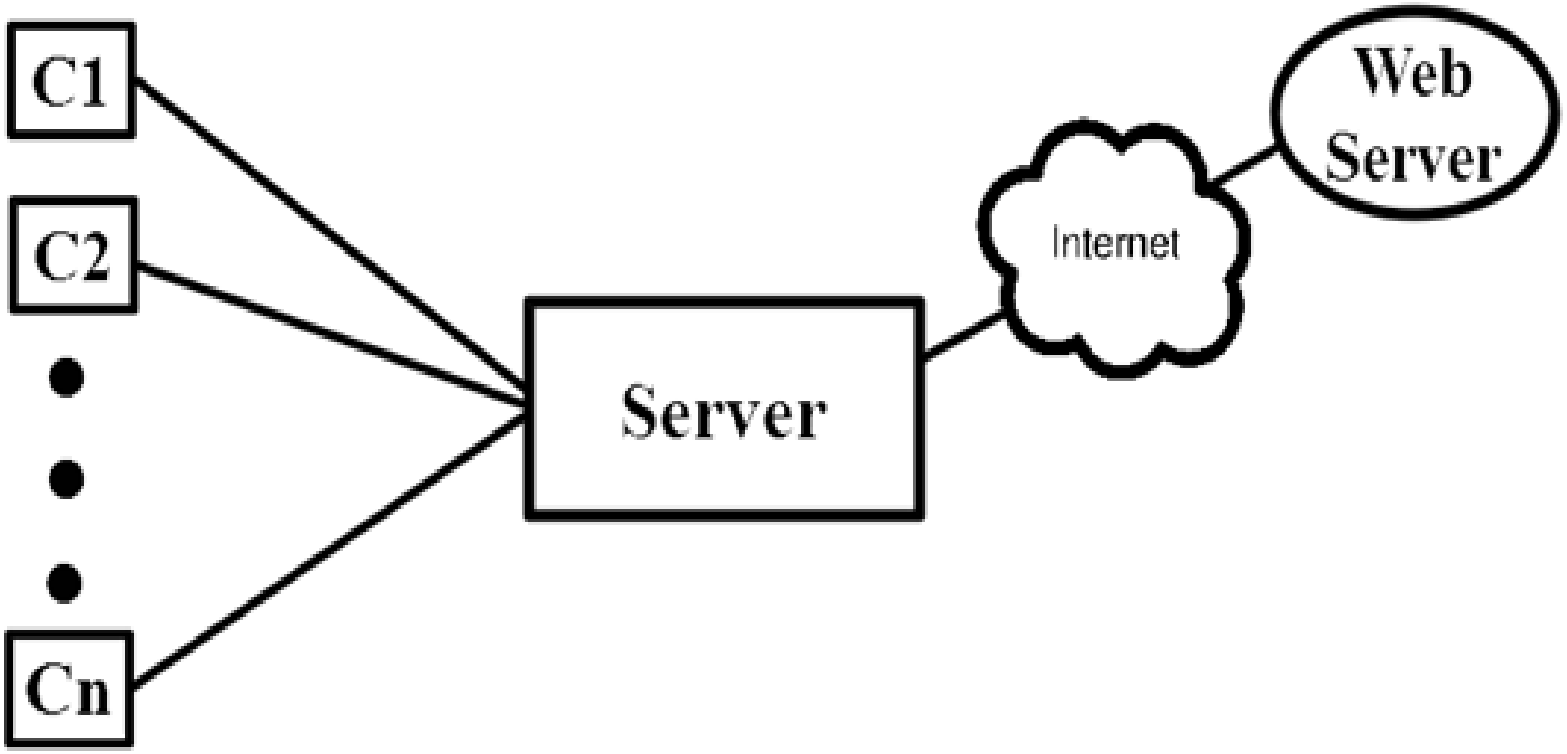
Multiprogrammed Server

- Project 11 -

Mauro Abozzi
Stefano Cicero
Matteo Luciani Brunini
