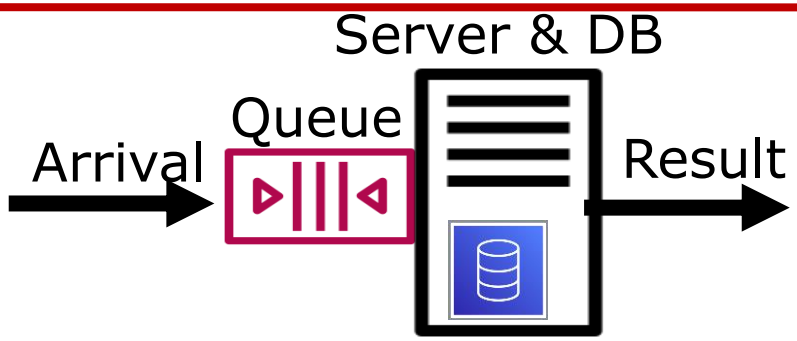
Carnegie Mellon University

Performance

Outline

- **Introduction**
- Pipeline of servers
- Scaling

Basics



- Requests for service arrive at the system
- They are placed in a queue
- They are served in turn and a response is generated.
- Latency: time between request arrival and generation of response
- Throughput: number of requests that can be served in a unit time

More detail

- The request for service may involve database access as well as computation.
- Database access involves network requests.
- Accesses across a network are slower than computation from memory.

Reducing latency for a single server

- With a single server, latency may be reduced by
  - Reducing the time for required computation
    - Use a better algorithm
    - Reduce system overhead.
    - Use a host with more resources (faster processor, more memory, more disk)
  - Changing some database accesses to memory accesses (caching)

Concern

- Caching may result in inconsistency between data in cache and data in database.
  - Consistency must be managed

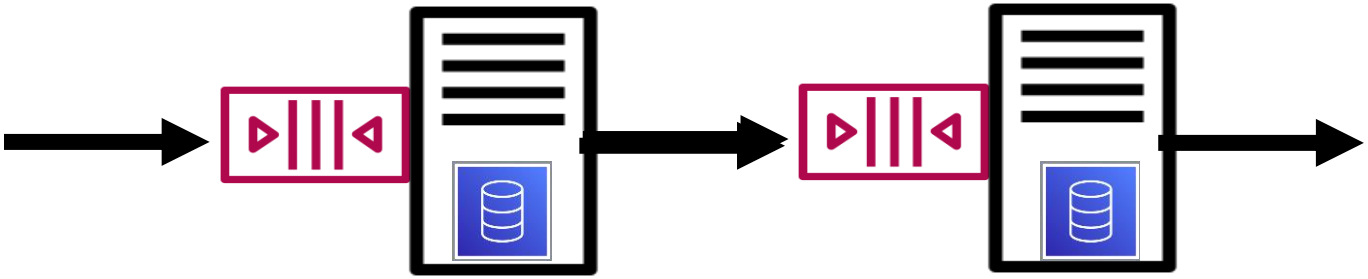
Estimating latency

- The latency of a server can be estimated
  - based on history of similar servers
  - Through development of a prototype.

Outline

- Introduction
- **Pipeline of servers**
- Scaling

Two servers



- Output of first server is input of second.
- Output of server 1 travels over a network to server 2.
- Latency is time at server 1 + network transport time + time at server 2.

Reducing the latency of two servers

- Latency can be reduced by either reducing the latency of a server or reducing the network transport time.
- Reducing the latency of a server is as with a single server
- Reducing the network transport time can be achieved by
  - Using a different protocol
  - Using a faster network
  - Reducing the volume of information sent.

Pipeline

- Multiple servers can be strung together into a pipeline
- Reducing the latency of a pipeline is achieved by reducing the latency of the servers or the network transport.

Relation to requirements

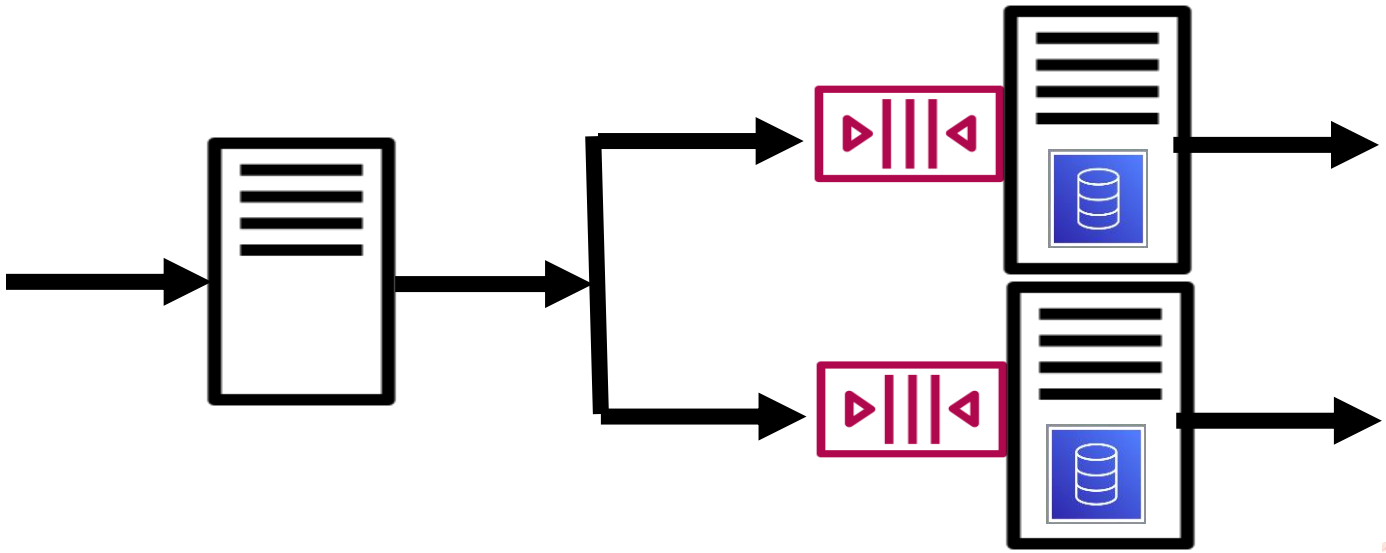
- Suppose there is a latency requirement for a set of cascading servers.
- How does that translate into requirements for the individual servers?
- Each server and network transport is given a budget where the sum of the budgets is less than overall requirement.

Outline

- Introduction
- Pipeline of servers
- **Scaling**

Horizontal scaling

- Reducing the overall latency for a collection of independent requests can be done by having multiple servers running in parallel.



Horizontal scaling

- Requests arrive at a distribution mechanism
- The distribution mechanism sends each request to one of the parallel servers.
- Each of the parallel servers may be the beginning of a pipeline
- The assumption is that the servers are equivalent.

State

- Horizontal scaling works best if the servers are stateless.
- The database may be shared across servers and may be used to store necessary state.
- Data consistency is a concern and must be managed.

Summary

- Latency is the time between a request arriving at the system and a response being generated.
- Meeting a request with a pipeline of servers means that a budget for each of the servers must be set
- Horizontal scaling involves creating multiple servers and distributing requests among them.
- Management of statement must be considered during the design.