Tracing Integration in Lighthouse

Sayan

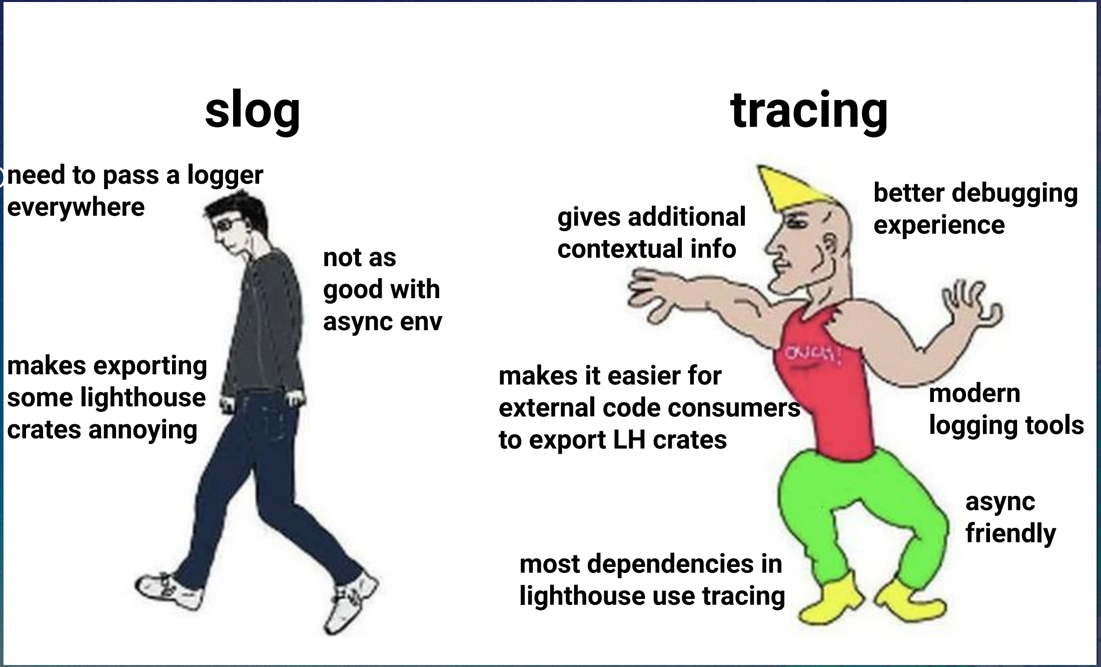
protocol fellow, EPF

Section 1

Background

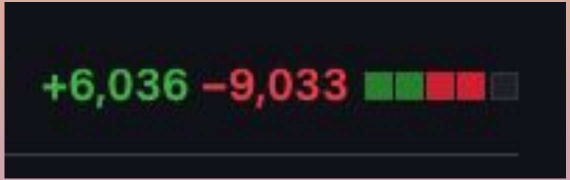
Background

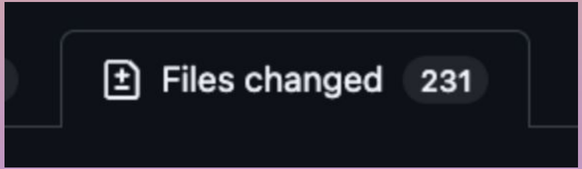
- What's `tracing` ?
  - A cool async-friendly structured logging framewo
- Why did we want to migrate Lighthouse from `slog` to `tracing` though?

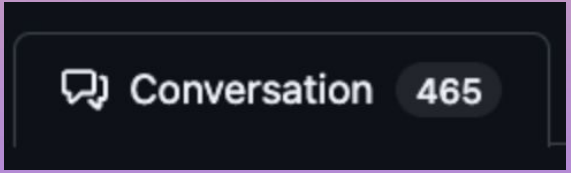

slog
need to pass a logger everywhere
not as good with async env
makes exporting some lighthouse crates annoying
tracing
gives additional contextual info
better debugging experience
OUCH!
makes it easier for external code consumers to export LH crates
modern logging tools
async friendly
most dependencies in lighthouse use tracing

so,what have I done so far?

