Measuring Real World Data Availability

Larry Lancaster and Alan Rowe – Network Appliance, Inc.

ABSTRACT

This paper examines how server marketing claims of high reliability (e.g., 99.999%) stack up against real world data measurements. Our goals were to: measure discretionary NFS data availability, compare data availability between standalone and clustered systems, and draw some conclusions about best practices for customers.

We explain our methodology for measuring, filtering, and categorizing availability-related data. Through careful data and error analysis, we arrive at discretionary NFS data availability estimates for NetApp filers in the real world. We conclude that NetApp clusters provide over four-nines availability in the field.

Introduction

The server marketing world is full of claims of high availability. These claims usually take the form of "3-nines" (99.9% availability), "4-nines" (99.99% availability), or even "5-nines" (99.999% availability) [Rich99]. One is left to wonder at how these claims compare to real-world availability.

Vendors are known to partner in a marketing relationship with the purported goal of providing 5-nines availability. However, even then the "guaranteed" availability can be much lower [HP01], and the quantitative basis on which the validity of the guarantee should be predicted or disputed remains unclear.

Among compute server vendors recently, Microsoft commissioned the Aberdeen Group to perform a study which was based upon real-world data [Aber01]. A Microsoft-developed system monitoring tool collected outage data from systems at various Microsoft customer sites.

Data from 10 companies was considered in the Microsoft study, for a total of about 330 observed system-years. The result was that the hand-picked set of customers were seeing about 99.96% availability in the field. Whether and on what basis customers, systems, and/or outages were excluded remains unclear.

Among dedicated file server vendors, however, we could find no recent analogous study of real-world data availability. While similar studies may have been commissioned and carried out, they may not have been made public. We believe that a sound empirical basis for studying file server data availability is necessary to establish, track, and refine product quality.

Since October 1999, Network Appliance has been shipping versions of DataONTAP (our proprietary file server OS) with availability metrics support. This development has allowed us to perform a trailing-year analysis of NFS data availability across all systems running the appropriate releases and configured to send the data to Network Appliance.

Goals

Our goals for this study were threefold. First, we wanted to reliably measure discretionary NFS data availability,[1] within the complex and uncontrolled environment of the real world. We define *discretionary availability* for a product-service (the product being a NetApp filer and the service being NFS file service, in this instance) as one minus the fraction of time the service is unavailable due to the failure of any one or more component(s) of the product.

Discretionary availability is not impacted by intentional downtime or downtime due to external power failures, operator blunders, installation, testing, or moving. By "the real world" we mean the distribution of NetApp customer environments. This distribution includes countries and localities with poor power distribution, localities far away from parts depots and Customer Support centers, and customer testing facilities that aren't always knowable to us. We believe there is no more meaningful laboratory in which to measure a product's discretionary availability than the distributed laboratory of the product's customer base.

Second, we wanted to be able to compare data availability between standalone and clustered systems. Clustered configurations have been proposed and implemented for both general purpose servers [Barber97] and specialized file server appliances [Kleiman98]. The purpose of such an architecture is to provide high availability (HA) through failover capability. We wanted to see if the prediction of HA for clustered configurations was realized in real-world environments.

And third, we wanted to be able to draw some conclusions about best practices for customers. This is an ongoing study for us, but we wanted to gather at least some basic information immediately.

Methodology

Real-World Data Collection

DataONTAP has a feature called Autosupport. When enabled, the file server will send an email whenever something noteworthy happens (including after an outage), and weekly no matter what. The

[1]We support both NFS, CIFS, and multi-protocol NFS plus CIFS. For our first study, we chose systems running at least NFS. Other studies are ongoing.

customer can choose to have emails sent back to Network Appliance and/or internally within his/her own company.

If they're sent to Network Appliance, and they're from systems running recent enough releases to give us the availability-related ("availtime") data we need, they become part of our sample pool of raw emails.

SMTP is not an entirely reliable transport. Even if it were, there are sometimes network problems or filer problems surrounding an outage, such that an email may not be properly generated or received at or around that time. We needed to design a data collection and reporting system which would work around these kinds of difficulties.

Availtime Deltas

Availtime data consists of a set of cumulative counters. Relevant to this study are the cumulative seconds since NFS was first licensed, and the cumulative seconds of unplanned NFS downtime since NFS was first licensed.

