CS131: Programming Languages

Fall 2015
Week #8

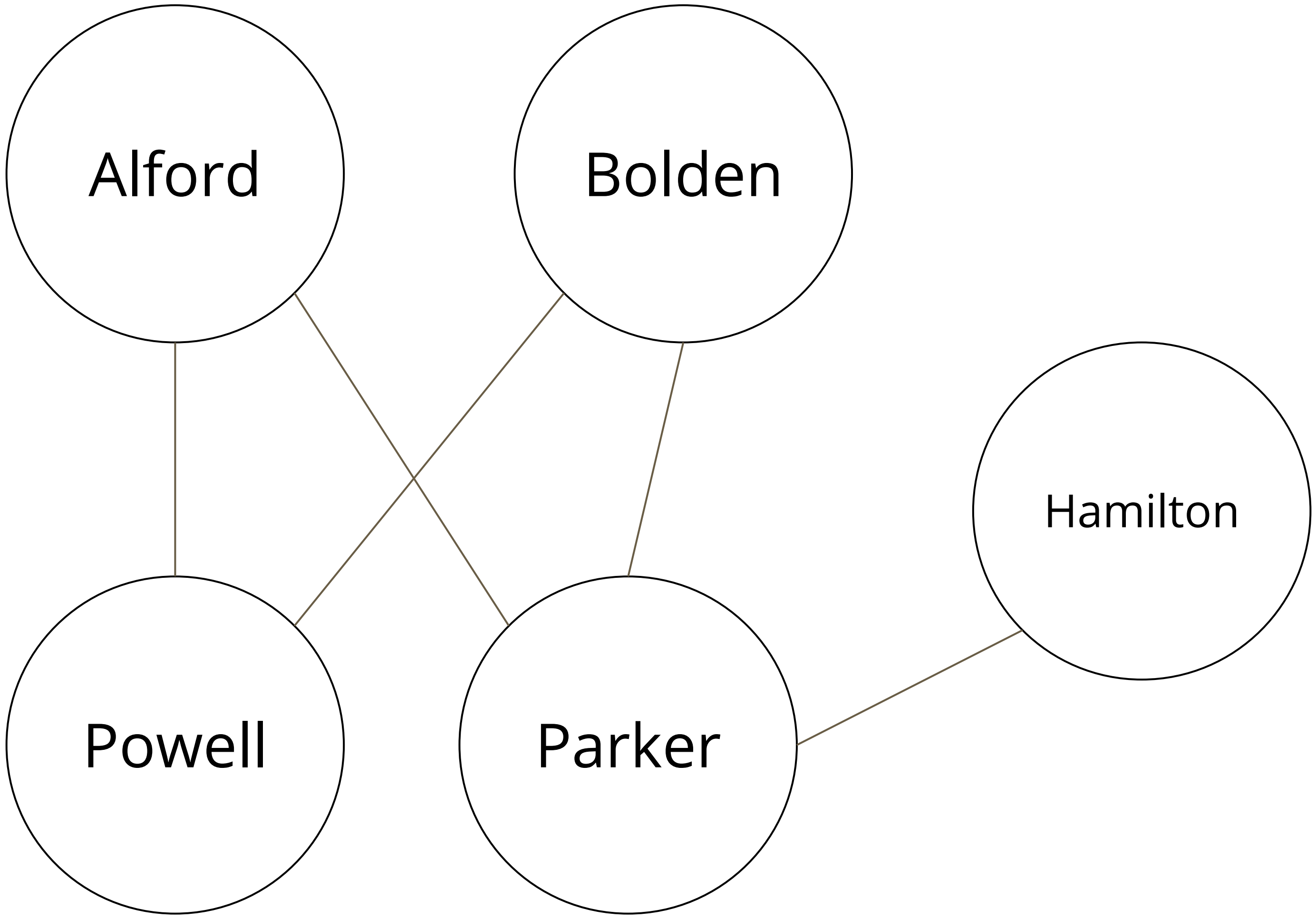
Announcements

- Project due 11/30, 11:55 PM
    - Late policy applies
- HW 6 due 12/3

Today

- Project
- HW 6

ProxyHerd Layout



Running the Reactor

```
from twisted.internet import reactor
def main():
    ...
    reactor.listenTCP(port, factory)
    reactor.run()
```

What is the Factory?

