Cycles, Cells and Platters: An Empirical Analysis of Hardware Failures on a Million Consumer PCs

Edmund B. Nightingale
Microsoft Research
ed.nightingale@microsoft.com

John R. Douceur
Microsoft Research
johndo@microsoft.com

Vince Orgovan
Microsoft Corporation
vinceo@microsoft.com

Abstract

We present the first large-scale analysis of hardware failure rates on a million consumer PCs. We find that many failures are neither transient nor independent. Instead, a large portion of hardware induced failures are *recurrent*: a machine that crashes from a fault in hardware is up to two orders of magnitude more likely to crash a second time. For example, machines with at least 30 days of accumulated CPU time over an 8 month period had a 1 in 190 chance of crashing due to a CPU subsystem fault. Further, machines that crashed once had a probability of 1 in 3.3 of crashing a second time. Our study examines failures due to faults within the CPU, DRAM and disk subsystems. Our analysis spans desktops and laptops, CPU vendor, overclocking, underclocking, generic vs. brand name, and characteristics such as machine speed and calendar age. Among our many results, we find that CPU fault rates are correlated with the number of cycles executed, underclocked machines are significantly more reliable than machines running at their rated speed, and laptops are more reliable than desktops.

Categories and Subject Descriptors D.4.5 [*Operating Systems*]: Reliability

General Terms Measurement, reliability

Keywords Fault tolerance, hardware faults

1. Introduction

We present the first large-scale analysis of hardware failure rates on consumer PCs by studying failures in the CPU, DRAM, and disk subsystems. We find that many failures are neither transient (one-off) nor independent (memoryless). Instead, a large portion of hardware-induced failures are recurrent. Hardware crashes increase in likelihood by up to two orders of magnitude after a first such crash occurs, and DRAM failures are likely to recur in the same location.

Three conditions motivate a hardware-failure study particularly targeted at consumer machines. First, in contrast to servers, consumer machines tend to lack significant error-resiliency features, such as ECC for memory and RAID for disks, which increases the likelihood that a hardware error will have an impact on a machine's operation. Second, whereas the failure of a single server in a data center can often be masked by the application-level logic that partitions tasks among machines, the failure of a single consumer machine directly affects the user of that machine. Moreover, there are good reasons to expect that server-class and consumer-class machines will exhibit different failure characteristics: Consumer machines are built with cheaper components, and they run in harsher environments, with wider temperature variation, greater mechanical stresses, more ambient dust, and more frequent power cycling. Third, although there now exists a significant body of work on analyzing hardware failures in server machines [Bairavasundaram et al. 2007, 2008; Kalyanakrishnam et al. 1999; Oppenheimer et al. 2003; Pinheiro et al. 2007; Schroeder and Gibson 2006, 2007; Schroeder et al. 2009; Xu et al. 1999], there is a dearth of information on consumer machines, even though they constitute the vast majority of machines sold and used each year.

Studying consumer machines brings new challenges not present when studying server-class machines. Consumer machines are geographically remote, widely distributed, and independently administered, which complicates data collection. There are no field reports from repair technicians or outage logs filed by customers, as used by other studies [Gray 1987, 1990; Kalyanakrishnam et al. 1999; Oppenheimer et al. 2003; Schroeder and Gibson 2006; Xu et al. 1999]. The general lack of ECC precludes tracking ECC errors as a means for determining memory failures, as used in server studies [Constantinescu 2003; Schroeder et al. 2009].

To address these challenges, we employ data sets from the Windows Error Reporting (WER) system [Glerum et al. 2009], which was built to support diagnosis of software faults that occur in the field. Through post hoc analysis of re-

EuroSys'11, April 10–13, 2011, Salzburg, Austria.

ports from roughly one million machines, we are able to isolate several narrow classes of failures that are highly likely to have a root cause in hardware: machine-check exceptions reported by the CPU, single-bit errors in the kernel region of DRAM, and OS-critical read failures in the disk subsystem.

This methodology has two unfortunate limitations. First, since WER logs are generated only in response to a crash, our study is blind to hardware failures that do not lead to system crashes. In particular, the only DRAM bit errors that cause system crashes are those that occur within the roughly 1.5% of memory that is occupied by kernel code pages. Errors elsewhere in the memory may cause application crashes or data corruption, but we cannot observe these.

Second, our study sheds no light on the relative frequency of hardware-induced versus software-induced crashes. The three types of failures we study constitute a minority of WER entries; however, most types of CPU, memory, and disk failures produce symptoms that are indistinguishable from kernel-software failures. On the other hand, such a result would be relevant only for Windows machines, whereas absolute measures of hardware failures are relevant for any operating system, or at least any OS that relies on a functioning CPU, error-free memory, and a responsive disk.

Despite these limitations, our study has found a number of interesting results. For instance, even small degrees of overclocking significantly degrade machine reliability, and small degrees of underclocking improve reliability over running at rated speed. We also find that faster CPUs tend to become faulty more quickly than slower CPUs, and laptops have lower failure rates than desktops. Beyond our results, this study serves to inform the community that hardware faults on consumer machines are not rare, independent, or always transient. While prior work focused on measuring hardware faults that may or may not cause an OS failure, our study is the first to measure observed OS failure rates due to faulty hardware.

In the sequel, we analyze the probability of machine failure from each failure type, as well as the conditional probability that a failure will recur (§4). We analyze the spatial locality of DRAM failures, which informs conjectures about the underlying fault. We analyze failure behavior of various machine classes, by partitioning machines into populations of overclocked vs. non-overclocked, underclocked vs. rated-speed, white box (generic) vs. brand name, and desktops vs. laptops (§5). We evaluate the interdependence of the three failure types, revealing a nonintuitive set of dependency relations. We analyze the effect of various machine characteristics on failure rate, including CPU speed, memory size, and calendar age (§6). We estimate the likelihood that underlying faults are intermittent rather than transient (§7) in a temporal analysis of recurring failures. Based on these and other results, we propose several research directions for a hardware-fault-tolerant operating system (§8).

2. Prior work

Ours is the first large-scale study of hardware failures in consumer machines, and the first study of CPU subsystem failures on any class of machines. Furthermore, many of our analyses have not been previously conducted, even for server machines. The effect on failure rates of overclocking and underclocking, brand name vs. white box, and memory size have not been previously studied. There is no prior work examining the relative frequency of intermittent vs. transient hardware faults, nor in studying the spatial locality of DRAM failures. Some of our comparisons, such as desktop vs. laptop, are not even meaningful in server systems.

However, some of our findings mirror those in prior work. Like Schroeder et al. [2009], who conducted detailed studies of DRAM failure rates on server machines, we find that DRAM errors are far more likely to occur than would be expected from studies of error rates induced by active radiation sources [Constantinescu 2002; Micron 1997; Ziegler et al. 1998] or cosmic rays [Seifert et al. 2001; Ziegler et al. 1996]. Also, like Pinheiro et al. [2007] and Schroeder and Gibson [2007], who examined disk failure rates in large data-center installations, we find that disk MTTF times are much lower than those specified on disk data sheets. Our findings on conditional failure probability mirror the findings of recent studies of DRAM failures [Schroeder et al. 2009] and disk-subsystem failures [Jiang et al. 2008] on server machines: an increase of up to two orders of magnitude in observed failure rates after a first failure occurs.

Some of our findings contrast with those of prior studies. In studying the correlation between age and DRAM failure rates, Schroeder et al. [2009] found no infant mortality and no increasing error rates at very old ages; by contrast, we find minor evidence for both of these phenomena, consistent with a bathtub curve. Constantinescu [2003], in a study of ECC errors on 193 servers, concluded that the risk of intermittent faults increases as chip densities and frequencies increase; by contrast, we find that memory size (which roughly corresponds to density) has only weak correlation to failure rate, and CPU frequency has no effect on the rate of failures per CPU cycle for non-overclocked machines.

