(http://www.flickr.com/photos/ashesoftimes/8079234402)

Chris Ball (https://blog.printf.net/)

(HTTPS://BLOG.PRINTF.NET/)

External links:

photographs (https://plus.google.com/u/0/photos/112938759017605010116/albums/5637573932501306529)

on github (http://github.com/cjb)

on google+ (http://gplus.to/chrisball)

on linkedin (http://www.linkedin.com/in/chrisjball)

on twitter (http://twitter.com/cjbprime)

on youtube (http://youtube.com/user/cjbprime/videos)

Most popular tags:

ethics (/articles/tag/ethics)

gnome (/articles/tag/gnome)

guitar (/articles/tag/guitar)

kernel (/articles/tag/kernel)

linux (/articles/tag/linux)

olpc (/articles/tag/olpc)

photography (/articles/tag/photography)

WP-SVBTLE (HTTPS://GITHUB.COM/GRAVITYONMARS/WP-SVBTLE)

Announcing GitTorrent: A Decentralized GitHub (https://blog.printf.net/articles/2015/05/29/announcing-gittorrent-a-decentralized-github/)

85 COMMENTS (HTTPS://BLOG.PRINTF.NET/ARTICLES/2015/05/29/ANNOUNCING-GITTORRENT-A-DECENTRALIZED-GITHUB/#DISQUS_THREAD)

May 29, 2015

(This post is an *aspirational transcript* of the talk I gave to the Data Terra Nemo (http://dtn.is) conference in May 2015. If you'd like to watch the less eloquent version of the same talk that I *actually* gave, the video should be available soon!)

I've been working on building a decentralized GitHub, and I'd like to talk about what this means and why it matters — and more importantly, show you how it can be done and real GitTorrent code I've implemented so far.

Why a decentralized GitHub?

First, the practical reasons: GitHub might become untrustworthy, get hacked — or get DDOS'd by China, as happened (http://arstechnica.com/security/2015/03/massive-denial-of-service-attack-on-github-tied-to-chinese-government/) while I was working on this project! I know GitHub seems to be doing many things right at the moment, but there often comes a point at which companies that have raised $100M in Venture Capital funding start making decisions that their users would strongly prefer them not to.

There are philosophical reasons, too: GitHub is closed source, so we can't make it better ourselves. Mako Hill (http://mako.cc) has an essay called Free Software Needs Free Tools (http://mako.cc/writing/hill-free_tools.html), which describes the problems with depending on proprietary software to produce free software, and I think he's right. To look at it another way: the experience of our collaboration around open source projects is currently being defined by the unmodifiable tools that GitHub has decided that we should use.

So that's the practical and philosophical, and I guess I'll call the third reason the "ironical". It is a massive irony to move from many servers running the CVS and Subversion protocols, to a single centralized server speaking the decentralized Git protocol. Google Code announced its shutdown (http://google-opensource.blogspot.com/2015/03/farewell-to-google-code.html) a few months ago, and their rationale was explicitly along the lines of "everyone's using GitHub anyway, so we don't need to exist anymore". We're quickly heading towards a single central service for all of the world's source code.

So, especially at this conference, I expect you'll agree with me that this level of centralization is unwise.

Isn't Git already decentralized?

You might be thinking that while GitHub is centralized, the Git protocol is decentralized — when you clone a repository, your copy is as good as anyone else's. Isn't that enough?

I don't think so, and to explain why I'd like you to imagine someone arguing that we can do without BitTorrent because we have FTP. We would not advocate replacing BitTorrent with FTP, and the suggestion doesn't even make sense! First — there's no index of which hosts have which files in FTP, so we wouldn't know where to look for anything. And second — even if we knew who owned copies of the file we wanted, those computers aren't going to be running an anonymous FTP server.

Just like Git, FTP doesn't turn clients into servers in the way that a peer-to-peer protocol does. So that's why Git isn't *already* the decentralized GitHub — you don't know where anything's stored, and even if you did, those machines aren't running Git servers that you're allowed to talk to. I think we can fix that.

Let's GitTorrent a repo!

Let's jump in with a demo of GitTorrent – that is, cloning a Git repository that's hosted on BitTorrent:

(http://www.flickr.com/photos/ashesoftimes/8079234402)

Chris Ball (https://blog.printf.net/)

(HTTPS://BLOG.PRINTF.NET/)

External links:

photographs (https://plus.google.com/u/0/photos/112938759017605010116/albums/5637573932501306529)

on github (http://github.com/cjb)

on google+ (http://gplus.to/chrisball)

on linkedin (http://www.linkedin.com/in/chrisjball)

on twitter (http://twitter.com/cjbprime)

on youtube (http://youtube.com/user/cjbprime/videos)

Most popular tags:

ethics (/articles/tag/ethics)

gnome (/articles/tag/gnome)

guitar (/articles/tag/guitar)

kernel (/articles/tag/kernel)

linux (/articles/tag/linux)

olpc (/articles/tag/olpc)

photography (/articles/tag/photography)

WP-SVBTLE (HTTPS://GITHUB.COM/GRAVITYONMARS/WP-SVBTLE)

```
1  λ git clone gittorrent://github.com/cjb/recursers
2  Cloning into 'recursers'...
3
4  Okay, we want to get: 5fbfea8de70ddc686dafdd24b690893f98eb9475
5
6  Adding swarm peer: 192.34.86.36:30000
7
8  Downloading git pack with infohash: 9d98510a9fee5d3f603e08dcb565f0675bd4b6a2
9
10 Receiving objects: 100% (47/47), 11.47 KiB | 0 bytes/s, done.
11 Resolving deltas: 100% (10/10), done.
12 Checking connectivity... done.
```

Hey everyone: we just cloned a git repository over BitTorrent! So, let's go through this line by line.

Lines 1-2: Git actually has an extensible mechanism for network protocols built in. The way it works is that my `git clone` line gets turned into "run the `git-remote-gittorrent` command and give it the URL as an argument". So we can do whatever we want to perform the actual download, and we're responsible for writing git objects into the new directory and telling Git when we're done, and we didn't have to modify Git at all to make this work.

So `git-remote-gittorrent` takes it from here. First we connect to GitHub to find out what the latest revision for this repository is, so that we know what we want to get. GitHub tells us it's `5fbfea8de..`.

Lines 4-6: Then we go out to the GitTorrent network, which is a distributed hash table just like BitTorrent's, and ask if anyone has a copy of commit `5fbdea8de..`. Someone said yes! We make a GitTorrent connection to them. The way that BitTorrent's distributed hash table works is that there's a single operation, `get_nodes(hash)` which tells you who can send you content that you want, like this:

```
get_nodes('5fbfea8de70ddc686dafdd24b690893f98eb9475') =
  [192.34.86.36:30000, ...]
```

Now, in standard BitTorrent with "trackerless torrents", you ask for the *files* that you want by their content, and you'd get them and be happy. But a repository the size of the Linux kernel has four million commits, so just receiving the one commit `5fbdea8de..` wouldn't be helpful; we'd have to make another four million requests for all the other commits too. Nor do we want to get every commit in the repository every time we 'git pull'. So we have to do something else.

Lines 8-12: Git has solved this problem — it has this "smart protocol format" for negotiating an exchange of git objects. We can think of it this way:

Imagine that your repository has 20 commits, 1-20. And the 15th commit is `bbbb` and the most recent 20th commit is `aaaa`. The Git protocol negotiation would look like this:

```
1> have aaaa
2> want aaaa
2> have bbbb
```

Because of the way the git graph works, node 1> here can look up where `bbbb` is on the graph, see that you're only asking for five commits, and create you a "packfile" with just those objects. Just by a three-step communication.

That's what we're doing here with GitTorrent. We ask for the commit we want and connect to a node with BitTorrent, but once connected we conduct this Smart Protocol negotiation in an overlay connection on top of the BitTorrent wire protocol, in what's called a BitTorrent Extension. Then the remote node makes us a packfile and tells us the hash of that packfile, and then we start downloading that packfile from it and any other nodes who are seeding it using Standard BitTorrent. We can authenticate the packfile we receive, because after we uncompress it we know which Git commit our graph is supposed to end up at; if we don't end up there, the other node lied to us, and we should try talking to someone else instead.

So that's what just happened in this terminal. We got a packfile made for us with this hash — and it's one that includes every object because this is a fresh clone — we downloaded and unpacked it, and now we have a local git repository.

This was a git clone where everything up to the actual downloading of git objects happened as it would in the normal GitHub way. If GitHub decided tomorrow that it's sick of being in the disks and bandwidth business, it could encourage its users to run this version of GitTorrent, and it would be like having a peer

(http://www.flickr.com/photos/ashesoftimes/8079234402)

Chris Ball (https://blog.printf.net/)

(HTTPS://BLOG.PRINTF.NET/)

External links:

photographs (https://plus.google.com/u/0/photos/112938759017605010116/albums/5637573932501306529)

on github (http://github.com/cjb)

on google+ (http://gplus.to/chrisball)

on linkedin (http://www.linkedin.com/in/chrisjball)

on twitter (http://twitter.com/cjbprime)

on youtube (http://youtube.com/user/cjbprime/videos)

Most popular tags:

ethics (/articles/tag/ethics)

gnome (/articles/tag/gnome)

guitar (/articles/tag/guitar)

kernel (/articles/tag/kernel)

linux (/articles/tag/linux)

olpc (/articles/tag/olpc)

photography (/articles/tag/photography)

WP-SVBTLE (HTTPS://GITHUB.COM/GRAVITYONMARS/WP-SVBTLE)

to peer "content delivery network" (http://en.wikipedia.org/wiki/Content_delivery_network) for GitHub, falling back to using GitHub's servers in the case where the commits you want aren't already present in the CDN.

Was that actually decentralized?

That's some progress, but you'll have noticed that the very first thing we did was talk to GitHub to find out which hash we were ultimately aiming for. If we're really trying to decentralize GitHub, we'll need to do much better than that, which means we need some way for the owner of a repository to let us know what the hash of the latest version of that repository is. In short, we now have a global database of git objects that we can download, but now we need to know what objects we want — we need to emulate the part of github where you go to `/user/repo`, and you know that you're receiving the very latest version of that user's repo.

So, let's do better. When all you have is a hammer, everything looks like a nail, and my hammer is this distributed hash table we just built to keep track of which nodes have which commits. Very recently, substack (http://substack.net/) noticed that there's a BitTorrent extension for making each node be partly responsible for maintaining a network-wide key-value store, and he coded it up. It adds two more operations to the DHT, `get()` and `put()`, and `put()` gives you 1000 bytes per key to place a message into the network that can be looked up later, with your answer repeated by other nodes after you've left the network. There are two types of key — the first is immutable keys, which work as you might expect, you just take the hash of the data you want to store, and your data is stored with that hash as the key.

The second type of key is a mutable key, and in this case the key you look up is the hash of a public key to a crypto keypair, and the owner of that keypair can publish signed updates as values under that key. Updates come with a sequence number, so anytime a client sees an update for a mutable key, it checks if the update has a newer sequence number than the value it's currently recorded, and it checks if the update is signed by the public key corresponding to the hash table key, which proves that the update came from the key's owner. If both of those things are true then it'll update to this newer value and start redistributing it. This has many possible uses, but my use for it is as the place to store what your repositories are called and what their latest revision is. So you'd make a local Git commit, push it to the network, and push an update to your personal mutable key that reflects that there's a new latest commit. Here's a code description of the new operations:

```
// Immutable key put
hash(value) = put({
  value: 'some data'
})

// Mutable key put
hash(key) = put({
  value: 'some data',
  key: key,
  seq: n
})

// Get
value = get(hash)
```

So now if I want to tell someone to clone my GitHub repo on GitTorrent, I don't give them the github.com URL, instead I give them this long hex number that is the hash of my public key, which is used as a mutable key on the distributed hash table.

