Queuing Theory

From HnP6

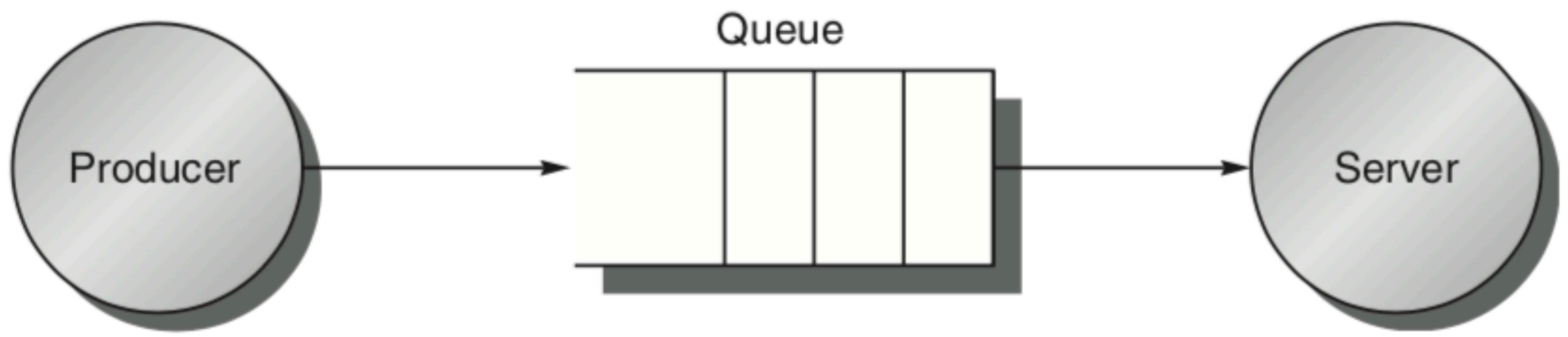

Figure D.8 The traditional producer-server model of response time and throughput. Response time begins when a task is placed in the buffer and ends when it is completed by the server. Throughput is the number of tasks completed by the server in unit time.

1. *Entry time*—The time for the user to enter the command.
2. *System response time*—The time between when the user enters the command and the complete response is displayed.
3. *Think time*—The time from the reception of the response until the user begins to enter the next command.

