

Using Sockets

Communication between Programs

client/server interaction $\sim$ telephone exchange

Remote Server and Client

getting IP addresses of computers

code for remote client and server

Two Clients and one Talk Host

swapping information between two clients

code for client and server

Monte Carlo for π

a pleasingly parallel computation

MCS 275 Lecture 25
Programming Tools and File Management
Jan Vershelde, 12 March 2008

MCS 275 L-25

12 March 2008

Communication
between Programs

client/server interaction $\sim$
telephone exchange

Remote Server
and Client

getting IP addresses of
computers

code for remote client and
server

Two Clients and
one Talk Host

swapping information
between two clients

code for client and server

Monte Carlo for π

a pleasingly parallel
computation

Using Sockets

MCS 275 L-25

12 March 2008

Communication between Programs

client/server interaction $\sim$ telephone exchange

Communication
between Programs

client/server interaction $\sim$
telephone exchange

Remote Server and Client

getting IP addresses of computers
code for remote client and server

Remote Server
and Client

getting IP addresses of
computers
code for remote client and
server

Two Clients and one Talk Host

swapping information between two clients
code for client and server

Two Clients and
one Talk Host

swapping information
between two clients
code for client and server

Monte Carlo for π

a pleasingly parallel computation

Monte Carlo for π

a pleasingly parallel
computation

Analogy with Telephone Exchange

client/server communication with sockets

Analogy between a telephone exchange and sockets:

1. dial company on 1-312-666-9000
connect to IP address 127.0.0.1
2. call answered by reception
connection established to remote host
3. ask for computer center
route using specified port (8732)
4. call answered by computer center
server handles request from client
5. hang up the phone
close sockets

MCS 275 L-25

12 March 2008

Communication
between Programs

client/server interaction ~
telephone exchange

Remote Server
and Client

getting IP addresses of
computers
code for remote client and
server

Two Clients and
one Talk Host

swapping information
between two clients
code for client and server

Monte Carlo for π

a pleasingly parallel
computation

Analogy with Telephone Exchange

client/server communication with sockets

MCS 275 L-25

12 March 2008

Analogy between a telephone exchange and sockets:

1. dial company on 1-312-666-9000
connect to IP address 127.0.0.1
2. call answered by reception
connection established to remote host
3. ask for computer center
route using specified port (8732)
4. call answered by computer center
server handles request from client
5. hang up the phone
close sockets

Communication
between Programs

client/server interaction ~
telephone exchange

Remote Server
and Client

getting IP addresses of
computers
code for remote client and
server

Two Clients and
one Talk Host

swapping information
between two clients
code for client and server

Monte Carlo for π

a pleasingly parallel
computation

Analogy with Telephone Exchange

client/server communication with sockets

MCS 275 L-25

12 March 2008

Analogy between a telephone exchange and sockets:

1. dial company on 1-312-666-9000
connect to IP address 127.0.0.1
2. call answered by reception
connection established to remote host
3. ask for computer center
route using specified port (8732)
4. call answered by computer center
server handles request from client
5. hang up the phone
close sockets

Communication
between Programs

client/server interaction ~
telephone exchange

Remote Server
and Client

getting IP addresses of
computers
code for remote client and
server

Two Clients and
one Talk Host

swapping information
between two clients
code for client and server

Monte Carlo for π

a pleasingly parallel
computation

Analogy with Telephone Exchange

client/server communication with sockets

MCS 275 L-25

12 March 2008

Analogy between a telephone exchange and sockets:

1. dial company on 1-312-666-9000
connect to IP address 127.0.0.1
2. call answered by reception
connection established to remote host
3. ask for computer center
route using specified port (8732)
4. call answered by computer center
server handles request from client
5. hang up the phone
close sockets

Communication
between Programs

client/server interaction ~
telephone exchange

Remote Server
and Client

getting IP addresses of
computers
code for remote client and
server

Two Clients and
one Talk Host

swapping information
between two clients
code for client and server

Monte Carlo for π

a pleasingly parallel
computation

Analogy with Telephone Exchange

client/server communication with sockets

MCS 275 L-25

12 March 2008

Analogy between a telephone exchange and sockets:

1. dial company on 1-312-666-9000
connect to IP address 127.0.0.1
2. call answered by reception
connection established to remote host
3. ask for computer center
route using specified port (8732)
4. call answered by computer center
server handles request from client
5. hang up the phone
close sockets

Communication
between Programs

client/server interaction ~
telephone exchange

Remote Server
and Client

getting IP addresses of
computers
code for remote client and
server

Two Clients and
one Talk Host

swapping information
between two clients
code for client and server

Monte Carlo for π

a pleasingly parallel
computation

Methods on Socket Objects

MCS 275 L-25

12 March 2008

Communication
between Programs

client/server interaction ~
telephone exchange

Remote Server
and Client

getting IP addresses of
computers

code for remote client and
server

Two Clients and
one Talk Host

swapping information
between two clients

code for client and server

Monte Carlo for π

a pleasingly parallel
computation

Most commonly used methods:

method	description
<code>accept()</code>	accepts connection and returns new socket for passing data
<code>bind()</code>	binds a socket to an address
<code>close()</code>	closes the socket
<code>connect(a)</code>	connects to address a
<code>listen(c)</code>	listend for connections
<code>recv(b)</code>	receives data in buffer of size b
<code>send(d)</code>	sends data in d

Methods on Socket Objects

MCS 275 L-25

12 March 2008

Most commonly used methods:

method	description
<code>accept()</code>	accepts connection and returns new socket for passing data
<code>bind()</code>	binds a socket to an address
<code>close()</code>	closes the socket
<code>connect(a)</code>	connects to address a
<code>listen(c)</code>	listend for connections
<code>recv(b)</code>	receives data in buffer of size b
<code>send(d)</code>	sends data in d

Communication
between Programs

client/server interaction ~
telephone exchange

Remote Server
and Client

getting IP addresses of
computers

code for remote client and
server

Two Clients and
one Talk Host

swapping information
between two clients

code for client and server

Monte Carlo for π

a pleasingly parallel
computation

Methods on Socket Objects

MCS 275 L-25

12 March 2008

Most commonly used methods:

method	description
<code>accept()</code>	accepts connection and returns new socket for passing data
<code>bind()</code>	binds a socket to an address
<code>close()</code>	closes the socket
<code>connect(a)</code>	connects to address a
<code>listen(c)</code>	listend for connections
<code>recv(b)</code>	receives data in buffer of size b
<code>send(d)</code>	sends data in d

