Mon Dec 23, 2019

Understanding the term, terminfo and pointer into postings and positions along with the document store.

Sun Dec 22, 2019

First day working on fst. I read some of the blog post by burntsushi and got some of the fst code up and running with the examples.

Sat Dec 21 (winter solstice, hb is going to the party in OR)

So the year is ending on a good note, and my rust skills are in pretty good shape. I am finally starting to feel a bit productive, after starting to code in Rust on Scott's bday this year, so its been over 5 months now.

Today I got the code working to rebuild a json document based on the keys in the tantivy schema. It involved using a Map from serde-json, crossbeam channels for the first time, HashSets. I am pretty confident I can take a few weeks off from coding while I am back home in NM and get right back into some time in mid January once I am back home in OR.

Wed Dec 18

Up and running (again) with tantivy-cli. I got a new working example up and running, and pretty much understand how it works. Will outline the next steps in coming days. But for now I got the data coming from Hn and going into Redis, Sled, and Files which is where I wanted to get to. Now I am going to get up and running with easily getting the data into Tantivy.

Tuesday Dec 17, 2019

Moving on now to a new architecture with channels. So both IDs and Json Strings will go on channels moving forward.

Sunday Dec 15, 2019

Starting to get sled working with the hacker news json data. First cut at sled up and running and finished.

Tuesday Dec 10, 2019

The day of flying to Florida I got the data into redis in the repo redsled. I just took the code from yesterday and added the method to write the data that was already in a vector out to redis.

Monday Dec 09, 2019

The day before going to Fl I got the lifetimes example working noted in last Friday's entry. This is kind of HUGE and the next step in learning Rust. As I look over the Rust Book and what I still don't know, this was something I definitely needed to get down. So now I can take the rest of the day off :) See you in Florida. Its been a very productive time in Rust land here in Pittsburgh. I pretty much worked almost every single day and didn't take any days off while I was home. So until I get back to Oregon in mid January you can rest assured knowing you aren't a slacker when it comes to work :)

Friday Dec 06, 2019

So I have been coding in Rust pretty much every day since arriving here in Pittsburgh and from the beginning on July 10. Today is the first time I ran into the borrow checker, lifetimes, and ownership, so this is pretty cool.

Here is the problem. I am reading data out of a file and passing that data into a Vector which then gets passed along somewhere else. The line iterator of the file owns the data and I want to be able to have a lifetime stating that the data lives on after the line iterator goes out of scope but the vector still can have a reference to the data.

The fact that I am able to create a problem that uses these concepts is pretty cool. This may take a while to wrap my head around so stay tuned.

Thursday Dec 05

Booked a ticket today to fly to ABQ on Dec 25 for $189. Believe it or not but this one line of code took me a couple of hours to figure out. Where keys is a vector, and I was trying to assign it to a variable via let x = but all it wanted was Vector on its own line calling the sort method. I was sorting a Vector of hash keys which are the hackernews ids for stories only.

```
 let mut keys = get_hashmap_keys("hn-story-19".to_string()).unwrap();
keys.sort();
```

Monday Dec 02

So the null checks and everything else is running, now moving on to storing the data using both sled and probably redis too.

Saturday Nov 30

Lunch at Cafe 33 Taiwanese, yummy.
Got null checks working in the repo hn-api-examples

Friday Nov 29, 2019

hn-api-examples is up and running and I am parsing tens of thousands of hn json's with no errors. only thing left to do is handle the case where there is a null.

make sure you understand this line of code:

```
Item::Comment(comment) => Some(&comment.by.as_ref().unwrap()),
```

Monday Nov 18

Up and running and the serde deserialization of a Hackernews json object is up and running.

Saturday Nov 16

Working on serde serial, deserial of hackernews in the context of spaceapi-rs. It is starting to work...

Thursday Nov 14

Note when I started coding in Rust we were at 1.36.0 and now today I installed 1.39.0

So the past couple of days in Pittsburgh have been trying and a bit brutal getting my async, tokio code working, but I am there with this rust-hackernews code. It iterates over a set of hackernews favorite ids, gets the JSON body from here, and writes the data to a redis hashmap. Pretty cool, and I believe it helped today to install the latest stable version of rust 1.39.0 which is the first version to support async-await and here is the hackernews post which got 1100 points, wow ! Initially I tried to tackle it using the hyper client code but then quickly realized that reqwest which was a much easier route to take.

Saturday Nov 9

I am in pretty good shape with the tantivy schema mod, and today I got a good handle on NamedFieldDocument which I learned is mainly used to help with document conversion to json.