Here's a demo of that:

(http://www.flickr.com/photos/ashesoftimes/8079234402)

Chris Ball (https://blog.printf.net/)

(HTTPS://BLOG.PRINTF.NET/)

External links:

photographs (https://plus.google.com/u/0/photos/112938759017605010116/albums/5637573932501306529)

on github (http://github.com/cjb)

on google+ (http://gplus.to/chrisball)

on linkedin (http://www.linkedin.com/in/chrisjball)

on twitter (http://twitter.com/cjbprime)

on youtube (http://youtube.com/user/cjbprime/videos)

Most popular tags:

ethics (/articles/tag/ethics)

gnome (/articles/tag/gnome)

guitar (/articles/tag/guitar)

kernel (/articles/tag/kernel)

linux (/articles/tag/linux)

olpc (/articles/tag/olpc)

photography (/articles/tag/photography)

WP-SVBTLE (HTTPS://GITHUB.COM/GRAVITYONMARS/WP-SVBTLE)

```
λ git clone gittorrent://81e24205d4bac8496d3e13282c90ead5045f09ea/recursers

Cloning into 'recursers'...

Mutable key 81e24205d4bac8496d3e13282c90ead5045f09ea returned:
name:         Chris Ball
email:        chris@printf.net
repositories:
  recursers:
    master: 5fbfea8de70ddc686dafdd24b690893f98eb9475

Okay, we want to get: 5fbfea8de70ddc686dafdd24b690893f98eb9475

Adding swarm peer: 192.34.86.36:30000

Downloading git pack with infohash: 9d98510a9fee5d3f603e08dcb565f0675bd4b6a2

Receiving objects: 100% (47/47), 11.47 KiB | 0 bytes/s, done.
Resolving deltas: 100% (10/10), done.
Checking connectivity... done.
```

In this demo we again cloned a Git repository over BitTorrent, but we didn't need to talk to GitHub at all, because we found out what commit we were aiming for by asking our distributed hash table instead. Now we've got true decentralization for our Git downloads!

There's one final dissatisfaction here, which is that long strings of hex digits do not make convenient usernames. We've actually reached the limits of what we can achieve with our trusty distributed hash table because usernames are rivalrous, meaning that two different people could submit updates claiming ownership of the same username, and we wouldn't have any way to resolve their argument. We need a method of "distributed consensus" to give out usernames and know who their owners are. The method I find most promising is actually Bitcoin's blockchain (http://en.wikipedia.org/wiki/Block_chain_(transaction_database)) — the shared consensus that makes this cryptocurrency possible.

The deal is that there's a certain type of Bitcoin transaction, called an OP_RETURN transaction (http://blog.coinprism.com/2015/02/11/80-bytes-op-return/), that instead of transferring money from one wallet to another, leaves a comment as your transaction that gets embedded in the blockchain forever. Until recently you were limited to 40 bytes of comment per transaction, and it's been raised to 80 bytes per transaction (https://github.com/bitcoin/bitcoin/commit/fcf646c9b08) as of Bitcoin Core 0.11. Making any Bitcoin transaction on the blockchain I believe currently costs around $0.08 USD, so you pay your 8 cents to the miners and the network in compensation for polluting the blockchain with your 80 bytes of data.

If we can leave comments on the blockchain, then we can leave a comment saying "Hey, I'd like the username Chris, and the hash of my public key is <*x*>", and if multiple people ask for the same username, this time we'll all agree on which public key asked for it first, because blockchains are an append-only data structure where everyone can see the full history. That's the real beauty of Bitcoin — this currency stuff is frankly kind of uninteresting to me, but they figured out how to solve distributed consensus in a robust way. So the comment in the transaction might be:

```
@gittorrent!cjb!81e24205d4bac8496d3e13282c90ead5045f09ea

(@service!username!pubkey)
```

It's interesting, though — maybe that "gittorrent" at the beginning doesn't have to be there at all. Maybe this could be a way to register one username for every site that's interested in decentralized user accounts with Bitcoin, and then you'd already own that username on all of them. This could be a separate module, a separate software project, that you drop in to your decentralized app to get user accounts that Just Work, in Python or Node or Go or whatever you're writing software in. Maybe the app would monitor the blockchain and write to a database table, and then there'd be a plugin for web and network service frameworks that knows how to understand the contents of that table.

It surprised me that nothing like this seems to exist already in the decentralization community. I'd be happy to work on a project like this and make GitTorrent sit on top of it, so please let me know if you're interested in helping with that.

By the way, username registration becomes a little more complicated than I just said, because the miners could see your message, and decide to replace it before adding it to the blockchain, as a registration of your username to *them* instead of you. This is the equivalent of going to a domain name registrar and

(http://www.flickr.com/photos/ashesoftimes/8079234402/)

Chris Ball (https://blog.printf.net/)

(HTTPS://BLOG.PRINTF.NET/)

External links:

photographs (https://plus.google.com/u/0/photos/112938759017605010116/albums/5637573932501306529)

on github (http://github.com/cjb)

on google+ (http://gplus.to/chrisball)

on linkedin (http://www.linkedin.com/in/chrisjball)

on twitter (http://twitter.com/cjbprime)

on youtube (http://youtube.com/user/cjbprime/videos)

Most popular tags:

ethics (/articles/tag/ethics)

gnome (/articles/tag/gnome)

guitar (/articles/tag/guitar)

kernel (/articles/tag/kernel)

linux (/articles/tag/linux)

olpc (/articles/tag/olpc)

photography (/articles/tag/photography)

WP-SVBTLE (HTTPS://GITHUB.COM/GRAVITYONMARS/WP-SVBTLE)

typing the domain you want in their search box to see if it's available — and at that moment of your search the registrar could turn around and register it for themselves, and then tell you to pay them a thousand bucks to give it to you. It's no good.

If you care about avoiding this, Bitcoin has a way around it, and it works by making registration a two-step process. Your first message would be asking to reserve a username by supplying just the hash of that username. The miners don't know from the hash what the username is so they can't beat you to registering it, and once you see that your reservation's been included in the blockchain and that no-one else got a reservation in first, you can send on a second comment that says "okay, now I want to use my reservation token, and here's the plain text of that username that I reserved". Then it's yours.

(I didn't invent this scheme. There's a project called Blockname (https://github.com/telehash/blockname), from Jeremie Miller (https://twitter.com/jeremie), that works in exactly this way, using Bitcoin's OP_RETURN transaction for DNS registrations on bitcoin's blockchain. The only difference is that Blockname is performing domain name registrations, and I'm performing a mapping from usernames to hashes of public keys. I've also just been pointed at Blockstore (https://github.com/namesystem/blockstore), which is extremely similar.)

So to wrap up, we've created a global BitTorrent swarm of Git objects, and worked on user account registration so that we can go from a user experience that looks like this:

```
git clone gittorrent://github.com/cjb/foo
```

to this:

```
git clone gittorrent://81e24205d4bac8496d3e13282c90ead5045f09ea/foo
```

to this:

```
git clone gittorrent://cjb/foo
```

And at this point I think we've arrived at a decentralized replacement for the core feature of GitHub: finding and downloading Git repositories.

Closing thoughts

There's still plenty more to do — for example, this doesn't do anything with comments or issues or pull requests, which are all very important aspects of GitHub.

For issues, the solution I like is actually storing issues in files inside the code repository, which gives you nice properties like merging a branch means applying both the code changes and the issue changes — such as resolving an issue — on that branch. One implementation of this idea is Bugs Everywhere (http://bugseverywhere.org).

We could also imagine issues and pull requests living on Secure Scuttlebutt (https://github.com/ssbc/secure-scuttlebutt), which synchronizes append-only message streams across decentralized networks.

I'm happy just to have got this far, though, and I'd love to hear your comments on this design. The design of GitTorrent itself is (ironically enough) on GitHub (https://github.com/cjb/GitTorrent/blob/master/README.md) and I'd welcome pull requests to make any aspect of it better.

I'd like to say a few thank yous — first to Feross Aboukhadijeh (http://feross.org), who wrote the BitTorrent libraries that I'm using here. Feross's enthusiasm for peer-to-peer and the way that he runs community around his "mad science" projects made me feel excited and welcome to contribute, and that's part of why I ended up working on this project.

I'm also able to work on this because I'm taking time off from work at the moment to attend the Recurse Center (https://www.recurse.com) in New York City. This is the place that used to be called "Hacker School" and it changed its name (https://www.recurse.com/blog/77-hacker-school-is-now-the-recurse-center) recently; the first reason for the name change was that they wanted to get away from the connotations of a school where people are taught things, when it's really more like a retreat for programmers to improve their programming through project work for three months, and I'm very thankful to them for allowing me to attend.

(http://www.flickr.com/photos/ashesoftimes/8079234402)

Chris Ball (https://blog.printf.net/)

(HTTPS://BLOG.PRINTF.NET/)

External links:

photographs (https://plus.google.com/u/0/photos/112938759017605010116/albums/5637573932501306529)

on github (http://github.com/cjb)

on google+ (http://gplus.to/chrisball)

on linkedin (http://www.linkedin.com/in/chrisjball)

on twitter (http://twitter.com/cjbprime)

on youtube (http://youtube.com/user/cjbprime/videos)

Most popular tags:

ethics (/articles/tag/ethics)
gnome (/articles/tag/gnome)
guitar (/articles/tag/guitar)
kernel (/articles/tag/kernel)
linux (/articles/tag/linux)
olpc (/articles/tag/olpc)
photography (/articles/tag/photography)

WP-SVBTLE (HTTPS://GITHUB.COM/GRAVITYONMARS/WP-SVBTLE)

The second reason they decided to change their name because their international attendees kept showing up at the US border and saying “I’m here for Hacker School!” and.. they didn’t have a good time.

Finally, I’d like to end with a few more words about why I think this type of work is interesting and important. There’s a certain grand, global scale of project, let’s pick GitHub and Wikipedia as exemplars, where the only way to have the project be able to exist at global scale after it becomes popular is to raise tens of millions of dollars a year, as GitHub and Wikipedia have, to spend running it, hoarding disks and bandwidth in big data centers. That limits the kind of projects we can create and imagine at that scale to those that we can make a business plan for raising tens of millions of dollars a year to run. I hope that having decentralized and peer to peer algorithms allows us to think about creating ambitious software that doesn’t require that level of investment, and just instead requires its users to cooperate and share with each other.

Thank you all very much for listening.

(You can check out GitTorrent on GitHub (http://github.com/cjb/gittorrent), and discuss it on Hacker News (https://news.ycombinator.com/item?id=9625840). You could also follow me on Twitter (https://twitter.com/cjbprime).)

Comments

Julien P *said* on May 29, 2015 at 7:45 pm (https://blog.printf.net/articles/2015/05/29/announcing-gittorrent-a-decentralized-github/#comment-257511):

Very interesting idea.
I’ll follow the project on github for now and try it.

Reply ↓ (https://blog.printf.net/articles/2015/05/29/announcing-gittorrent-a-decentralized-github/?replytocom=257511#respond)

julien *said* on May 29, 2015 at 7:47 pm (https://blog.printf.net/articles/2015/05/29/announcing-gittorrent-a-decentralized-github/#comment-257512):

I love your idea.

I have question about security, is it safe when I got a commit from another user?

Reply ↓ (https://blog.printf.net/articles/2015/05/29/announcing-gittorrent-a-decentralized-github/?replytocom=257512#respond)

cjb *said* on May 29, 2015 at 7:49 pm (https://blog.printf.net/articles/2015/05/29/announcing-gittorrent-a-decentralized-github/#comment-257514):

It seems safe to me. You (by which I mean the GitTorrent software) just uncompress it and see if it matches the hash you know you’re supposed to end up at. If so, you got what you wanted. If not, discard it immediately. It seems to me that the only problem could be if there’s an exploit in git’s pack uncompressor, which would be a huge problem for anything using git, not just this.

Reply ↓ (https://blog.printf.net/articles/2015/05/29/announcing-gittorrent-a-decentralized-github/?replytocom=257514#respond)

Ivan *said* on May 30, 2015 at 10:11 am (https://blog.printf.net/articles/2015/05/29/announcing-gittorrent-a-decentralized-github/#comment-257818):

Git uses sha1 hash which is considered insecure by today’s standards.