- https://github.com/sigp/lighthouse/pull/6339
- Migrated the entire codebase from `slog` to `tracing`
- Re-implemented the custom-formatter
- Re-implemented the SSE Logging component
- File Logging

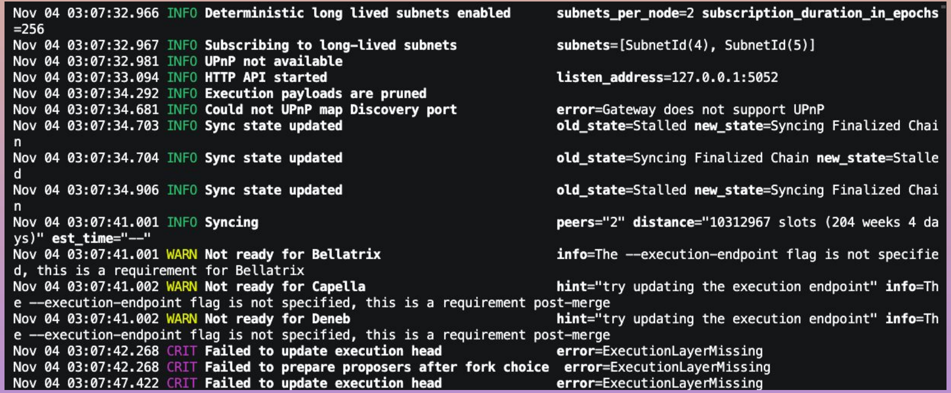






so,what have I done so far?

- Migrated the entire codebase from `slog` to `tracing`
- Re-implemented the custom-formatter
- Re-implemented the SSE Logging component
- File Logging
- https://github.com/sigp/lighthouse/pull/6339



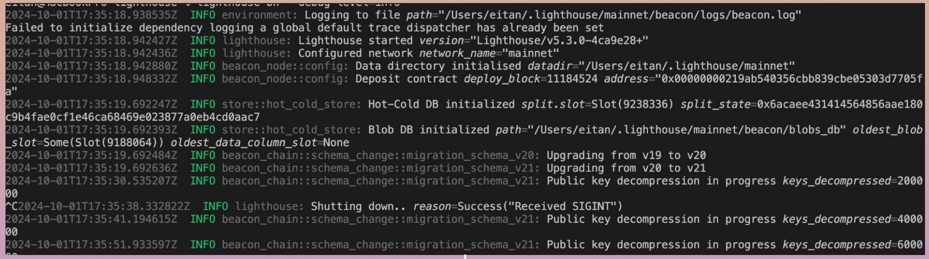
```
Nov 04 03:07:32.966 INFO Deterministic long lived subnets enabled      subnets_per_node=2 subscription_duration_in_epochs
=256
Nov 04 03:07:32.967 INFO Subscribing to long-lived subnets             subnets=[SubnetId(4), SubnetId(5)]
Nov 04 03:07:32.981 INFO UPnP not available
Nov 04 03:07:33.094 INFO HTTP API started                              listen_address=127.0.0.1:5052
Nov 04 03:07:34.292 INFO Execution payloads are pruned
Nov 04 03:07:34.681 INFO Could not UPnP map Discovery port             error=Gateway does not support UPnP
Nov 04 03:07:34.703 INFO Sync state updated                            old_state=Stalled new_state=Syncing Finalized Chai
n
Nov 04 03:07:34.704 INFO Sync state updated                            old_state=Syncing Finalized Chain new_state=Stalle
d
Nov 04 03:07:34.906 INFO Sync state updated                            old_state=Stalled new_state=Syncing Finalized Chai
n
Nov 04 03:07:41.001 INFO Syncing                                       peers="2" distance="10312967 slots (204 weeks 4 da
ys)" est_time="--"
Nov 04 03:07:41.001 WARN Not ready for Bellatrix                       info=The --execution-endpoint flag is not specifie
d, this is a requirement for Bellatrix
Nov 04 03:07:41.002 WARN Not ready for Capella                         hint="try updating the execution endpoint" info=Th
e --execution-endpoint flag is not specified, this is a requirement post-merge
Nov 04 03:07:41.002 WARN Not ready for Deneb                           hint="try updating the execution endpoint" info=Th
e --execution-endpoint flag is not specified, this is a requirement post-merge
Nov 04 03:07:42.268 CRIT Failed to update execution head               error=ExecutionLayerMissing
Nov 04 03:07:42.268 CRIT Failed to prepare proposers after fork choice  error=ExecutionLayerMissing
Nov 04 03:07:47.422 CRIT Failed to update execution head               error=ExecutionLayerMissing
```

