The Saddest Moment

JAMES MICKENS

Reprinted from ;login: *logout,* May 2013

James Mickens is a researcher in the Distributed Systems group at Microsoft's Redmond lab. His current research focuses on Web applications, with an emphasis on the design of JavaScript frameworks that allow developers to diagnose and fix bugs in widely deployed Web applications. James also works on fast, scalable storage systems for datacenters. James received his PhD in computer science from the University of Michigan, and a bachelor's degree in computer science from Georgia Tech.
mickens@microsoft.com

Whenever I go to a conference and I discover that there will be a presentation about Byzantine fault tolerance, I always feel an immediate, unshakable sense of sadness, kind of like when you realize that bad things can happen to good people, or that Keanu Reeves will almost certainly make more money than you over arbitrary time scales. Watching a presentation on Byzantine fault tolerance is similar to watching a foreign film from a depressing nation that used to be controlled by the Soviets—the only difference is that computers and networks are constantly failing instead of young Kapruskin being unable to reunite with the girl he fell in love with while he was working in a coal mine beneath an orphanage that was atop a prison that was inside the abstract concept of World War II. "How can you make a reliable computer service?" the presenter will ask in an innocent voice before continuing, "It may be difficult if you can't trust anything and the entire concept of happiness is a lie designed by unseen overlords of endless deceptive power." The presenter never explicitly says that last part, but everybody understands what's happening. Making distributed systems reliable is inherently impossible; we cling to Byzantine fault tolerance like Charlton Heston clings to his guns, hoping that a series of complex

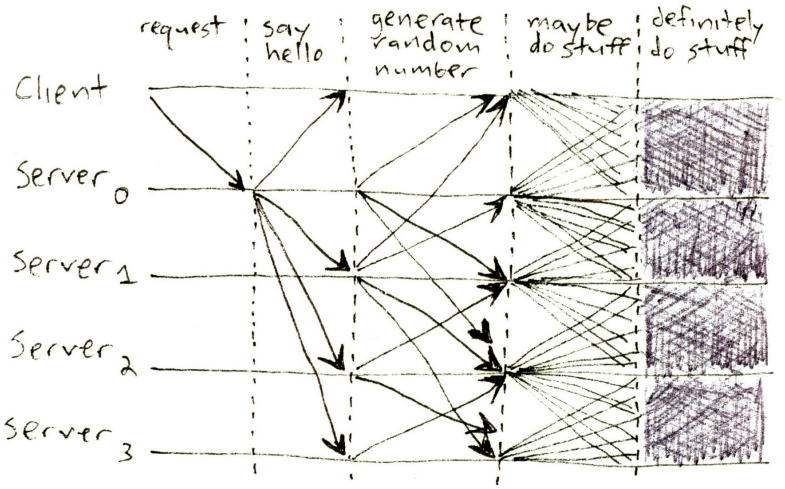

Figure 1: Typical Figure 2 from Byzantine fault paper: Our network protocol

software protocols will somehow protect us from the oncoming storm of furious apes who have somehow learned how to wear pants and maliciously tamper with our network packets.

Every paper on Byzantine fault tolerance contains a diagram that looks like Figure 1. The caption will say something like "Figure 2: Our network protocol." The caption should really say, "One day, a computer wanted to issue a command to an online service. This simple dream resulted in the generation of 16 gajillion messages. An attacker may try to interfere with the reception of 1/f of these messages. Luckily, 1/f is much less than a gajillion for any reasonable value of f. Thus, at least 15 gajillion messages will survive the attacker's interference. These messages will do things that only Cthulu understands; we are at peace with his dreadful mysteries, and we hope that you feel the same way. Note that, with careful optimization, only 14 gajillion messages are necessary. This is still too many messages; however, if the system sends fewer than 14 gajillion messages, it will be vulnerable to accusations that it only handles reasonable failure cases, and not the demented ones that previous researchers spitefully introduced in earlier papers in a desperate attempt to distinguish themselves from even more prior (yet similarly demented) work. As always, we are nailed to a cross of our own construction."

