nix-casync chunk size analysis

January 15, 2022

[1]:
```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
```

[ ]:
```
df = pd.read_csv(
    "2pr5achd242cna6qfk086qy0ffxgsyv2-cacnk-sizes.csv",
    names=["size_compressed", "size_uncompressed", "chunk_name"]
)

df.drop("chunk_name", inplace=True, axis=1)
```

[3]:
```
df.describe()
```

[3]:

	size_compressed	size_uncompressed
count	3.421999e+07	3.421999e+07
mean	3.199065e+04	7.620992e+04
std	3.654161e+04	6.091455e+04
min	2.000000e+01	0.000000e+00
25%	8.991000e+03	3.154900e+04
50%	1.967100e+04	5.577700e+04
75%	4.001800e+04	9.966600e+04
max	2.621640e+05	2.621440e+05

[4]:
```
df.head()
```

[4]:

	size_compressed	size_uncompressed
0	13875	40468
1	36751	79176
2	9943	99748
3	18596	160993
4	7240	25549

[5]:
```
df.tail()
```

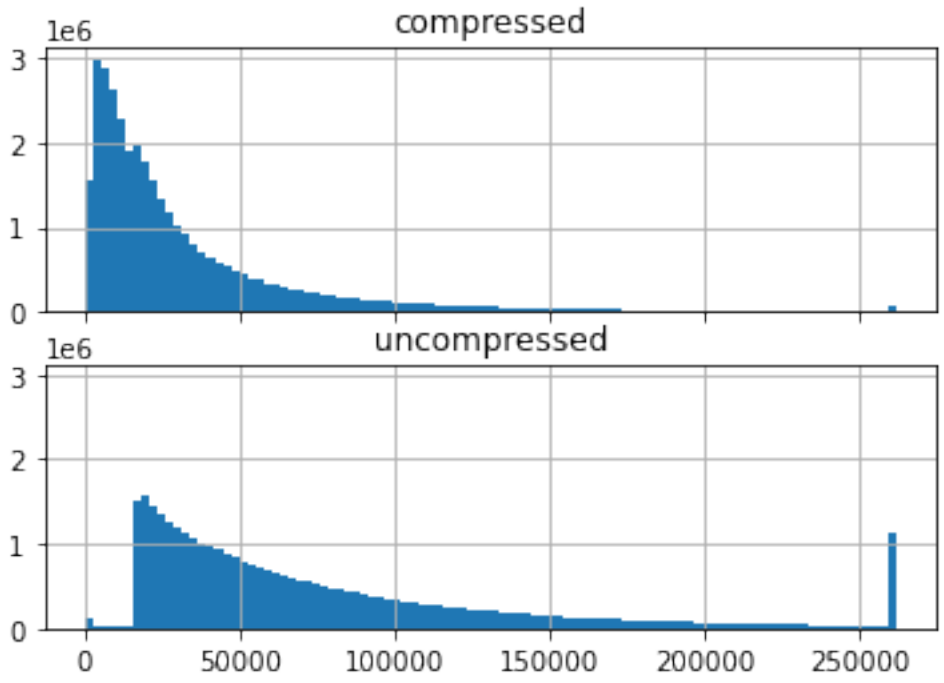
[5]:

	size_compressed	size_uncompressed
34219989	16404	16384
34219990	3512	18289

```
34219991            41175              41165
34219992           125010             129096
34219993            11244              28283
```

[6]:
```
fig, ax = plt.subplots(2,1, sharex=True, sharey=True)
ax[0].set_title("compressed")
ax[1].set_title("uncompressed")
df['size_compressed'].hist(bins=100, ax=ax[0])
df['size_uncompressed'].hist(bins=100, ax=ax[1])
```

[6]: <AxesSubplot:title={'center':'uncompressed'}>



[7]:
```
# find chunk size 0
percentage = (df['size_uncompressed'] < 1).mean()

print(f"Percentage of chunks with size 0: {percentage*100:0.1f}%")
```

```
Percentage of chunks with size 0: 0.0%
```

