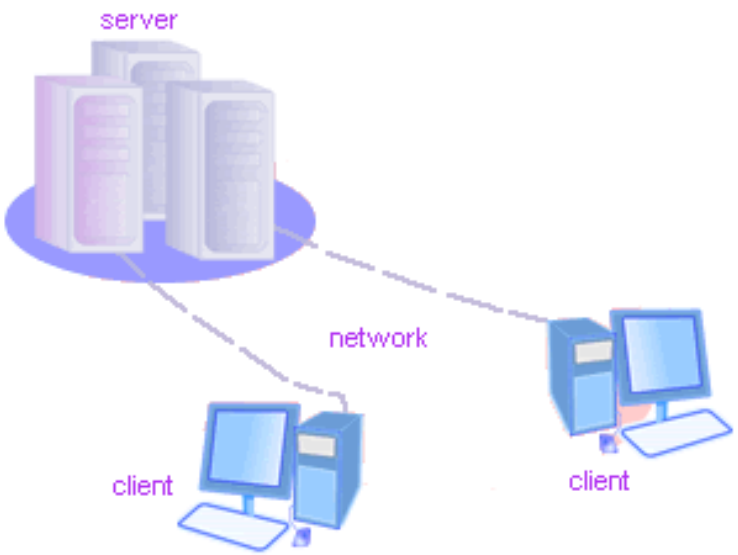
Twisty Twitter Herd



CS 131

Henry Yu

The Prototype: A description

- Consists of proxy servers
- Accepts incoming TCP connections
  - IAMAT message: updates database
  - WHATSAT message: send query to twitter
- Servers have a set rule to who they talk to
  - Relationship two-way
  - Blake talks to everyone but Howard and Metta
  - Gasol talks with Bryant and Howard
  - Metta talks with Bryant

Flooding Algorithm

- Not all servers talk to each other
- It propagates through
- All the servers will eventually all have the information
  - All maintain their own database
- Allows faster queries since its all in their database
  - Ex: doesn't have to ask a "Master Server"

Advantages

- “Deferred Objects”
  - Handles blocking functions
- Factory class
  - Automatically declares protocols
  - Multiple clients can connect w/o additional effort
- Handles all low-level calls
  - Ex: listening to port and relaying single messages
- Built-in methods can be overloaded

Twisted Performance

(3pm)	2 max_tweets	100 max_tweets
6 concurrent requests	.387-.518	.406-.563
1 single request	0.468	0.538
(3am)	2 max_tweets	100 max_tweets
6 concurrent requests	.084-.132	.096.135
1 single request	0.099	0.104

- Network traffic due to time of day is much larger factor
- Increase in data size -> small increase in response time
- Increase in concurrent user -> no noticeable effect on response time

Language Issues

- Maintainability
  - Easy to read and write
  - Open source
  - Object oriented = modularity
- Efficiency
  - Interpreted language -> dynamic type checking
    - Not an issue since this is due to data processing
      - Our application doesn't have a lot of this; mostly just transferring data
- Multithreading
  - Safe due to Global Interpreter Lock

Conclusion

- Advantages
  - Extreme ease with its library
  - Minimal cost for developing
    - Open source + easy to write
- Disadvantages
  - Minor drawbacks in processor speeds
- Twisted is a promising and elegant solution to our application