In a paper about Byzantine fault tolerance, the related work section will frequently say, "Compare the protocol diagram of our system to that of the best prior work. Our protocol is clearly better." The paper will present two graphs that look like Figure 2. Trying to determine which one of these hateful diagrams is better is like gazing at two unfathomable seaweed bundles that washed up on the beach and trying to determine which one is marginally less alienating. Listen, regardless of which Byzantine fault tolerance protocol you pick, Twitter will still have fewer than two nines of availability. As it turns out, Ted the Poorly Paid Datacenter Operator will not send 15 cryptographically signed messages before he accidentally spills coffee on the air conditioning unit and then overwrites your tape backups with bootleg recordings of Nickelback. Ted will just do these things and then go home, because that's what Ted does. His extensive home collection of "Thundercats" cartoons will not watch itself. Ted is needed, and Ted will heed the call of duty.

Every paper on Byzantine fault tolerance introduces a new kind of data consistency. This new type of consistency will have an ostensibly straightforward yet practically inscrutable name like "leap year triple-writer dirty-mirror asynchronous semi-consistency." In Section 3.2 ("An Intuitive Overview"), the authors will provide some plainspoken, spiritually appealing arguments about why their system prevents triple-conflicted write hazards in the presence of malicious servers and unexpected outbreaks of the bubonic plague. "Intuitively, a malicious server cannot lie to a client because each message is an encrypted, nested, signed, mutually-attested log entry with pointers to other encrypted and nested (but not signed) log entries."

Interestingly, these kinds of intuitive arguments are not intuitive. A successful intuitive explanation must invoke experiences that I have in real life. I have never had a real-life experience that resembled a Byzantine fault tolerant protocol. For example, suppose that I am at work, and I want to go to lunch with some of my co-workers. Here is what that experience would look like if it resembled a Byzantine fault tolerant protocol:

JAMES: I announce my desire to go to lunch.

BRYAN: I verify that I heard that you want to go to lunch.

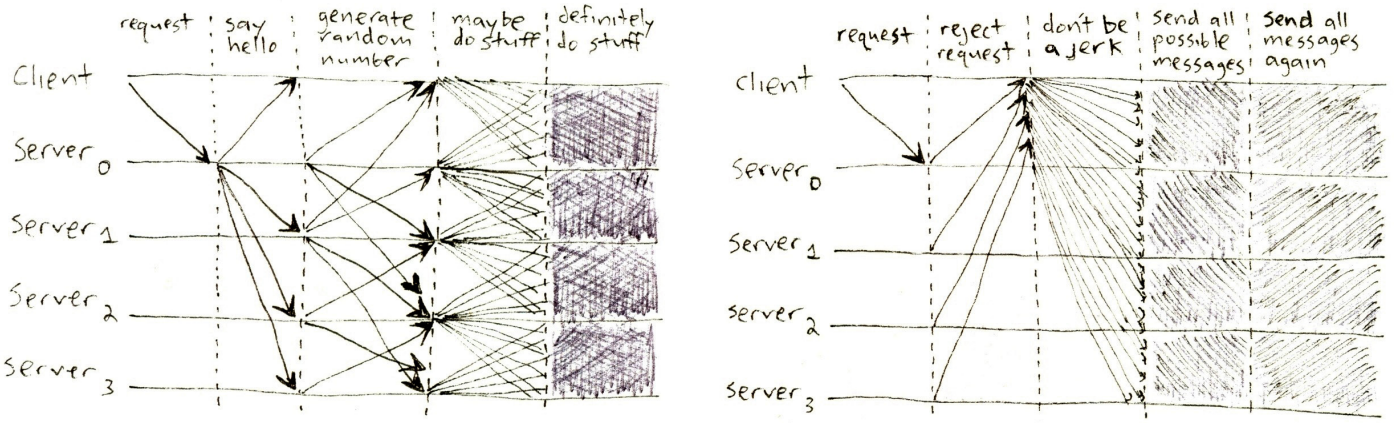

Figure 2: Our new protocol is clearly better.

