

sector_state

April 27, 2021

1 Sector State Metrics Dashboard

1.1 Setting-up

```
[1]: %load_ext autotime  
  
%load_ext autoreload  
  
%autoreload 2
```

time: 8.55 ms (started: 2021-04-27 01:51:50 +00:00)

```
[2]: # External dependences  
import pandas as pd  
import numpy as np  
import plotly.express as px  
  
# Move path to parent folder  
import sys  
sys.path.insert(1, '../')  
  
# Internal dependences  
from filecoin_metrics.connection import get_connection, get_connection_string  
from filecoin_metrics.metrics import *
```

time: 476 ms (started: 2021-04-27 01:51:50 +00:00)

```
[3]: conn_string = get_connection_string('../config/sentinel-conn-string.txt')  
connection = get_connection(conn_string)
```

time: 1.18 s (started: 2021-04-27 01:51:50 +00:00)

1.2 Visualizations

Rate of missing PoST, weekly, network-wide

```
[4]: s = rate_missing_post_network_weekly(connection)  
px.bar(s, log_y=True)
```

time: 1min 49s (started: 2021-04-27 01:51:51 +00:00)

Fraction of active sectors with Missing PoST, weekly, network-wide

```
[5]: s = fraction_missing_post_network_weekly(connection)
px.bar(s)
```

time: 1min 25s (started: 2021-04-27 01:53:41 +00:00)

Rate of missing PoST, weekly, per miner `d = rate_missing_post_miner_weekly(connection)`

Declare Fault count per miner, top 10 `s = declare_fault_count_per_miner(connection)`
`print(s.sort_values(ascending=False).head(10))`

Declare Fault rate, weekly, network-wide

```
[6]: from filecoin_metrics.metrics import declare_fault_weekly

s = declare_fault_weekly(connection)

fig_df = s.reset_index()

fig = px.bar(fig_df,
             x='timestamp',
             y='declare_fault_count',
             title='Weekly Declare Fault Count',
             log_y=True)
fig.show()
```

time: 594 ms (started: 2021-04-27 01:55:07 +00:00)

Rate of early termination, weekly, network-wide

```
[ ]:
```

Average termination fee, weekly, network-wide

```
[ ]:
```

1.2.1 Renewal Events

```
[7]: from filecoin_metrics.metrics import renewal_count_per_epoch

s = renewal_count_per_epoch(connection)
INTERVAL = '1w'

s_count = (s.resample(INTERVAL)
           .sum()
           .backfill()
           )
```

```

s_cum = (s.cumsum()
         .resample(INTERVAL)
         .median()
         .backfill()
         )

s_cum.name = 'renewal_count_cumulative'

fig_df = (pd.DataFrame([s_count, s_cum])
         .T
         .reset_index()
         .melt(id_vars=['timestamp'])
         )

fig = px.bar(fig_df,
             x='timestamp',
             y='value',
             title='Renewal Events Count',
             facet_col='variable',
             log_y=True)

fig.show()

```

time: 22.4 s (started: 2021-04-27 01:55:07 +00:00)

Renewal Gap, weekly, per miner

[7]:

time: 13.9 s (started: 2021-02-17 17:12:59 -03:00)