Much of the prior work in this area addresses issues that our study does not. Our data is unable to conclusively address the relative frequency of hardware versus software failures, but the coarse-grained division between hardware, software, and operator error was the focus of a study at three internet services [Oppenheimer et al. 2003] and of two studies of Windows NT server installations [Kalyanakrishnam et al. 1999; Xu et al. 1999]. WER logs also tell us nothing about machine temperature or the relocation of components among machines, as studied by Pinheiro et al. [2007], Schroeder and Gibson [2007], and Schroeder et al. [2009]. A Windows machine will submit a WER error log only if the machine recovers after a crash; by contrast, Schroeder and Gibson [2006] studied the sources of unrecoverable failures in 4,700 com-

puters across several high-performance computing clusters. The study of disk failures by Bairavasundaram et al. [2008] showed data corruption was neither independent across disks nor independent within a single disk, roughly analogous to our study of spatial locality of DRAM errors.

3. Methodology

3.1 Terminology

We begin with a brief summary of terms used throughout the paper, derived from definitions by Patterson [2002]. We distinguish between a *failure*, which is deviant behavior (e.g., a crash), and a *fault*, which is a defect in a component. A failure is *recurrent* if it occurs more than once. Faults are characterized by duration as either permanent, intermittent, or transient. A *permanent* fault is a persistent defect that causes consistent failure, such as a burned-out chip. An *intermittent* fault is a persistent defect that causes zero or more failures, such as a speck of conductive dust partially bridging two traces. A *transient* fault is an instantaneous defect causing a single failure, such as an alpha particle striking a transistor.

In general, machines did not operate continuously for the entire study period. When reporting results with respect to time, we aggregate each machine's not-necessarily-contiguous CPU time into a quantity we call *TACT*, for Total Accumulated CPU Time. CPU time is defined as time when the machine is running. For example, an idle CPU accumulates TACT, but no TACT is accumulated by a machine that is in sleep mode.

3.2 Data sets and analysis

Our analyses employ two data sets from the Windows Error Reporting process [Glerum et al. 2009], in which Windows computers send crash and/or status reports to Microsoft.

The *RAC* data set is part of the Windows' Customer Experience Improvement Program (CEIP), in which users opt-in to allow their machines to send periodic machine status reports to Microsoft. These reports indicate not only when crashes occurred but also periods during which crashes did not occur. If a machine crashes so severely that a crash report is not generated, than those reports will not be present in our data. Therefore, our analysis can be considered conservative, since it does not capture such incidents. This data set is used for all of the analyses in this paper other than Section 4.3.

The *ATLAS* data set collects and processes a crash report whenever the Windows OS reboots after a crash. This data set includes detailed crash information that we use to spatially analyze DRAM errors in Section 4.3. We also use this data to obtain BIOS date information for the calendar age study in Section 6.4.

Both the ATLAS and RAC data sets are represented in SQL databases. More details about the collection of failure data is detailed by Glerum et al. [2009]. The analysis took roughly one person-year of effort to complete. Of this time, approximately 6 to 7 months was spent in exploratory work, and the remainder of the time was spent conducting the analyses that appear in this paper. Some of these analyses, such as those in Sections 4 and 5, were completed by executing approximately 10,000 lines of SQL written by the authors against the two databases previously listed. Other portions of the analysis, such as those in Section 6 and 7, required pulling multiple pieces of data out of the database and conducting additional work in Excel and Mathematica.

3.2.1 Managing selection bias

The biggest weakness of our methodology is selection bias. Since users volunteer their machines for either or both of our data sets, our results may not be representative of the general population of machines in the world. In particular, the occurrence or type of a hardware failure might be correlated with the machine owner's inclination to participate in the CEIP or to report a failure to ATLAS.

Although we cannot guarantee that self-selection bias has not affected our results, the large size of our population (nearly one million machines) and the wide diversity of machine characteristics (§6) suggest that our results apply to a large and diverse set of machines.

Moreover, since the processes by which users express participation in RAC and ATLAS are different from each other, we theorize that a strong correlation between these two data sets suggests that both are likely to be representative of the world population from which they were separately drawn. Although the two databases have different populations of users and machines, they both use the same failure categorization scheme, which is far more specific than our three main failure types. We sorted these categories and found matches for the top 34 categories, which account for over 99% of all hardware-related failures, according to their occurrence frequency within each database. The resulting sort orders are correlated with a Spearman's coefficient of $\rho = 0.92$, which is a significance of at least 99.9%.

3.3 Types of failures

Of the many types of hardware failures, we study only three: one in the CPU and its associated components, one in DRAM, and one in the disk subsystem. Our specific choice of failure types was motivated by our confidence that post-hoc analysis can determine that a crash is due to a particular hardware fault and is not caused by a software bug.

In addition to dictating our choice of failure types, our use of crash logs has two additional consequences. First, we observe only failures that cause system crashes, and not a broader class of failures that includes application crashes or non-crash errors. Second, we cannot observe failures due to permanent faults, since a machine with a permanent, crash-inducing fault would be unable to recover to produce, record, and transmit a crash log.

```
kd> !chkimg -d -db !Ntfs
  f9161dbd - Ntfs!NtfsCommonCleanup+18d3
    [ 53:73 ]
1 error : !Ntfs (f9161dbd)
[correct] f9161dbd 53  push ebx
[faulty]  f9161dbd 738d jae Ntfs!NtfsCommonCleanup+0x1acf
```

This figure shows the output of the Windows debugger when detecting a kernel crash caused by a single-bit flip in memory. In this example, the NTFS driver resides in a code-page marked as read-only by the MMU. When a single bit was flipped (`0x53 to 0x73`), a 1 byte instruction was decoded as a 2-byte instruction (`0x738d`), causing execution to stray into an arbitrary function.

Figure 1. Example of an error caused by a 1-bit failure

3.3.1 CPU subsystem failures

Our first type of failure occurs within the CPU or closely associated components in the core chipset. This failure occurs when the CPU issues a machine-check exception (MCE) [Intel], which indicates a detected violation of an internal invariant. Causes include bus errors, microcode bugs, and parity errors in the CPU's caches.

3.3.2 DRAM failures

Our second type of failure results from an incorrect value in a *kernel code page*, meaning regions of DRAM that hold code for the OS kernel and kernel-level drivers. These pages have two important properties that allow us to identify hardware as the cause of a failure.

The first property is that the intended content of these pages is known, because it is code from the Windows kernel or from a Microsoft or third-party driver. The binary files are digitally signed, so if they become corrupted on disk, they will fail to load into memory. System crash logs include a dump of small region of memory near the crash location, called a *mini-dump*. Comparison of the mini-dump to the intended content reveals any modifications to the page. In addition, manual inspection is often able to relate the observed code change to the cause of the crash, as shown in Figure 1. This relation substantiates the conclusion that the DRAM actually contained the observed erroneous value, in contrast to an alternate explanation in which the state of the mini-dump file became corrupted on disk.

The second property is that over 90% of kernel code pages are marked read-only, so the MMU prevents software from directly modifying the page contents. Therefore, any changes to this memory state must be either written by DMA hardware or generated within the memory itself. To discount the possibility of DMA writes, we limit our study to *one-bit failures*, meaning that only a single bit in the mini-dump shows an incorrect value.

It is still possible for a buggy DMA to cause such an error, by spuriously writing a single byte that happens to differ in only one bit from the proper value for that location. However, when we calculate the likelihood of a randomly written byte differing from the correct value by exactly one bit, it turns out to be far less probable than the rate we observe: Of the 256 possible values that might randomly overwrite a single byte, one of these values coincidentally equals the correct value, eight of these values differ from the correct value in a single bit, and the remaining 247 values differ by more than one bit. Thus, a spurious one-byte DMA would cause $\frac{8}{247}$ as many single-bit errors as multi-bit errors, but we observe over six times as many single-bit errors as multi-bit errors.

A consequence of confining our analysis to kernel code pages is that we will miss DRAM failures in the vast majority of memory. On a typical machine kernel code pages occupy roughly 30 MB of memory, which is 1.5% of the memory on the average system in our study. While our analysis provides insight into the number of operating system crashes caused by DRAM faults in kernel code pages, no general conclusion can be drawn about the distribution or frequency of DRAM faults across all of the DRAM within a machine. The analysis provides no insight into the distribution of faults, and we can make no assumption since the faults are caused by an unknown source.

3.3.3 Disk subsystem failures

Our third type of failure occurs in the disk subsystem. This failure occurs when the operating system is unable to proceed without reading data from the disk, such as when the memory manager tries to read in virtual-memory state that has been paged out. If the disk does not complete the read, even with multiple retries, the system crashes. This type of failure can be caused by a problem in the system bus, in the bus controller, in the cable connecting the disk to the controller, or in the disk itself.