Communication
between Programs

client/server interaction ~
telephone exchange

Remote Server
and Client

getting IP addresses of
computers

code for remote client and
server

Two Clients and
one Talk Host

swapping information
between two clients

code for client and server

Monte Carlo for π

a pleasingly parallel
computation

Methods on Socket Objects

MCS 275 L-25

12 March 2008

Most commonly used methods:

method	description
<code>accept()</code>	accepts connection and returns new socket for passing data
<code>bind()</code>	binds a socket to an address
<code>close()</code>	closes the socket
<code>connect(a)</code>	connects to address a
<code>listen(c)</code>	listend for connections
<code>recv(b)</code>	receives data in buffer of size b
<code>send(d)</code>	sends data in d

Communication
between Programs

client/server interaction ~
telephone exchange

Remote Server
and Client

getting IP addresses of
computers

code for remote client and
server

Two Clients and
one Talk Host

swapping information
between two clients

code for client and server

Monte Carlo for π

a pleasingly parallel
computation

Methods on Socket Objects

MCS 275 L-25

12 March 2008

Most commonly used methods:

method	description
<code>accept()</code>	accepts connection and returns new socket for passing data
<code>bind()</code>	binds a socket to an address
<code>close()</code>	closes the socket
<code>connect(a)</code>	connects to address a
<code>listen(c)</code>	listend for connections
<code>recv(b)</code>	receives data in buffer of size b
<code>send(d)</code>	sends data in d

Communication
between Programs

client/server interaction ~
telephone exchange

Remote Server
and Client

getting IP addresses of
computers

code for remote client and
server

Two Clients and
one Talk Host

swapping information
between two clients

code for client and server

Monte Carlo for π

a pleasingly parallel
computation

Methods on Socket Objects

MCS 275 L-25

12 March 2008

Most commonly used methods:

method	description
<code>accept()</code>	accepts connection and returns new socket for passing data
<code>bind()</code>	binds a socket to an address
<code>close()</code>	closes the socket
<code>connect(a)</code>	connects to address a
<code>listen(c)</code>	listend for connections
<code>recv(b)</code>	receives data in buffer of size b
<code>send(d)</code>	sends data in d

Communication
between Programs

client/server interaction ~
telephone exchange

Remote Server
and Client

getting IP addresses of
computers

code for remote client and
server

Two Clients and
one Talk Host

swapping information
between two clients

code for client and server

Monte Carlo for π

a pleasingly parallel
computation

Methods on Socket Objects

MCS 275 L-25

12 March 2008

Most commonly used methods:

method	description
<code>accept()</code>	accepts connection and returns new socket for passing data
<code>bind()</code>	binds a socket to an address
<code>close()</code>	closes the socket
<code>connect(a)</code>	connects to address a
<code>listen(c)</code>	listend for connections
<code>recv(b)</code>	receives data in buffer of size b
<code>send(d)</code>	sends data in d

Communication
between Programs

client/server interaction ~
telephone exchange

Remote Server
and Client

getting IP addresses of
computers

code for remote client and
server

Two Clients and
one Talk Host

swapping information
between two clients

code for client and server

Monte Carlo for π

a pleasingly parallel
computation

Using Sockets

MCS 275 L-25

12 March 2008

Communication between Programs

client/server interaction ~ telephone exchange

Communication between Programs

client/server interaction ~
telephone exchange

Remote Server and Client

getting IP addresses of computers

code for remote client and server

Remote Server and Client

getting IP addresses of
computers

code for remote client and
server

Two Clients and one Talk Host

swapping information between two clients

code for client and server

Two Clients and one Talk Host

swapping information
between two clients

code for client and server

Monte Carlo for π

a pleasingly parallel computation

Monte Carlo for π

a pleasingly parallel
computation

getting IP addresses of computers

using `ifconfig` and pipes

On a Mac: open System Preferences -> Network.

On Unix: `ifconfig` configures network interface parameters. With no parameters, `ifconfig` displays current configuration.

To select what we need, we type at the prompt \$

```
$ ifconfig | grep "inet " | grep -v 127.0.0.1  
| cut -d\ -f2
```

The firewall on the server computer must allow incoming connections for Python.

See System Preferences -> Security.

In the next example, the connection is wireless.

getting IP addresses of computers

using `ifconfig` and pipes

On a Mac: open System Preferences -> Network.

On Unix: `ifconfig` configures network interface parameters. With no parameters, `ifconfig` displays current configuration.

To select what we need, we type at the prompt \$

```
$ ifconfig | grep "inet " | grep -v 127.0.0.1  
| cut -d\ -f2
```

The firewall on the server computer must allow incoming connections for Python.

See System Preferences -> Security.

In the next example, the connection is wireless.

getting IP addresses of computers

using `ifconfig` and pipes

On a Mac: open System Preferences -> Network.

On Unix: `ifconfig` configures network interface parameters. With no parameters, `ifconfig` displays current configuration.

To select what we need, we type at the prompt \$

```
$ ifconfig | grep "inet " | grep -v 127.0.0.1 \
  | cut -d\ -f2
```

The firewall on the server computer must allow incoming connections for Python.

See System Preferences -> Security.

In the next example, the connection is wireless.

getting IP addresses of computers

using `ifconfig` and pipes

On a Mac: open System Preferences -> Network.

On Unix: `ifconfig` configures network interface parameters. With no parameters, `ifconfig` displays current configuration.

To select what we need, we type at the prompt \$

```
$ ifconfig | grep "inet " | grep -v 127.0.0.1 \
  | cut -d\ -f2
```

The firewall on the server computer must allow incoming connections for Python.

See System Preferences -> Security.

In the next example, the connection is wireless.

