High availability

From Wikipedia, the free encyclopedia

"Always-on" redirects here. For the software restriction, see Always-on DRM.

High availability (**HA**) is a characteristic of a system which aims to ensure an agreed level of operational performance, usually uptime, for a higher than normal period.

Modernization has resulted in an increased reliance on these systems. For example, hospitals and data centers require high availability of their systems to perform routine daily activities. Availability refers to the ability of the user community to obtain a service or good, access the system, whether to submit new work, update or alter existing work, or collect the results of previous work. If a user cannot access the system, it is – from the users point of view – *unavailable*.[1] Generally, the term *downtime* is used to refer to periods when a system is unavailable.

Contents [hide]

Principles [edit]

There are three principles of systems design in reliability engineering which can help achieve high availability.

日本語
Norsk bokmål
Polski
Română
Русский
Suomi
Türkçe
中文
Edit links

1. Elimination of single points of failure. This means adding redundancy to the system so that failure of a component does not mean failure of the entire system.
2. Reliable crossover. In redundant systems, the crossover point itself tends to become a single point of failure. Reliable systems must provide for reliable crossover.
3. Detection of failures as they occur. If the two principles above are observed, then a user may never see a failure – but the maintenance activity must.

Scheduled and unscheduled downtime [edit]

This section **does not cite any sources**. Please help improve this section by adding citations to reliable sources. Unsourced material may be challenged and removed. *(June 2008) (Learn how and when to remove this template message)*

A distinction can be made between scheduled and unscheduled downtime. Typically, scheduled downtime is a result of maintenance that is disruptive to system operation and usually cannot be avoided with a currently installed system design. Scheduled downtime events might include patches to system software that require a reboot or system configuration changes that only take effect upon a reboot. In general, scheduled downtime is usually the result of some logical, management-initiated event. Unscheduled downtime events typically arise from some physical event, such as a hardware or software failure or environmental anomaly. Examples of unscheduled downtime events include power outages, failed CPU or RAM components (or possibly other failed hardware components), an over-temperature related shutdown, logically or physically severed network connections, security breaches, or various application, middleware, and operating system failures.

If users can be warned away from scheduled downtimes, then the distinction is useful. But if the requirement is for true high availability, then downtime is downtime whether or not it is scheduled.

Many computing sites exclude scheduled downtime from availability calculations, assuming that it has little or no impact upon the computing user community. By doing this, they can claim to have phenomenally high availability, which might give the illusion of continuous availability. Systems that exhibit truly continuous availability are comparatively rare and higher priced, and most have carefully implemented specialty designs that eliminate any single point of failure and allow online hardware, network, operating system, middleware, and application upgrades, patches, and replacements. For certain systems, scheduled downtime does not matter, for example system downtime at an office building after everybody has gone home for the night.

Percentage calculation [edit]

Availability is usually expressed as a percentage of uptime in a given year. The following table shows the downtime that will be allowed for a particular percentage of availability, presuming that the system is required to operate continuously. Service level agreements often refer to monthly downtime or availability in order to calculate service credits to match monthly billing cycles. The following table shows the translation from a given availability percentage to the corresponding amount of time a system would be unavailable.

Availability %	Downtime per year[note 1]	Downtime per month	Downtime per week	Downtime per day
90% ("one nine")	**36.53 days**	**73.05 hours**	**16.80 hours**	**2.40 hours**
95% ("one and a half nines")	18.26 days	36.53 hours	8.40 hours	1.20 hours
97%	10.96 days	21.92 hours	5.04 hours	43.20 minutes

