sector_state_miners

April 11, 2021

1 Stats for the sectors, segmented by miner group

1.1 Setting-up

[1]:
```
%load_ext autotime

%load_ext autoreload

%autoreload 2
```

```
time: 9.03 ms (started: 2021-04-11 01:14:56 +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, '../')

import plotly
plotly.offline.init_notebook_mode()
```

```
time: 492 ms (started: 2021-04-11 01:14:56 +00:00)
```

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

```
time: 1.28 s (started: 2021-04-11 01:14:57 +00:00)
```

[4]:
```
UPGRADE_DATE = '2020-11-25 00:00:00'
```

```
time: 11.4 ms (started: 2021-04-11 01:14:58 +00:00)
```

[5]:
```
QUERY = """
/* Get the last state of the sectors */
```

```
with sector_states as (
        select
        msi.*,
        max(msi.height) over (partition by msi.sector_id, msi.miner_id) as␣
 ↪max_height
        from miner_sector_infos msi
        where msi.activation_epoch > 0
        and msi.expiration_epoch > msi.height /* Get only active sectors */
        order by max_height
)
select
ss.miner_id,
count(*) as sector_count,
sum(ss.initial_pledge::numeric) / 1e18 as initial_pledge_in_fil,
count(*) * 32 as network_power_in_gb,
date_trunc('DAY', to_timestamp(height_to_unix(ss.activation_epoch))) as␣
 ↪activation_date,
date_trunc('DAY', to_timestamp(height_to_unix(ss.expiration_epoch))) as␣
 ↪expiration_date
from sector_states as ss
where ss.max_height = ss.height /* get the last state of the info */
group by activation_date, expiration_date, miner_id
order by activation_date, expiration_date, miner_id
"""

query_df = (pd.read_sql(QUERY, connection)
              .assign(network_power_in_pib=lambda df: df.network_power_in_gb /␣
 ↪(1024 ** 2))
              .assign(initial_pledge_in_thousand_fil=lambda df: df.
 ↪initial_pledge_in_fil / 1000))
```

time: 12min 49s (started: 2021-04-11 01:14:58 +00:00)

[6]:
```
import json

path = '../config/miners.json'

with open(path, 'r') as fid:
    miners_map = json.load(fid)

miner_id_map = {miner_id: miner
                for miner, miner_ids
                in miners_map.items()
                for miner_id in miner_ids}

metrics = {'miner': query_df.miner_id.map(miner_id_map).fillna('None'),
           'is_v1': lambda x:  x['activation_date'] < UPGRADE_DATE}
```

```
query_df = query_df.assign(**metrics)
```

```
time: 65.9 ms (started: 2021-04-11 01:27:48 +00:00)
```

[7]:
```
def resample_and_bar_plot(df, resample_rule, time_column, value_column, title,␣
↪**kwargs):
    fig_df = df.groupby('miner').resample(resample_rule, on=time_column,␣
↪label='left').sum()
    fig = px.bar(fig_df.reset_index(),
                 x=time_column,
                 y=value_column,
                 color='miner',
                 title=title,
                 **kwargs)
    return fig


def resample_and_bar_plot_relative(df, resample_rule, time_column,␣
↪value_column, title, **kwargs):
    fig_df = df.groupby('miner').resample(resample_rule, on=time_column,␣
↪label='left').sum()
    y = fig_df.groupby(time_column).sum()
    fig_df /= y
    fig = px.bar(fig_df.reset_index(),
                 x=time_column,
                 y=value_column,
                 color='miner',
                 title=title,
                 **kwargs)
    return fig
```

```
time: 11.1 ms (started: 2021-04-11 01:27:48 +00:00)
```

1.2 Sector Count

[8]:
```
df = query_df.copy()

print("Basic stats")
print("---")
print(f"Total sectors (#): {df.sector_count.sum()}")
print(f"Raw bytes power (PiB): {df.network_power_in_gb.sum() / (1024 ** 2) :.
↪3g}")
print(f"Initial pledge (FIL): {df.initial_pledge_in_fil.sum()}")
print("---")
```

```
Basic stats
```

```
---
Total sectors (#): 106483186
Raw bytes power (PiB): 3.25e+03
Initial pledge (FIL): 34719962.147866756
---
time: 19.4 ms (started: 2021-04-11 01:27:48 +00:00)
```

[9]:
```
resample_rule = '1m'
time_column = 'expiration_date'
value_column = 'sector_count'
title = 'Count of Expiring  Sectors (#)'


groups = [pd.Grouper(key='expiration_date', freq=resample_rule),
          'is_v1']

fig_df = (df.groupby(groups)
            .sum()
         )


fig = px.bar(fig_df.reset_index(),
             x=time_column,
             y=value_column,
             color='is_v1',
             title=title)
fig.show()
```

```
time: 345 ms (started: 2021-04-11 01:27:48 +00:00)
```