Using Sockets

MCS 275 L-25

12 March 2008

Communication between Programs

client/server interaction ~ telephone exchange

Communication
between Programs

client/server interaction ~
telephone exchange

Remote Server and Client

getting IP addresses of computers
code for remote client and server

Remote Server
and Client

getting IP addresses of
computers

code for remote client and
server

Two Clients and one Talk Host

swapping information between two clients
code for client and server

Two Clients and
one Talk Host

swapping information
between two clients

code for client and server

Monte Carlo for π

a pleasingly parallel computation

Monte Carlo for π

a pleasingly parallel
computation

Code for the Client

in remote_client.py

```
from socket import *

server = '131.193.178.186' # IP address
number = 11267             # port number
buffer = 80               # buffer size

server_address = (server, number)
client = socket(AF_INET, SOCK_STREAM)
try:
    client.connect(server_address)
    print 'client is connected'
    data = raw_input('Give message : ')
    client.send(data)
except:
    print 'connection failed'

client.close()
```

MCS 275 L-25

12 March 2008

Communication
between Programs

client/server interaction ~
telephone exchange

Remote Server
and Client

getting IP addresses of
computers

code for remote client and
server

Two Clients and
one Talk Host

swapping information
between two clients

code for client and server

Monte Carlo for π

a pleasingly parallel
computation

Code for the Client

in remote_client.py

```
from socket import *

server = '131.193.178.186' # IP address
number = 11267             # port number
buffer = 80                # buffer size

server_address = (server, number)
client = socket(AF_INET, SOCK_STREAM)
try:
    client.connect(server_address)
    print 'client is connected'
    data = raw_input('Give message : ')
    client.send(data)
except:
    print 'connection failed'

client.close()
```

MCS 275 L-25

12 March 2008

Communication
between Programs

client/server interaction ~
telephone exchange

Remote Server
and Client

getting IP addresses of
computers

code for remote client and
server

Two Clients and
one Talk Host

swapping information
between two clients

code for client and server

Monte Carlo for π

a pleasingly parallel
computation

Code for the Client

in `remote_client.py`

```
from socket import *

server = '131.193.178.186' # IP address
number = 11267             # port number
buffer = 80                # buffer size

server_address = (server, number)
client = socket(AF_INET, SOCK_STREAM)
try:
    client.connect(server_address)
    print 'client is connected'
    data = raw_input('Give message : ')
    client.send(data)
except:
    print 'connection failed'

client.close()
```

MCS 275 L-25

12 March 2008

Communication
between Programs

client/server interaction ~
telephone exchange

Remote Server
and Client

getting IP addresses of
computers

code for remote client and
server

Two Clients and
one Talk Host

swapping information
between two clients

code for client and server

Monte Carlo for π

a pleasingly parallel
computation

Code for the Client

in `remote_client.py`

```
from socket import *

server = '131.193.178.186' # IP address
number = 11267             # port number
buffer = 80               # buffer size

server_address = (server, number)
client = socket(AF_INET, SOCK_STREAM)
try:
    client.connect(server_address)
    print 'client is connected'
    data = raw_input('Give message : ')
    client.send(data)
except:
    print 'connection failed'

client.close()
```

MCS 275 L-25

12 March 2008

Communication
between Programs

client/server interaction ~
telephone exchange

Remote Server
and Client

getting IP addresses of
computers

code for remote client and
server

Two Clients and
one Talk Host

swapping information
between two clients

code for client and server

Monte Carlo for π

a pleasingly parallel
computation

Code for the Client

in `remote_client.py`

```
from socket import *

server = '131.193.178.186' # IP address
number = 11267             # port number
buffer = 80                # buffer size

server_address = (server, number)
client = socket(AF_INET, SOCK_STREAM)
try:
    client.connect(server_address)
    print 'client is connected'
    data = raw_input('Give message : ')
    client.send(data)
except:
    print 'connection failed'

client.close()
```

MCS 275 L-25

12 March 2008

Communication
between Programs

client/server interaction ~
telephone exchange

Remote Server
and Client

getting IP addresses of
computers

code for remote client and
server

Two Clients and
one Talk Host

swapping information
between two clients

code for client and server

Monte Carlo for π

a pleasingly parallel
computation

Code of the Server

Part I of `remote_server.py`

```
from socket import *

hostname = ''      # so any address can be used
number = 11267     # number for the port
buffer = 80        # size of the buffer

server_address = (hostname, number)
server = socket(AF_INET, SOCK_STREAM)
server.bind(server_address)
server.listen(1)
```

MCS 275 L-25

12 March 2008

Communication
between Programs

client/server interaction ~
telephone exchange

Remote Server
and Client

getting IP addresses of
computers

code for remote client and
server

Two Clients and
one Talk Host

swapping information
between two clients

code for client and server

Monte Carlo for π

a pleasingly parallel
computation

Code of the Server

Part I of `remote_server.py`

```
from socket import *

hostname = ''      # so any address can be used
number = 11267    # number for the port
buffer = 80       # size of the buffer

server_address = (hostname, number)
server = socket(AF_INET, SOCK_STREAM)
server.bind(server_address)
server.listen(1)
```

MCS 275 L-25

12 March 2008

Communication
between Programs

client/server interaction ~
telephone exchange

Remote Server
and Client

getting IP addresses of
computers

code for remote client and
server

Two Clients and
one Talk Host

swapping information
between two clients

code for client and server

Monte Carlo for π

a pleasingly parallel
computation

Code of the Server

Part I of `remote_server.py`

```
from socket import *

hostname = ''      # so any address can be used
number = 11267    # number for the port
buffer = 80       # size of the buffer

server_address = (hostname, number)
server = socket(AF_INET, SOCK_STREAM)
server.bind(server_address)
server.listen(1)
```

MCS 275 L-25

12 March 2008

Communication
between Programs

client/server interaction ~
telephone exchange

Remote Server
and Client

getting IP addresses of
computers

code for remote client and
server

Two Clients and
one Talk Host

swapping information
between two clients

code for client and server

Monte Carlo for π

a pleasingly parallel
computation

Code for the Server

Part II of remote_server.py

```
print 'server waits for connection'
client, client_address = server.accept()

if client_address[0] == '131.193.41.130':
    print 'server accepted connection from ', \
        client_address
    print 'server waits for data'
    data = client.recv(buffer)
    print 'server received ', data
else:
    print 'server does not accept data from ', \
        client_address
    client.shutdown(SHUT_RDWR)

server.close()
```

MCS 275 L-25

12 March 2008

Communication
between Programs

client/server interaction ~
telephone exchange

Remote Server
and Client

getting IP addresses of
computers

code for remote client and
server

Two Clients and
one Talk Host

swapping information
between two clients

code for client and server

Monte Carlo for π

a pleasingly parallel
computation

Code for the Server

Part II of remote_server.py

```
print 'server waits for connection'
client, client_address = server.accept()

if client_address[0] == '131.193.41.130':
    print 'server accepted connection from ', \
        client_address
    print 'server waits for data'
    data = client.recv(buffer)
    print 'server received ', data
else:
    print 'server does not accept data from ', \
        client_address
    client.shutdown(SHUT_RDWR)

server.close()
```