Because availtime counters are cumulative, the availtime information from consecutively received pairs of emails can be differenced to tell how much time passed between generation of the emails, generation of the emails, and during how much of that time the service was unavailable. The resulting pairs of differences are called "availtime deltas," and form the basis for a rigorous event-based model of availability. Each delta is associated with a pair of consecutively received emails from the same system.

Logs and Configuration Information for Diagnosis

In addition to availtime data, autosupport data contains detailed configuration information and cumulative system logs to help us automatically diagnose system failures, if any, associated with an outage or a set of outages. Those emails can be examined to see what was reported in the portion of the log new from one received email to the next, and whether any configuration information changed over that period of time.

Because system logs are cumulative and are regularly sent to Network Appliance on a weekly basis, clues to the cause of an outage will still be available in the log when the next email is received. System configuration information between consecutively received emails can still be compared on a before-and-after basis to determine if parts were swapped or other configuration items were changed.

Data Filtering

In order to measure discretionary availability as defined in the previous section, we need to filter out intentional outages, installation, testing, and moving. We also need to be able to deal with garbage data.

To filter out intentional outages, we consider all *unplanned* NFS outages in this study – that is, outages where the operator did not intentionally reboot or halt the machine, unless he specified that a core be generated (a diagnostic reboot or halt is considered unplanned). This distinction is made by the filer in recording the availtime data, so that no post-processing is required.

An unplanned outage is a serious event. Simple failures do not cause these. For example, a single disk failure just causes a spare disk to be assigned to the RAID group instead; memory and NVRAM has ECC logic; systems have dual power supplies and multiple fans; single motherboard or adapter failures can be dealt with by clustering; disk Fibre Channel loop failures can be handled by dual-pathing the disks, and so on. That's why it's important for us to examine data relating to all unplanned outages, in order to determine the cause of failure.

Next, we filter out garbage data with a set of basic sanity checks to insure that the availtime deltas don't defy reason. For example, setting the clock forward or backward by a year – something that happens more frequently in the field than one might at first imagine – might cause the total time difference to be either 53 or -51 weeks.

Filtering out installation, testing, and moving is a more difficult proposition. One step we take to address installation and related testing is to exclude from analysis availtime deltas from any system for the first 28 days after NFS is first licensed. This is hardly a sufficient measure. NFS might be licensed on a demo-eval unit, or as part of the manufacturing test process. To do a better job, we introduce the concept of an "availtime system."

Availtime Systems

An availtime system can be thought of as an extended system key which uniquely identifies a system with reference to where it is and how it is deployed. The components of the key include system ID, OS version, domain name, hostname, and clustered indicator (indicating whether the system is one node in a cluster). One unique combination of these values comprises an availtime system, and only availtime deltas between consecutively received emails from the same availtime system are included for analysis.

The domain name and the hostname insure that systems which are installed, are moved, or change owners don't generate input to the analysis. Changes in the other keys help prevent the inclusion of availtime deltas related to testing of new system configurations.

Testing is not completely screened out by this filtering process. Some customers are known to have dedicated machines just for the purpose of testing, and we have no systematic way of excluding these systems.

Other Sources of Availtime Data Exclusion

We do not filter based upon customer, system, location, filer model, release, or system management practices for the purposes of this study. The real world is a rough and nasty place. Literally, if a filer died and the customer decided ‘‘it's Friday, I'm tired, I'll fix the problem on Monday,’’ a three day outage would be included in this study.

All machines are not equivalent products, or in identical settings, other than being NetApp filers serving NFS. All eligible availtime deltas from all eligible availtime systems are included in this analysis. As such, the results of this study apply as generally as possible to our products and our customers.

It is possible that a customer chooses not to enable the Autosupport mechanism, or does not run a DataONTAP release recent enough to support availability analysis. In either case, we obviously cannot include their data in this study.

Failure Categorization

In order to measure discretionary availability as defined in the previous section, we need to be able to determine which outages fall into the categories of operator error and power failure, and which were specifically related to the product-service. We took as a starting template for outage categorization a study done by Jim Gray [Gray90].

Our study differs in some important ways, though, so we modified our outage categories as appropriate. The categories we use are disk subsystem hardware, non-disk subsystem hardware, software, operator, power, and likely power failure.

