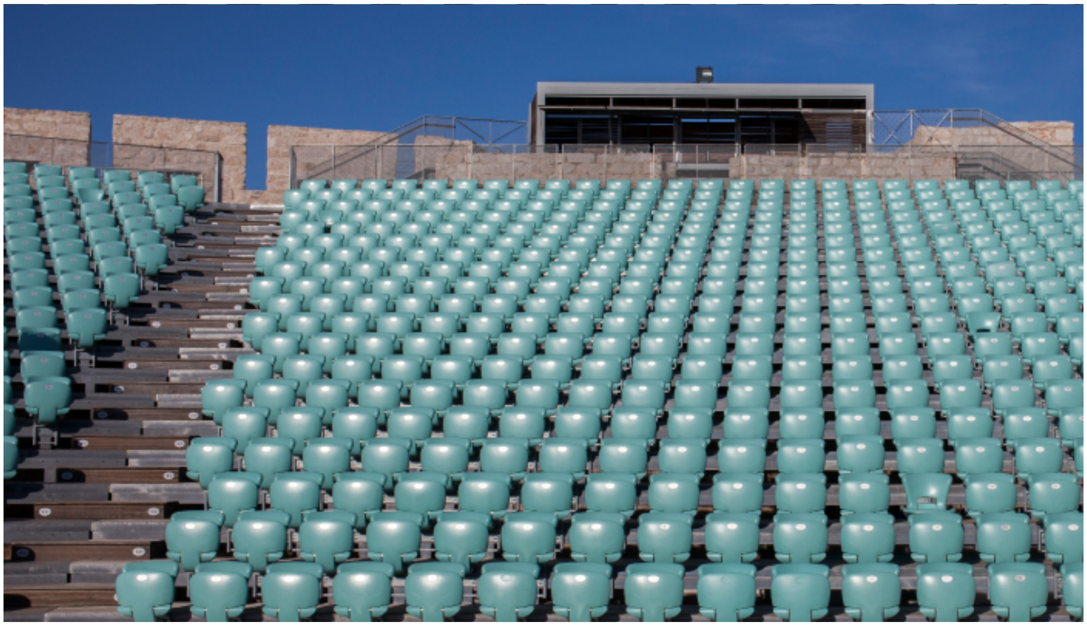
bmc.com

N-Modular Redundancy Explained: N, N+1, N+2, 2N, 2N+1, 2N+2, 3N/2

August 30, 2021 5 minute read Shanika Wickramasinghe, Muhammad Raza

9-12 minutes



No matter if it’s software or hardware, any IT system should have built-in measures to ensure smooth operations—especially when you encounter unexpected issues. This means that any IT system should be dependable enough to handle unexpected situations which allow users to use the IT system confidently.

One way to achieve this dependability? Introducing redundancy into the system. Introducing redundancy helps to:

- Minimize service interruptions due to failures
- Create a fault tolerance architecture

In simple terms, redundancy refers to introducing multiple or duplicates of exciting components that can be used to carry out the required tasks in case of failure. There are different redundancy principles that can be applied to systems.

In this article, we will have a look at N-Modular redundancy and its corresponding redundancy levels.

What is N-modular redundancy?

Redundancy is a critical component in any good IT system. There are different kinds of redundancy levels with redundancy, and we'll look at these below:

- N
- N+1, N+2
- 2N, 2N+1, 2N+2
- 3N/2

N redundancy

N redundancy refers to the bare minimum of required components for an IT system to operate. This level is characterized by two factors:

- No redundancy solution is available for the system.
- The system will be non-functional and inaccessible in case of a failure until the issue is diagnosed and resolved.

No system should operate at this redundancy level. This level does not provide any redundancy to the system. Instead, users should take this level as the base level and build on top of it to introduce proper redundancy to an IT system.

N+1, N+2 redundancy

As the name suggests, N+1 refers to the base level of resources required for the system functionality—plus a single backup. This is the minimum requirement for introducing redundancy to an IT system.

At this stage, the system can function while providing a single redundancy solution. This redundancy level is appropriate for a small IT system, but it's not suited for medium- to large-scale systems.