[8]:
```
max_chunk_size = 262144
tolerance = 1000

max_val = max_chunk_size - tolerance

for col, s_data in df.iteritems():
```

```
    percentage = (s_data > max_val).mean()

    print(f"[{col}] Percentage of chunks hitting the max size: {percentage*100:
↪0.1f}%")
```

```
[size_compressed] Percentage of chunks hitting the max size: 0.2%
[size_uncompressed] Percentage of chunks hitting the max size: 3.3%
```

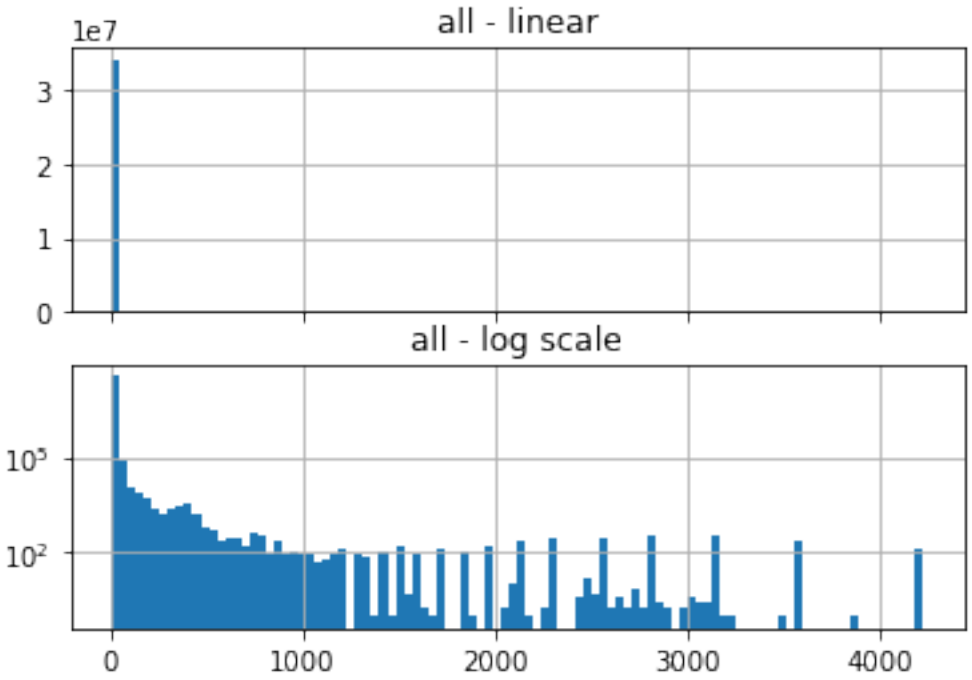
[9]:
```
def plot_compression_rate_distribution(
    df: pd.DataFrame,
    title: str
) -> None:

    compression_rate = df['size_uncompressed'] / (df['size_compressed']+1)

    fig, ax = plt.subplots(2,1, sharex=True)
    ax[0].set_title(f"{title} - linear")
    ax[1].set_title(f"{title} - log scale")

    compression_rate.hist(bins=100, log=False, ax=ax[0])
    compression_rate.hist(bins=100, log=True, ax=ax[1])
```

[10]:
```
plot_compression_rate_distribution(
    df=df,
    title="all"
)
```



[ ]:
```
quantiles = df['size_uncompressed'].quantile(np.linspace(0,1,5)).values

print(quantiles)

for idx in range(len(quantiles)-1):
    quantile_low = quantiles[idx]
    quantile_high = quantiles[idx+1]

    plot_compression_rate_distribution(
        df=df[ (df['size_uncompressed'] > quantile_low) &␣
↪(df['size_uncompressed'] < quantile_high)],
        title=f"chunk size in [{int(quantile_low)}, {int(quantile_high)}]"'
    )
```

```
[     0.  31549.  55777.  99666. 262144.]
```