Tuesday Nov 05, 2019

In the morning before flying out to Pittsburgh. More work on tantivy, now looking at the Document and the schema crate in tantivy in relationship to the spaceapi-server concept but using this api to store tantivy documents, elastic-search documents, and possibly the concept of writing a tutorial on how to port data from elastic to tantivy using vector ?

also, with redis in mind fully understand how to use the RedisPool. I will basically get this for free as I unwind the spaceapi-server. For a second opinion on redis-pool take a look at sidekiq as well...

Saturday October 26, 2019

This week I started working on search in rust, more specifically, **tantivy and sonic**. I am not sure how I got here, but basically I wanted to store all of my hackernews favorite json files on redis or sled and then be able to search across them when it hit smack in the middle of the face that why not use a search engine library, and then I found tantivy and sonic. Life continues to twist and turn and its very strange how you discover the rocks and then eventually you turn them over.

Tuesday October 22, 2019

Maria's birthday and Day 1 of tantivy.

Friday Sep 27

Finally after many months of trying to understand Rust I am starting to code. The past 2 days have been working on reading the data back from Redis that maman writes to it... So I am finally am able to get at the data, and in doing so, am actually writing some Rust code.

Thursday Sep 26

Some work on maman and getting serde to serialize and deserialize data from redis to a file etc...

Monday Sep 23

Found out about jonhoo and his amazing Rust podcasts on Youtube. Making more progress on understanding Rust streams and futures. Today was a good day of understanding and getting more turned on to the concept of Pinning, which is well elucidated in one of John's videos.

Thur Sep 19

See worklog trail on vector, http, hyper, async, await, future understanding...

Wed Sep 11

Starting to understand events in the context of vector.

Tuesday Sep 10

Sendlines is sending data into the elastic sink.

Sendlines is up and running and I am successfully sending events from this rust example into the server; very cool and a significant event (NPI) in the evolution of my vector journey.

Sunday Sep 01, 2019

Found vector via this blog post about tracing. I have been interested in both tokio and log tracing and so when I saw that vector was using both tokio and tracing I was happy. Vector is going to be my ***etcd*** project for taking my Rust programming skills to the next level. At this moment it has 2177 stars which is enough to give the project some validity.

Other mission critical crates:

- serde
- indexmap

Star Count over time

- 2019 september = 2177

Friday Aug 23

Very deep in the bowels of Rust. Today was typical across the board. Started out by looking at and understanding Codecs in Tokio followed by my first real foray into reading about macros followed by a glimpse into the internals of the rust compiler by just a peripheral reading of the rust compiler docs.

Wed Aug 21

Looking at the hyper and tower web frameworks in the context of tokio and futures.

Tuesday Aug 13

Starting to understand methods, versus free standing functions and traits.

Monday Aug 12

First time really starting to code using sled as my basis for learning.

Saturday Aug 10

Starting to look into persistence engines such as sled.

Thursday Aug 8

Cheryl and Michael's 8/8 party. More work on diesel in the context of Rocket. Working on a better understanding of lifetimes and macros in rust.

Sunday Aug 4

Postgresql up and running on my mac in prep for some diesel and crate.io work.

Wednesday July 31

More coding Tokio and Actix examples. This stuff is interesting.

Wednesday July 24

More coding in Rust with Tokio and a better understanding of async. Also how it compares to Golang, pros and cons.

Monday July 22

Up and running with Rust and talking to redis. So I am just going to start doing more programming in Rust. Next up a better understanding of a rust hacker news client.

Wednesday July 17, 2019

So I have spent the past month working on the internals of flux and the month prior to that on influxdb. On May 10, 2019 I started working on the influxdb ecosystem in the context of a "test" system to work with the hashicorp ecosystem. I have spent enough time on the internals of flux that in the past 2 months while in the influxdb world that I learned two things. Two things in two months, pretty cool. And the 2 things are:

1) Rust
2) Programming Language Theory of Types

So with that I will parallel track both above topics deeper and then circle back on Flux in both worlds. But the above two topics will not be a full time endeavor and so back into the influxdb world in tht context of equity data, but keeping abreast of the work going on in flux at the source code level. Specifically Rust and Affo's work on the redesign of Compilation and Execution along with Pull Request 1332.

Friday July 12, 2019

So I will remember Scott's birthday 2019 as the time when I started programming in Rust. My first programming in Rust was version 1.36.0

I will remember my first golang project in Ashland in November of 2014. This is the first November we spent in Ashland as November 2013 was the month we flew back home to NM from Oregon having just bought our house August 30, 2013.

