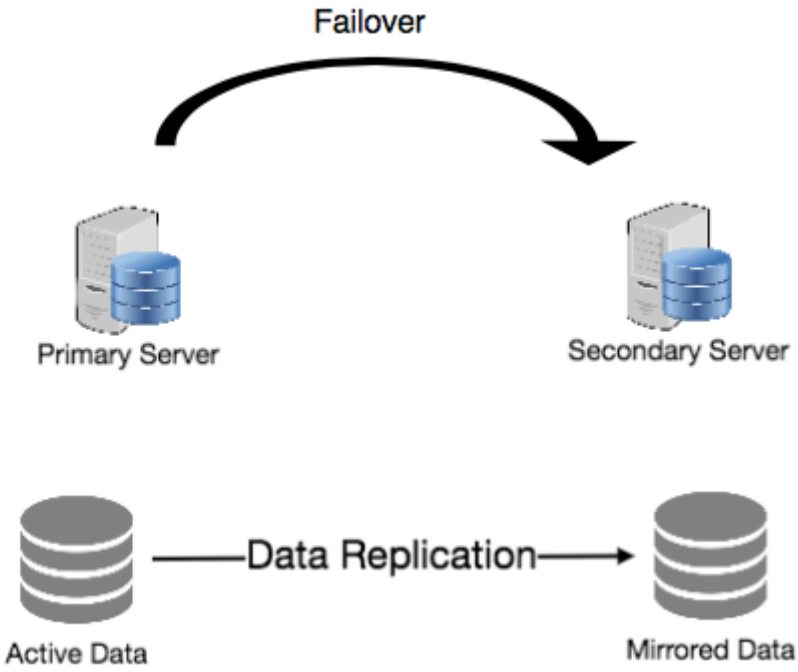
System Design — Redundancy and Replication

Concepts and considerations for Redundancy and Replication in System Design

Larry | Peng Yang
Apr 6, 2020 · 6 min read ★



1. Concepts

- Duplication of critical data or services with the intention of increased reliability and availability of the system.

2. Server failover

- Remove single points of failure and provide backups (e.g. server failover).

3. Shared-nothing architecture

- Each node can operate independently of one another.

- No central service managing state or orchestrating activities.
- New servers can be added without special conditions or knowledge.
- No single point of failure.

4. Models of Redundancy

Link: https://www.ni.com/ja-jp/innovations/white-papers/08/redundant-system-basic-concepts.html

4.1 Standby Redundancy

- Standby redundancy, also known as *Backup Redundancy* is when you have an identical secondary unit to back up the primary unit. The secondary unit typically does not monitor the system but is there just as a spare.
- The standby unit is not usually kept in sync with the primary unit, so it must reconcile its input and output signals on the takeover of the Device Under Control (DUC).
- You also need a third party to be the watchdog, which monitors the system to decide when a switchover condition is met and command the system to switch control to the standby unit and a voter.
- In Standby redundancy, there are two basic types, *Cold Standby* and *Hot Standby*.
  1. **Cold Standby**: The secondary unit is powered off, thus preserving the reliability of the unit, so it takes time to bring it online and it makes it more challenging to reconcile synchronization issues.
  2. **Hot Standby**: The secondary unit is powered up and can optionally monitor the DUC. It can also be used as the watchdog and/or voter to decide when to switch over. It shortens the downtime, which in turn increases the availability of the system.

4.2 N Modular Redundancy

- N Modular Redundancy, also known as Parallel Redundancy, refers to the approach of having multiply units running in parallel.
- All units are highly synchronized and receive the same input information at the same time.
- Their output values are then compared and a voter decides which output values should be used. This model easily provides bumpless switchovers.

- This model typically has faster switchover times than Hot Standby models, but the system is at more risk of encountering a common mode failure across all the units.

- In N Modular Redundancy, there are three main typologies: *Dual Modular Redundancy, Triple Modular Redundancy,* and *Quadruple Redundancy*.
  1. **Dual Modular Redundancy (DMR)** uses two functional equivalent units, thus either can control the DUC.
  2. **Triple Modular Redundancy (TMR)** uses three functionally equivalent units to provide redundant backup.
  3. **Quadruple Modular Redundancy (QMR)** is fundamentally similar to TMR but using four units instead of three to increase reliability.

