-adapter links, on the other hand, require no special configuration at the network switch. This includes a single-adapter EtherChannel and the backup adapter connection. It is also possible to run an EtherChannel between two AIX systems without a network switch. We implemented this non-switch EtherChannel connection in our test environment by cabling the adapters directly in a two-machine setup.

EtherChannel in HACMP Environments

In recent years there has been a significant progress in the way that we configure IPAT within HACMP. The three main IP Address Takeover (IPAT) scenarios are depicted in Figures 1a, 1b, and 1c.

The first topology model, IPAT via Replacement, involves boot and standby adapters on separate subnets. The boot address is replaced by the service IP address when cluster services are started. Although effective, this model is unconventional for environments that need to implement multiple service IP addresses. Cluster administrators were forced to customize their environment with pre- and post-events to set up any additional aliases and make sure that they were removed before another failover.

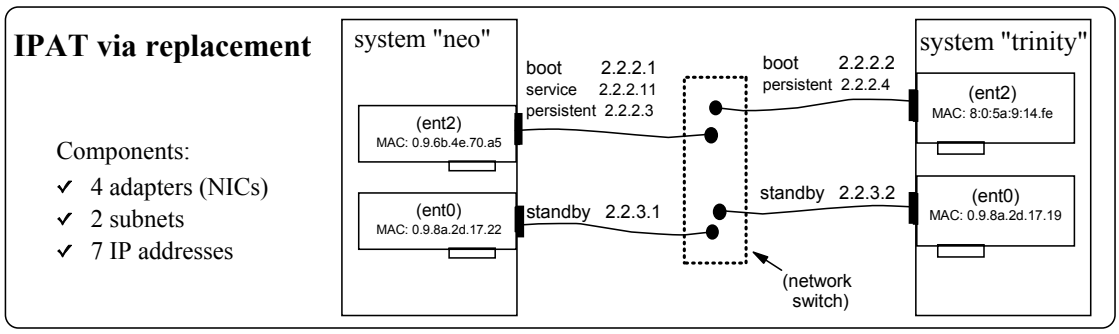

Figure 1a. Traditional HACMP IP Address Takeover (IPAT) via replacement scenario

HACMP V4.5 introduced IPAT via Aliasing as the new default topology. In this new model, the standby adapter function has been replaced with another boot. The subnet requirements are different in that an additional subnet is required. Each boot needs its own subnet and any service or persistent IP addresses will operate on its own subnet, for a total of three subnets. The boot IP addressess no longer disappear when cluster services are started and service IP address is acquired. This design is different from the previous because multiple service IP addresses exist within the same HACMP network and are handled via aliasing.

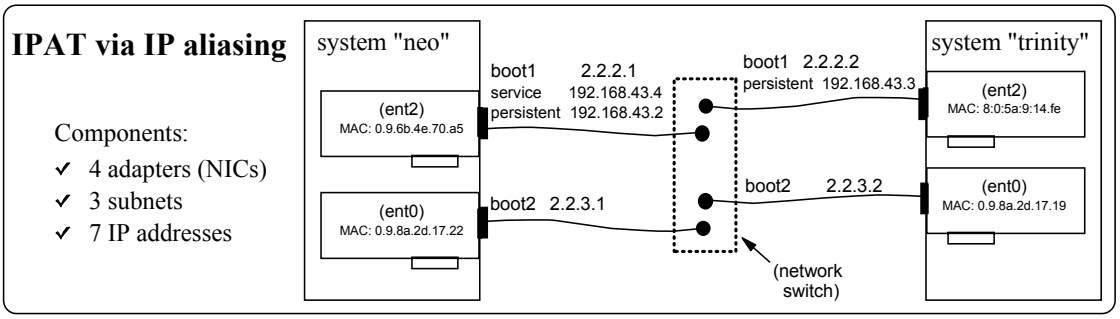

Figure 1b. HACMP IPAT via IP aliasing scenario

The third model, shown in Figure 1c, masks the underlying Ethernet adapters behind a single "ent" interface and handles the redundancy and load balancing under the covers. It is not a replacement for either of the previous models -- it works with both. Because the EtherChannels on each node are configured to be redundant, we can define each one within HACMP as a single adapter network using IP aliasing. Since only one adapter is defined on each node, only two subnets are required -- one for the boot (the base IP address on each node) and one for the highly available service(s).