98%	7.31 days	14.61 hours	3.36 hours	28.80 minutes
99% ("two nines")	**3.65 days**	**7.31 hours**	**1.68 hours**	**14.40 minutes**
99.5% ("two and a half nines")	1.83 days	3.65 hours	50.40 minutes	7.20 minutes
99.8%	17.53 hours	87.66 minutes	20.16 minutes	2.88 minutes
99.9% ("three nines")	**8.77 hours**	**43.83 minutes**	**10.08 minutes**	**1.44 minutes**
99.95% ("three and a half nines")	4.38 hours	21.92 minutes	5.04 minutes	43.20 seconds
99.99% ("four nines")	**52.60 minutes**	**4.38 minutes**	**1.01 minutes**	**8.64 seconds**
99.995% ("four and a half nines")	26.30 minutes	2.19 minutes	30.24 seconds	4.32 seconds
99.999% ("five nines")	**5.26 minutes**	**26.30 seconds**	**6.05 seconds**	**864.00 milliseconds**
99.9999% ("six nines")	**31.56 seconds**	**2.63 seconds**	**604.80 milliseconds**	**86.40 milliseconds**
99.99999% ("seven nines")	**3.16 seconds**	**262.98 milliseconds**	**60.48 milliseconds**	**8.64 milliseconds**
99.999999% ("eight nines")	**315.58 milliseconds**	**26.30 milliseconds**	**6.05 milliseconds**	**864.00 microseconds**
99.9999999% ("nine nines")	**31.56 milliseconds**	**2.63 milliseconds**	**604.80 microseconds**	**86.40 microseconds**

Uptime and availability can be used synonymously as long as the items being discussed are kept consistent. That is, a system can be up, but its services are not available, as in the case of a network outage. This can also be viewed as a system that is available to be worked on, but its services are not up from a functional perspective (as opposed to software service/process perspective). The perspective is important here - whether the item being discussed is the server hardware, server OS, functional service, software service/process...etc. Keep the perspective consistent throughout a discussion, then uptime and availability can be used synonymously.

"Nines" [edit]

Main article: Nine (purity)

Percentages of a particular order of magnitude are sometimes referred to by the number of nines or "class of nines" in the digits. For example, electricity that is delivered without interruptions (blackouts, brownouts or surges) 99.999% of the time would have 5 nines reliability, or class five.[2] In particular, the term is used in connection with mainframes[3][4] or enterprise computing, often as part of a service-level agreement.

Similarly, percentages ending in a 5 have conventional names, traditionally the number of nines, then "five", so 99.95% is "three nines five", abbreviated 3N5.[5][6] This is casually referred to as "three and a half nines",[7] but this is incorrect: a 5 is only a factor of 2, while a 9 is a factor of 10, so a 5 is 0.3 nines (per below formula: $\log_{10} 2 \approx 0.3$):[note 2] 99.95% availability is 3.3 nines, not 3.5 nines.[8] More simply, going from 99.9% availability to 99.95% availability is a factor of 2 (0.1% to 0.05% unavailability), but going from 99.95% to 99.99% availability is a factor of 5 (0.05% to 0.01% unavailability), over twice as much.[note 3]

A formulation of the *class of 9s* c based on a system's unavailability x would be

$$c := \lfloor -\log_{10} x \rfloor$$

(cf. Floor and ceiling functions).

A similar measurement is sometimes used to describe the purity of substances.

In general, the number of nines is not often used by a network engineer when modeling and measuring availability because it is hard to apply in formula. More often, the unavailability expressed as a probability (like 0.00001), or a downtime per year is quoted. Availability specified as a number of nines is often seen in marketing documents.[*citation needed*] The use of the "nines" has been called into question, since it does not appropriately reflect that the impact of unavailability varies with its time of occurrence.[9] For large amounts of 9s, the "unavailability" index (measure of downtime rather than uptime) is easier to handle. For example, this is why an "unavailability" rather than availability metric is used in hard disk or data link bit error rates.

Measurement and interpretation [edit]

Availability measurement is subject to some degree of interpretation. A system that has been up for 365 days in a non-leap year might have been eclipsed by a network failure that lasted for 9 hours during a peak usage period; the user community will see the system as unavailable, whereas the system administrator will claim 100% uptime. However, given the true definition of availability, the system will be approximately 99.9% available, or three nines (8751 hours of available time out of 8760 hours per non-leap year). Also, systems experiencing performance problems are often deemed partially or entirely unavailable by users, even when the systems are continuing to function. Similarly, unavailability of select application functions might go unnoticed by administrators yet be devastating to users — a true availability measure is holistic.

