THE EXPERT'S VOICE® IN COMPUTER CLUSTER

Oracle Solaris and Veritas Cluster

An Easy-build Guide

Vijay Shankar Upreti

Apress®

Oracle Solaris and Veritas Cluster

An Easy-build Guide

Vijay Shankar Upreti

Apress®

Oracle Solaris and Veritas Cluster: An Easy-build Guide

ISBN-13 (pbk): 978-1-4842-1832-7

ISBN-13 (electronic): 978-1-4842-1833-4

Managing Director: Welmoed Spahr
Lead Editor: Pramila Balan
Editorial Board: Steve Anglin, Pramila Balan, Louise Corrigan, Jonathan Gennick, Robert Hutchinson, Celestin Suresh John, Michelle Lowman, James Markham, Susan McDermott, Matthew Moodie, Jeffrey Pepper, Douglas Pundick, Ben Renow-Clarke, Gwenan Spearing
Coordinating Editor: Prachi Mehta
Copy Editor: Karen Jameson
Compositor: SPi Global
Indexer: SPi Global
Artist: SPi Global

Distributed to the book trade worldwide by Springer Nature, 233 Spring Street, 6th Floor, New York, NY 10013. Phone 1-800-SPRINGER, fax (201) 348-4505, e-mail orders-ny@springer-sbm.com, or visit www.springeronline.com. Apress Media, LLC is a California LLC and the sole member (owner) is Springer Science + Business Media Finance Inc (SSBM Finance Inc). SSBM Finance Inc is a Delaware corporation.

For information on translations, please e-mail rights@apress.com, or visit www.apress.com.

Apress and friends of ED books may be purchased in bulk for academic, corporate, or promotional use. eBook versions and licenses are also available for most titles. For more information, reference our Special Bulk Sales–eBook Licensing web page at www.apress.com/bulk-sales.

Any source code or other supplementary materials referenced by the author in this text is available to readers at www.apress.com/9781484218327. For detailed information about how to locate your book's source code, go to www.apress.com/source-code/. Readers can also access source code at SpringerLink in the Supplementary Material section for each chapter.

Dedicated to my parents

Dr. Jaydutt Upreti

Smt Kamla Upreti

YOUR BLESSINGS ARE DIVINE POWER

Contents at a Glance

Contents

About the Author

Vijay S. Upreti, is science graduate, comes with nearly 20 years in field of IT. Started his career in 1996, being Systems administrator and rose to his last position worked as Principal Architect. Vijay, worked for Datapro Information Technology Ltd, Inter University Center for Astronomy and Astrophysics, Mahindra British Telecom (now TechMahindra), Tech Mahindra, Bulldog Broadband UK, Cable&Wireless Worldwide (now part of Vodafone) UK, Sun Microsystems India Pvt, Target Corporation India Pvt. Ltd and Wipro Technologies. Throughout his experience, Vijay was engaged in the IT Infrastructure strategies, planning, design, implementation and operational support activities at various levels in Unix and Linux technologies.

Currently Vijay is working as an Independent consultant for Datacenter and Cloud technologies.

Acknowledgments

First and foremost, I would like to thank all my peers, colleagues, juniors and my bosses in past, who encouraged me to pen the skill I gained through my experience of more than 19 years in the IT infrastructure domain. It has taken a while to complete the book, but the idea has always been to ensure that the book helps those who are novice to clustering and would like to get a better understanding from "Concepts to implementation and configuration level."

Working in past organizations like Datapro Information Technology Ltd Pune, IUCAA Pune - India, Mahindra British Telecom (Now Tech Mahindra) – India and UK, Bulldog Broadband – UK, Cable & Wireless UK, Sun Microsystems India Pvt Ltd, Target Corporation India and Wipro Technologies India has been learning at every stage. Sun Microsystems exposed me to a vital opportunities on Sun Solaris Technologies which helped me in acquiring required skills sets.

I had multiple opportunities to implement local and geographic cluster in my past organizations. I had the privilege to work and interact with some great minds and highly skilled teams and individuals. With specific names, a big thanks to Divya Oberoi, (Faculty member at National Center for Radio Astrophysics - TIFR, Pune), Fiaz Mir (Strategy Architect – Vodafone, UK), Dominic Bundy (IT Cloud specialist), Guido Previde Massara (Owner and Director of FooBar consulting Limited), Satyajit Tripathi (Manager Oracle India Pvt Ltd), Anitha Iyer (Sr. Director, Software Engineering at Symantec, India), Poornima Srinivasan, IT Consultant and Leader, Target Corporation India Pvt Ltd, Sanjay Rekhi (Senior Technology Executive). Friends have always been technically associated with me. I would like to personally quote few individuals here, Shirshendu Bhattacharya (Program Manager, Google), Arijit Dey (Sr. Architect, Wipro), Rangarajan Vasudeva (Lead Engineer, Wipro Technologies), Mahatma Reddy (Sr Engineer) and Gurubalan T (Sr. Engineer, Oracle India).

Blessings of my parents (Dr J. D Upreti and Mrs Kamla Upreti), my elder brothers (Mr Ravindra Upreti and Mr Sanjay Upreti) and sisters (Smt Gayatri Joshi and Smt Savitri Pant) and all other relatives for their sustained encouragement and support has been a big moral boost. My wife Dr. Lata Upreti and son Aniruddha Upreti have been a great support for me, and helped me in best way they could in their own capacity while writing this book. A big thanks to a beautiful family for their support and encouragement.

The book could not have been completed by the support of my reviewer Mr. Hemachandran Namachivayam, Principle Engineer at Oracle Corporation India. His vast knowledge in Cluster technologies has been quite helpful in completing the book.

And finally the publishing of this book would not have been possible without the help of Mr. Celestin Suresh John and Ms. Prachi Mehta, who had been a dedicated resource from Apress. They were a continuous source of encouragement, and assisted me in getting the book published in time. Sincerely acknowledge your efforts and assistance.

Introduction

What Is This Book About?