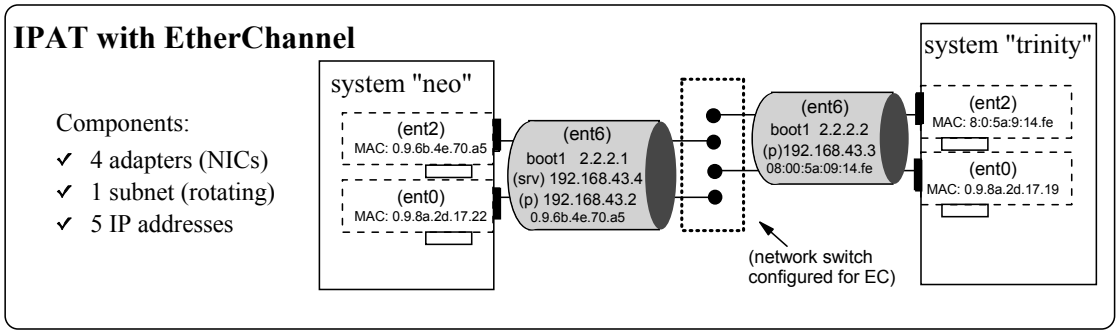

Figure 1c. HACMP IPAT with EtherChannel

In Figure 1c, the "*en6*" IP interface is configured atop the *ent6* adapter (the EtherChannel pseudo device). A persistent IP address was defined on each of the EtherChannels in order to maintain connectivity to that subnet when HACMP services are not online. The sample topology as shown via *cllsif:*

```
Adapter Type Network Net Type Attribute Node IP Addr Hardware Addr Interface Name Global Name Netmask
neo_boot1     boot    channet ether public neo     2.2.2.1       en6   255.255.255.0
neoec_srv     service channet ether public neo     192.168.43.4        255.255.255.0
trinity_boot1 boot    channet ether public trinity 2.2.2.2       en6   255.255.255.0
neoec_srv     service channet ether public trinity 192.168.43.4        255.255.255.0
```

Although we did not configure one for our tests, we still recommend configuration of some type of serial network to prevent situations where the cluster can become partitioned. The same applies for the use of a *netmon.cf* file.

Once configured, the loss of traffic on the links is viewed in the `netstat -v` output and errors will be logged in the error report. Since the failovers are handled by the EtherChannel logic, HACMP adapter maintenance is minimized. We would no longer expect to see local SWAP_ADAPTER, FAIL_INTERFACE or FAIL_STBY events, nor the removal of routes in the event of a local adapter failure. The failure is seamless to HACMP.

Test Environment Overview:

Our test environment was constructed using these main components.

- two pSeries p630 systems (named *neo* and *trinity*)
- AIX V5.2 plus May 2004 update CD 5200-03 – **Requirements outlined in announcement flash**
- HACMP v5.1 (5.1.0.5)
- Ethernet network connections ent0 through ent6:
  - ent1 - administrative network connection, attached via an Ethernet network switch
  - ent4 – unused
  - ent6 - EtherChannel (comprised of ent2, ent3 and ent0)
- three UTP Ethernet crossover cables (see the References section)

Figure 2 illustrates the test environment. Our lab systems, *neo* and *trinity,* are identical p630 nodes. Each system has an integrated Ethernet adapter (ent0) and a 4-Port Ethernet adapter (ent1-ent4).

The first port of the 4-port adapter (ent1) provides day-to-day access via the production network. We neither used nor disrupted this interface in our tests.

The last port of the 4-port adapter (ent4) remained unconfigured.

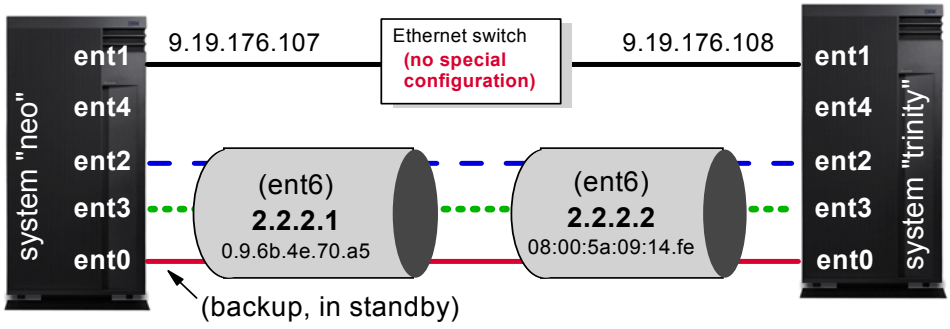

Figure 2. EtherChannel and HACMP test environment

EtherChannel planning. Ethernet switch manufacturers expect attachment of the individual links in the EtherChannel at the same network switch. Connect the backup adapter to a second switch for added redundancy. Remember that the backup adapter is idle during normal operation until it becomes the last adapter standing in the EtherChannel. At that point, the EtherChannel backup adapter utilizes the path over the second switch.

