How High Availability Works And Who Implements What

James Bottomley
SteelEye Technology

LinuxWorld San Francisco 2003

Cluster Types

- The rule of building HA clustering software is that anything beyond two nodes is hard.
- HA Clusters attempt to work with commodity.
- Thus, The world is essentially divided into three types of cluster
  - Resource Driven
  - Quorate
  - Two Node Only

Two Node Only

- Clusters which were designed with two nodes as the fundamental limit
  - Distinct from clusters which may operate with 2 or more nodes
- Large numbers of simplifying assumptions may be made in the design
- Major benefit is simplicity
  - "Complexity is the enemy of HA"

Quorate Clusters

- Quorate clusters are centrally controlled
  - Analagous to single CPU controlled by 1 clock
  - Cluster must form first before actions taken
  - Cluster directs all actions based on its controlling view of the cluster membership
  - Membership must be well defined
  - Actions generally agreed to by all cluster members (single cluster view)
  - Only a single cluster entity may exist at one time

Resource Driven Clusters

- Resource driven clusters are more chaotic
  - Act like Asynchronous CPU designs (actions trickle through instead of being co−ordinated centrally)
  - There is no central controlling cluster
  - Actions controlled for a given resource by cluster member who "owns" the resource
  - Other member acquiescence to actions by owning node not required

Resource Driven Clusters (2)

- Resource driver clusters (continued)
  - No central cluster means no monotonic instance numbers
  - Cluster may form with partial communications
  - Multiple resources => multiple owning nodes each of which may take an action simultaneously
  - Multiple independent sub−clusters may form

Why Choose Two Node Only?

- Simplicity
  - less to go wrong, therefore should be more robust.
  - Cheaper (costs more to build and test >2 node clusters).
  - Maybe you have a single application that will **never** need to scale beyond two nodes
- Simple transactional websites, dual redundant fileservers.

Why Choose more than Two Nodes?

- Need the implementation complexity
  - More than one application
  - Need better control over the location of cluster resources to maximise operational efficiency.
  - May want to add more servers later to smooth operations or spread the load.
- Need Protection from Cascade Failures
  - Things tend to fail in groups.
  - Such a grouping of failures is called a cascade.

Why Quorate?

- It’s an extremely old, tried and tested technology.
  - Used by the VAX (paragon of clustering virtues)
- Cluster failure modes are easy to predict and to analyse.
- It’s centrally controlled which is often seen as an advantage in clustering philosophies.

Why Resource Driven?

- Easier to design and build (no central control layer need be constructed)
  - Simplicity is desirable in HA (less to go wrong)
- Better scaling properties (in large clusters with large numbers of resources)
- Better disaster survivability (formation of multiple sub clusters usually gives better recovery characteristics)

Problems with Resource Driven Clusters

- Harder to analyse.
  - Chaotic behaviour makes provability difficult.
  - Disliked by acadaemia for this reason.
- Multi−threaded failover characteristics may cause OS resource problems.
- Single cluster view hard to obtain
  - makes administration difficult
- Definitely not like the good old VAX

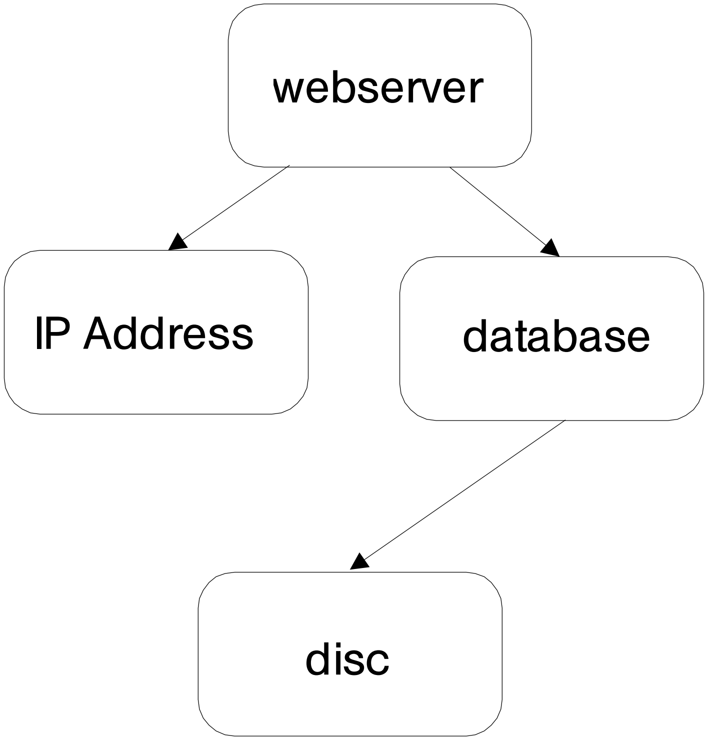
Elements of a Two Node Only Cluster

- Each node watches the other.
- When a heartbeat fails, assume the other node is down
- Employ STONITH technology to prevent the "Split Brain" problem.
- That’s it...

