enterprisestorageforum.com

What Is Fault Tolerance? | Enterprise Storage Forum

Paul Rubens

11-14 minutes

Fault tolerance is a concept used in many fields, but it is particularly important to data storage and information technology infrastructure. In this context, fault tolerance refers to the ability of a computer system or storage subsystem to suffer failures in component hardware or software parts yet continue to function without a service interruption – and without losing data or compromising safety.

Fault tolerance in systems can encompass the entirety of the data storage platform, from SSD to HDD to RAID to NAS. Issues in fault tolerance are numerous, but the ultimate goal of a fault tolerant system is to provide protection – but this idea is more complex than it sounds.

How Does Fault Tolerance Work?

At the most basic level, fault tolerance can be built into a system by ensuring that it has no single point of failure. This requires that there is no single component which, if it stopped working properly, would cause the entire system to stop working completely.

A typical single point of failure in a standard system is the power supply unit (PSU) which takes in the main alternating current (AC) supply and converts this into direct current (DC) of various voltages to power different components. If the PSU fails, then all the components it powers will also fail, generally leading to a catastrophic failure of the entire system.

Fault tolerance typically follows one of these two models:

- **Normal functioning** Under some circumstances, a fault tolerant system encountering a fault may continue to function as normal, without any change in throughput, response time or other performance metric.
- **Graceful degradation** Other fault tolerant systems will, in the face of certain faults, experience “graceful degradation” in performance. That is to say that the impact of a fault on the performance of the system will be in proportion to the severity of the fault. So a small fault will have a small impact rather than a major impact or even cause the system as a whole to fail. A highly fault tolerant system will continue to work even when it experiences one or multiple severe faults.

A secure fault tolerance plan requires multiple data repositories to *ensure redundancy.*

Building In Fault Tolerance

The key purpose of creating fault tolerance is to avoid (or at least minimize as far as possible) the possibility that the functionality of the system ever becomes unavailable because of a fault in one or more of its components.

Fault tolerance is necessary in systems that are used to protect people’s safety (such as air traffic control hardware and software systems), and in systems which security, data protection and integrity, and high value transactions depend on.

Redundancy

To remove a single point of failure and provide fault tolerance, fault tolerant systems use the concept of “redundancy.” In practice, in the above example, this would mean equipping the system with one or more extra PSUs which are redundant in the sense that they are not required to power the system when the primary PSU is functioning normally.

However, if the primary PSU fails (or a fault such as overheating is detected which indicates that it is about to fail) then it can be taken out of service and one of the redundant PSUs can kick in without any interruption to the functioning of the overall system.

Ideally, redundancy would be provided for all components in a system, but in practice this is usually too expensive. For that reason designers calculate how likely a component is to fail, how important it is to the system, and how expensive it is to make redundant, before selecting the most best candidates for redundancy.

An alternative approach is to treat redundancy at the system level, having an alternate entire computer system which can kick in in the event of a system failure.

Diversity

In some cases, it may not be possible to provide redundancy, and an example of this is the main electrical supply which normally comes from the public electricity grid. If the main electricity supply fails (perhaps due to a power station failure or interruption to power lines during a storm) then it is usually not possible to access an alternative public electricity grid.

In this case fault tolerance can be achieved by diversity, which in practice means getting an electricity supply from another source entirely – most likely a backup electricity generator which kicks in automatically in case of a main power failure.

In some cases the “diverse” option (in this case the generator) may not have the same capacity as the primary option, which may necessitate a graceful degradation of service until the primary option can be restored.

Replication

A more complex way to achieve fault tolerance is through “replication.” This involves running multiple identical versions of a system or subsystem, and checking that their functioning always results in identical results. If results differ then some procedure is called up to decide which system is faulty.

Most commonly a “democratic” system is used, so if three systems provide identical results and a fourth provides a different result then the fourth is assumed to be faulty.

An alternative approach is to rerun a procedure for which the correct result is known to check which system comes up with a different result, indicating that it is faulty.

Replication can be carried out at the component level – for example by having three processors all working simultaneously – or it can be carried out at the system level, with a cluster of identical computer systems working simultaneously.

Elements of Fault Tolerant Systems

Hardware systems

A typical computer system or data storage system includes a central processing unit (CPU), system memory (RAM), secondary storage systems such as hard disk drives, along with a PSU, network interface, and motherboard.