Availability must be measured to be determined, ideally with comprehensive monitoring tools ("instrumentation") that are themselves highly available. If there is a lack of instrumentation, systems supporting high volume transaction processing throughout the day and night, such as credit card processing systems or telephone switches, are often inherently better monitored, at least by the users themselves, than systems which experience periodic lulls in demand.

An alternative metric is mean time between failures (MTBF).

Closely related concepts [edit]

Recovery time (or estimated time of repair (ETR), also known as recovery time objective (RTO) is closely related to availability, that is the total time required for a planned outage or the time required to fully recover from an unplanned outage. Another metric is mean time to recovery (MTTR). Recovery time could be infinite with certain system designs and failures, i.e. full recovery is impossible. One such example is a fire or flood that destroys a data center and its systems when there is no secondary disaster recovery data center.

Another related concept is data availability, that is the degree to which databases and other information storage systems faithfully record and report system transactions. Information management often focuses separately on data availability, or Recovery Point Objective, in order to determine acceptable (or actual) data loss with various failure events. Some users can tolerate application service interruptions but cannot tolerate data loss.

A service level agreement ("SLA") formalizes an organization's availability objectives and requirements.

Military control systems [edit]

High availability is one of the primary requirements of the control systems in unmanned vehicles and autonomous maritime vessels. If the controlling system becomes unavailable, the Ground Combat Vehicle (GCV) or ASW Continuous Trail Unmanned Vessel (ACTUV) would be lost.

System design [edit]

Adding more components to an overall system design can undermine efforts to achieve high availability because complex systems inherently have more potential failure points and are more difficult to implement correctly. While some analysts would put forth the theory that the most highly available systems adhere to a simple architecture (a single, high quality, multi-purpose physical system with comprehensive internal hardware redundancy), this architecture suffers from the requirement that the entire system must be brought down for patching and operating system upgrades. More advanced system designs allow for systems to be patched and upgraded without compromising service availability (see load balancing and failover).

High availability requires less human intervention to restore operation in complex systems; the reason for this being that the most common cause for outages is human error.[10]

Redundancy is used to create systems with high levels of availability (e.g. aircraft flight computers). In this case it is required to have high levels of failure detectability and avoidance of common cause failures. Two kinds of redundancy are passive redundancy and active redundancy.

Passive redundancy is used to achieve high availability by including enough excess capacity in the design to accommodate a performance decline. The simplest example is a boat with two separate engines driving two separate propellers. The boat continues toward its destination despite failure of a single engine or propeller. A more complex example is multiple redundant power generation facilities within a large system involving electric power transmission. Malfunction of single components is not considered to be a failure unless the resulting performance decline exceeds the specification limits for the entire system.

Active redundancy is used in complex systems to achieve high availability with no performance decline. Multiple items of the same kind are incorporated into a design that includes a method to detect failure and automatically reconfigure the system to bypass failed items using a voting scheme. This is used with complex computing systems that are linked. Internet routing is derived from early work by Birman and Joseph in this area.[11] Active redundancy may introduce more complex failure modes into a system, such as continuous system reconfiguration due to faulty voting logic.

Zero downtime system design means that modeling and simulation indicates mean time between failures significantly exceeds the period of time between planned maintenance, upgrade events, or system lifetime. Zero downtime involves massive redundancy, which is needed for some types of aircraft and for most kinds of communications satellites. Global Positioning System is an example of a zero downtime system.

Fault instrumentation can be used in systems with limited redundancy to achieve high availability. Maintenance actions occur during brief periods of down-time only after a fault indicator activates. Failure is only significant if this occurs during a mission critical period.