Choose the adapters for the EtherChannel carefully. The goal is to avoid a single point of failure. In the test environment, we had an integrated Ethernet adapter and a single 4-port Ethernet adapter on each system so we chose to configure the integrated adapter as the backup so that the channel would continue to run even if the 4-port adapter failed.

EtherChannel back-to-back connection details and special considerations. We configured a two-link EtherChannel with backup link for the test. We eliminated the 4-Port Ethernet adapter as a single-point-of-failure by configuring the integrated ent0 port as the EtherChannel backup adapter -- adapters ent2 and ent3 became the main channel and ent0 become the backup link. Normally, the two-link main channel would be connected to an Ethernet switch configured for EtherChannel (as shown in Figure 1c) and the backup adapter would be connected to a second Ethernet switch for switch redundancy. However, we didn't have extra Ethernet switches in our lab so we created a simple test environment without a network switch by connecting the three ports directly on the two systems via crossover cables. This gave us the connectivity for a great two-system test, without acquiring and configuring the switches.

This simple setup is terrific for testing EtherChannel behavior. We used the `netstat -v ent6` command to view the distribution of the traffic (send/receive statistics) across the adapters in the EtherChannel. It does, however, limit the connectivity to two systems which was okay for our tests. Also, our non-switch environment reflected the AIX EtherChannel software time for triggering the backup adapter – making the swap seem

instantaneous. In a switch-based setup, there is a short delay after the backup adapter awakes as the switch registers the new system. In this two-system setup, each system is sending to only one IP address so we configured utilized both adapter in the EtherChannel with "round robin" mode. In "standard" mode, there is only one target IP address so the algorithm would always selects the same adapter. Configuring round robin mode optimizes bandwidth and uses all adapters with minimal exposure to out-of-order packets.

All of the ports in the EtherChannel were configured for the same speed, duplex mode and TCP/IP MTU size. This is the normal and expected configuration for EtherChannel. Although adapter mismatches may work in certain circumstances and AIX software doesn't prohibit the configuration, users avoid troubleshooting headaches by starting out with matching configurations. The network switches are typically more restrictive than AIX, so expect the switch to enforce the matched configuration.

Configuration Procedures:

We set up our cluster via the following basic steps. Details on each step, as completed for system *neo*, follow.

1. Check the Ethernet adapter configurations and adapter cabling
2. Create EtherChannel interface
3. Configure IPs on new interface (en6) via TCP/IP
4. Add boot & service IPs to HACMP topology
5. Create a resource group and assign it the service IP
6. Synchronize cluster
7. Start cluster services
8. Test redundancy of NICs and make sure HACMP does not detect it

Start with unconfigured adapters, cabled together as shown in Figure 2. Our adapters had been configured previously so we removed the ODM interface definitions via `smitty inet.` We completed these basic steps on both systems, using the IP interfaces, MAC and IP addresses shown in Figure 2.

Notes:
To avoid potential problems with HACMP auto-discovery with adapter interfaces (en#) previously configured, remove the interfaces in the *smitty inet* fastpath. Alternatively, `ifconfig down` the interface, `detach` it, and `rmdev` the definition to remove it from the ODM.

In Gigabit Ethernet adapter environments, users can improve data transfer performance by configuring jumbo frames. To configure a Gigabit EtherChannel, enable jumbo frames in steps 1 and 2 and set the 9000-byte MTU via fast path `smitty chif` once the interface has been created in step 3.

Step 1. Check the Ethernet adapter configurations and adapter cabling.
The adapters that will become a part of the EtherChannel should be configured for the same speed and duplex mode. We configured ent0, ent2 and ent3 for 100 Mbps, full duplex.

1. Configure speed and mode via fastpath `smitty ethernet.`
2. Verify that the three adapters pairs are cabled together between the systems as shown in Figure 2.

Notes:
- At this point, one could test these links by configuring IP addresses on each side. That's probably a good idea if the cabling method is new. Just remember to remove the configuration prior to the next step.
- Configuring the EC automatically triggers key changes in underlying adapters (e.g. link polling, alternate address, and so on. However, while jumbo frames usage can be enabled/disabled in SMIT, this change is not reflected at runtime.

Step 2. Configure EtherChannel.