Simone Tavoletta

System Overview

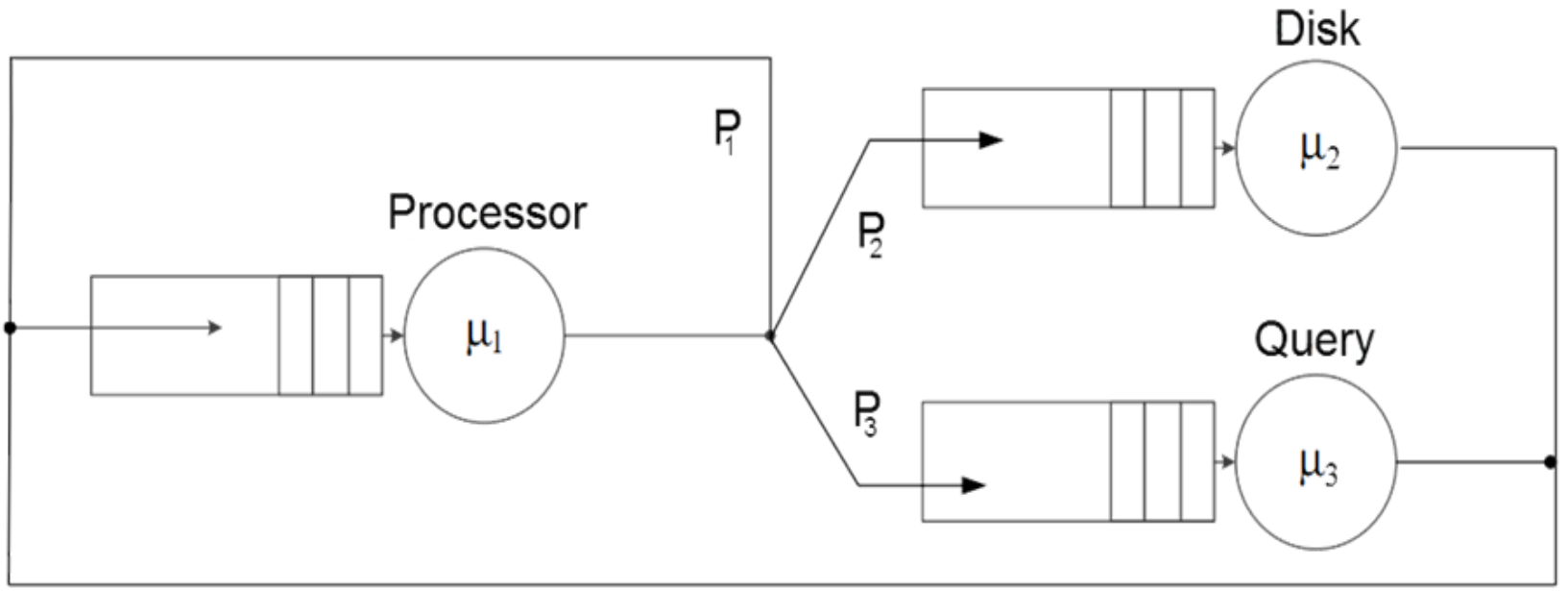
Multiprogrammed server which manages N different clients