RICH: I also verify that I heard that you want to go to lunch.

CHRIS: YOU DO NOT WANT TO GO TO LUNCH.

JAMES: OH NO. LET ME TELL YOU AGAIN THAT I WANT TO GO TO LUNCH.

CHRIS: YOU DO NOT WANT TO GO TO LUNCH.

BRYAN: CHRIS IS FAULTY.

CHRIS: CHRIS IS NOT FAULTY.

RICH: I VERIFY THAT BRYAN SAYS THAT CHRIS IS FAULTY.

BRYAN: I VERIFY MY VERIFICATION OF MY CLAIM THAT RICH CLAIMS THAT I KNOW CHRIS.

JAMES: I AM SO HUNGRY.

CHRIS: YOU ARE NOT HUNGRY.

RICH: I DECLARE CHRIS TO BE FAULTY.

CHRIS: I DECLARE RICH TO BE FAULTY.

JAMES: I DECLARE JAMES TO BE SLIPPING INTO A DIABETIC COMA.

RICH: I have already left for the cafeteria.

In conclusion, I think that humanity should stop publishing papers about Byzantine fault tolerance. I do not blame my fellow researchers for trying to publish in this area, in the same limited sense that I do not blame crackheads for wanting to acquire and then consume cocaine. The desire to make systems more reliable is a powerful one; unfortunately, this addiction, if left unchecked, will inescapably lead to madness and/or tech reports that contain 167 pages of diagrams and proofs. Even if we break the will of the machines with formalism and cryptography, we will never be able to put Ted inside of an encrypted, nested log, and while the datacenter burns and we frantically call Ted's pager, we will realize that Ted has already left for the cafeteria.

REGISTER TODAY

23rd USENIX Security Symposium

AUGUST 20–22, 2014 • SAN DIEGO, CA

The USENIX Security Symposium brings together researchers, practitioners, system programmers and engineers, and others interested in the latest advances in the security of computer systems and networks. The Symposium will include a 3-day technical program with refereed papers, invited talks, posters, panel discussions, and Birds-of-a-Feather sessions. Program highlights include:

Keynote Address by Phil Lapsley, author of *Exploding the Phone: The Untold Story of the Teenagers and Outlaws Who Hacked Ma Bell*

Invited Talk: "Battling Human Trafficking with Big Data" by Rolando R. Lopez, *Orphan Secure*

Panel Discussion: "The Future of Crypto: Getting from Here to Guarantees" with Daniel J. Bernstein, Matt Blaze, and Tanja Lange

Invited Talk: "Insight into the NSA's Weakening of Crypto Standards" by Joseph Menn, *Reuters*

The following co-located events will precede the Symposium on August 18–19, 2014:

EVT/WOTE '14: 2014 Electronic Voting Technology Workshop/Workshop on Trustworthy Elections
USENIX Journal of Election Technology and Systems (JETS)
Published in conjunction with EVT/WOTE
www.usenix.org/jets

CSET '14: 7th Workshop on Cyber Security Experimentation and Test

NEW! 3GSE '14: 2014 USENIX Summit on Gaming, Games, and Gamification in Security Education

FOCI '14: 4th USENIX Workshop on Free and Open Communications on the Internet

HotSec '14: 2014 USENIX Summit on Hot Topics in Security

HealthTech '14: 2014 USENIX Summit on Health Information Technologies
Safety, Security, Privacy, and Interoperability of Health Information Technologies

WOOT '14: 8th USENIX Workshop on Offensive Technologies

www.usenix.org/sec14

usenix
THE ADVANCED COMPUTING SYSTEMS ASSOCIATION

Stay Connected...

twitter.com/USENIXSecurity
www.usenix.org/youtube
www.usenix.org/gplus
www.usenix.org/facebook
www.usenix.org/linkedin
www.usenix.org/blog