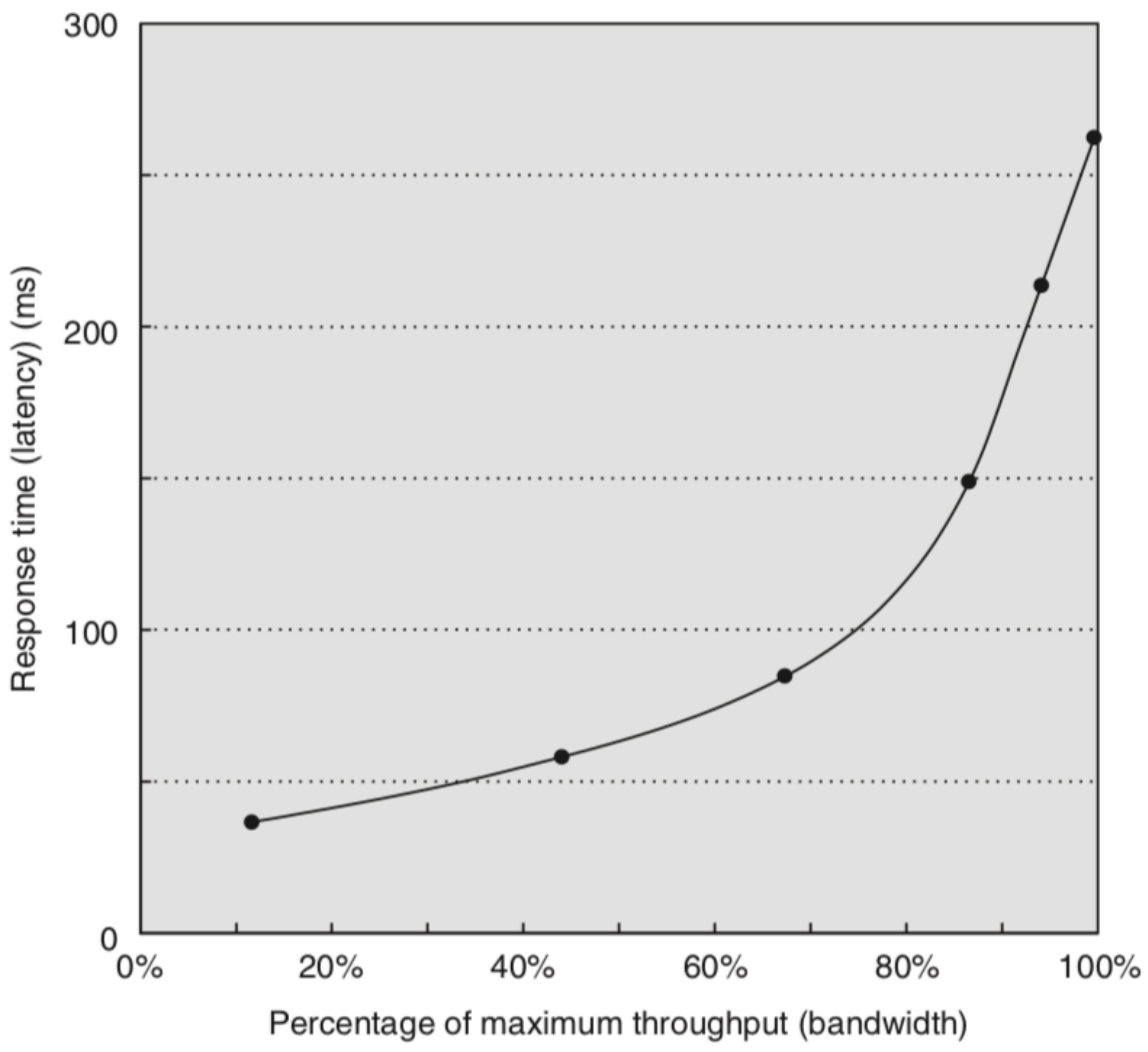

Figure D.9 Throughput versus response time. Latency is normally reported as

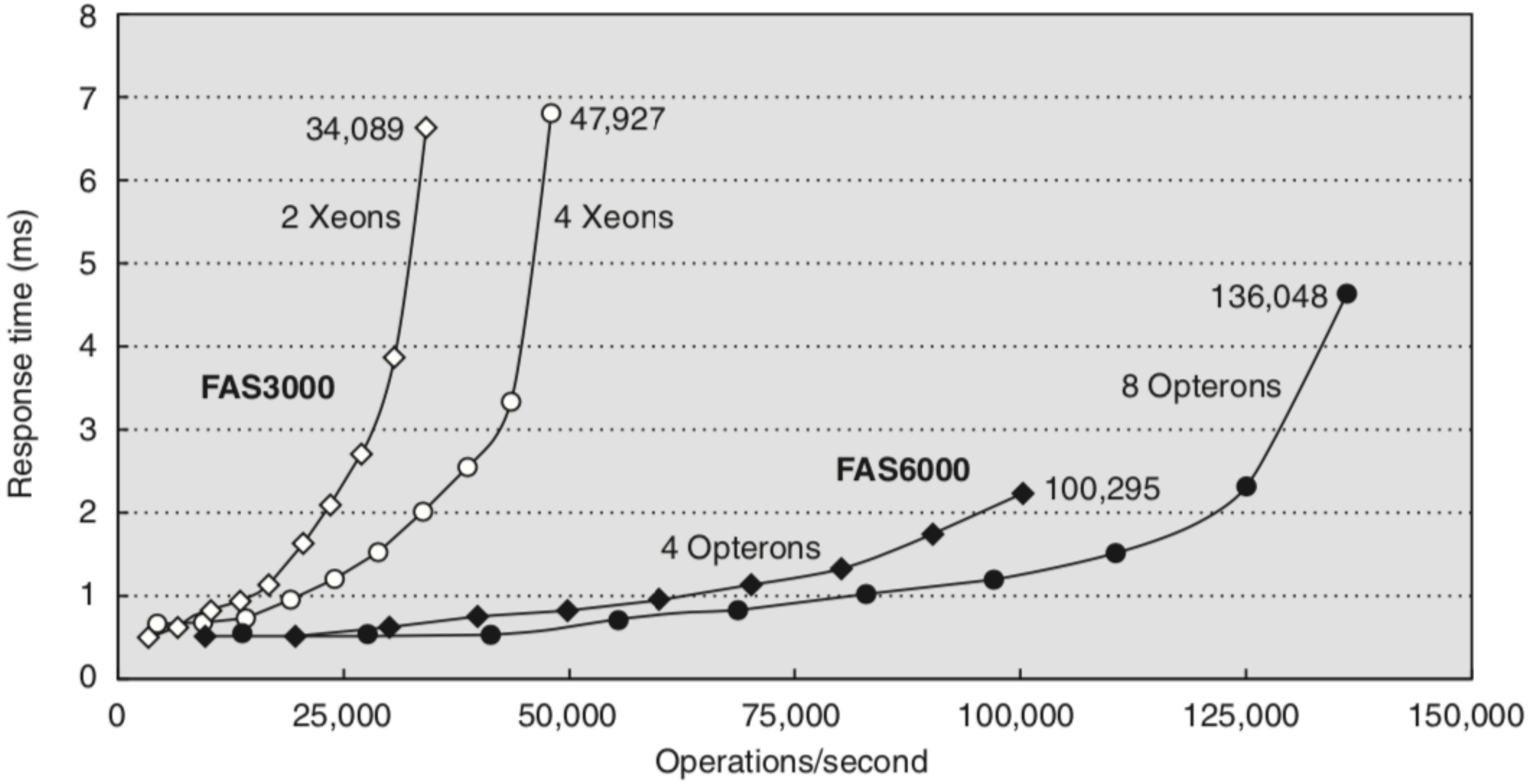

Figure D.13 SPEC SFS97_R1 performance for the NetApp FAS3050c NFS servers in

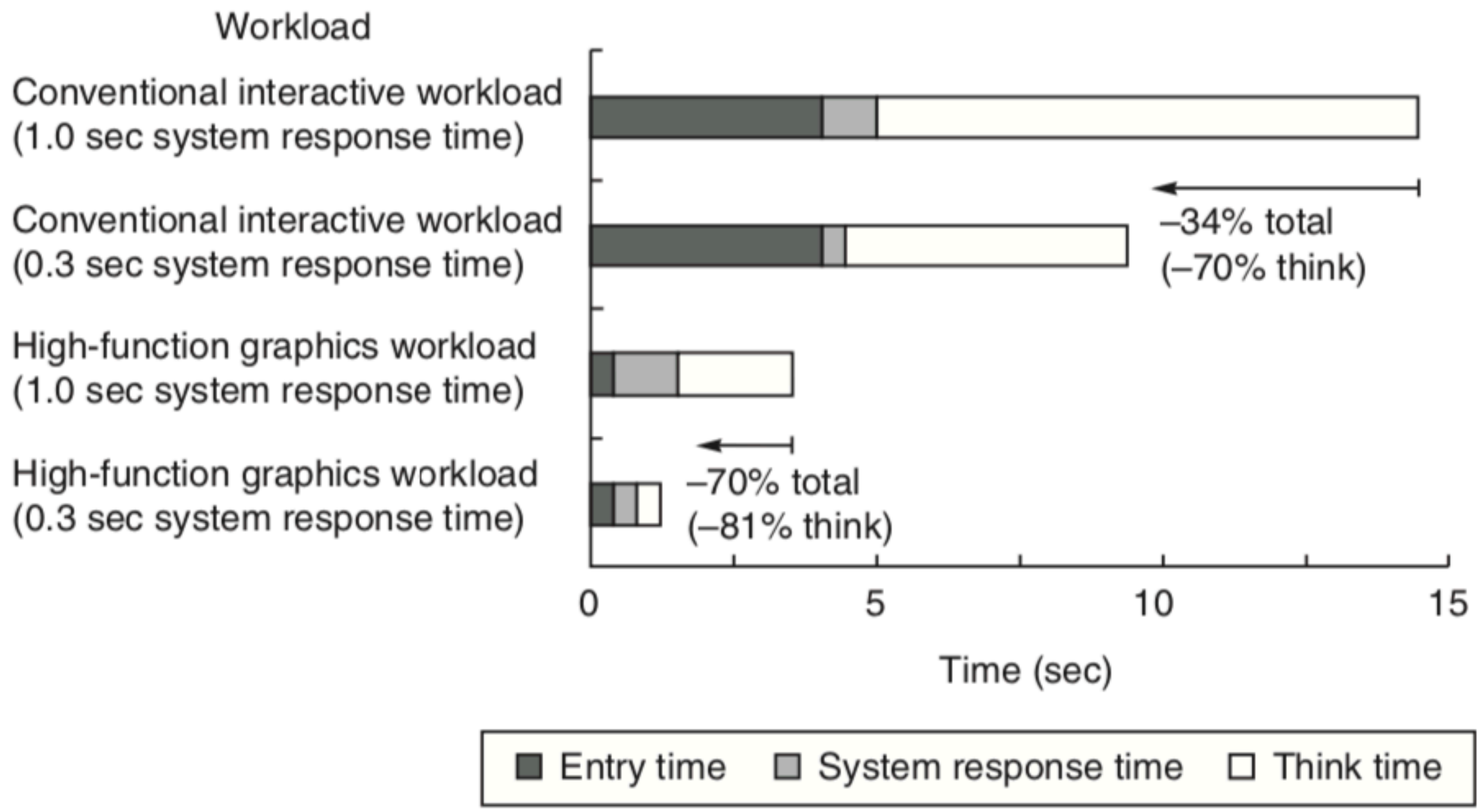

Figure D.10 A user transaction with an interactive computer divided into entry time, system response time, and user think time for a conventional system and graphics

I/O benchmark	Response time restriction	Throughput metric
TPC-C: Complex Query OLTP	$\geq$90% of transaction must meet response time limit; 5 seconds for most types of transactions	New order transactions per minute
TPC-W: Transactional Web benchmark	$\geq$90% of Web interactions must meet response time limit; 3 seconds for most types of Web interactions	Web interactions per second
SPECsfs97	Average response time $\leq$40 ms	NFS operations per second

Figure D.11 Response time restrictions for three I/O benchmarks.

Figure D.15 Treating the I/O system as a black box. This leads to a simple but important observation: If the system is in steady state, then the number of tasks entering the system must equal the number of tasks leaving the system. This *flow-balanced* state is necessary but not sufficient for steady state. If the system has been observed or measured for a sufficiently long time and mean waiting times stabilize, then we say that the system has reached steady state.

