

metrics_openc1

August 14, 2025

```
[11]: # !pip install numpy matplotlib pandas tqdm
```

```
[12]: import json

import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
```

```
[13]: with open("metrics_openc1.json", encoding="utf-8") as f:
      data = json.load(f)
```

```
[14]: df = pd.DataFrame(data)
      df.sort_values("duration", ascending=False)
```

```
[14]:
```

	duration	value	block_size	size	runtime \
122	0.254616	0.033328	64	10000000	OpenCL
134	0.230729	0.033328	512	10000000	OpenCL
138	0.208531	0.033328	1024	10000000	OpenCL
126	0.185868	0.033328	128	10000000	OpenCL
119	0.182566	0.033328	32	10000000	clBLAS
..
53	0.003115	0.033326	1024	10000000	OpenBLAS
17	0.002985	0.033326	256	10000000	OpenBLAS
33	0.002955	0.033326	32	10000000	OpenBLAS
25	0.002934	0.033326	1024	10000000	OpenBLAS
21	0.002904	0.033326	512	10000000	OpenBLAS

	device	grid_size
122	Intel(R) Arc(TM) Graphics	256.0
134	Intel(R) Arc(TM) Graphics	256.0
138	Intel(R) Arc(TM) Graphics	256.0
126	Intel(R) Arc(TM) Graphics	256.0
119	Intel(R) Arc(TM) Graphics	256.0
..
53	CPU	32.0
17	CPU	16.0
33	CPU	32.0
25	CPU	16.0

[196 rows x 7 columns]

```
[ ]: runtimes = df["runtime"].unique()

target_runtime = 'OpenCL'
df_filtered = df[df['runtime'] == target_runtime]

block_sizes = sorted(df_filtered['block_size'].unique())
counts = sorted(df_filtered['grid_size'].unique())

X, Y = np.meshgrid(block_sizes, counts)
Z = np.zeros_like(X, dtype=float)

for i, count in enumerate(counts):
    for j, block in enumerate(block_sizes):
        match = df_filtered[
            (df_filtered['block_size'] == block) &
            (df_filtered['grid_size'] == count)
        ]
        if not match.empty:
            Z[i, j] = match['duration'].values[0]
        else:
            Z[i, j] = np.nan

fig = plt.figure(figsize=(10, 7))
ax = fig.add_subplot(111, projection='3d')
surf = ax.plot_surface(X, Y, Z, cmap='plasma', edgecolor='k')

ax.set_xlabel('Block Size')
ax.set_ylabel('Number of Blocks')
ax.set_zlabel('Execution Time (s)')
ax.set_title(f'Execution Time for {target_runtime}')
fig.colorbar(surf, shrink=0.5, aspect=10)

plt.show()
```

Execution Time for OpenCL

