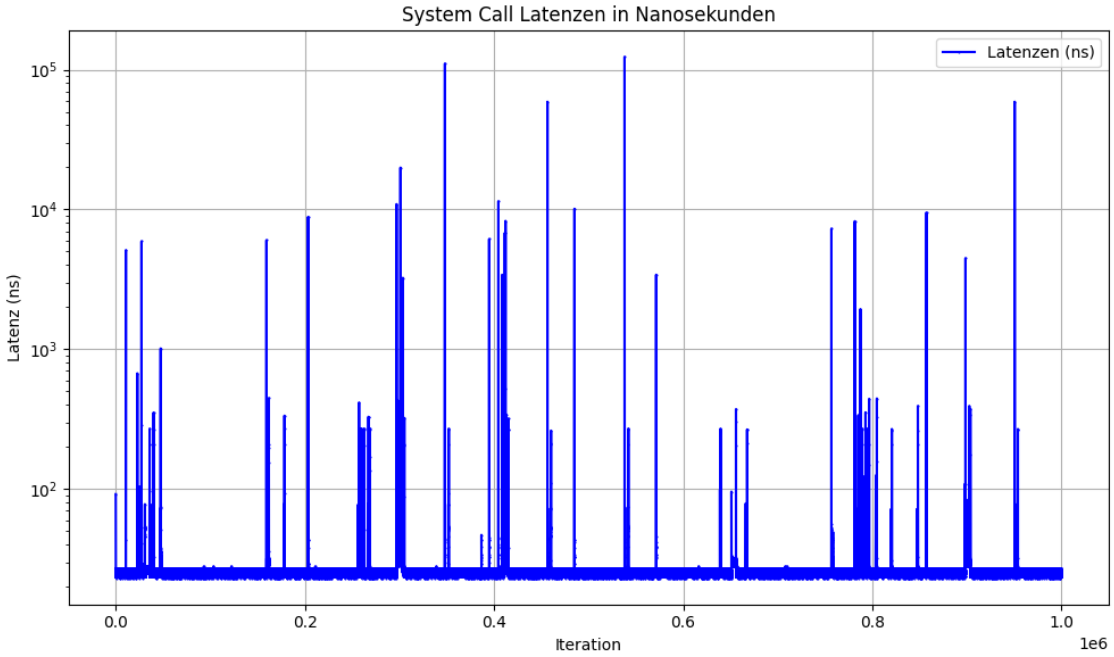
TimeContext

November 19, 2024

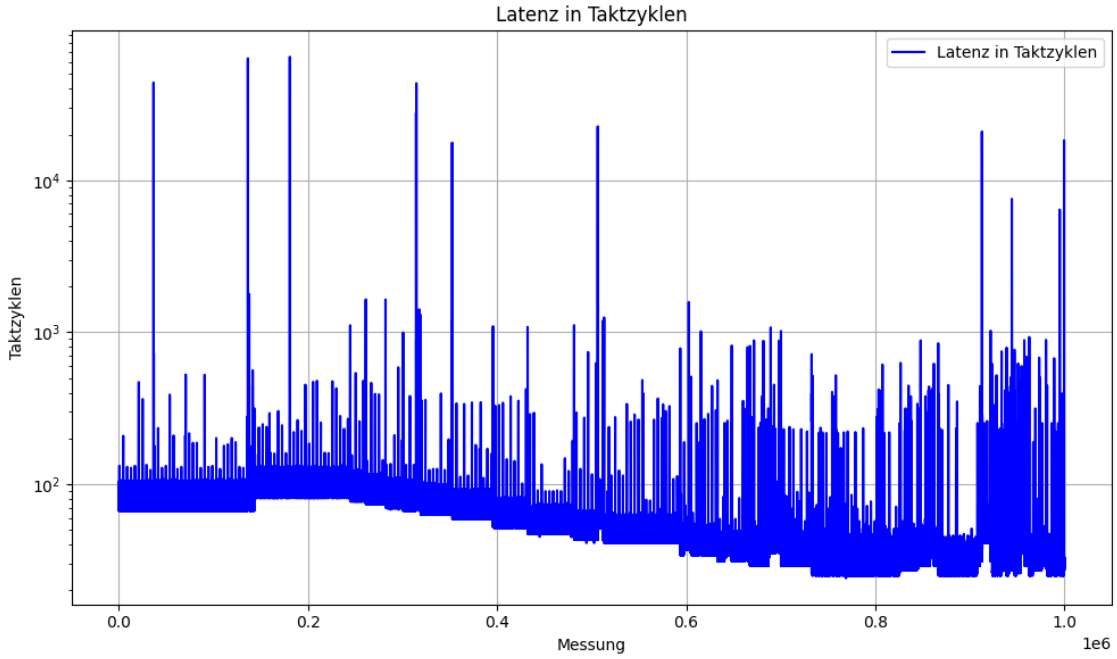
[1]:
```
import matplotlib.pyplot as plt
import pandas as pd

latencies = []
with open("latencies.txt", "r") as file:
    for line in file:
        latencies.append(int(line.strip()))

plt.figure(figsize=(10, 6))

plt.plot(latencies, label="Latenzen (ns)", color='blue', linestyle='-',␣
 ↪marker='.', markersize=1)

plt.title("System Call Latenzen in Nanosekunden")
plt.xlabel("Iteration")
plt.ylabel("Latenz (ns)")
plt.yscale('log')
plt.grid(True)
plt.legend()

plt.tight_layout()
plt.show()
```



[2]:
```
mean_syscall = pd.Series(latencies).mean()
mean_syscall
```

[2]:
```
25.836756
```

[3]:
```
file_name = "latencies_tx.txt"
data = pd.read_csv(file_name)

measurements = data['Messung']
latency_ticks = data['Latenz_tz']

plt.figure(figsize=(10, 6))

plt.plot(measurements, latency_ticks, label="Latenz in Taktzyklen",␣
 ↪color="blue")
plt.xlabel("Messung")
plt.ylabel("Taktzyklen")
plt.yscale('log')
plt.title("Latenz in Taktzyklen")
plt.grid(True)
plt.legend()

plt.tight_layout()
plt.show()
```



[4]:
```
mean = data['Latenz_tz'].mean()
mean
```

[4]: 53.608293

[5]:
```
without_peak = data[data['Latenz_tz'] < 1000]
avg_tz = without_peak['Latenz_tz'].mean()
```

[6]:
```
with_peak = data[data['Latenz_tz'] > 1000]
avg_tz_cc = with_peak['Latenz_tz'].mean()
```

[7]:
```
def tz_in_ns(tz, ps):
    return tz / ps
```

[8]:
```
print(f'Durchschnittliche Zeit Abruf RDTSC: {tz_in_ns(avg_tz, 3.0):.2f} ns')
```

```
Durchschnittliche Zeit Abruf RDTSC: 17.75 ns
```

[9]:
```
print(f'Durchschnittliche Zeit eines Kontextwechsels: {tz_in_ns(avg_tz_cc, 3.0):
  ↪.2f} ns')
```

```
Durchschnittliche Zeit eines Kontextwechsels: 4004.14 ns
```

[10]:
```
print(f'Durchschnittliche Zeit eines Syscalls ohne Ausführung der Abfrage:␣
  ↪{mean_syscall - tz_in_ns(avg_tz, 3.0):.2f} ns')
```

```
Durchschnittliche Zeit eines Syscalls ohne Ausführung der Abfrage: 8.09 ns
```