RESILIENCY VS REDUNDANCY: WHAT'S THE DIFFERENCE?

Resilience and redundancy offer ways to yield a dependable system—known as system dependability. Both resilience and redundancy are critical for the design and deployment of computer networks, data centers, and IT infrastructure. Despite the critical nature of both, resiliency and redundancy are not the same thing.

In this article, we'll explain system dependability, the differences in resiliency and redundancy, and how they both contribute to your system's dependability.

What is system dependability?

System dependability is defined as "the trustworthiness of a computing system which allows reliance to be justifiably placed on the service it delivers". A computing system and its components must be strong enough to consistently deliver its service to end users.

A key attribute of dependability is the reliability of the system, so that the rendered service is available throughout its operating duration at an acceptable performance level. Both resilience and redundancy help achieve a system's dependability, but they are not interchangeable strategies.

Resiliency and redundancy

In the domain of computer networking, resilience and redundancy establish fault tolerance within a system, allowing it to remain functional despite the occurrence of issues such as power outage,

cyber-attacks, system overload, and other causes of downtime. In this context, the terms can be defined as follows:

- **Redundancy** is the intentional duplication of system components in order to increase a system's dependability.
- **Resilience** refers to a system's ability to recover from a fault and maintain persistency of service dependability in the face of faults.

With these definitions, redundancy and resilience are not interchangeable but complementary: **Redundancy is required to enhance a system's resilience, and resilience of individual system components ensures that redundant components can recover to functional state following a fault occurrence.** Redundancy is an absolute measure of the additional components supporting system resilience, whereas resiliency is a relative (and continuous) measure of the impact of fault on the system operation.

How to build redundancy

Functional capabilities of computer system components encompass hardware, software, information, and time-based performance. Therefore, introducing redundancy into a system with respect to these functional categories can help break the fault-error-failure chain.

When a fault occurs, it produces a performance error, resulting in the system's failure to deliver a dependable service. Introducing redundancy in the form of additional hardware components, multiple software features, error checks for data, and consistency in information or system performance with respect to time reduce the possibility of a fault leading to system failure.

Redundancy can take various forms and configurations, including:

- **Active redundancy.** Components are actively participating in system operation and sharing workload distribution. For example, you may transmit data over two fiber optic cables in parallel instead of just one. If one cable breaks, the system remains functional. Because the redundant component is already activated and involved in system functionality, the system is strong enough to recover from a state of fault. However, actively redundant components may also share a single point of failure, in which case the additional system elements may not contribute to system performance. For instance, connections loosening due to heat or other external damages can cause both fiber optics cables to disconnect simultaneously.
- **Passive redundancy.** Application of redundant components only following the fault state. The additional components are present but not actively involved in system functionality. Recovery may take time since the activation process and connection to the system of the redundant components is not instantaneous in real-world applications. For instance, with a cold backup system additional hard drives are available to store data, but these additional drives connect to perform the necessary storage operations only when the primary storage volume fails.
- **Homogeneous redundancy.** Application of redundant components of the same type, such as using the same brand and model of the hardware in a redundant configuration. The additions may yield exponential improvement in system resilience. However, this also makes the system susceptible to a single cause of failure. If the hardware is vulnerable to a certain security exploit, then the redundant hardware will not offer tolerance to that exploit.
- **Diverse redundancy.** Application of redundant components different to the primary component. This configuration reduces the possibility of a single cause of failure impacting

system dependability. For instance, trains use a mix of electric, pneumatic, hydraulic, and air braking systems. If a primary braking system fails due to excessive wear and tear or malfunction, secondary and tertiary braking systems take over immediately to reduce the risk of collision. Generally, diverse redundancy is quite complex and expensive to maintain.

A variety of redundancy configurations can be used for the design and deployment of IT infrastructure with a range of availability requirements, cost, and complexity considerations.

How to build resiliency

Redundancy is a measure to enhance resilience, allowing for a proactive or reactive response to impactful changes facing the system. Resilience requirements can be diverse and evolving, especially in scalable environments where a fault at one system element can cascade across the system. Other measures of resilience can include the system's ability to:

- Forecast potential faults
- Isolate impacted components
- Protect against potential faults
- Remove faults and recover from a fault state
- Restore optimal system performance

Resilience may be measured in terms of the availability of a system at any given instance, as a continuous function of the frequency or delays in occurrence of the faults and the speed of recovery from a fault state. Therefore, important metrics for resilience include the system's mean time to failure (MTTF) and the mean time to recovery (MTTR).

The choice for configuration of redundancy maps directly to the MTTF and MTTR of a system. An optimal choice for redundancy configuration will aim to achieve the highest MTTF and lowest MTTR while reducing the cost and complexity of the redundant system. In other words, redundancy configuration must be chosen such that the system remains available for the prolonged time period on average (MTTF), but when it does fail, the average recovery time to optimal state (MTTR) should be the lowest, without incurring unacceptable cost and complexity of the system. Other useful metrics include the mean time between failure (MTBF), which is the mathematical sum of MTTF and MTTR.

The choice for resilience and redundancy is based on the requirements on dependability of the system, with respect to the cost, complexity, and practical impediments involved. In terms of goals and objectives, resilience is often considered as an important requirement as it may be used to describe dependability attributes such as availability, reliability, and performance. However, redundancy in itself is never considered as a goal, since the requirements associated with dependability should be fulfilled with a redundancy configuration incurring the lowest possible cost and complexity.

Additional resources

Redundancy vs Resiliency from **Vology**