4. Measuring hardware failure rates

We begin by quantifying the probability that a machine will fail from one of the three types of failures examined in the study. We measured the failure rates of approximately 950,000 machines voluntarily and anonymously reporting to the RAC database for a period of 8 months in 2008.

For each type of failure, we analyze the probability that a machine will crash while under observation. We also analyze the conditional probability that a machine will crash from a hardware failure if it has already experienced one or more failures of the same type. We begin by analyzing two subsets of machines in our study: 577,155 machines with at least 5 days of TACT, and 124,591 machines with at least 30 days of TACT. We note that even 5 days of TACT may represent many more calendar days in the study if a machine is turned on for only a few hours each day.

4.1 CPU subsystem (MCE) failures

As shown in the first row of Figure 2, machines that accumulate at least 5 days of TACT have a 1 in 330 chance of

Failure	min TACT	Pr[1st failure]	Pr[2nd fail | 1 fail]	Pr[3rd fail | 2 fails]
CPU subsystem (MCE)	5 days	1 in 330	1 in 3.3	1 in 1.8
CPU subsystem (MCE)	30 days	1 in 190	1 in 2.9	1 in 1.7
Memory (DRAM one-bit flip)	5 days	1 in 2700	1 in 9.0	1 in 2.2
Memory (DRAM one-bit flip)	30 days	1 in 1700	1 in 12	1 in 2.0
Disk subsystem	5 days	1 in 470	1 in 3.4	1 in 1.9
Disk subsystem	30 days	1 in 270	1 in 3.5	1 in 1.7

Figure 2. The (conditional) probability of an OS crash from various hardware failures

crashing due to an MCE during the 8 month observation period. After a machine has crashed once, its crash probability increases by a factor of 100, and the probability continues to increase with subsequent crashes. The second row in the figure shows that the same trend holds for machines with at least 30 days of TACT, but the initial probability of failure is higher. Further analysis shows that, of the machines with at least 5 days of TACT that experience a recurring crash from an MCE, 84% of machines experience a recurrence within 10 days of TACT, and 97% of machines experience a recurrence within a month.

4.2 One-bit DRAM failures

The middle two rows of Figure 2 show the crash probability for one-bit DRAM failures, which are broadly similar to the trends for CPU subsystem failures: The failure probability jumps by more than two orders of magnitude after a first failure is observed, and the probability further increases with subsequent crashes. The initial failure probabilities are nearly an order of magnitude lower than those for CPU failures, but this gap is almost erased after two repeated crashes. In addition, since we are capturing DRAM errors in only 1.5% of the address space, it is possible that DRAM error rates across all of DRAM may be far higher than what we have observed. Further analysis shows that, of the machines with at least 5 days of TACT that crash from a repeated one-bit DRAM failure, 94% experience the second crash within 10 days of TACT, and 100% crash within 30 days.

Memory manufacturers often use a metric called FITS (failures per billion hours of uptime) [Micron 1994] to describe the probability of an Alpha particle or neutrino flipping a bit in memory. Our data suggests that FITS is an insufficient metric upon which to make decisions for building software reliability or determining whether hardware protections such as ECC should be used. Unfortunately, ECC memory is seen as a "premium" part, and is often used only in server machines. Field studies—such as ours—may observe many different environmental effects, such as dirt, heat, or assembly defects, all of which all can conspire to increase the probability of memory errors.

4.3 Spatial analysis of one-bit DRAM failures

DRAM faults provide a unique opportunity to gain more insight into recurring hardware faults. Each mini-dump sent to Microsoft contains sufficient information to determine the memory location of a one-bit fault. We can thus determine whether recurring one-bit faults show spatial locality, which would substantiate the hypothesis that such recurrences are not merely coincidental.

Unfortunately, we have a fairly limited set of crash reports on which to base this analysis, for two reasons. First, the RAC database contains only post-processed crash results, not actual mini-dumps, so we must analyze mini-dumps from ATLAS instead. Therefore, we analyzed 381,315 one-bit failures reported to the ATLAS database, only 22,319 of which are recurrent. Second, we can determine the physical address only for the 2707 recurrent failures that occurred within a particular 3 MB portion of the 30 MB Windows kernel image, because the remainder of the image is not loaded at a deterministic physical address.

Our result is that, among machines that experienced recurrent DRAM failures, 79% experienced at least two such failures at the same physical address with the same bit flipped. In fact, the rate of spatial locality may even be higher, due to a weakness in the ATLAS data set: Each machine in ATLAS is tagged with an ID that is not guaranteed to be unique, so many observations of two failures on the "same" machine at different locations may in fact be failures on two different machines. The identifier collision rate is hard to quantify but known to be non-zero, so 79% is a conservative value for spatial locality.

There is one important caveat to this result: The 3 MB portion of the kernel image used for this analysis is the only part of kernel code not protected by the MMU, so it is possible that the one-bit errors are due to erratic software rather than DRAM failures. However, we observed **zero** spatial locality among failures on different machines. Therefore, for software to cause the behavior we have seen, a stray thread would have to flip the same bit across different boots of the same machine, but never flip the same bit across two different machines. Although this is possible, we conjecture it to be considerably more probable that these recurring crashes are caused by a spatially local defect in a DRAM chip.

4.4 Disk subsystem

The last two rows of Figure 2 show the crash probability for disk subsystem failures. The first-failure probability is one in several hundred, followed by a roughly two-order-of-magnitude increase after a machine has failed once, and increasing thereafter. Further analysis found that, of the ma-

	CPU Vendor A		CPU Vendor B	
	No OC	OC	No OC	OC
Pr[1st]	1 in 400	1 in 21	1 in 390	1 in 86
Pr[2nd|1]	1 in 3.9	1 in 2.4	1 in 2.9	1 in 3.5
Pr[3rd|2]	1 in 1.9	1 in 2.1	1 in 1.5	1 in 1.3

Figure 3. CPU failures and the effects of overclocking

chines with at least 5 days of TACT, 86% of recurring disk subsystem failures recur within 10 days of TACT, and 99% of recurrences happen within 30 days.

These crashes are costly because all data items not yet written to disk are permanently lost when a machine crashes from a disk fault. Therefore, this data suggests that when a machine fails from a disk subsystem error, it is prudent to *immediately* replace the disk, rather than waiting to see if the failure recurs.

4.5 Are failures memoryless?

The trends shown in Figure 2 strongly suggest that the occurrence rates of hardware failures are not memoryless. To substantiate this conclusion rigorously, we performed a χ^2 test to compare per-machine failure inter-occurrence times to an exponential distribution. With a better than 0.001 level of significance, observed failure inter-occurrence times are not exponential and therefore not memoryless.

This is an important result because it suggests that mean time to fail (MTTF) would be an inherently inappropriate metric for hardware failures on consumer machines. MTTF has relevance in a data center, where an operator cares about aggregate failure behavior: Thanks to the law of large numbers, an operator can use the Poisson distribution to set confidence bounds on the expected count of hardware failures in a year, even though the failure inter-occurrence time for each machine is not exponentially distributed. However, for a consumer machine running a single-system OS, MTTF provides very little insight into how frequently failures will occur in practice. For instance, the MTTF for our studied population is 2.07×10^8 CPU seconds, which is 6.56 CPU years; however, after a failure occurs, the MTTF drops to 1.17×10^6 seconds, which is 13.5 CPU days.

5. Effect of machine class

Next, we partition the data set into various classes of machines, to determine their effect on OS failure rates.

5.1 Effects of Overclocking

The first class we examine is overclocked machines. When a CPU is fabricated, the manufacturer rates the speed at which the part passes a set of tests. We refer to this as the CPU's "rated" speed. Some end users and/or OEMs will overclock the part to improve performance. However, this increases the risk of malfunction, either from directly exceeding component tolerances (such as gate delays) or from indirectly affected environmental factors (such as overheating).