Configure the EtherChannel through the fastpath `smitty etherchannel` and select the appropriate adapters via F7. In our configuration, ent2 and ent3 comprise the main channel and ent0 is the backup adapter. Processing the following menu creates the new EtherChannel interface (ent6) as pictured in Figure 2.

```
                        Add an EtherChannel / Link Aggregation

                                                        [Entry Fields]
  EtherChannel / Link Aggregation Adapters             ent2, ent3
  Enable Alternate Address                             no
  Alternate Address                                   []
  Enable Gigabit Ethernet Jumbo Frames                 no
  Mode                                                 round robin
  Hash Mode                                            default
  Backup Adapter                                       ent0
  Internet Address to Ping                            []
  Number of Retries                                   []                      +#
  Retry Timeout (sec)                                 []                      +#
```

Notes:

- By default, the entire EtherChannel uses the MAC address of the first adapter in the channel. Use the Alternate Address fields to set a different MAC value.
- As previously explained, we selected round robin mode because both links will be utilized in this two-IP address environment. Please refer to the EtherChannel documentation to learn about the different modes and select the one that will best suit your configuration.
- Poor EtherChannel aggregate performance and/or "round robin failure behavior" indicate mismatches. Check for mismatched jumbo frames, switch aggregation configuration and resultant rapid MAC address movement between switch ports.

Step 3. Configure the IP addresses on the EtherChannel.

Now configure the IP interface (en6) on the EtherChannel using fastpath `smitty chinet`. We repeated this step on node *trinity* using an address of 2.2.2.2, also on en6.

```
            Change / Show a Standard Ethernet Interface

  Type or select values in entry fields.
  Press Enter AFTER making all desired changes.

                                                      [Entry Fields]
    Network Interface Name                            en6
    INTERNET ADDRESS (dotted decimal)                [2.2.2.1]
    Network MASK (hexadecimal or dotted decimal)     [255.255.255.0]
    Current STATE                                     up                     +
    Use Address Resolution Protocol (ARP)?            yes                    +
    BROADCAST ADDRESS (dotted decimal)               []
    Interface Specific Network Options
      ('NULL' will unset the option)
      rfc1323                                        []
      tcp_mssdflt                                    []
      tcp_nodelay                                    []
      tcp_recvspace                                  []
      tcp_sendspace                                  []
```

Note:

- This screen created the en6 IP interface. Remember to look for en6 when running familiar TCP/IP commands. The interfaces for the individual adapters that comprise the EtherChannel (en0, en2 and en3) do not exist.

Step 4. Configure HACMP Topology.
For the purposes of our testing we chose to use IP aliasing when defining our HACMP network (channet). We configured our boot IP addresses on each EtherChannel (neo_boot 2.2.2.1, trinity_boot 2.2.2.2). Then defined our service IP address (bound to multiple nodes) 192.168.43.4 and our persistent IP addresses, 192.168.43.X on node and 192.168.43.X on trinity.

We selected the IP addresses above in order to test with a 1-way cascading configuration. We could have just as well have selected a service IP address on the same subnet as our boots and configured a rotating configuration.

Although omitted from our setup, HACMP administrators should configure a non-IP serial network in a production environment.

Step 5. Configure HACMP Resources.
We configured one cascading resource group with a single service IP address defined to it. Since our focus was on the NIC redundancy testing, we simplied the configuration by omitting additional resources.

Step 5a.
Use fastpath `smitty hacmp` → *Initialization and Standard Configuration* → *Configure HACMP Resource Groups* → *Add a Resource Group* → *Cascading* (from pick list)

```
      Add a Resource Group with a Cascading Management Policy (standard)

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                [Entry Fields]
* Resource Group Name                           [testec]
* Participating Node Names / Default Node Priority   [neo trinity]               +
```

Press <Enter> to create the resource group. The next step is to add the service IP address to the resource group.

Step 5b.
Use fastpath `smitty hacmp` → *Initialization and Standard Configuration* → *Configure HACMP Resource Groups* → *Change/Show Resources for a Resource Group (standard)* → *Choose previously created resource group*

```
             Change/Show Resources for a Cascading Resource Group

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                [Entry Fields]
  Resource Group Name                           testec
  Participating Node Names (Default Node Priority)    neo trinity

* Service IP Labels/Addresses                   [neoec_svc]             +
  Volume Groups                                 []                      +
  Filesystems (empty is ALL for VGs specified)  []                      +
  Application Servers                           []                      +
```

Step 6. Synchronize the cluster.
Though HACMP familiarity is assumed for this doc, we wanted to show what type of message is displayed when HACMP topology is configured as a single adapter network.

Use fastpath `smitty hacmp` → *Initialization and Standard Configuration* → *Verify and Synchronize HACMP Configuration --> <Enter>*

```
Verifying Cluster Topology...
```

```
WARNING: There may be an insufficient number of communication interfaces defined
on node: neo, network: net_ether_01. Multiple communication interfaces are
recommended for networks that will use IP aliasing.

WARNING: There may be an insufficient number of communication interfaces defined
on node: trinity, network: net_ether_01. Multiple communication interfaces are
recommended for networks that will use IP aliasing.
```

The above warning message would be expected in this particular environment. Since it is considered a single adapter network, configuring a netmon.cf is desired as well.

