To better understand the distribution of traffic among different agents, we can first sort them by the total size of their requests, then tally the cumulative traffic.

In [1]:
```
import pandas as pd
import numpy as np
from scipy import stats
import plotly.express as px
import plotly.graph_objects as go
from plotly.subplots import make_subplots
pd.set_option('max_columns', None)
```

In [2]:
```
df_groupby_user = pd.read_csv('data_groupby_user.csv', index_col=0)
df_groupby_user.shape
```

Out[2]:
```
(21264, 3)
```

In [3]:
```
df1 = df_groupby_user[['agent', 'request_sum']]
# sort by request size
df1 = df1.sort_values(by=['request_sum'], ascending=False)
df1 = df1['request_sum'].reset_index()
df1 = df1.drop(['index'], axis=1)
# calculate cumulative sum
df1['request_size_cumulative'] = df1.cumsum()
df1['request_size_cumulative'] = df1['request_size_cumulative']/pow(1024,4)
# calculate percentage
total_size = df1.iloc[-1]['request_size_cumulative']
df1['percentage'] = df1['request_size_cumulative'] / total_size
df1 = df1.reset_index()
# convert x to percentage
total_user = df1.shape[0]
df1['idx_percentage'] = df1['index'] / total_user
```

In [4]:
```
total_size
```

Out[4]:
```
5.736327559936399
```

In [5]:
```
# find percentage of id corresponding 80% of request
# df1[(df1['percentage']>0.799)&(df1['percentage']<0.801)]
# 0.057
```

In [6]:
```
fig = px.line(df1, x="idx_percentage", y="request_size_cumulative", title='Cumulative traffic by agent')

fig.update_layout(xaxis=dict(tickformat=".0%"))
fig.update_xaxes(title="agent (sorted by request size descending)")
fig.update_yaxes(title="cumulative traffic in TB")

fig.add_hline(y=total_size*0.8, line_width=1, line_dash="dash", line_color="grey",
              annotation_text="80% of total traffic",
              annotation_position="bottom right")
fig.add_vline(x=0.057, line_width=1, line_dash="dash", line_color="grey",
              annotation_text="5.7% of total agent",
              annotation_position="bottom right")

fig.show()
```