may 2020

Crash report of the I/O failure at 1984 on 15 November 2017

1984

0. Table of contents

1. Preface

This report was commissioned by Iceland-based web hosting company 1984 to detail the total system failure that the company suffered on 15 November 2017, how its customers were affected by it, how recovery efforts went, and which factors led to the incident.

The report also investigates what lessons the company has drawn from the crash and which reforms have been put into place at 1984 in order to make sure that such an incident does not happen again.

The report was edited by Halldór Auðar Svansson and other contributors to it were Mörður Áslaugarson, Freysteinn Alfreðsson and Sigurður Þorfinnur Einarsson.

2. Executive summary

At the time of the crash, 1984 was running an IBM BladeCenter setup that had been put into place in 2011. The crash resulted in 9 out of 14 physical blades going down completely, which shut down 62 of 85 web servers, resulting in around 7.300 web sites going down and around 23.000 email users losing access to their inboxes, with approximately 200 VPS clients being affected as well.

Recovery efforts prioritized users‘ access to email, which took around a week to complete. Recovery of web sites took longer, with around 97% of web sites back up on 24 November with issues related to custom setups of web sites cropping up for around a month after that. Recovery was confounded by factors such as slow backups and a lot of manual effort being required in order to get everything back.

An analysis of the causes of the crash finds that it was not due to any outside cause, nor due to a hardware failure, but rather due to the system setup itself. Aberrant behavior in a SAN switch was enough to cause a cascading effect that ultimately brought down most of the physical blades. Although the precise cause of this could not be pinpointed with certainty, it is quite likely

that it can be traced to a bug in the Linux kernel, as only blades that were running a specific kernel version were affected.

The main lessons that 1984 has drawn from the incident are that greater redundancy is required in the system setup and that clear recovery procedures need to be put into place in order to allow for a rapid and firm response in worst-case scenarios. Reforms along these lines have been implemented at the company.

3. Introduction

On 15 November 2017, Iceland-based web hosting company 1984 encountered a failure of both their virtual machines and physical blades. The virtual machines and blades slowly started to fail one by one without any traceable cause. The first virtual machine went down in the night, was recovered but then immediately went down again. With the arrival of the morning and throughout the day, more and more virtual machines started to fail in the same manner.

By 16 November, 1984 had announced a “total system failure” as a result of the crash. The crash had affected a large number of web servers hosted at 1984, as well as mail servers and virtual private servers, rendering services utterly inaccessible for the majority of customers. 1984 promised regular customers a full recovery of their data from backup as well as vowed to recover the data of VPS customers to the extent possible. The company eventually delivered on that promise, although full recovery took a great deal of work and time.

This report details the timeline of the crash and recovery from it, as well as an overview of the efforts that were made to analyze the causes and their effects. It also summarizes what 1984 has learned from the crash, such as measures the company has put in place in order to ensure that a similar event does not reoccur and to increase redundancy in case it still does.

4. The system

At the time of the crash, 1984 was running on a 14-blade IBM BladeCenter setup. BladeCenter infrastructure components, such as IBM H BladeCenter chassis, network switches in BladeCenter, and SAN switches, were put into production in early 2011, in consultation with equipment vendor Nýherji. The hardware was not under warranty at the time of the crash, and 1984 did not have an active service contract with Nýherji.

More detailed information on the system setup at the time of the crash can be found in Appendix 1.

5. The crash

On 15 November 2017, at 3:21, a system administrator at 1984 received an alert message from Nagios, a tool used for health and status checks. The message reported that a virtual server was down in its entirety.

These alerts were unusual because usually only services running on the virtual server are reported down, such as web, database or mail services. In this case, however, the virtual server, and all the attached services crashed. While trying to boot the server again, it became clear that something more serious than the usual crash factors (kernel panic, overload, memory leaks, etc.) was at play. It turned out that the file system was corrupt and not repairable using standard tools.

At 3:39, another server on a different physical server (blade) went down in the same fashion. That one, however, was bootable and became available again within a minute.

After unsuccessfully trying to repair the filesystem on the virtual server that had been first to crash, 1984 staff decided to recover it from a day-old backup. The respective clients were notified, and 1984 began copying from backup to a new filesystem on a different physical server (blade). When the copying was finished, the server was started.