Failure type	No OC	OC
DRAM one-bit flip	1 in 2800	1 in 560
Disk subsystem	1 in 480	1 in 430

Figure 4. Disk, DRAM and the effects of overclocking

5.1.1 Identifying an overclocked machine

We measured the effects of overclocking on machine reliability from two major parts vendors of x86 compatible CPUs. When the Windows operating system boots, it measures the actual clock speed of the machine. Identifying the rated frequency involved a bit more work. One of the authors created a SQL table that allowed us to match a CPU model number, which is identifiable in the SMBIOS table, to a CPU rated frequency, which was identified through a time-intensive, but one-time manual analysis. However, we were able to identify the rated frequency for only a subset of the machines in our study. For our analysis of machines with at least 5 days of TACT, we could identify the rated speed of 477,464 out of 577,155 machines. Any machine for which we could not identify the rated speed was ignored.

We can compare the measured frequency and the rated frequency to see if the machine is overclocked. However, CPUs are often sold at a rated frequency that differs slightly from the frequency at which the manufacturer established reliable operation. For example, a 2 GHz chip might run at 1.995 GHz in practice. To conservatively measure only those machines explicitly overclocked by users or resellers, we divide the machines into three groups: *non-overclocked* machines, whose actual speed is within 0.5% above the rated speed (97% of machines), *unknown* machines, whose actual speed is between 0.5% and 5% above the rated speed (1% of machines), and *overclocked* machines, whose actual speed is more than 5% faster than the rated speed (2% of machines). This analysis ignores machines in the unknown category.

A potential concern with this analysis is that TACT might correlate with how over- or underclocked a machine is, which would conflate our results. However, we calculated negligible correlation ($r = 0.0004$) between TACT and the ratio of actual CPU speed to rated CPU speed.

5.1.2 Overclocking analysis

Figure 3 shows the impact of an overclocked device on the probability that a machine will crash from a machine check exception. For brevity, results are shown only for machines with at least 5 days of TACT; similar trends hold for machines with at least 30 days of TACT. We have divided the analysis between two CPU vendors, labeled "Vendor A" and "Vendor B." The table shows that CPUs from Vendor A are nearly 20x as likely to crash a machine during the 8 month observation period when they are overclocked, and CPUs from Vendor B are over 4x as likely. After a failure occurs, all machines, irrespective of CPU vendor or overclocking, are significantly more likely to crash from additional machine check exceptions.

Failure type	UC	rated
CPU subsystem (MCE)	1 in 460	1 in 330
DRAM one-bit flip	1 in 3600	1 in 2000
Disk subsystem	1 in 560	1 in 380

Figure 5. The effects of underclocking

Failure type	Brand name	White box
CPU subsystem (MCE)	1 in 120	1 in 93
DRAM one-bit flip	1 in 2700	1 in 950
Disk subsystem	1 in 180	1 in 180

Figure 6. Brand name vs. White box

Failure type	Desktops	Laptops
CPU subsystem (MCE)	1 in 120	1 in 310
DRAM one-bit flip	1 in 2700	1 in 3700
Disk subsystem	1 in 180	1 in 280

Figure 7. Desktops vs. Laptops

Figure 4 shows the probability of crashing at least once from a disk subsystem failure or a DRAM error for overclocked and non-overclocked machines with at least 5 days of TACT. For the sake of brevity, we omit the measurements of the 2nd and 3rd failure, since they converge on a high probability of failure when a machine has previously crashed at least once. Overclocking increases the likelihood of one-bit flip failures by a factor of 4; however, it does not have much impact on the probability of disk subsystem failures.

5.2 Effects of underclocking

As noted in Section 5.1.1, CPUs that are not explicitly overclocked by users still run slightly above or below their rated frequency. We were interested to see whether there was any variance in reliability among non-overclocked machines. Therefore, we further partitioned the non-overclocked machines into *underclocked* machines, which run below their rated frequency (65% of machines), and *rated* machines, which run at or no more than 0.5% above their rated frequency (32% of machines). As shown in Figure 5, underclocked machines are between 39% and 80% less likely to crash during the 8 month observation period than machines with CPUs running at their rated frequency.

5.3 Brand name vs. white box

The next comparison examines whether "brand name" machines are more reliable than "white box" machines. We identify a machine as brand name if it comes from one of the top 20 OEM computer manufacturers as measured by worldwide sales volume. To avoid conflation with other factors, we remove overclocked machines and laptops from our analysis. We limited our study to failures that occur within the first 30 days of TACT, for machines that have at least 30 recorded days of TACT. This is critical because brand name machines have an average of 9% more TACT per machine than white box machines, so including all observed failures would conflate our results. Figure 6 shows that white box machines have less reliable CPUs and dramatically less reliable DRAMs than brand name machines, although disks are just as reliable.

5.4 Desktop vs. laptop machines

Our next analysis compares the reliability of desktop and laptop machines. To avoid conflation with other factors, we remove overclocked and white box machines from our analysis. Because desktops have 35% more TACT than laptops, we only count failures within the first 30 days of TACT, for machines with at least 30 days of TACT.

Although one might expect the typically abusive environment of laptops to make them less reliable, Figure 7 shows the opposite. Laptops are between 25% and 60% **less** likely than desktop machines to crash from a hardware fault over the first 30 days of observed TACT. We hypothesize that the durability features built into laptops (such as motion-robust hard drives) make these machines more robust to failures in general. Perhaps the physical environment of a home or office is not much gentler to a computer as the difference in engineering warrants.

5.5 Interdependence of failure types

Finally, we ran an analysis to determine whether the probability of crashing from each type of hardware failure is independent of crashing from each of the other types of hardware failure. We ran a contingency-table analysis using a χ^2 test. (Details are in Appendix A.) We found that (1) CPU failures and DRAM failures are dependent, and (2) CPU failures and disk failures are dependent, but (3) DRAM failures and disk failures are independent. These results are statistically strong, with levels of significance better than 10^{-6} for the dependence results and a level of significance of 0.3 for the independence result.

This is bizarre. The simplest cases would be those in which all three are dependent, or all three are independent, or two are dependent but each is independent of the third. Yet we found two independent of each other but each separately dependent on the third. This implies the existence of at least two different factors, one (such as a flaw in the memory bus) that affects the CPU and DRAM but not the disk, and another (such as a flaw in the PCI bus controller) that affects the CPU and disk but not the DRAM. Lacking the data for further analysis, we offer these ideas as purely conjecture.

6. Effect of machine characteristics

We now examine the effect of various machine characteristics on failure rate. The specific characteristics we consider are absolute CPU speed, CPU speed ratio (speed relative to rated speed), memory size, and calendar age.

These analyses are complicated because our sample population has a highly non-uniform distribution of each of these characteristics, as shown in Figure 8. In addition, the

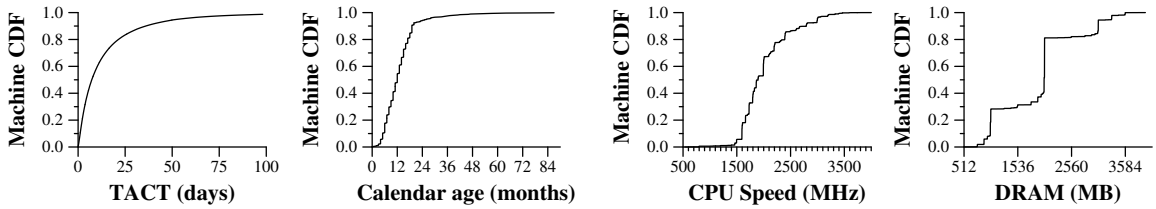


Figure 8. Characterization of machines in the study

amount of sample time (in TACT) may be correlated to machine characteristics, most obviously to calendar age, since younger machines have shorter lifetimes to sample.

To factor out these conflating issues, our primary analytical tool is the P-P plot[1], a graphical technique that compares two cumulative distribution functions by plotting one of the functions against the other [Gibbons and Chakraborti 2003]. (An example of P-P plot construction is shown in Appendix B.) For most of the graphs in this section, we plot a CDF of failures (versus the characteristic under study) against the CDF of TACT (versus the characteristic under study). If failures were to occur independently of the characteristic, the resulting P-P plot would be a straight line along the line $y = x$. Deviations from straight-line behavior show particular forms of dependence.