MCS 275 L-25

12 March 2008

Communication
between Programs

client/server interaction ~
telephone exchange

Remote Server
and Client

getting IP addresses of
computers

code for remote client and
server

Two Clients and
one Talk Host

swapping information
between two clients

code for client and server

Monte Carlo for π

a pleasingly parallel
computation

Code for the Server

Part II of remote_server.py

```
print 'server waits for connection'
client, client_address = server.accept()

if client_address[0] == '131.193.41.130':
    print 'server accepted connection from ', \
        client_address
    print 'server waits for data'
    data = client.recv(buffer)
    print 'server received ', data
else:
    print 'server does not accept data from ', \
        client_address
    client.shutdown(SHUT_RDWR)

server.close()
```

MCS 275 L-25

12 March 2008

Communication
between Programs

client/server interaction ~
telephone exchange

Remote Server
and Client

getting IP addresses of
computers
code for remote client and
server

Two Clients and
one Talk Host

swapping information
between two clients
code for client and server

Monte Carlo for π

a pleasingly parallel
computation

Code for the Server

Part II of remote_server.py

```
print 'server waits for connection'
client, client_address = server.accept()

if client_address[0] == '131.193.41.130':
    print 'server accepted connection from ', \
        client_address
    print 'server waits for data'
    data = client.recv(buffer)
    print 'server received ', data
else:
    print 'server does not accept data from ', \
        client_address
    client.shutdown(SHUT_RDWR)

server.close()
```

MCS 275 L-25

12 March 2008

Communication
between Programs

client/server interaction ~
telephone exchange

Remote Server
and Client

getting IP addresses of
computers
code for remote client and
server

Two Clients and
one Talk Host

swapping information
between two clients
code for client and server

Monte Carlo for π

a pleasingly parallel
computation

Using Sockets

MCS 275 L-25

12 March 2008

Communication between Programs

client/server interaction $\sim$ telephone exchange

Communication
between Programs

client/server interaction $\sim$
telephone exchange

Remote Server and Client

getting IP addresses of computers
code for remote client and server

Remote Server
and Client

getting IP addresses of
computers
code for remote client and
server

Two Clients and one Talk Host

swapping information between two clients
code for client and server

Two Clients and
one Talk Host

swapping information
between two clients
code for client and server

Monte Carlo for π

a pleasingly parallel computation

Monte Carlo for π

a pleasingly parallel
computation

Protocol to swap Client Data

two clients and one server

Often the server acts as moderator between clients.

Suppose two clients want to exchange information.

Protocol:

1. server listens to two incoming clients
2. both clients connect to the server
and the server accepts two connections
3. both clients send their data to the server
4. server sends the data of the first client to the second
and the data of the second client to the first.

→ same program for the clients

Protocol to swap Client Data

two clients and one server

Often the server acts as moderator between clients.

Suppose two clients want to exchange information.

Protocol:

1. server listens to two incoming clients
2. both clients connect to the server
and the server accepts two connections
3. both clients send their data to the server
4. server sends the data of the first client to the second
and the data of the second client to the first.

→ same program for the clients

MCS 275 L-25

12 March 2008

Communication
between Programs

client/server interaction ~
telephone exchange

Remote Server
and Client

getting IP addresses of
computers
code for remote client and
server

Two Clients and
one Talk Host

swapping information
between two clients
code for client and server

Monte Carlo for π

a pleasingly parallel
computation

Protocol to swap Client Data

two clients and one server

Often the server acts as moderator between clients.

Suppose two clients want to exchange information.

Protocol:

1. server listens to two incoming clients
2. both clients connect to the server
and the server accepts two connections
3. both clients send their data to the server
4. server sends the data of the first client to the second
and the data of the second client to the first.

→ same program for the clients

Communication between Programs

client/server interaction ~
telephone exchange

Remote Server and Client

getting IP addresses of
computers
code for remote client and
server

Two Clients and one Talk Host

swapping information
between two clients
code for client and server

Monte Carlo for π

a pleasingly parallel
computation

Protocol to swap Client Data

two clients and one server

Often the server acts as moderator between clients.

Suppose two clients want to exchange information.

Protocol:

1. server listens to two incoming clients
2. both clients connect to the server
and the server accepts two connections
3. both clients send their data to the server
4. server sends the data of the first client to the second
and the data of the second client to the first.

→ same program for the clients

Protocol to swap Client Data

two clients and one server

Often the server acts as moderator between clients.

Suppose two clients want to exchange information.

Protocol:

1. server listens to two incoming clients
2. both clients connect to the server
and the server accepts two connections
3. both clients send their data to the server
4. server sends the data of the first client to the second
and the data of the second client to the first.

→ same program for the clients

Two Clients swapping Data

server is a talk host

Terminal window at the server:

```
$ python talk_host.py
talk host waits for first client
server accepted connection from ..\
    .. ('127.0.0.1', 49158)
talk host waits for second client
server accepted connection from ..\
    .. ('127.0.0.1', 49159)
talk host waits for first client
talk host received "this is alpha"
talk host waits for second client
talk host received "this is beta"
talk host sends "this is beta" to first client
talk host sends "this is alpha" to second client
$
```

MCS 275 L-25

12 March 2008

Communication
between Programs

client/server interaction ~
telephone exchange

Remote Server
and Client

getting IP addresses of
computers
code for remote client and
server

Two Clients and
one Talk Host

swapping information
between two clients
code for client and server

Monte Carlo for π

a pleasingly parallel
computation

Two Clients swapping Data

server is a talk host

Terminal window at the server:

```
$ python talk_host.py
talk host waits for first client
server accepted connection from ..\
    .. ('127.0.0.1', 49158)
talk host waits for second client
server accepted connection from ..\
    .. ('127.0.0.1', 49159)
talk host waits for first client
talk host received "this is alpha"
talk host waits for second client
talk host received "this is beta"
talk host sends "this is beta" to first client
talk host sends "this is alpha" to second client
$
```

MCS 275 L-25

12 March 2008

Communication
between Programs

client/server interaction ~
telephone exchange

Remote Server
and Client

getting IP addresses of
computers
code for remote client and
server

Two Clients and
one Talk Host

swapping information
between two clients
code for client and server

Monte Carlo for π

a pleasingly parallel
computation

Two Clients swapping Data

server is a talk host

Terminal window at the server:

```
$ python talk_host.py
talk host waits for first client
server accepted connection from ..\
    .. ('127.0.0.1', 49158)
talk host waits for second client
server accepted connection from ..\
    .. ('127.0.0.1', 49159)
talk host waits for first client
talk host received "this is alpha"
talk host waits for second client
talk host received "this is beta"
talk host sends "this is beta" to first client
talk host sends "this is alpha" to second client
$
```