Our algorithms for attributing the cause of an outage to one of these categories include scanning the cumulative system logs, examining the system configuration information before and after the outage, and looking for site-wide events by correlating availtime deltas and system-logged information across multiple systems at the same site.

When scanning the system logs we can find each of the downtime events within the log, and look both back (earlier) and forward (later) in the logs from that point to find more information about the outage. When examining configuration information from before and after the outage, we can determine if the customer had misconfigured something, or if a part was swapped out during that time. When correlating events across multiple systems at the same site, we can determine that a site-wide outage took place.

Examples of Outage Classification

Disk subsystem hardware failure is a category of failure which affects access to multiple disks in the same RAID group. Typical causes include loss of an adapter or cable or one of the disk shelf circuits in a non-clustered or non-dual-pathed configuration. We can detect this by looking for messages in the system log which indicate failure of the loop, such as:

```
Wed Oct 11 09:32:47 GMT [isp2100_timeout]:
Offlining loop attached to Fibre Channel
adapter 4.
```

Non-disk subsystem hardware failure includes all other hardware failures. We can detect these failures by syslog messages which are specific to hardware failures; for example, a bad (and uncorrectably bad) memory chip:

```
Fri Oct 13 13:45:14 GMT [callout]:
1 UNCORR PROC ERR 86 eia:87a0000000
fs:005d eis:c5000000 isr:100000000 ...
CPU ECC error on DIMM J28 at
address=0xa000000f, bit=39.
```

Software failure is visible when we examine the logs and detect either that the software panicked, or that the administrator deliberately created a core file:

```
Fri Feb  9 01:44:16 GMT
[savecore_admin:info]:saving 1048M
to /etc/crash/core.0.nz (assertion
"!sk_is_mp_mode || (sk_n_domains == 2
|| _sk_own_domain(procp->domain))"
failed...)
```

Power failure consists of two different categories. One is **definite** power failure, which we can detect by observing that multiple systems suffered a dual power supply failure at the same site at about the same time. The other is a **likely** power failure, which we can recognize by noting that both power supplies failed on the system at about the same time.

Operator failure includes outages due to mis-administration. Our system is an appliance and so is much less susceptible to mis-administration than the general-purpose computers noted in [Gray90]. However, operator failures do exist in real-world data.

For example, one way we can detect an operator failure is to note that both power supplies failed on a system at about the same time, and once the system came back up, parts unrelated to the power supplies had been swapped. This observation indicates that an administrator pulled the plug on a machine in order to swap hardware, without first performing a clean shutdown.

Waiting for Parts and Customer Deferral

In order to measure discretionary availability as defined in the previous section, we need to be able to determine how much of the hardware- and software-related downtime as measured using the availtime data gathered through Autosupport is specifically related to the product-service.

There are two significant downtime factors which cannot be measured through Autosupport-derived data, but which must be factored out to estimate discretionary data availability. The first is the amount of time spent waiting for parts to arrive at a customer site. The second is the amount of time the customer chooses to wait because he/she regards the outage as uncritical, and taking care of the problem immediately would be inconvenient.

All outage-related data from our Customer Satisfaction department relating to the same physical systems and the same trailing year period as the Autosupport data were thoroughly examined and audited over a six-month period, case by case, to determine the portions of downtime that were attributable to product failure and diagnosis, customer deferral, and waiting for parts.

This data was collected and audited in a painstaking manual process, and so can not be mapped one-to-one with availtime events. However, the overall ratio is the best available estimator for the product-related proportion of downtime. We can use this result in conjunction with the availtime-based product-related availability figures to estimate discretionary availability.

Data and Analysis

Categorization of Empirical Availtime Data

Over the trailing year ending April 28, 2001, the availtime deltas forming the sample pool for this study spanned 4400 observed system-years. We summarize the empirical data by showing the relative impact on availability from each of our downtime categories: disk subsystem hardware, non-disk subsystem hardware, software, operator, power, and likely power failures. Since one of the goals of this study was to look at the differences in discretionary availability between systems in clustered and standalone configurations, we broke out the data along these lines, as well.

We compare results for the entire sample base against results for a "select" subset of systems, thought to be well-managed production systems at large, international, well-known customer sites each with 100 or more systems in both standalone and clustered configurations. We also compare these results against results for one particularly large customer, whose site we consider to be the best-managed overall, with frequent on-site visits from NetApp Customer Support representatives. The empirical data are summarized in Figure 1.

