

[Pull Requests on Nushell](#)  
[Link to this doc](#)

Friday March 05, 2021

The last couple of days I have been working on scope; and today I nailed down how to startup / init nushell by sourcing an alias file located in my nuconfig repo.

Thursday Feb 25, 2021 [Sharon arrives tomorrow]

I am loving [stylishthemes/github-dark](#)

Two more commits to nushell in the past week; so I am cranking along on making progress.

Sunday Feb 14, 2021

Day one of Bevy, I got it up and running and the examples actually work out of the box, everyone I tried actually works which is very impressive. Don't worry; I will attempt to spend a small amount of time looking into this for now.

Day 1 of delta and tickrs too.

Saturday Feb 13, 2021

Starting to do more work on understanding lex and parse\_block which used to be called block and I renamed to parse\_block which was the content of PR #3047.

Friday Feb 12, 2021

Got home from Ashland and Kfalls 2 days ago on Wednesday and on Thursday I did [PR #3047](#). This morning Jonathan had fixed all

of the clippy warnings that got thrown post release of Rust 1.50. So I was able to merge main and Jonathan posted my PR.

Sunday Jan 31, 2021

After a week of work on #2972 I can move on for now. See the end of the issue and the discord conversation for more details. The bottom line is @lhkipp came up with a simple solution to fix the lexer; and it probably is the most elegant simple solution. But I don't feel I am up to the paygrade yet to go around fixing the lexer at this moment. Hopefully someone else will step up and do this work, or I will feel more confident to do this work at a later point in time.

Friday Jan 29, 2021

I have been working all week on [Issue 2972](#). It is making me understand a lot more about the lexer and the parser than I ever thought I would have to know. "The universe continues..." This is a good project... Also this week on [Hacker News](#) I discovered an article where it states how long and how much work it takes to acquire the knowledge to better understand a big complex piece of software.

Friday Jan 22, 2021

Yesterday I started working in earnest on nushell scripts. Darren has been working for awhile on this stuff and he has a nice repo with scripts in it; so I am using that as a model in which to move forward.

[Jonathan's Big Todo List](#)

Friday Jan 15

So now that the nu-cli refactor is pretty much done; and I have taken a few days off on any hard core work; I am starting to think about what my next step in nushell is going to be...

As an aside, this is the beauty of what you are doing. You get to decide and get excited about what you are going to work on and work on next. So it looks like my focus could possibly be on the state in nushell, the context, the environment, and how all of that stuff is tied together and works in the bigger picture. This is a logical next step for me, and will get me more into the programming aspects of things as opposed to the mechanics of moving a bunch of code around.

This will also be related to the work on Jonathan's todo list of getting a new command going which....

*maybe: would it be interesting to add the ability to query the context for what commands, variables, and aliases are in scope?*

Monday Jan 11

Huge day. Too much. It ended with Jonathan taking over because one of the tests on windows was failing and he was able to take the ball over the line. I was talking with Dad at 5pm after working all day since literally 7 this morning on getting the nu-core-commands crate up and running.

Sunday January 10, 2021

Will post when commit #5 lands, its now in the queue.

Monday January 4, 2021

Cranking along on nushell with commit #4 landing...

My first commit 2828 was on Dec 28, 2020

# 2021

So I end the year on this note about my first day of programming in Rust which I noted again on Monday Oct 26 was on Scott's birthday 2019.

Wednesday Dec 30, 2020

One more commit lands, and one appears to be ready to land. I am on a roll and I am finally having fun programming in Rust after a very long while.

Monday Dec 28, 2020

I awoke this morning to a great piece of news...

My [first commit 2828 on Dec 28](#) into nushell was accepted by Jonathan. We are up and running at work.

Saturday Dec 26, 2020

Day 2 of Jonathan giving me the go ahead to refactor nu-cli. This is going to be a very interesting project, and challenging. Good luck.

Tuesday Dec 22, 2020

Starting to work on my own cli that calls into the nushell api parser et al. Finally starting to understand this stuff; as I am now writing code that calls into nushell and I am better able to understand what dependencies I need in Cargo.toml.

Sunday Dec 20, 2020

Continue down Jonathan's update path. Lets get this stuff now, and get it documented and move on past this initial phase. Put it out there so others will understand too.

