A Target of Five Nines is Appropriate for Internet Services
CS 553 Position Paper - I

1 Introduction

Importance of Internet Services The growth in scale and utility of the Internet has been explosive starting the early 1990s. The commercialization of the Internet and the arrival of convenient tools such as the search engines and the browsers have made the virtual stores, libraries, museums, and learning institutions as navigable to a cyberuser as a car user would navigate the parking lots of the real world shopping malls. The result being that more and more of one's daily activities is done online. Consider me, I order my airline e-tickets, groceries, music, and apparel online, pay my bills and manage my bank accounts through web interfaces, and in in fact it irks me that my landowner does not have a *Paypal* account! It is no wonder then that the WWW has turned into a multi-billion dollar industry, and every company, including Joe-SixPack is rushing towards *e-tailors* such as IBM, to put up a web-window to their clients.

The other comparable predecessor technology that collapsed distances between interacting people, the PSTN (Public Switched Telephone Network) is an excellent example which provides us insight into the success of Internet services, and also what would sustain this phenomenon. One of the main reason for the popularity of the PSTN apart from its utility is that its available most of the time. Phone companies report that their services are down for less than an hour every year! Though they make various simplifying assumptions in their calculations, such as not considering faulty instruments and cables on the client side , this degree of availability is truly commendable. Would people be such patrons of the phone services if the service we not available for some portion of the day, everyday? Unlikely. Hence availability of a service is truly an important factor in determining its success.

So the question now is, "Do Internet services provide the same availability as the PSTN, and if better or worse why so?" Various studies [6, 2] have shown that current Internet services provide anywhere between two to three nines of availability, which amounts to downtimes of 5 hours to 2 days per year. This is two orders of magnitude worse than the PSTN!

Why is the PSTN more available? The PSTN has been around now for more than a few decades and over these years it has mostly standardized its services. Apart from the addition of a few extra services such as mailboxes, call-forwarding, three-way calls, etc., the basic service has remained identical to that provided say 50 years ago. Phones which worked back then, can still be used in the analog mode to call people without any problems. This standardization of service enables: a better understanding of the service, standardization of hardware and software components and intimate knowledge of its weak points. This in turn results in being able to maintain the system to provide higher availability. In contrast todays Internet services provide diverse services, dished out through a myriad combination of hardware and software components, which are put together in any of numerous configurations. In my opinion, this inherent variety, complexity and non-standardization of todays Internet services are the major culprits in the crime of low availability. To provide a flavor for complexity and variety in such services, [6] presents a case study of three services, and they differ in size - varying from 500 to 2000 nodes, type of machines - while some use Sun's Sparc others use IBM-PC clones, type of operating system - copyright Solaris to open-source Linux-like OSes, and not to mention vastly different arrangement of these nodes and their corresponding interactions. One of the results of this diversity is that good practices (e.g., for high availability) in one service is not necessarily directly applicable to others in the industry, while this is mostly true for the PSTN. This self-feeding cycle in the case of PSTN has enabled it to achieve high availability, in contrast to its unfortunate successor. However, the Internet industry is still in its growing years, and there's much that can be done to improve the availability scenario.

Before we get into arguing how many *nines* of availability Internet services should aim for, let us briefly walk-through what the "nines" imply.

System of 9s Availability of systems is expressed as a percentage of time the system delivers correct service. If you view the system as servicing independent requests from clients, the availability can be expressed as the percentage of requests that were serviced correctly. Industry often expresses availability of components and services as *number of nines*. A service with 2 nines availability delivers correct service for 99% of the time the system is scheduled to be operational, and is said to be in class 2 set of services. Table 1 tabulates the relation between the *class* of nines and the corresponding downtime experienced by each. Note that all reported availabilities do not take into consideration the time for *scheduled* repairs or maintenance, and includes only unscheduled outages.

Class	Availability %	Downtime/year
1	90	5 weeks
2	99	3.6 days
3	99.9	530 min
4	99.99	53 min
5	99.999	5.3 min
6	99.9999	30 secs

Table 1: *Availability classes and corresponding downtimes per year of operation.*

My Take on Availability in Internet Services In this paper, I will present justification for Internet services to strive for high availability equivalent to that of the PSTN. Through a study of the losses incurred by Internet services due to unavailability, and a walk through of the additional costs incurred to deliver high availability, I will present evidence to suggest that a goal of five nines for these services is an appropriate one, in that it is necessary, beneficial and an achievable target. Before we move onto the defense of the above, I will first address some contradictions to my stand on the topic in the next section.