4.3 1:N Redundancy

- 1:N is a design technique used where you have **a single backup for multiple systems** and this backup is able to function in the place of any single one of the active systems. This technique offers redundancy at a much lower cost than the other models by using **one standby unit for several primary units**.

- This approach only works well when the primary units all have very similar functions, thus allowing the standby to back up any of the primary units if one of them fails.

- The drawbacks of this approach are the added complexity of deciding when to switch and of a switch matrix that can reroute the signals correctly and efficiently.

5. Redundancy at different layers

5.1 Network Redundancy

- Layer 2 redundancy (switches)
  1. **Active/Standby** using Spanning Tree Protocol
  2. **Active/Active** using per VLAN spanning tree protocol and Multiple Spanning Tree Protocol

- Layer 3 redundancy
  1. **First Hop Redundancy Protocols** are designed to provide redundancy to clients by representing multiple default gateways in a group with a single IP address.

5.2 VM/Server Redundancy

- **How VMware HA Works**: VMware HA provides high availability for virtual machines by pooling them and the hosts they reside on into a cluster. Hosts in the

cluster are monitored and in the event of a failure, **the virtual machines on a failed host are restarted on alternate hosts**.

- **Primary and Secondary Hosts in a VMware HA Cluster:**
  1. When you add a host to a VMware HA cluster, an agent is uploaded to the host and configured to communicate with other agents in the cluster.
  2. The first five hosts added to the cluster are designated as primary hosts, and all subsequent hosts are designated as secondary hosts.
  3. The primary hosts maintain and replicate all cluster state and are used to initiate failover actions.
  4. If a primary host is removed from the cluster, VMware HA promotes another (secondary) host to primary status.
  5. If a primary host is going to be offline for an extended period of time, you should remove it from the cluster, so that it can be replaced by a secondary host.
  6. Any host that joins the cluster must communicate with an existing primary host to complete its configuration (except when you are adding the first host to the cluster).
  7. At least one primary host must be functional for VMware HA to operate correctly.
  8. If all primary hosts are unavailable (not responding), no hosts can be successfully configured for VMware HA. You should consider this limit of five primary hosts per cluster when planning the scale of your cluster.

- One of the primary hosts is also designated as the active primary host and its responsibilities include:
  1. Deciding where to restart virtual machines.
  2. Keeping track of failed restart attempts.
  3. Determining when it is appropriate to keep trying to restart a virtual machine.

- If the active primary host fails, another primary host replaces it.

5.3 Database Redundancy

- Have more than one copy of your data in a database system. It can be either at the table level or at the field level. Usually, the copies are called a replica.

- e.g. Replicas in ClustrixDB are distributed across the cluster for redundancy and to balance reads, writes, and disk usage.

5.4 Storage Redundancy

- Azure Storage Redundancy

Locally Redundant Storage (LRS)

LRS ensures that your data stays within a single data center in your chosen region. Data is replicated three times. LRS is cheaper than the other types of redundancy and doesn't provide protection against data center failures.

Zone-Redundant Storage (ZRS)

Only available for block blobs, ZRS keeps three copies of your data across two or three data centers, either within your chosen region or across two regions.

Geo-Redundant Storage (GRS)

This is the type of redundancy that Microsoft recommends by default, and it keeps six copies of your data. Three copies stay in the primary region, and the remaining three are replicated to a secondary region.

Read-Access Geo-Redundant Storage (RA-GRS)

The default redundancy setting, RA-GRS replicates data to a secondary region, where apps also get read access to the data.

5.5 Application Redundancy

Redundant applications, normally server applications, provide backup capability in the event that an application fails. That is, if one server (the primary server) goes out of service for some reason, such as lost connectivity, the other server (the backup server) can act as the primary server, with little or no loss of service.

Other Topics for System Design

- System Design — Load Balancing
- System Design — Caching
- System Design — Sharding / Data Partitioning
- System Design — Indexes
- System Design — Proxies
- System Design — Message Queues

- System Design — Redundancy and Replication
- System Design — SQL vs. NoSQL
- System Design — CAP Problem
- System Design — Consistent Hashing
- System Design — Client-Server Communication
- System Design — Storage
- System Design — Other Topics
- Object-Oriented Programming — Basic Design Patterns in C++

System Design Interview Redundancy Replication