Step 7. Start cluster services.
Execute s*mitty clstart* on each node and wait for node_up _complete.

Step 8. Test procedures.
The majority of the testing focused around pulling physical cables to see how the system responded and to make sure HACMP was unaware.

While performing each test, we ran a ping from an outside client node, to both boot IPs and the service IP. In none of our testing did we drop a single ping packet.

1. Pulled the cable from ent3. This resulted in continued service surviving on ent2. This was verified with *netstat* and *entstat* commands, as shown in the Appendix 2. Along with the surviving ping running from the client. AIX makes note of this in the error report. HACMP however, is none the wiser that a failure occurred. The following errors showed in the errpt:

   ```
   F77ECAC2   0624145904 T H ent3      ETHERNET NETWORK RECOVERY MODE
   8650BE3F   0624145904 I H ent6      ETHERCHANNEL RECOVERY
   F77ECAC2   0624145904 T H ent2      ETHERNET NETWORK RECOVERY MODE
   ```

2. Pulled the cable from ent2. This caused the standby adapter of ent0 to takeover the services. Much like the previous tests, AIX noted failure in the error report, but not HACMP. Now since we used crossover cables, this had a dual effect of causing similar errors and swaps on both nodes.

   Note: AIX V5.2 (at July 2004 level) and V5.3 administrators can force the failure via the new `/usr/lib/methods/ethchan_config -f ent6` command.

3. We then pulled the lone surviving adapter of ent0. This, in turn, resulted in a full EtherChannel failure, which was noticed a failed network by HACMP and caused a takeover to occur. However, because of our crossover cable configuration, the resource group could not properly fall over as no communications were available. The resource group just went offline.

4. We then stopped cluster services, plugged the cables back in and rebooted nodes.

5. Restarted cluster services on node *trinity* and the let it acquire the resources back.

That concluded our basic testing with EtherChannel and HACMP.

Changing the Adapter Configuration in a running EtherChannel.

In this new version of AIX V 5.2, a new function, Dynamic Adapter Membership, allows the user to add, delete and alter the configuration of adapters within the EtherChannel without disrupting the EtherChannel. Here are some special considerations for this environment.

- When adding a new adapter to an existing EtherChannel, you must consider the EtherChannel pseudo device capabilities which comprise the common attributes of all underlying capabilities. They can be viewed `entstat` or `netstat -v`. If their capabilities do not match, you will not be able to add the adapters dynamically and will have to select the option to apply the updates the database only. In that case, the changes take effect when the system is rebooted or the EtherChannel is removed with an `rmdev` and reconfigured.

- When testing for failover behavior, think through configuration implications at switch. When reconfiguring, remember to recable and reconfigure at switch -- the switch aggregation configuration must reflect new, moved, and deleted adapters! Try this simple procedure when replacing a failed adapter:
    1. Unplug appropriate cable(s) at adapter or at switch side. Make desired EtherChannel adapter configuration changes in SMIT
    2. Make associated EtherChannel configuration changes at network switch
    3. Replug cables into the correct adapter or switch ports

Observations and Conclusions.

Our overall thoughts about the implementation of EtherChannels in an HACMP environment were very positive. Although the configuration will require some additional initial planning, it was very quick and easy to setup. We were especially pleased with the recovery times of our testing; they were almost instantaneous and had no impact on our running cluster. We were also pleased at how the implementation of this model eliminates the removal of routes in HACMP events associated with local adapter swaps, making the failure time shorter and easier to troubleshoot.

In summary, the simplicity and overall benefits of the EtherChannel model make it a very promising choice when planning a new environment that needs HACMP's availability with scalable network bandwidth and redundancy. The dynamic scalability and possibilities for even greater redundancy in crossover environments were an even bigger incentive to consider migration to this type of configuration.

Appendix 1 – Useful Commands:

The AIX administrator will find familiar commands are used with the EtherChannel interface. One unique command, ***/user/lib/methods/etherchan config –f ent,*** was introduced in AIX V 5.2 (the July, 2004 update) and AIX V5.3, allowing users to force failover between the main and backup channel. This command is useful for testing the behavior prior to implementing the solution in a production environment.