Thirty-two minutes later, after attempting a MySql repair, the new server crashed with the same symptoms. The filesystem was corrupt and beyond repair. 1984 had nothing else to try except to get the server image restored from backup again. They started copying again and then contacted a technician from their equipment vendor, Nýherji, who was familiar with their setup.

Two to three hours passed with no incident. However, the first blade then became unresponsive again, with a kernel panic being indicated in the console. The server was down, and 1984 staff attempted to boot it up again, but it would not boot because the boot sector and filesystem were corrupt. While they were attempting this, a massive, cascading crash started where blades essentially came crashing down like dominoes. It was at this point when it became readily apparent that the issue at hand was a system-wide crash.

6. Effects of the crash

The crash resulted in 9 out of 14 physical blades going down completely. These blades were hosting mixed services on virtual machines, such as shared hosting web and mail services; support services such as DNS, MX, SMTP; and Virtual Private Server clients.

At the time of the crash, 1984 hosted 85 web servers, 62 of which went down. This crash affected around 3,300 shared hosting clients, resulting in approximately 7,300 web sites going down. Around 23,000 e-mail users also lost access to their mailboxes, and approximately 200 VPS clients were affected as well.

7. Recovery

When it became apparent that what 1984 had on their hands was no ordinary malfunction or operational hitch, their equipment vendor (Nýherji) was notified. They immediately offered the use of their facilities and staff, and a crisis center and a crisis team were formed at the Nýherji offices. All hands were called on

deck. Nýherji contributed their foremost experts to the team, including experts in storage, networking, Linux systems administration and hardware technicians. This crisis team stayed on the case, monitoring developments and brainstorming about possible causes and possible courses of action, trying every trick in the book to contain the developments they witnessed.

Because of the nature of the crash, with so many physical servers having gone down completely, the recovery of users' data and restoration of services necessitated a restoration from backups.

Thankfully, all shared servers and mail had been backed up in their entirety to an encrypted server in Germany. However, the capability of the backup server was not enough to boot all these servers on that hardware, as this was a storage machine that was not meant to run any services. To attempt to get those backups up and running, 1984 rented multiple physical servers from the same ISP in Germany, to which they started to copy data from the backup server. This turned out to be extremely time consuming, as only a fixed amount of copy transfer could be run at once so as to not overload the network or smother the ability of the backup server.

Prioritizing e-mail recovery

When 1984 realized how time consuming the recovery process was and that the restoration could possibly take days, they tried to think of ways to quickly deal with the mail users who were backed up on the servers. This was seen as the number one priority. These mail users had had limited or no access to their mail for two days, while the MX servers were handling incoming mail queuing and waiting for a destination server to come online so they could deliver the mail.

On the morning of 17 November, 1984 decided to create multiple stand-alone mail servers, divide the mail users between them and then put them in production so that the MX servers could at least deliver new incoming mail to users. This turned out to be a great relief for the mail users; for even though they still had no access to their mail history or IMAP folders, at least they could receive the mail that was sent during the down time and thereafter. Following this, 1984 made scripts that successfully synced all mailboxes from backup. In a few cases, a backup had failed, but 1984 staff was able to recover that data from the ruins of the blade setup. This way, all mailboxes were successfully recovered and updated. The whole syncing and retrieving process took more than a week.

When mail services came back online, the main challenge was helping those who had forgotten their passwords that had been stored in mail clients to create new passwords. To deal with this challenge, a temporary solution was created for generating new passwords and sending them directly via SMS to the phone number on record.

Recovering web servers

Getting web services online turned out to be more problematic than 1984 had anticipated.

These were spread over around 70 instances of virtual servers along with MySql database servers. Many of the databases were large, some of them multiple gigabytes. In a few cases, 1984 staff were able to run the database server from backup and have MySql fix any issues using the innodb log. However, many of the database servers would not run at all or would only run if forced recovery levels were high (which usually meant a huge data loss), so these servers had to be restored using database dumps. In the worst instances, MySql user and privilege data was corrupt and had to be restored by parsing config files using custom-made scripts. These scripts would examine every database dump belonging to every customer and then attempt to find out what kind of system the user was running and extract the MySql password from config files. This approach worked for clients using the most common web systems, such as WordPress, Joomla and Drupal, but failed for other clients who were using custom systems. Those had to be fixed manually as complaints came in, and they came in at a high rate during the weeks following the crash.