(http://www.flickr.com/photos/ashesoftimes/8079234402)

Chris Ball (https://blog.printf.net/)

(HTTPS://BLOG.PRINTF.NET/)

External links:

photographs (https://plus.google.com/u/0/photos/112938759017605010116/albums/5637573932501306529)

on github (http://github.com/cjb)

on google+ (http://gplus.to/chrisball)

on linkedin (http://www.linkedin.com/in/chrisjball)

on twitter (http://twitter.com/cjbprime)

on youtube (http://youtube.com/user/cjbprime/videos)

Most popular tags:

ethics (/articles/tag/ethics)

gnome (/articles/tag/gnome)

guitar (/articles/tag/guitar)

kernel (/articles/tag/kernel)

linux (/articles/tag/linux)

olpc (/articles/tag/olpc)

photography (/articles/tag/photography)

WP-SVBTLE (HTTPS://GITHUB.COM/GRAVITYONMARS/WP-SVBTLE)

If I understand correctly, with sha1's weak collision resistance, it would be relatively cheap to create alternate histories and contents of commits.

Reply ↓ (https://blog.printf.net/articles/2015/05/29/announcing-gittorrent-a-decentralized-github/?replytocom=257818#respond)

cjb *said* on May 30, 2015 at 10:40 am (https://blog.printf.net/articles/2015/05/29/announcing-gittorrent-a-decentralized-github/#comment-257831):

No, you're overstating how broken sha1 is. Finding collisions for a specific hash is not yet any kind of cheap.

Reply ↓ (https://blog.printf.net/articles/2015/05/29/announcing-gittorrent-a-decentralized-github/?replytocom=257831#respond)

Ivan *said* on May 30, 2015 at 10:54 am (https://blog.printf.net/articles/2015/05/29/an gittorrent-a-decentralized-github/#comment-257837):

While that is true, creating a pair of hashes and sending one in a pull request shouldn't be.

Reply ↓ (https://blog.printf.net/articles/2015/05/29/an gittorrent-a-decentralized-github/?replytocom=257837#respond)

Nick *said* on May 30, 2015 at 8:22 pm (https://blog.printf.net/articles/2015/05/29/announcing-gittorrent-a-decentralized-github/#comment-258029):

Nobody has ever found a single SHA-1 collision. So it would not be 'relatively cheap'.

Reply ↓ (https://blog.printf.net/articles/2015/05/29/announcing-gittorrent-a-decentralized-github/?replytocom=258029#respond)

Ivan *said* on June 10, 2015 at 1:54 pm (https://blog.printf.net/articles/2015/05/29/an gittorrent-a-decentralized-github/#comment-263206):

I stand corrected.

Shneier's estimate was $2.77M in 2012. It may have been a low one.

Nevertheless, sha-1 in git doesn't seem significantly more secure than in certificates where it is being phased out.

(http://www.flickr.com/photos/ashesoftimes/8079234402)

Chris Ball (https://blog.printf.net/)

(HTTPS://BLOG.PRINTF.NET/)

External links:

photographs (https://plus.google.com/u/0/photos/112938759017605010116/albums/5637573932501306529)

on github (http://github.com/cjb)

on google+ (http://gplus.to/chrisball)

on linkedin (http://www.linkedin.com/in/chrisjball)

on twitter (http://twitter.com/cjbprime)

on youtube (http://youtube.com/user/cjbprime/videos)

Most popular tags:

ethics (/articles/tag/ethics)
gnome (/articles/tag/gnome)
guitar (/articles/tag/guitar)
kernel (/articles/tag/kernel)
linux (/articles/tag/linux)
olpc (/articles/tag/olpc)
photography (/articles/tag/photography)

WP-SVBTLE (HTTPS://GITHUB.COM/GRAVITYONMARS/WP-SVBTLE)

Reply ↓ (https://blog.printf.net/articles/2015/05/29/an gittorrent-a-decentralized-github/?replytocom=263206#respond)

cjb *said* on June 10, 2015 at 2:52 pm (https://blog.printf.net/artic gittorrent-a-decentralized-github/#comment-263227):

Schneier's cost estimate is for creating *random* collisions, but you said it would be "relatively cheap to create alternate histories and contents of commits", which involves creating *specific* sha1 collisions, which has never happened before and has no known cost estimate.

It's a good idea to move away from sha1 where we can. But it's not because there's a real vulnerability to git's use of sha1.

Ivan *said* on June 10, 2015 at 3:42 pm (https://blog.printf.net/artic gittorrent-a-decentralized-github/#comment-263246):

@cjb:

There are no publicly known collisions of any kind, i.e. s1, s2, such that s1 ≠ s2, sha1(s1) = sha1(s2).

(http://www.flickr.com/photos/ashesoftimes/8079234402)

Chris Ball (https://blog.printf.net/)

(HTTPS://BLOG.PRINTF.NET/)

External links:

photographs (https://plus.google.com/u/0/photos/112938759017605010116/albums/5637573932501306529)

on github (http://github.com/cjb)

on google+ (http://gplus.to/chrisball)

on linkedin (http://www.linkedin.com/in/chrisjball)

on twitter (http://twitter.com/cjbprime)

on youtube (http://youtube.com/user/cjbprime/videos)

Most popular tags:

ethics (/articles/tag/ethics)

gnome (/articles/tag/gnome)

guitar (/articles/tag/guitar)

kernel (/articles/tag/kernel)

linux (/articles/tag/linux)

olpc (/articles/tag/olpc)

photography (/articles/tag/photography)

WP-SVBTLE (HTTPS://GITHUB.COM/GRAVITYONMARS/WP-SVBTLE)

I'm not sure I follow what you mean by a *specific* collision.

raphael *said* on July 7, 2015 at 6:01 pm (https://blog.printf.net/articl gittorrent-a-decentralized-github/#comment-274091):

@Ivan:

You're mistaking collision attacks with pre-image atacks. The latter would pose a threat to git if they were possible, but not the first.

Collision attacks are what Chris is calling random collisions: you just find two random inputs that result on the same hash, whatever these inputs are — any two random strings of bytes. Pre-image attacks on the other hand happen when you have a certain input with its associated hash (e.g. a git commit) and you need to find a different input that will result on this same hash. This is way harder than a collision, and we don't even have a theoretical attack of this kind on SHA1. Not only that but for this to be considered a threat to git, you need to be able to find your

(http://www.flickr.com/photos/ashesoftimes/8079234402)

Chris Ball (https://blog.printf.net/)

(HTTPS://BLOG.PRINTF.NET/)

External links:

photographs (https://plus.google.com/u/0/photos/112938759017605010116/albums/5637573932501306529)

on github (http://github.com/cjb)

on google+ (http://gplus.to/chrisball)

on linkedin (http://www.linkedin.com/in/chrisjball)

on twitter (http://twitter.com/cjbprime)

on youtube (http://youtube.com/user/cjbprime/videos)

Most popular tags:

ethics (/articles/tag/ethics)

gnome (/articles/tag/gnome)

guitar (/articles/tag/guitar)

kernel (/articles/tag/kernel)

linux (/articles/tag/linux)

olpc (/articles/tag/olpc)

photography (/articles/tag/photography)

WP-SVBTLE (HTTPS://GITHUB.COM/GRAVITYONMARS/WP-SVBTLE)

second input within the set of valid git commits, which is waaay smaller than the set of all possible random bytestrings. Not happening anytime soon.

---

lesto *said* on June 5, 2015 at 11:26 pm (https://blog.printf.net/articles/2015/05/29/announcing-gittorrent-a-decentralized-github/#comment-260974):

problem is “uncompressing”.
AFAIK there is no way to prevent a “decompress bomb”

Reply ↓ (https://blog.printf.net/articles/2015/05/29/announcing-gittorrent-a-decentralized-github/?replytocom=260974#respond)

cjb *said* on June 5, 2015 at 11:35 pm (https://blog.printf.net/articles/2015/05/29/announcing-gittorrent-a-decentralized-github/#comment-260978):

> AFAIK there is no way to prevent a “decompress bomb”

Worst case, the person publishing the Git sha1 (e.g. on their mutable key) can also publish what uncompressed object file size you should end up at; if you go past that, terminate decompression. That gives some mild protection against hash collisions attacks too. 🙂

Reply ↓ (https://blog.printf.net/articles/2015/05/29/announcing-gittorrent-a-decentralized-github/?replytocom=260978#respond)

---

sha *said* on May 30, 2015 at 7:52 am (https://blog.printf.net/articles/2015/05/29/announcing-gittorrent-a-decentralized-github/#comment-257771):

If you have some trusted way of knowing that the commit with hash 3a987487d1098a42ef1 is the one you want, and you clone a repo and the tip does in fact have the commit hash 3a987487d1098a42ef1, then you’re good.

So what you need is a trusted way of knowing that. If that hash is signed with a public key like explained above, and you trust that public key, then you have a trusted way of knowing it.

Reply ↓ (https://blog.printf.net/articles/2015/05/29/announcing-gittorrent-a-decentralized-github/?replytocom=257771#respond)

(http://www.flickr.com/photos/ashesoftimes/8079234402)

Chris Ball (https://blog.printf.net/)

(HTTPS://BLOG.PRINTF.NET/)

External links:

photographs (https://plus.google.com/u/0/photos/112938759017605010116/albums/5637573932501306529)

on github (http://github.com/cjb)

on google+ (http://gplus.to/chrisball)

on linkedin (http://www.linkedin.com/in/chrisjball)

on twitter (http://twitter.com/cjbprime)

on youtube (http://youtube.com/user/cjbprime/videos)

Most popular tags:

ethics (/articles/tag/ethics)
gnome (/articles/tag/gnome)
guitar (/articles/tag/guitar)
kernel (/articles/tag/kernel)
linux (/articles/tag/linux)
olpc (/articles/tag/olpc)
photography (/articles/tag/photography)

WP-SVBTLE (HTTPS://GITHUB.COM/GRAVITYONMARS/WP-SVBTLE)

Jonathan Baldwin *said* on May 29, 2015 at 7:49 pm (https://blog.printf.net/articles/2015/05/29/announcing-gittorrent-a-decentralized-github/#comment-257513):

I've been thinking about such a project for a while. But I have one major criticism of the way you plan to roll this out:

Use of OP_RETURN is deprecated – the Bitcoin community frowns upon using it's network for storing metadata, and OP_RETURN exists solely as damage control for the metadata stored on it in more mischevious and damaging ways. Have you considered using another cryptocurrency, such as Namecoin, to do what you want? Namecoin is specifically designed for applications like this.

https://bitcoin.org/en/release/v0.9.0#opreturn-and-data-in-the-block-chain (https://bitcoin.org/en/release/v0.9.0#opreturn-and-data-in-the-block-chain)

https://namecoin.info/ (https://namecoin.info/)

Reply ↓ (https://blog.printf.net/articles/2015/05/29/announcing-gittorrent-a-decentralized-github/?replytocom=257513#respond)

cjb *said* on May 29, 2015 at 7:52 pm (https://blog.printf.net/articles/2015/05/29/announcing-gittorrent-a-decentralized-github/#comment-257517):

Thanks. It's a shame about OP_RETURN. I wrote a reply on HN which I'll copy here:

I have a mild bias against altcoins, and have heard bad things about Namecoin in particular: that the anti-spam incentives aren't good, leading to illegal files stored in the blockchain itself, and that there's no compact representation (like Bitcoin's Simplified Payment Verification) for determining whether a claimed name is valid without consulting a full history.

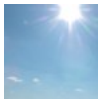
As I understand it, these two design flaws combine to mean that you have to store some very illegal files to use a namecoin resolver, which doesn't sound good to me. (I may be mistaken, since the bad things I heard about Namecoin came from Bitcoin people..)

Reply ↓ (https://blog.printf.net/articles/2015/05/29/announcing-gittorrent-a-decentralized-github/?replytocom=257517#respond)

Jonathan Baldwin *said* on May 29, 2015 at 8:01 pm (https://blog.printf.net/articles/2015/05/29/announcing-gittorrent-a-decentralized-github/#comment-257523):

Carry on then, that sounds like a good enough reason to keep using OP_RETURN. Ultimately, the metadata problem is something that the Bitcoin people are going to have to deal with sooner or later, OP_RETURN just postpones it for them.