The following commands, shown with sample output from the test environment, are useful for system analysis and troubleshooting.

```
neo.dfw.ibm.com /#lsdev -Cc adapter | grep en
ent0   Available 1L-08  10/100 Mbps Ethernet PCI Adapter II (1410ff01)
ent1   Available 12-00  IBM 4-Port 10/100 Base-TX Ethernet PCI Adapter (23100020)(Port 1)
ent2   Available 12-08  IBM 4-Port 10/100 Base-TX Ethernet PCI Adapter (23100020)(Port 2)
ent3   Available 12-10  IBM 4-Port 10/100 Base-TX Ethernet PCI Adapter (23100020)(Port 3)
ent4   Available 12-18  IBM 4-Port 10/100 Base-TX Ethernet PCI Adapter (23100020)(Port 4)
ent5   Available 14-08  10/100 Mbps Ethernet PCI Adapter II (1410ff01)
ent6   Available        EtherChannel / IEEE 802.3ad Link Aggregation
```

```
neo.dfw.ibm.com /#lsdev -Cc if
en0 Available 1L-08 Standard Ethernet Network Interface
en1 Available 12-00 Standard Ethernet Network Interface
en2 Available 12-08 Standard Ethernet Network Interface
en3 Available 12-10 Standard Ethernet Network Interface
en4 Available 12-18 Standard Ethernet Network Interface
en5 Defined   14-08 Standard Ethernet Network Interface
en6 Available       Standard Ethernet Network Interface
```

```
neo.dfw.ibm.com /#netstat -in
Name  Mtu   Network     Address            Ipkts Ierrs    Opkts Oerrs  Coll
en1   1500  link#2      0.9.6b.4e.70.a4    71007     0    52693     0     0
en1   1500  9.19.176    9.19.176.107       71007     0    52693     0     0
en4   1500  link#5      0.9.6b.4e.70.a7       53     0       63     0     0
en4   1500  1.1.1       1.1.1.1               53     0       63     0     0
en6   1500  link#7      0.9.6b.4e.70.a5     4824     0     4886     0     0
en6   1500  2.2.2       2.2.2.1             4824     0     4886     0     0
en6   1500  192.168.43  192.168.43.4        4824     0     4886     0     0
lo0   16896 link#1                         19432     0    20276     0     0
lo0   16896 127         127.0.0.1          19432     0    20276     0     0
lo0   16896 ::1                            19432     0    20276     0     0
```

Appendix 2 – Full netstat Command Output:

The `netstat -v ent6` command output is extremely useful during configuration, analysis, and troubleshooting stages. Unfortunately, it is also lengthy.