As an aid to visual interpretation, our P-P plots show a dashed line for $y = x$. They also show dotted lines that outline a 99% confidence envelope that should contain any P-P plot for which the two CDFs are uncorrelated. The size of the envelope differs among plots, because the sample variance in the failure CDF varies with the count of observed failures.

Because these P-P plots weight machines in proportion to their TACT, there is no need to discard machines with less than 5 days of TACT, as we did in previous sections.

6.1 CPU speed

We first consider the characteristic of CPU speed. For this study, we limit our analysis to machine records in the RAC data set that contain valid entries for both actual CPU speed and rated CPU speed. As a result, we conducted the analysis on 83% of the machines.

Figure 9 shows P-P plots for our three failure types versus TACT, relative to CPU speed. For CPU subsystem failures, we also show one plot—Figure 9(b)—that includes only non-overclocked machines. All of the graphs in Figure 9 show P-P plots that significantly deviate (with $> 99\%$ confidence) from the line $y = x$. Specifically, all of these plots pull to the right, indicating that failures become more common as CPU speed increases. This result suggests that one can minimize the likelihood of failure over a given time period by operating at the slowest CPU speed sufficient to achieve desired performance.

[1] An alternative approach is to use a technique such as principal component analysis (PCA). However, PCA requires the data to be jointly normally distributed. This is a strong assumption that is not defensible for our data.

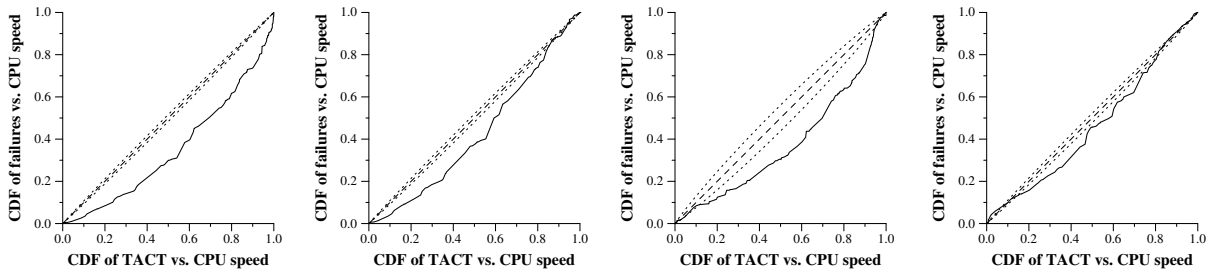
Before discussing each failure type in turn, we consider one factor that could contribute to this tendency across all failure types: Faster CPUs execute more cycles per CPU second, so even if the failure rate *per cycle* were independent of CPU speed, the failure rate *per CPU second* would not be. To factor this out, we also generate P-P plots versus total aggregate CPU cycles (TACC), as shown in Figure 10.

CPU subsystem: Figure 9(a) shows that the frequency of CPU failures increases dramatically as CPU speed increases. Much of this can be accounted for by a faster machine's greater count of cycles per CPU second, as evidenced by the comparative weakness of this trend in Figure 10(a). Alternatively, the trend is weakened when considering only non-overclocked machines, as shown in Figure 9(b). These two factors together can account for nearly all of the effect of CPU speed on CPU failure rate, as shown by the lack of any consistent pull in Figure 10(b). In other words, we see no evidence that the per-cycle failure rate of non-overclocked CPUs is dependent on CPU speed.

Memory: Figure 9(c) shows a positive correlation between memory failures and CPU speed, except at the extrema of the distribution. The 99% confidence envelope is notably wider in this graph than in Figure 9(a), because our data set contains an order-of-magnitude fewer memory failures than CPU subsystem failures.

Figure 10(c) shows that this positive correlation decreases when we account for the increase in cycles per CPU second. We do not know why this is so; one possibility is that faster CPUs tend to load and flush their caches more frequently, which leads to a greater frequency of memory operations per second. In any event, the positive correlation for the middle of the distribution is still significant in Figure 10(c). This might, for instance, indicate a correlation between DRAM errors and package temperature, since faster CPUs tend to run hotter, thus warming nearby DRAM.

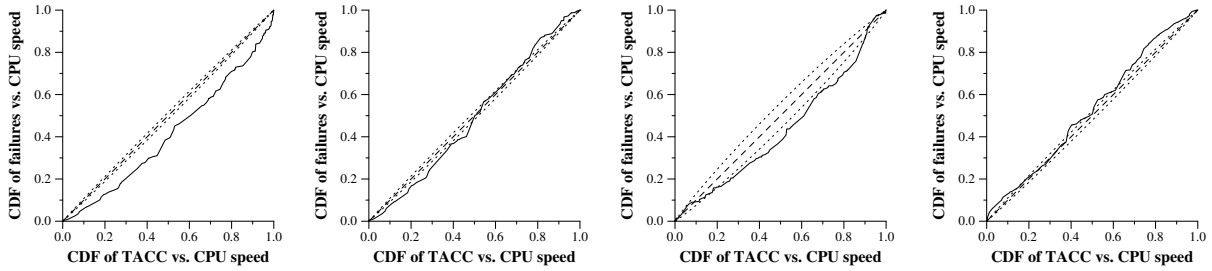
Disk subsystem: Figures 9(d) and 10(d) show that disk subsystem errors have a complicated relationship to CPU speed. When calculated per unit time, the correlation is positive, but when calculated per CPU cycle, the correlation is negative. We offer the following (untested) hypothesis to explain this behavior: We conjecture that (1) disk subsystem failures are correlated with disk activity; (2) disk activity per unit time increases as CPU speed increases, because the processor spends less time in critical-path computations between disk operations; and (3) the increase in disk activity per unit time is sub-linear with respect to CPU speed, be-



(a) CPU failures, all (b) CPU failures, non-OC (c) memory failures (d) disk subsystem failures

The dashed line $y = x$ shows where a perfectly independent plot would lie. The dotted lines outline a 99% confidence envelope for independence. Whereas (a), (c), and (d) show all machines for which CPU speed could be determined, (b) shows only machines whose CPU speeds are no more than 0.5% over rated speed.

Figure 9. P-P plots of failure count versus total aggregate CPU seconds (TACT), relative to CPU speed



(a) CPU failures, all (b) CPU failures, non-OC (c) memory failures (d) disk subsystem failures

Figure 10. P-P plots of failure count versus total aggregate CPU cycles (TACC), relative to CPU speed

cause the disk activity rate is also constrained by the disk speed. If all three of these conditions hold, a disk connected to a faster CPU will perform more operations per unit time, leading to a higher error rate per CPU second; however, the CPU will perform more cycles per disk operation, leading to a lower error rate per CPU cycle.

6.2 CPU speed ratio

Sections 5.1 and 5.2 investigated the effect of over- and underclocking on machine stability. In the present subsection, we study this effect more broadly. We generate P-P plots for each failure type versus TACT, relative to the ratio of actual CPU speed to rated CPU speed, as shown in Figure 11.

Unlike the other characteristics we study, CPU speed ratio naturally divides into well-demarcated regions. Ratios below 1 are machines that are underclocked; ratios of exactly 1 are machines that are clocked at their rated speed; and ratios above 1 are machines that are overclocked. As described in Section 5.1.1, we divide overclocked machines into those that may be within manufacturers' tolerances and those that are clearly not; the dividing point is a ratio of 1.05. These regions are demarcated in Figure 11 with long-dash lines.

All three of these P-P plots show significant positive correlation between failures and CPU speed ratio, meaning that failures become more common as the ratio of actual to rated CPU speed increases.

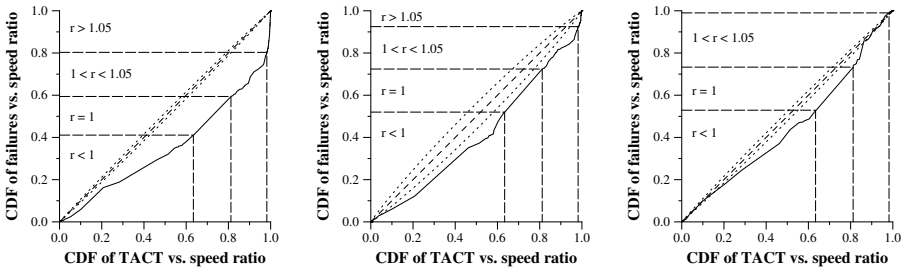
CPU subsystem: Figure 11(a) shows the most dramatic rightward pull of all our P-P plots, over all machine characteristics and all failure types. The less-than-unity slope in the $r < 1$ region graphically illustrates the lower CPU failure rates of underclocked machines (c.f. §5.2), and the nearly vertical slope in the $r > 1.05$ region shows the profound effect of overclocking on CPU failures (c.f. §5.1). The sharp knee at $r = 1.05$ validates our choice of 5% as a cutoff for judging overclocking.