Thursday Dec 17

More work on Jonathan's massive update as we move toward multiline scripts in nushell.

Friday Dec 11

Got bluemesa back up and running and added args to getfun. also starting to relook at parsing html via `pandas.read_html` which is used by `yahoo_fin`. Yesterday I tried parsing the yahoo finance quote page with rust but that was a complete failure so I am back to looking into parsing with pandas. It has 2 rock solid parsers including beautifulsoup which is also used to parse the edgar filings for xml / xbrl.

Tuesday Dec 8

As I was looking through github for hackernews repos I found [hn](#) which is a command line hn client. In there they use scraper which grabs the hn top posts. Day 1 of rust programs with the scraper technology.

Monday Dec 7

Starting to take a look at Jonathan's fork of nushell and the merging of nunu back into nushell. Last evening he told me where the fork was located. Also got hackernews favorites up and running with a fresh cut of all of the favorites through today.

Friday Dec 4, 2020

I am up to speed on nunu and jonathan is starting to merge it into nushell. rlox is in good shape now too. I want to switch gears now and re-focus my efforts on storing data in redis and sled and be up to speed on how to do it. So I am going to hold off on redis for awhile and just focus on sled. We will come back to redis later, but for now sled should get the job done, (I think)...

Saturday Nov 21, my 60th birthday

Found another implementation of lox called rlox by radogost. This is very helpful for understanding how to write lang code in rust. Keep going; as this is very helpful.

Friday Nov 20, 2020

Last night I sent in a PR for rulox and Mario responded back within a few hours. I awoke this morning and already had an email from him. I am doing a deep dive into programming languages and a much better understanding of how they work on the inside. [Crafting Interpreters](#) is a marvelous read and will be my seminal introduction (once again) to this aspect of computer science. Bill Macready would be happy to know I am here. My path in my work is all related and clearly this fork on the trail is related to my work the past number of months with nushell. We are at a point in the product where building a correct scripting language is required and so --- I need to understand this stuff. I love my job !

Tuesday Nov 10

Back into Bluemesa specifically grabbing more data from Edgar. Using ubnt as my model data set.

Wednesday Nov 04

Spent some more time on sled. I believe I am going to move back into bluemesa work to get a grip on how to push all 3 platforms forward: {bluemesa, nushell and sled}.

Sunday Nov 01

The corvallisoddfellows.org domain name is transferred over to us from Wes in Hawaii, and the web site is up and running on gh-pages. Congrats, this is a big win for me and a testament to

the ease of use of mdbook, which is rock solid.

Saturday Oct 31

I think I now finally am starting to understand the whole type system and what it is all about for nushell. I think the light went on when I started thinking about html template engines in the context of zola as well as what is going on with serde json and how the output that you see whether it be json, a nushell table, or an html page is moving the data through the Value to what the end user sees. More on this later.

Monday Oct 26

So I went back the other day and found out that the first day I started programming in Rust was on Scott's birthday 2019. It has now been well over a year and I consider myself a bit more than a beginner, but at least I am still focused on the language, and becoming more knowledgable. Today I came back in once again to review Chapter 13 of the Rust Book on closures and iterators. They are all over the nushell code and so I need to fully understand the concepts.

Sunday Oct 25

I learned today that mdbook allows us to copy the complete theme directory over to your local mdbook repo where you are creating the book and then it will use those themes instead of the default ones in the binary. This means that I don't need to have my own custom copy of mdbook lying around but can simply embed the theme files I need for each particular mdbook repo that I create. So even though the odd fellow mdbook needs some changes, another mdbook may want other ones.

Saturday Oct 24

Worked on a website for Bonnie and Oddfellows and had some

minor breakthroughs. First I decided to use mdBook and was able to fix the sidebar so it stayed open by default and reduced the sidebar from 300px to 200px giving the user a lot more real estate on the right side. I like mdBook and will continue to use it more and more in other projects; plus it is written in Rust !

Thursday Oct 22

Going through the async/await book in reference to the futures trait and the stream trait in nushell and all of the calls associated with the futures crate. So basically we are doing a deep dive into everything behind futures in relation to nushell. This is actually the first time I have revisited this concept since working with Tokio awhile back and the first time looking at it since I started working on nushell back in July 2020. Its amazing how all of this stuff doesn't really stick around in my brain until I relearn it or re-remember it again.