Reply ↓ (https://blog.printf.net/articles/2015/05/29/announcing-gittorrent-a-decentralized-github/?replytocom=257523#respond)



Dāvis *said* on May 29, 2015 at 8:17 pm (https://blog.printf.net/articles/2015/05/29/announcing-gittorrent-a-decentralized-github/#comment-257532):

Do you realize that it's exactly same for Bitcoin's blockchain right? See http://www.righto.com/2014/02/ascii-bernanke-wikileaks-photographs.html (http://www.righto.com/2014/02/ascii-bernanke-wikileaks-photographs.html)

Namecoin was forked from same Bitcoin codebase and if they would have manpower they could rebase it on top of latest Bitcoin Core. I think Namecoin is the right way to go for

(http://www.flickr.com/photos/ashesoftimes/8079234402)

Chris Ball (https://blog.printf.net/)

(HTTPS://BLOG.PRINTF.NET/)

External links:

photographs (https://plus.google.com/u/0/photos/112938759017605010116/albums/5637573932501306529)

on github (http://github.com/cjb)

on google+ (http://gplus.to/chrisball)

on linkedin (http://www.linkedin.com/in/chrisjball)

on twitter (http://twitter.com/cjbprime)

on youtube (http://youtube.com/user/cjbprime/videos)

Most popular tags:

ethics (/articles/tag/ethics)
gnome (/articles/tag/gnome)
guitar (/articles/tag/guitar)
kernel (/articles/tag/kernel)
linux (/articles/tag/linux)
olpc (/articles/tag/olpc)
photography (/articles/tag/photography)

WP-SVBTLE (HTTPS://GITHUB.COM/GRAVITYONMARS/WP-SVBTLE)

decentralized identities.

Reply ↓ (https://blog.printf.net/articles/2015/05/29/announcing-gittorrent-a-decentralized-github/?replytocom=257532#respond)

cjb *said* on May 29, 2015 at 8:22 pm (https://blog.printf.net/articles/2015/05/29/announcing-gittorrent-a-decentralized-github/#comment-257534):

I wrote about this on HN. The theoretic capability is the same, but the cost incentives are different. Storing a 4MB image at 80 bytes per $0.08 OP_RETURN transaction would cost you $4000 on Bitcoin's network, so no-one would actually do it.

Reply ↓ (https://blog.printf.net/articles/2015/05/29/announcing-gittorrent-a-decentralized-github/?replytocom=257534#respond)

Ron *said* on May 29, 2015 at 9:14 pm (https://blog.printf.net/articles/2015/05/29/an gittorrent-a-decentralized-github/#comment-257562):

Again, awesome idea, just beware of the people deceitfully "warning" you against using Bitcoin. They're just financially vested in Namecoin, and are essentially trying to pump their stock portfolio.

Reply ↓ (https://blog.printf.net/articles/2015/05/29/an gittorrent-a-decentralized-github/?replytocom=257562#respond)

Sok Puppette *said* on May 30, 2015 at 10:31 am (https://blog.printf.net/articles/2015/05/29/an gittorrent-a-decentralized-github/#comment-257826):

"No-one would actually do it" is a pretty strong claim.

Remember, it only takes ONE person to do it. Or one organization. Are you going to claim that there is not, and never will be, anybody in the whole world who would invest $4000 to cloud the legal status of running a full Bitcoin node? No person, no corporation, no government clandestine service? And a perfectly clear image can be well under 100KB, not 4MB. So that's $100. There are trolls who would spend that for fun.

(http://www.flickr.com/photos/ashesoftimes/8079234402)

Chris Ball (https://blog.printf.net/)

(HTTPS://BLOG.PRINTF.NET/)

External links:

photographs (https://plus.google.com/u/0/photos/112938759017605010116/albums/5637573932501306529)

on github (http://github.com/cjb)

on google+ (http://gplus.to/chrisball)

on linkedin (http://www.linkedin.com/in/chrisjball)

on twitter (http://twitter.com/cjbprime)

on youtube (http://youtube.com/user/cjbprime/videos)

Most popular tags:

ethics (/articles/tag/ethics)

gnome (/articles/tag/gnome)

guitar (/articles/tag/guitar)

kernel (/articles/tag/kernel)

linux (/articles/tag/linux)

olpc (/articles/tag/olpc)

photography (/articles/tag/photography)

WP-SVBTLE (HTTPS://GITHUB.COM/GRAVITYONMARS/WP-SVBTLE)

gittorrent is a GREAT IDEA, and a big contribution, by the way. Thank you.

Reply ↓ (https://blog.printf.net/articles/2015/05/29/announcing-gittorrent-a-decentralized-github/?replytocom=257826#respond)

Ron *said* on May 29, 2015 at 9:10 pm (https://blog.printf.net/articles/2015/05/29/announcing-gittorrent-a-decentralized-github/#comment-257560):

Bitcoin is the ultimate distributed database first, it's a currency second. Don't pay the blockchain bloat nazis any mind. As long as you're paying BTC fees, you can do whatever you want to the Bitcoin blockchain. @cjb has a killer idea, decentralized GitHub – if your first reaction is to tell him OP_RETURN is "deprecated", which is just utter malarky, you should be ashamed!

Reply ↓ (https://blog.printf.net/articles/2015/05/29/announcing-gittorrent-a-decentralized-github/?replytocom=257560#respond)

Amin *said* on May 30, 2015 at 4:14 pm (https://blog.printf.net/articles/2015/05/29/announcing-gittorrent-a-decentralized-github/#comment-257949):

Maybe some Bitcoin developers are deprecating it, but the fact that it's recently been increased from 40 bytes to 80 bytes is a tacit endorsement of its use.

Reply ↓ (https://blog.printf.net/articles/2015/05/29/announcing-gittorrent-a-decentralized-github/?replytocom=257949#respond)

necrophcodr (http://necrophcodr.me) *said* on May 29, 2015 at 7:58 pm (https://blog.printf.net/articles/2015/05/29/announcing-gittorrent-a-decentralized-github/#comment-257521):

This is a really cool idea. Is there any chance you could do a writeup, or checkup, about implementation of this concept into the Fossil DVCS (http://fossil-scm.org/ (http://fossil-scm.org/)) ?
It sounds like that would be the ideal DVCS to have this feature, considering that the entire documentation as well as any current issues and much more would be cloned, and fully available offline for anyone to view. Potentially, anyway.

Reply ↓ (https://blog.printf.net/articles/2015/05/29/announcing-gittorrent-a-decentralized-github/?replytocom=257521#respond)

A. F. Dudley *said* on May 29, 2015 at 8:37 pm (https://blog.printf.net/articles/2015/05/29/announcing-gittorrent-a-decentralized-github/#comment-257543):

Have you looked at IPFS?
http://ipfs.io/ (http://ipfs.io/)
https://github.com/ipfs/go-ipfs (https://github.com/ipfs/go-ipfs)

Reply ↓ (https://blog.printf.net/articles/2015/05/29/announcing-gittorrent-a-decentralized-github/?replytocom=257543#respond)

(http://www.flickr.com/photos/ashesoftimes/8079234402)

Chris Ball (https://blog.printf.net/)

(HTTPS://BLOG.PRINTF.NET/)

External links:

photographs (https://plus.google.com/u/0/photos/112938759017605010116/albums/5637573932501306529)

on github (http://github.com/cjb)

on google+ (http://gplus.to/chrisball)

on linkedin (http://www.linkedin.com/in/chrisjball)

on twitter (http://twitter.com/cjbprime)

on youtube (http://youtube.com/user/cjbprime/videos)

Most popular tags:

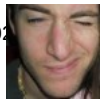
ethics (/articles/tag/ethics)
gnome (/articles/tag/gnome)
guitar (/articles/tag/guitar)
kernel (/articles/tag/kernel)
linux (/articles/tag/linux)
olpc (/articles/tag/olpc)
photography (/articles/tag/photography)

WP-SVBTLE (HTTPS://GITHUB.COM/GRAVITYONMARS/WP-SVBTLE)

cjb *said* on May 29, 2015 at 8:38 pm (https://blog.printf.net/articles/2015/05/29/announcing-gittorrent-a-decentralized-github/#comment-257544):

I like IPFS, but I'm not sure how to build the part where packfiles are negotiated on top of it. I'll try to figure it out in the future.

Reply ↓ (https://blog.printf.net/articles/2015/05/29/announcing-gittorrent-a-decentralized-github/?replytocom=257544#respond)



sam *said* on May 29, 2015 at 8:46 pm (https://blog.printf.net/articles/2015/05/29/announcing-gittorrent-a-decentralized-github/#comment-257549):

Love the idea! One question though, how do we validate the data being seeded? I not familiar with bit-torrent, so this might already be solved, just afraid of someone potentially injecting dangerous code into the distributed stream. Maybe a centralized server could be used to house checksum data so that the can project downloaded can hopefully be verified and validated. Even if a centralized server is still needed for that purpose, it is definitely a step in the right direction. I still imagine a centralized server would be needed as a base seed for projects that do not have enough users.

Reply ↓ (https://blog.printf.net/articles/2015/05/29/announcing-gittorrent-a-decentralized-github/?

cjb *said* on May 29, 2015 at 8:48 pm (https://blog.printf.net/articles/2015/05/29/announcing-gittorrent-a-decentralized-github/#comment-257550):

You have to find out what hash you're aiming for first. In the mode where you're downloading a repo that's also on GitHub, we ask GitHub. In the mode where you're internal to the network, it uses the mutable keys, so you should ensure that the person you want to download from has given you their real key.

Reply ↓ (https://blog.printf.net/articles/2015/05/29/announcing-gittorrent-a-decentralized-github/?replytocom=257550#respond)

Just John *said* on May 29, 2015 at 8:52 pm (https://blog.printf.net/articles/2015/05/29/announcing-gittorrent-a-decentralized-github/#comment-257551):

What happens when there is a *single* security issue, and all of the source code that resides on such a network is compromised. Or, is the intention such that *only* open source projects would exist on this? If the latter is the case, I fear you've spent a little too much time eating artisinal toast.

Reply ↓ (https://blog.printf.net/articles/2015/05/29/announcing-gittorrent-a-decentralized-github/?replytocom=257551#respond)

cjb *said* on May 29, 2015 at 8:58 pm (https://blog.printf.net/articles/2015/05/29/announcing-gittorrent-a-decentralized-github/#comment-257552):

You could ask exactly the same question of GitHub, no?

(I haven't thought about whether this system could support closed source repos too. It's not my main use case.)

Reply ↓ (https://blog.printf.net/articles/2015/05/29/announcing-gittorrent-a-decentralized-github/?replytocom=257552#respond)

isovector *said* on June 25, 2015 at 5:58 pm (https://blog.printf.net/articles/2015/05/29/announcing-gittorrent-a-decentralized-github/#comment-270369):

(http://www.flickr.com/photos/ashesoftimes/8079234402)

Chris Ball (https://blog.printf.net/)

(HTTPS://BLOG.PRINTF.NET/)

External links:

photog (https://plus.google.com/u/0/photos/112938759017605010116/albums/5637573932501306529)

on github (http://github.com/cjb)

on google+ (http://gplus.to/chrisball)

on linkedin (http://www.linkedin.com/in/chrisjball)

on twitter (http://twitter.com/cjbprime)

on youtube (http://youtube.com/user/cjbprime/videos)

Most popular tags:

ethics (/articles/tag/ethics)

gnome (/articles/tag/gnome)

guitar (/articles/tag/guitar)

kernel (/articles/tag/kernel)

linux (/articles/tag/linux)

olpc (/articles/tag/olpc)

photography (/articles/tag/photography)

WP-SVBTLE (HTTPS://GITHUB.COM/GRAVITYONMARS/WP-SVBTLE)

Presumably you could just have a silent encryption layer running before you commit the git objects?

Reply ↓ (https://blog.printf.net/articles/2015/05/29/announcing-gittorrent-a-decentralized-github/?replytocom=270369#respond)

---

clacke (https://microca.st/clacke) *said* on March 14, 2016 at 12:14 pm (https://blog.printf.net/articles/2015/05/29/announcing-gittorrent-a-decentralized-github/#comment-324247):

Check out https://github.com/joeyh/git-remote-gcrypt/ (https://github.com/joeyh/git-remote-gcrypt/), perhaps? Take the encrypted git repo and distribute that over gittorrent.