Section 2

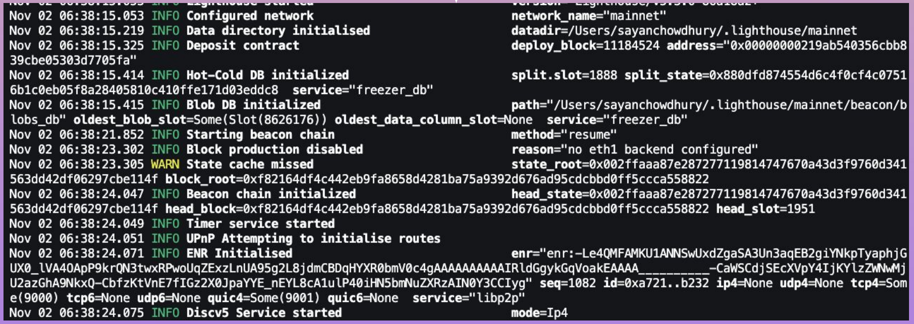
Challenges

some challenges & how we handled those

- **Keeping the format same:**
  - It was really important to keep the same exact formatting from slog to tracing, as internal infra and external users rely on that formatting for their staking setup



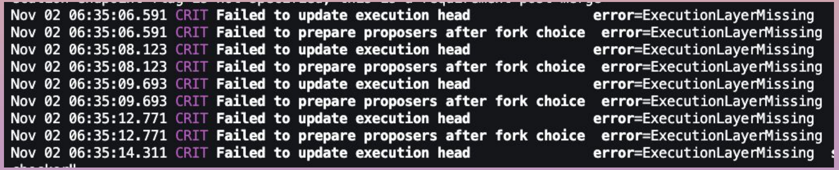
```
2024-10-01T17:35:18.938535Z  INFO environment: Logging to file path="/Users/eitan/.lighthouse/mainnet/beacon/logs/beacon.log"
Failed to initialize dependency logging a global default trace dispatcher has already been set
2024-10-01T17:35:18.942427Z  INFO lighthouse: Lighthouse started version="Lighthouse/v5.3.0-4ca9e28+"
2024-10-01T17:35:18.942436Z  INFO lighthouse: Configured network network_name="mainnet"
2024-10-01T17:35:18.942880Z  INFO beacon_node::config: Data directory initialised datadir="/Users/eitan/.lighthouse/mainnet"
2024-10-01T17:35:18.948332Z  INFO beacon_node::config: Deposit contract deploy_block=11184524 address="0x00000000219ab540356cbb839cbe05303d7705f
a"
2024-10-01T17:35:19.692247Z  INFO store::hot_cold_store: Hot-Cold DB initialized split.slot=Slot(9238336) split_state=0x6acaee431414564856aae180
c9b4fae0cf1e46ca68469e023877a0eb4cd0aac7
2024-10-01T17:35:19.692393Z  INFO store::hot_cold_store: Blob DB initialized path="/Users/eitan/.lighthouse/mainnet/beacon/blobs_db" oldest_blob
_slot=Some(Slot(9188064)) oldest_data_column_slot=None
2024-10-01T17:35:19.692484Z  INFO beacon_chain::schema_change::migration_schema_v20: Upgrading from v19 to v20
2024-10-01T17:35:19.692636Z  INFO beacon_chain::schema_change::migration_schema_v21: Upgrading from v20 to v21
2024-10-01T17:35:30.535207Z  INFO beacon_chain::schema_change::migration_schema_v21: Public key decompression in progress keys_decompressed=2000
00
^C2024-10-01T17:35:38.332822Z  INFO lighthouse: Shutting down.. reason=Success("Received SIGINT")
2024-10-01T17:35:41.194615Z  INFO beacon_chain::schema_change::migration_schema_v21: Public key decompression in progress keys_decompressed=4000
00
2024-10-01T17:35:51.933597Z  INFO beacon_chain::schema_change::migration_schema_v21: Public key decompression in progress keys_decompressed=6000
```



```
Nov 02 06:38:15.053 INFO Configured network                            network_name="mainnet"