The book is focused on understanding high availability concepts: Oracle Solaris Cluster and Veritas Cluster framework, installation and configuration with some examples of setting up applications as high available, and providing a cheat sheet on the list of commands used for setting up a cluster. The book assists with setting up clustering on VirtualBox-based virtual. Easy steps mentioned in the book will help readers to have a basic level understanding of high availability, cluster, and setting up cluster environments. A quick note on Oracle Solaris and Veritas Clusters:

> Oracle Solaris Cluster is a high availability software, originally created by Sun Microsystems and acquired by Oracle Corporations in 2010. Oracle Solaris Cluster helps in building high available environments using Active-Passive application failover. Oracle Solaris Cluster also comes with Disaster Recovery-based Geographic clustering functionality cross-site/geography using a high speed DWDM network backbone along with storage replication. The Oracle Solaris Cluster environment can be used for any kind of application and Database supported to be running in Oracle Solaris environments, having Stop/Stop and Probe methods.
>
> Veritas Cluster software is High Availability Software provided by Veritas, which was later acquired by Symantec in 2005, known as Symantec Cluster. (https://www.symantec.com/en/in/cluster-server/), ensuring 24/7 high availability with minimal or no manual intervention. Likewise, any OS cluster solution providers, such as Symantec Cluster Server, also provides disaster recovery (DR) solutions. Veritas Cluster Server detects risks to application availability through monitoring triggers and ensures automated recovery of applications for high availability and disaster recovery. Veritas cluster also supports any kind of application that can be ported and configured under the given platform (e.g., Solaris, Linux, or Windows).

The book is aimed at providing elaborate steps on setting up virtual host-based lab clusters (Oracle Solaris and Veritas) on personal devices (Laptop or Desktop) for cluster installation, configuration, management, and administration. The book also covers some example functions of web- or java-based graphical user interface (GUI) for cluster management and administration.

Due to limitations of the Laptop/Desktop configuration, the setup is limited to only a two-node cluster build, although if configuration permits, you can add as many virtual hosts to the cluster using the steps mentioned below.

Who Will Get Help from the Book?

This book is targeted to anyone working in the IT Infrastructure domain with a basic understanding of the Unix/Solaris environment. Additionally the book will help engineers in IT Operations and support, who rarely get an opportunity to build live in OS clustered environments.

It is a single book that covers the basic high availability concepts, clustering framework, and components and installation and setup for both Oracle Solaris Cluster and Veritas Cluster. The easy steps provided here will surely help Systems Administrators and System Build Engineers to gain confidence in understanding availability and cluster build.

The easy writing as well as setup on the personal desktop/laptop will also help graduate students and those who would like to pursue careers in the field of IT Infrastructure domain. The real-life cluster setup simulated on the virtual environment is a starting point to gain a good level of cluster installation and configuration skills. The beginning of each chapter also helps readers understand the availability concepts for IT Infrastructure and Data Center.

How This Book Is Designed

The book is designed to first have a brief understanding of High Availability concepts and then elaborates on Oracle Solaris and Veritas Cluster framework, installation, and configuration steps, managing cluster resources using command line and GUI. The book also covers some more examples of cluster setup. The book ends with some frequently used command lines and their options for day-to-day cluster management.

Chapter 1: Availability Concepts The chapter covers the concepts of availability. It starts with the understanding of IT Infrastructure and Data Center, specific availability challenges, and how to address these challenges. It also discusses Operating System-level clustering concepts. And finally it covers how to understand the business value driven through availability.

Chapter 2: Cluster Introduction The chapter starts with a quick overview of cluster framework and then moves further on to Cluster Architecture. The chapter will detail Oracle and Veritas specific cluster frameworks.

Chapter 3: Cluster Build Preparations and Understanding VirtualBox The focus of this chapter is to describe preparation required for the cluster build. It details VirtualBox and its components. VirtualBox will be the virtual host solution as a part of a Lab exercise for setting up two-node clusters.

Chapter 4: Oracle Solaris Cluster Build This chapter is focused on the Oracle Cluster Build process, starting from planning to the cluster implementation. The chapter will cover prerequisites, installation, and configuration of two-node Oracle Solaris cluster environments built under VirtualBox.

Chapter 5: Setting Up Apache and NFS Cluster Data Services This chapter will cover adding two example applications – Apache and NFS – as a part of cluster failover services. The chapter will start with adding shared storages move further and with setting up metaset and mirroring of disks and then adding cluster resource groups and resources along with their dependencies.

Chapter 6: Veritas Clustering This chapter is about the Veritas Cluster and starts with the design phase of Veritas Cluster. The chapter then covers Veritas Cluster implementation, through cluster installation and setting up a Veritas Cluster framework.

Chapter 7: Setting Up Apache and NFS Services in Veritas Cluster Similar to Chapter 5 for setting up Apache and NFS applications in Oracle Solaris Cluster, this chapter also describes steps for bringing Apache and NFS applications under a cluster framework for high availability. Again, here it will first start with adding shared storage and configuring.

Chapter 8: Graphical User Interface for Cluster Management This chapter has information on Graphical User Interface as provided by both Oracle Solaris Cluster and Veritas Cluster. This chapter starts with initiating GUI, login, and then managing clusters through GUI. In addition to showing components of GUI, there are some examples of switching over cluster resource groups.

Chapter 9: Additional Examples – Cluster Configurations This chapter will have some more command-line examples of cluster configuration. It starts with steps for setting up Oracle Solaris Geographic Cluster. There are also examples of setting up NFS service using ZFS filesystems and zone clusters under Oracle Solaris Cluster build. And the last example in this chapter is about adding a customer build application to Veritas and Oracle Solaris clusters.

Chapter 10: Command-Line Cheat Sheet This is the last chapter of the book, covering some commonly used command lines for both Oracle Solaris Cluster and Veritas Cluster setup, administration, and other operational activities.

CHAPTER 1

Availability Concepts

Availability

Availability is the time that business services are operational. In other words, availability is a synonym of business continuity. And when it comes to IT Infrastructure, systems, site, or data center, availability reflects the span of time applications, servers, site, or any other infrastructure components that are up and running and providing value to the business.

The focus of this book is based on the IT Infrastructure and data center availability leading to business uptime.

IT Infrastructure availability ensures that its components are fault tolerant, resilient, and provide high availability and reliability within site and disaster recovery cross-site/data centers in order to minimize the impact of planned or unplanned outages. Reliability ensures systems have longer lives to be up and running, and redundancy helps ensure system reliability.

Expectations on high availability may vary depending upon the service level being adopted and agreed upon with the business stakeholders. And in terms of disaster recovery, the availability is the reflection of time to return to normal business services, either through failover of services to the disaster recovery site or through restoration from backup, ensuring minimal data loss and reduced time of recovery.

