Making Facebook Self-Healing

由 Alethea Power 于 2011年9月15日上午 8:35 发布

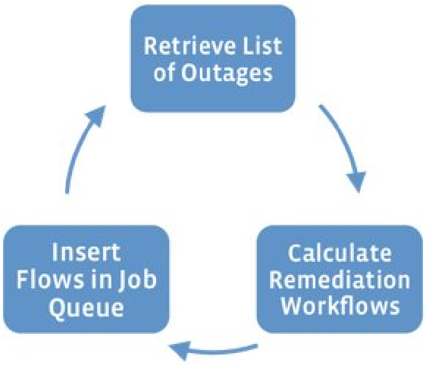
Facebook Engineering
Facebook Engineering 的网志
所有网志
经由RSS 获取网志
嵌入帖子

When your infrastructure is the size of Facebook’s, there are always broken servers and pieces of software that have gone down or are generally misbehaving. In most cases, our systems are engineered such that these issues cause little or no impact to people using the site. But sometimes small outages can become bigger outages, causing errors or poor performance on the site. If a piece of broken software or hardware does impact the site, then it's important that we fix it or replace it as quickly as possible. Even if it's not causing issues for users yet, it could in the future so we need to take care of it quickly.

Facebook's Site Reliability team is dedicated to keeping the site up and fast and stable. We handle everything from the smallest outages on individual servers to the largest outages across the entire site. When I joined the Site Reliability team a couple of years ago, it was clear that the infrastructure was growing too fast for us to be able to handle small repetitive outages manually. We had to find an automated way to handle these sorts of issues so that the human engineers could focus on solving and preventing the larger, more complex outages. So, I started writing scripts when I had time to automate the fixes for various types of broken servers and pieces of software.

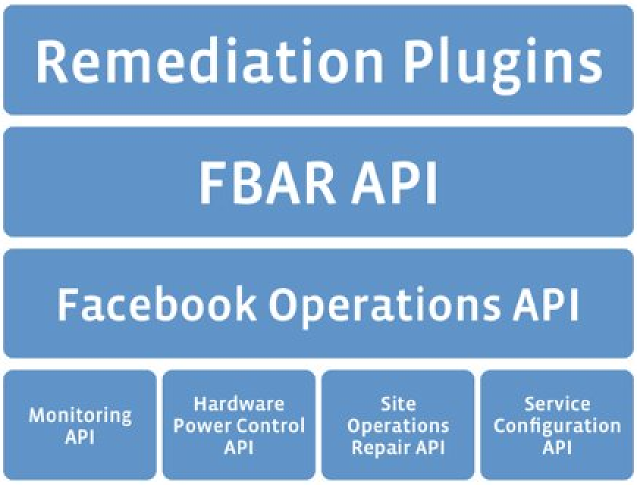
Introducing FBAR (Not FUBAR)
Over time, I developed the scripts more and more. As they got better, they saved me more time, which I used to continue improving them. Eventually my team started benefiting from my scripts enough that I was asked to work on them full time. I separated out the common parts into generic APIs that model our infrastructure and I turned the rest into remediation modules that use these APIs to implement the business logic for individual components of the Facebook back end. Then I wrote a daemonized service that executes workflows comprising these remediation plugins to handle outages detected by our monitoring system. I named the whole system "Facebook Auto-Remediation" or "FBAR" for short. (I originally wanted to name it "FUBAR", but I couldn't come up with anything good for the "U" to stand for, so "FBAR" it is.)

To understand how FBAR works, let's look at what happens when an individual server goes down. Imagine a hard drive goes bad on one of our Web servers. First, the monitoring system will detect the failed hardware and report this outage as an "alert". FBAR's "Alert Fetch Loop" runs continuously in the background querying the monitoring system to find new alerts. When it finds alerts, it processes them and calculates appropriate workflows to execute to handle the outages. The workflows get placed on a job queue for the FBAR Job Engine to execute.



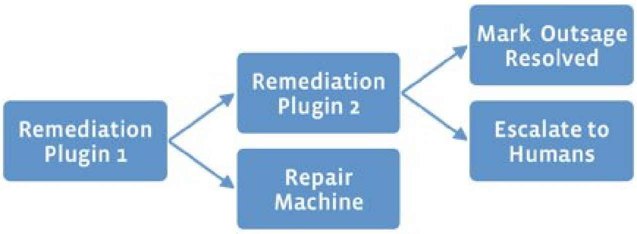
FBAR Alert Fetch Loop

The FBAR Job Engine will then pull the job for this server off of the job queue and begin executing remediation plugins in precedence order. Each plugin is written against the FBAR API. This API gives the plugin access to hardware and configuration data about the host and to the alert that describes the detected outage. The API also provides access to power control, command execution on the host and to the host's entries in our site-wide service configuration database.


Remediation Plugins
FBAR API
Facebook Operations API
Monitoring API
Hardware Power Control API
Site Operations Repair API
Service Configuration API

FBAR API Stack

When the job runs on our hypothetical Web server, the first remediation plugin would verify that the machine has damaged hardware, classify the failure type as *hard_drive*, then return that data to FBAR. At this point the workflow would branch. Rather than moving on to handle the next outage (like SSH or HTTP), FBAR would execute the plugin to remove the Web server from production service and then flag the machine as needing a part replacement.