Sunday Oct 18

Finally realized that the tests in nu-cli need a binary to run and that is why main.rs gets sucked into the mix. It was not obvious till now that the nu! macro calls into the nu binary, upon thinking about it makes sense.

Friday Oct 16

```
nupro value/evaluate.rs
nucli  src/evaluate/mod.rs
evaluator::evaluate_baseline_expr
```

Wednesday Oct 14, 2020

Spent all day on a bug in nushell where tests fail unless config.toml is empty or the skip welcome message is not set. not sure today was productive, but I am very happy as all my tests are passing... and I know why !! it had to do with **config.toml**



Monday Oct 12, 2020

Deep dive into the underlying commands and how free data gets parse in nushell including all of the core commands {parse, each, split, etc...}

Sunday Oct 11, 2020

Working on Value as well as regex associated with the nushell parse command.

Saturday Oct 3, 2020

More work on exactly what a value is in Rust.

Thursday Oct 1, 2020

Massive dive into true understanding of Literals, Expressions, and Values in both the context of nushell and Rust.

Monday Sep 28

Luke's birthday. flexi\_logger is up and running inside nushell and logging all log messages either to a file or to the console. It works exactly like I wanted it to. Once I figured out flexi\_logger this afternoon I had the whole main.rs nushell file pared down and back up and running with the new logging code in about 1 hour. Very cool ! Rust logging is working the way it should, and nushell is very well designed from a logging perspective... Once you understand this stuff it is really trivial and easy. Today was an example of how things fall into place. However, it took me a number of days to finally get here and come up with this simple solution.

Friday Sep 25

A week has passed. We are now deep into how Rust logging works. I want to integrate a new logging solution into the Nushell

code for personal use. For now, the focus is as simple as; writing the log data out to file instead of stdout. (I think)...

Friday Sep 18

Moving quickly, I now understand that nu-table is the crate that draws the table data. I now need to work out exactly how this data gets fed to draw\_table and when a table is drawn and when one is not... As always, more work to do. But I have gotten past the parser for now, and moved on to the other side of the equation. There is a lot more to understand with nushell; but I think we are making progress. Stay tuned here for more details :)

Wed Sep 16

still smoky during the day but by evening it cleared and we went for a big walk in WL. still working on lite\_parse, with a major deep dive into all aspects of how **lite\_parse** in nushell works.

Tuesday Sep 15

Day-1 working on the nushell-notes Day 1 of mdbook. This is the start of my adventure getting this book up and running.

Monday Sep 14

this evening we went for a walk at night and it was the first time since last Monday that the skies were clear. Its been a crappy week as far as smoke is concerned. I am way deep into understanding the crates {nu-parser, nu-source}... I assume this should go on at least another week or two, or until I fully understand these 2 crates plus {nu-protocol}.

Monday Sep 07

jturner and I had our first talk about nushell. He is a cool guy. This morning I read the RFC on dataframes and made some

comments on discord, and soon we were discussing the spec he wrote up. I am going to spend more time looking at streaming based on our talk today.

Friday Sep 04

Listened to most of this [podcast](#) which was shot in September 2019 of the 3 founders of nushell. Today was day 1 of looking into the plugins. It was motivated by this [chat](#) I had with jturner on 9/3 which was yesterday in the documentation thread. I was trying to understand why sys wasn't working and it was because I was copying the nu binary over to another directory that did not have the plugins in there. Once I fired up the /target/debug nu version everything worked and I pulled in all of the default plugins. Then today I got the sqlite plugin working and started to work on understanding how the underlying plugin architecture using jsonrpc works.

Wednesday Sep 02

Massive dive into nushell ! Its all I have been doing since the last entry on August 24. Yesterday I discovered the github repo [nushell/demo](#) which was great to find because for the past week I had been seeing the method **parse and eval** inside nu\_cli but it was not referenced anywhere in the whole nushell repo.

Whenever I did a rg on it, nothing came up that was using it. And then yesterday I discovered the wasm demo repo by reading over the old nushell blog posts. Wasm was introduced in the [17.0 version blogpost](#). So tonight, I got run\_nu up and running with a simple example which is currently in my nuwasm repo which will probably eventually be moved somewhere else but for now thats its landing spot.