Error Analysis

To continue forward and obtain meaningful results, we need a basic model of the data to determine how to proceed. In particular, it is our goal to estimate discretionary filer-NFS data availability for the population of Network Appliance filers, in both clustered and standalone configurations, and to have some idea of the uncertainty in these estimates.

Our sample pool of availtime deltas contains about 4400 system years worth of data. However, these data are reported by only a small portion of our entire customer installed base, on a self-selecting basis as described in the previous section. And, we're only looking at these systems for a one year period. We'd like to be able to estimate the mean availability across the population of either standalone or clustered filers, at large and in general. As discussed in [Burgess00], we need a measure of standard error (SE) to tell us how accurate our estimates are.

There are a number of sources of measurement error in our estimates, as well. By virtue of the way

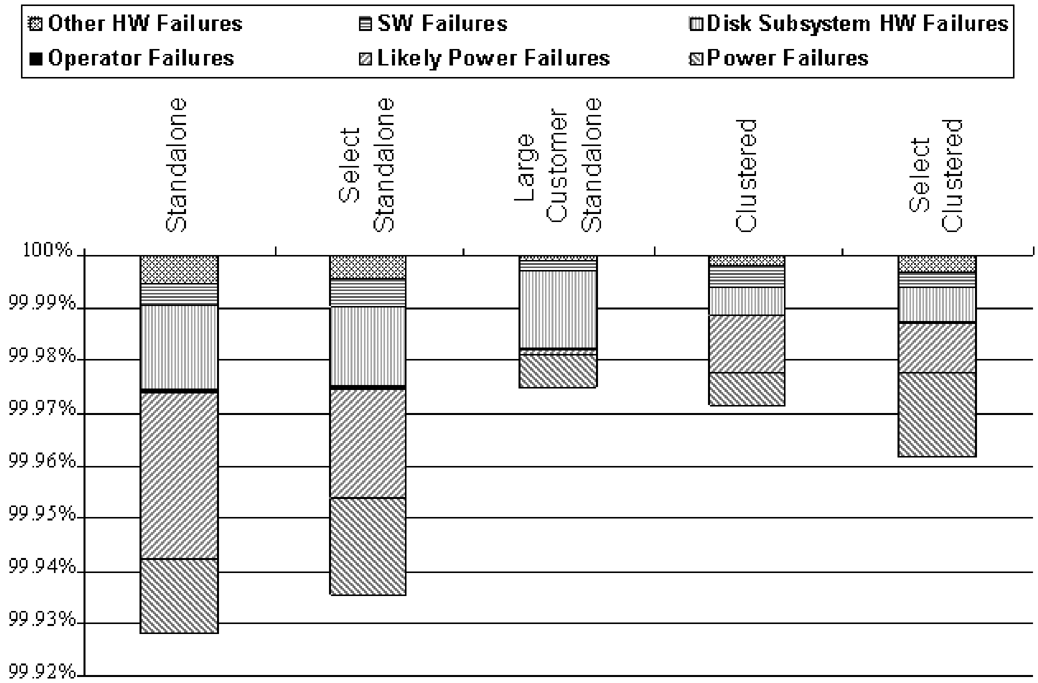


Figure 1: Empirical data categorization: availability impact by category. This graph shows the results of our outage categorization, and the overall impact of each category on the measured availability. We were surprised to find that operator failure accounts for a trivial portion of overall downtime. Note the size of the power failure categories: a UPS is worth the expense, if the goal is an HA environment. Note also the dramatic decrease in HW-related downtime when moving from a standalone to a clustered configuration.

DataONTAP availability metrics support is designed, for example, the tabulated availtime deltas have an unbiased ten second granularity. Also, as described in the previous section, our estimate of the product-related portion of downtime – relative to waiting for parts and customer deferral – was obtained by hand on a case-by-case basis, making it subject to error and rounding. All of these sources of random error can be addressed with a SE, as well.

However, calculating a meaningful SE on an estimate of population availability is much more difficult than one might at first imagine. We know that the availtime deltas which comprise our sample pool are not independent and identically distributed (IID). The availtime deltas contributed by any one system tend to be highly dependent among themselves.