Availability Challenges

Having observed some major disasters in the IT Infrastructure and data center space, availability is a top priority for business continuity. We need to understand the challenges introduced due to the availability aspect of IT Infrastructure and/or data centers.

But before even starting to understand the challenges in managing and supporting IT Infrastructure specific to server availability, let's first understand the backbone of IT Infrastructure, which is a data center and its primary purpose.

> Data Center is a collection of server/computers connected using network and storage devices controlled by software and aims to provide seamless business functioning. A data center is the brain and also one of the sources of energy to drive the business faster with the least or no interruption.
>
> In the early age of data center design, there were computer rooms; the computers were built in a room dedicated to IT. With the rapid growth of IT requirements, risks on the data integrity and security, modularity, availability, and scalability challenges, the data centers were moved out of company premises to a dedicated site; and they kept network, security, and storage as different dedicated components of Data Center Management.

So what are the data center availability challenges? Here are some of the critical data center measures that lead to availability challenges.

At the very beginning, lack of planning for the selection of an IT site or data center site could have severe availability impact due to natural disasters such as floods, earthquakes, tsunamis, etc. Similarly unplanned floor design may lead to a disaster due to flood or any other water clogging.

Not having sufficient fire protection and policies is another big challenge to the availability and may lead to data center disasters. The next challenge is insufficient physical security, such as not having Photo Identity, absence of CCTV, not maintaining visitors' log registers, etc.

Hardware devices, such as Servers, Storages, and Network and their components installed with no redundancy will have an immediate impact on availability. Most of the traditional data centers suffer the disruption of services due to non-redundant devices.

Electrical power is one of the most critical parts of data center uptime. Not having dual power feeds, backup powers supplies, or green energy power sources are some of the critical areas of challenges for availability. Similarly, absence of temperature control in the data center will have a disastrous impact on running high-computing temperature-sensitive devices.

Unskilled or not, having the right level of skill and untrained technical resources are big contributors to risks on data center availability.

Not having monitoring solutions at different levels such as heating and cooling measurements of the data center to the monitoring of applications, servers, network- and storage-based devices will have direct impacts on the host availability. Also, having no defined and applied backup and restoration policy will have an impact on the host availability.

Absence of host security measures in terms of hardware and software security policies is another challenge to availability.

Addressing Availability Challenges

As discussed in the previous section, availability is the most important factor that ensures application and business uptime. To address the challenges posed to the data center availability, each of the above challenges, be it site selection, floor planning, fire safety, power supply redundancy, servers/network and storage-level redundancy, physical security, Operating System-level security, network security, resource skills, and monitoring solutions should be addressed and resolved.

When it comes to server availability, it reflects hardware redundancy and software-based high availability. In other words, redundant hardware components along with software-driven fault tolerant solutions ensure server availability. Figure 1-1 further gives detailed components impacting and contributing to the server availability.

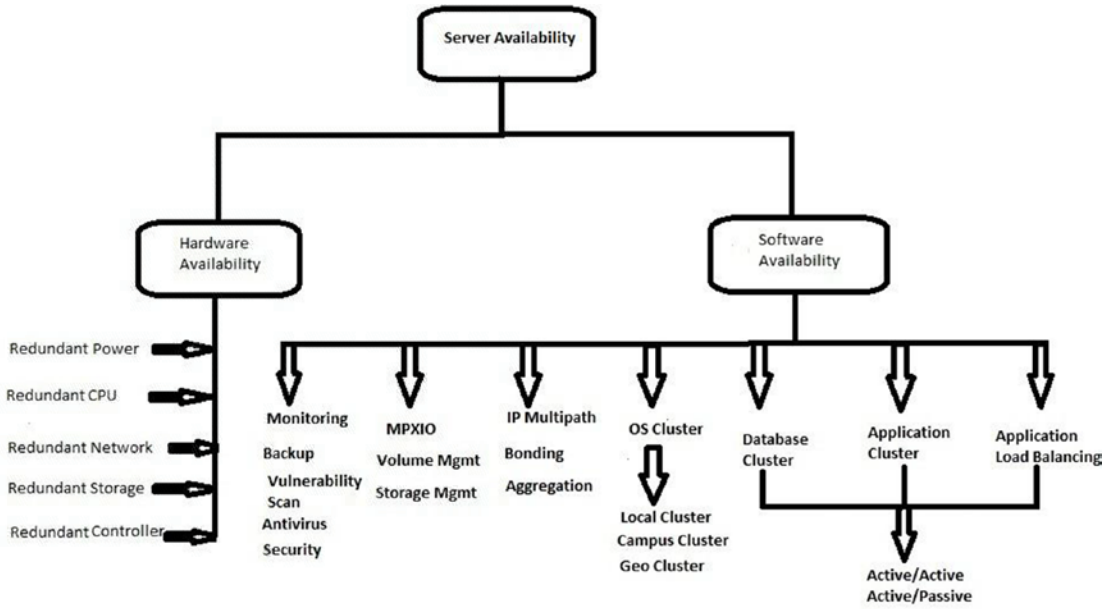

Figure 1-1. *High Availability Components*

As explained above, Server Availability is mainly covered as hardware and software availability. On hardware availability, introducing redundant hardware for critical components of the server will provide high resilience and increased availability.

The most critical part of addressing availability challenges is to ensure the server has redundant power points. Power supplies should be driven from two difference sources of power generators and with a minimal time to switch. Battery backup/generator power backups also ensure minimal or no server downtime.

The next important component for host availability is the CPU, and having more than one CPU can ensure, in the case of failure, that one of the processors that processes is failed over to a working CPU.

Next in the line addressing server availability challenges is to have redundant network components, both at switch level and server-level port redundancy. The network redundancy is later coupled with software-based network redundancy such as port aggregation or IP Multipathing or Interface Bonding to ensure network interface failures are handled well with very insignificant or no impact on the application availability.

Storage-based redundancy is achieved at two stages. One is at the local storage, by using either hardware- or software-based raid configuration; and second, using SAN- or NAS-based storage LUNs, which are configured for high availability. OS-based MPXIO software ensures the multipath.

On a larger scale for the data center (DC) availability, based on business availability requirements, data centers are classified and certified in four categories:

Tier I – Single non-redundant distribution paths; 99.671% availability allows 28.817 hours of downtime.

Tier II – Along with tier I capabilities, redundant site capacity, with 99.741% availability and allows 22.688 hours of downtime.

