the Availability Digest

www.availabilitydigest.com
@availabilitydig

How ‘Fat’ Are Your Fingers?

July 2017

What is a ‘fat finger?’ A fat finger can cause a command to be entered improperly. Fundamentally, a fat-finger error occurs when a key on a keyboard is depressed, and the finger depressing the key overlaps an adjoining key and depresses it also.

However, for purposes of this article, we extend the definition of a fat-finger error to include any erroneously entered keyboard command that unintentionally compromises an IT system or takes the system down. Fat-finger commands happen all too frequently. We give some examples of fat-finger outages in this article. We also suggest a simple fix that should eliminate fat-finger errors, though it seems that this procedure is seldom used.

The Origen of Fat-Finger Errors

The term ‘fat-finger error’ was originally coined in the brokerage industry to describe a trade that was improperly entered. An often quoted example occurred in December 2005 when an employee (a trainee) of Japanese broker Mizuho entered an order to sell 610,000 shares of J-Com, a manpower recruitment firm, for one yen when he meant to enter an order to sell one share at 610,000 yen.[1] The error cost Mizuho about $280 million. The head of the Tokyo Stock Exchange resigned over the error.

In May 2001 a dealer at Lehman Brothers’ London office wiped £30 billion off the London Stock Exchange when he inadvertently ordered sales of shares in blue-chip companies such as BP and AstraZeneca that were 100 times larger than intended. The hapless trader keyed in £300 million for a trade instead of £3 million.

Some Fat-Finger Outage Examples

According to the Ponemon Institute, at least 22% of data center outages are caused by fat-finger human errors. These outages have affected even the largest of organizations such as Amazon, New York City, Credit Suisse, and the State of Virginia. We describe these and other outages below and offer a simple fix for fat-finger errors after our examples.

The Alaska Permanent Fund and the $38 Billion Keystroke

Perhaps the most costly fat-finger error occurred in Alaska. Alaska is a major exporter of oil. Alaskan oil revenues are accumulated in a fund by the Alaska Department of Revenue, and the earnings from the

[1] ‘Fat finger’ trade costs Tokyo shares boss his job, *Independent*; December 21, 2005. http://www.independent.co.uk/news/world/asia/fat-finger-trade-costs-tokyo-shares-boss-his-job-520283.html

fund are distributed to Alaskan residents. Each resident receives about $1,000 per year from the fund. The fund balance is currently about USD $38 billion.

With a classic fat-finger error, a technician mistakenly deleted the oil-fund database. No problem – the database was backed up in another disk system. However, under the pressure of the moment, he managed to reformat the backup disk system.

Salvation still awaited around the corner. A recent version of the oil-fund database was resident on magnetic tape. However, the tape proved unreadable. Nine months of transactions were lost. It took a team of seventy people two months to rescan 800,000 documents stored in more than 300 cardboard boxes to manually recreate the database.

Console Command Takes Down Active/Active System

A European company, who for good reason does not want to be identified, has for years used a two-node active/active system to perform its critical processing. It was a common practice of the company to roll upgrades across the two nodes without affecting users.

The company would first route all transaction traffic to one node. It would then take the other node down and upgrade it. After thoroughly testing the upgraded node it would return it to service. It would then route all transaction traffic to the upgraded node and repeat the process for the other node.

Over the years, rolling upgrades through the two nodes in this fashion had become second nature to the IT staff. Unfortunately, this led to a degree of complacency that proved fatal to the system. During one upgrade, all users were moved from Node A to Node B. When the IT staff went to stop Node A so that it could be upgraded, a fat-finger error stopped Node B instead. The entire active/active system crashed.

Amazon Availability Zone Down for Four Days

Amazon is known for the high availability of its services. To ensure reliability, it operates 42 Availability Zones in sixteen regions around the world. A user may run his application in multiple Availability Zones to ensure continuity in the event of an Availability Zone failure.

Amazon scheduled maintenance to increase the capacity of its Availability Zone in its U.S. East Region. In order to accomplish this, it had to reroute traffic from that Availability Zone to another Availability Zone. However, instead of rerouting traffic to an alternate high-capacity network, it mistakenly routed it to a slower network. The long transmission delays effectively isolated the slower processing nodes from their data. The entire Availability Zone went down. It took Amazon four days to recover all services.

Amazon's S3 Storage Taken Down by 'Fat Finger'

It happened to Amazon again. Amazon provides its S3 Simple Storage Service to users so that they can store objects via a simple web interface. This has proven to be a highly reliable storage service, and many web sites depend upon S3 to store their data.