N+2 refers to the next step up, and it consists of the resources needed for the system functionality plus two separate backups. This further increases the redundancy of an IT system as well as the confidence of end-users about the system since there are two separate backups. In this case, the system functionality can be restored through the other backup even if a single backup is corrupted.

In addition to N+1 and N+2, there may be instances where even more backups are maintained. These are referred to as N+X, where X stands for any number of backups to ensure the functionality of the system. This can be +3,+4,+5… Still, most command levels will be N+1 and N+2 unless there is a specific or unique requirement to keep multiple copies, such as a compliance policy.

2N, 2N+1, 2N+2 redundancy

N refers to the minimum number of resources (amount) required to operate an IT system. 2N simply means that there is twice the amount of required resources/capacity available in the system.

For a simple example, let's consider a server in a data center that has ten servers with an additional ten servers that act as a dedicated capacity. This reserves a combination of 20 servers in total, providing 2N redundancy. This way, 2N always provides an excess capacity to the IT system.

2N+1 correlates to a system with twice the required resources/capacity to function normally plus a backup as an additional redundancy step. These additional backup systems can provide redundancy even if there is an issue with all of the additional capacity.

2N+2 refers to a system with additional capacity plus two backups to provide one of the highest levels of redundancy. In this method, if there are ten servers in a data center, it will have another ten identical servers as reserved capacity while having two more servers that will act as backups in case of an emergency.

2N+2 is considered the highest level of redundancy methodology that is commonly used in the IT industry.

3N/2, 4N/3 redundancy

3N/2, 4N/3, or more specifically AN/B, refers to a redundancy methodology where additional capacity is based on the load of the system.

For example, consider 3N/2 redundancy applied to power delivery infrastructure in a data center environment. In this instance, there will be separate power delivery systems powering two workloads (two servers). This results in each power delivery system only using 67% of the available capacity at a specific time (or the inverse of the 3N/2 ratio).

If we consider a 4N/3 scenario, four power delivery systems will power three servers, resulting in each power delivery system utilizing 75% of the available capacity.

Redundancies can degrade performance

All the methodologies mentioned above provide some kind of redundancy to an IT system.

However, it’s crucial to understand that using backups (spare components/resources) may lead to performance degradations. That’s because these spare resources may not be identical to the original system resources in terms of capacity. This only affects backups or (+X) scenarios when dealing with capacity reservation (XN), as the additional capacity is identical to the operating system without any impact on the performance of the system.

Redundancy configurations

Active, passive, and load sharing (standby) are redundancy configurations available when implementing a redundancy methodology.

- **Active**. In an active configuration, the redundant component is operated simultaneously with the original component. However, in case the original fails, the redundant component will be used.
- **Passive**. In a passive config, redundant component is available yet not operational while the original component is active. It will be activated to provide functionality in the event of a failure.
- **Load sharing (standby)**. This fulfills the availability gap until the original or active component is completely available. Additionally, load sharing can be used here as a partial or a temporary redundancy method to provide additional capacity.

The configuration and the methodology you choose will depend on facts such as:

- User and system requirements
- Cost
- Resource availability
- Compliance requirements

The higher the level of redundancy requirements in a data center, a more complex redundancy methodology will be required to handle it. While this complexity will lead to higher costs and resource requirements to implement and manage properly, it will provide the best redundancy for the underlying system.

Redundancy increases reliability & availability

The dependability of an IT system will differ based on its reliability and availability. Redundancy is one of the best ways to increase the reliability and availability of a system using different redundancy methodologies.

One factor that affects the complexity and requirements of a redundancy method is the failure rate of individual components of a system:

- If individual components have low failure rates, a simple redundancy policy can be applied to provide high availability.
- Components with higher failure rates require complex redundancy policies to mitigate any issues and provide guaranteed availability.

Another factor to consider when implementing a redundancy methodology is the workload of the system, as higher workloads cause system components to be constantly stressed, leading to faster component degradation.

(Understand how redundancy affects availability.)

Redundancy is critical

Redundancy is an essential component in any IT system. It ensures that an IT system can function properly in the event of an unforeseen issue within this rapidly evolving technological landscape. Thus, redundancy allows users to use the IT system reliably, leading to a more satisfying end-user experience.