Elements of a Quorate Cluster

- Every node communicates with every other
- Cluster is formed by a voting membership (which may include a tie breaker device)
- Cluster will form in Quorate Majority (Majority with minimum votes necessary to form a quorum)
- Cluster **cannot** form without a quorum
  - May mitigate this by giving the only counting vote to the tie breaker device.

Elements of Resource Driven Clusters



- Resources comprise Hierarchies
- Hierarchies are fundamental units
- At least one resource of a hierarchy must be ownable.
- arrows represent dependencies

Resource Ownership Properties

- Classes of resources are ownable
- Ownability implies two properties
  - May I own (i.e. test of ownership)
  - Take ownership (must be exclusive)
- Disk resources implement ownability usually with reservations
- May also introduce ownership carrying resources (similar to a quorum disc)

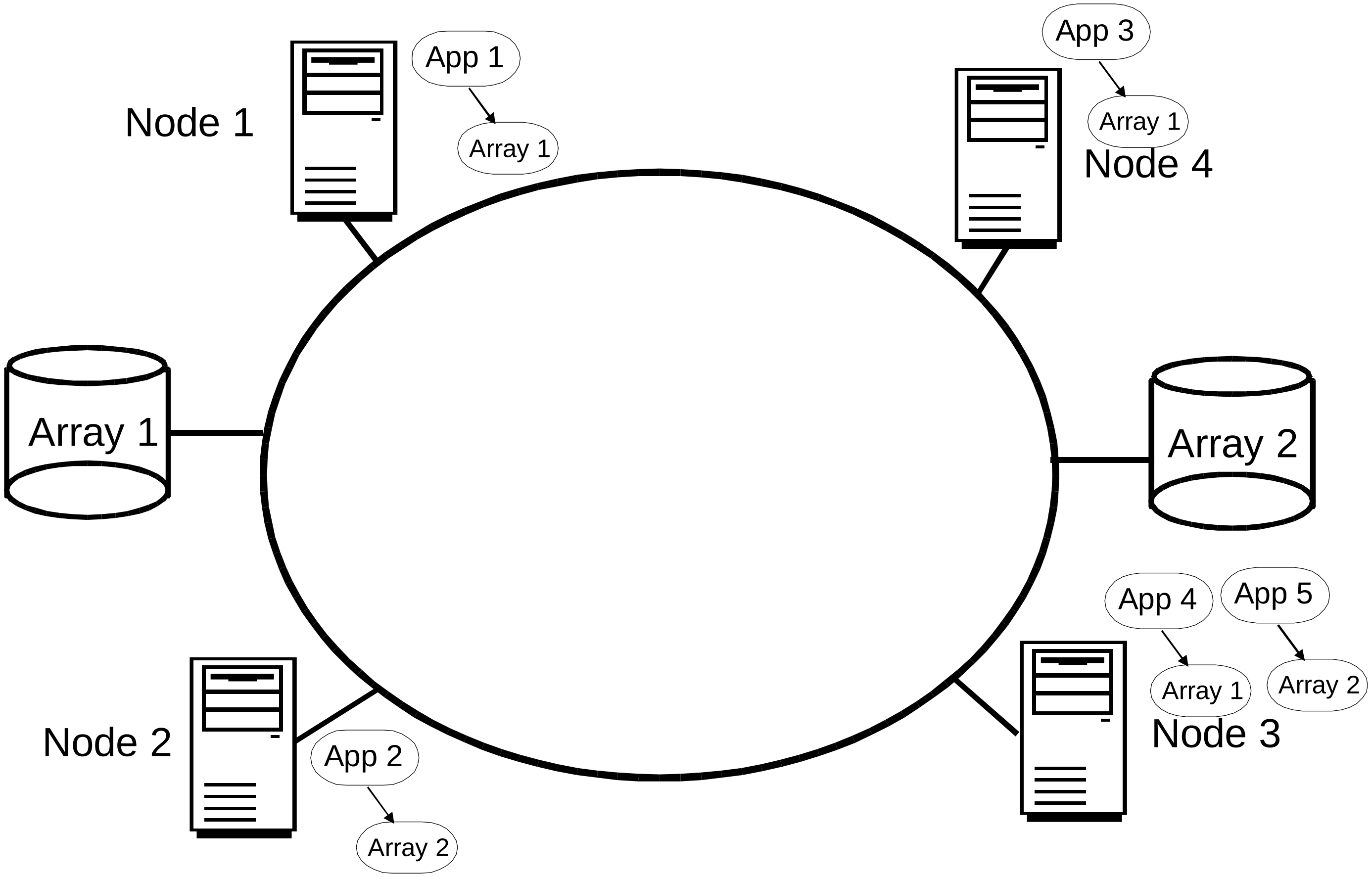
SCSI Reservations

- Tailor made for resource ownership
- Reservation will enforce exclusive access to the owning node. Another node may not accidentally or maliciously interfere with the data
- Ownership is at the disc level, not the partition level (multiple partitions move together)
- Reservations can cause OS problems (i.e. can’t read the partition table)

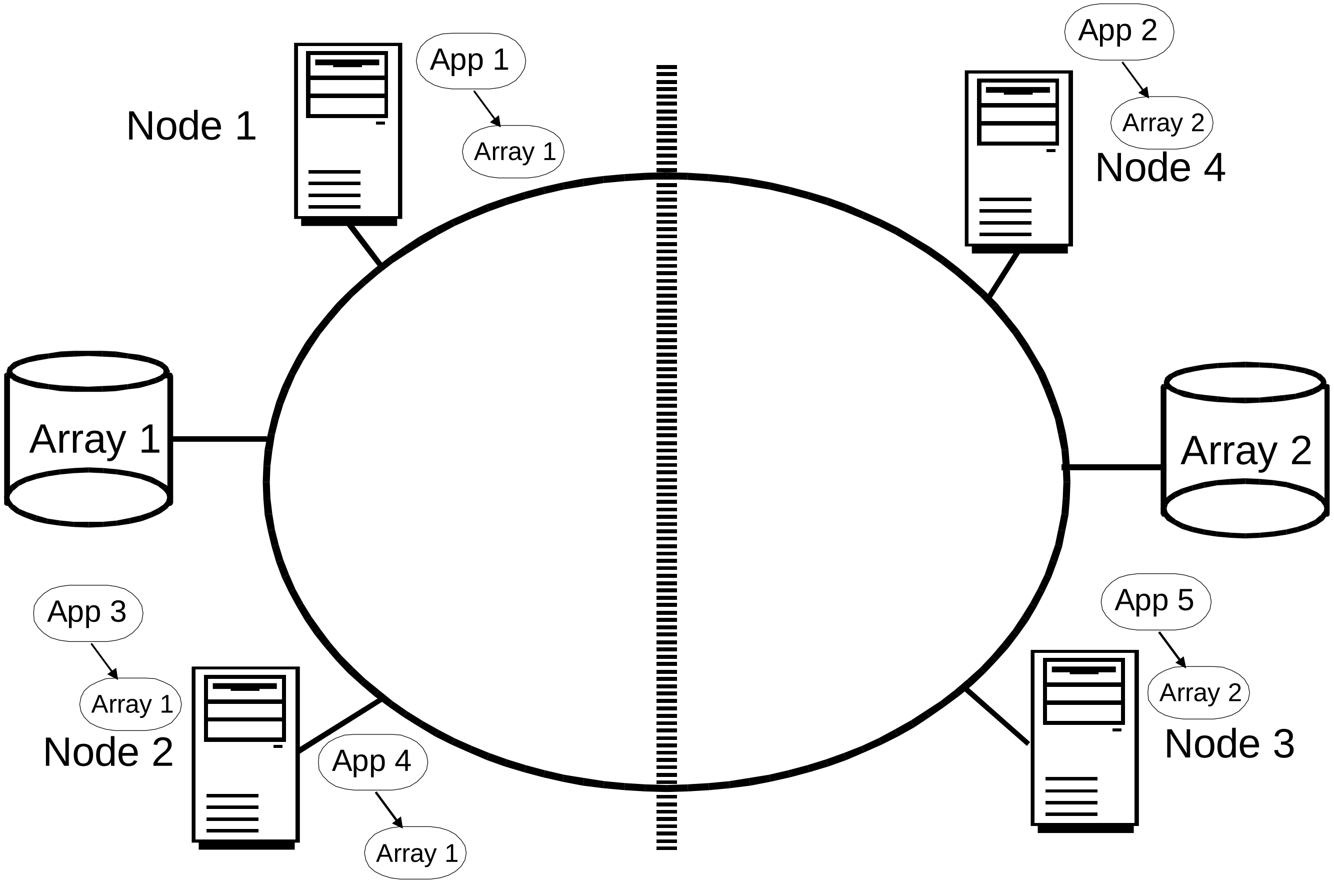
Hierarchy Ownership

- Nodes own the hierarchy
- To own a hierarchy, a node must own all of its ownable resources.
- To prevent ownership deadlock, hierarchies need a deterministic ownership acquisition ordering.
- As soon as a node owns a hierarchy, it may proceed to recover that hierarchy regardless of what is going on in the cluster.

