short_sector_state_before_upgrade

February 28, 2021

1 Stats for the Upcoming Sector Expiration before the PoRep Upgrade

1.1 Setting-up

[1]:
```
%load_ext autotime

%load_ext autoreload

%autoreload 2
```

```
time: 12.6 ms (started: 2021-02-24 16:16:37 -03:00)
```

[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, '../')
```

```
time: 670 ms (started: 2021-02-24 16:16:38 -03:00)
```

[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: 3.32 s (started: 2021-02-24 16:16:38 -03:00)
```

[4]:
```
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.activation_epoch)  < 1152000 /* Get␣
 ↪sectors with less than 400d lifetime */
        and msi.activation_epoch  < 265200 /* Get sectors activated before the␣
 ↪update */
        and msi.expiration_epoch > msi.height /* Get only active sectors */
        order by max_height
)
select
count(*) as sector_count,
sum(ss.initial_pledge::numeric) / 1e18 as initial_pledge_in_fil,
count(*) * 32 as network_power_in_gb,
date_trunc('WEEK', to_timestamp(height_to_unix(ss.activation_epoch))) as␣
 ↪activation_date,
date_trunc('WEEK', 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
order by activation_date, expiration_date
"""

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: 38.3 s (started: 2021-02-24 16:16:42 -03:00)
```

1.2 Visualizations

[5]:
```
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 (#): 7091151
Raw bytes power (PiB): 216
Initial pledge (FIL): 1463528.4263541135
---
time: 11.6 ms (started: 2021-02-24 16:17:20 -03:00)
```

[6]:
```
def resample_and_bar_plot(df, resample_rule, time_column, value_column, title):
    fig_df = df.resample(resample_rule, on=time_column).sum()
    fig = px.bar(fig_df.reset_index(),
                 x=time_column,
                 y=value_column,
                 log_y=True,
                 barmode='group',
                 title=title)
    return fig
```

```
time: 12.5 ms (started: 2021-02-24 16:17:20 -03:00)
```

[7]:
```
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: 312 ms (started: 2021-02-24 16:17:20 -03:00)
```

[8]:
```
resample_rule = '1m'
time_column = 'expiration_date'
value_column = ['initial_pledge_in_fil', 'network_power_in_pib']
title = 'Sum of Initial Pledge (kFIL) and RB Network Power (PiB) across␣
 ↪expiration dates'
resample_and_bar_plot(df, resample_rule, time_column, value_column, title).
 ↪show()
```

```
time: 60.7 ms (started: 2021-02-24 16:17:20 -03:00)
```

[9]:
```
resample_rule = '1m'
time_column = 'activation_date'
value_column = ['initial_pledge_in_thousand_fil', 'network_power_in_pib']
title = 'Sum of Initial Pledge (kFIL) and RB Network Power (PiB) across␣
 ↪activation dates'
resample_and_bar_plot(df, resample_rule, time_column, value_column, title).
 ↪show()
```

```
time: 60.8 ms (started: 2021-02-24 16:17:20 -03:00)
```