[10]:
```
resample_rule = '1d'
time_column = 'expiration_date'
value_column = 'sector_count'
title = 'Count of Expiring  Sectors Before 15Jun2021 (#)'


groups = [pd.Grouper(key='expiration_date', freq=resample_rule),
          'is_v1']

fig_df = (df.query("expiration_date < '2021-06-15 00:00+00:00'")
            .groupby(groups)
            .sum()
            .reset_index()
         )


fig = px.bar(fig_df,
```

```
            x=time_column,
            y=value_column,
            color=fig_df.is_v1,
            title=title,
            log_y=True)
fig.show()
```

time: 93.3 ms (started: 2021-04-11 01:27:49 +00:00)

[11]:
```
resample_rule = '1d'
time_column = 'expiration_date'
value_column = 'sector_count'
title = 'Count of Expiring  Sectors Before 15Jun2021, grouped by Miner (#)'


groups = [pd.Grouper(key='expiration_date', freq=resample_rule),
          'is_v1',
          'miner']

fig_df = (df.query("expiration_date < '2021-06-15 00:00+00:00'")
            .groupby(groups)
            .sum()
            .reset_index()
         )


fig = px.bar(fig_df,
             x=time_column,
             y=value_column,
             color='miner',
             facet_col='is_v1',
             title=title)
fig.show()
```

time: 165 ms (started: 2021-04-11 01:27:49 +00:00)

[12]:
```
resample_rule = '1m'
time_column = 'expiration_date'
value_column = 'sector_count'
title = 'Upcoming Sector Expiration Count (#)'

resample_and_bar_plot(df,
                      resample_rule,
                      time_column,
                      value_column,
                      title).show()
```

time: 272 ms (started: 2021-04-11 01:27:49 +00:00)

[13]:
```
resample_rule = '1m'
time_column = 'expiration_date'
value_column = 'sector_count'
title = 'Upcoming Sector Expiration Count, grouped by sector version (#)'


groups = ['miner',
          pd.Grouper(key='expiration_date', freq=resample_rule),
          'is_v1']

fig_df = (df.groupby(groups)
            .sum()
            .reset_index()
         )

fig = px.bar(fig_df.reset_index(),
             x=time_column,
             y=value_column,
             color='miner',
             facet_col='is_v1',
             title=title)
fig.show()
```

```
time: 237 ms (started: 2021-04-11 01:27:49 +00:00)
```

[14]:
```
resample_rule = '1m'
time_column = 'expiration_date'
value_column = 'sector_count'
title = 'Percentage of Expiring V1 Sectors (%)'


groups = [pd.Grouper(key='expiration_date', freq=resample_rule),
          'is_v1']

fig_df = (df.groupby(groups)
            .sum()
            .reset_index(level='is_v1')
         )

v1_df = fig_df.query('is_v1 == True').fillna(0)
v2_df = fig_df.query('is_v1 == False').fillna(0)
fig_df = (v1_df / (v1_df + v2_df))

fig = px.bar(fig_df.reset_index(),
             x=time_column,
             y=value_column,
             title=title)
```

```
fig.show()
```

time: 167 ms (started: 2021-04-11 01:27:49 +00:00)

[15]:
```
resample_rule = '1m'
time_column = 'expiration_date'
value_column = 'sector_count'
title = 'Percentage of Expiring Old Sectors per Miner Group (%)'


groups = ['miner',
          pd.Grouper(key='expiration_date', freq=resample_rule),
          'is_v1']

fig_df = (df.groupby(groups)
            .sum()
            .reset_index(level='is_v1')
         )

v1_df = fig_df.query('is_v1 == True').fillna(0)
v2_df = fig_df.query('is_v1 == False').fillna(0)
fig_df = (v1_df / (v1_df + v2_df))

fig = px.bar(fig_df.reset_index(),
             x=time_column,
             y=value_column,
             color='miner',
             animation_frame='miner',
             title=title)
fig.show()
```

time: 225 ms (started: 2021-04-11 01:27:50 +00:00)

[16]:
```
resample_rule = '1m'
time_column = 'activation_date'
value_column = 'sector_count'
title = 'Count of Sector Activation Date (#)'


resample_and_bar_plot(df, resample_rule, time_column, value_column, title).
↪show()
```

time: 190 ms (started: 2021-04-11 01:27:50 +00:00)

[17]:
```
resample_rule = '1w'
time_column = 'activation_date'
value_column = 'sector_count'
```

```
title = 'Activated Sector Count, grouped by sector version (#)'


groups = ['miner',
          pd.Grouper(key='activation_date', freq=resample_rule),
          'is_v1']

fig_df = (df.groupby(groups)
            .sum()
            .reset_index()
         )

fig = px.bar(fig_df.reset_index(),
             x=time_column,
             y=value_column,
             color='miner',
             facet_col='is_v1',
             title=title)
fig.show()
```

```
time: 201 ms (started: 2021-04-11 01:27:50 +00:00)
```