The timeline of the recovery of web sites was as follows: On 18 November, 1984 announced that about a third of the web sites had been recovered. On 20 November, that number had gone up to around 80% and then up to about 97% on 24 November. Issues related to custom setups kept cropping up and were fixed for about a month after that.

Attempting to recover Virtual Private Servers

When this part of the recovery was more or less stable but there was still a lot of manual work to be done for hundreds of websites, 1984 turned to their

Virtual Private Servers, counting approximately 200 instances. Eventually, about 96% of VPSs were recovered. As all block devices were damaged, both the one holding the root filesystem of the host and those holding the filesystem for the virtual servers, 1984 staff had a difficult time retrieving data from the root system in order to be able to know the layout of the LUN devices, which were all using LVM.

Under LVM, the layout of the volume groups, physical devices and logical volumes is kept in the root system directory /etc/lvm. This data was unfortunately not backed up, but it was still recoverable from other sources within the filesystem. Using a tool called scalpel, 1984 staff ran through the boot disk trying to gather files that could be used to reconstruct the LVM on the physical media. This turned out to be a huge challenge, as what they retrieved was sometimes only partial data, backup data from a different state of the physical media, data from physical media that had been detached for some time and moved to another machine, etc. Fortunately, LVM is built in such a way that there is room for mistakes, so LVM recovery can be run on a physical device even if the recovery data is not from that disk. Even if it's old and skewed, it does not damage the disk. If that doesn't work, one can flush it and start again with another file; as long as one doesn't try to write to the logical volumes found on the disk, all is fine.

When 1984 staff thought they had retrieved the LVM for the physical media, they copied the block devices to another machine in another data center. In a few cases, when they started working on the copy, they found out they had the wrong layout and had to do it again. Fixing each image (copy of the block device) was problematic. They used testdisk to fix the filesystem, which did a fairly good job for a certain type of layout, such as for LVM layouts where the MBR had been wiped out but data was still retrievable. Because of the large scale, they decided they would not try to get these instances running again, but instead, they would make all the data available to the customers by putting it on an SFTP server and giving the customer access to it. A fresh VPS instance was also made available to the customer to restore his data to. That VPS was for 6 months free of charge, as an attempt to compensate the customer somewhat.

Needless to say, many of the VPS customers took their data and business elsewhere, while a few did stay on.

8. The aftermath – customer reactions

According to 1984, the reaction of customers when the crash took place was mostly positive, caring and supportive. They received greetings and messages of support from thousands of customers, wishing them well. 1984 knows of only about two dozen cases where shared hosting customers announced that they would be taking their business elsewhere. Many customers even offered to pay for a few years of service up front to support 1984.

The services rendered by 1984 are in nearly all cases in the form of subscriptions that are renewed after a period of time, so renewal rates are a good measure of whether business is being lost. They expected renewal rates to decrease significantly after the crash, but to the company's amazement, they stayed on target, as no statistically significant change in renewal rates was detected. The rate of new business, however, suffered the following year and was only at about 65% of the level that 1984 had estimated for that year. This turned out to be a temporary effect though, as by 2019, new business has recovered to the levels that had been estimated for that year before the crash happened.

All in all, it seems that a large majority of 1984's customers were understanding and stayed loyal to the company after the crash, and though the crash did mean a dip in new business for the company, even this seems to have been temporary. As a total loss of services for days on end can certainly be a very trying experience for many customers, this continued customer loyalty seems to indicate that the level of trust that 1984 enjoyed before the crash, as well as their handling of recovery efforts, helped to ameliorate what could otherwise have been a huge blow to the company's reputation and business.

9. Analysis of causes

Parallel to recovery efforts, 1984 staff and supporting personnel attempted to trace the causes of the crash. Analysis of causes goes hand in hand with recovery, as understanding the underlying causes facilitates efforts

to fix what has gone wrong. However, analysis deserves a chapter of its own, as it is a discipline of its own and has further goals than recovery. This analysis includes learning from mistakes in order to ensure that they will not happen again.

It was immediately apparent that due to the extent of the crash, the causes would need thorough analysis. 1984 laid out a plan to isolate the problem by analyzing different parts of the setup independently. Their priority was to create a map of the whole system. This map included all hardware and software components and how they were connected. Each component was then tested individually by using self-test features and custom-made tests.

