Example 1

Consider a service, a remote storage, deployed on a single server *S*.
Such a service is enforced by a software component A (single thread) which is able to manage each request (i.e. a retrieval of data) in *0.1s* on average. Let us assume that 1) on average *5 requests per second* arrive at *S*, that 2) the arrivals are exponentially distributed, and that 3) the service time of *A* also follows an exponential distribution.
Compute the expected response time perceived by an user of the service, i.e., the expected amount of time an user waits to get a response for a request.

Example 2

Consider a service, a remote storage, deployed on a single server S.
Such a service is enforced by two software components *A* and *B* (both single-threaded) that manage the requests sequentially (i.e., each request is processed first by *A* and then by *B*). On average, software components *A* and *B* can handle *10* and *6* requests per second, respectively. Assume that 1) an average of *5 requests per second* arrive at *S*, 2) the arrivals are exponentially distributed, and 3) the service time of *A* and *B* also follow an exponential distribution.
Compute the expected response time perceived by a user of the service, i.e., the expected time a user waits to receive a response to a request.
Can the system handle a workload of *8 requests per second*?
If not, is it possible to handle the increased workload by vertical scaling one the component doubling its service rate? What will be the resulting response time?

Example 3

Consider a remote storage service deployed on a single server S.
Such a service is enforced by two software components *A* and *B* (both single-threaded). Component A caches part of all the data stored on S, component B stores the whole data. The probability of a cache miss is 0.2. Component A needs to process a request both in case of a cache-hit and cache-miss. On average, software components *A* and *B* can handle *10* and *6* requests per second, respectively. Assume that 1) an average of *5 requests per second* arrive at *S*, 2) the arrivals are exponentially distributed, and 3) the service time of *A* and *B* also follow an exponential distribution.
Compute the expected response time perceived by a user of the service, i.e., the expected time a user waits to receive a response to a request.
Can the system handle a workload of *8 requests per second*?
If not, is it possible to handle the increased workload by vertical scaling one the component doubling its service rate? What will be the resulting response time?

Example 1

Consider a service, a remote storage, deployed on a single server *S*.
Such a service is enforced by a software component A (single thread) which is able to manage each request (i.e. a retrieval of data) in *0.1s* on average. Let us assume that 1) on average *5 requests per second* arrive at *S*, that 2) the arrivals are exponentially distributed, and that 3) the service time of *A* also follows an exponential distribution.
Compute the expected response time perceived by an user of the service, i.e., the expected amount of time an user waits to get a response for a request.

1 server → M M 1

IMPORTANT

arrival rate : $\lambda = 5$ req|s

service rate : $\mu = 0.1\ s = \frac{1}{0.1} = 10$ req|s

$\lambda < \mu$: system STABLE

Response time $MM1 = \frac{1}{K\mu - \lambda} = \frac{1}{10-5} = \frac{1}{5} = 0{,}2\ s$

Example 2

Consider a service, a remote storage, deployed on a single server S.
Such a service is enforced by two software components *A* and *B* (both single-threaded) that manage the requests sequentially (i.e., each request is processed first by *A* and then by *B*). On average, software components *A* and *B* can handle *10* and *6* requests per second, respectively.
Assume that 1) an average of *5 requests per second* arrive at *S*, 2) the arrivals are exponentially distributed, and 3) the service time of *A* and *B* also follow an exponential distribution.
Compute the expected response time perceived by a user of the service, i.e., the expected time a user waits to receive a response to a request.
Can the system handle a workload of *8 requests per second*?
If not, is it possible to handle the increased workload by vertical scaling one the component doubling its service rate? What will be the resulting response time?

handle

MM1

service rate

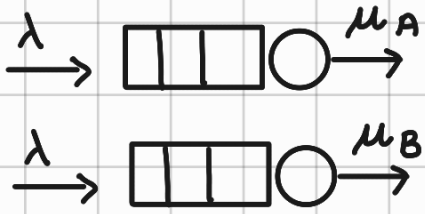
arrival rate

A: $\mu_A = 10$ req/s

B: $\mu_B = 6$ req/s

$\lambda = 5$ req/s

System = A + B



$$MM1 = \frac{1}{k\mu - \lambda}$$

$$MM1_A = \frac{1}{\mu_A - \lambda} = \frac{1}{10-5} = \frac{1}{5} = 0{,}2\,s$$

$$MM1_B = \frac{1}{\mu_B - \lambda} = \frac{1}{6-5} = 1\,s$$

→ B acts as a BOTTLENECK

↓

we have to improve B

① $R = 1{,}2\,s$

② $\lambda = 8$ req/s

No

vertical scaling

③ $2 \cdot \mu_B = 2 \cdot 6$ req/s $\rightarrow$ $\mu_B = 12$ req/s

WORKLOAD
$\lambda = 8$ req/s

$$NM1_A = \frac{1}{\mu_A - \lambda} = \frac{1}{10-8} = 0{,}5\ s$$

$$NM1_B = \frac{1}{\mu_B - \lambda} = \frac{1}{12-8} = 0{,}25\ s$$

$R = 0{,}75$ s

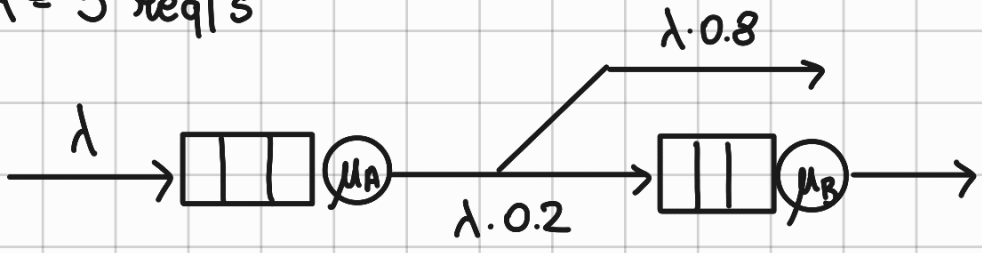
Example 3

Consider a remote storage service deployed on a single server S.
Such a service is enforced by two software components *A* and *B* (both single-threaded).
Component A caches part of all the data stored on S, component B stores the whole data. The probability of a cache miss is 0.2. Component A needs to process a request both in case of a cache-hit and cache-miss. On average, software components *A* and *B* can handle *10* and *6* requests per second, respectively. Assume that 1) an average of *5 requests per second* arrive at *S*, 2) the arrivals are exponentially distributed, and 3) the service time of *A* and *B* also follow an exponential distribution.
Compute the expected response time perceived by a user of the service, i.e., the expected time a user waits to receive a response to a request.
Can the system handle a workload of *8 requests per second*?
If not, is it possible to handle the increased workload by vertical scaling one the component doubling its service rate? What will be the resulting response time?

$\mu_A = 10$ req/s $\qquad P_{cache\ miss} = 0.2$

$\mu_B = 6$ req/s

$\lambda = 5$ req/s



$$R_{cache\ hit} = R_A = \frac{1}{10-5} = 0{,}2\ s$$

$$R_{cache\ miss} = R_A + R_B = \frac{1}{10-5} + \frac{1}{\mu_B - \lambda \cdot 0.2}$$

$$= 0{,}2 + \frac{1}{6-1} = 0{,}4\ s$$

$$R = 0{,}8 \cdot R_{HIT} + 0{,}2 \cdot R_{MISS}$$