

```

from simulator import MM1QueueSimulator
import argparse

#parameters
arrival_rate = 1.0 #λ
service_rate = 2.0 #μ
simulation_time = 2000.0 #seconds
warmup = simulation_time / 3

parser = argparse.ArgumentParser(description="M/M/1 Queue Simulation")
parser.add_argument("--arrival_rate", type=float, default=arrival_rate, help=f"Arrival rate (λ, default {arrival_rate})")
parser.add_argument("--service_rate", type=float, default=service_rate, help=f"Service rate (μ, default {service_rate})")
parser.add_argument("--simulation_time", type=float, default=simulation_time, help=f"Time of the simulation (default {simulation_time})")
parser.add_argument("--warmup", type=float, default=warmup, help=f"Time of the simulation (default simulation_time / 3)")
args = parser.parse_args()

arrival_rate = args.arrival_rate
service_rate = args.service_rate
if(service_rate <= arrival_rate):
    print("Cannot continue: system is not stable! Must have μ > λ!")
    exit(0)
simulation_time = args.simulation_time
warmup = simulation_time / 3
warmup = args.warmup
print(f"You are running the simulator with an arrival_rate of {arrival_rate}, a service_rate of {service_rate} and a simulation_time of {simulation_time} (warm

#####SINGULAR CASE: very noise results#####
print("Single run 1")
simulator = MM1QueueSimulator(arrival_rate, service_rate, simulation_time)
simulator.simulate()

rho = arrival_rate / service_rate
theoretical_avg = rho / (1 - rho)
empirical_average = simulator.average_packets_in_system()

print(f"\nTheoretical average number of packets in system: {theoretical_avg:.4f}\n")

print(f"Empirical average number of packets in system: {empirical_average:.4f}")
simulator.plot_together()

empirical_average = simulator.average_packets_in_system(warmup)
print(f"[Warmup {warmup:.3f}] Empirical average number of packets in system: {empirical_average:.4f}")
#simulator.plot_cumulative_packets_in_system(warmup)
#simulator.plot_packets_in_system(warmup)

#####
print("Single run 2")
simulator = MM1QueueSimulator(arrival_rate, service_rate, simulation_time)
simulator.simulate()

rho = arrival_rate / service_rate
theoretical_avg = rho / (1 - rho)
empirical_average = simulator.average_packets_in_system()

print(f"\nTheoretical average number of packets in system: {theoretical_avg:.4f}\n")

print(f"Empirical average number of packets in system: {empirical_average:.4f}")
simulator.plot_together()

empirical_average = simulator.average_packets_in_system(warmup)
print(f"[Warmup {warmup:.3f}] Empirical average number of packets in system: {empirical_average:.4f}")
#simulator.plot_cumulative_packets_in_system(warmup)
#simulator.plot_packets_in_system(warmup)

#Not robust results: we must run the simulator multiple times

```