Nov 02 06:38:15.219 INFO Data directory initialised                    datadir=/Users/sayanchowdhury/.lighthouse/mainnet
Nov 02 06:38:15.325 INFO Deposit contract                              deploy_block=11184524 address="0x00000000219ab540356cbb8
39cbe05303d7705fa"
Nov 02 06:38:15.414 INFO Hot-Cold DB initialized                       split.slot=1888 split_state=0x880dfd874554d6c4f0cf4c0751
6b1c0eb05f8a28405810c410ffe171d03eddc8  service="freezer_db"
Nov 02 06:38:15.415 INFO Blob DB initialized                           path="/Users/sayanchowdhury/.lighthouse/mainnet/beacon/b
lobs_db" oldest_blob_slot=Some(Slot(8626176)) oldest_data_column_slot=None  service="freezer_db"
Nov 02 06:38:21.852 INFO Starting beacon chain                         method="resume"
Nov 02 06:38:23.302 INFO Block production disabled                     reason="no eth1 backend configured"
Nov 02 06:38:23.305 WARN State cache missed                            state_root=0x002ffaaa87e287277119814747670a43d3f9760d341
563dd42df06297cbe114f block_root=0xf82164df4c442eb9fa8658d4281ba75a9392d676ad95cdcbbd0ff5ccca558822
Nov 02 06:38:24.047 INFO Beacon chain initialized                      head_state=0x002ffaaa87e287277119814747670a43d3f9760d341
563dd42df06297cbe114f head_block=0xf82164df4c442eb9fa8658d4281ba75a9392d676ad95cdcbbd0ff5ccca558822 head_slot=1951
Nov 02 06:38:24.049 INFO Timer service started
Nov 02 06:38:24.051 INFO UPnP Attempting to initialise routes
Nov 02 06:38:24.071 INFO ENR Initialised                               enr="enr:-Le4QMFAMKU1ANNSwUxdZgaSA3Un3aqEB2giYNkpTyaphjG
UX0_lVA4OApP9krQN3twxRPwoUqZExzLnUA95g2L8jdmCBDqHYXR0bmV0c4gAAAAAAAAAIRldGgykGqVoakEAAAA__________-CaWSCdjSEcXVpY4IjKYlzZWNwMj
U2azGhA9NkxQ-CbfzKtVnE7fIGz2X0JpaYYE_nEYL8cA1ulP40iHN5bmNuZXRzAIN0Y3CCIyg" seq=1082 id=0xa721..b232 ip4=None udp4=None tcp4=Som
e(9000) tcp6=None udp6=None quic4=Some(9001) quic6=None  service="libp2p"
Nov 02 06:38:24.075 INFO Discv5 Service started                        mode=Ip4
```

some challenges & how we handled those

- **Keeping the format same:**
    - tracing doesn’t provide `crit!`, so we had to build a custom implementation
        - which was basically a wrapper around `tracing`’s error macro