Reply ↓ (https://blog.printf.net/articles/2015/05/29/announcing-gittorrent-a-decentralized-github/?replytocom=324247#respond)

Peter TB Brett *said* on May 29, 2015 at 9:59 pm (https://blog.printf.net/articles/2015/05/29/announcing-gittorrent-a-decentralized-github/#comment-257580):

How does GitTorrent cope with commit hash collisions? As the number of projects using git increases, the probability of two commits having the same SHA-1 approaches unity.

Reply ↓ (https://blog.printf.net/articles/2015/05/29/announcing-gittorrent-a-decentralized-github/?replytocom=257580#respond)

cjb *said* on May 30, 2015 at 2:45 am (https://blog.printf.net/articles/2015/05/29/announcing-gittorrent-a-decentralized-github/#comment-257672):

It would have to be two published refs colliding (i.e. two different repos with the same sha1 for master/HEAD at the same time) to be a collision on the DHT; I'm not worried about it.

Reply ↓ (https://blog.printf.net/articles/2015/05/29/announcing-gittorrent-a-decentralized-github/?replytocom=257672#respond)

Jesse *said* on May 11, 2016 at 8:31 pm (https://blog.printf.net/articles/2015/05/29/announcing-gittorrent-a-decentralized-github/#comment-328642):

While technically true, the probability approaches unity *very* slowly. For SHA-1, the probability of a *random* collision between any two 160-bit hashes would remain less than 50% over a period of 100 years during which the entire population of the planet (~9 billion) generated new commits at an average rate of one commit per person per second. Practically speaking, the odds are much higher that a random electronic glitch will cause the wrong data to be returned than that there would be a true random collision between commit hashes.

(Needless to say, this does not take into account non-random collisions resulting from attacks against the SHA-1 algorithm. This is just a straightforward application of the Birthday Paradox formula.)

Reply ↓ (https://blog.printf.net/articles/2015/05/29/announcing-gittorrent-a-decentralized-github/?replytocom=328642#respond)

Duy Nguyen *said* on May 30, 2015 at 12:11 am (https://blog.printf.net/articles/2015/05/29/announcing-gittorrent-a-decentralized-github/#comment-257615):

(http://www.flickr.com/photos/ashesoftimes/8079234402)

Chris Ball (https://blog.printf.net/)

(HTTPS://BLOG.PRINTF.NET/)

External links:

photographs (https://plus.google.com/u/0/photos/112938759017605010116/albums/5637573932501306529)

on github (http://github.com/cjb)

on google+ (http://gplus.to/chrisball)

on linkedin (http://www.linkedin.com/in/chrisjball)

on twitter (http://twitter.com/cjbprime)

on youtube (http://youtube.com/user/cjbprime/videos)

Most popular tags:

ethics (/articles/tag/ethics)

gnome (/articles/tag/gnome)

guitar (/articles/tag/guitar)

kernel (/articles/tag/kernel)

linux (/articles/tag/linux)

olpc (/articles/tag/olpc)

photography (/articles/tag/photography)

WP-SVBTLE (HTTPS://GITHUB.COM/GRAVITYONMARS/WP-SVBTLE)

I'm not sure about this

"Then the remote node makes us a packfile and tells us the hash of that packfile, and then we start downloading that packfile from it _and any other nodes who are seeding it using Standard BitTorrent._"

packfile generation is unstable (by design) Even if you give git-pack-objects the same input, it may generate different files. How come other nodes seed the same packfile?

Reply ↓ (https://blog.printf.net/articles/2015/05/29/announcing-gittorrent-a-decentralized-github/?replytocom=257615#respond)

cjb *said* on May 30, 2015 at 12:17 am (https://blog.printf.net/articles/2015/05/29/announcing-gittorrent-a-decentralized-github/#comment-257620):

Oh! That's interesting. Do you have a reference for packfiles being unstable? They don't seem so in practice.

Reply ↓ (https://blog.printf.net/articles/2015/05/29/announcing-gittorrent-a-decentralized-github/?replytocom=257620#respond)

Duy Nguyen *said* on May 30, 2015 at 3:35 am (https://blog.printf.net/articles/2015/05/29/announcing-gittorrent-a-decentralized-github/#comment-257691):

http://article.gmane.org/gmane.comp.version-control.git/164643 (http://article.gmane.org/gmane.comp.version-control.git/164643)

Reply ↓ (https://blog.printf.net/articles/2015/05/29/announcing-gittorrent-a-decentralized-github/?replytocom=257691#respond)

cjb *said* on May 30, 2015 at 10:35 am (https://blog.printf.net/articles/2015/05/29/announcing-gittorrent-a-decentralized-github/#comment-257827):

Thanks! I should probably switch to using an alternate method of packfile generation that is guaranteed deterministic.

Reply ↓ (https://blog.printf.net/articles/2015/05/29/announcing-gittorrent-a-decentralized-github/?replytocom=257827#respond)

Guest *said* on June 1, 2015 at 7:40 pm (https://blog.printf.net/articles/2015/05/29/announcing-gittorrent-a-decentralized-github/#comment-258853):

Even with deterministic packfiles, you're still hoping that lots of people not only are interested in commit aaaa, but also want(ed) to update there from commit bbbb. That sounds like reducing shareability by quite a lot. Assume a repo that creates a new commit each hour, and 24 clients running "git fetch" in a cronjob, each on a different hour. There'd be no sharing of bandwidth at all, each client wants a completely different packfile.

Reply ↓ (https://blog.printf.net/articles/2015/05/29/announcing-gittorrent-a-decentralized-github/?replytocom=258853#respond)

cjb *said* on June 1, 2015 at 8:41 pm (https://blog.printf.net/articles/2015/05/29/announcing-gittorrent-a-decentralized-github/#comment-258879):

(http://www.flickr.com/photos/ashesoftimes/8079234402)

Chris Ball (https://blog.printf.net/)

(HTTPS://BLOG.PRINTF.NET/)

External links:

photographs (https://plus.google.com/u/0/photos/112938759017605010116/albums/5637573932501306529)

on github (http://github.com/cjb)

on google+ (http://gplus.to/chrisball)

on linkedin (http://www.linkedin.com/in/chrisjball)

on twitter (http://twitter.com/cjbprime)

on youtube (http://youtube.com/user/cjbprime/videos)

Most popular tags:

ethics (/articles/tag/ethics)
gnome (/articles/tag/gnome)
guitar (/articles/tag/guitar)
kernel (/articles/tag/kernel)
linux (/articles/tag/linux)
olpc (/articles/tag/olpc)
photography (/articles/tag/photography)

WP-SVBTLE (HTTPS://GITHUB.COM/GRAVITYONMARS/WP-SVBTLE)

You're right. With this design, swarming downloads become a "nice to have" performance optimization for repos that are very popular or not updated all the time, rather than something integral.

We probably need to move away from BitTorrent to do better (which I'd be willing to do if it's worth it). IPFS hosting the Git DAG might work?

Reply ↓ (https://blog.printf.net/articles/2015/05/29/announcing-gittorrent-a-decentralized-github/?replytocom=258879#respond)

BillDStrong (http://ndxtreme.com) *said* on June 13, 2015 at 8:19 am (https://blog.printf.net/articles/2015/05/29/announcing-gittorrent-a-decentralized-github/#comment-264531):

IPFS has a sister project called filecoin that deals with data that is not popular. It allows interested users to pay to have data kept. I haven't looked into it too much, but might be an interesting benefit if it pans out.

Reply ↓ (https://blog.printf.net/articles/2015/05/29/announcing-gittorrent-a-decentralized-github/?replytocom=264531#respond)

Web (http://www.fussball-tipps.org) *said* on May 30, 2015 at 3:20 am (https://blog.printf.net/articles/2015/05/29/announcing-gittorrent-a-decentralized-github/#comment-257685):

This is fascinating work and a very sound idea; I like the idea of my personal device/server being an active distribution node not just contributor to the open source projects I contribute and support.

Reply ↓ (https://blog.printf.net/articles/2015/05/29/announcing-gittorrent-a-decentralized-github/?replytocom=257685#respond)

Bora M. Alper (https://boramalper.org) *said* on May 30, 2015 at 5:32 am (https://blog.printf.net/articles/2015/05/29/announcing-gittorrent-a-decentralized-github/#comment-257733):

Are you the author of GitTorrent on Google Code (https://code.google.com/p/gittorrent/ (https://code.google.com/p/gittorrent/))?

Reply ↓ (https://blog.printf.net/articles/2015/05/29/announcing-gittorrent-a-decentralized-github/?replytocom=257733#respond)

cjb *said* on May 30, 2015 at 10:36 am (https://blog.printf.net/articles/2015/05/29/announcing-gittorrent-a-decentralized-github/#comment-257828):

Nope! But I talked to both of them, and they're happy for me to use the project name. (The Google Code GitTorrent stalled out over five years ago; they moved on to MirrorSync.)

Reply ↓ (https://blog.printf.net/articles/2015/05/29/announcing-gittorrent-a-decentralized-github/?replytocom=257828#respond)

(http://www.flickr.com/photos/ashesoftimes/8079234402)

Chris Ball (https://blog.printf.net/)

(HTTPS://BLOG.PRINTF.NET/)

External links:

photographs (https://plus.google.com/u/0/photos/112938759017605010116/a... 6375739325013065 29)

on github (http://github.com/cjb)

on google+ (http://gplus.to/chrisball)

on linkedin (http://www.linkedin.com/in/chrisjball)

on twitter (http://twitter.com/cjbprime)

on youtube (http://youtube.com/user/cjbprime/videos)

Most popular tags:

ethics (/articles/tag/ethics)
gnome (/articles/tag/gnome)
guitar (/articles/tag/guitar)
kernel (/articles/tag/kernel)
linux (/articles/tag/linux)
olpc (/articles/tag/olpc)
photography (/articles/tag/photography)

WP-SVBTLE (HTTPS://GITHUB.COM/GRAVITYONMARS/WP-SVBTLE)

Ron *said* on May 30, 2015 at 7:59 am (https://blog.printf.net/articles/2015/05/29/announcing-gittorrent-a-decentralized-github/#comment-257776):

Great work. We need more innovators like you!

Reply ↓ (https://blog.printf.net/articles/2015/05/29/announcing-gittorrent-a-decentralized-github/?replytocom=257776#respond)

LowEel *said* on May 30, 2015 at 9:31 am (https://blog.printf.net/articles/2015/05/29/announcing-gittorrent-a-decentralized-github/#comment-257810):

I live the idea.

I only ask a question: hos is the serving side? It means you are always running a tracker, or a dht, or?

I mean, imagine tomorrow the police shots down some tracker because of copyright infringement. If this is the same tracker we use for distributing our code torrents, our code is lost or corrupted or impossible to retrieve then?

Reply ↓ (https://blog.printf.net/articles/2015/05/29/announcing-gittorrent-a-decentralized-github/?replytocom=257810#respond)

cjb *said* on May 30, 2015 at 10:37 am (https://blog.printf.net/articles/2015/05/29/announcing-gittorrent-a-decentralized-github/#comment-257829):

We don't use trackers, just the DHT. If the BitTorrent DHT was going to be shut down due to copyright infringment, it would have happened already. It's fine.

Reply ↓ (https://blog.printf.net/articles/2015/05/29/announcing-gittorrent-a-decentralized-github/?replytocom=257829#respond)

Sol *said* on May 30, 2015 at 10:54 am (https://blog.printf.net/articles/2015/05/29/announcing-gittorrent-a-decentralized-github/#comment-257836):

Don't you think that it is enough to use old good domains for distributed consensus on user names?

You put some file with your key and may be even more meta information about you at: https://my-domain.com/username (https://my-domain.com/username) and then you can use gittorent://my-domain.com/username/myrepo

Owner of the domain is the owner of the key!

Reply ↓ (https://blog.printf.net/articles/2015/05/29/announcing-gittorrent-a-decentralized-github/?replytocom=257836#respond)

cjb *said* on May 30, 2015 at 10:57 am (https://blog.printf.net/articles/2015/05/29/announcing-gittorrent-a-decentralized-github/#comment-257838):

Because I'm decentralizing GitHub, I wanted to have an answer for "github.com/someuser". I agree in general that a decentralized system should prefer something non-rivalrous like DNS or email addresses, so users don't have to go through the hassle of e.g. Bitcoin.