However, in practice, the distinction between gathering information and drawing findings from it is not always clear-cut. This chapter splits the analysis into two main parts. The first part focuses on what information was gathered during the analysis, and the second focuses on the findings.

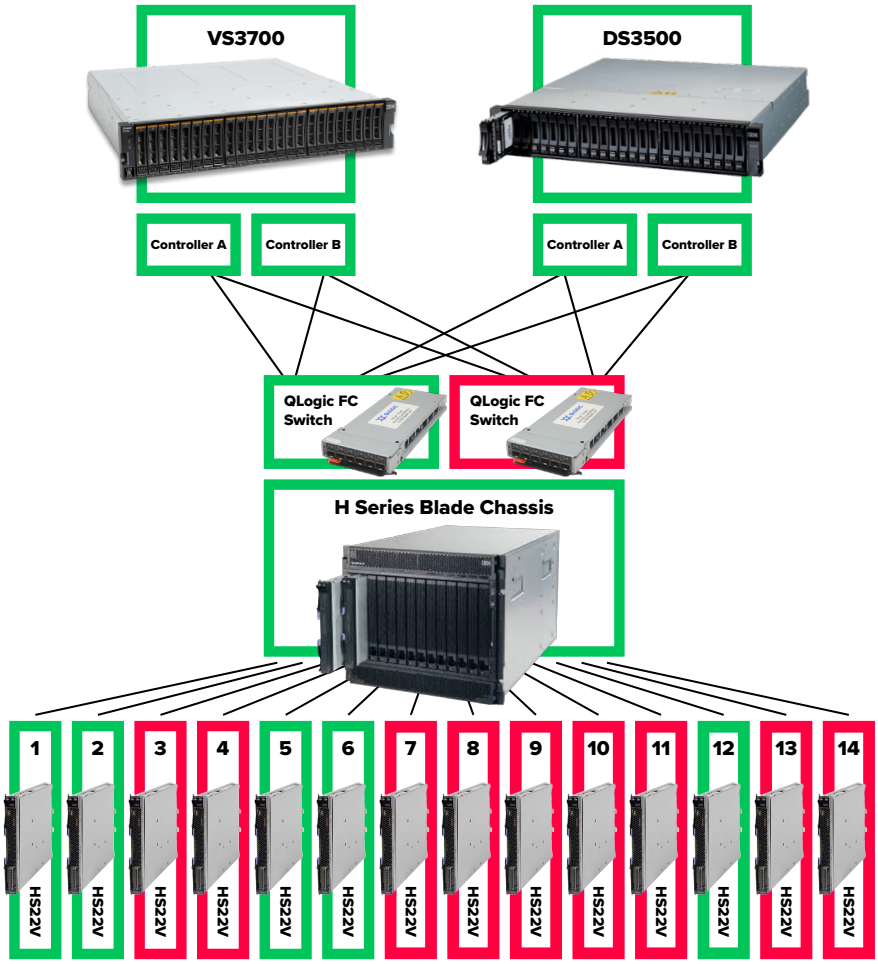

Figure 1: *Depicts the hardware setup affected by the crash. The hardware components in red were affected in the crash, while the components in green were working as intended.*

Information gathering

In this section, we list each component that was analyzed by 1984. Figure 1 shows a hardware overview of these components and highlights which components were affected in the crash.

Fiber-channel switches

Logs from the SANs and blades show that one of the switches dropped all connections. However, there were no visible problems with the fiber-channel switches in their management interface, nor was there anything in the fiber-channel logs that explained why it lost connectivity to the SANs. Due to the lack of information on the problem from the switches, 1984 replaced them after the analysis.

SANs

Nýherji analyzed the SANs and concluded that the hardware was functioning correctly. The logs showed that the fiber-channel ports lost connectivity to all four controllers on both SANs and that they had issued failover requests. Following the failover requests, there were numerous errors from the host-side of the type “not ready”, “ill requests” and “unit attentions”. Further analysis also revealed that the SANs host-tables showed that not all the blades were logged into the SANs.