Tier III – Along with tier II capabilities, Multiple distribution availability paths (dual powered), with 99.982% availability and allows 1.5768 hours of downtime.

Tier IV – Along with tier III capabilities, HVAC systems are multipowered, with 99.995% availability and 26.28 minutes or .438 hours of unavailability.

And finally, when it comes to host monitoring (hardware, OS, and application), it should also be integrated with the event generators and events further categorized into different service levels based on the criticality of the event generated, ensuring minimum disruption to the services.

OS Clustering Concepts

Operating System cluster architecture is a solution driven by highly available and scalable systems ensuring application and business uptime, data integrity, and increased performance by performing concurrent processing.

Clustering improves the system's availability to end users, and overall tolerance to faults and component failures. In case of failure, a cluster in action ensures failed server applications are automatically shut down and switched over to the other servers. Typically the definition of cluster is merely a collection/group of object, but on the server technology, clustering is further expanded to server availability and addressing fault tolerance.

Clustering is defined as below:

"Cluster analysis or clustering is the task of grouping a set of objects in such a way that objects in the same group (called a cluster) are more similar (in some sense or another) to each other than to those in other groups (clusters)." – *Wikipedia* definition

"Group of independent servers (usually in close proximity to one another) interconnected through a dedicated network to work as one centralized data processing resource. Clusters are capable of performing multiple complex instructions by distributing workload across all connected servers." – *Business Dictionary* (Read more: `http://www.businessdictionary.com/definition/cluster.html#ixzz3dmQvJEdk`)

OS-based cluster is classified based on the location of the cluster nodes installed. They are Local Cluster, Campus Cluster, Metropolitan Cluster, and Geographic Cluster.

> Local cluster is the cluster environment built in the same Data Center Site. In this configuration disks are shared across all cluster nodes for failover.
>
> Campus Cluster, also known as stretched cluster, is stretched across two or more Data Centers within the limited range of distance, and mirrored storages are visible across all other cluster nodes such as local cluster configuration.
>
> Metropolitan cluster setup with Metropolitan area, still limited by distances. The setup for Metropolitan cluster is similar to the one in Campus cluster, except that storage is set up for replication instead of disk mirroring, using Storage remote replication technologies. Metropolitan provides disaster recover in addition to the local high availability.
>
> Geographic cluster, which is ideally configured for a dedicated disaster recovery cluster configuration, is set up across two or more data centers geographically separated.

Local and campus clusters are set up with similar cluster configurations and using the same cluster software, although Metropolitan and Geographic clusters use a different software-based configuration, network configuration, and storage configuration (like replication True Copy, SRDF, etc.).

Business Value of Availability

Nonavailability of IT services has a direct impact on business value. Disruption to the business continuity impacts end users, client base, stakeholders, brand image, company's share value, and revenue growth. A highly available IT infrastructure and data center backbone of an enterprise caters to the bulk needs of customer requirements, reduces the time to respond to customers' needs, and helps in product automation – ultimately helping in revenue growth.

Below are some examples of incidents of data center disasters due to fire, flood, or lack of security that caused systems and services unavailability for hours and resulted in a heavy loss of revenues and a severe impact on the company's brand image.

On April 20, 2014, a fire broke out in in the Samsung SDS data center housed in the building. The fire caused users of Samsung devices users, including smartphones, tablets, and smart TVs, to lose access to data they may have been trying to retrieve. (`http://www.Data Centerknowledge.com/archives/2014/04/20/data-center-fire-leads-outage-samsung-devices/`)

Heavy rain broke a ceiling panel in Datacom's (an Australian IT solutions company), colocation facility in 2010, destroying storage area networks, servers, and routers belonging to its customers. On September 9, 2009, an enormous flash flood hit Istanbul. Vodafone's data center was destroyed.

The addition of a single second adjustment applied to UT (Universal Time) due to Earth's rotation speed to the world's atomic clocks caused problems for a number of IT systems in 2012, when several popular web sites, including LinkedIn, Reddit, Mozilla, and The Pirate Bay, went down.

In 2007, NaviSite (now owned by Time Warner) was moving customer accounts from Alabanza's main data center in Baltimore to a facility in Andover, Massachusetts. They literally unplugged the servers, put them on a truck, and drove the servers for over 420 miles. Many web sites hosted by Alabanza were reportedly offline for as long as the drive and reinstallation work took to complete.

A few years ago there was a data center robbery in London. UK newspapers reported an "Ocean's Eleven"-type heist with the thieves taking more than $4 million of equipment.

Target Corporation, December 2013: 70 million customers' credit/debit card information was breached, amounting to a loss of about $162 million.

It is quite obvious that any impact on the availability of the data center components such as server, network, or storage will have a direct impact on the customer's business.

CHAPTER 2

Cluster Introduction: Architecture for Oracle Solaris Cluster and Veritas Cluster

Introduction to Cluster Framework

As defined in the previous chapter, clustering enables multiple servers to be treated as a part of single cluster of servers and provides high availability solutions to the application environment. Clustering can help in increased performance and ensures greater scalability and reduced costs through optimized utilization of participant servers. The backbone of cluster lies on the server, network, and storage capacities along with the operating system-level clustering software environment. A cluster framework ensures applications and/or database environments start with defined dependencies and ensures successful failover or failback initiated by monitoring triggers or through manual intervention. Cluster framework also ensures an easy scalability through addition or removal of servers, known as cluster nodes, to the cluster.

OS Clustering Architecture

A cluster architecture is a group of two or more computers/server/hosts/nodes built with fault tolerant components and connected using network, storage, and software components to make a single virtual or logical server, available anytime in either of the physical computer/server/hosts or nodes. On the other view, cluster framework is an architecture that allows systems to combine together as an aggregated host, a way of harnessing of multiple processors to work in parallel, sharing memory across multiple hosts So cluster is not just a solution to availability in the form of failover and fault tolerant services but also stretches to scalability and performance.

A high-level two-node cluster architecture is explained in Figure 2-1.