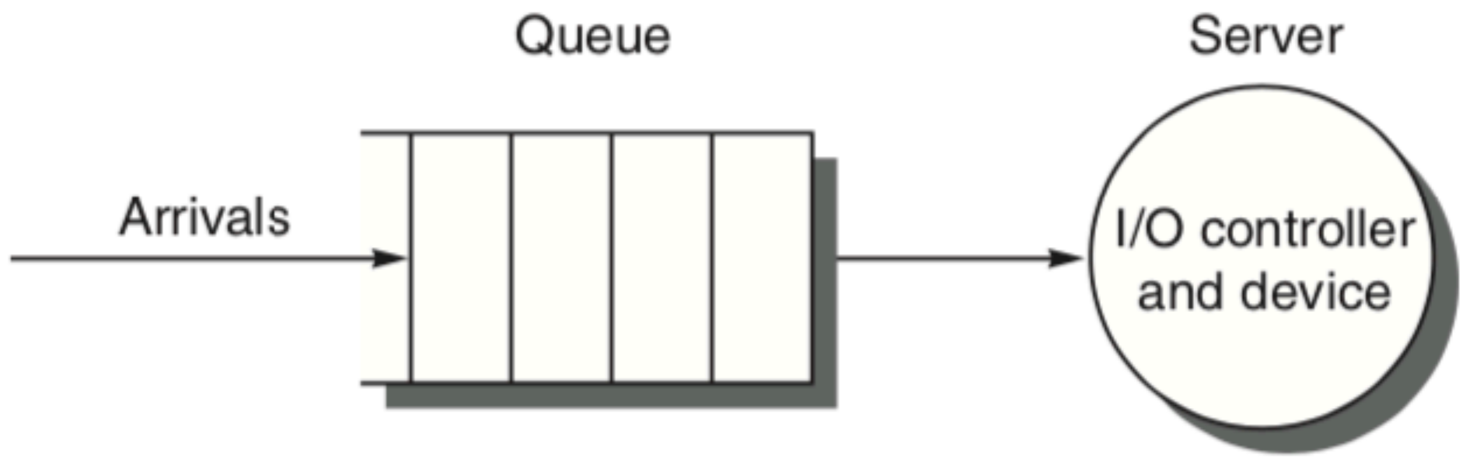

Figure D.16 The single-server model for this section. In this situation, an I/O request "departs" by being completed by the server.

1. *Entry time*—The time for the user to enter the command.
2. *System response time*—The time between when the user enters the command and the complete response is displayed.
3. *Think time*—The time from the reception of the response until the user begins to enter the next command.

Little's Law Equations

From HnP6

Little's law and a series of definitions lead to several useful equations:

- $\text{Time}_{\text{server}}$—Average time to service a task; average service rate is $1/\text{Time}_{\text{server}}$, traditionally represented by the symbol μ in many queuing texts.
- $\text{Time}_{\text{queue}}$—Average time per task in the queue.
- $\text{Time}_{\text{system}}$—Average time/task in the system, or the response time, which is the sum of $\text{Time}_{\text{queue}}$ and $\text{Time}_{\text{server}}$.
- Arrival rate—Average number of arriving tasks/second, traditionally represented by the symbol λ in many queuing texts.
- $\text{Length}_{\text{server}}$—Average number of tasks in service.
- $\text{Length}_{\text{queue}}$—Average length of queue.
- $\text{Length}_{\text{system}}$—Average number of tasks in system, which is the sum of $\text{Length}_{\text{queue}}$ and $\text{Length}_{\text{server}}$.

One common misunderstanding can be made clearer by these definitions: whether the question is how long a task must wait in the queue before service starts ($\text{Time}_{\text{queue}}$) or how long a task takes until it is completed ($\text{Time}_{\text{system}}$). The latter term is what we mean by response time, and the relationship between the terms is $\text{Time}_{\text{system}} = \text{Time}_{\text{queue}} + \text{Time}_{\text{server}}$.

Example Suppose an I/O system with a single disk gets on average 50 I/O requests per second. Assume the average time for a disk to service an I/O request is 10 ms. What is the utilization of the I/O system?

Answer Using the equation above, with 10 ms represented as 0.01 seconds, we get: 50

$$\text{Server utilization} = \text{Arrival rate} \times \text{Time}_{\text{server}} = \frac{50}{\text{sec}} \times 0.01\,\text{sec} = 0.50$$

Therefore, the I/O system utilization is 0.5.

Example Using the definitions and formulas above, derive the average time waiting in the queue ($\text{Time}_{\text{queue}}$) in terms of the average service time ($\text{Time}_{\text{server}}$) and server utilization.