Thursday July 11, 2019

Day 1, Rust motivated by this issue on Fluxlang along with this blog post by Paul Dix at Influxdata.

As I think back on my programming career I can only really think of learning 5 new programming languages in the 21st century. In order it was Javascript, Python, Ruby, Golang and now Rust.

I learned Java in 1995 at Sfi and rode that wave through all of my career at Vitria into my time at Justice Systems when Brad turned me on to Javascript.

I learned Python prior to moving out to Caltech because Google's first implementation of their cloud infrastructure code used Python. At the time Guido was working at Google.

I learned Ruby at the tail end of my time at Caltech as the final version of CRData was written in Ruby. I wrote the first version in Python.

In Buenos Aires I exclusively programmed in Javascript for my classroom image project and then upon my return to Placitas where I heavily got into Node.js for my project with Milewise and their CEO Vinay Pulim.

I did all of my programming at Spinnakr in Ruby until the final year when I rewrote part of our code at Spinnakr in Golang.

I should mention that I briefly looked at Rust in the past but it might have only been for a day or two.

Sunday July 7, 2019

Big time progress on understanding this repo

```
csv.from(csv: data) |> sum(column: "value")
```

Now I have some more stuff to sink my teeth into and move forward from here. This could be the starting point for lots more independent testing of the different flux functions.

Thursday July 4, 2019

So I got the gdayt code up and running, grabbing a set of filenames from a directory. I also have the date code working in go-examples/csv. So I am able to convert the ubnt.csv file into a file that can be written to influxdb. With that data in influxdb I will now circle back to understanding the flux scripts.

Monday July 1

First day back to work, and started working on remembering how to write data to influxdb and then starting to work on reading a CSV file and doing the time calculation correctly on the date.

Saturday June 29

David Kleiman's birthday. I am back from Canada and we had a great trip. We drove home in one day from the Osoyoos border and had a nice Thai lunch in Yakima. Here are my plans moving forward, if you can stay focused and get this job done you will be in good shape. The last major task I undertook was the material-ui tutorial. And you got that job done. I will go back through this work log and see what I have done this year, but not right now. Stay tuned for an update on the progress to date at the half year point.

In a nutshell here is what I need to do...
1) Get daily closing data into InfluxDb
2) Get fundamental data into InfluxDb. For now this will include just the spreadsheet items I have listed quarterly for Ubnt.
3) Once the data is loaded up then calculate the 50 and 200 day moving averages using flux.
4) Also calculate the trailing PE and forward PE.

Thursday June 13

Boyd's 78th birthday today. I got fluxspec up and running which is very cool. It just worked out of the box after a failed attempt with spec-md. I also realized today why you should always create a branch with a public repo prior to commiting. It enables you to update the master branch from the public repo, and continue the branch you created untouched. Can't believe it took me this long to understand this concept. I realized it when my other pull request wasn't taken and I had to go out and get the latest code. When I did that and commited back I had to do a merge. Git continues to amaze me.

Tuesday June 11

So I have been spending the past number of days exclusively on the internals of flux and I am at a place where I can now move back into Influxdb and start writing some more sophisticated queries. Combined with the test data from the generate command and my flux knowledge I should begin to be able to create a cookbook or tutorial of flux queries in parallel with a deeper understanding of flux internals.

I am getting ready for our trip to Canada so it might be awhile before my next entry, but at least we have a plan when we get back from the adventure.

Friday May 31

I am now building flux queries for the first time. Just got this going a couple of hours ago. It is Friday evening and we are making more progress on the Influx ecosystem. Yesterday I learned about schemas and the influxd generate command which gives me the ability to generate out some really cool test data. This product continues to impress me.

Wednesday May 23

For the first time got data from a text file into influxdb using the line protocol. I was also able to see the data in the chronograph ui. This is great news ! I have been messing around with this stuff for awhile and this is the first real progress.

Today I went to KP and had a visit with Sheila Jhansale who gave me a good report. I got an EKG done for the first time and she read it and said things looked good. Earlier in the day Hb and I walked in Minto Park and had lunch at the salem coop.

Friday May 17, 2019

Found go-quote which will be a big win for getting equity data into Influxdb.

Friday May 10, 2019

Start looking at the ***Influxdb ecosystem*** in the context of having it be a test bed for both hashicorp and docker.

Tuesday May 07, 2019

Can not believe I have not posted anything for over a month. This just shows you where I have been at in my life and what has been going on... So to quickly summarize the whole time that past month has been spent coming more and more up to speed on the Hashicorp ecosystem. I have gone through everything at a high level and pretty much understand TVCN {Terraform, Vault, Consul, Nomad}. Now in the context of running a nomad job for redis I am revisiting the docker concepts and commands. Its great that I have my new computer back because now I can run docker on my Mac which is pretty darn handy.