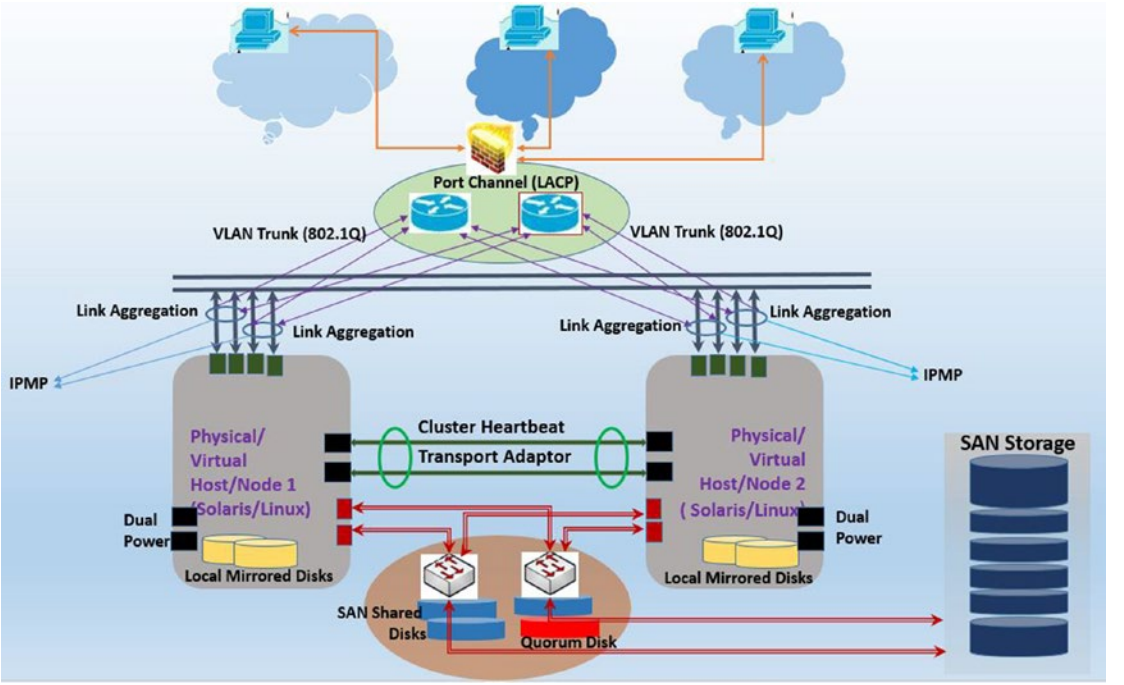

Figure 2-1. *Cluster Framework*

As shown in the above diagram, the basic design of a cluster framework consists of two or more cluster hosts/nodes – physical or virtual, Operating System, supported cluster software, network, and shared storage (SAN or NAS shared storage). The figure also shows some other aspects: redundancy design to support the cluster framework (like dual power supplies, Link Aggregation, and IPMP configuration).

Cluster Network

Network configurations should be set up with both server-level network card redundancy as well as having a redundant switch for failover. Network ports be configured to provide maximum redundancy allowing network survival due to the failure of the card at the server end or a failure of the switch.

A core component of cluster configuration is cluster heartbeat (transport interconnect). Cluster heartbeat plays a critical role for keeping a cluster alive. Heartbeat interconnects are used for pinging each cluster node, ensuring they are part of the cluster.

The other network component of cluster configuration is the core network (or company's internal network) that is used for communication between cluster resources (e.g., across applications or communications from application to database).

And the last component of the cluster network configuration is the communication to or from end users connecting from a public network to the back-end application under cluster configuration.

Redundant cluster network configuration is done differently for different kinds of configuration requirements. One of the configurations is to aggregate multiple network ports at the host level. The technology used for aggregating ports is either link aggregation (UNIX) or interface bonding (Linux) or NIC teaming. Link aggregation or interface bonding provides network redundancy/load balancing as well as providing combined bandwidth to the virtual network pipe created out of bonding. At the network configuration level, Port Channel – LACP will need to be configured to support the link aggregation configuration at the server end.

Additionally, IP Multipathing can be set up either on top of link aggregation or configured independently to further support the network redundancy via failover policies adopted (active/active or active/standby).

These will specifically be explained on the specific clustering framework (for Oracle Solaris Cluster or Veritas Cluster).

Cluster Storage

For the storage configuration, the cluster environment is set up by having both SAN fabric and host storage port-level redundancy. Cluster storage is a shared storage carved from SAN- or NAS-based redundant storage environments. Cluster storage LUNs (Logical Units), also known as multihome disks, are presented to all participant cluster nodes and made active/standby, based on which node is the active cluster node. Software MPXIO is used to create single virtual paths out of multiple redundant paths created out of SAN fabric and host storage ports. Ideally, disks pulled out of SAN storage are raid controlled and provide sufficient disk redundancy, although to have better resilient configuration, it's better to obtain storage from two separate storage appliances.

Quorum Device

Cluster uses quorum voting to prevent split brain and amnesia. Quorum determines the number of failures of node a cluster can sustain and for any further failure cluster must panic. Quorum disk is used for supporting the Cluster quorum.

Cluster Split Brain

Split brain occurs when cluster interconnects between cluster nodes break. In that case, each broken cluster node partitions to form a separate cluster and tries to bring up cluster services simultaneously and access the respective shared storage leading to data corruption. Additionally, it might duplicate network addresses, as each new partitioned clusters might own the logical hosts created as a part of the cluster.