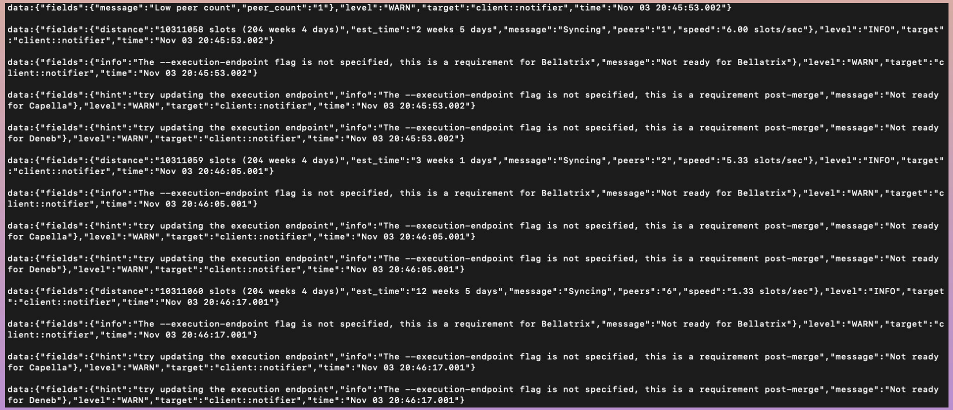
```
Nov 02 06:35:06.591 CRIT Failed to update execution head                error=ExecutionLayerMissing
Nov 02 06:35:06.591 CRIT Failed to prepare proposers after fork choice  error=ExecutionLayerMissing
Nov 02 06:35:08.123 CRIT Failed to update execution head                error=ExecutionLayerMissing
Nov 02 06:35:08.123 CRIT Failed to prepare proposers after fork choice  error=ExecutionLayerMissing
Nov 02 06:35:09.693 CRIT Failed to update execution head                error=ExecutionLayerMissing
Nov 02 06:35:09.693 CRIT Failed to prepare proposers after fork choice  error=ExecutionLayerMissing
Nov 02 06:35:12.771 CRIT Failed to update execution head                error=ExecutionLayerMissing
Nov 02 06:35:12.771 CRIT Failed to prepare proposers after fork choice  error=ExecutionLayerMissing
Nov 02 06:35:14.311 CRIT Failed to update execution head                error=ExecutionLayerMissing
```

some challenges & how we handled those

- **SSE Logging (with slog)**
    - With slog,the implementation was way too complex
        - They were using slog_async ,
            - it collects all the logs into a single task. And then task could then print and write to files asynchronously.
        - HTTP API was in a separate task. There was no way for the HTTP API to see any of the logs
        - They modified the slog_async task thing to create a channel.
            - The HTTP API optionally gets this channel, then the slog task could clone the logs and dump them into the channel.
        - For some reason, slog records that didn't allow sending the data through the channel
            - So they had to make the `AsyncRecord` thing