Wednesday April 10
Final day of work in Florida was completed with starting a deep dive into Terraform. The key point is that a separate process is running that Terraform talks to. Terraform calls into the particular Golang API for AWS, Digital Ocean or whatever other cloud its running on.

Terraform appears to be pretty cool. So now I have rounded off my initial understanding of the Hashicorp ecosystem and Terraform was the final shoe to drop in understanding. I have successfully tackled and completed phase I of the Hashicorp ecosystem. A lot more work needs to be done, but a good first step has been made.

Friday April 05
More work on the details of Vault src code including starting to watch some stuff on Youtube and look at the Google Groups account for vault.

Monday April 01
Vault Initialization Sequence is FINALLY nailed down

Friday March 29

Starting a deep dive into the Vault src code along with connecting up Vault to consul as one of the physical backends.

Wednesday March 27

Day 1 of Hashicorp Vault

Tuesday March 26, 2019

Please see worklog18 on Friday Nov 2, 2018 for my first mention of Hashicorp again.

Day 1 looking at the hashicorp ecosystem including nomad after reading this blog post. The code has to work out of the box and build and feel right. Traefik just seemed to be a bit broken and did NOT build out of the box. Nomad and the whole ecosystem seems to be the right chemistry and feels right and passes the smell test.

Sunday March 24

Spent some time looking at traefik.

Saturday March 23

Got my reverse-proxy working in go-examples.

Friday March 22

Ended the week on a deep dive into etcd.

Monday March 18

Worked all day on gostar3 and pretty much have it working.

Saturday March 16

Working on etcd, dgraph, tile38 on in the context of storage across more than one machine. dgraph accomplishes this via just using etcd raft, but I am not sure yet how this works. etcd uses boltdb for storing the key value pairs (on each node ?) I believe. So more research across the board on all of these topics. Could I use buntdb with raft and badger ?

Friday March 8, 2019

So I now have the btree up and running and printing out the items in the node along with the number of children each has. From this information I can pretty much discern what the tree actually looks like. I don't think there is a need to take the time to display it in the browser. As long as I can understand the nodes and the children and levels we are good.

Tuesday March 5, 2019

Continued deep dive into understanding B-Trees in depth.

Saturday March 2, 2019

Starting to look into B-Trees in the eventual context of Tile38.

Tuesday Feb 26, 2019

Arrive in Florida, Marsha picked us up at the airport and we went to Nino's for dinner.

Thursday Feb 21, 2019

So I am still working on just understanding how to build go repos with and WITHOUT go.mod. Today I banged my head against the wall for about an hour or two on NOT KNOWING that the

```
go get -t github.com/jpincas/tormenta
```

-t flag instructs get to also download the packages required to build the tests for the specified packages.

Wednesday Feb 20 [Eve's Birthday]

So needless to say a lot has happened. I am back in Pittsburgh with my Dad, my Mom passed Sunday Feb 10, 10 days ago. This is my first entry since leaving OR. I am diverting now since I have my old macbook and its important to relearn again Golang. So that is what I will be doing in concert with some other things. Right now I am looking at http load testing and on the other side the http server.

Friday Feb 8

I now understand a proxy, reverse and forward. Spending more time focusing specifically on Envoy and the details of that since it is at the core of Istio and more broadly used across the cloud landscape.

Thursday Feb 7, 2019

C++ sucks as bad as it always has. I tried building the Envoy Istio Proxy on my mac and it was a total joke. It took like 30 minutes to build and it still wasn't done. I eventually killed it off, that was the last time I will ever try and build a C++ binary like that again unless I am getting paid to build something like that. Pathetic how outdated the process of building something like that is compared to Golang which builds in like one minute for the same level of complexity. That is the difference between technology that is 30 years old versus 10 years old. No wonder no one ever writes C or C++ code.

Wednesday Feb 6, 2019

Hb coming home from Redmond today.
So, I love my job and here is why.

All of this stuff came out of the need for something in my professional life. As I started using DO more and more, and I was diving deeper into Istio I noted on their website yesterday the link Docker for Desktop.

Day 2 of Docker Desktop.
Day 1 of Prometheus.

```
docker run --name prometheus -d -p 127.0.0.1:9090:9090 prom/prometheus
```

Having the ability to run this above command is just GIGANTIC. It downloaded prom and I was up and running in literally less than 20 seconds. Docker Desktop is going to be a huge thing in my life going forward; this is clearly going to be one of my go to tools until the next big thing (like this) comes along.