SAN LUNs

All of the hypervisor operating system disks booted from the SAN. Therefore, the boot disks were analyzed by placing a local operating system disk into one of the blades and mapping all the LUNs to that machine. The analysis showed that some of the hypervisor disks were corrupt and that the MBR/GPT partition tables were compromised. No I/O pattern in the corrupt regions of the operating system disks was found.

Many of the virtual machine LUNs that were corrupt had zeroed-out regions. This I/O pattern often included the MBR/GPT table and the LVM metadata area, rendering the file-systems inaccessible.

Blades

All hardware self-tests ran without issues, indicating that the blades were functioning correctly. As can be seen in Figure 1, the crashed blades were not continuous but spread around the blade center. So no pattern was found there either. The 1984 team also installed an operating system on a local disk of one of the affected blades, which operated without issues.

Firmware Blade servers were taken in and out of service within this environment over the years, and firmware upgrades on blades were made irregularly and on no particular schedule. No firmware updates took place since the deployment of SAN switches, network switches and SAN controllers. However, a comparison of the BIOS/UEFI versions on the blades showed no pattern related to the crashed blades.

Hypervisors After logging into the hypervisors, it was apparent that the kernels were compromised. In most cases when an I/O request times out, the kernel creates a stack trace to indicate when and where it happened. However, in this case, the kernels had no stack traces and showed signs of corrupt memory. These signs were in the form of non-I/O-related system calls failing, sluggish behavior and corrupt information displayed in some proc and sysfs entries. Unfortunately, none of these hypervisors were running kdump. Therefore, it was not possible to create a kernel dump for further analysis of how the kernel became corrupt.

The compromised kernels were running the Debian Linux Kernel 3.16.0-4, except one, which was running 4.9.0-3. The uncompromised machines were all running different kernels that were either newer or much older—these kernels were 3.2.0-4, 4.9.0-4 and 4.9.0-6. However, one of the failed machine's kernel version could not be determined. All the kernels were using the open-source driver for fiber-channel.

The logs of all the hypervisors were analyzed and reconstructed from LUNs where the operating system was corrupt. From the recovered logs from the time of the crash, they showed that most problems related to block devices and that file systems started to appear after the lost fiber-channel paths to the SANs.

Findings

For readability, we decided to split this part into four sections: Malicious actor, Hardware, Software and Conclusions. This split is because different readers of the report may be primarily, or even solely, interested in certain aspects of the analysis and specific theories as to the causes. This split also corresponds to a logical progression of analysis, where the system is analyzed stepwise, from external causes, then on to the hardware and then finally

to the software. The final, brief Conclusions part summarizes the findings succinctly.

Malicious actor

In a full-system crash such as the one that 1984 experienced, the possibility of a malicious actor compromising the system from outside immediately springs to mind as a genuine possibility.

This possibility was taken very seriously. No indications of a malicious actor in any of the observed events become apparent through the information gathering process. The nature of the crash also was simply not consistent with that scenario, as this would have been a sophisticated method that would have required the attacker to have broad knowledge of the internal setup, as the machines are only accessible through vastly different virtual machines. That someone should go through such lengths when far more direct methods of taking a system down are available is simply not a likely event.

Even so, it was considered prudent to consult a security expert. To this end, Syndis ehf. was contacted and a security and cyberattack expert was immediately deployed to the crisis center at Nýherji's offices. Through this work, the possibility of an outside malicious actor was essentially ruled out entirely and effort was put into analyzing possible internal causes within the hardware and software.

Hardware

All hardware self-tests passed on all equipment, and after recovery, all of the hardware was usable with new functioning operating systems without issues.

Specifically, these tests were run in order to see if the fault could be reproduced:

1. The fio I/O benchmark tool was run onto a LUN from a local operating system disk from one of the blades. No loss on that LUN was reproduced, while the problem persisted on the other blades.
2. After all the data had been migrated off the SANs and the blade center, new operating systems running with the same setup as the crashed

blades were installed. The fio tool was then run to create an extensive I/O to the SANs for multiple hours without seeing any data loss nor any trace of the problem recurring.

As was the case with the scenario of a malicious actor having compromised the system, it was thus rather quickly deemed highly unlikely that a hardware malfunction caused the crash. This conclusion was further supported by the fact that the compromised blades were not physically continuous, but rather spread out around the blade center.

