the Availability Digest

www.availabilitydigest.com
@availabilitydig

Can Applications Achieve Seven 9s Availability?

March 2018

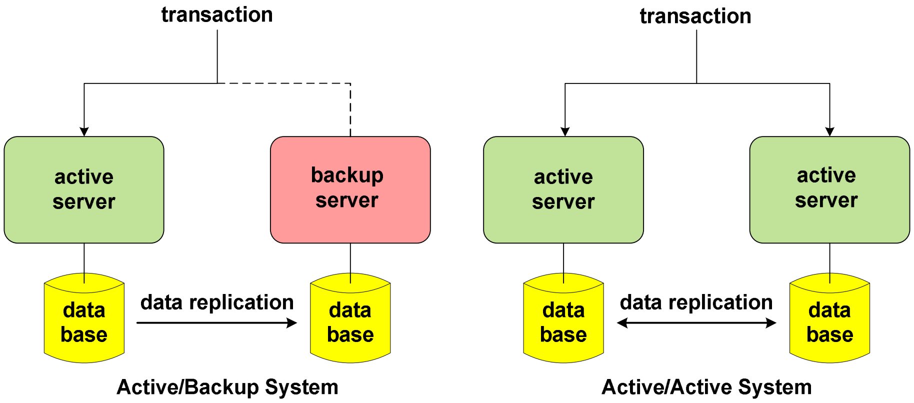
A company with which I am familiar is promoting its software solution as having seven 9s availability. Is this possible? We are used to measuring the availability of applications in the three- to five-9s range. Seven 9s means that the application will be down for no more than an average of three seconds per year.

The answer is that it depends upon what you are evaluating. If it is just a software application, then seven 9s is quite possible. Once a software application has been 'burned in' through use over a long period of time, there is not much that can go wrong with it. True, there may well be sections of code that have not been exercised extensively, such as error-handling routines; and these sections may still have coding errors that could lead to a software fault. However, by their very natures, these error-handling routines are seldom executed. The result is that the software application is highly unlikely to fail and could easily achieve an availability of seven 9s.

How Does a Hardware/Software System Achieve Seven 9s?

But if we are evaluating the availability of an application, we must take into account the hardware upon which it is running. A typical hardware server has an availability of three or four 9s. An application running on such a system, no matter how reliable the software, will have only three or four 9s availability.

The reliability of the hardware configuration can be increased via redundancy. This is a common practice when implementing a highly available system. One such architecture is an active/backup system. In this architecture, two servers are provided. One is the active server that is processing all of the transactions. The other is the backup server. The database of the backup server is kept synchronized with that of the active server via data replication. In this way, the backup server is immediately ready to take over transaction processing should the active server fail.



Another architecture is an active/active system. In this configuration, two servers are also provided. However, both servers are processing transactions. Their databases are kept synchronized via bidirectional replication. Should one server fail, all transactions are simply sent to the surviving server.

An active/active system has a higher availability than an active/backup system because there is no failover time (which can be measured in minutes in an active/backup system as the backup server is brought online). In fact, if the servers have an availability of four 9s (common in today's technology), an active/active system can achieve an availability of seven 9s.

This is the case of the application that I mentioned in the introduction. The software runs in an active/active architecture, and numerous installations have been sold worldwide. In such a case, seven 9s availability very well may be an accurate claim.

But What About Other Factors?

However, there are other factors that impact availability; and they also should be taken into account when calculating uptime. For instance, network failures can take a system down. In an active/active system, a network failure can separate the two servers so that they cannot replicate their databases. In that case, one server must be chosen as the transaction processor; and the system itself no longer is redundant.

Human error can take a system down. If an operator erroneously causes one server in an active/active system to fail, the system will still be operational; but it will no longer be a redundant system with fast failover.

Likewise, an environmental issue can take a system down. For instance, anything from a flood to an overheated data center can cause system failures. If the systems in an active/active architecture are separated geographically and are located in separate data centers, then an environmental fault will take down only one side of the active/active system; and user services will still be provided.

When a license agreement or an SLA is negotiated, does a software company have the right to promote seven 9s without taking into consideration the availability issues of the other system components? If all that is negotiated as having seven 9s availability is the software, that's fine. But buyer beware. Given all the factors that can contribute to downtime, it is unlikely that an overall availability of seven 9s is achievable.

Summary

Companies may claim extraordinary availabilities such as seven 9s for their applications. But when they negotiate SLAs , they should be careful about what their declarations entail. Do seven 9s availability encompass only their software applications, or are they attributing seven 9s capability to the servers, the network, the IT staff, the environmental factors? Given the liabilities that may be attached to an SLA that is not satisfied, it's crucial that both buyer and seller set accurate expectations.