Memory: Figure 11(b) shows a similar trend to Figure 11(a), although it is less dramatic. In the $r < 1$ region, underclocked machines account for 63% of TACT but only 52% of memory errors. In the next two regions, machines at rated speed and those up to 5% over rated speed account for 35% of TACT and 41% of memory errors. In the $r > 1.05$ region, overclocked machines account for 2% of TACT and 8% of memory errors.

Disk subsystem: Figure 11(c) also shows a positive correlation, but it is more spread out than for the other two failure types. In marked contrast to Figures 11(a) and 11(b), the upper-right end of the plot falls almost perfectly on the $y = x$ line: The top 14% of overclocked CPU seconds account for 14% of all disk subsystem failures. This is consistent with the result from Section 5.1.2 that overclocking does not have a large effect on disk subsystem faults.

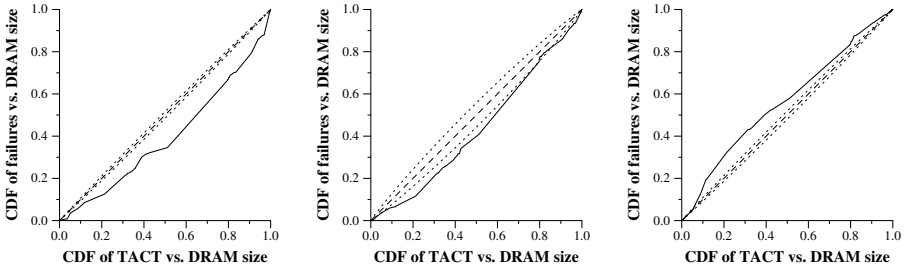
6.3 Memory size

Next, we consider the characteristic of machine DRAM size, for which P-P plots of failures versus TACT are shown in Figure 12. While one might expect that DRAM size has lim-



(a) CPU subsystem failures (b) memory failures (c) disk subsystem failures

Figure 11. P-P plots of failure count versus TACT, relative to the ratio of CPU speed to rated speed



(a) CPU subsystem failures (b) memory failures (c) disk subsystem failures

Figure 12. P-P plots of failure count versus TACT, relative to DRAM size

ited impact on the stability of a machine, we instead find that our three failure types have widely differing relationships to this characteristic.

CPU subsystem: Figure 12(a) shows that CPU subsystem failures have a strong positive correlation to the amount of DRAM in a machine. This is the only strong correlation we have observed for which we can offer no interesting hypothesis. We have been unable to imagine a plausible causal chain that would link memory size to CPU subsystem failures. Instead, we conjecture that this may be an artifact of some other correlative factor. For example, DRAM size may coincidentally correlate with particular CPU models.

Memory: Figure 12(b) shows that memory failures have a weak positive correlation to memory size. The curve barely leaves the 99% confidence envelope for uncorrelated P-P plots. We cannot infer much meaning from this graph, because the only DRAM failures we catch are in kernel code pages, and the size of the kernel text segment is independent of the overall DRAM size. Therefore, if the overall rate of DRAM failures were to increase linearly with increasing DRAM size, our study would not reveal this trend.

Disk subsystem: Figure 12(c) shows that disk subsystem failures have a negative correlation to memory size. This is intuitively reasonable, since the only type of disk subsystem failure we catch is an OS-critical disk read failure, such as paging in virtual-memory state. As memory size increases, paging activity decreases, which reduces the opportunities for such errors to manifest. This explanation is consistent with the observation that the strongest negative correlation (the greatest deviation of P-P plot slope from unity) is in the lower part of the distribution, where the effect of memory size on paging activity would be most pronounced.

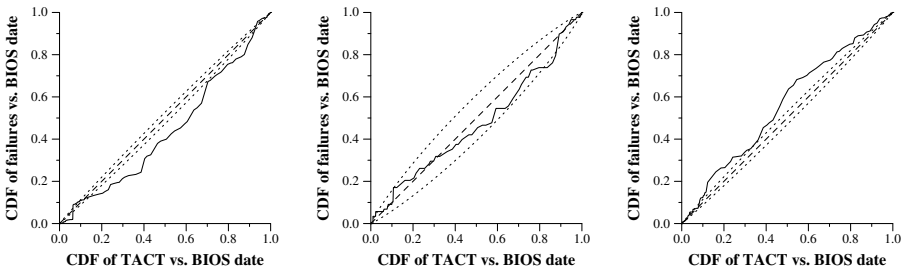
6.4 Calendar age

Lastly, we consider the characteristic of calendar age. To approximate a machine's date of manufacture, we use the release date of the BIOS version on the machine. This is accurate to within several months, because new BIOS versions are typically released every few months over the lifetime of a machine model. The RAC database does not contain BIOS dates, but ATLAS does, so we match machine models across the two data sets. This limits us to the 44% of machines in RAC with models we could identify via ATLAS. Moreover, we cannot determine BIOS date for a model that never reported a crash via ATLAS, so there is selection bias against highly reliable models.

Figure 13 shows P-P plots for our three failure types versus TACT, relative to BIOS date. As with the characteristic of DRAM size, our three failure types differ widely in relation to BIOS date.

In discussing failure rate versus age, it is common to invoke the bathtub curve [Wilkins 2009], a compound distribution that combines the failure-inducing effects of burn-in and wear-out, leading to higher failure rates among both young and old machines, with lower failure rates in the middle. Our P-P plots show evidence of both of these phenomena, albeit not consistently across failure types.

CPU subsystem: Figure 13(a) shows that CPU subsystem failures have a positive correlation to BIOS date, meaning that younger machines show a greater incidence of CPU subsystem failures. This could be due to burn-in, but this



(a) CPU subsystem failures (b) memory failures (c) disk subsystem failures

Figure 13. P-P plots of failure count versus TACT, relative to BIOS date

leaves open the question of why there is no evidence of wear-out. A possibility is that machine owners rarely keep a CPU long enough for it to wear out, in contrast to DIMMs and disks, which may be relocated from discarded old machines to replacement new machines.

Memory: Figure 13(b) shows that memory failures have no significant correlation to BIOS date, at the 99% confidence level. With lower confidence, we can see a serpentine shape, in which young and old machines account for a disproportionate share of failures, consistent with a bathtub curve. Unfortunately, among machines for which we were able to determine a BIOS date, the count of memory failures is so small that our confidence envelope is very large. A larger sample might well show a significant bathtub shape.

Disk subsystem: Figure 13(c) shows that disk subsystem failures have a negative correlation to BIOS date, meaning that older machines show a greater incidence of disk subsystem failures. Since disks are mechanical devices, it would not be surprising to find that wear-out is the dominant effect of age-related failure. Alternatively, it could be that disk manufacturers do an effective job of burning-in disks at the factory, so burn-in effects are not visible in our data from consumer machines.

7. Temporal analysis

In this section, we briefly analyze the temporal behavior of faulty machines. In particular, we estimate the fraction of non-permanent faults that are intermittent as opposed to transient, using 755,539 records from the RAC data set.

As a first-level approximation, we assume that every recurring failure indicates an intermittent fault, and that every non-recurring failure indicates a transient fault. Figure 14 shows the measured rates of recurring failures for each failure type. Of the machines that experienced a CPU subsystem crash, 30% experienced such a crash more than once. The comparable values for memory and disk subsystem crashes are 15% and 29%.

The recurrence or non-recurrence of a failure is only an approximation for the intermittence or transience of a fault, for two reasons: First, a transient fault can manifest as a recurring failure if multiple faults coincidentally develop on the same machine. Second, an intermittent fault can manifest as a non-recurring failure if only a single failure instance occurs during the observation period. We have developed estimators for the probabilities that these conditions occur, based on empirically-derived models of failure co-occurrence patterns, failure recurrence times, and machine observation periods. From these, we have analytically computed maximum-likelihood estimates for the probability of intermittent faults. (This is briefly described in Appendix C.) Figure 14 shows the resulting estimates for each failure type. Of the machines that experienced a CPU subsystem crash, an estimated 39% have an underlying intermittent fault. The comparable values for memory and disk subsystem crashes are 19% and 39%.