Monday August 24

Yesterday I started into the nushell discord which is giving me a

big insight as to how nushell works. Today I downloaded the first version just to get a glimpse of the original ideas behind the program. In concert with this info, I was able to convert the ui xbrl 10q file into a table and then back into json and the whole thing worked. This is amazing ! Also spent some time looking at toml to get a handle on their very simple, very intuitive format.

Thursday August 20

Deep in the throes of Nushell, and making HUGE progress on understanding the inner workings of it. Hopefully this will be relevant going forward. Stay tuned.

Monday August 17

Pedro's bday. Today I launched in a big way back into nushell. So I will attempt to work on 2 projects at the same time which could be a bit tricky. Stay tuned.

Friday August 14

Contextref in soup-xbrl is ready to be written out to a dictionary. Then I will do the other tags. Idea is to have a complete dictionary representation of the xbrl 10q / 10k file and then write that out to whatever including json, csv, redis, sqlite etc... We are making massive progress on this front.

Tuesday August 11

Starting to break out a new class called ContextRef. Will document here more what it does.

Wednesday August 5

Decided today to go away next Monday for a week or so. I learned today that our shed will not be delivered for several weeks so next Monday is a good day to go away. I now fully

understand data\_processing and passing in the context\_ids.

So here is whats on tap when I get back from my trip. I may update this further prior to departure but here goes for now.

So I am just at the point where I am understanding how to grab all of the correct context and associated context\_ids that I need to get certain data points and data sets. I will work out exactly what data points in the 10Q's are associated with what particular contexts but this for now is where all of the work will be going upon return. Capturing all of the data is possibly a function of getting the correct 3 month intervals associated contexts. That is step 1. Getting all of the associated quarterly data.

Once this is done then you can move on to getting the annual data. And lastly the 2 quarters / 6 months and 3 quarters / 9 months worth of data.

But for now just getting all of the 3 month data is not obvious as the current 2 context id's that I am pulling does appear to get all of the data and I also don't know out of all of the data points on the 10q what data points are 3 month data points and what data points are other contextrefs.

Monday August 3

More parsing of the edgar xbrl docs. I now have the name of the tag along with the value and starting to work on the child tags and grabbing the context data.

Friday July 31

Day 2 of working on xbrl. I now understand the work I need to

do to move forward. Be able to get the date and the context id's automagically from the file name and a particular xml element.

Thursday July 30

### **Day One of Parsing XBRL Files**

Lots of work on parsing XBRL and winnowing down the parsing for the quarterly UI files. We are up and running and getting a small subset of GAAP and cleaning up the xbrl parser file with just the bare essentials.

Tuesday July 28

Got all of the ubnt data going back to the beginning of time from their website and archived it in numerous locations. Now I can dive in further and make sure I have all of the data. This is a great model to better understand cash flow and the other interesting notes of a much bigger picture. Also calculated the avg share price of UI's purchases since the beginning of time. They spent just over 2B to purchase almost 28M shares at an average price of \$75. I used the cash flow statement to see how much stock they bought back each year starting with year 2012.

Monday July 27

Major, massive push to a better understanding of the cashflow statement. In fact I got so excited, that I put together a google sheet for the first time for [ubnt's 2020 cash flow](#) statements. I copied over all of the numbers from the 10Q's and reconciled that the numbers were exactly correct.

I have been wanting to do this for years, and get a better hold

and understanding of how all of these numbers flow. I think my conversation yesterday with Lukas got me to do this today.

The main thing I wanted to see was how all of the numbers literally add up, and now I can prove to myself exactly what the heck is going on with the cash flow statement.

Sunday July 26

First contact with Lukas at Causal after reading their blog post [Financial Statements: A Beginner's Guide](#).

Thursday July 23

Starting to get a handle on nushell. Python regex up and running removing {M,B,%,}\$ from fun and mcap numbers so that nushell sorts correctly. It doesn't like to sort with other characters in the numbers. Yesterday I got a handle on how to search and go back in time on discord. Now I can go back to the beginning of the nushell history and get a better handle on how things work.