- Class that builds new protocol instances

```
from twisted.internet import protocol
class YourFactory(protocol.ServerFactory):
   def __init__ ...
   def buildProtocol(self, addr):
      return YourProtocol(self)
```

- Run the reactor for each server
  - A new protocol will be created for each new connection

What is the Protocol?

- Class that implements event methods

```
from twisted.protocols.basic import LineReceiver
class YourProtocol(LineReceiver):
   def connectionMade(self):
      ...
   def lineReceived(self, line):
      ...
   def connectionLost(self, reason):
      ...
```

Handling Commands

```
IAMAT kiwi.cs.ucla.edu +34.068930-118.445127 1400794645.3...
```

- IAMAT contains
  - Command name
  - client ID
    - Can be any string of non-whitespace
  - LatLng
  - Time client thinks it sent the message
  - Args are separated by whitespace
    - Can be tab, space, etc

Handling Commands

```
AT Alford +0.563873386 kiwi.cs.ucla.edu +34.068930-
118.445127 1400794699.108893381
```

- Server response to IAMAT
  - Command name = AT
  - server ID
  - Time difference between client time and receipt
  - Client ID
  - LatLng
  - Time server thinks it sent the message

Handling Commands

```
WHATSAT kiwi.cs.ucla.edu 10 5
```

- WHATSAT contains
  - Command name
  - client ID
  - Radius in km
  - Number of results desired

Handling Commands

```
AT Alford +0.563873386 kiwi.cs.ucla.edu +34.068930-
118.445127 1400794699.108893381

json { ... }
```

- Server response to WHATSAT
    - Command name
    - server ID
        - of most recent IAMAT
    - Time difference between client time and receipt
        - of most recent IAMAT
    - Client ID
    - LatLng of most recent IAMAT
    - Time server thinks it sent the message

Server Propagation

- Servers must send information to each other
- Client can ask any server for info, they should all have it
- Use a flooding algorithm
  - When a server receives an IAMAT, forward the information to all connected servers
    - Forward the information in whatever format you design
  - When a server receives the forwarded info, forward it to all connected servers
  - If you see info you already sent, ignore it

Client Protocol

- Client connects and sends commands via telnet
    - No need to create Protocol for this
- However, server needs to act like client during propagation
    - Create Factory & Protocol for this
    - How will the protocol differ; what event are you concerned with?

Logging

- Log everything!
  - Made connection
  - Every command received
  - Connections lost

Project API Key

See url on course webpage

twisted.web.client.getPage

d = getPage(url)

d.addCallbacks(callback=<fun>, errback=<fun>)

```
https://maps.googleapis.
com/maps/api/place/nearbysearch/json?location=+34.068930,
-118.445127&radius=10&sensor=false&key=API_KEY
```

Download Twisted

First check:

```
$ python
>>> import twisted
```

http://twistedmatrix.com/Releases/Twisted/15.0/

or

```
pip install twisted
```

Testing TCP Ports

Email TAs for allocation

Report Contents

- 5 pages max
- 80% - talk about your Project implementation
  - How your protocol works with client messages
  - How your servers communicate with each other
  - Biggest hurdles
- 20% - Node.js vs Twisted
  - Discussion of how your implementation would change if you used Node.js

HW 6

- Virtual Machine
    - Creates sandboxed OS on host computer
    - Uses host's resources
    - Abstracts an entire machine
        - Has own network interface, filesystem, kernel, etc
        - Contains a copy of all the binaries the OS would have
    - Large overhead

HW 6

- Linux Containers
- Have own network interface, file system, etc
- But share the kernel with the host
- Load in an image
    - Contains all the binaries needed to run an application
    - Created by the user

HW 6

- Docker
  - Based on Linux Containers (LXC)
  - Written in Go
- Your task
  - Research Docker, LXC, Go, & why Go is used in Docker
  - Discuss how you might implement Docker in:
    - Java, Python, and (Rust | Scala | Clojure | Groovy)

HW 6

- Alternative implementation: DockAlt
- Things to discuss:
    - Impact of language style (imperative, functional, etc)
    - Impact of static/dynamic typing
    - Library availability
        - Does it have a LXC library?
    - Modularization
        - How does the module system differ from Go, and would this affect implementation?