MCS 275 L-25

12 March 2008

Communication
between Programs

client/server interaction ~
telephone exchange

Remote Server
and Client

getting IP addresses of
computers
code for remote client and
server

Two Clients and
one Talk Host

swapping information
between two clients
code for client and server

Monte Carlo for π

a pleasingly parallel
computation

Two Clients swapping Data

server is a talk host

Terminal window at the server:

```
$ python talk_host.py
talk host waits for first client
server accepted connection from ..\
    .. ('127.0.0.1', 49158)
talk host waits for second client
server accepted connection from ..\
    .. ('127.0.0.1', 49159)
talk host waits for first client
talk host received "this is alpha"
talk host waits for second client
talk host received "this is beta"
talk host sends "this is beta" to first client
talk host sends "this is alpha" to second client
$
```

MCS 275 L-25

12 March 2008

Communication
between Programs

client/server interaction ~
telephone exchange

Remote Server
and Client

getting IP addresses of
computers
code for remote client and
server

Two Clients and
one Talk Host

swapping information
between two clients
code for client and server

Monte Carlo for π

a pleasingly parallel
computation

Two Clients swapping Data

server is a talk host

Terminal window at the server:

```
$ python talk_host.py
talk host waits for first client
server accepted connection from ..\
    .. ('127.0.0.1', 49158)
talk host waits for second client
server accepted connection from ..\
    .. ('127.0.0.1', 49159)
talk host waits for first client
talk host received "this is alpha"
talk host waits for second client
talk host received "this is beta"
talk host sends "this is beta" to first client
talk host sends "this is alpha" to second client
$
```

MCS 275 L-25

12 March 2008

Communication
between Programs

client/server interaction ~
telephone exchange

Remote Server
and Client

getting IP addresses of
computers

code for remote client and
server

Two Clients and
one Talk Host

swapping information
between two clients

code for client and server

Monte Carlo for π

a pleasingly parallel
computation

Running the Clients

MCS 275 L-25

12 March 2008

Terminal window for the first client:

```
$ python talk_client.py
client is connected
Give message : this is alpha
client waits for reply
client received "this is beta"
$
```

Terminal window for the second client:

```
$ python talk_client.py
client is connected
Give message : this is beta
client waits for reply
client received "this is alpha"
$
```

Communication
between Programs

client/server interaction ~
telephone exchange

Remote Server
and Client

getting IP addresses of
computers
code for remote client and
server

Two Clients and
one Talk Host

swapping information
between two clients
code for client and server

Monte Carlo for π

a pleasingly parallel
computation

Running the Clients

MCS 275 L-25

12 March 2008

Terminal window for the first client:

```
$ python talk_client.py
client is connected
Give message : this is alpha
client waits for reply
client received "this is beta"
$
```

Terminal window for the second client:

```
$ python talk_client.py
client is connected
Give message : this is beta
client waits for reply
client received "this is alpha"
$
```

Communication
between Programs

client/server interaction ~
telephone exchange

Remote Server
and Client

getting IP addresses of
computers
code for remote client and
server

Two Clients and
one Talk Host

swapping information
between two clients
code for client and server

Monte Carlo for π

a pleasingly parallel
computation

Using Sockets

MCS 275 L-25

12 March 2008

Communication between Programs

client/server interaction ~ telephone exchange

Communication between Programs

client/server interaction ~
telephone exchange

Remote Server and Client

getting IP addresses of computers
code for remote client and server

Remote Server and Client

getting IP addresses of
computers
code for remote client and
server

Two Clients and one Talk Host

swapping information between two clients
code for client and server

Two Clients and one Talk Host

swapping information
between two clients
code for client and server

Monte Carlo for π

a pleasingly parallel computation

Monte Carlo for π

a pleasingly parallel
computation

Code for the Client

in talk_client.py

```
from socket import *

hostname = 'localhost' # on same host
number = 11267          # same port number
buffer = 80            # size of the buffer

server_address = (hostname, number)
client = socket(AF_INET, SOCK_STREAM)
client.connect(server_address)

print 'client is connected'
alpha = raw_input('Give message : ')
client.send(alpha)

print 'client waits for reply'
beta = client.recv(buffer)
print 'client received \'' + beta + '\''
client.close()
```

MCS 275 L-25

12 March 2008

Communication
between Programs

client/server interaction ~
telephone exchange

Remote Server
and Client

getting IP addresses of
computers
code for remote client and
server

Two Clients and
one Talk Host

swapping information
between two clients
code for client and server

Monte Carlo for π

a pleasingly parallel
computation

Code for the Client

in talk_client.py

```
from socket import *

hostname = 'localhost' # on same host
number = 11267          # same port number
buffer = 80            # size of the buffer

server_address = (hostname, number)
client = socket(AF_INET, SOCK_STREAM)
client.connect(server_address)

print 'client is connected'
alpha = raw_input('Give message : ')
client.send(alpha)

print 'client waits for reply'
beta = client.recv(buffer)
print 'client received \'' + beta + '\''
client.close()
```

MCS 275 L-25

12 March 2008

Communication
between Programs

client/server interaction ~
telephone exchange

Remote Server
and Client

getting IP addresses of
computers
code for remote client and
server

Two Clients and
one Talk Host

swapping information
between two clients
code for client and server

Monte Carlo for π

a pleasingly parallel
computation

Code for the Client

in talk_client.py

```
from socket import *

hostname = 'localhost' # on same host
number = 11267          # same port number
buffer = 80             # size of the buffer

server_address = (hostname, number)
client = socket(AF_INET, SOCK_STREAM)
client.connect(server_address)

print 'client is connected'
alpha = raw_input('Give message : ')
client.send(alpha)

print 'client waits for reply'
beta = client.recv(buffer)
print 'client received \'' + beta + '\''
client.close()
```

MCS 275 L-25

12 March 2008

Communication
between Programs

client/server interaction ~
telephone exchange

Remote Server
and Client

getting IP addresses of
computers
code for remote client and
server

Two Clients and
one Talk Host

swapping information
between two clients
code for client and server

Monte Carlo for π

a pleasingly parallel
computation

Code for the Client

in `talk_client.py`

```
from socket import *

hostname = 'localhost' # on same host
number = 11267          # same port number
buffer = 80             # size of the buffer

server_address = (hostname, number)
client = socket(AF_INET, SOCK_STREAM)
client.connect(server_address)

print 'client is connected'
alpha = raw_input('Give message : ')
client.send(alpha)

print 'client waits for reply'
beta = client.recv(buffer)
print 'client received \'' + beta + '\''
client.close()
```