Tuesday July 21

So I have been pretty busy working on some different things. First of all bluemesa is in a good spot. The next thing I need to do is start sorting all of the data inside the CSV. I got the mcap fundamental data from Fidelity now pretty much up and running.

What I did was download all of the stocks whose market cap is greater than > 1B and broken it up into categories where the largest one is > 90B and on down. Then I got all of those symbols sorted out and accompanying names of the companies in the redis key symbol hash. Then from there I was able to create

a CSV file from the Yahoo Fun data. And then I hit the road block which sent me down my 2nd rabbit hole.

Nushell only sorts integer columns but not floats. So I dove into nushell to see what was up and came across an issue with running

```
cargo run src/main.rs
```

I found a solution but not sure of what the cause of the problem was. In the meantime, I went back to take a 2nd look at alacritty to make sure I understand what that is. Its a terminal emulator that brings up a new window. Clearly, nothing to do with shells but it took me a bit of time to remember what a terminal emulator is...

Sunday July 12, 2020

Day one of nushell. this may not be the exact first day, but it is close enough to this day, to revise history and note that my first day of nushell was exactly one year after my first mention in my blog of rust, or more exactly day 1 of rust. so nushell came one year to the day of rust.

Thursday July 9

python-xbrl-test up and running. starting to look at xbrl and the mapping between [fundamental accounting concepts and gaap](#).

Friday July 3, 2020

Now that we have symbols in place for the Russell 3000 in the form of IWV we can now venture off to start to use these symbols to get fundamental data via Project Edgar. Step 1 was to have the ability to download the Edgar files and then parse the XBRL.



That is where the beautifulsoup comes in which is what I am currently working on.

Wednesday July 1

Got iwv symbols integrated into the blumesa system. so now we have a lot more symbol data to play with...

Tuesday June 30

In search of symbols and I found them from [ishares IWV](#). This has just under 3k symbols which is about 90% of the US equity markets so this will be enough symbols to populate my entire symbol db and then I will add any additional symbols as needed but this bootstraps my symbol library / db. In the afternoon I learned more about spreadsheets and how to reference one cell in another cell. I have been trying to understand how to do this for a couple of sessions and today I figured it out, very cool. Spreadsheets are really powerful and it has been just the past few years that I have learned better how to use them. Yesterday I spoke to Eli Scavron for awhile, its been many months since we last spoke.

Friday June 19

We learned today about the Juneteenth holiday for the first time. Google did a doodle on it. I also figured out how Python modules and packages work. The key breakthrough was adding the `/j/tmp32/blumesa/py` to the end of the PYTHONPATH. I am really truly loving Python. When I was programming in the language back at Caltech I did not have the background or knowledge of really understanding Javascript, Ruby, Go and Rust and then coming back at Python for a 2nd time gave me a lot

more "dry powder" to understand it. I am already a very happy and productive Python programmer and am cranking out code fairly fast for bluemesa.

Monday June 15

**This was my first commit to bluemesa.** We are starting to get up and running with whole new repo about "equity analysis". Some of the tools will be in Python and some of the tools will also be written in Rust. We are off to a good start.

Friday June 12

I can not believe how easy it is to code in Python. I am blown away by the elegance and simplicity of the language. Compared to Rust Python is a breeze. And its also way simpler than Javascript tambien.

Wed June 10

Up and running with Alacritty. Eventually this and the terminal work could be possibly integrated together. A better understanding of screen buffers, terminal emulators and the knowledge surrounding this subject.

Monday June 8

Starting to look at ben ashford's redis client that uses a very nice design of the RESP protocol.

Sunday June 7

Starting to look at Rust tui's (terminal UIs).

Saturday June 6, 2020

Checked out the rust redox-os.org OS. Also going to spend lots more time building and designing terminal UIs in rust and use the python [rich](#) version as one of my models.

Friday June 05, 2020

Up and running with Python, Go, and Rust grpc server and client along with a better understanding of protocol buffers. Earlier in the week I worked on Rust serde and a better understanding of Rust traits. Lots more work needs to be done on getting more up to speed on the advanced concepts in Rust.

Friday May 22

Watched the [Youtube video](#) by Wes McKinney author of pandas and arrow

Thursday May 21

Diana Martin died today as I turned 59.5. Also, starting to take a look at Apache Arrow and Parquet which is mentioned by Paul Dix in a [blog post](#).