This is the complete output reported when we unplugged the ent3 adapter in our test environment. Key information is bolded in red. Notice that the main channel continues to run, but only the ent2 adapter is carrying the full load. The backup adapter is unused. During testing, we occasionally reset the statistics using the `entstat -r` command so we could more easily see the packet and byte count effects in the current test.

```
-------------------------------------------------------------
ETHERNET STATISTICS (ent6) :
Device Type: EtherChannel
Hardware Address: 00:09:6b:4e:70:a5
Elapsed Time: 0 days 1 hours 34 minutes 19 seconds

Transmit Statistics:                          Receive Statistics:
--------------------                          -------------------
Packets: 4403                                 Packets: 4352
Bytes: 754772                                 Bytes: 1653330
Interrupts: 10                                Interrupts: 4331
Transmit Errors: 0                            Receive Errors: 0
Packets Dropped: 0                            Packets Dropped: 0
                                              Bad Packets: 0
Max Packets on S/W Transmit Queue: 32
S/W Transmit Queue Overflow: 0
Current S/W+H/W Transmit Queue Length: 2

Elapsed Time: 0 days 0 hours 0 minutes 0 seconds
Broadcast Packets: 247                        Broadcast Packets: 165
Multicast Packets: 0                          Multicast Packets: 0
No Carrier Sense: 0                           CRC Errors: 0
DMA Underrun: 0                               DMA Overrun: 0
Lost CTS Errors: 0                            Alignment Errors: 0
Max Collision Errors: 0                       No Resource Errors: 0
Late Collision Errors: 0                      Receive Collision Errors: 0
Deferred: 0                                   Packet Too Short Errors: 0
SQE Test: 0                                   Packet Too Long Errors: 0
Timeout Errors: 0                             Packets Discarded by Adapter: 0
Single Collision Count: 0                     Receiver Start Count: 0
Multiple Collision Count: 0
Current HW Transmit Queue Length: 2

General Statistics:
-------------------
No mbuf Errors: 0
Adapter Reset Count: 0
Adapter Data Rate: 200
Driver Flags: Up Broadcast Running
        Simplex 64BitSupport PrivateSegment
        DataRateSet

=============================================================
=============================================================
```

```
Statistics for every adapter in the EtherChannel:
-------------------------------------------------

Number of adapters: 3
Active channel: primary channel
Operating mode: Round-robin mode

------------------------------------------------------------

ETHERNET STATISTICS (ent2) :
Device Type: IBM 4-Port 10/100 Base-TX Ethernet PCI Adapter (23100020)
Hardware Address: 00:09:6b:4e:70:a5

Transmit Statistics:                          Receive Statistics:
--------------------                          -------------------
Packets: 2556                                 Packets: 2527
Bytes: 416299                                 Bytes: 877793
Interrupts: 4                                 Interrupts: 2514
Transmit Errors: 0                            Receive Errors: 0
Packets Dropped: 0                            Packets Dropped: 0
                                              Bad Packets: 0
Max Packets on S/W Transmit Queue: 16
S/W Transmit Queue Overflow: 0
Current S/W+H/W Transmit Queue Length: 1

Broadcast Packets: 120                        Broadcast Packets: 88
Multicast Packets: 0                          Multicast Packets: 0
No Carrier Sense: 0                           CRC Errors: 0
DMA Underrun: 0                               DMA Overrun: 0
Lost CTS Errors: 0                            Alignment Errors: 0
Max Collision Errors: 0                       No Resource Errors: 0
Late Collision Errors: 0                      Receive Collision Errors: 0
Deferred: 0                                   Packet Too Short Errors: 0
SQE Test: 0                                   Packet Too Long Errors: 0
Timeout Errors: 0                             Packets Discarded by Adapter: 0
Single Collision Count: 0                     Receiver Start Count: 0
Multiple Collision Count: 0
Current HW Transmit Queue Length: 1

General Statistics:
-------------------
No mbuf Errors: 0
Adapter Reset Count: 0
Adapter Data Rate: 200
Driver Flags: Up Broadcast Running
        Simplex AlternateAddress 64BitSupport
        PrivateSegment DataRateSet

IBM 4-Port 10/100 Base-TX Ethernet PCI Adapter Specific Statistics:
------------------------------------------------
Chip Version: 26
RJ45 Port Link Status : up
Media Speed Selected: 100 Mbps Full Duplex
Media Speed Running: 100 Mbps Full Duplex
Receive Pool Buffer Size: 384
Free Receive Pool Buffers: 128
```

```
No Receive Pool Buffer Errors: 0
Inter Packet Gap: 96
Adapter Restarts due to IOCTL commands: 0
Packets with Transmit collisions:
 1 collisions: 0           6 collisions: 0          11 collisions: 0
 2 collisions: 0           7 collisions: 0          12 collisions: 0
 3 collisions: 0           8 collisions: 0          13 collisions: 0
 4 collisions: 0           9 collisions: 0          14 collisions: 0
 5 collisions: 0          10 collisions: 0          15 collisions: 0
Excessive deferral errors: 0x0

-------------------------------------------------------------

ETHERNET STATISTICS (ent3) :
Device Type: IBM 4-Port 10/100 Base-TX Ethernet PCI Adapter (23100020)
Hardware Address: 00:09:6b:4e:70:a5

Transmit Statistics:                          Receive Statistics:
--------------------                          -------------------
Packets: 1847                                 Packets: 1825
Bytes: 338473                                 Bytes: 775537
Interrupts: 5                                 Interrupts: 1817
Transmit Errors: 0                            Receive Errors: 0
Packets Dropped: 0                            Packets Dropped: 0
                                              Bad Packets: 0