	Recurrent (measured)	Intermittent (estimated)
CPU (MCE) failures	30%	39%
DRAM failures	15%	19%
DISK failures	29%	39%

Figure 14. Failure and fault probabilities

8. A hardware-fault-tolerant OS

Although many distributed systems regard hardware failures to be normal occurrences, single-machine systems commonly consider hardware failures to be drastic exceptions that lead inexorably to crashes. While some research [Kadav et al. 2009; Rinard et al. 2004] has already begun to change this view, the results in the present paper suggest more work is warranted towards an operating system designed with faulty hardware as a first-order concern.

For example, a hardware-fault-tolerant (HWFT) OS might map out faulty memory locations (§4.3), just as disks map out bad sectors. In a multi-core system, the HWFT OS could map out intermittently bad cores (§7). More interestingly, the HWFT OS might respond to an MCE (§4.1) by migrating to a properly functioning core, or it might minimize susceptibility to MCEs by executing redundantly on multiple cores [Bressoud and Schneider 1996]. A HWFT OS might be structured such that after boot, no disk read is so critical as to warrant a crash on failure (§3.3.3). Kernel data structures could be designed to be robust against bit errors (§4.2). Dynamic frequency scaling, currently used for power and

energy management, could be used to improve reliability (§5.2), running at rated speed only for performance-critical operations. We expect that many other ideas will occur to operating-system researchers who begin to think of hardware failures as commonplace even on single machines.

9. Conclusion

This paper presents the first large-scale study of hardware-induced operating system failures on consumer machines. Using post-hoc analysis of machine status reports and OS crash logs, we find that:

- **Failure rates are non-trivial.** Hardware crash rates are up to 1 in 190 over an 8 month observation period.
- **Recurrent failures are common.** Hardware crashes increase in likelihood by up to two orders of magnitude after a first such crash occurs, and 20% to 40% of machines have faults that are intermittent rather than transient.
- **Recurrent failures happen quickly.** As many as 97% of recurring failures occur within 10 days of the first failure on a machine.
- **CPU speed matters.** Even minor overclocking significantly degrades reliability, and minor underclocking improves reliability. Even absent overclocking, faster CPUs become faulty more rapidly than slower CPUs.
- **DRAM faults have spatial locality.** Almost 80% of machines that crashed more than once from a 1-bit DRAM failure had a recurrence at the same physical address.
- **Configuration matters.** Brand name desktop machines are more reliable than white box desktops, and brand name laptops are more reliable than brand name desktops. Machines with more DRAM suffer more one-bit and CPU errors, but fewer disk failures.

Figure 15 shows a terse summary of our findings, with comparisons against previous results where available.

Appendix A—contingency table analysis

In Section 5.5, we described an interdependence analysis of our three failure types. Figure 16 shows the contingency tables used for this analysis. Each cell shows the exact count of machines and (in parentheses) the expected count, to three decimal places, if the indicated failure types were independent. For instance, 5 machines were observed to experience both CPU and DRAM failures, whereas the expected count is 0.549 machines. The caption of each table shows the calculated χ^2 value and the resulting p-value.

Appendix B—construction of P-P plots

Figure 17 shows the two CDFs used to construct the P-P plot in Figure 9(a). If machines had a uniform failure rate per CPU second of execution, independent of CPU speed, we would expect these two CDFs to look the same. Since the

machines	DRAM failures	no DRAM failures
CPU failures	5 (0.549)	2091 (2100)
no CPU failures	250 (254)	971,191 (971,000)

(a) CPU and DRAM failures: $\chi^2 = 36$, p-value $= 1.8 \times 10^{-9}$

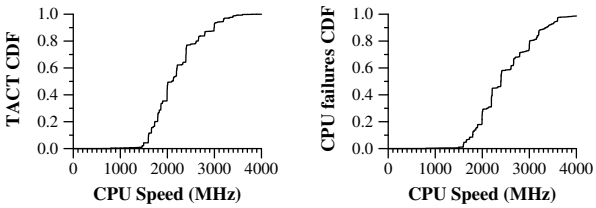
machines	Disk failures	no Disk failures
CPU failures	13 (3.15)	2083 (2090)
no CPU failures	1452 (1460)	969,989 (970,000)

(b) CPU and disk failures: $\chi^2 = 31$, p-value $= 2.8 \times 10^{-8}$

machines	Disk failures	no Disk failures
DRAM failures	1 (0.384)	254 (255)
no DRAM failures	1464 (1460)	971,818 (972,000)

(c) DRAM and disk failures: $\chi^2 = 1.0$, p-value $= 0.31$

Figure 16. Contingency tables for failure types



(a) TACT vs. CPU speed (b) CPU failures vs. CPU speed

Figure 17. CDFs used to construct P-P plot in Figure 9(a)

CDFs are not the same, their differences provide information about the dependence of failure rate on CPU speed. For instance, machines with CPU speeds below 2000 MHz were responsible for 46% of all TACT (Figure 17(a)) but only 26% of all CPU failures (Figure 17(b)).

It requires detailed study to extract specific comparison points from the two CDFs. To relieve the reader of this burden, we construct a P-P plot as follows: We sweep through the range of CPU speeds in the X-axes of Figure 17. At each speed, we take the Y-value from each curve and plot the CDF value of CPU failures against the CDF value of TACT. For instance, at a CPU speed of 2000 MHz, we plot the point $(0.46, 0.26)$. The resulting curve, as seen in Figure 9(a), shows the overall effect of CPU speed on the relationship between CPU failures and TACT.

To plot the 99% confidence envelope, we rely on the central limit theorem. Each curve of the envelope is a P-P plot of two normal-distribution CDFs against each other. The CDFs have unity standard deviation, and their means differ by $\alpha/\sqrt{N}$, where N is the count of observed failures. We set $\alpha = 2.58$ because 99% of the probability mass of a normal distribution is within 2.58 standard deviations of the mean.

Appendix C—modeling failure recurrence

To estimate the probabilities of transient versus intermittent faults, we determine parameter values that are most likely to have produced the observed results. We cannot use a

System	Topic	Finding	Comparison
CPU	initial failure rate	1 in 190	
DRAM	initial failure rate	1 in 1700	consistent with Schroeder et al. [2009]
Disk	initial failure rate	1 in 270	consistent with Pinheiro et al. [2007] consistent with Schroeder and Gibson [2007]
CPU	rate after first failure	2 order-of-magnitude increase	
DRAM	rate after first failure	2 order-of-magnitude increase	consistent with Schroeder et al. [2009]
Disk	rate after first failure	2 order-of-magnitude increase	consistent with Jiang et al. [2008]
DRAM	physical address locality	identical for 70% of machines	
all	failure memorylessness	failures are not Poisson	
all	overclocking	failure rate increase 11% to 19x	
all	underclocking	failure rate decrease 39% to 80%	
all	brand name / white box	brand name up to 3x more reliable	
all	laptop / desktop	laptops 25% to 60% more reliable	
cross	CPU & DRAM	failures dependent	
cross	CPU & disk	failures dependent	
cross	DRAM & disk	failures independent	
CPU	increasing CPU speed	fails incr. per time, const. per cycle	partly agree with Constantinescu [2003]
DRAM	increasing CPU speed	failures increase per time & cycle	
Disk	increasing CPU speed	fails incr. per time, decr. per cycle	
CPU	increasing DRAM size	failure rate increase	
DRAM	increasing DRAM size	failure rate increase (weak)	weaker than Constantinescu [2003]
Disk	increasing DRAM size	failure rate decrease	
CPU	calendar age	rates higher on young machines	
Disk	calendar age	rates higher on old machines	
all	intermittent faults	15%–39% of faulty machines	

Figure 15. Summary of our results with comparisons to prior research

standard maximum-likelihood estimator, because our data does not fit a standard distribution.