[18]:
```
resample_rule = '7d'
time_column = 'activation_date'
value_column = 'sector_count'
title = 'Count of Sector Activation Date (#)'


groups = ['miner',
          pd.Grouper(key='activation_date', freq=resample_rule),
          pd.Grouper(key='expiration_date', freq=resample_rule)]
fig_df = df.groupby(groups).sum().reset_index()
px.density_heatmap(fig_df,
                   x='activation_date',
                   y='expiration_date',
                   z='sector_count',
                   animation_frame='miner')
```

```
time: 298 ms (started: 2021-04-11 01:27:50 +00:00)
```

[19]:
```
resample_rule = '1d'
time_column = 'activation_date'
value_column = 'sector_count'
title = 'Count of Sector Activation Date (#)'


groups = ['miner',
```

```
          pd.Grouper(key='activation_date', freq=resample_rule),
          pd.Grouper(key='expiration_date', freq=resample_rule)]
fig_df = df.groupby(groups).sum().reset_index()
fig = px.density_contour(fig_df,
                    x='activation_date',
                    y='expiration_date',
                    z='sector_count',
                    histfunc='sum',
                    color='miner')
fig.show()
```

time: 883 ms (started: 2021-04-11 01:27:51 +00:00)

[20]:
```
resample_rule = '1m'
time_column = 'expiration_date'
value_column = 'sector_count'
title = 'Upcoming Sector Expiration Count (%)'
resample_and_bar_plot_relative(df, resample_rule, time_column, value_column,␣
 ↪title).show()
```

time: 271 ms (started: 2021-04-11 01:27:51 +00:00)

[21]:
```
resample_rule = '1m'
time_column = 'activation_date'
value_column = 'sector_count'
title = 'Count of Sector Activation Date (%)'
resample_and_bar_plot_relative(df, resample_rule, time_column, value_column,␣
 ↪title).show()
```

time: 198 ms (started: 2021-04-11 01:27:52 +00:00)

1.3 Initial Pledge

[22]:
```
resample_rule = '1m'
time_column = 'expiration_date'
value_column = 'initial_pledge_in_fil'
title = 'Initial Pledge (FIL) of Expiring  Sectors (#)'


groups = [pd.Grouper(key='expiration_date', freq=resample_rule),
          'is_v1']

fig_df = (df.groupby(groups)
             .sum()
         )
```

```
fig = px.bar(fig_df.reset_index(),
             x=time_column,
             y=value_column,
             color='is_v1',
             title=title)
fig.show()
```

time: 155 ms (started: 2021-04-11 01:27:52 +00:00)

[23]:
```
resample_rule = '1m'
time_column = 'expiration_date'
value_column = 'initial_pledge_in_fil'
title = 'Initial Pledge (FIL) of Expiring Sectors, grouped by Miner and Sector␣
 ↪Version'


groups = ['miner',
          pd.Grouper(key='expiration_date', freq=resample_rule),
          'is_v1']

fig_df = (df.groupby(groups)
            .sum()
            .reset_index()
         )


fig = px.bar(fig_df,
             x=time_column,
             y=value_column,
             color='miner',
             facet_col='is_v1',
             title=title)
fig.show()
```

time: 250 ms (started: 2021-04-11 01:27:52 +00:00)

[24]:
```
resample_rule = '1m'
time_column = 'activation_date'
value_column = ['initial_pledge_in_thousand_fil']
title = 'Sum of Initial Pledge (FIL) across activation dates'
resample_and_bar_plot(df, resample_rule, time_column, value_column, title).
 ↪show()
```

time: 196 ms (started: 2021-04-11 01:27:52 +00:00)

1.4 RB Network Power

[25]:
```
resample_rule = '1m'
time_column = 'expiration_date'
value_column = 'network_power_in_pib'
title = 'RB Network Power (PiB) of Expiring  Sectors (#)'


groups = [pd.Grouper(key='expiration_date', freq=resample_rule),
          'is_v1']

fig_df = (df.groupby(groups)
            .sum()
         )


fig = px.bar(fig_df.reset_index(),
             x=time_column,
             y=value_column,
             color='is_v1',
             title=title)
fig.show()
```

```
time: 155 ms (started: 2021-04-11 01:27:53 +00:00)
```

[26]:
```
resample_rule = '1m'
time_column = 'expiration_date'
value_column = 'network_power_in_pib'
title = 'RB Network Power (PiB) of Expiring Sectors, grouped by Miner and␣
 ↪Sector Version'


groups = ['miner',
          pd.Grouper(key='expiration_date', freq=resample_rule),
          'is_v1']

fig_df = (df.groupby(groups)
            .sum()
            .reset_index()
         )


fig = px.bar(fig_df,
             x=time_column,
             y=value_column,
             color='miner',
             facet_col='is_v1',
             title=title)
```

```
fig.show()
```

```
time: 260 ms (started: 2021-04-11 01:27:53 +00:00)
```

[27]:
```
resample_rule = '1m'
time_column = 'activation_date'
value_column = ['network_power_in_pib']
title = 'Sum of RB Network Power (PiB) across activation dates'
resample_and_bar_plot(df, resample_rule, time_column, value_column, title).
 ↪show()
```

```
time: 194 ms (started: 2021-04-11 01:27:53 +00:00)
```