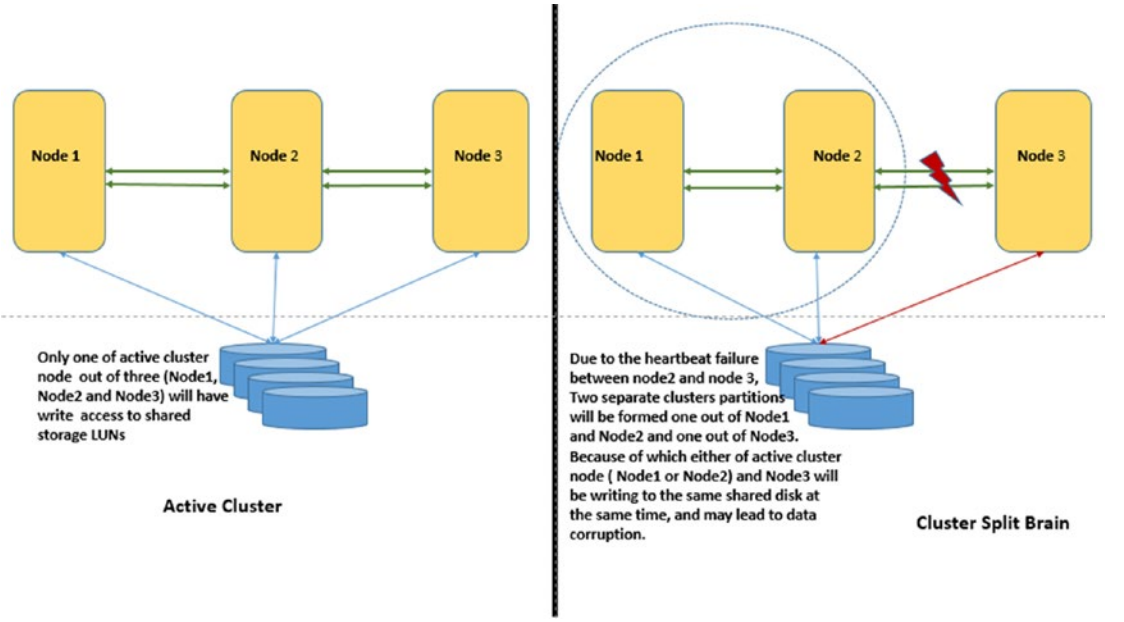

Figure 2-2. *Split Brain*

Quorum disk resolves this by ensuring a partition cluster with the majority of votes will be allowed to survive (like the above cluster made of Node1 and Node2) and other cluster partition (Node3) will be forced to panic and fence the node from disks to avoid data corruption.

Cluster Amnesia

As stated in Oracle Solaris documentation, "Amnesia is a failure mode in which a node starts with the stale cluster configuration information. This is a synchronization error due to the cluster configuration information not having been propagated to all of the nodes."

Cluster amnesia occurs when cluster nodes leave the cluster due to technical issues. Cluster uses the cluster configuration database to keep this updated across all cluster nodes. Although for the nodes going down due to technical reasons, it will not have an updated cluster repository. When all cluster nodes are restarted, the cluster with the latest configuration should be started first, meaning the cluster node brought down last should be brought up first. But if this is not done, the cluster itself will not know which cluster node contains the right cluster repository and may lead to a stale cluster configuration database.

Figure 2-3 below explains further the process of cluster amnesia.

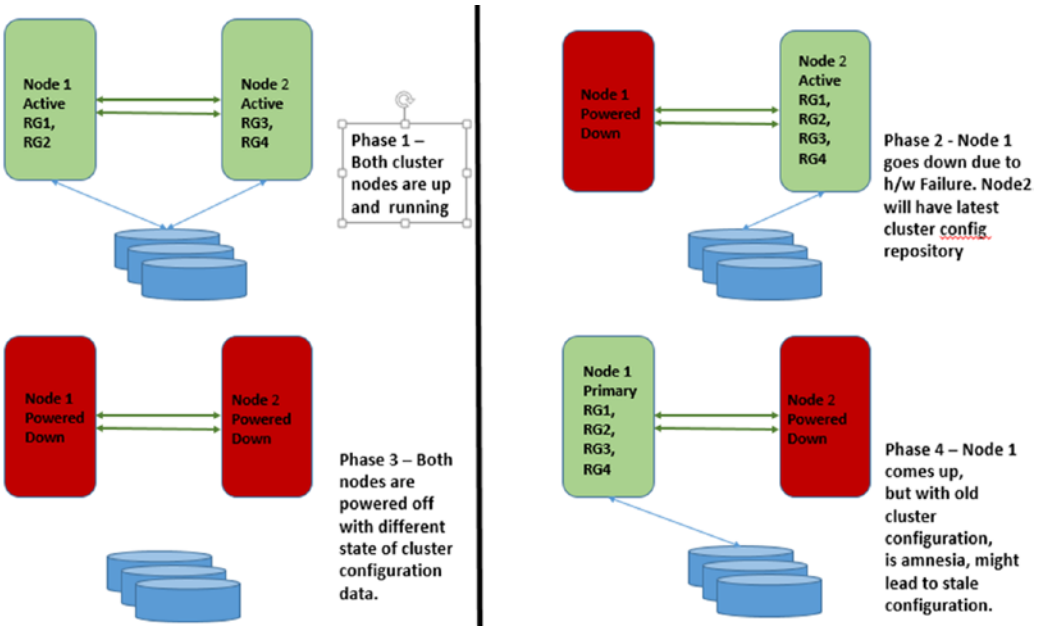

Figure 2-3. *Amnesia*

To avoid this, cluster quorum guarantees at the time of cluster reboot that it has at least one node (either quorum server or quorum disk) with the latest cluster configuration.

Oracle/Solaris Cluster Framework

Oracle Solaris Cluster design is the same as what is shown in Figure 2-1. Figure 2-4 shows a cluster design for an Oracle Solaris Cluster.

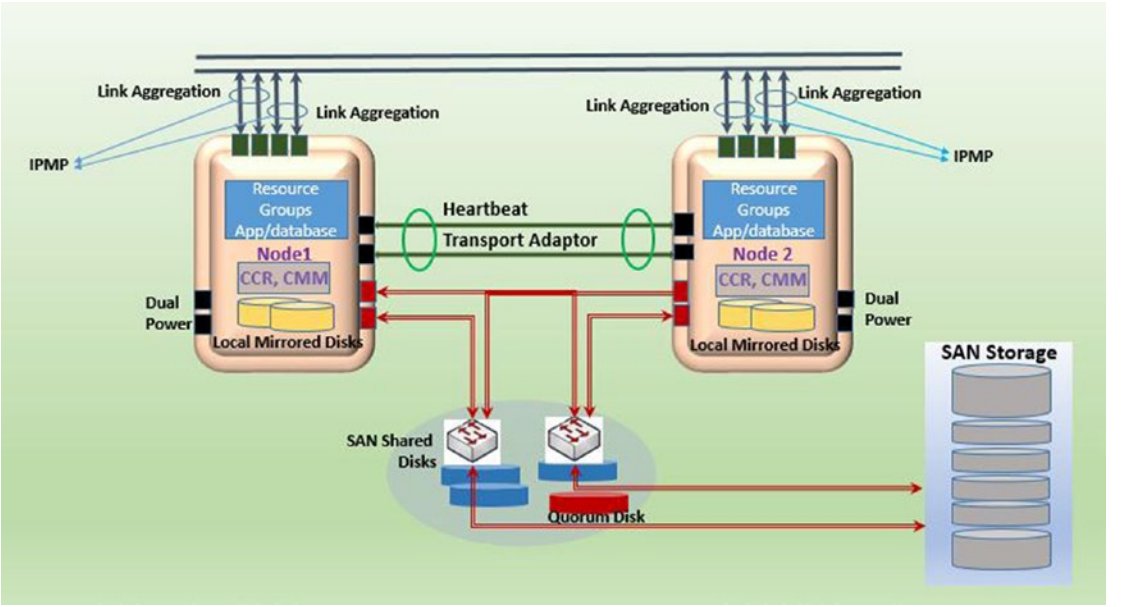

Figure 2-4. *A cluster design for an Oracle Solaris Cluster*

Oracle Solaris Cluster software has two components: Kernel and User Land.

