Grid Computing

Network and Client/Server Essentials  
Paul A. Farrell  
Fall 2006

The Grid: Core Technologies  
Maozhen Li, Mark Baker  
From material by Amy Apon at U. Arkansas

Brief Network Tutorial

Application (e.g., FTP, HTTP, telnet)  
Transport (e.g., TCP, UDP)  
Network (e.g., IP)  
Data Link (e.g., Ethernet)  
Physical (e.g., cables, etc.)

When a message is sent

- The application constructs a message

When a message is sent

- The message is packaged (encapsulated) with a header from the transport layer (e.g., TCP) and sent to the network layer

TCP  user data
When a message is sent

- The network layer adds a header

![IP packet](image1)

When a message is sent

- The data link layer adds a header, and the frame is sent out on the network

![Ethernet frame](image2)

When a message is sent

1. Application builds message in user memory
2. Message is copied to kernel and TCP/IP and Ethernet headers are added
3. Message is sent onto network to receiver
4. Message arrives to user memory and the application is notified

Connectionless or connection-oriented

- If a protocol is connection-oriented then the client and server must connect before communication takes place – like a telephone call
- If a protocol is connectionless then there is no connection, just messaging – like sending a letter in the mail

![Connectionless or connection-oriented](image3)
### Client/Server Model

<table>
<thead>
<tr>
<th>Server</th>
<th>Client</th>
</tr>
</thead>
</table>
| • Starts first  
• Waits for contact from a client  
• Responds to requests | • Starts second  
• Contacts a server with a request  
• Waits for response from server |

### Types of Servers

<table>
<thead>
<tr>
<th>Iterative</th>
<th>Concurrent</th>
</tr>
</thead>
</table>
| Stateful  | iterative  
stateful  
concurrent  
stateful  | Stateless  
iteration  
stateless  
concurrent  
stateless  |

### Stateful Server

- Maintains some information between requests
- Requires smaller messages, since some information is kept between contacts
- May become confused if a connection terminates abnormally (if the design is not fault tolerant)
- Example: FTP

### Stateless Server

- Requires larger messages. That is, the message must contain all information about the request since no state information is kept.
- Example: HTTP
Iterative Server

while (1) {
    accept a connection (or request) from a client
    service the client
    close the connection (if necessary)
}

Concurrent Server

while (1) {
    accept a connection/request from client
    start a new thread to handle this client
    /* the thread must close the connection! */
}

Internet Addressing

- Suppose you type:
  http://www.cs.kent.edu/~farrell

This invokes the HTTP protocol (over TCP/IP), and the computer "www.cs.kent.edu" is sent a message

- Before the message can be sent the name www.cs.kent.edu must be resolved to an IP address
- A Domain Name Server (DNS) may be called to find the IP address of www.cs.kent.edu
- Each IP machine is usually configured with the name of a DNS server.
- Some IP names and addresses can also be stored in /etc/hostfile
- The IP address appears in the IP header of the message
Internet Addressing

Find the home page of user farrell

http://www.cs.kent.edu/~farrell/

Contact the HTTP server on the computer named www.cs.kent.edu

Same as IP address 131.123.35.61

"http" says: send the message to port 80

- The message includes both a host address and a port number!
- The port number is in the TCP header
- Different servers listen on different ports
- This allows messages to be routed to multiple servers on a machine
- The HTTP server listens to port 80
- The HTTP server responds when a client contacts it

⇒ You can write a server that listens to any port not already in use!

- A port number is a 16-bit integer. Ports below 1024 are reserved for system use.
- Well-known ports include FTP, Telnet, SMTP, etc.
- In Linux the standard (well-known) services are specified in /etc/services