Max Packets on S/W Transmit Queue: 16
S/W Transmit Queue Overflow: 0
Current S/W+H/W Transmit Queue Length: 1

Broadcast Packets: 127                        Broadcast Packets: 77
Multicast Packets: 0                          Multicast Packets: 0
No Carrier Sense: 0                           CRC Errors: 0
DMA Underrun: 0                               DMA Overrun: 0
Lost CTS Errors: 0                            Alignment Errors: 0
Max Collision Errors: 0                       No Resource Errors: 0
Late Collision Errors: 0                      Receive Collision Errors: 0
Deferred: 0                                   Packet Too Short Errors: 0
SQE Test: 0                                   Packet Too Long Errors: 0
Timeout Errors: 0                             Packets Discarded by Adapter: 0
Single Collision Count: 0                     Receiver Start Count: 0
Multiple Collision Count: 0
Current HW Transmit Queue Length: 1

General Statistics:
-------------------
No mbuf Errors: 0
Adapter Reset Count: 0
Adapter Data Rate: 200
Driver Flags: Up Broadcast Simplex
        Limbo AlternateAddress 64BitSupport
        PrivateSegment DataRateSet

IBM 4-Port 10/100 Base-TX Ethernet PCI Adapter Specific Statistics:
------------------------------------------------
Chip Version: 26
RJ45 Port Link Status : down
Media Speed Selected: 100 Mbps Full Duplex
```

```
Media Speed Running: Unknown
Receive Pool Buffer Size: 384
Free Receive Pool Buffers: 128
No Receive Pool Buffer Errors: 0
Inter Packet Gap: 96
Adapter Restarts due to IOCTL commands: 0
Packets with Transmit collisions:
 1 collisions: 0           6 collisions: 0          11 collisions: 0
 2 collisions: 0           7 collisions: 0          12 collisions: 0
 3 collisions: 0           8 collisions: 0          13 collisions: 0
 4 collisions: 0           9 collisions: 0          14 collisions: 0
 5 collisions: 0          10 collisions: 0          15 collisions: 0
Excessive deferral errors: 0x0

-------------------------------------------------------------
Backup adapter - ent0:
======================

ETHERNET STATISTICS (ent0) :
Device Type: 10/100 Mbps Ethernet PCI Adapter II (1410ff01)
Hardware Address: 00:09:6b:4e:70:a5

Transmit Statistics:                          Receive Statistics:
--------------------                          -------------------
Packets: 0                                    Packets: 0
Bytes: 0                                      Bytes: 0
Interrupts: 1                                 Interrupts: 0
Transmit Errors: 0                            Receive Errors: 0
Packets Dropped: 0                            Packets Dropped: 0
                                              Bad Packets: 0
Max Packets on S/W Transmit Queue: 0
S/W Transmit Queue Overflow: 0
Current S/W+H/W Transmit Queue Length: 0

Broadcast Packets: 0                          Broadcast Packets: 0
Multicast Packets: 0                          Multicast Packets: 0
No Carrier Sense: 0                           CRC Errors: 0
DMA Underrun: 0                               DMA Overrun: 0
Lost CTS Errors: 0                            Alignment Errors: 0
Max Collision Errors: 0                       No Resource Errors: 0
Late Collision Errors: 0                      Receive Collision Errors: 0
Deferred: 0                                   Packet Too Short Errors: 0
SQE Test: 0                                   Packet Too Long Errors: 0
Timeout Errors: 0                             Packets Discarded by Adapter: 0
Single Collision Count: 0                     Receiver Start Count: 0
Multiple Collision Count: 0
Current HW Transmit Queue Length: 0

General Statistics:
-------------------
No mbuf Errors: 0
Adapter Reset Count: 1
Adapter Data Rate: 200
Driver Flags: Up Broadcast Running
        Simplex AlternateAddress 64BitSupport
        ChecksumOffload PrivateSegment DataRateSet
```

```
10/100 Mbps Ethernet PCI Adapter II (1410ff01) Specific Statistics:
--------------------------------------------------------------------
Link Status : up
Media Speed Selected: 100 Mbps Full Duplex
Media Speed Running: 100 Mbps Full Duplex
Receive Pool Buffer Size: 1024
Free Receive Pool Buffers: 1024
No Receive Pool Buffer Errors: 0
Receive Buffer Too Small Errors: 0
Entries to transmit timeout routine: 0
Transmit IPsec packets: 0
Transmit IPsec packets dropped: 0
Receive IPsec packets: 0
Receive IPsec packets dropped: 0
Inbound IPsec SA offload count: 0
Transmit Large Send packets: 0
Transmit Large Send packets dropped: 0
Packets with Transmit collisions:
 1 collisions: 0           6 collisions: 0          11 collisions: 0
 2 collisions: 0           7 collisions: 0          12 collisions: 0
 3 collisions: 0           8 collisions: 0          13 collisions: 0
 4 collisions: 0           9 collisions: 0          14 collisions: 0
 5 collisions: 0          10 collisions: 0          15 collisions: 0
```

References:

Additional information is available from these publications.

- **HACMP Anouncement of Support for EtherChannel in AIX V 5.1 and 5.2** Flash 10284
  http://www.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/FLASH10284

- **AIX 5L Version 5.3 System Management Guide: Communications and Networks** (TCP/IP section)
  http://publib16.boulder.ibm.com/doc_link/en_US/a_doc_lib/aixbman/commadmn/commadmntfrm.htm

- **Crossover Cable for 10/100/Gigabit Ethernet Operations on pSeries RS/6000 Systems** technical document 101802
  http://www.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/TD101802

- **AIX EtherChannel Load Balancing Options** technical document 101260
  http://www.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/TD101260

These information respositories are available to users within the IBM corporate network.

- **Advanced Technical Support website -- Networking topic** (on IBM intranet only)
  http://w3-1.ibm.com/support/americas/pseries/network.html

- **Advanced Technical Support website -- High Availability topic** (on IBM intranet only)
  http://w3-1.ibm.com/support/americas/pseries/high_availability.html