Answer All tasks in the queue ($\text{Length}_{\text{queue}}$) ahead of the new task must be completed before the task can be serviced; each takes on average $\text{Time}_{\text{server}}$. If a task is at the server, it takes average residual service time to complete. The chance the server is busy is *server utilization*; hence, the expected time for service is Server utilization $\times$ Average residual service time. This leads to our initial formula:

$$\begin{aligned}\text{Time}_{\text{queue}} = {} & \text{Length}_{\text{queue}} \times \text{Time}_{\text{server}} \\ & + \text{Server utilization} \times \text{Average residual service time}\end{aligned}$$

Replacing the average residual service time by its definition and $\text{Length}_{\text{queue}}$ by Arrival rate $\times$ $\text{Time}_{\text{queue}}$ yields

$$\begin{aligned}\text{Time}_{\text{queue}} = {} & \text{Server utilization} \times \left[1/2 \times \text{Time}_{\text{server}} \times \left(1 + \text{C}^2\right)\right] \\ & + \left(\text{Arrival rate} \times \text{Time}_{\text{queue}}\right) \times \text{Time}_{\text{server}}\end{aligned}$$

Since this section is concerned with exponential distributions, C^2 is 1. Thus

$$\text{Time}_{\text{queue}} = \text{Server utilization} \times \text{Time}_{\text{server}} + \left(\text{Arrival rate} \times \text{Time}_{\text{queue}}\right) \times \text{Time}_{\text{server}}$$

Rearranging the last term, let us replace Arrival rate $\times$ $\text{Time}_{\text{server}}$ by Server utilization:

$$\begin{aligned}\text{Time}_{\text{queue}} & = \text{Server utilization} \times \text{Time}_{\text{server}} + (\text{Arrival rate} \times \text{Time}_{\text{server}}) \times \text{Time}_{\text{queue}} \\ & = \text{Server utilization} \times \text{Time}_{\text{server}} + \text{Server utilization} \times \text{Time}_{\text{queue}}\end{aligned}$$

Rearranging terms and simplifying gives us the desired equation:

$$\begin{aligned}\text{Time}_{\text{queue}} & = \text{Server utilization} \times \text{Time}_{\text{server}} + \text{Server utilization} \times \text{Time}_{\text{queue}} \\ \text{Time}_{\text{queue}} - \text{Server utilization} \times \text{Time}_{\text{queue}} & = \text{Server utilization} \times \text{Time}_{\text{server}} \\ \text{Time}_{\text{queue}} \times (1 - \text{Server utilization}) & = \text{Server utilization} \times \text{Time}_{\text{server}} \\ \text{Time}_{\text{queue}} & = \text{Time}_{\text{server}} \times \frac{\text{Server utilization}}{(1 - \text{Server utilization})}\end{aligned}$$

Example For the system in the example on page D-26, which has a server utilization of 0.5, what is the mean number of I/O requests in the queue?

Answer Using the equation above,

$$\text{Length}_{\text{queue}} = \frac{\text{Server utilization}^2}{(1 - \text{Server utilization})} = \frac{0.5^2}{(1 - 0.5)} = \frac{0.25}{0.50} = 0.5$$

Therefore, there are 0.5 requests on average in the queue.

Queuing Model Assumptions

Let's review the assumptions about the queuing model:

- The system is in equilibrium.
- The times between two successive requests arriving, called the *interarrival times*, are exponentially distributed, which characterizes the arrival rate mentioned earlier.
- The number of sources of requests is unlimited. (This is called an *infinite population model* in queuing theory; finite population models are used when arrival rates vary with the number of jobs already in the system.)
- The server can start on the next job immediately after finishing the prior one.
- There is no limit to the length of the queue, and it follows the first in, first out order discipline, so all tasks in line must be completed.
- There is one server.

Such a queue is called *M/M/1:*

M = exponentially random request arrival ($C^2 = 1$), with *M* standing for A. A. Markov, the mathematician who defined and analyzed the memoryless processes mentioned earlier

M = exponentially random service time ($C^2 = 1$), with *M* again for Markov

1 = single server

Example Suppose a processor sends 40 disk I/Os per second, these requests are exponentially distributed, and the average service time of an older disk is 20 ms. Answer the following questions:

1. On average, how utilized is the disk?
2. What is the average time spent in the queue?
3. What is the average response time for a disk request, including the queuing time and disk service time?

Answer Let's restate these facts:

Average number of arriving tasks/second is 40.

Average disk time to service a task is 20 ms (0.02 sec).