Software

Logs showed that all the problems emerged after the fiber-channel ports on one of the switches went down. It triggered the multipath functionality on all of the blades. However, only blades that were running specific versions of the Linux kernel were affected. When I/O times out in the Linux kernel, the kernel usually either produces a stack trace and kills the affected program or panics the kernel if the I/O is more critical. In the case of the crash, the operating system kernels were unstable and kept running even when there were I/O issues.

Unstable kernels can happen when a malicious actor exploits a kernel bug. However, it is far more likely that the crash happened when the fiber-channel switch dropped out. This is further supported by the fact that the problems started happening almost immediately after the fiber-channel switch dropped out. 1984, therefore, found that the most likely explanation for the crash was that there was a bug in the Linux kernel related to the open-source fiber-channel driver and multipath functionality, specifically within the kernel version that the affected blades were running.

A test was run where the blades were forced to change fiber-channel paths by disabling fiber-channel ports on the switches during the I/O load using the fio tool. The fault was not reproduced, and the multipath functionality worked without issues. It could therefore be concluded that if the fault were a result of a kernel bug, then the kernels would need to have been running for a long time. It is likely that kernel data structures would have needed to be in the correct position for the bug to be reproducible. Further analysis was also confounded by the fact that kdump was not running on any of the machines.

The argument that the crash was due to a kernel bug, therefore, remains a hypothesis, albeit a highly likely one.

Conclusions

— From the analysis of the causes of the crash, it is apparent that the fault does not lie with any malicious outside actor or with compromised security, but within the system setup.

— Specifically, the fault most likely lies with differing kernel versions. If this hypothesis is correct, the crash could have been avoided with more frequent kernel upgrades, but ironically, also with less frequent kernel upgrades, as both older and newer kernel versions handled the situation gracefully.

10. Lessons learned

A report on the crash would be incomplete without an honest discussion of what lessons 1984 has been able to draw from it. The following is a summary of the main lessons:

The most important lesson learned was to assume in the future, that the worst-case scenario can happen and to be ready for it when it occurs. 1984 had already explicitly taken this approach regarding security as a healthy way to prevent being compromised. However, when it came to hardware and setup, the worst-case scenario had not explicitly been assumed, as 1984 was under the impression that the setup was quite robust. Never in their wildest nightmares had they anticipated that their setup would come tumbling down as rapidly as it did. Recovery and analysis efforts were complicated by everyone scrambling to fix things quickly.

There was no standard operating procedure in place, and logs detailing who did what when were not kept. 1984 now assumes that this might happen again and has prepared a disaster recovery plan on that premise, as a part of the 1984 internal security policy.

A SAN / blade center setup was, at some point in time, considered to be

redundant. It was sold as such, but 1984 has learned from bitter experience that redundancy is not worth anything when there is a single point of failure. 1984 has, therefore, taken numerous steps in order to ensure a more robust and genuinely redundant setup, which are detailed in Appendix 2.

11. Appendix 1: The setup at the time of the crash

List of hardware components:

SANs

— V3700
— DS3500

SAN switches

— QLOGIC: Q-Logic 6-Port Fiber-Channel Switch Module (Part: 26K6479/26K6481)

Fiber-channel cards

— QLogic 4Gb Fiber-Channel Expansion Card (CIOv) (Part: 46M6067)

Blades chassis

— IBM H Series BladeCenter Chassis

Blades

— IBM HS22V (Type: 7871)

List of firmware:

SAN firmware

— DS3500/10.84.G5.30 and V3700/7.3.0.6

Disk firmware

— DS3500/SB27,B547 and V3700/B56J

SAN switch firmware

— 5.5.2.10.0 07/07/2009

Fiber-channel card firmware

— N/A (updated during and/or after the crash)

Blade management module firmware

— BPET68L 12/06/2018

Blade firmware

— P9E163A
— P9E151B
— P9E155B
— P9E158A
— P9E161A
— P9E163A

Please keep in mind that some firmware was updated during the crash. Unfortunately, it was not documented which firmware was updated.