Kernel land is the area that is controlled using OS kernel libraries and for setting up global file systems and devices, setting up network communications, the Sun Volume Manager for setting up multihome devices and for creating cluster configuration repositories, and managing cluster memberships and cluster communications.

User land is the area that is for user-controlled libraries and binaries for cluster command and libraries, and for setting up cluster resources and resource groups through the Resource Group Manager, making data services scalable and failover data services using the command line, etc.

Oracle Solaris Cluster Topologies

Cluster topology helps to understand the way shared devices are connected to cluster nodes in order to provide maximum availability.

There are three supported cluster topologies available for the Oracle Solaris cluster configuration, known as Clustered pairs, Pair+N and N+1 (Star), and N*N topologies.

In Cluster pairs topology, cluster nodes are configured in pair and storages are shared across paired cluster nodes, within the same cluster framework. This way the failover of the respective application will occur within the paired cluster nodes. An example of this configuration is when different types of applications (like application and database) are part of the same cluster administrative framework.

Pair+N topology means a pair of hosts are connected to shared storage and other hosts that are not directly connected to storages but uses cluster interconnects to access shared storage.

N+1 (Star) topology means a combination of primary hosts and one secondary/standby host. The secondary host should have a connection to all shared storages and have sufficient capacity to take the load of primary host in case of failure.

N*N, the most commonly used cluster topology, is configured by having each node connected to each shared storage, and the application can failover to either of the hosts available.

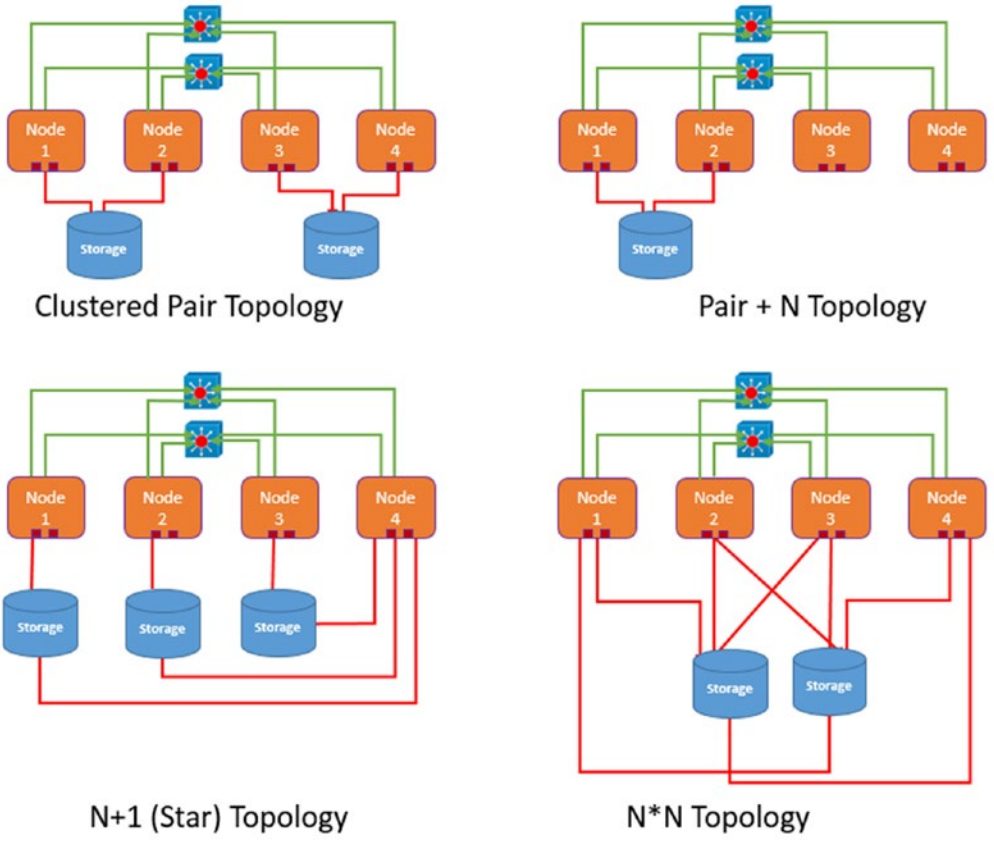

Figure 2-5.

Oracle Solaris Cluster Components

A critical component of the cluster design is the network configuration used at different layers. Each network layer serves the purpose of communication within cluster environments and outside the cluster. These network layers are **Public Network –** Network that the outside world or end user connects with the back-end host, typically web tier-based connectivity. **Core Network –** Company's internal network for communication (typically used for Application to DB connectivity), **Private Interconnect (Heartbeat connection) –** This is a heartbeat connection used only for intercommunications between hosts connected for availability polling. **Console network –** Host management console connection for remotely managing headless servers, and last, **Cluster Management Network** – Usually part of the console management network but optionally chosen to be part of a core network or separate dedicated network for cluster management.

The next components as parts of a cluster framework are shared disks. These disks can be shared as a part of SAN storage, or NAS storage, and they should be visible by all cluster nodes to ensure failover storage-based application services.

Further Oracle Solaris cluster-specific components are defined based on their specific cluster function to support the high availability feature.