To provide fault tolerance, the fault tolerant computer system or fault tolerant data storage system may use various elements. This includes replication for the CPU, redundancy for the PSU and RAM, hard drives configured in some form of RAID array which involves both redundancy and replication, and diversity of power supply with the provision of a backup generator.

Fault tolerant networking may be provided by redundant network interface cards (NICs), and/or a variety of diverse networking options such as a wired LAN NIC and a wireless LAN adapter.

Software systems

Software can be designed to be fault tolerant so that it can continue to operate even when it encounters an error, exception, or invalid input as long as it has been designed to handle such errors rather than defaulting to reporting an error and halting.

In particular, networking protocols such as TCP/IP have been developed expressly to enable the creation of fault tolerant networks. TCP/IP can continue to function in an environment where individual network links or nodes may become unavailable unexpectedly. It can adapt to the varying conditions in order to get packets to their destinations via whatever routes are available whenever possible.

Software systems can also use replication to provide fault tolerance: a critically important database can be continuously replicated to another server, so that if the server hosting the primary database goes down then operations can instantly be redirected to the replica database.

Alternatively, some services, notably web servers, can be placed behind a load balancer so that multiple servers all provide the same service. If one server develops a fault then the load balancer simply sends all web requests to the other ones until the faulty one is repaired.

This of course begs the question of what happens if the load balancer fails, and the answer is usually a failover system which instantly transfers web requests to a server at an alternative location. Since this failover location may not have the same resources as the primary data center this may result in graceful degradation until normal operations can be resumed.

Power sources

As mentioned earlier, many fault tolerant systems include multiple PSUs to provide redundancy in case of a PSU failure. And since it is usually not possible to obtain redundant main power supplies, most organizations rely in diversity in the form of power from an alternative source. This is typically a generator that starts up automatically in the event of a main power failure to ensure that hardware, storage, HVAC and other systems have the power they require.

What is the Difference Between High Availability and Fault Tolerance?

There is often some confusion between the concepts of high availability vs fault tolerance. At the most basic level, high availability refers to systems that suffer minimal service interruptions, whilst systems with fault tolerance are designed never to experience service interruptions.

In practice the difference may be small – many highly available systems aim for so-called “five nines,” or 99.999% uptime, which equates to just a few minutes of downtime a year.

But the principals that govern the two concepts are very different. Fault tolerant systems are designed to detect faults and remediate the problem (perhaps by swapping in a redundant component) without interruption, while highly available systems generally use standard hardware and aim to restore service quickly after an outage has occurred.

The reason why high availability is often deemed acceptable instead of fault tolerance usually comes down to cost: building fault tolerance into a system is far more expensive than accepting that short outages may occur from time to time. Many organization use a combination of the two: fault tolerant systems for the most critical activities, and high availability for less important ones.

Factors to Consider in Fault Tolerance

Cost

By far the biggest disadvantage of fault tolerance is that it leads to the building of systems which are far more costly than fault intolerant systems. That is because, among other reasons, they usually require multiple versions of the same components to provide redundancy.

They may also require extra equipment such as generators which need to be maintained and tested regularly even if they are never used. The cost is not only financial: fault tolerant systems may take up more valuable data center space.

The extra cost of fault tolerance is largely inevitable given the extra hardware involved. This means that organizations need to think long and hard about whether the advantages of fault tolerance vs. high availability are worth the extra costs.

Quality Degradation

To compensate for the increased cost of fault tolerance, there is often an almost inevitable tendency for organizations to accept the use of lower cost and inferior quality redundant components, since the reliability of an individual component is no longer critical. This can lead to an increase in support and maintenance costs, and if the components are of too poor quality it can even make the system as a whole less reliable than a fault-intolerant system.

To avoid this scenario it is necessary to monitor the performance of and lifespan of individual components both in relation to their cost, and in absolute terms.

Testing and Fault Detection Difficulties

By its very nature, fault tolerance makes it harder to spot when things are not working “properly” because component failures do not lead to systemic failure.

That means that more resources (and therefore expenditure) is often required to test and monitor the health of a system built for fault tolerance, and in some cases this may involve developing or acquiring custom software or procedures to help carry out the task.