MCS 275 L-25

12 March 2008

Communication
between Programs

client/server interaction ~
telephone exchange

Remote Server
and Client

getting IP addresses of
computers
code for remote client and
server

Two Clients and
one Talk Host

swapping information
between two clients
code for client and server

Monte Carlo for π

a pleasingly parallel
computation

Code for the Client

in `talk_client.py`

```
from socket import *

hostname = 'localhost' # on same host
number = 11267          # same port number
buffer = 80             # size of the buffer

server_address = (hostname, number)
client = socket(AF_INET, SOCK_STREAM)
client.connect(server_address)

print 'client is connected'
alpha = raw_input('Give message : ')
client.send(alpha)

print 'client waits for reply'
beta = client.recv(buffer)
print 'client received \'' + beta + '\''
client.close()
```

MCS 275 L-25

12 March 2008

Communication
between Programs

client/server interaction ~
telephone exchange

Remote Server
and Client

getting IP addresses of
computers
code for remote client and
server

Two Clients and
one Talk Host

swapping information
between two clients
code for client and server

Monte Carlo for π

a pleasingly parallel
computation

Code for starting Server

part I of `talk_host.py`

```
from socket import *

hostname = ''      # so any address can be used
number = 11267     # number for the port
buffer = 80        # size of the buffer

server_address = (hostname, number)
server = socket(AF_INET, SOCK_STREAM)
server.bind(server_address)
server.listen(2)
```

MCS 275 L-25

12 March 2008

Communication
between Programs

client/server interaction ~
telephone exchange

Remote Server
and Client

getting IP addresses of
computers
code for remote client and
server

Two Clients and
one Talk Host

swapping information
between two clients
code for client and server

Monte Carlo for π

a pleasingly parallel
computation

Code for accepting Connections

part II of talk_host.py

```
print 'talk host waits for first client'  
first, first_address = server.accept()  
print 'server accepted connection from ',\  
      first_address  
print 'talk host waits for second client'  
second, second_address = server.accept()  
print 'server accepted connection from ',\  
      second_address
```

MCS 275 L-25

12 March 2008

Communication
between Programs

client/server interaction ~
telephone exchange

Remote Server
and Client

getting IP addresses of
computers
code for remote client and
server

Two Clients and
one Talk Host

swapping information
between two clients
code for client and server

Monte Carlo for π

a pleasingly parallel
computation

Code for accepting Connections

part II of talk_host.py

```
print 'talk host waits for first client'
first, first_address = server.accept()
print 'server accepted connection from ', \
    first_address
print 'talk host waits for second client'
second, second_address = server.accept()
print 'server accepted connection from ', \
    second_address
```

MCS 275 L-25

12 March 2008

Communication
between Programs

client/server interaction ~
telephone exchange

Remote Server
and Client

getting IP addresses of
computers
code for remote client and
server

Two Clients and
one Talk Host

swapping information
between two clients
code for client and server

Monte Carlo for π

a pleasingly parallel
computation

Code for exchanging Data

end of talk_host.py

```
print 'talk host waits for first client'
alpha = first.recv(buffer)
print 'talk host received \'' + alpha + '\''
print 'talk host waits for second client'
beta = second.recv(buffer)
print 'talk host received \'' + beta + '\''
print 'talk host sends \'' + beta + '\'' \
      + ' to first client'
first.send(beta)
print 'talk host sends \'' + alpha + '\'' \
      + ' to second client'
second.send(alpha)
server.close()
```

MCS 275 L-25

12 March 2008

Communication
between Programs

client/server interaction ~
telephone exchange

Remote Server
and Client

getting IP addresses of
computers
code for remote client and
server

Two Clients and
one Talk Host

swapping information
between two clients
code for client and server

Monte Carlo for π

a pleasingly parallel
computation

Code for exchanging Data

end of talk_host.py

```
print 'talk host waits for first client'
alpha = first.recv(buffer)
print 'talk host received \'' + alpha + '\''
print 'talk host waits for second client'
beta = second.recv(buffer)
print 'talk host received \'' + beta + '\''
print 'talk host sends \'' + beta + '\'' \
      + ' to first client'
first.send(beta)
print 'talk host sends \'' + alpha + '\'' \
      + ' to second client'
second.send(alpha)
server.close()
```

MCS 275 L-25

12 March 2008

Communication
between Programs

client/server interaction ~
telephone exchange

Remote Server
and Client

getting IP addresses of
computers
code for remote client and
server

Two Clients and
one Talk Host

swapping information
between two clients
code for client and server

Monte Carlo for π

a pleasingly parallel
computation

Code for exchanging Data

end of talk_host.py

```
print 'talk host waits for first client'
alpha = first.recv(buffer)
print 'talk host received \'' + alpha + '\''
print 'talk host waits for second client'
beta = second.recv(buffer)
print 'talk host received \'' + beta + '\''

print 'talk host sends \'' + beta + '\'' \
      + ' to first client'
first.send(beta)
print 'talk host sends \'' + alpha + '\'' \
      + ' to second client'
second.send(alpha)
server.close()
```

MCS 275 L-25

12 March 2008

Communication
between Programs

client/server interaction ~
telephone exchange

Remote Server
and Client

getting IP addresses of
computers
code for remote client and
server

Two Clients and
one Talk Host

swapping information
between two clients
code for client and server

Monte Carlo for π

a pleasingly parallel
computation

Code for exchanging Data

end of talk_host.py

```
print 'talk host waits for first client'
alpha = first.recv(buffer)
print 'talk host received \'' + alpha + '\''
print 'talk host waits for second client'
beta = second.recv(buffer)
print 'talk host received \'' + beta + '\''
print 'talk host sends \'' + beta + '\'' \
      + ' to first client'
first.send(beta)
print 'talk host sends \'' + alpha + '\'' \
      + ' to second client'
second.send(alpha)
server.close()
```

MCS 275 L-25

12 March 2008

Communication
between Programs

client/server interaction ~
telephone exchange

Remote Server
and Client

getting IP addresses of
computers
code for remote client and
server

Two Clients and
one Talk Host

swapping information
between two clients
code for client and server

Monte Carlo for π

a pleasingly parallel
computation

Code for exchanging Data

end of talk_host.py

```
print 'talk host waits for first client'
alpha = first.recv(buffer)
print 'talk host received \'' + alpha + '\''
print 'talk host waits for second client'
beta = second.recv(buffer)
print 'talk host received \'' + beta + '\''
print 'talk host sends \'' + beta + '\'' \
      + ' to first client'
first.send(beta)
print 'talk host sends \'' + alpha + '\'' \
      + ' to second client'
second.send(alpha)
server.close()
```

