Performance Visualization Options

To effectively visualize and analyze the performance data related to the events (search and update) from your `forwarder_perf` table, you can use a variety of charts and Key Performance Indicators (KPIs). Here are some suggestions:

Charts

1. **Bar Chart**:
   - **Average Time Taken for Each Event**: Calculate the average duration for search and update events. A bar chart can display the average times, with separate bars for search and update events.

```python
import matplotlib.pyplot as plt

# Example data calculation
avg_search_time = (df['search_ended'] - df['search_started']).mean()
avg_update_time = (df['update_ended'] - df['update_started']).mean()

# Plotting
events = ['Search', 'Update']
times = [avg_search_time, avg_update_time]

plt.bar(events, times, color=['blue', 'green'])
plt.xlabel('Event')
plt.ylabel('Average Time (seconds)')
plt.title('Average Time Taken for Search and Update Events')
plt.show()
```

2. **Line Chart**:
   - **Trend Over Time**: Show how the time taken for search and update events changes over time. You can plot the durations against the timestamps when events started or ended.

```python
# Assuming df is your DataFrame containing the performance data
df['search_duration'] = (df['search_ended'] - df['search_started']).dt.total_seconds()
df['update_duration'] = (df['update_ended'] - df['update_started']).dt.total_seconds()

plt.plot(df['search_started'], df['search_duration'], label='Search')
plt.plot(df['update_started'], df['update_duration'], label='Update')
plt.xlabel('Time')
plt.ylabel('Duration (seconds)')
plt.title('Event Duration Over Time')
plt.legend()
plt.show()
```

3. **Box Plot**:
    - **Distribution of Durations**: A box plot can show the distribution, median, and outliers for the durations of search and update events.

```python
data = [df['search_duration'].dropna(), df['update_duration'].dropna()]
plt.boxplot(data, labels=['Search', 'Update'])
plt.ylabel('Duration (seconds)')
plt.title('Distribution of Event Durations')
plt.show()
```

4. **Heatmap**:
    - **Frequency of Events**: If you want to see the frequency of events at different times of the day or week, a heatmap can be useful.

```python
import seaborn as sns

df['hour'] = df['search_started'].dt.hour
heatmap_data = df.groupby('hour').size().unstack()

sns.heatmap(heatmap_data, cmap='Blues')
plt.xlabel('Hour of Day')
plt.ylabel('Event Count')
plt.title('Event Frequency by Hour')
plt.show()
```

KPIs

1. **Average Duration**:
    - **Average Search Duration**: The average time taken for search events to complete.
    - **Average Update Duration**: The average time taken for update events to complete.

```python
avg_search_duration = df['search_duration'].mean()
avg_update_duration = df['update_duration'].mean()
```

2. **Event Counts**:
    - **Number of Searches**: Total number of search events.
    - **Number of Updates**: Total number of update events.

```python
search_count = df['search_started'].notna().sum()
```

```
update_count = df['update_started'].notna().sum()
```

3. **Max and Min Durations**:
   - **Longest and Shortest Search Duration**: The maximum and minimum time taken for a search event.
   - **Longest and Shortest Update Duration**: The maximum and minimum time taken for an update event.

```python
max_search_duration = df['search_duration'].max()
min_search_duration = df['search_duration'].min()
max_update_duration = df['update_duration'].max()
min_update_duration = df['update_duration'].min()
```

4. **Standard Deviation**:
   - **Variation in Durations**: The standard deviation of search and update durations to understand variability.

```python
std_search_duration = df['search_duration'].std()
std_update_duration = df['update_duration'].std()
```

5. **Success Rate**:
   - **Event Completion Rate**: The rate at which events are successfully started and completed.

```python
search_success_rate = search_count / len(df)
update_success_rate = update_count / len(df)
```

Dashboard Integration

To integrate these visualizations and KPIs into a dashboard, you can use tools like:

- **Grafana**: For real-time monitoring and visualization.
- **Tableau**: For more detailed and interactive data exploration.
- **Dash by Plotly**: To create web-based analytical applications.

In the dashboard, you can arrange these charts and KPIs to provide a comprehensive view of the event performance, enabling stakeholders to monitor and analyze the data effectively.