The server utilization is then

$$\text{Server utilization} = \text{Arrival rate} \times \text{Time}_{\text{server}} = 40 \times 0.02 = 0.8$$

Since the service times are exponentially distributed, we can use the simplified formula for the average time spent waiting in line:

$$\text{Time}_{\text{queue}} = \text{Time}_{\text{server}} \times \frac{\text{Server utilization}}{(1 - \text{Server utilization})}$$

$$= 20\text{ ms} \times \frac{0.8}{1 - 0.8} = 20 \times \frac{0.8}{0.2} = 20 \times 4 = 80\text{ ms}$$

The average response time is

$$\text{Time system} = \text{Time}_{\text{queue}} + \text{Time}_{\text{server}} = 80 + 20\text{ ms} = 100\text{ ms}$$

Thus, on average we spend 80% of our time waiting in the queue!

Example Suppose we get a new, faster disk. Recalculate the answers to the questions above, assuming the disk service time is 10 ms.

Answer The disk utilization is then

$$\text{Server utilization} = \text{Arrival rate} \times \text{Time}_{\text{server}} = 40 \times 0.01 = 0.4$$

The formula for the average time spent waiting in line:

$$\begin{aligned}\text{Time}_{\text{queue}} &= \text{Time}_{\text{server}} \times \frac{\text{Server utilization}}{(1 - \text{Server utilization})} \\ &= 10\text{ ms} \times \frac{0.4}{1-0.4} = 10 \times \frac{0.4}{0.6} = 10 \times \frac{2}{3} = 6.7\text{ ms}\end{aligned}$$

The average response time is $10 + 6.7$ ms or 16.7 ms, 6.0 times faster than the old response time even though the new service time is only 2.0 times faster.