Empirical outage durations and availabilities are far from normally distributed, both at the availtime delta level and when rolled up to the availtime system level. We have very little in the way of a parametric model for calculating uncertainty in our estimate. Taking these facts together, there is no quick and easy formula to obtain SEs for our population estimates that would be valid in this case.

The Bootstrap

We turn to a bootstrap [Efron93] resampling procedure to calculate our SEs. With a very basic and sparse set of assumptions, we can reasonably compute the SEs of our measurements.

Recall that an availtime delta is a vector containing both the difference in total time and the difference in downtime between consecutively received Autosupport emails. Let D_i represent the sum of all availtime deltas for one availtime system over the course of the observation period. Here, $(i: 1 \cdots N)$ indexes the N availtime systems comprising our sample pool, S_0. The sample availability parameter of interest $A(S_0)$ is calculated as:

$$A_0 = 1 - \frac{\sum_i Ddown_i}{\sum_i Dtotal_i}$$

where $Ddown_i$ is the downtime component of D_i and $Dtotal_i$ is the total time component of D_i.

Assume that the D_i are random vectors IID w. r. t. some unknown probability distribution P. (This assumption is strongly supported by the qualitative similarity of the empirical hardware + software data availability components between the "select" systems and "all" systems, as seen in Figures 1 and 3.)

The bootstrap resampling procedure, then, is as follows. Construct a sequence of new sample pools, S_j, indexed by $(j: 1 \cdots M)$. Each S_j is created by sampling N of the D_i at random, with replacement, from S_0. For each S_j, compute the sample availability parameter, $A_j(S_j)$. Then, compute the unbiased SE of the $A_j(S_j)$ over j. The bootstrap lets us substitute the resulting SE (as N and M grow large) for the SE which would be seen if the S_j were drawn from P rather than from S_0.

Using the same basic technique with minor modifications, we can derive SEs for the portion of

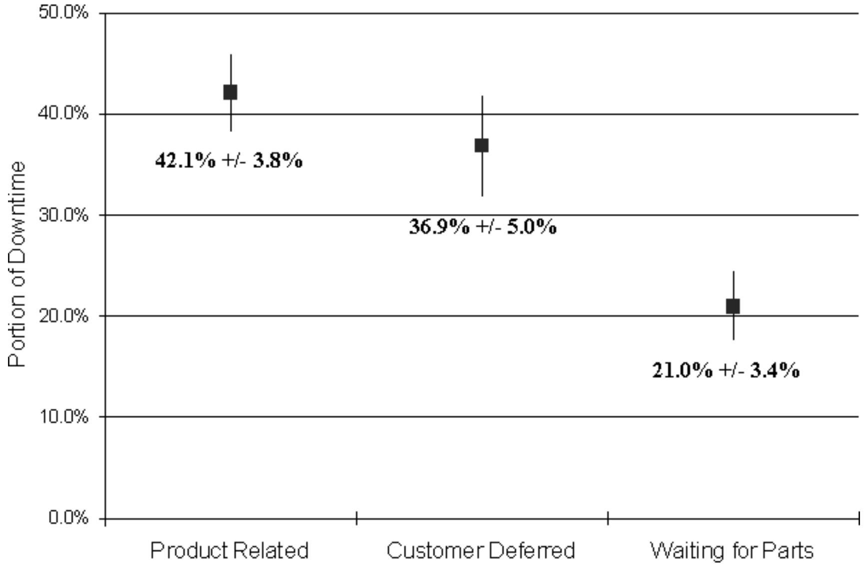


Figure 2: Portions of downtime due to product failure, waiting for parts, and customer deferral. This graph shows a categorized distribution of downtime, based upon a study of Customer Support case records over the same period of time as the rest of this study. Bootstrapped SE bars are also included (M = 5000 resamples). These results are used to obtain the conditional probability that a unit of downtime is product related, for the purpose of estimating discretionary availability. Note the large amount of "Customer Deferred" downtime: any field study of availability will be confronted with factoring this out. Also note the size of the "Waiting for Parts" category: this shows that customers would do well to keep a stock of commonly-used parts on-hand.

availtime-calculated downtime which is actually due to waiting for parts and customer deferral (see Figure 2).

The bootstrap is not without its weaknesses. For example, outliers captured in S_0 will tend to exaggerate the calculated SE. But the effect of this problem is to understate, rather than overstate, certainty in the estimate of $A(P)$. In other words, it leads to more conservative reporting. We wholeheartedly recommend the bootstrap technique to those seeking a measure of uncertainty around estimates when a classical determination of such a measure is not readily available.