Wednesday May 20

Back into Rust with a look at protobuf and grpc in the context of Hyperium and Lucio Franco's tonic. I spent all day kind of slashing around looking for the "right" first package solution that I could sink my teeth into. Of course, I need grpc for my sp500 project, but in and of itself it looks to be a very interesting area

to focus on for awhile. Especially in concert with using Python. The combination of these two languages may turn out to be a match made in heaven !:)

Wednesday May 13

Wrote my first "real" Python program [here](#). It took in the sp500 list of quotes for each industry group and created a json file with just the industry group symbol and the set of symbols in that group. I like Python alot, and I am impressed by how simple of a language it is... Coming back to it really for the first time since programming in it at Caltech for the first version of crdata. The 2nd version was re-written in Ruby.

Tuesday May 05

Ta is completely up and running, it just all works, and I fully understand now how to use the api. Getting back into python was a brilliant and lucky move on my part especially in the area of finance. And the whole motivation came initially from my work in influxdb and flux and realizing that instead of doing my processing in flux which is not the right way to go, was do possibly explore Python. Turns out to be a lucky and good choice. Influxdb and Paul Dix was also the motivator for me getting into Rust initially when I found out and saw that the Flux parser was written in Rust mainly for compatibility with wasm. Its a very cool world on the technology front.

Monday May 04

Reading ui.csv files into ta and running the algorithms on strategies that just take in close as an input parameter.

Friday May 01, 2020

As I started getting into my first Python package ta I realized I needed to review Python modules, packages, and classes.

Thursday 30

Begin spending more time looking at [ta](#).

Wednesday 29

Starting to spend more time looking at pandas and matplotlib.

Monday April 27

Deep dive into numpy, wow ! And to think I was using this at caltech in the good old days and I am back here again literally more than 10 years later like no time has passed. all of this is (of course) in the context of looking at flux versus other ways to process and compute on the time series equity data. So using flux makes no sense for computation, flux only needs to be used for getting data in and getting data out of influx and literally nothing else. numpy and pandas makes life so easy, and I am just starting to use it.

Saturday April 25

Revisited python virtual envs and started using pipx which seems to do the trick for me.

Thursday April 23, 2020

E's 26th Bday. Summary for the week so far... I think I now understand wasmtime and what a wasm standalone runtime is. Also, ran a bunch of python tests on different technical analysis software written in rust.

Monday April 20, 2020

Day 1 of Python. Day 1 of Catalina Version 10.15.4. What a mess, they changed the shell to zsh and made the root directory read only which means you have to run a command everytime you reboot which is a pretty big hassle in my view. I got Python up and running and some nice tests running which talks to Influxdb. So the Influxdb Python client is starting to get up and running and it appears to be pretty nice. The idea is to do most of the processing in a language like Rust or Python and do minimal processing in Flux for now. I learned yesterday that all of the flux code has to be in one file because user packages are not yet working.

Sunday April 19, 2020

This was a big week of Flux and a better understanding of the language. I believe I am back to where I was last July 2019. So here is my list of stuff going forward in no particular order.

The thesis of this list is Rust programming...

- get back into tantivy with a focus on the client
- work on rust client for influxdb 2.0
- work on RSI (relative strength index)
- both in rust and go
- eventually port the go version into influx
- <https://github.com/greyblake/ta-rs>
- <https://github.com/markcheno/go-talib>
- <https://github.com/markcheno/go-quote>

Monday April 13

Diving deeper into Influx. Yesterday I had a conversation with Paul Dix about the piece he wrote recently about a tutorial he

wrote on new upcoming features in Flux. I am spending more time once again understanding Flux and the pieces that surround reading data out of influxdb and into the Flux engine. All of that stuff comes from [from](#).

Also huge breakthrough in understanding of the fact that I can

```
go run cmd/influxd/main.go  
go build cmd/influxd/main.go
```

All day I was trying to rip out src code and understand how UI is related so that I can remove stuff or pare it down and in the end this is all I had to do... You do get a 500 Internal Server error but how all of this stuff is tied to the go build process is for future research, for now the fact that I can quickly

**build an influxd binary**

without going through the whole UI yarn cluster-f--k of building the complete React UI system is a HUGE win.