Tuesday Feb 5, 2019

I am up and running with Docker Desktop and Kubernetes on the Mac using the Docker dmg file which seems to work smoothly.

Monday Feb 4, 2019

So I am fully up and running with Istio in terms of understanding at least how to run the helm command.

```
kubectl apply -f istio.yaml
```

The fact that this one command brings up everything in istio is very cool and actually quite amazing. Lots more work to do, but progress is clearly being made. I worked all day yesterday as well on this istio stuff, Hb went off to Redmond to be with the girls and I stayed home and had fun working.

Saturday Feb 2, 2019

I woke up this morning and istio and the sidecar proxy finally got figured out. The idea is simply this... Every application needs to call out to some type of API. Why not always have everyone call out to Envoy (the sidecar proxy) and then they can figure out, monitor, load balance and deal with who gets the request. Its like having a smart router sitting there figuring out how to route the traffic. Also, on the return trip the same thing. And the other huge win is that it is **not** EMBEDDED in the application. It doesn't have to be. In fact, the application doesn't know anything about the sidecar proxy, it just thinks its calling out to someone, and then getting a response back from someone.

Friday Feb 1, 2019

First time ever getting istio up and running fully on DO. My first attempt failed miserably as I was trying to bring it up on a 1 Gig machine. Then I went with the big guns of 4 Gigs and it worked beautifully.

Thursday Jan 31, 2019

So the effects of hearing about different stuff in my world comes together when you keep hearing about the same stuff. I came across this YouTube Post about Envoy several weeks ago and then when it appeared again in istio I knew I needed to spend more time with it... Plus it is totally related to the Docker and

Kubernetes work so it is a natural extension.

Wed Jan 30, 2019

Day 1 of hearing about and taking a look at istio which means sail in greek. So I am completed (for now) with Elasticsearch and writing out my HN Favorites to it. It all works, and Kibana works too.

Tuesday Jan 29

Messing around with Ruby, Rvm, Jekyll on my new Macbook. It took me hours to get things figured out because of the stupid

ruby version **versus** ruby --version ***command***

Any way, I don't believe I will ever make that mistake again with Ruby. And here are my Github Notes about that in the future.

Wed Jan 23

I am writing my HN Favorites out to a JSON file with hackernews-favorites.

Monday Jan 21

See entry from yesterday. Today I figured out how the redishacker code works. This code reads json data out of hackernews and writes it to redis. The code is a bit tricky, but I am pretty sure I now understand most of it. I also got a source code version of kibana up and running, which is actually pretty impressive. It just worked out of the box which is a pretty good sign.

Sunday Jan 20, 2019

Spent all day Sunday working on golang. I am sort of, kind of a bit back into golang. At least now I have a bit of a handle on go.mod as I struggled through getting my hackernews, redis, elasticsearch code back up and running. The key find was finding my code buried in another github repo. Boy, all of those github accounts actually come in handy and today was one of those big wins in finding the code. The 2nd big hurdle after finding the code was getting the modules stuff to work. I eventually ended up having to create a whole new repo with BulkStringRequest code in it; as forking olivere/elastic was not working as I could see the extra 2 files were not in ~/go/pkg... Any way, great news ! Now I have to figure out how the code actually works and then drive forward...

Wednesday Jan 16

Lots more work on Docker, especially in regard to getting nginx to run and all of the commands associated with it. I am documenting those commands now in my github repo which is actually a better spot than doing it in google docs. So the thing that drove me to continue on with Golang was realizing that getting a real live application deployed that works with Docker and Kubernetes probably makes sense. An application that I can scale out and demo to people that I know what the heck I am doing. So I am pulling out of the wood work my old code on redis, elasticsearch etc...

Thursday Jan 10, 2019

A week has passed and a lot has happened, the main accomplishment was researching, communicating, and purchasing a new computer which I desperately needed. I bought an early 2015 Macbook Pro for $800 on Tuesday evening at 8PM at Starbucks. I purchased my first Macbook Pro on December 28, 2011. So basically more than seven years has passed since I purchased my last computer. I got it up and running yesterday; it only took me a few hours. It was good to basically go prior to Ken coming over to go for a walk around the park.

Thursday Jan 03, 2019

First post of the new year and first post where I am starting this document with a blank slate. You will note that worklog18 encompassed all of my time since being fired by Buzzsumo back in the late summer of 2016, the last time I had a job where I was getting paid money. A lot of time has passed, and here we are. Started the day with getting nginx up and running on my DO, Digital Ocean droplet.

References

worklog18