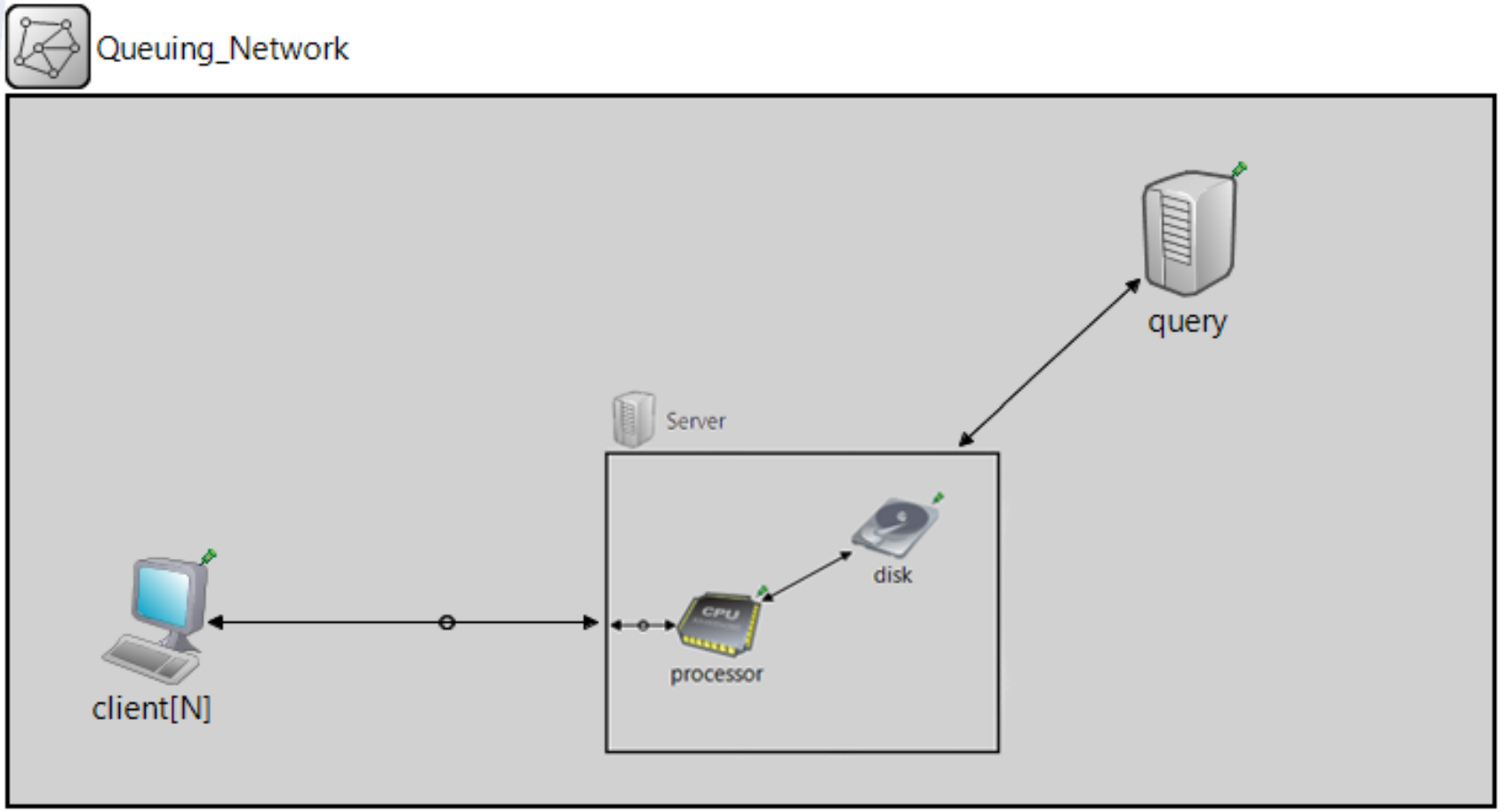
System Overview

After processing has occured with probability:

- **p1** → Transaction terminated
- **p2** → disk access is required
- **p3 = 1-p1-p2** → remote query is issued



OMNeT++ Model



Experiment Design

Scenarios (service time mean in seconds)

1. Processor = Disk = Query → (0.1 = 0.1 = 0.1)
2. Processor << Disk << Query → (0.01 << 0.1 << 1.0)
3. Query << Processor << Disk → (0.01 << 0.1 << 1.0)
4. Disk << Query << Processor → (0.01 << 0.1 << 1.0)