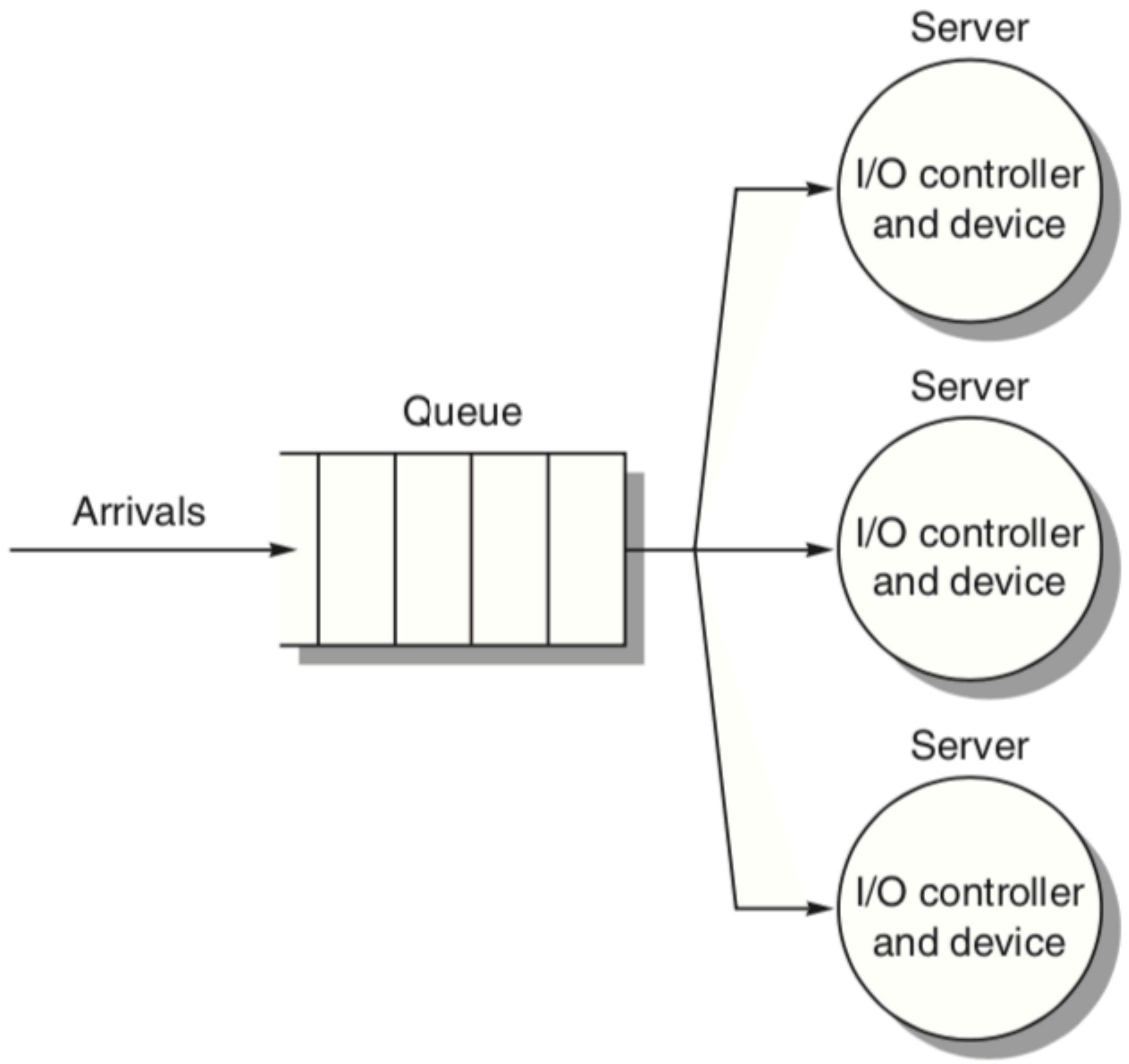

Figure D.17 The M/M/m multiple-server model.

Example Suppose instead of a new, faster disk, we add a second slow disk and duplicate the data so that reads can be serviced by either disk. Let's assume that the requests are all reads. Recalculate the answers to the earlier questions, this time using an M/M/m queue.

Answer The average utilization of the two disks is then

$$\text{Server utilization} = \frac{\text{Arrival rate} \times \text{Time}_{\text{server}}}{\text{N}_{\text{servers}}} = \frac{40 \times 0.02}{2} = 0.4$$

We first calculate the probability of no tasks in the queue:

$$\text{Prob}_{0\ \text{tasks}} = \left[1 + \frac{(2 \times \text{Utilization})^2}{2! \times (1 - \text{Utilization})} + \sum_{n=1}^{1} \frac{(2 \times \text{Utilization})^n}{n!}\right]^{-1}$$

$$= \left[1 + \frac{(2 \times 0.4)^2}{2 \times (1 - 0.4)} + (2 \times 0.4)\right]^{-1} = \left[1 + \frac{0.640}{1.2} + 0.800\right]^{-1}$$

$$= [1 + 0.533 + 0.800]^{-1} = 2.333^{-1}$$

We use this result to calculate the probability of tasks in the queue:

$$\text{Prob}_{\text{tasks} \geq \text{N}_{\text{servers}}} = \frac{2 \times \text{Utilization}^2}{2! \times (1 - \text{Utilization})} \times \text{Prob}_{0\ \text{tasks}}$$

$$= \frac{(2 \times 0.4)^2}{2 \times (1 - 0.4)} \times 2.333^{-1} = \frac{0.640}{1.2} \times 2.333^{-1}$$

$$= 0.533/2.333 = 0.229$$

Finally, the time waiting in the queue:

$$\text{Time}_{\text{queue}} = \text{Time}_{\text{server}} \times \frac{\text{Prob}_{\text{tasks} \geq \text{N}_{\text{servers}}}}{\text{N}_{\text{servers}} \times (1 - \text{Utilization})}$$

$$= 0.020 \times \frac{0.229}{2 \times (1 - 0.4)} = 0.020 \times \frac{0.229}{1.2}$$

$$= 0.020 \times 0.190 = 0.0038$$

The average response time is $20 + 3.8$ ms or 23.8 ms. For this workload, two disks cut the queue waiting time by a factor of 21 over a single slow disk and a factor of 1.75 versus a single fast disk. The mean service time of a system with a single fast disk, however, is still 1.4 times faster than one with two disks since the disk service time is 2.0 times faster.