Cluster Membership Monitor (CMM)

The Cluster Membership Monitor **(CMM) ensures** the health of a cluster node through active membership communicated via private interconnects. A critical cluster component protecting data integrity across distributed computing operations. At the time of cluster changes due to failures of cluster node or changes to cluster configuration, the CMM initiates reconfiguration using transport interconnects to send and receive status alerts. So basically, CMM is the core of cluster configuration ensuring persistent configuration. Further deeper into the technical understanding, CMM is a distributed set of agents that communicates and exchanges messages to perform consistent membership view on all nodes, driving synchronized reconfiguration in response to membership changes (nodes leaving or joining cluster), cluster partitioning (split brains, etc) is taken care of for safer reboot of cluster nodes, ensuring defected cluster nodes are cleanly left out of the cluster for repair, uses cluster heartbeats for keeping control of cluster membership and in case of change of cluster membership, it reinitiates the cluster configuration.

Cluster Configuration Repository

The CCR is a cluster database for storing information that pertains to the configuration and state of the cluster. All cluster nodes are synchronized for this configuration database. The CCR contains Node names, Disk Group, device configuration, current cluster status, and a list of nodes that can master each disk group.

Heartbeats/Cluster Interconnect

The core of the cluster uptime is heartbeat, which monitors cluster nodes ensuring reconfiguration of cluster repositories at the time of node joining or leaving the cluster group. Cluster heartbeats operate over dedicated private interconnects (cluster interconnects), helping seamless cluster configuration synchronization.

Cluster Membership

Every cluster node is assigned a cluster member id by assuring it to be part of cluster.

Quorum

The Cluster Membership Monitor (CMM) and quorum in combination assures, at most, one instance of the same cluster is operational at any time, at the time of broken cluster interconnect. Sun Cluster uses quorum voting to prevent split brain and amnesia.

As explained before, split brain means two nodes access the same device leading to data corruption so failure fencing is used to prevent split brain. Amnesia occurs when there is a change in cluster configuration while one of the nodes is down. When this node comes up, there will be an amnesia as to which is the primary cluster node, which keeps the correct cluster configuration data. If nodes are shut down one after the other, the last node leaving the cluster should be the first booting up in order to ensure CCR consistency.

Disk (I/O) Fencing

Helps preserve data integrity and non-cluster nodes are prevented from accessing and updating any shared disk/LUN. Or in other words, failure fencing protects the data on shared storage devices against undesired writes accessed by nodes that are no longer in the cluster and are connected to the shared disk.

In a two-node cluster, SCSI-2 reservation is used to fence a failed node and prevent it from writing to dual host-shared disks. For more than two nodes, SCSI-3 Persistence Group Reservation (PGR) provides fencing and helps shared disks' data consistency.

Fault Monitors

There are a number of monitors that are constantly monitoring the cluster and detecting faults; the cluster is monitoring applications, disks, network, etc. These are Data Service monitoring, Disk-Path monitoring, and IP Multipath monitoring respectively.

Cluster Resource Group and Resources

These are Cluster Application/Database or web services running as a part of cluster services for failover and/or load balancing. Cluster resources are components of Cluster Resource groups set up with dependencies. Some of the example cluster resources are NIC-based cluster resource, Floating IP-based cluster resource, HA Storage Cluster resource, and application/database-based cluster resources.

Data Services

A data service is the application cluster resources, like an Apache server or an Oracle database; the cluster will manage the resource and its dependencies, and it will be under the control of the Resource Group Manager (RGM). The RGM performs 1. Start and Stop the Data Service 2. Monitor the Data Service (faults, etc) and 3. Help in failing over the data services. Cluster resource groups and resources are configured with dependencies to ensure before the start of application necessary network and storage components are in place. Cluster resources are configured with start/stop and probe methods.

Cluster Daemons

Some of the core cluster daemons are listed below.

cluster – System kernel process, the core of cluster daemon. This process cannot be killed because it is always in the kernel.

failfastd – The failfast daemon allows the kernel to panic if certain essential daemons have stopped responding or failed.

cl_eventd – This daemon registers and forwards cluster events (such as nodes entering and leaving the cluster).

qd_userd – This daemon server is used for probing the health of Quorum devices.

rgmd – This is the resource group manager daemon.

rpc.pmfd – This is the process monitoring facility.

pnmd – This is the public network management daemon for monitoring IPMP information.

cl_eventlogd – This daemon is used for logging logs cluster events into a binary log file.

cl_ccrad – This daemon helps in accessing CCR.

scdpmd – This daemon monitors the status of disk paths.

sc_zonesd – This daemon monitors the state of Solaris 10 non-global zones.

IP Multipathing

IPMP provides reliability, availability, and network performance for systems with multiple physical interfaces. IPMP keeps monitoring failed interface, allows to failover and failback network interfaces. In Oracle Solaris, two types of IPMP methods are used: Link-Based IPMP and Probe-Based IPMP. In Link-based IPMP, configuration mpathd daemon uses an interface driver to check the status of the interface and uses no TEST address for failure detection. Link-based IPMP is the default IPMP configuration. In Probe-based IPMP configuration, in.mpathd daemon sends out probes onto a TEST address to a target system on the same subnet. It uses a multicast address to probe the target system.

IPMP is configured with options as deprecated, failover, and standby. **Deprecated** is used when interface is used as a test address for IPMP (basically the interface is deprecated to be used only for test communication and not for any application data transfer). **Failover** option to ifconfig is used to make interface as failover or no failover when the interface fails. And the option **Standby** is used for the interface to be configured as standby.

Oracle Solaris IPMP involves in.mpathd daemon, /etc/default/mpathd configuration file and ifconfig command options for IPMP configuration.

The important parameters in mpathd configuration file are the following:

1. **FAILURE_DETECTION_TIME**: Time taken by mpathd to detect a NIC failure in ms (default value – 10 seconds)
2. **FAILBACK**: To enable or disable failback after the failed link becomes available (default value – yes)
3. **TRACK_INTERFACES_ONLY_WITH_GROUPS** – If turned on, interfaces configured as part of IPMP are only monitored (default value – yes)

Below are examples of **Probe-based** and **Link-based** IPMP configurations:

```
        Active interface(s):   e1000g0
                               e1000g1
Standby interface(s):  -
        Data IP addresse(s):   10.0.0.50
        Test IP addresse(s):   10.0.0.51
                               10.0.0.52
```

Active/Active configuration

To configure probe-based IPMP – command line

```
# ifconfig e1000g0 plumb 10.0.0.50 netmask + broadcast + group ipmp0 up addif 10.0.0.53
netmask + broadcast + deprecated -failover up
# ifconfig e1000g1 plumb 10.0.0.52 netmask + broadcast + deprecated -failover group
ipmp0 up
```