Cases

A. p1=70%, p2=20%, p3=10%
B. p1=10%, p2=20%, p3=70%
C. p1=10%, p2=70%, p3=20%
D. p1=33%, p2=33%, p3=34%

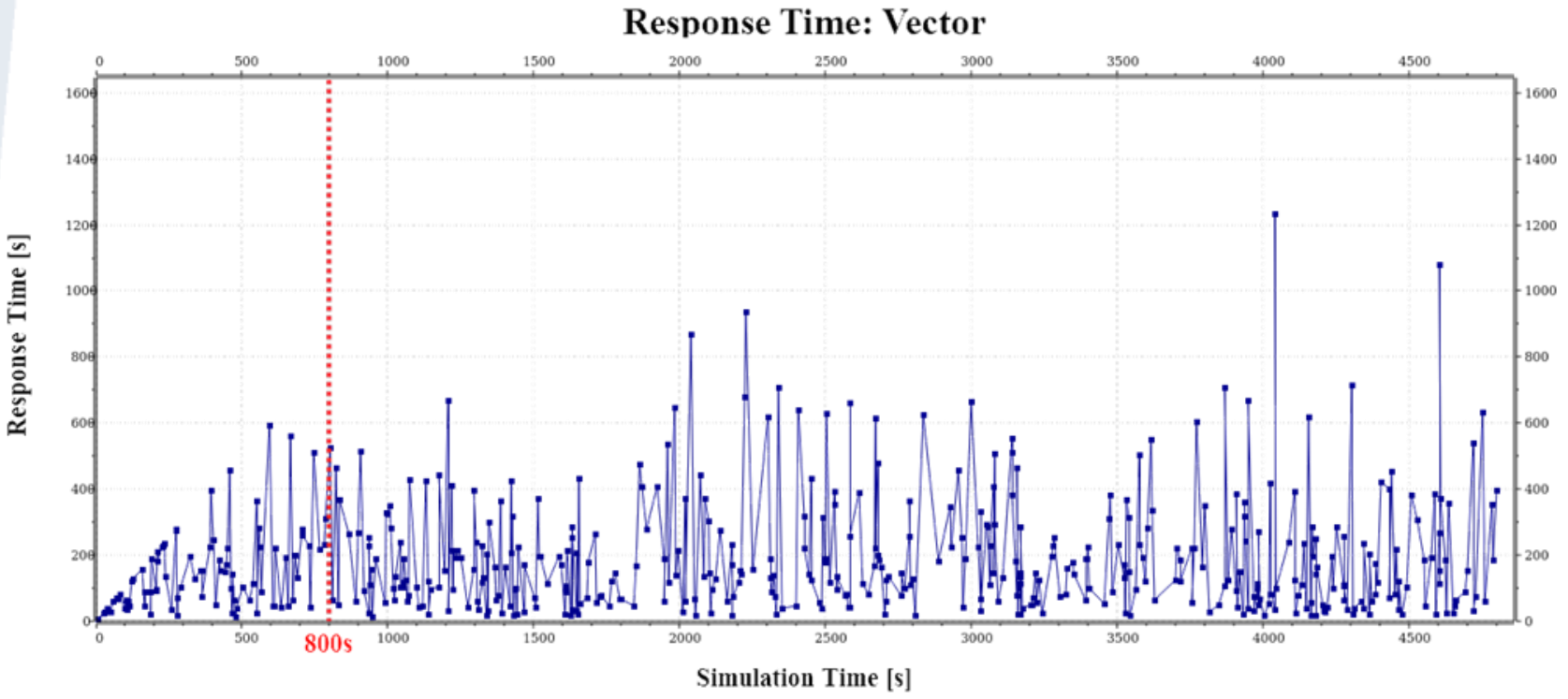
Experiment Design

Warm-up estimation:

N = N_{max} = **20** clients

Scenario 4 Case C (worst case)

