A Simple Way to Estimate the Cost of Downtime

David A. Patterson
Computer Science Division, University of California at Berkeley

Contact Author: patterson@cs.berkeley.edu
Regular LISA paper, Long Presentation
April 25, 2002

Extended Abstract, submitted to LISA conference 2002

It is time for the systems community of researchers and developers to broaden the agenda beyond performance. The 10000X increase in performance over the last 20 years means that other aspects of computing have risen in relative importance. We see three challenges for the future as [Patterson2002a]:

1. **Synergy with humanity**: We need to make the technology match human nature, both for users of personal gadgets and for those who operate the services accessed by those gadgets. Thus, we must care as much about the human cost of operating information technology as about the cost of purchasing it.
2. **Dependability**: We need to create a technology that the world can really depend upon, since its already doing it with a technology that does not deserve our trust.
3. **Security/Privacy**: We need to invent the technologies that helps make our societies secure without sacrificing privacy.

Although these goals cover all computing devices, we are especially interested in services over a network, which covers both Internet sites and enterprise data centers.

The systems we have created are fast and cheap, but undependable. To understand why they are undependable, we conducted two surveys on the causes of downtime, with unexpected results [Patterson 2002b]. In our first survey, we collected failure data on the U.S. Public Switched Telephone Network (PSTN). In our second, we collected failure data from three Internet sites. Based on that data, Figure 1 shows the percentage of failures due to operators, hardware failures, software failures, and overload. The surveys are notably consistent in their suggestion that operators are the leading cause of failure. The high fraction of operator error suggests that little attention has been paid to making systems easy to operate.

This figure inspired the Recovery Oriented Computing (ROC) project [Patterson 2002b]. It takes the perspective that hardware faults, software bugs, and operator errors are facts to be coped with, not problems to be solved. By concentrating on Mean Time to Repair (MTTR) rather than Mean Time to Failure (MTTF), ROC reduces recovery time and thus offers higher availability. Since a portion of system administration is dealing with failures [Anderson 2000], ROC may also reduce total cost of ownership. The ROC project is investigating several techniques to reduce the consequences of operator error.

Improving dependability and lowering cost of ownership is likely to require more resources. For example, an undo system for operator actions would likely need more disk space than conventional systems. The marketplace may not accept such innovations if products that use them are more expensive and the subsequent benefits cannot be quantified by lower cost of ownership.

Hence this paper, which seeks to define a simple and useful estimate of the cost of unavailability.

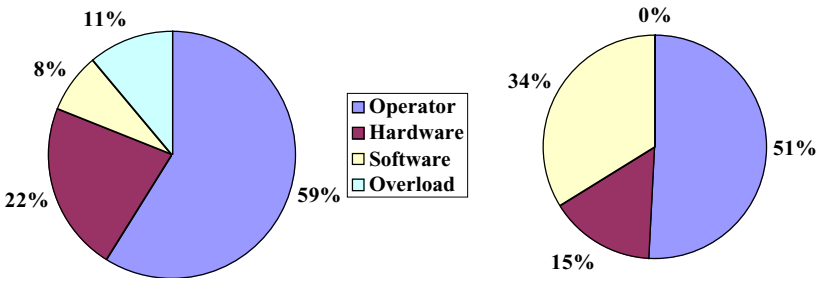

Figure 1. Percentage of failures by operator, hardware, software, and overload for PSTN and three Internet sites. Note that the mature software of the PSTN is much less of a problem than Internet site software, yet the Internet sites have such frequent fluctuations in demand that they have overprovisioned sufficiently so that overload failures are rare. The PSTN data measured blocked calls during an outage in the year 2000. (This figure does not show vandalism, which is responsible for 0.5% of blocked calls.) We collected this data from the FCC; it represents over 200 telephone outages in the U.S. that affected at least 30,000 customers or lasted at least 30 minutes. Rather than only reporting outages, telephone switches record the number of attempted calls blocked during an outage, which is an attractive metric. The Internet site data measured outages in 2001. We collected this data from companies in return for anonymity; it represents six weeks to six months of services with 500 to 5000 computers. (The figure does not include environmental causes, which are responsible for 1% of the outages. Also, 25% of outages had no identifiable cause, and are not included in the data.)

Prior work on estimating the cost of downtime is usually measuring the loss of income for online companies or other services that cannot possibly function if their computers are down. Table 1 is a typical example.