Monday April 06

More progress... I got the whole flux script up and running today. I am actually able to see results in both the influx UI as well as running flux script via 'influx query @ex00.txt'. I am impressed, this stuff actually works really well and I accidentally discovered the simple solution to the 200 day moving average and it matches up with the result on yahoo; very cool. Its just the mean calculation using the past 200 days. What a life !

Friday April 03

I am happy to say after several weeks of hard work that I am finally reading a yahoo historical equity csv file and writing out the data in [influxdb line protocol format](#). Here is the [influx-point-lineprotocol](#) repo. The hard part the past 2 weeks

was a better understanding of rust structs along with the ownership, borrowing and especially the reference concept in Chapter 4 of the Rust Book.

Wednesday April 01

Reading the ownership chapter in the Rust book, Chapter 4 and this time I believe its starting to sink in better. Especially now since I really needed to understand structs and how they work in regard to the concept of a Reference in Rust. I think I am finally starting to get the point. Boy does this stuff take a while to get.

Monday March 30

Big time deep dive into structs in Rust including borrowing and ownership. I need to understand this crap.

Tuesday March 24

First day of working getting Point up and running inside the src tree with a real crate and real structs with methods all being called from example. This is my first time ever really starting to understand this concept in Rust. Its amazing how long I have been working in this language to finally get to this place. This stuff takes a lot of time.

Friday March 20, 2020

So I spent the day looking at the Point implementation in Influxdb along with the rust client for 1.7x, along with the line-protocol for 1.7x. I am going to do an array of structs with {measurement, time, fieldset and tagset}

Tuesday March 17, 2020

First real day of coding back up and running with Rust. Read equity csv files and writing the data to an InfluxDb file format.



Friday March 6

Final check in prior to heading down to Ashland. Lots of work on Influxdb since last update. Tons more to do... Not sure this is the right path, but I am heading down it...

Friday Feb 28

Start working again on understanding flux.

Thursday Feb 27

Finished up initial work on understanding influxd generate along with pkg/data/gen and how it melds random numbers into the spec which gets generated via the schema.toml file.

Friday Feb 21

Toml is the key to understanding schemas in influx, so we are going through a review of the toml spec.

Tuesday Feb 18, 2020

Got the influxdb models package up and running standalone and sent in my first PR to make the models package independent of tsdb. Discussed with [Jonathan Sternberg on Slack today](#)...

Thursday Feb 13, 2020

So I got the xbrl xml ubnt file parsed into json and beyond via Javascript. I think now it might be easy to do in Rust as well, see my repo **parse-xbrl-test** for more details. I now believe that I will be both a better Rust and Go programmer because coming back to Go now I am better able to understand what is going on. Hit upon the influxdata/line-protocol github repo and realized how simple the Metric is and how they built a HUGE idea and company

and product out of this very simple line-protocol Metric data structure.

Tuesday Feb 11, 2020

Back into influxdb possibly... Got it up and running on my machine. Might continue exploring time series data for ubnt and fundamental and technical mix signals.

Monday Feb 10, 2020

Deep dive into bayard for the first time. My initial hit is that I am very impressed. Start looking at how to write a javascript client so you can embed a search bar and talk to bayard in javascript instead of using rust and wasm ? The idea would be to use grpc.

Monday Feb 3, 2020

Get tantivy-cli up and running on 11.3 based on a better understanding of Iterators, maps, filters and collect.

Monday Jan 27, 2020

Finally after months of work I now am reading a file of hackernews lines in json that contain an ID and a TITLE and writing those to a tantivy index and then searching across that data. Getting the data into redis properly and pre and post processing that stuff took probably more work than it should have but when one is learning a new programming language everything takes longer. Hopefully I will stick around Rust for awhile.

Sunday Jan 05, 2020

Breakthrough on understanding that the fastfield code is just one of the SegmentComponents and that each individual segment component has its own serializer, file format, and reader / writer which are independent of each other.

Friday Jan 03, 2020

More work on the directory package in Tantivy including the general concept behind mmap...

## 2020

*So I am only keeping part of 2019 to keep 2020 in context to what I was working on in 2019. I go back to my start of tantivy which goes back to Maria's birthday on Oct 22, 2019...*

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 burntstushi 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, 2019

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.