Warm-up = **800s** Simulation-time = **4800s**

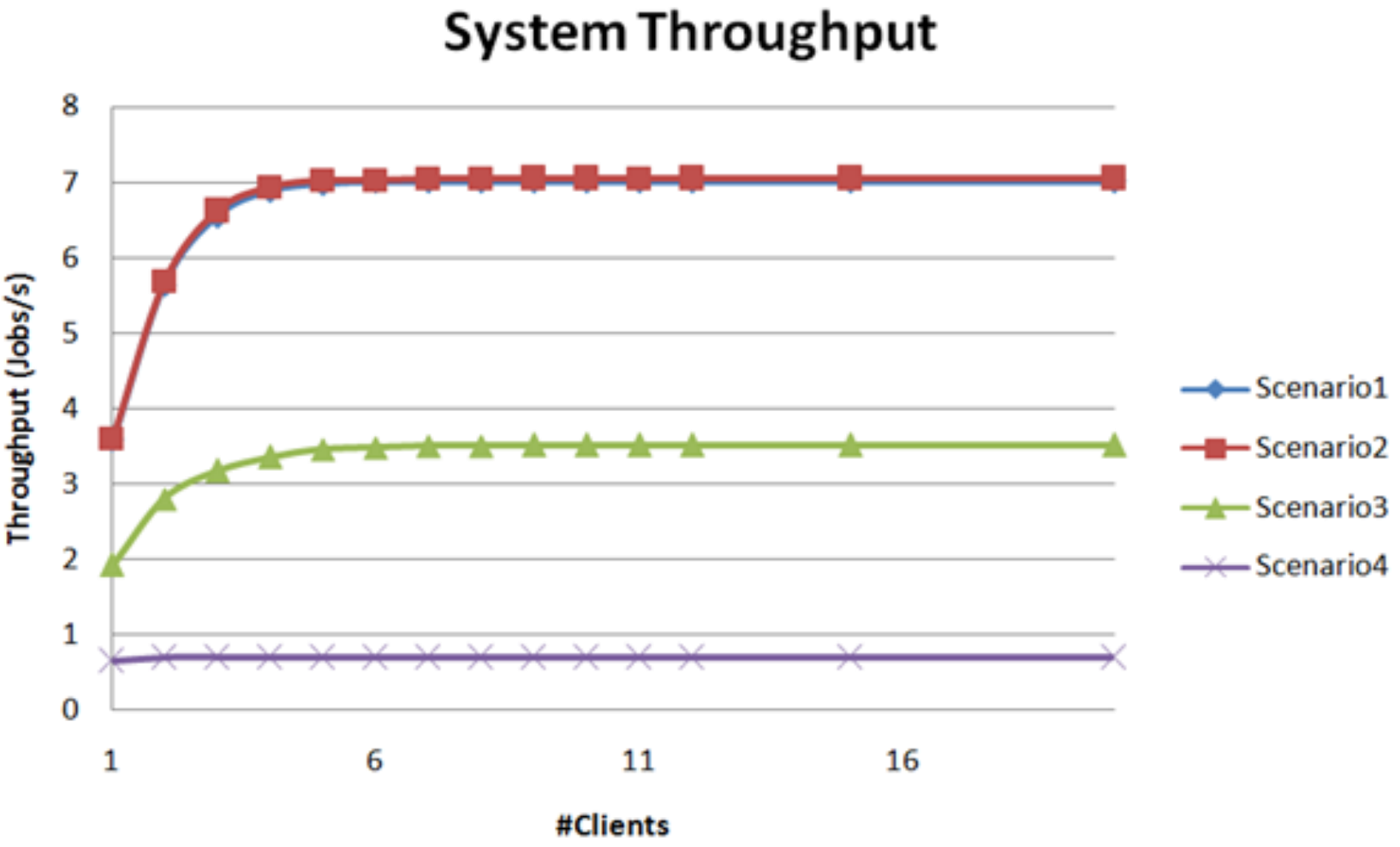


Performance Analysis

Throughput

1- α = 0.99 n = 10

Case A - p1=70% p2=20% p3=10%

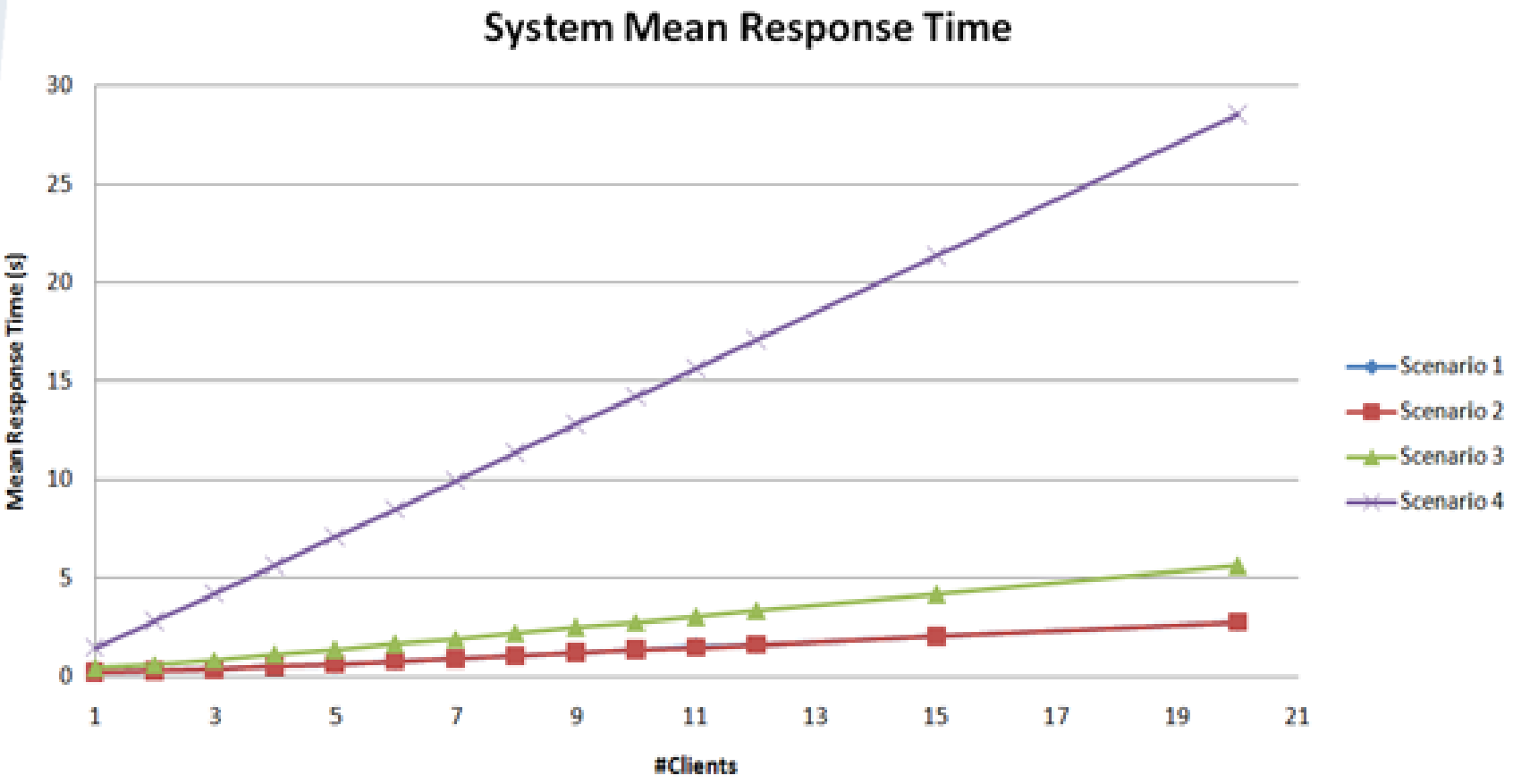


Performance Analysis

Response Time

1- α = 0.99 n = 10

Case A - p1=70% p2=20% p3=10%

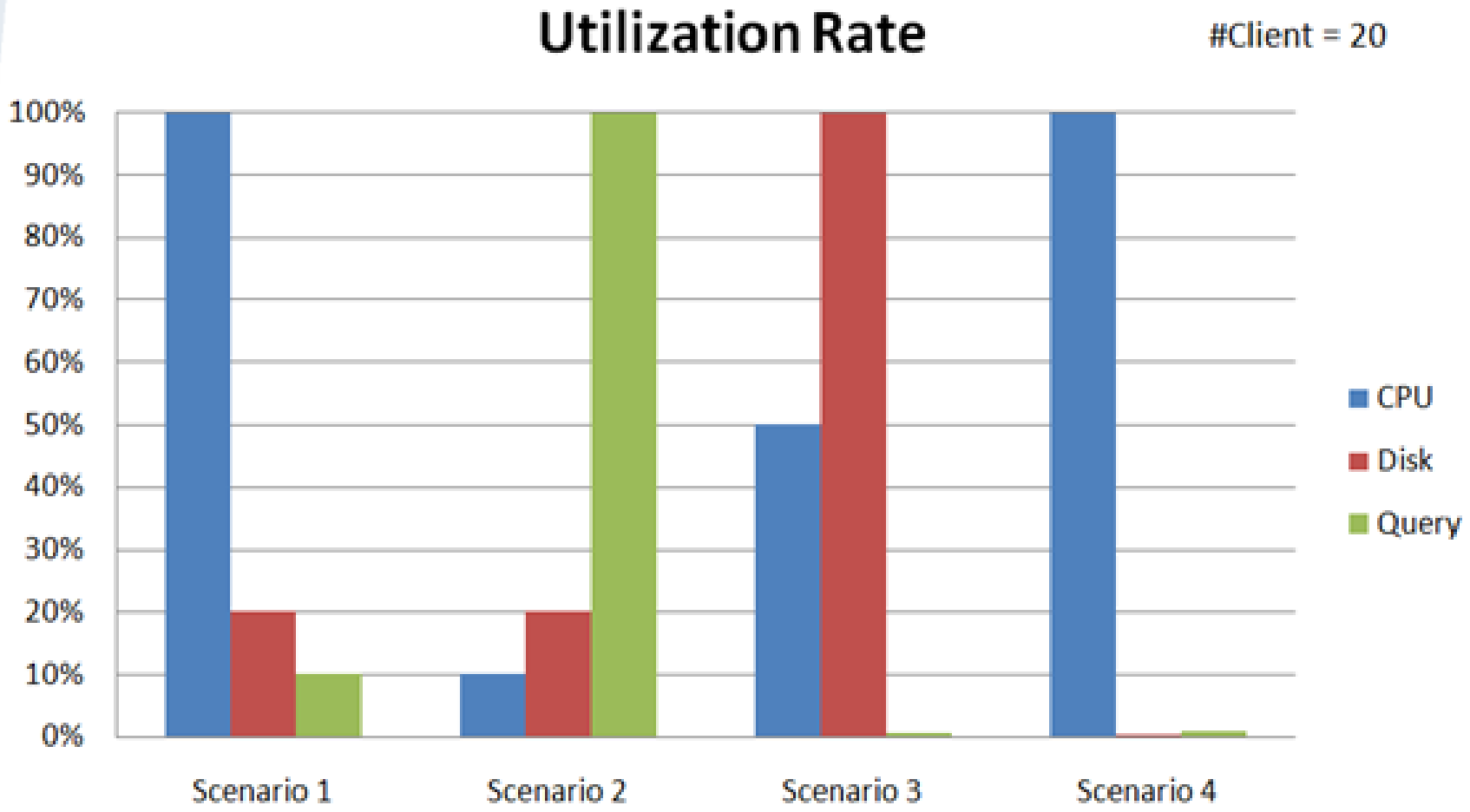


Performance Analysis

Utilization rate

1- α = 0.99 n = 10

Case A - p1=70% p2=20% p3=10%



Conclusion

- ✓ Scenario 1 shows better performance compared to all the others because the system is well balanced and the service time means are small enough
- ✓ Scenario 1 represents all the cases in which the three components have the same order of magnitude with respect to the service time means
- ✓ If probabilities are unknown, to get an optimal performance, select the speed of the devices by looking at the quickest one
- ✓ Otherwise, update the bottleneck and put also the cpu to the same speed
- ✓ Generally, the processor must not be slower than the other components