Reply ↓ (https://blog.printf.net/articles/2015/05/29/announcing-gittorrent-a-decentralized-github/?replytocom=257838#respond)

Gastlag *said* on June 1, 2015 at 9:15 am (https://blog.printf.net/articles/2015/05/29/announcing-gittorrent-a-decentralized-github/#comment-258637):

(http://www.flickr.com/photos/ashesoftimes/8079234402)

Chris Ball (https://blog.printf.net/)

(HTTPS://BLOG.PRINTF.NET/)

External links:

photographs (https://plus.google.com/u/0/photos/112938759017605010116/albums/5637573983250130659)

on github (http://github.com/cjb)

on google+ (http://gplus.to/chrisball)

on linkedin (http://www.linkedin.com/in/chrisjball)

on twitter (http://twitter.com/cjbprime)

on youtube (http://youtube.com/user/cjbprime/videos)

Most popular tags:

ethics (/articles/tag/ethics)
gnome (/articles/tag/gnome)
guitar (/articles/tag/guitar)
kernel (/articles/tag/kernel)
linux (/articles/tag/linux)
olpc (/articles/tag/olpc)
photography (/articles/tag/photography)

WP-SVBTLE (HTTPS://GITHUB.COM/GRAVITYONMARS/WP-SVBTLE)

Hello, Do you know GnuNet (https://gnunet.org/ (https://gnunet.org/)) and GnuName System (https://gnunet.org/gns (https://gnunet.org/gns)) ?

This comparaison could interess you : http://seenthis.net/messages/358071 (http://seenthis.net/messages/358071)

Reply ↓ (https://blog.printf.net/articles/2015/05/29/announcing-gittorrent-a-decentralized-github/?replytocom=258637#respond)

Pingback: *GitTorrent, un GitHub descentralizado - Detrás del pingüino (http://dplinux.net/gittorrent-un-github-descentralizado/)*

lp *said* on May 30, 2015 at 12:27 pm (https://blog.printf.net/articles/2015/05/29/announcing-gittorrent-a-decentralized-github/#comment-257867):

Regarding the name-resolve and your concerns about DNSchain and the deprecation of OP_RETURN in Bitcoin and also this line from your post:

>It surprised me that nothing like this seems to exist already in the decentralization community. I'd be happy to work on a project like this and make GitTorrent sit on top of it, so please let me know if you're interested in helping with that.

I would like to point you towards a project called dename:

– https://github.com/andres-erbsen/dename (https://github.com/andres-erbsen/dename)
– https://www.youtube.com/watch?v=-By4OnyC4Ig (https://www.youtube.com/watch?v=-By4OnyC4Ig) (2nd talk, 12mins in)

It might be exactly what you are looking for (:

Reply ↓ (https://blog.printf.net/articles/2015/05/29/announcing-gittorrent-a-decentralized-github/?replytocom=257867#respond)

cjb *said* on May 30, 2015 at 12:29 pm (https://blog.printf.net/articles/2015/05/29/announcing-gittorrent-a-decentralized-github/#comment-257869):

Thanks! Very interesting.

Reply ↓ (https://blog.printf.net/articles/2015/05/29/announcing-gittorrent-a-decentralized-github/?replytocom=257869#respond)

Andrés G. Aragoneses *said* on October 5, 2015 at 7:05 pm (https://blog.printf.net/articles/2015/05/29/announcing-gittorrent-a-decentralized-github/#comment-304893):

What are Chris' concerns about DNSChain? He didn't mention it in his blog post AFAICS.

Reply ↓ (https://blog.printf.net/articles/2015/05/29/announcing-gittorrent-a-decentralized-github/?replytocom=304893#respond)

CruelAngel *said* on May 30, 2015 at 1:25 pm (https://blog.printf.net/articles/2015/05/29/announcing-gittorrent-a-decentralized-github/#comment-257880):

Noobish question: What happens to projects that are not seeded anymore or not seeded as of yet? GitHub's validity for small projects is that you don't have to host your server for your repo, but if it works like bittorrent, then there has to be someone who seeds the repo, or noone can access it later.

(http://www.flickr.com/photos/ashesoftimes/8079234402)

Chris Ball (https://blog.printf.net/)

(HTTPS://BLOG.PRINTF.NET/)

External links:

photographs (https://plus.google.com/u/0/photos/112938759017605010116/albums/5637573932501306529)

on github (http://github.com/cjb)

on google+ (http://gplus.to/chrisball)

on linkedin (http://www.linkedin.com/in/chrisjball)

on twitter (http://twitter.com/cjbprime)

on youtube (http://youtube.com/user/cjbprime/videos)

Most popular tags:

ethics (/articles/tag/ethics)
gnome (/articles/tag/gnome)
guitar (/articles/tag/guitar)
kernel (/articles/tag/kernel)
linux (/articles/tag/linux)
olpc (/articles/tag/olpc)
photography (/articles/tag/photography)

WP-SVBTLE (HTTPS://GITHUB.COM/GRAVITYONMARS/WP-SVBTLE)

Reply ↓ (https://blog.printf.net/articles/2015/05/29/announcing-gittorrent-a-decentralized-github/?replytocom=257880#respond)

cjb *said* on May 30, 2015 at 1:44 pm (https://blog.printf.net/articles/2015/05/29/announcing-gittorrent-a-decentralized-github/#comment-257887):

It's a good question. We could set up some reciprocal hosting ("you seed my repos and I'll seed yours"), just have people donate spare space, encourage groups like the Internet Archive to help, or pay people to seed for you in the same way you can currently pay GitHub to store private repos for you.

Reply ↓ (https://blog.printf.net/articles/2015/05/29/announcing-gittorrent-a-decentralized-github/?replytocom=257887#respond)

JuanPablo (http://juanpabloaj.com/) *said* on May 30, 2015 at 1:59 pm (https://blog.printf.net/articles/2015/05/29/announcing-gittorrent-a-decentralized-github/#comment-257895):

Hi,
very interesting post and a great package, thanks a lot!
maybe over the package is posible build a decentralized content visualizer

Example: a decentralized wikipedia, every article is a git repo, if you would like read some article, the "visualizer" clone the article and you can read, and now you are sharing the article.

Reply ↓ (https://blog.printf.net/articles/2015/05/29/announcing-gittorrent-a-decentralized-github/?replytocom=257895#respond)

cjb *said* on May 30, 2015 at 2:01 pm (https://blog.printf.net/articles/2015/05/29/announcing-gittorrent-a-decentralized-github/#comment-257897):

Wikipedia's actually much harder to decentralize than GitHub, because resolving edit conflicts on Wikipedia is harder.

If you just wanted to browse Wikipedia p2p, though, someone's already done that 🙂 https://github.com/mafintosh/peerwiki (https://github.com/mafintosh/peerwiki)

Reply ↓ (https://blog.printf.net/articles/2015/05/29/announcing-gittorrent-a-decentralized-github/?replytocom=257897#respond)

tr4s *said* on May 30, 2015 at 2:16 pm (https://blog.printf.net/articles/2015/05/29/announcing-gittorrent-a-decentralized-github/#comment-257901):

How is the new head's hash propagated through the network? Is there any guarantee on how much time it's going to take before everyone get an update?

Reply ↓ (https://blog.printf.net/articles/2015/05/29/announcing-gittorrent-a-decentralized-github/?replytocom=257901#respond)

cjb *said* on May 30, 2015 at 2:32 pm (https://blog.printf.net/articles/2015/05/29/announcing-gittorrent-a-decentralized-github/#comment-257906):

It's announced to the Distributed Hash Table the same way that a normal BitTorrent DHT announcement of a new peer is, so you could read about that to learn more. I don't have numbers on time, but should be fast.

(http://www.flickr.com/photos/ashesoftimes/8079234402)

Chris Ball (https://blog.printf.net/)

(HTTPS://BLOG.PRINTF.NET/)

External links:

photographs (https://plus.google.com/u/0/photos/112938759017605010116/albums/5637573932501306529)

on github (http://github.com/cjb)

on google+ (http://gplus.to/chrisball)

on linkedin (http://www.linkedin.com/in/chrisjball)

on twitter (http://twitter.com/cjbprime)

on youtube (http://youtube.com/user/cjbprime/videos)

Most popular tags:

ethics (/articles/tag/ethics)

gnome (/articles/tag/gnome)

guitar (/articles/tag/guitar)

kernel (/articles/tag/kernel)

linux (/articles/tag/linux)

olpc (/articles/tag/olpc)

photography (/articles/tag/photography)

WP-SVBTLE (HTTPS://GITHUB.COM/GRAVITYONMARS/WP-SVBTLE)

Reply ↓ (https://blog.printf.net/articles/2015/05/29/announcing-gittorrent-a-decentralized-github/?replytocom=257906#respond)

Noam Kfir *said* on May 30, 2015 at 6:05 pm (https://blog.printf.net/articles/2015/05/29/announcing-gittorrent-a-decentralized-github/#comment-257992):

The idea is amazing and, I believe, will become increasingly necessary.

I think you should reconsider the reliance on a specific cryptocurrency's blockchain. You're interested in identity, not in currency and not in transaction history. Identity is much more complex.

For example, a write-only blockchain with an essentially "first come first serve" approach that works pretty well for currency is very often not the ideal for identity. Also, domains and email are fickle and transient. Oh, and simple things like a 20GB blockchain is a hurdle many people won't want to jump over.

Conflict resolution, including mutating identities, has to be built into the system to properly model the real world.

Reply ↓ (https://blog.printf.net/articles/2015/05/29/announcing-gittorrent-a-decentralized-github/?replytocom=257992#respond)

cjb *said* on May 30, 2015 at 6:31 pm (https://blog.printf.net/articles/2015/05/29/announcing-gittorrent-a-decentralized-github/#comment-258003):

It's not really reliant on Bitcoin. Bitcoin's providing a way to map from a username to a hash-of-pubkey. All that really matters is that you can get to that hash-of-pubkey somehow. I'll accept pull requests to support other methods, such as DNS records.

"First come first serve" is worse on other systems than this one. GitHub/Twitter/etc gives out usernames for free, but GitTorrent charges $0.08 (or $0.16 to avoid races).

> 20GB blockchain is a hurdle

You don't need to store the whole thing, just scan it once, so it's not quite so bad. If someone doesn't want to do that, I could publish a list of usernames alongside gittorrent — it's introducing trust, but anyone would be able to run the same scripts against the blockchain to verify that my list is the correct one, so it's not introducing any real centralization. It's just an optimization.

> mutating identities

Yes, supporting name transfer/name expiry/etc would be good to add.

Reply ↓ (https://blog.printf.net/articles/2015/05/29/announcing-gittorrent-a-decentralized-github/?replytocom=258003#respond)

Adrian Knoth *said* on May 31, 2015 at 11:37 am (https://blog.printf.net/articles/2015/05/29/announcing-gittorrent-a-decentralized-github/#comment-258278):

Hi!

I've been waiting for this since 2010 when I read the following mail on debian-devel: https://lists.debian.org/debian-user/2010/09/msg00052.html (https://lists.debian.org/debian-user/2010/09/msg00052.html)

Debian packaging often happens in git repositories. With GitTorrent, these repos don't need to be stored on a centralised server (git.debian.org) but could be distributed as well.

(http://www.flickr.com/photos/ashesoftimes/8079234402)

Chris Ball (https://blog.printf.net/)

(HTTPS://BLOG.PRINTF.NET/)

External links:

photographs (https://plus.google.com/u/0/photos/112938759017605010116/albums/5637573932501306529)

on github (http://github.com/cjb)

on google+ (http://gplus.to/chrisball)

on linkedin (http://www.linkedin.com/in/chrisjball)

on twitter (http://twitter.com/cjbprime)

on youtube (http://youtube.com/user/cjbprime/videos)

Most popular tags:

ethics (/articles/tag/ethics)
gnome (/articles/tag/gnome)
guitar (/articles/tag/guitar)
kernel (/articles/tag/kernel)
linux (/articles/tag/linux)
olpc (/articles/tag/olpc)
photography (/articles/tag/photography)

WP-SVBTLE (HTTPS://GITHUB.COM/GRAVITYONMARS/WP-SVBTLE)

We usually sign our git tags to testify that everything up to this commit is correct and denotes the official package. More importantly, since all the maintainers have cross-signed their keys, we can prove authorship with a WoT.