some challenges & how we handled those

- SSE Logging(with tracing)
    - It was super easy to implement with \`tracing\`
    - Had to remove that code totally first & then re-implement the SSE Logging component from scratch



```
data:{"fields":{"message":"Low peer count","peer_count":"1"},"level":"WARN","target":"client::notifier","time":"Nov 03 20:45:53.002"}

data:{"fields":{"distance":"10311058 slots (204 weeks 4 days)","est_time":"2 weeks 5 days","message":"Syncing","peers":"1","speed":"6.00 slots/sec"},"level":"INFO","target"
:"client::notifier","time":"Nov 03 20:45:53.002"}

data:{"fields":{"info":"The --execution-endpoint flag is not specified, this is a requirement for Bellatrix","message":"Not ready for Bellatrix"},"level":"WARN","target":"c
lient::notifier","time":"Nov 03 20:45:53.002"}

data:{"fields":{"hint":"try updating the execution endpoint","info":"The --execution-endpoint flag is not specified, this is a requirement post-merge","message":"Not ready
for Capella"},"level":"WARN","target":"client::notifier","time":"Nov 03 20:45:53.002"}

data:{"fields":{"hint":"try updating the execution endpoint","info":"The --execution-endpoint flag is not specified, this is a requirement post-merge","message":"Not ready
for Deneb"},"level":"WARN","target":"client::notifier","time":"Nov 03 20:45:53.002"}

data:{"fields":{"distance":"10311059 slots (204 weeks 4 days)","est_time":"3 weeks 1 days","message":"Syncing","peers":"2","speed":"5.33 slots/sec"},"level":"INFO","target"
:"client::notifier","time":"Nov 03 20:46:05.001"}

data:{"fields":{"info":"The --execution-endpoint flag is not specified, this is a requirement for Bellatrix","message":"Not ready for Bellatrix"},"level":"WARN","target":"c
lient::notifier","time":"Nov 03 20:46:05.001"}

data:{"fields":{"hint":"try updating the execution endpoint","info":"The --execution-endpoint flag is not specified, this is a requirement post-merge","message":"Not ready
for Capella"},"level":"WARN","target":"client::notifier","time":"Nov 03 20:46:05.001"}

data:{"fields":{"hint":"try updating the execution endpoint","info":"The --execution-endpoint flag is not specified, this is a requirement post-merge","message":"Not ready
for Deneb"},"level":"WARN","target":"client::notifier","time":"Nov 03 20:46:05.001"}

data:{"fields":{"distance":"10311060 slots (204 weeks 4 days)","est_time":"12 weeks 5 days","message":"Syncing","peers":"6","speed":"1.33 slots/sec"},"level":"INFO","target
":"client::notifier","time":"Nov 03 20:46:17.001"}

data:{"fields":{"info":"The --execution-endpoint flag is not specified, this is a requirement for Bellatrix","message":"Not ready for Bellatrix"},"level":"WARN","target":"c
lient::notifier","time":"Nov 03 20:46:17.001"}

data:{"fields":{"hint":"try updating the execution endpoint","info":"The --execution-endpoint flag is not specified, this is a requirement post-merge","message":"Not ready
for Capella"},"level":"WARN","target":"client::notifier","time":"Nov 03 20:46:17.001"}

data:{"fields":{"hint":"try updating the execution endpoint","info":"The --execution-endpoint flag is not specified, this is a requirement post-merge","message":"Not ready
for Deneb"},"level":"WARN","target":"client::notifier","time":"Nov 03 20:46:17.001"}
```

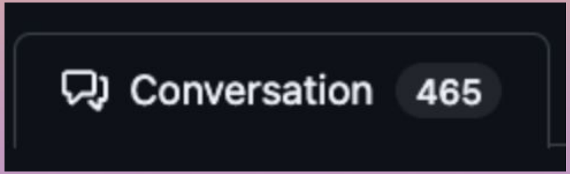
some challenges & how we handled those

- **spans vs slog::o! macro**
  - Both work differently
    - slog::o! attaches static metadata to loggers
      - good for adding consistent information across logs
    - Spans creates scoped, dynamic contexts tied to specific code blocks
      - good for tracking execution flow

- https://github.com/ThreeHrSleep/lighthouse/pull/1(for the slasher service)
- https://github.com/ThreeHrSleep/lighthouse/pull/2 (for the eth RPC)

some challenges & how we handled those

- **Misc.**
  - We thought about doing this change across multiple PRs, but breaking it up into chunks would've broken SSE logging.
  - Quite a bit of back & forth
  - Severe skill issues
    - I was really really bad at Rust



Section 3

Next steps

Next Steps

Adding more spans

Need to add more spans so that service names show up in the logs like before

More feedback-fix iterations

still lots of iterations left

More testing/QA

Need to make sure we introduce least amount of breaking changes

Flags to enable/disable additional context

For better debugging experience.

Section 4

Overall Experience

Overall Experience

Last five months were absolute blast

- Working with Lighthouse team was awesome
- Got better at Rust (arguably)
- Loved the Office hours/AMAs

eserilev left a comm

Great work so far!

macladson left a

This is nice work,

Huge Thanks to

- Age,Eitan,Mac
- Josh & Mário

Huge Thanks to

- Age,Eitan,Mac
- Josh & Mário

Thanks for sitting through :)

Sayan

github: @ThreeHrSleep

twitter: @sayan_011