Remediation Workflow

When the data center technician has replaced the bad drive on the machine, they would flag the machine as repaired. At this point, FBAR again takes control of the machine and verifies that it is ready for production service and re-enables it. The only human interaction with the machine is when a person replaces the physical hard drive. The rest of the process happens automatically without any manual intervention.

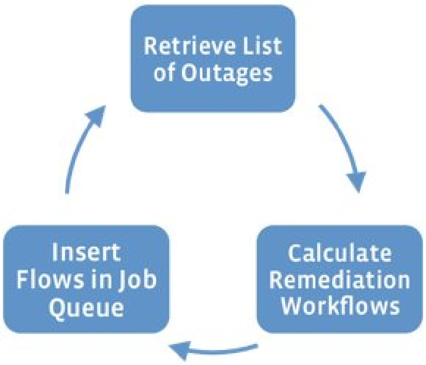
Automating the Work of Hundreds

Today, the FBAR service is developed and maintained by two full time engineers, but according to the most recent metrics, it’s doing the work of approximately 200 full time system administrators. FBAR now manages more than 50% of the Facebook infrastructure and we’ve found that services have dramatic increases in reliability when they go under FBAR control. Recently, we've opened up development of remediation plugins to other teams working on Facebook's back end services so they can implement their service-specific business logic. As these teams write their own remediation plugins, we're expanding FBAR coverage to more and more of the infrastructure. This is making the site more and more reliable for end users while reducing the workload of the supporting engineers.

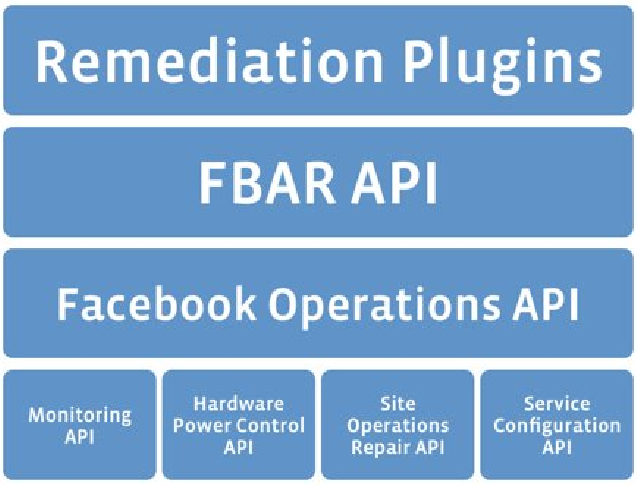
Facebook is an amazing place to work for many reasons but I think my favorite part of the job is that engineers like me are encouraged to come up with our own ideas and implement them. Management here is very technical and there is very little bureaucracy, so when someone builds something that works, it gets adopted quickly. Even though Facebook is one of the biggest websites in the world it still feels like a start-up work environment because there's so much room for individual employees to have a huge impact.

Like building infrastructure? Facebook is hiring infrastructure engineers. Apply here.

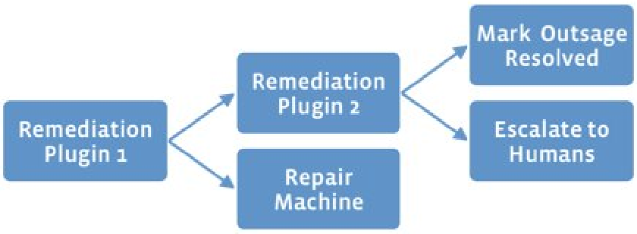
Patrick is a software engineer at Facebook.



FBAR Alert Fetch Loop


Remediation Plugins
FBAR API
Facebook Operations API
Monitoring API
Hardware Power Control API
Site Operations Repair API
Service Configuration API

FBAR API Stack



Remediation Workflow

赞 · 评论 · 分享

Golak Sarangi 、 Kapil Mohan 、 Sylvain Kalache 和其他 948 人赞了

214个分享

查看之前的评论

Boku No Graham A dream come true for stalkers and nosy parents alike and I doubt even the people who fall in those categories appreciate anyone on their friends list being easily to see every activity they do thanks to the ticker nonsense. arf, arf
2011年9月21日下午 11:14 · 4

Liz McMillan STOP w/the yellow message Box Bubble it's annoying!!!
2011年9月22日上午 6:35 · 2

Rajkumar Letsride Mundel As per me, There is no need of Right Hand side link for Profile near Home link. As we have already big link on left hand side for profile. Hope you like Facebook Engineering Facebook Facebook India
2011年9月22日上午 11:08 · 4

Paul Terrill FB Engineers -- use your time making FB more reliable; leave the interface alone! Listen to the people or expect revolt.
2011年9月22日下午 1:53 · 3

Veronica Tamayo can't use photo site goes into newsfeed?
2011年9月23日上午 3:51

Kevin Murphy i disagree. facebook needs to work on foundation. getting top heavy: when you can not even finnish typing a simple note before fb switches pages is bad engineering. navigation within fb is too hard. needs work trust me. if you do not believe me just ask around and see what you find. maybe if you type with one or two fingers but...
2011年9月23日上午 5:04 · 3

Ajeet Singh Panwar I m requesting to all frnds that send me frnd request i want to conect new frnds nd i want to share my experience with him so those who intrsted in me pls send me frnd request.
2011年9月23日上午 8:01