2 Countering the Antagonists

Who cares? Client systems are down most of the time anyway... This would be irrefutable if there was just one client access point used for accessing the service. A person even having the slightest idea of stochastic processes and queueing theoretical framework would never make such a statement about the current Internet service architecture. I agree that end-user machines are cheap and built out of commodity components, and certainly not built to work 24 X 7 X 365, and they may have an availability of 2 nines or less. However, the client base is so large, and so widely distributed, that it could only be the "Hand of God" intervening, if all the clients and service were to go down in a synchronizing manner. A set of 5 *independent* clients, each with just a single nine of availability, each accessing the same service, would be equivalent to a single client with 5 nines availability. With the scale of todays Internet reaching over 160 million hosts[9], distributed over 24 timezones, and whose interconnection structure is a multigraph, a person making the above claim would be absolutely misinformed.

Internet Services are too Complex! 99.999% for Internet services is a Fantasy! Well, XXX may have this on his slide presentation, where the topic of the presentation would be "Top Ten Reasons why Fault-Tolerance and High Availability should be Banned from Academic Research". Cheezy comments aside, we have to view this as a challenge for computer scientists. A recent study by our team at Panic Lab[7], showed that by applying simple, well-defined software and some well-understood hardware techniques allowed us to reach close to 4 nines availability for a simple Internet service. These were mostly accepted techniques in distributed systems for fault-tolerance, and I believe that further study of the specific environment of Internet systems (e.g., cluster-based systems would allow the application of systematic approaches to achieve 5 nines availability.

Before we get into a discussion of ”How we can achieve such high availability?”, a more important question is ”Is such high availability necessary at all?”. I will address these questions in the following section by providing an intuition about the cost of running services at lower availabilities.

3 Cost of Unavailability in Internet Services

Various emperical studies [5] have pointed to the sizeable loss in revenue attributed to downtimes in Internet services. The absolute losses vary from service to service, with brokerage firms standing to lose upto 6 million Euros per hour of downtime, while *less*-valued services such as airline reservations incurring upto a hundred thousand Euros[8] in lost transactions. The one hour of downtime would correspond to a class 4 system with 4 nines availability. So services which belong to class 4 or lower lose significantly in annual revenue. While this is substantial in itself, there are additional costs incurred as a result of service failures. The overall cost can be broken into three categories:lost transactions, equipment repair/replacement and customer disgruntlement. For the first category of lost transactions cost T_c, we have seen that its clearly quantifiable by observing one's own service, while for the other two the impact is not clearly quantifiable. The repair costs R_c, can include hardware replacement costs, purchase of improved software components (e.g., software upgrades), and if the components are not commodity then additional costs of bringing in field workers of the customized infrastructure provider(such as HP or IBM). While the cost of these depend on the severity of the failure, and could be specific to the particular Internet service, a reasonable estimate may be arrived at by looking at the MTTRs (as gathered from experience, previous empirical studies, and manufacturer guarantees). Finally, I believe an approximate model could be developed to determine losses incurred due to customer disgruntlement G_c. The latter would require the knowledge of parameters such as how frequently the customer accesses the service f, how long has the customer been using the service d, the correlation of probability that a client would switch services p with the downtime of the service D, and the loss in revenue incurred by the service L assuming an average customer would stick with the service for a period T. The overall cost of unavailability C_{UA} can be given by:

$$C_{UA} = T_c + R_c + G_c$$

$$where\ G_c = y(f, d, p, D, L, T)$$

The reason, I presented a discussion of the components comprising the overall cost, is to understand the relation between them and the total cost. While T_c clearly holds a linear relationship, the behavior of the other two are ambiguous. Repair cost, R_c in most failure scenarios can be thought of as a constant. However, looking at G_c, intuitively, p and hence G_c is directly proportional to the downtime D experienced by the service. As much as I would have liked to have workable equations for the latter two components, we will work with the intuition that the cost of unavailability is directly proportional to the downtime incurred by the service.

4 Cost of Achieving Ultra-Availability

Price to pay Gray's book[4] gives a good labeling of systems based on the class of availability, which also characterizes the effort that goes into achieving the corresponding availability. Class 1 systems are *unmanaged*, class 3 systems are *well-managed*, class 6 systems are *ultra-availability* systems. Though the term *management* is used to represent the effort/cost required to achieve higher availability, in reality it involves other contributing factors. The main contributors include: fault-tolerant hardware, additional software techniques, and increased maintenance costs. Fault-tolerant hardware costs H_c, encompass replication of critical components such as disks, switches, routers and links as is done in HP Non-Stop Servers[1] as also the addition of components which enable quick failure detection and failover such as Cisco's LocalDirector[3]. The second contributor is the cost of deploying fault-tolerant software S_c. This requires either purchasing well-supported, pricy software or employing highly qualified programmers and researchers to build robust software layers to deal with hardware failures and to recover correctly. The third contributor, maintainence costs M_c, includes environmental costs such as cooling, increased *operator-hours-per-system* costs, as also costs of providing advanced management interfaces to personnel such as service configuration and health monitoring tools. So the overall cost for achieving higher availability can be given by:

$$C_{HA} = H_c + S_c + M_c$$

5 Super-Linear Benefits?

The important aspect to note in the above equation for C_{HA} is: intuitively, none of the components indicate a linear increase in costs with higher availability. In other words, for an increase in availability from a class of 1 nines to 2 nines, an order of magnitude increase in availability, does not imply a 10-fold increase in costs. In fact there is reason to believe that the increase in availability for increased expenditure may be super-linear! In a recent availability study[7] of simple cluster-based web service we observed that addition of simple software techniques can improve the availability by an order of magnitude from a 2 and a half nines system to a 3 and a half nines system. We evaluated the effort required to achieve this higher availability and I reproduce it here in Table 2. The first column represents the additional software techniques applied, and the second column is the number of lines of code required for implementing the technique, while the third column is the percentage reduction in unavailability over a basic version which delivered 2.5 nines. What is not mentioned in the table and is from personal experience is that writing the above code and testing the system for correct behavior took about 3 weeks, which transformed to hours is about 120 man hours. If the programmer was paid a salary if 50 Euros an hour, then the effort can be transformed to a total value of 6000 Euros. If you consider the application to be servicing client requests in an airline reservation system, which might incur a loss of 100,000 Euros per hour of downtime, then the savings accrued over a year by going from 2.5 nines to 3.5 nines roughly amounts to 3,500,000 Euros!!! Note that this considers only the T_c component of C_{UA}. This simple example shows that small investment towards higher availability can deliver huge monetary gains or savings according to one's point of view.

Enhancement	**Additional NCSL**	**Unavailability Reduction**
Membership	1307	71%
Queue Monitoring	138	38%
Queue Monitoring + Membership	1445	83%
Queue Monitoring + Membership + FME	1596	88%

Table 2: *Comparison of effort, measured in non-commented source lines (NCSL), required to implement the software techniques, with the reduction in unavailability over the basic version. Data from [7]*

Is 5 nines the most we can do? Intuitively, as also from our study[7] we have seen that, improving the availability of services gets tougher as we reach the higher classes. Marginal increases in availability requires the application of many software and hardware techniques. While achieving five nines seems within our reach, we cannot say the same of 7 or 10 nines for Internet services. However, once we get to the class 5, I'm sure we can put a bold step forward to aim for the next class of availability.

6 Other Benefits...

Appart from the savings in the costs discussed in the previous sections, striving for high availability can bring about other desirable effects such as standardization of services, ability to deliver critical services online, and also furthering of research in well-managed systems. These proximal effects would eventually enable humans build more complex reliable systems to satisfy the needs of tomorrow.

7 Conclusions

The discussion presented here provides a flavor for the incentives derived by services that could deliver 5 nines of availability. Clearly, there is lot to be coped with, namely, complexity, variety of ill-defined (I mean non-standardized) services, failures in hardware and software components, and human induced service disruptions. However, with systematic approaches to hardware/software fault-tolerance, well-defined interfaces, failure detection tools and trained personnel, 5 nines for Internet services is not beyond the possible realm.

References

[1] HP Non-Stop Servers. http://nonstop.tandem.com, Jan. 2003.

[2] A. Acharya. Reliability on the Cheap: How I Learned to Stop Worrying and Love Cheap PCs. Invited Talk at the Second Workshop on Evaluating and Architecting System dependabilitY(EASY), Nov. 2002.

[3] Cisco Systems. Cisco LocalDirector 400 Series, Feb. 2003. Available at http//www.cisco.com/warp/public/cc/pd/cxsr/400/index.shtml.

[4] J. Gray and A. Reuter. *Transaction Processing: Concepts and Techniques*. Morgan Kaufmann, 1993.

[5] R. Kembel. *Fibre Channel: A Comprehensive Introduction*. Northwest Learning Assoc, 2000.

[6] M. Merzbacher and D. Patterson. Measuring end-user availability on the web: Practical experience. In *International Performance and Dependability Symposium (DSN)*, Washington DC, June 2002.

[7] K. Nagaraja, N. Krishnan, R. Bianchini, R. P. Martin, and T. D. Nguyen. Quantifying and Improving the Availability of Cooperative Cluster-Based Services. Technical Report DCS-TR-517, Department of Computer Science, Rutgers University, Jan 2003.

[8] D. Patterson. Recovery Oriented Computing: A New Research Agenda for a New Century. Keynote talk at the 8th International Symposium on High-Performance Computer Architecture (HPCA 8), Feb. 2002.

[9] R. H. Zakon. Hobbes' internet timeline v5.6. http://www.zakon.org/robert/internet/timeline/, Jan. 2003.