Putting everything together, GitTorrent allows for fully distributed Debian development. One could even just clone all the repos, verify the signatures on the tags and recompile the binaries from source if they don't trust their local Debian mirror.

Nice work!

Reply ↓ (https://blog.printf.net/articles/2015/05/29/announcing-gittorrent-a-decentralized-github/?replytocom=258278#respond)

Kristofer *said* on May 31, 2015 at 12:07 pm (https://blog.printf.net/articles/2015/05/29/announcing-gittorrent-a-decentralized-github/#comment-258291):

I don't know if it has been pointed out yet but "because the miners could see your message, and decide to modify it before adding it to the blockchain" is not true. Pretty much all transactions include a signature in them which would become invalid if a miner changed a single byte of your message. The miners could however just ignore your message but as long as there are legit miners left (aka 51% attack) it will eventually get on the blockchain.

Another possibility could perhaps be that not only a miner ignores your message but creates a new one because the scheme doesn't care who broadcasts the name claims. The success of this would again be proportional to the hash rate of the miner but is a valid threat. (which might be what you intended to say in the first place but "modify" is not the right word here)

Reply ↓ (https://blog.printf.net/articles/2015/05/29/announcing-gittorrent-a-decentralized-github/?replytocom=258291#respond)

cjb *said* on May 31, 2015 at 12:32 pm (https://blog.printf.net/articles/2015/05/29/announcing-gittorrent-a-decentralized-github/#comment-258303):

Thanks! You're the first. Changed "modify" to "replace".

Reply ↓ (https://blog.printf.net/articles/2015/05/29/announcing-gittorrent-a-decentralized-github/?replytocom=258303#respond)

Aaron Toponce (https://pthree.org) *said* on May 31, 2015 at 1:13 pm (https://blog.printf.net/articles/2015/05/29/announcing-gittorrent-a-decentralized-github/#comment-258310):

The one-word-per-line-nested-comments are killing puppies. Anything you can do to fix that?

Reply ↓ (https://blog.printf.net/articles/2015/05/29/announcing-gittorrent-a-decentralized-github/?replytocom=258310#respond)

cjb *said* on June 5, 2015 at 5:57 pm (https://blog.printf.net/articles/2015/05/29/announcing-gittorrent-a-decentralized-github/#comment-260822):

Fixed!

Reply ↓ (https://blog.printf.net/articles/2015/05/29/announcing-gittorrent-a-decentralized-github/?replytocom=260822#respond)

Andrew *said* on June 1, 2015 at 11:15 am (https://blog.printf.net/articles/2015/05/29/announcing-gittorrent-a-decentralized-github/#comment-258677):

(http://www.flickr.com/photos/ashesoftimes/8079234402)

Chris Ball (https://blog.printf.net/)

(HTTPS://BLOG.PRINTF.NET/)

External links:

photographs (https://plus.google.com/u/0/photos/112938759017605010116/albums/5637573932501306529)

on github (http://github.com/cjb)

on google+ (http://gplus.to/chrisball)

on linkedin (http://www.linkedin.com/in/chrisjball)

on twitter (http://twitter.com/cjbprime)

on youtube (http://youtube.com/user/cjbprime/videos)

Most popular tags:

ethics (/articles/tag/ethics)
gnome (/articles/tag/gnome)
guitar (/articles/tag/guitar)
kernel (/articles/tag/kernel)
linux (/articles/tag/linux)
olpc (/articles/tag/olpc)
photography (/articles/tag/photography)

WP-SVBTLE (HTTPS://GITHUB.COM/GRAVITYONMARS/WP-SVBTLE)

Your discussion about how to register usernames using the blockchain is almost identical to the way the peer-to-peer microblogging app Twister works. You would probably find that project interesting: http://twister.net.co/ (http://twister.net.co/)

Reply ↓ (https://blog.printf.net/articles/2015/05/29/announcing-gittorrent-a-decentralized-github/?replytocom=258677#respond)

Andy Chambers *said* on June 2, 2015 at 4:05 am (https://blog.printf.net/articles/2015/05/29/announcing-gittorrent-a-decentralized-github/#comment-259053):

There's something I don't understand. Once a peer has figured out what needs to be sent, how can other peers participate in sending it unless they have previously sent the exact same set of changes?

Reply ↓ (https://blog.printf.net/articles/2015/05/29/announcing-gittorrent-a-decentralized-github/?replytocom=259053#respond)

cjb *said* on June 2, 2015 at 11:22 am (https://blog.printf.net/articles/2015/05/29/announcing-gittorrent-a-decentralized-github/#comment-259165):

You're exactly right, they can't. Swarming only works for popular packs in this design. I'm looking into moving from BitTorrent to IPFS to fix this.

Reply ↓ (https://blog.printf.net/articles/2015/05/29/announcing-gittorrent-a-decentralized-github/?replytocom=259165#respond)

Kiran *said* on June 3, 2015 at 12:56 pm (https://blog.printf.net/articles/2015/05/29/announcing-gittorrent-a-decentralized-github/#comment-259684):

I wanted to have code, bugs and testcases in single repository. At last my wish is coming true :).

Please let me know how are you going to store bugs.

I think we should also write a client side app to view and modify the repository contents.

Thanks for great work.

~Kiran

Reply ↓ (https://blog.printf.net/articles/2015/05/29/announcing-gittorrent-a-decentralized-github/?replytocom=259684#respond)

Daniel Marbach (http://www.planetgeek.ch) *said* on June 9, 2015 at 7:15 am (https://blog.printf.net/articles/2015/05/29/announcing-gittorrent-a-decentralized-github/#comment-262590):

Hi Chris,

Have you checked:
https://www.ethereum.org/ (https://www.ethereum.org/)

Might be a good help
Regards
Daniel

Reply ↓ (https://blog.printf.net/articles/2015/05/29/announcing-gittorrent-a-decentralized-github/?replytocom=262590#respond)

Rusty Russell (http://rusty.ozlabs.org) *said* on June 11, 2015 at 11:37 pm (https://blog.printf.net/articles/2015/05/29/announcing-gittorrent-a-decentralized-github/#comment-

(http://www.flickr.com/photos/ashesoftimes/8079234402/)

Chris Ball (https://blog.printf.net/)

(HTTPS://BLOG.PRINTF.NET/)

External links:

photographs (https://plus.google.com/u/0/photos/112938759017605010116/albums/5637573932501306529)

on github (http://github.com/cjb)

on google+ (http://gplus.to/chrisball)

on linkedin (http://www.linkedin.com/in/chrisjball)

on twitter (http://twitter.com/cjbprime)

on youtube (http://youtube.com/user/cjbprime/videos)

Most popular tags:

ethics (/articles/tag/ethics)
gnome (/articles/tag/gnome)
guitar (/articles/tag/guitar)
kernel (/articles/tag/kernel)
linux (/articles/tag/linux)
olpc (/articles/tag/olpc)
photography (/articles/tag/photography)

WP-SVBTLE (HTTPS://GITHUB.COM/GRAVITYONMARS/WP-SVBTLE)

263934):

Naming uniqueness can't be proven by merkle proofs, unfortunately. But if you use – you avoid the uniqueness race for short names as well as almost always getting a unique handle (if someone else gets the same name in your block, try again).

Reply ↓ (https://blog.printf.net/articles/2015/05/29/announcing-gittorrent-a-decentralized-github/?replytocom=263934#respond)

BillDStrong (http://ndxtreme.com) *said* on June 13, 2015 at 8:29 am (https://blog.printf.net/articles/2015/05/29/announcing-gittorrent-a-decentralized-github/#comment-264533):

The biggest benefit I see with Github is not its centralized nature, but its uptime. No matter what time or day it is, it is up. How do you answer the availability question for one owner projects? How do they spread?

I think the tech is cool, and I applaud the goals. I like them for the same reasons I am interested in IPFS. (They have some goals to allow layers such as git on top of them, btw. No telling when they will get to it.)

IPFS is more generalized, and does solve the problem by their filecoin idea, as well as server hosting. Git itself always focused on server hosting as that is its main benefit for large groups. Bittorent doesn't have that problem, as it was content agnostic, and relied on the fact that popular content would remain popular. But this doesn't work for small software houses spread over the world with, say, three coders all working on their laptops that go to sleep at different times, and sometimes they are not accessible.

Reply ↓ (https://blog.printf.net/articles/2015/05/29/announcing-gittorrent-a-decentralized-github/?replytocom=264533#respond)

Ryan Hellyer (https://geek.hellyer.kiwi/) *said* on June 17, 2015 at 4:29 pm (https://blog.printf.net/articles/2015/05/29/announcing-gittorrent-a-decentralized-github/#comment-266758):

Fascinating concept. So long as hash collisions are not a problem (I have no idea how hard it is to brute force the hash algorithm used in Git), then this sounds like a sane and useful idea.

Reply ↓ (https://blog.printf.net/articles/2015/05/29/announcing-gittorrent-a-decentralized-github/?replytocom=266758#respond)

Pingback: *GitTorrent – An Oxymoron? | 21st century storage: more than just faster disks (http://storagetarget.com/2015/06/23/gittorrent-an-oxymoron/)*

Drew (http://VancouverTechPodcast.ca) *said* on January 28, 2016 at 3:15 am (https://blog.printf.net/articles/2015/05/29/announcing-gittorrent-a-decentralized-github/#comment-322165):

Sorry for necro-ing an older post, but it seems particularly relevant today, as GitHub was down for around 90 minutes!

An entertaining idea around GitTorrent: https://news.ycombinator.com/item?id=10984775 (https://news.ycombinator.com/item?id=10984775)

🙂

Reply ↓ (https://blog.printf.net/articles/2015/05/29/announcing-gittorrent-a-decentralized-github/?replytocom=322165#respond)

C. Scott Ananian (http://cscott.net) *said* on January 28, 2016 at 5:30 am (https://blog.printf.net/articles/2015/05/29/announcing-gittorrent-a-decentralized-github/#comment-

(http://www.flickr.com/photos/ashesoftimes/8079234402)

Chris Ball (https://blog.printf.net/)

(HTTPS://BLOG.PRINTF.NET/)

External links:

photographs (https://plus.google.com/u/0/photos/112938759017605010116/albums/5637573932501306529)

on github (http://github.com/cjb)

on google+ (http://gplus.to/chrisball)

on linkedin (http://www.linkedin.com/in/chrisjball)

on twitter (http://twitter.com/cjbprime)

on youtube (http://youtube.com/user/cjbprime/videos)

Most popular tags:

ethics (/articles/tag/ethics)

gnome (/articles/tag/gnome)

guitar (/articles/tag/guitar)

kernel (/articles/tag/kernel)

linux (/articles/tag/linux)

olpc (/articles/tag/olpc)

photography (/articles/tag/photography)

WP-SVBTLE (HTTPS://GITHUB.COM/GRAVITYONMARS/WP-SVBTLE)

322168):

Although not entirely decentralized, I like the idea of using a hierarchy and multiple roots to bootstrap the username system. Something like:
gittorrent://cscott.net/username/reponame

Where a TXT entry on cscott.net gives an initial hash for the distribution key/value store, and this is used to publish a username->key registry. This lets you bootstrap a number of different username mappings, instead of relying on a single immutable registration in a blockchain. Trust is delegated to the domain owner to maintain your name mapping. If you don't trust cscott.net, use a different domain/registry.

Wrt swarming the packs, one option is using something like the rsync rolling checksum algorithm to decide pack boundaries. This makes it much more likely that folks can share packs.

For instance, if the commit history is AAA (root commit), BBB, CCC, DDD, EEE (latest commit), then we pick a hash algorithm h and a value N, and for each commit C compute h(C) % N, and see if the result is 0 (which will be the case 1/N of the time). Say CCC is a commit for which this is true. Then a request for EEE will actually give you the pack from EEE to CCC and direct you to request CCC to complete the clone. The requests for CCC (and earlier) are now much more likely to be swarmable.