Modeling and simulation is used to evaluate the theoretical reliability for large systems. The outcome of this kind of model is used to evaluate different design options. A model of the entire system is created, and the model is stressed by removing components. Redundancy simulation involves the N-x criteria. N represents the total number of components in the system. x is the number of components used to stress the system. N-1 means the model is stressed by evaluating performance with all possible combinations where one component is faulted. N-2 means the model is stressed by evaluating performance with all possible combinations where two component are faulted simultaneously.

Reasons for unavailability [edit]

A survey among academic availability experts in 2010 ranked reasons for unavailability of enterprise IT systems. All reasons refer to **not following best practice** in each of the following areas (in order of importance):[12]

1. Monitoring of the relevant components
2. Requirements and procurement

3. Operations
4. Avoidance of network failures
5. Avoidance of internal application failures
6. Avoidance of external services that fail
7. Physical environment
8. Network redundancy
9. Technical solution of backup
10. Process solution of backup
11. Physical location
12. Infrastructure redundancy
13. Storage architecture redundancy

A book on the factors themselves was published in 2003.[13]

Costs of unavailability [edit]

In a 1998 report from IBM Global Services, unavailable systems were estimated to have cost American businesses $4.54 billion in 1996, due to lost productivity and revenues.[14]

See also [edit]

- Disaster recovery
- Fault-tolerance
- High-availability cluster
- Overall equipment effectiveness
- Reliability, availability and serviceability (computing)
- Reliability engineering
- Resilience (network)
- Ubiquitous computing

Notes [edit]

1. ^ Using 365.25 days per year. For consistency, all times are rounded to two decimal digits.
2. ^ See mathematical coincidences concerning base 2 for details on this approximation.
3. ^ "Twice as much" on a logarithmic scale, meaning two *factors* of 2: $\times 2 \times 2 < \times 5$

References [edit]

1. ^ Floyd Piedad, Michael Hawkins (2001). *High Availability: Design, Techniques, and Processes*. Prentice Hall. ISBN 9780130962881.
2. ^ Lecture Notes M. Nesterenko, Kent State University

3. ^ Introduction to the new mainframe: Large scale commercial computing Chapter 5 Availability IBM (2006)
4. ^ IBM zEnterprise EC12 Business Value Video at *youtube.com*
5. ^ *Precious metals, Volume 4*. Pergamon Press. 1981. p. page 262. ISBN 9780080253695.
6. ^ *PVD for Microelectronics: Sputter Desposition to Semiconductor Manufacturing*. 1998. p. 387.
7. ^ Murphy, Niall Richard; Beyer, Betsy; Petoff, Jennifer; Jones, Chris (2016). *Site Reliability Engineering: How Google Runs Production Systems*. p. 38.
8. ^ Josh Deprez (April 23, 2016). "Nines of Nines".
9. ^ Evan L. Marcus, *The myth of the nines*
10. ^ "Top Seven Considerations for Configuration Management for Virtual and Cloud Infrastructures". Gartner. October 27, 2010. Retrieved October 13, 2013.
11. ^ RFC 992
12. ^ Ulrik Franke, Pontus Johnson, Johan König, Liv Marcks von Würtemberg: Availability of enterprise IT systems – an expert-based Bayesian model, *Proc. Fourth International Workshop on Software Quality and Maintainability* (WSQM 2010), Madrid, [1]
13. ^ Marcus, Evan; Stern, Hal (2003). *Blueprints for high availability* (Second ed.). Indianapolis, IN: John Wiley & Sons. ISBN 0-471-43026-9.
14. ^ IBM Global Services, *Improving systems availability*, IBM Global Services, 1998, [2]

External links [edit]

- Lecture Notes on Enterprise Computing University of Tübingen
- Lecture notes on Embedded Systems Engineering by Prof. Phil Koopman
- Uptime Calculator (SLA)

Categories: System administration | Quality control | Applied probability | Reliability engineering | Measurement