Discretionary Data Availability

Beginning with the empirical data summarized in Figure 1, we factor out power and operator failures, leaving all hardware-and software-related outages. This extraction takes us as close as possible to an estimation of discretionary availability using availtime data alone.

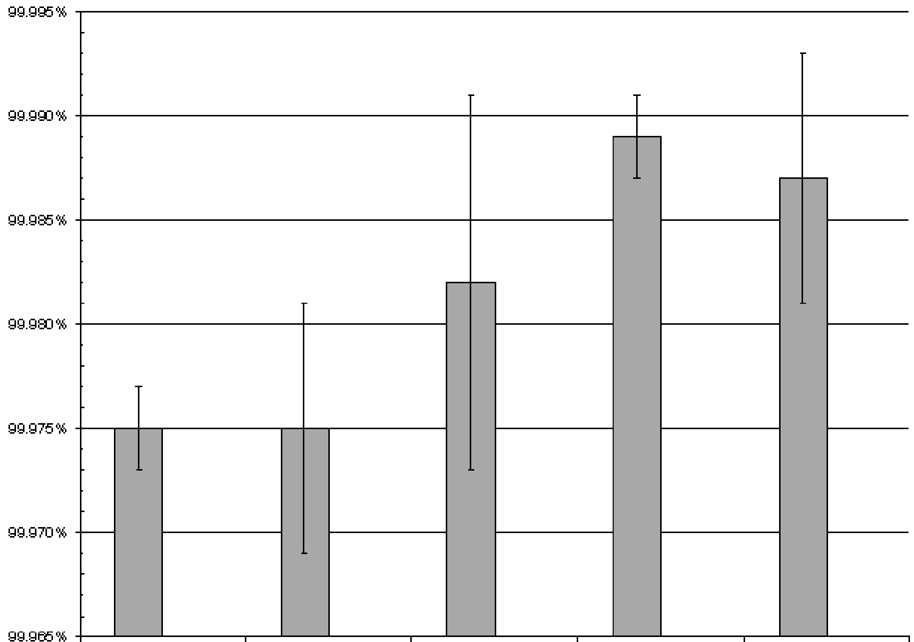
We then apply the bootstrap resampling procedure to compute SEs (M = 5000 resamples), and display the results in Figure 3. We already have the results of the Customer Support analysis of waiting for parts and customer deferral in Figure 2, also with bootstrapped SEs (M = 5000 resamples). It remains only to combine these results into a cohesive whole.

We proceed as follows: the availtime-derived availability A_a, as shown in Figure 3, is equal to $1 - P(down)$, where $P(down)$ is the probability that a system-service will be down in any small sliver of time due to product failure, waiting for parts, or customer deferral. The probability that a system-service is down in any small sliver of time due to product failure, given that the system is down, is shown by the first data point in Figure 2: call this $P(prod|down)$.

What we want is the discretionary availability A_d, which is given by $1 - P(prod)$, where $P(prod)$ is the probability that a system-service will be down in any small sliver of time due to product failure. Taking these probabilities as independent, Bayes' rule tells us that $A_d = 1 - P(prod) = 1 - P(down)P(prod|down)$.

To get an SE for A_d, we notice that $SE(A_d) = SE(P(down)P(prod|down))$. We can then solve for $SE(A_d)$ by using the well-known technique for propagating SEs of two independent random variables into their product, so that

$$\frac{SE^2(A_d)}{E^2(1 - A_d)} \approx \frac{SE^2(A_a)}{E^2(1 - A_a)} + \frac{SE^2(P(prod|down))}{P^2(prod|down)}.$$



	All Standalone	Select Standalone	Large Customer Standalone	All Clustered	Select Clustered
Availability	99.975%	99.975%	99.982%	99.989%	99.987%
+/- SE	0.002%	0.006%	0.009%	0.002%	0.006%

Figure 3: NFS data availability from availtime data (includes waiting for parts and customer deferred downtime). This graph shows the estimates of NFS data availability gotten from availtime data alone, for different sets of availtime systems. Bootstrapped SE bars are also included (M = 5000 resamples). This graph excludes the operator failures and power outages depicted in Figure 1. Note the overlap of the "All" and "Select" categories, for both clustered and standalone configurations. This implies that data availability can reasonably be considered a property of the population. Note also that the difference between aggregate clustered and standalone configuration availabilities is highly significant. This difference demonstrates that clustering is an effective tool for achieving HA.

