

```
In [1]: import sys
sys.path.append("../")
```

```
In [2]: import cmdbench
```

```
In [3]: stress_benchmark_write = cmdbench.benchmark_command("node write_test.js")
```

```
In [4]: first_iteration_write = stress_benchmark_write.get_statistics()
print("GNU Time for writing: %s" % first_iteration_write.gnu_time.disk)
print("psutil for writing: %s" % first_iteration_write.psutil.disk)
```

```
GNU Time for writing: {
  'file_system_inputs': (mean: 0.0, stdev: 0.0, min: 0, max: 0)
  'file_system_outputs': (mean: 13107200.0, stdev: 0.0, min: 13107200,
max: 13107200)
}
```

```
psutil for writing: {
  'io_counters': {
    'read_bytes': (mean: 0.0, stdev: 0.0, min: 0, max: 0)
    'read_chars': (mean: 8167.0, stdev: 0.0, min: 8167, max: 8167)
    'read_count': (mean: 17.0, stdev: 0.0, min: 17, max: 17)
    'write_bytes': (mean: 13107200.0, stdev: 0.0, min: 13107200, max:
13107200)
    'write_chars': (mean: 13107220.0, stdev: 0.0, min: 13107220, max:
13107220)
    'write_count': (mean: 104.0, stdev: 0.0, min: 104, max: 104)
  },
}
```

```
In [5]: stress_benchmark_read = cmdbench.benchmark_command("python read_binary_t
# print(stress_benchmark_read.get_single_iteration().general.stdout_data)
```

```
In [6]: first_iteration_read = stress_benchmark_read.get_statistics()
print("GNU Time for reading: %s" % first_iteration_read.gnu_time.disk)
print("psutil for reading: %s" % first_iteration_read.psutil.disk)
```

```
GNU Time for reading: {
  'file_system_inputs': (mean: 0.0, stdev: 0.0, min: 0, max: 0)
  'file_system_outputs': (mean: 0.0, stdev: 0.0, min: 0, max: 0)
}

psutil for reading: {
  'io_counters': {
    'read_bytes': (mean: 0.0, stdev: 0.0, min: 0, max: 0)
    'read_chars': (mean: 13318271.0, stdev: 0.0, min: 13318271, max: 1
3318271)
    'read_count': (mean: 261.0, stdev: 0.0, min: 261, max: 261)
    'write_bytes': (mean: 0.0, stdev: 0.0, min: 0, max: 0)
    'write_chars': (mean: 290.0, stdev: 0.0, min: 290, max: 290)
    'write_count': (mean: 1.0, stdev: 0.0, min: 1, max: 1)
  },
}
```

```
In [7]: !g++ -std=c++17 read_binary_test.cpp -o read_binary_test
```

```
In [8]: stress_benchmark_read = cmdbench.benchmark_command("./read_binary_test",
print(stress_benchmark_read.get_single_iteration().general.stdout_data)
```

Total file bytes: 13107300

```
In [9]: first_iteration_read = stress_benchmark_read.get_statistics()
print("GNU Time for reading: %s" % first_iteration_read.gnu_time.disk)
print("psutil for reading: %s" % first_iteration_read.psutil.disk)
```

```
GNU Time for reading: {
  'file_system_inputs': (mean: 0.0, stdev: 0.0, min: 0, max: 0)
  'file_system_outputs': (mean: 0.0, stdev: 0.0, min: 0, max: 0)
}

psutil for reading: {
  'io_counters': {
    'read_bytes': (mean: 0.0, stdev: 0.0, min: 0, max: 0)
    'read_chars': (mean: 13111644.0, stdev: 0.0, min: 13111644, max: 1
3111644)
    'read_count': (mean: 1809.0, stdev: 0.0, min: 1809, max: 1809)
    'write_bytes': (mean: 0.0, stdev: 0.0, min: 0, max: 0)
    'write_chars': (mean: 27.0, stdev: 0.0, min: 27, max: 27)
    'write_count': (mean: 1.0, stdev: 0.0, min: 1, max: 1)
  },
}
```

```
In [10]: # Samples/ms
# len(stress_benchmark_read.get_single_iteration().time_series.sample_m
```