One day, a technician executed a command to remove a small number of servers for maintenance purposes. However, through a fat-finger error, he executed an incorrect command that removed a large set of servers. S3 went down, as did dozens of major web sites that depended upon S3. The outage lasted for eleven hours before all storage services were restored.

GitLab Suffers Massive Backup Failure

Git is a version control system used by several development organizations to manage their code versions. GitLab is a Git repository manager. To guarantee the availability of the stored code versions, GitLab replicates the versions to a backup system.

An unexpected massive increase in load caused GitLab’s data replication facility to fail. Replication had gotten so far behind that the data to be replicated was being rolled off the system before replication could occur.

To restore replication, the database of the secondary system had to be cleared. Through a fat-finger error, the database administrator mistakenly cleared the database of the primary server. About 300 GB of live production data was lost. It took 36 hours to recover services.

Australia’s Telstra Downed by Fat Finger

Telstra is Australia’s largest telecommunication company, serving over 70% of the Australian population. Telstra customers connect to Telstra via ten communication nodes. Should a node fail, the customers on the failed node can be switched to surviving nodes.

Using its normal operating procedures, Telstra took down a node for maintenance. Normally, the customers on that node would have switched to the other operating nodes. However, through a fat-finger error, the technician mistakenly switched all those customers back to the downed node.

10% of Telstra’s customers – about 1.5 million people and businesses – were without landline or mobile service for four hours until Telstra could reconfigure the system.

New York City’s 911 Service Goes Down

New York City installed a new 911 system in 2013. Due to the critical nature of the system, the city tested it exhaustively for six months before deploying it into service.

Nevertheless, the system went down four times in its first three days of service. Three of these outages were caused by problems with server and storage systems. However, one was caused when an operator mistakenly shut down a server. This caused all of the dispatch servers to crash.

The city reverted to manual operations to continue 911 services until the system could be restored. Fortunately, no one was put in danger during the outage.

TWC Internet Outage Affects Millions

Time Warner Cable is an American cable television company servicing tens of millions of subscribers. During one overnight maintenance procedure, an erroneous configuration was propagated throughout its national backbone network.

This resulted in a confused network that could not pass messages from sources to destinations. The outage left twelve million subscribers in 29 states without Internet service for four hours.

State of Virginia Down for Days

The U.S. State of Virginia lost dozens of computers for over a week due to a fat-finger error. One side of a redundant Storage Area Network (SAN) failed, but the SAN failed over and continued operating properly.

The resulting outage was not caused by an erroneously fat-finger-entered command, but rather by a fat-finger manual error. In an attempt to repair the SAN, a technician mistakenly pulled the good controller card. This not only took down the SAN but caused severe data corruption as writes to the SAN were interrupted in the middle of execution.

The SAN was rebuilt from magnetic tape, taking almost a week to recover. Even worse, all data since the last tape backup was lost.

Credit Suisse Fined for Fat Finger Error

Credit Suisse is a leading financial services company and trades regularly in several stock markets. In one batch order of 167 securities that it placed on the Australian Stock Exchange, it included some improper trades. It bought two securities at a price well above the last trade. These were the only trades in the securities that day and caused the prices of the two securities to jump inappropriately.

Credit Suisse was fined USD $74,000 by the Australian Securities and Investments Commission for the improper trades.

NAB Takes Out Partner Banks for Two Days

The National Australia Bank conducts nightly batch runs on behalf of its partner banks. Its customer banks depend upon the results of the batch run to conduct business the next day.

One night, an IT staff person at NAB uploaded an erroneous control file containing instructions for the batch processing cycle. The control file caused many transactions to be duplicated.

As a result, the partner banks had no reliable record of the previous days transactions and were unable to service their customers. It took almost two weeks before full customer servicing could be resumed.

WestHost Data Center Downed

This is another case of a data center being downed not by a fat-finger erroneous command being entered, but by a fat-finger action by a technician who did the wrong thing.

WestHost operates a major website hosting data center. Hundreds of web sites depend upon its services.

During a routing preventive maintenance test, a technician failed to disable the fire suppression system. The system was triggered, and the piercing noise from the sirens damaged several disks. Unfortunately, many of the backup disks were also in the same facility and were damaged.

It took six days for WestHost to recover all services.

Let’s Eliminate Fat-Finger Errors

The solution to fat-finger errors is quite straightforward. We employ redundancy in our servers and databases for redundancy. Why not employ redundancy with humans during critical operations?

If a critical command that can take a system down is to be entered, use two humans. One enters the command, while the other checks the command before it is executed. In this way, fat-finger errors can be eliminated.

Do any of you use this simple technique to ensure system availability? I have never seen it in action.