Here, $E(.)$ denotes the expectation (average). The results are displayed in Figure 4. This completes our estimation of real-world filer-NFS discretionary data availability.

Conclusions and Direction

It is possible to track discretionary product-service availability in the real world: We believe that the similarity between the results for all sites and the results for "select" sites in Figure 4 also demonstrates that the procedures we're using to generate, collect, filter, and aggregate the data are giving a fair and objective view of discretionary availability.

NetApp clusters provide 99.99% availability: Figure 4 shows that nodes in a clustered configuration provided greater than 4-nines discretionary NFS data availability in the field. Nodes in a standalone configuration are shown to be within the margin of error of 4-nines, as well.

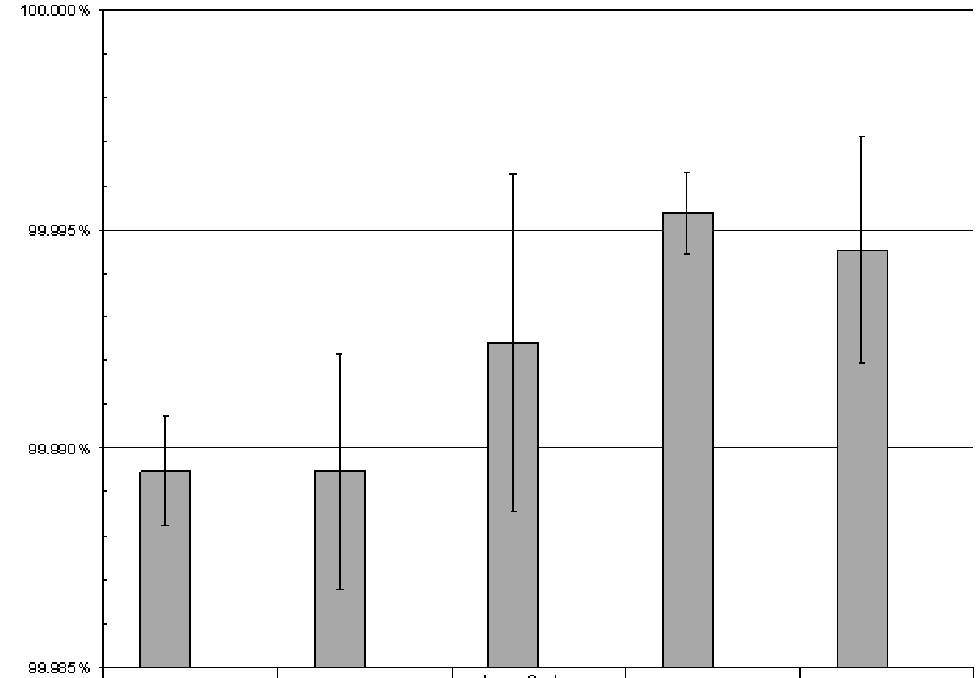
A clustered configuration does provide higher data availability in the real world: As Figure 4 shows, clustered systems provided significantly better availability than standalone systems. A common rule-of-thumb for determining whether two sample means are different, given rigorously measured SEs, is to check whether the error bars of the two measurements are mutually exclusive. In this case, they are separated by this distance twice over.

This observation fits with prediction, since the point of a cluster is to allow one head to serve data for its own and another data node while the second head is down. This results in only a brief non-discretionary outage during takeover, during which NFS file service would be unavailable.

Redundancy of paths to disks also provides higher data availability. All clustered nodes have that while most unclustered nodes do not, since the feature was only introduced within the timeframe measured here.

All customers can achieve enterprise-level availability: As Figure 4 shows, and as we were surprised to find, availability is not significantly separated between "select" systems and all systems in either a clustered or a standalone configuration. We believe that this uniformity reflects the ease-of-use and productized reliability that are goals of the appliance concept.

Figure 1 supports this notion. Operator failure represents a trivial portion of the overall downtime, suggesting that ease-of-administration is key to high data availability in the real-world.