Brokerage operations	$6,450,000
Credit card authorization	$2,600,000
Ebay	$225,000
Amazon.com	$180,000
Package shipping services	$150,000
Home shopping channel	$113,000
Catalog sales center	$90,000
Airline reservation center	$89,000
Cellular service activation	$41,000
On-line network fees	$25,000
ATM service fees	$14,000

Table 1. **Cost of one hour of downtime**. From InternetWeek 4/3/2000 and *Fibre Channel: A Comprehensive Introduction*, R. Kembel 2000, p.8.
... based on a survey done by Contingency Planning Research."

Such companies are not the only ones that lose income if there is an outage. More importantly, such a table ignores the loss to a company of wasting the time of employees who cannot get their work done during an outage, even if it does not affect sales.

Thus, we need a formula that is easy to calculate so that administrators and CIOs in any institution can determine the costs of outage. It should capture both the cost of lost productivity of employees and the cost of lost income from missed sales.

Toward that end, we define two terms. *Employee costs per hour* is simply the total salaries and benefits of all employees per week divided by the average number of

working hours per week. (We picked a week since there is less ambiguity in the number of hours than in a month.) *Average income per hour* is just the total income of an institution per week divided by average number of hours per week an institution is open for business. Note that this term includes two factors: income associated with a web site and income supported by the internal information technology infrastructure. We believe these employee costs and income are not too difficult to calculate, especially since they are input to an estimate, and hence do not have to be precise to the last penny.

For example, publicly traded companies must report their income and expenses every quarter, so quarterly statements have the raw data to calculate these terms. The finance department of small companies must know both these terms to pay the bills and issue paychecks. Even departments in public universities and government agencies without conventional income sources have their salaries in the public record.

Once we quantify these two terms, we include them in a straightforward formula:

Estimated average cost of 1 hour of downtime =

Employee costs per hour * Fraction employees affected by outage
+ Average Income per hour * Fraction income affected by outage

The other two terms just need to be educated guesses or ranges that make sense for your institution. To establish the argument for systems that may be more expensive to buy but less expensive to own, administrators only need to give an example of what the costs might be using the formula above. CIOs could decide on their own fractions in making their decisions, as there is plenty of room for debate.

For example, depending on the company, one hour of downtime may not lead to lost income, as customers may just wait and order later. Employees may simply do other work for an hour that does not involve a computer. Depending on centralization of services, an outage may only affect a portion of the employees. There may also be different systems for employees and for sales, so an outage might affect just one part.

Before giving examples, we need to qualify this estimate. It ignores the cost of repair, such as the cost of overtime by operators or bringing in consultants. We assume these costs are small relative to the other terms. Second, the estimate ignores daily and seasonal variations in income; as some hours in some days are more expensive than others. For example, a company running a lottery is likely to have rapid increase in income as the deadline approaches. Perhaps a best case and worst case cost of downtime might be useful as well as average, and its clear how to calculate them from the formula.

<<Note to LISA committee; full paper would give several examples of institutions with different parameters filled in. For example; EECS department at Berkeley, Amazon, Sun. Suggestions of examples are welcome.>>

Our hypothesis is that by quantifying some of the terms it will be easy to come up with plausible estimates of the cost of downtime, or perhaps the importance of downtime for different services in the data center. If such estimates become part of the CIO and system administrator community, they may pave the way for new generations of products and processes that pay greater attention to dependability and cost of ownership than to the conventional focus of performance and cost of purchase.

References

[Anderson 1999] E. Anderson and D. Patterson, "A Retrospective on Twelve Years of LISA Proceedings," *Proc. 13th Systems Administration Conference - LISA 1999*, Nov 1999, Seattle, Washington, p.95-107.

[Patterson 2002a] D. Patterson, " A Call to Arms: A New Manifesto for Systems*," CRA Grand Challenges Workshop*, June 2002.

[Patterson 2002b] D. Patterson, A. Brown, P. Broadwell, G. Candea, M. Chen, J. Cutler, P. Enriquez, A. Fox, E. Kiciman, M. Merzbacher, D. Oppenheimer, N. Sastry, W. Tetzlaff, J. Traupman, N. Treuhaft. *Recovery-Oriented Computing (ROC): Motivation, Definition, Techniques, and Case Studies*. U.C. Berkeley Computer Science Technical Report UCB//CSD-02-1175, March 15, 2002.