We estimate the counts M_T and M_I of machines experiencing transient and intermittent faults. These unknown counts are related to the known counts M_N and M_R of machines that reported non-recurring and recurring failures, via the following two equations:

$$M_N = (1-\theta)M_T + \phi M_I$$

$$M_R = \theta M_T + (1-\phi)M_I$$

where θ is the probability that a transient fault will manifest as a recurring failure, and the ϕ is the probability that an intermittent fault will manifest as a non-recurring failure.

The probability θ is equal to the likelihood that a transient fault will afflict a machine that has already experienced a failure of the same type. This can be trivially estimated as the fraction of CPU time that accrued to machines that reported failures, relative to the total CPU time accrued to all machines.

The probability ϕ is the likelihood that a machine with an intermittent fault will not exhibit a recurring failure before the study period ends:

$$\phi = 1 - \int_0^\infty Pr[R = t \wedge W \geq t]\, dt$$

where R is the time between successive recurring failures and W is the time after a first failure that a machine remains in the study. This equation expands to:

$$\phi = 1 - \int_0^\infty Pr[W \geq t]\, Pr[R = t \,|\, W \geq t]\, dt$$

It is straightforward to measure both of the distributions in this equation. $Pr[W \geq t]$ is one minus the CDF of the time between the first failure and the end of the observation period, for all machines that experienced at least one failure. $Pr[R = t \,|\, W \geq t]$ is the PDF of the time between the first failure and the second failure, for all machines that experienced at least two failures during the observation period.

Acknowledgements

We thank our shepherd, Bianca Schroeder, and the anonymous reviewers from EuroSys 2011, OSDI 2010, and SOSP 2009 for comments that successively improved the paper. We thank Gretchen Loihle, Mingtian Ni, Miklos Szegedi, and Silviu Calinoiu for their help with the ATLAS and RAC databases, and insight into the nature of hardware failures. We thank Jeremy Elson, Chris Hawblitzel, Galen Hunt, Jay Lorch, James Mickens, David Molnar, and Reuben Olinsky for their insightful comments, and Landy Wang for providing insight into the Windows memory manager.

References

Lakshmi N. Bairavasundaram, Garth R. Goodson, Shankar Pasupathy, and Jiri Schindler. An Analysis of Latent Sector Errors in Disk Drives. In *Proceedings of the International Conference on Measurements and Modeling of Computer Systems (SIGMETRICS'07)*, San Diego, California, June 2007.

Lakshmi N. Bairavasundaram, Garth R. Goodson, Bianca Schroeder, Andrea C. Arpaci-Dusseau, and Remzi H. Arpaci-Dusseau. An Analysis of Data Corruption in the Storage Stack. In *Proceedings of the 6th USENIX Conference on File and Storage Technologies (FAST '08)*, San Jose, California, February 2008.

Thomas C. Bressoud and Fred B. Schneider. Hypervisor-based fault tolerance. *ACM Trans. Comput. Syst.*, 14(1):80–107, 1996. ISSN 0734-2071.

Cristian Constantinescu. Impact of deep submicron technology on dependability of vlsi circuits. In *International Conference on Dependable Systems and Networks (DSN'02)*, pages 205–209, 2002.

Cristian Constantinescu. Trends and challenges in vlsi circuit reliability. *IEEE Micro*, 23(4):14–19, July-Aug. 2003. ISSN 0272-1732.

Jean Dickinson Gibbons and Subhabrata Chakraborti. *Nonparametric Statistical Inference, Fourth Edition, Revised and Expanded*. CRC Press, 2003.

Kirk Glerum, Kinshuman Kinshumann, Steve Greenberg, Gabriel Aul, Vince Orgovan, Greg Nichols, David Grant, Gretchen Loihle, and Galen Hunt. Debugging in the (very) large: Ten years of implementation and experience. In *Proceedings of the 22nd ACM Symposium on Operating Systems Principles (SOSP '09)*, October 2009.

Jim Gray. Why do computers stop and what can we do about it. In *6th International Conference on Reliability and Distributed Databases*, 1987.

Jim Gray. A census of tandem system availability between 1985 and 1990. Technical Report 90.1, Tandem Computers, January 1990.

Intel. *Intel 64 and IA-32 Architectures Software Developers Manual*. Intel.

Weihang Jiang, Chongfeng Hu, Yuanyuan Zhou, and Arkady Kanevsky. Are disks the dominant contributor for storage failures? A comprehensive study of storage subsystem failure characteristics. In *6th USENIX Conference on File and Storage Technologies (FAST'08)*, San Jose, CA, February 2008.

Asmin Kadav, Matthew J. Renzelmann, and Michael M. Swift. Tolerating hardware device failures in software. In *22nd ACM Symposium on Operating Systems Principles (SOSP 09)*, pages 59–72, Big Sky, MT, October 2009.

M. Kalyanakrishnam, Z. Kalbarczyk, and R. Iyer. Failure data analysis of a lan of windows nt based computers. In *18th IEEE Symposium on Reliable Distributed Systems*, pages 178–187, 1999.

Micron. Dram soft eror rate calculations. Technical Report TN-04-28, Micron Technologies, 1994.

Micron. Module mean time between failures. Technical Report TN-04-45, Micron Technologies, 1997.

David Oppenheimer, Archana Ganapathi, and David A. Patterson. Why do internet services fail, and what can be done about it? In *4th Usenix Symposium on Internet Technologies and Systems (USITS 03)*, 2003.

David A. Patterson. An introduction to dependability. *login:*, 27 (4):61–65, August 2002.

Eduardo Pinheiro, Wolf-Dietrich Weber, and Luiz Andr Barroso. Failure trends in a large disk drive population. In *FAST '07: Proceedings of the 5th USENIX conference on File and Storage Technologies*, San Jose, CA, 2007.

Martin C. Rinard, Cristian Cadar, Daniel Dumitran, Daniel M. Roy, Tudor Leu, and Jr. William S. Beebee. Enhancing server availability and security through failure-oblivious computing. In *6th Symposium on Operating Systems Design and Implementation*, pages 303–316, San Francisco, CA, December 2004.

Bianca Schroeder and Garth A. Gibson. A large-scale study of failures in high-performance computing systems. In *Symposium on Dependable Systems and Networks (DSN 2006)*, 2006.

Bianca Schroeder and Garth A. Gibson. Disk failures in the real world: what does an mttf of 1,000,000 hours mean to you? In *Proceedings of the 5th USENIX conference on File and Storage Technologies (FAST 07)*, pages 1–16. USENIX Association, 2007.

Bianca Schroeder, Eduardo Pinheiro, and Wolf-Dietrich Weber. Dram errors in the wild:a large-scale field study. In *SIGMETRICS*, Seattle, WA, June 2009.

Norbet Seifert, David Moyer, Norman Leland, and Ray Hokinson. Historical trend in alpha-particle induced soft error rates of the alphatm microprocessor. In *39th Annual Reliability Physics Symposium*, pages 259–265, Orlando, FL, April 2001.

Dennis J. Wilkins. The bathtub curve and product failure behavior. Technical report, Fire-Dynamics.Com, October 2009.

Jun Xu, Z. Kalbarczyk, and R.K. Iyer. Networked windows nt system field failure data analysis. In *Pacific Rim International Symposium onDependable Computing*, pages 178–185, 1999.

J. F. Ziegler, H. W. Curtis, H. P. Muhlfeld, C. J. Montrose, B. Chin, M. Nicewicz, C. A. Russell, W. Y. Wang, L. B. Freeman, P. Hosier, L. E. LaFavea, J. L. Walsh, J. M. Orroa, G. J. Unger, J. M. Rossa, T. J. O'Gormana, B. Messinaa, T. D. Sullivana, A. J. Sykes, H. Yourkea, T. A. Enger, V. Tolat, T. S. Scotta, A. H. Taber, R. J. Sussman, W. A. Klein, and C. W. Wahaus. Ibm experiments in soft fails in computer elextronics. Technical Report 1, IBM, November 1996.

James F. Ziegler, Martin E. Nelson, James Dean Shell, Jerry Patterson, Carl J. Gelderloos, Hans P. Muhfield, and Charles J. Montrose. Cosmic ray soft error rates of 16-mb dram memory chips. *IEEE Journal of Solid-state circuits*, 33(2), February 1998.