List of software components:

Linuz distributions

— Debian Linux

Kernels (affected blades in bold

— Blade 1: 4.9.0-4-amd64
— Blade 2: 3.2.0-4-amd64
— **Blade 3: 3.16.0-4-amd64**
— **Blade 4: 3.16.0-4-amd64**
— **Blade 5: 4.9.0-3**
— Blade 6: 3.2.0-4-amd64
— **Blade 7: 3.16.0-4-amd64**
— **Blade 8: 3.16.0-4-amd64**
— **Blade 9: 3.16.0-4-amd64**
— **Blade 10: Unknown**
— **Blade 11: 3.16.0-4-amd64**
— Blade 12: 4.9.0-6-amd64
— **Blade 13: 3.16.0-4-amd64**
— **Blade 14: 3.16.0-4-amd64**

12. Appendix 2: The upgraded setup

System architecture and hardware

The cascading effect of the malfunctions that most likely caused the crash was made possible by the fact that the setup itself limited compartmentalization on the hardware level. The use of shared storage on the SAN created a situation where aberrant behavior in one hardware component (a SAN switch) affected a large part of the production setup, i.e. the blade servers, because they were all dependent on that component. 1984 has changed their hardware policy so that only stand-alone servers with local storage are in use on active production systems. The redundancy level enabled by the SAN setup is approached by other means on all new hardware under this new approach:

1. The local disks are set up in RAID 10.
2. The migration of virtual instances is quick and efficient, with minimum disruption.
3. Storage on all new hardware is done on enterprise-level SSD (NVMe) disks, increasing physical resilience and reliability.

The above signifies a changed general approach to systems operations on the 1984 hardware level.

Firmware and system software

1984 now has a comprehensive upgrade schedule in place for all equipment, both on the firmware level and the system software level. This is now a part of the internal security policy established in compliance with the General Data Protection Regulation (GDPR) as an Annex thereto.

Service redundancy and fallback

As opposed to previously operating a single data center, 1984 has now spread out to three data center locations running key services.

Mail

In the weeks after the crash, it became apparent that the unavailability of e-mail service had the highest general impact on users. Before the crash, the e-mail for shared hosting users was stored on the same server node as the web server and the database server for the user, as is standard practice in shared hosting environments. 1984 decided to change that. The mail service is now fully separated from the shared hosting environment and runs on dedicated mail storage servers. Those servers exist in a duplicate configuration so that a live, fully functional replica of the primary mail storage server is ready to take over service in case of a catastrophic failure of the primary one. The duplicate servers are in a different data center from the primary ones.

Web sites and user databases

Web services were all recovered from backups after the crash. The backup policy of 1984 proved to be fairly robust, but the sheer scale of the recovery was still problematic. Thousands of web sites and databases had to be recovered from backups and made operational again on new hardware in a different country, as the recovered web sites were made functional in Germany to begin with. This was inevitably time consuming, and full recovery of web services was only achieved about six weeks after the start of the crash.

1984 is a low-cost shared hosting provider, and as such, high-availability services have not been offered to customers. In the years before the crash, however, the trust placed in 1984's services steadily increased among users as a result of the increasingly long-term general stability and reliability of the company's services. 1984 values this trust highly and is committed to maintaining it as much as possible in spite of the crash, so even if they intend to remain a generally low-cost shared hosting provider, they have researched and decided on a course of action to make increased redundancy and higher operational resilience available to clients:

1. The WordPress CMS platform is currently by far the most popular among the users of shared hosting services; WordPress sites constitute about 75% of all web sites on the 1984 shared hosting platform. 1984 has signed a contract with a third party, Automattic Inc., to offer their service

packages for WordPress to customers. Backup of WordPress sites is included in many of those service packages, both daily automated backups and live automated backups. Those service packages will be available in the third quarter of 2020.

2. For users not wishing to start a business relationship with or entrust their data to a third party, an add-on service will also be provided directly by 1984, where sites and databases will have live, redundant instances in a different data center ready to take over service in case of catastrophic failure in the primary data center. This will achieve a live, automated backup for users. This service will be ready no later than the fourth quarter of 2020.

VPS service

1984 has introduced a snapshot feature, allowing the user to create either a live snapshot image or a cold snapshot which they can revert to and also download from the file server. We have also added the ability to migrate instances quickly to another host in case of an impending failure.

Nameservers/FreeDNS

While DNS services were unaffected by the crash, 1984 has still taken steps in order to make them more robust, spreading them to 4 different geological locations, running multiple nodes using multiple IPs. In the near future, 1984 plans on offering DNS on an anycast network as an add-on service to their FreeDNS service.


1984