If you double N for each recursive request, you end up with lg(commits in history/N) packs, all but the first few are swarmable.

Reply ↓ (https://blog.printf.net/articles/2015/05/29/announcing-gittorrent-a-decentralized-github/?replytocom=322168#respond)

sohalt (http://sohalt.net) *said* on January 28, 2016 at 10:33 am (https://blog.printf.net/articles/2015/05/29/announcing-gittorrent-a-decentralized-github/#comment-322172):

As mentioned already above it would be nice to have some way of identity management (username changes/depreciation). Alongside that I would like to know your thoughts on how to handle compromised private keys, either because they got stolen/leaked or the crypto doesn't stand up any more. Basically you would need to be able to rotate/change keys. Should be solvable by updating the name resolution system, only then you end up with the problem of having to guarantee only the owner of a user name to update the key. Which you could do for example by requesting a "key change message" be signed by the former key, but that only helps in the "update to newer crypto" scenario, not when the key got stolen. Alternatively you could embed a cryptographic hash of a token in the name registration payload with the token allowing a one time change of the name-key association (again providing a new hash of a new token to be used on the next change). This approach though only shifts the problem of having to keep the key safe to having to keep the token safe (which might provide a slight benefit, because the token is not needed in everyday use an can comfortably be stored on a piece of paper and is therefore less susceptible to compromise — although the same could be achieved by using an airgapped signing key and subkeys for day to day use).

Reply ↓ (https://blog.printf.net/articles/2015/05/29/announcing-gittorrent-a-decentralized-github/?replytocom=322172#respond)

C. Scott Ananian (http://cscott.net) *said* on January 28, 2016 at 2:40 pm (https://blog.printf.net/articles/2015/05/29/announcing-gittorrent-a-decentralized-github/#comment-322176):

My DNS-bootstrapped username registry would handle key rotation w/o a problem. The owner of the domain can update the public key stored in the TXT record, and/or update the keys stored in the bootstrapped distributed username registry on the user's behalf.

If one registry goes down, you can just switch to a different one. This would be equivalent to switching to a new username in the bitcoin-based registry, but the hierarchy of the DNS based system means that, to a human, the change appears as a switch from `cscott.net/cjb` to `printf.net/cjb` instead of as a switch from `cjb` to `cjb2`. I think keeping the 'username' part stable

(http://www.flickr.com/photos/ashesoftimes/8079234402)

Chris Ball (https://blog.printf.net/)

(HTTPS://BLOG.PRINTF.NET/)

External links:

photographs (https://plus.google.com/u/0/photos/112938759017605010116/albums/5687573932501306529)

on github (http://github.com/cjb)

on google+ (http://gplus.to/chrisball)

on linkedin (http://www.linkedin.com/in/chrisjball)

on twitter (http://twitter.com/cjbprime)

on youtube (http://youtube.com/user/cjbprime/videos)

Most popular tags:

ethics (/articles/tag/ethics)

gnome (/articles/tag/gnome)

guitar (/articles/tag/guitar)

kernel (/articles/tag/kernel)

linux (/articles/tag/linux)

olpc (/articles/tag/olpc)

photography (/articles/tag/photography)

WP-SVBTLE (HTTPS://GITHUB.COM/GRAVITYONMARS/WP-SVBTLE)

is more human-friendly, although you do have to contend with confusion attacks: csc0tt.net/cjb vs cscott.net/cjb, for example. But those exist even in the bitcoin-based scheme (cscott vs csc0tt), so it's a wash.

Another benefit of the DNS-based scheme is that github could decide to support it simply by publishing an appropriate TXT record and allowing users to upload a desired public key (or, better yet, by bootstrapping based on the SSH public keys they already have in their db). This would let github get out of the disks-and-bandwidth game and concentrate on being the best web UX for git repos (wherever/however they are stored).

Reply ↓ (https://blog.printf.net/articles/2015/05/29/announcing-gittorrent-a-decentralized-github/?replytocom=322176#respond)

C. Scott Ananian (http://cscott.net) *said* on February 8, 2016 at 4:51 pm (https://blog.printf.net/articles/2015/05/29/announcing-gittorrent-a-decentralized-github/#comment-322701):

Have you considered adding support for gittorrent to gitlab? This would allow a completely decentralized system (anyone can run their own gitlab UX allowsing access to the decentralized gitlab storage) while also allowing the single "public gitlab" to serve as a convenient centralized destination to simplify certain tasks — for example, username assignment, key management, guaranteed seeding of certain files, etc. The benefit of this model is that because of the inherent decentralization, it would be completely transparent to take over (say) seeding yourself, or make [illegible]ity, etc. You could also run your own gitlab server and use it to access the decentralized cloud of gittorrent projects, completely decoupling the UX (gitlab) from the implementation/store (gittorrent).

FWIW, it would also bring gitlab more on-par with services like the recent "Google Cloud Source Repositories" (http://venturebeat.com/2015/06/24/google-has-quietly-launched-a-github-competitor-source-code-repositories/ (http://venturebeat.com/2015/06/24/google-has-quietly-launched-a-github-competitor-source-code-repositories/)) — *anyone's* sources could be "stored/secured in the cloud", they just need to distribute gittorrent seeds around, and anyone's install of gitlab will be sufficient to work on any distributed project.

Reply ↓ (https://blog.printf.net/articles/2015/05/29/announcing-gittorrent-a-decentralized-github/?replytocom=322701#respond)

Chris (http://printf.net/) *said* on February 8, 2016 at 4:59 pm (https://blog.printf.net/articles/2015/05/29/announcing-gittorrent-a-decentralized-github/#comment-322704):

> Have you considered adding support for gittorrent to gitlab?

Totally hadn't! That's a neat idea.

Reply ↓ (https://blog.printf.net/articles/2015/05/29/announcing-gittorrent-a-decentralized-github/?replytocom=322704#respond)

Luke Kenneth Casson Leighton (http://lkcl.net) *said* on May 5, 2016 at 10:07 pm (https://blog.printf.net/articles/2015/05/29/announcing-gittorrent-a-decentralized-github/#comment-328247):

hi chris, it's good to see that somebody finally implemented gittorrent. i published an article about the concept back in 2008 (http://www.advogato.org/article/994.html (http://www.advogato.org/article/994.html)) and a guy called sam implemented something that he renamed "mirrorsync". sam didn't quite "grok" the concept in the same clear way that you clearly get it, and we also have, since then, had the addition of "blockchains".

chris: i see no reason why it should be necessary to rely on *bitcoin* for a blockchain. it should be perfectly and clearly logical and reasonable, especially if you are going to assume that there are DHT nodes out there, to simply run a completely independent blockchain service *at the same time*. with the advantage that you'd no longer be dependent on bitcoin, and, additionally, you'd be running a DHT so there would be no central servers. also, you really don't want a ton of

(http://www.flickr.com/photos/ashesoftimes/8079234402)

Chris Ball (https://blog.printf.net/)

(HTTPS://BLOG.PRINTF.NET/)

External links:

photographs (https://plus.google.com/u/0/photos/112938759017605010116/albums/5637573932501306529)

on github (http://github.com/cjb)

on google+ (http://gplus.to/chrisball)

on linkedin (http://www.linkedin.com/in/chrisjball)

on twitter (http://twitter.com/cjbprime)

on youtube (http://youtube.com/user/cjbprime/videos)

Most popular tags:

ethics (/articles/tag/ethics)

gnome (/articles/tag/gnome)

guitar (/articles/tag/guitar)

kernel (/articles/tag/kernel)

linux (/articles/tag/linux)

olpc (/articles/tag/olpc)

photography (/articles/tag/photography)

WP-SVBTLE (HTTPS://GITHUB.COM/GRAVITYONMARS/WP-SVBTLE)

bitcoins to have to download: i can't remember how big the current blockchain is (over a gigabyte?) but i sure as hell don't want to be downloading gigabytes worth of blockchain crap…. and then find that the project i'm sync'ing is 10k and contains 2 text files. that would be beyond *I*ronic, and bordering on *MO*ronic – it would be a huge burden that would actively discourage people from using the service.

other alternatives: plain-old GPG keys, especially those which have been registered with a key server as well as being part of a key-signing exercise (debian keyring for example).

a couple of really important things, though:

(1) due to the way that the pack-object is generated, there is NO GUARANTEE that the pack object is the exact same thing across multiple machines… or even the same machine (threads can execute out-of-order and return *different* results in the pack-object search algorithm). so you can't just "grab a commit range", it *will* be different.

so you're going to have to take an md5 checksum or sha1 checksum of the pack-object, and add an extra step to make sure that the pack-object is identical across multiple machines. the extra stage i considered is, you have an "auction". contact multiple machines, they all do the same "aaa bbbb" thing, they alll return a SHA1 answer as well as a file size, and also their available bandwidth allocated to uploads. then you begin downloading from that fastest machine *and* one other random machine, simultaneously. at some point you go, "hmmm, which one is quicker?" and you drop the slowest one. you get the idea, i'm sure, but i'm into "optimisation' here in the latter phase. the first phase, however, is ESSENTIAL.

(2) an option to only accept GPG-signed commits is ESSENTIAL in a distributed network. you do NOT want to be picking up random pack-objects from random unverified sources. some idiot, sooner rather than later, is guaranteed to try to f*** things up by answering with unadulterated random crud. you've already seen evidence of this in the film industry – not only fake torrents but fake clients uploading *literally* random crud, flooding the network in the hope of stopping downloads from happening. doesn't work, but they still try. gpg-signed commits has been a feature forever…. so use it. it's part of git infrastructure.

what would be nice is a combination of gpg and blockchain. it's probably already been done, somewhere.

p.s. irony: i've been around long enough to remember the precursor to blockchains. raph levien – the creator of the trust metrics algorithm behind advogato – was one of the people who researched and advocated it. but i've been around long enough to have forgotten the damn name. digitally-signed algorithms that were executed as verification for operations on publicly-accessible records such as DNS. if the algorithm (which was a formally-provable mathematical language) executed "true", an action was permitted, and of course it was distributed, so all recipients of the same distributed data could of course carry out the exact same operation, independently, and still maintain synchronous state. it was advocated for use in DNS (to make DNS decentralised – no more "registrars"). keynote! ha! got it! remembered it, yay! took…. minutes. argh. ha i forgot, it was published as an RFC: https://tools.ietf.org/html/rfc2704 (https://tools.ietf.org/html/rfc2704)

anyway, that – or similar – would do nicely here.

Reply ↓ (https://blog.printf.net/articles/2015/05/29/announcing-gittorrent-a-decentralized-github/?replytocom=328247#respond)

lesto *said* on June 6, 2018 at 8:00 am (https://blog.printf.net/articles/2015/05/29/announcing-gittorrent-a-decentralized-github/#comment-389832):

nice idea, but for the search i would more focus on having a known entry point; for example imagine situation at a work or at a coding jam where you know the IP of the other people working on the repo and you "just" need to collaborate with them

Reply ↓ (https://blog.printf.net/articles/2015/05/29/announcing-gittorrent-a-decentralized-github/?replytocom=389832#respond)

Leave a Reply

Your email address will not be published. Required fields are marked *

(http://www.flickr.com/photos/ashesoftimes/8079234402)

Chris Ball (https://blog.printf.net/)

(HTTPS://BLOG.PRINTF.NET/)

Comment

Name *

Email *

Website

Post Comment

← READ MORE (HTTPS://BLOG.PRINTF.NET/)

External links:

photographs (https://plus.google.com/u/0/photos/112938759017605010116/albums/5637573932501306529)

on github (http://github.com/cjb)

on google+ (http://gplus.to/chrisball)

on linkedin (http://www.linkedin.com/in/chrisjball)

on twitter (http://twitter.com/cjbprime)

on youtube (http://youtube.com/user/cjbprime/videos)

Most popular tags:

ethics (/articles/tag/ethics)

gnome (/articles/tag/gnome)

guitar (/articles/tag/guitar)

kernel (/articles/tag/kernel)

linux (/articles/tag/linux)

olpc (/articles/tag/olpc)

photography (/articles/tag/photography)

WP-SVBTLE (HTTPS://GITHUB.COM/GRAVITYONMARS/WP-SVBTLE)