MCS 275 L-25

12 March 2008

Communication
between Programs

client/server interaction ~
telephone exchange

Remote Server
and Client

getting IP addresses of
computers
code for remote client and
server

Two Clients and
one Talk Host

swapping information
between two clients
code for client and server

Monte Carlo for π

a pleasingly parallel
computation

Using Sockets

MCS 275 L-25

12 March 2008

Communication between Programs

client/server interaction $\sim$ telephone exchange

Communication between Programs

client/server interaction $\sim$
telephone exchange

Remote Server and Client

getting IP addresses of computers
code for remote client and server

Remote Server and Client

getting IP addresses of
computers
code for remote client and
server

Two Clients and one Talk Host

swapping information between two clients
code for client and server

Two Clients and one Talk Host

swapping information
between two clients
code for client and server

Monte Carlo for π

a pleasingly parallel computation

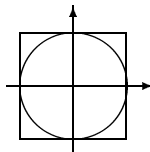
Monte Carlo for π

a pleasingly parallel
computation

Estimating π

Monte Carlo methods: throwing darts

The area of the unit disk is π :



Generate random uniformly distributed points with coordinates $(x, y) \in [-1, +1] \times [-1, +1]$.
We count a success when $x^2 + y^2 \leq 1$.

1. generate n points P in $[0, 1] \times [0, 1]$
2. $m := \{ (x, y) \in P : x^2 + y^2 \leq 1 \}$
3. the estimate is then $4 \times m/n$

→ dual core processor: use two clients.

pleasingly parallel: optimal speedup (twice as fast)

Communication
between Programs

client/server interaction ~
telephone exchange

Remote Server
and Client

getting IP addresses of
computers
code for remote client and
server

Two Clients and
one Talk Host

swapping information
between two clients
code for client and server

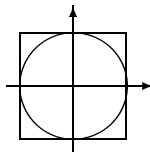
Monte Carlo for π

a pleasingly parallel
computation

Estimating π

Monte Carlo methods: throwing darts

The area of the unit disk is π :



Generate random uniformly distributed points with coordinates $(x, y) \in [-1, +1] \times [-1, +1]$.
We count a success when $x^2 + y^2 \leq 1$.

1. generate n points P in $[0, 1] \times [0, 1]$
2. $m := \{ (x, y) \in P : x^2 + y^2 \leq 1 \}$
3. the estimate is then $4 \times m/n$

→ dual core processor: use two clients.

pleasingly parallel: optimal speedup (twice as fast)

Communication
between Programs

client/server interaction ~
telephone exchange

Remote Server
and Client

getting IP addresses of
computers
code for remote client and
server

Two Clients and
one Talk Host

swapping information
between two clients
code for client and server

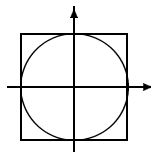
Monte Carlo for π

a pleasingly parallel
computation

Estimating π

Monte Carlo methods: throwing darts

The area of the unit disk is π :



Generate random uniformly distributed points with coordinates $(x, y) \in [-1, +1] \times [-1, +1]$.
We count a success when $x^2 + y^2 \leq 1$.

1. generate n points P in $[0, 1] \times [0, 1]$
2. $m := \{ (x, y) \in P : x^2 + y^2 \leq 1 \}$
3. the estimate is then $4 \times m/n$

→ dual core processor: use two clients.

pleasingly parallel: optimal speedup (twice as fast)

Communication
between Programs

client/server interaction ~
telephone exchange

Remote Server
and Client

getting IP addresses of
computers
code for remote client and
server

Two Clients and
one Talk Host

swapping information
between two clients
code for client and server

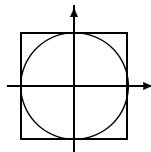
Monte Carlo for π

a pleasingly parallel
computation

Estimating π

Monte Carlo methods: throwing darts

The area of the unit disk is π :



Generate random uniformly distributed points with coordinates $(x, y) \in [-1, +1] \times [-1, +1]$.
We count a success when $x^2 + y^2 \leq 1$.

1. generate n points P in $[0, 1] \times [0, 1]$
2. $m := \{ (x, y) \in P : x^2 + y^2 \leq 1 \}$
3. the estimate is then $4 \times m/n$

→ dual core processor: use two clients.

pleasingly parallel: optimal speedup (twice as fast)

Communication
between Programs

client/server interaction ~
telephone exchange

Remote Server
and Client

getting IP addresses of
computers
code for remote client and
server

Two Clients and
one Talk Host

swapping information
between two clients
code for client and server

Monte Carlo for π

a pleasingly parallel
computation

Server and Client Terminals

server distributes seeds for random number generators

```
$ python mc4pi2.py
server waits for connections...
server waits for results...
approximation for pi = 3.141487
```

```
$ python mc4pi_client.py
client is connected
client received 1
client computes 0.785253200000
```

```
$ python mc4pi_client.py
client is connected
client received 2
client computes 0.785490300000
```

MCS 275 L-25

12 March 2008

Communication
between Programs

client/server interaction ~
telephone exchange

Remote Server
and Client

getting IP addresses of
computers
code for remote client and
server

Two Clients and
one Talk Host

swapping information
between two clients
code for client and server

Monte Carlo for π

a pleasingly parallel
computation

Server and Client Terminals

server distributes seeds for random number generators

```
$ python mc4pi2.py
server waits for connections...
server waits for results...
approximation for pi = 3.141487
```

```
$ python mc4pi_client.py
client is connected
client received 1
client computes 0.785253200000
```

```
$ python mc4pi_client.py
client is connected
client received 2
client computes 0.785490300000
```

MCS 275 L-25

12 March 2008

Communication
between Programs

client/server interaction ~
telephone exchange

Remote Server
and Client

getting IP addresses of
computers
code for remote client and
server

Two Clients and
one Talk Host

swapping information
between two clients
code for client and server

Monte Carlo for π

a pleasingly parallel
computation

Server and Client Terminals

server distributes seeds for random number generators

```
$ python mc4pi2.py
server waits for connections...
server waits for results...
approximation for pi = 3.141487
```

```
$ python mc4pi_client.py
client is connected
client received 1
client computes 0.785253200000
```

```
$ python mc4pi_client.py
client is connected
client received 2
client computes 0.785490300000
```

MCS 275 L-25

12 March 2008

Communication
between Programs

client/server interaction ~
telephone exchange

Remote Server
and Client

getting IP addresses of
computers
code for remote client and
server

Two Clients and
one Talk Host

swapping information
between two clients
code for client and server

Monte Carlo for π

a pleasingly parallel
computation

One Server distributes Seeds

in mc4pi2.py

```
...
server.listen(2)
print 'server waits for connections...'
first, first_address = server.accept()
second, second_address = server.accept()

first.send('1'); second.send('2')
print 'server waits for results...'

r1 = first.recv(buffer)
r2 = second.recv(buffer)
result = 2*(float(r1)+float(r2))
print 'approximation for pi =', result
server.close()
```