Cluster Partition Illustration



Cluster Partition Illustration (2)



Types Of Available HA Clusters for Linux

- There are a wide range available, both open source and proprietary.
- All cluster types are represented
- Open Source clusters often come with support as a purchaseable option
- Choose the cluster that is right for you
  - Initial Cluster Planning is the most necessary and most often neglected aspect of HA.

Red Hat/Mission Critical

- Two Node only
- Simple script driven interface
- Configuration options tend to be slightly inflexible
- No data replication or shared host based RAID
- Protects:
  - NFS, Samba, Apache, Oracle, MySQL, lp

Veritas Cluster Manager

- Resource Driven Cluster (needs VxVM for ownership model)
- Full Ease of Use Java based GUI
- Replication not (currently) available on Linux. No Host Based RAID support
- Protects:
  - MySql, Apache
  - More Application protection for Linux is in the works.

SteelEye LifeKeeper

- Resource Driven Cluster. Fencing done via SCSI reservations. Also includes STONITH support.
- Full Ease of Use Java based GUI
- Has replication and Host Based RAID support
- Protects:
  - NFS, Samba, MySQL, Oracle, Informix, DB2, Apache, SAP/R3, sendmail/SAMS, lp, generic applications, SDK

SGI Failsafe

- Quorate Cluster. Support for STONITH; Open Source
- Ease of Use web based GUI
- Supports replication (via drbd). No current support for Host Based RAID.
- Not currently under active development
- Protects
  - NFS, Samba, Apache, Oracle, DB2, SAP/R3

IBM Tivoli Clusters

- Quorate (hybrid) based
- Ease of use Web Based GUI
- No current support for either replication (except SAP/R3 replicated enqueue) or Host Based Raid
- Protects:
  - SAP/R3

Heartbeat

- Two Node only; Open Source
- CLI Configuration
- Replication (DRBD) and Host Based Raid Support
- Protects
  - NFS, Samba, Apache, Databases, sendmail, generic applications

Legato/Automated Availability Manager

- Multi node, neither Quorate nor Resource Driven.
- Centralised Management Console GUI
- No replication (execpt SRDF). No Host Based Raid
- Protects
  - NFS, Apache, Oracle, Sybase, Informix, generic services, Checkpoint Firewall

MC/ServiceGuard

- Quorate Cluster (2 node cluster requires extra machine for quorum service)
- Ease of Use GUI
- No Replication, supports Host Based Raid
- Protects:
  - NFS, Samba, Apache, sendmail

Polyserve Matrix Server

- Different Paradigm: Parallel Active (actually moving towards SSI like Mosix).
- Based around home grown cluster filesystem (and DLM).
  - Others available: Lustre, GFS etc.
- Rely on applications modified to be parallel active (e.g. like Oracle RAC).
- However, high barrier to producing "correct" parallel active applications.

Cluster File Systems

- Operation essentially similar to NFS
  - Except that in most CFS implementations, nodes talk directly to the disc
- Very Hard to do correctly
- Applications still need to co−ordinate correctly to ensure correct operation.
- Vendors: Polyserve, Lustre, Sistina,...
- Usually based on a DLM

Distributed Lock Managers

- Provide a cluster wide locking abstraction
- May also provide other facilities
  - Fast RPC (often via callback and notify)
  - Lock Information Blocks (for data exchange)
- All based to a certain extent on the original Oracle Lock Manager API
- Several now exist in Linux (ClusterFS, IBM etc).

Parallel Active Applications

- By and large, cannot be done unless the applications themselves co−operate
- Examples are Oracle RAC, OPS.
- However, can partition application namespaces up to give pseudo parallelism
- Sendmail for instance, uses the correct locking semantics to operate in a parallel environment.

High Performance Computing

- This is the other side of the Cluster coin
- HPC cluster farms tend to run multiple copies of the same application with slightly different data.
- Idea is to perform rapid calculations
- Data gathering back end is almost always a cluster filesystem
  - NFS was used a long time ago.

Conclusions

- Resource driven clusters are significantly different from Quorate ones
  - Both have advantages and disadvantages.
  - Correct choice depends on HA priorities.
- Resource driven clusters have greater flexibility and greater complexity
- Quorate clusters can be simpler but may have I/O fencing problems.