	All Standalone	Select Standalone	Large Customer Standalone	All Clustered	Select Clustered
Availability	99.989%	99.989%	99.992%	99.995%	99.995%
+/- SE	0.001%	0.003%	0.004%	0.001%	0.003%

Figure 4: Filer-NFS discretionary data availability: population estimates. This graph shows the estimated filer-NFS discretionary data availability, for different sets of availtime systems. The values are derived from the data shown in Figures 2 and 3. SE bars are also included, derived from those in Figures 2 and 3 using classical error propagation analysis. Note that the difference between aggregate clustered and standalone configuration availabilities remains highly significant after adjustment for discretionary outages. This difference demonstrates that clustering is an effective tool for achieving HA.

Power failures are common – use a UPS: As Figure 1 shows, power failures are commonplace in the field. We believe that a UPS is a critical component of an HA environment.

Parts failures can cause unnecessary wait – keep a stock of parts: Availability afforded the large customer depicted in Figures 1, 3, and 4 is qualitatively higher than that afforded other customers running standalone configurations, although the margin of error is high due to the relative size of the sample.

This large customer subscribes to the NetApp Global Advisor level of support. The advisors for this customer have recommended that the customer keep a large stock of spare parts on-site. Because about one-third of non-discretionary downtime can be attributed to waiting for parts (see Figure 2), we believe that one significant factor in this customer's success has been to avoid this delay.

More Global Advisor recommendations have evolved and spread to other sites since April 28, 2001. We look forward to examining the effects of these measures on availability in the year to come.

Future Direction

We have only just begun to make sense of this vast wealth of data. Future efforts will include further characterization of the causes of downtime at the availtime event level and enhancements to the information sent back to NetApp through Autosupport.

We also look forward to examining the effects of various measures we take to help improve discretionary data availability, and widening our scope to include various types of planned outages. Ultimately, we hope this study will help to change the competitive landscape by forcing our competitors to compete using rigorous availability figures based on data, not just hype

Author Information

Larry Lancaster earned a couple of degrees at Berkeley in the 1990s, took an LOA from grad school in 1998, worked for Decision Focus developing revenue maximization systems, and is now a tools programmer and metrics analyst at Network Appliance. Reach him at larryl@netapp.com .

Alan Rowe got a couple of degrees in Scotland in the late 60's, worked for Plessey on a capability machine, emigrated to Canada to work for Bell-Northern on the DMS digital switch, emigrated to the US to work for Tandem on the NonStop fault-tolerant system, and is now Technical Director at Network Appliance, where he cares about Available, Simple, and Fast file server appliances. He plans no further emigrations. Reach him at rowe@netapp.com.

References

[Aber01] Aberdeen Group, Inc., *Proving-the-Point: Interviews with Next – Generation Windows 2000 dot.com*, Microsoft Executive White Paper Updated – for Electronic Redistribution Only: http://www.microsoft.com/windows2000/server/evaluation/news/reviews/dotcoms.asp, Redmond, Microsoft, Feb. 6, 2001.

[Barber97] Barber, M. R. "Increased Server Availability and Flexibility through Failover Capability," *Proceedings of the Eleventh Systems Administration Conference*, Berkeley, USENIX Association, p. 89, 2001.

[Burgess00] Burgess, M., *Principles of Network and System Administration,* Chichester, Wiley, 2000.

[Efron93] Efron, B., R. J. Tibshirani, *An Introduction to the Bootstrap*, New York, Chapman and Hall, 1993.

[Gray90] Gray, J., *A Census of Tandem System Availability: 1985-1990,* Tandem Computers TR 90.1, January, 1990.

[HP01] Hewlett-Packard, Inc.,"Helping Companies Across the Globe Deploy Highly Available Solutions," HP-UX High Availability Software Home (as of June 4, 2001), http://www.hp.com/products1/unix/highavailability/5nines/5n5mbrief.pdf .

[Kleiman98] Kleiman, S. R., S. Schoenthal, A. Rowe, S. H. Rodrigues, A. Benjamin, "Using NUMA Interconnects to Implement Highly Available File Server Appliances." *Proceedings Hot Interconnects*, Vol. 6, August, 1998, and *IEEE Micro*, Jan./Feb. 1999.

[Rich99] Richards, D. "EMC drops a container load of products onto the storage market," *Computer Reseller News*, http://www.itnews.com.au/crn/news/008_0304i.htm, Mar. 11, 1999.