MCS 275 L-25

12 March 2008

Communication
between Programs

client/server interaction ~
telephone exchange

Remote Server
and Client

getting IP addresses of
computers
code for remote client and
server

Two Clients and
one Talk Host

swapping information
between two clients
code for client and server

Monte Carlo for π

a pleasingly parallel
computation

One Server distributes Seeds

in mc4pi2.py

```
...
server.listen(2)
print 'server waits for connections...'
first, first_address = server.accept()
second, second_address = server.accept()

first.send('1'); second.send('2')
print 'server waits for results...'

r1 = first.recv(buffer)
r2 = second.recv(buffer)
result = 2*(float(r1)+float(r2))
print 'approximation for pi =', result
server.close()
```

MCS 275 L-25

12 March 2008

Communication
between Programs

client/server interaction ~
telephone exchange

Remote Server
and Client

getting IP addresses of
computers
code for remote client and
server

Two Clients and
one Talk Host

swapping information
between two clients
code for client and server

Monte Carlo for π

a pleasingly parallel
computation

One Server distributes Seeds

in mc4pi2.py

```
...
server.listen(2)
print 'server waits for connections...'
first, first_address = server.accept()
second, second_address = server.accept()

first.send('1'); second.send('2')
print 'server waits for results...'

r1 = first.recv(buffer)
r2 = second.recv(buffer)
result = 2*(float(r1)+float(r2))
print 'approximation for pi =', result
server.close()
```

MCS 275 L-25

12 March 2008

Communication
between Programs

client/server interaction ~
telephone exchange

Remote Server
and Client

getting IP addresses of
computers
code for remote client and
server

Two Clients and
one Talk Host

swapping information
between two clients
code for client and server

Monte Carlo for π

a pleasingly parallel
computation

Code for the Client

part I of mc4pi_client.py

```
import random

from socket import *
hostname = 'localhost' # on same host
number = 11267          # same port number
buffer = 80             # size of the buffer
server_address = (hostname, number)
client = socket(AF_INET, SOCK_STREAM)
client.connect(server_address)
print 'client is connected'
s = client.recv(buffer)
print 'client received %s' % s

random.seed(int(s))
```

MCS 275 L-25

12 March 2008

Communication
between Programs

client/server interaction ~
telephone exchange

Remote Server
and Client

getting IP addresses of
computers
code for remote client and
server

Two Clients and
one Talk Host

swapping information
between two clients
code for client and server

Monte Carlo for π

a pleasingly parallel
computation

Code for the Client

part I of mc4pi_client.py

```
import random

from socket import *
hostname = 'localhost' # on same host
number = 11267          # same port number
buffer = 80             # size of the buffer
server_address = (hostname, number)
client = socket(AF_INET, SOCK_STREAM)
client.connect(server_address)
print 'client is connected'
s = client.recv(buffer)
print 'client received %s' % s

random.seed(int(s))
```

MCS 275 L-25

12 March 2008

Communication
between Programs

client/server interaction ~
telephone exchange

Remote Server
and Client

getting IP addresses of
computers
code for remote client and
server

Two Clients and
one Talk Host

swapping information
between two clients
code for client and server

Monte Carlo for π

a pleasingly parallel
computation

Code for the Client

part I of mc4pi_client.py

```
import random

from socket import *
hostname = 'localhost' # on same host
number = 11267          # same port number
buffer = 80             # size of the buffer
server_address = (hostname, number)
client = socket(AF_INET, SOCK_STREAM)
client.connect(server_address)
print 'client is connected'
s = client.recv(buffer)
print 'client received %s' % s

random.seed(int(s))
```

MCS 275 L-25

12 March 2008

Communication
between Programs

client/server interaction ~
telephone exchange

Remote Server
and Client

getting IP addresses of
computers
code for remote client and
server

Two Clients and
one Talk Host

swapping information
between two clients
code for client and server

Monte Carlo for π

a pleasingly parallel
computation

Code for the Client

part II of mc4pi_client.py

```
N = 10**7
k = 0
for i in range(0,N):
    x = random.uniform(0,1)
    y = random.uniform(0,1)
    if x**2 + y**2 <= 1: k = k + 1

R = float(k)/N
print 'client computes %.12f' % R

client.send(str(R))

client.close()
```

MCS 275 L-25

12 March 2008

Communication
between Programs

client/server interaction ~
telephone exchange

Remote Server
and Client

getting IP addresses of
computers
code for remote client and
server

Two Clients and
one Talk Host

swapping information
between two clients
code for client and server

Monte Carlo for π

a pleasingly parallel
computation

Code for the Client

part II of mc4pi_client.py

```
N = 10**7
k = 0
for i in range(0,N):
    x = random.uniform(0,1)
    y = random.uniform(0,1)
    if x**2 + y**2 <= 1: k = k + 1

R = float(k)/N
print 'client computes %.12f' % R

client.send(str(R))

client.close()
```

MCS 275 L-25

12 March 2008

Communication
between Programs

client/server interaction ~
telephone exchange

Remote Server
and Client

getting IP addresses of
computers
code for remote client and
server

Two Clients and
one Talk Host

swapping information
between two clients
code for client and server

Monte Carlo for π

a pleasingly parallel
computation

Summary + Assignments

MCS 275 L-25

12 March 2008

We covered more of chapter 12 in *Making Use of Python*; and introduced *interactive* parallel computation.

Exercises:

1. Explore the possibilities of a client/server interaction between `icarus.cc.uic.edu` and computers in the labs, or between your laptop and home computer.
2. Modify the code for `talk_host.py`: the user is prompted to enter the number of clients. Change the code for `talk_client.py`: the user enters the number of the client. Extend the data swapping protocol: data from the i th client goes to the $(i + 1)$ st client, data from the last client is sent to the first one.
3. The natural logarithm of 2 is $\ln(2) = \int_1^2 \frac{1}{x} dx$. Adjust the code for estimating π , to estimate $\ln(2)$ with an interactive parallel client/server computation.

Communication
between Programs

client/server interaction ~
telephone exchange

Remote Server
and Client

getting IP addresses of
computers
code for remote client and
server

Two Clients and
one Talk Host

swapping information
between two clients
code for client and server

Monte Carlo for π

a pleasingly parallel
computation