Pat Long, Jesse Klimov
Mentor: Jean Yang

# Characterizing Social Networks and their Queries

Second Annual MIT PRIMES Conference, May 20, 2012

# Current Problems with Social Network Programming

- Data leaks
  - Complex policies – data can slip through
- Programmer Frustration
  - Difficult to manage sensitive data

# Characteristics of Social Networks

- High clustering coefficient
- Very tightly linked
- Hubs
- 80/20 rule
- Scale-free
- Small diameter

# Jeeves – A Proposed Solution

- Scala library for managing sensitive data
- Adds policies and level variables
- Z3 Solver (Microsoft research project)

////Error: Constraint Environment

////Eval: False

////Access Denied

////This will be reported

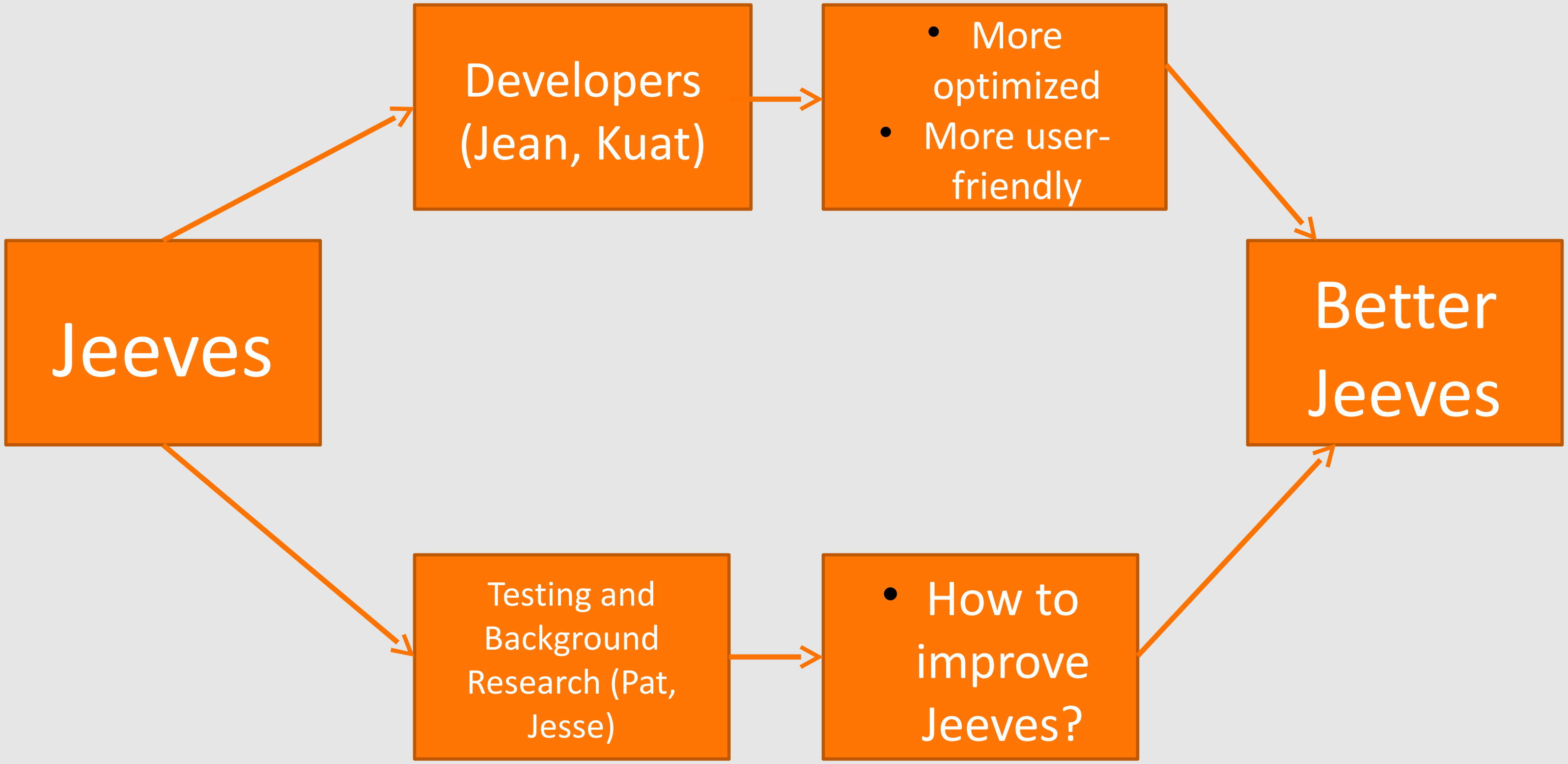
# Project Goals – Social Network Simulation

- Testing Jeeves's capabilities for real world applications
- Many simplifying assumptions
  - Jeeves is still at an early stage
- Queries
  - A few key categories
  - Many constraints per query

# Our Project

- Designed a framework to demonstrate how Jeeves can manage privacy policies
- Generated and analyzed social networks
  - Verified realistic properties
- Ran queries over these networks

# Overview of the Jeeves Project



Jeeves

Developers (Jean, Kuat)

- More optimized
- More user-friendly

Testing and Background Research (Pat, Jesse)

- How to improve Jeeves?

Better Jeeves

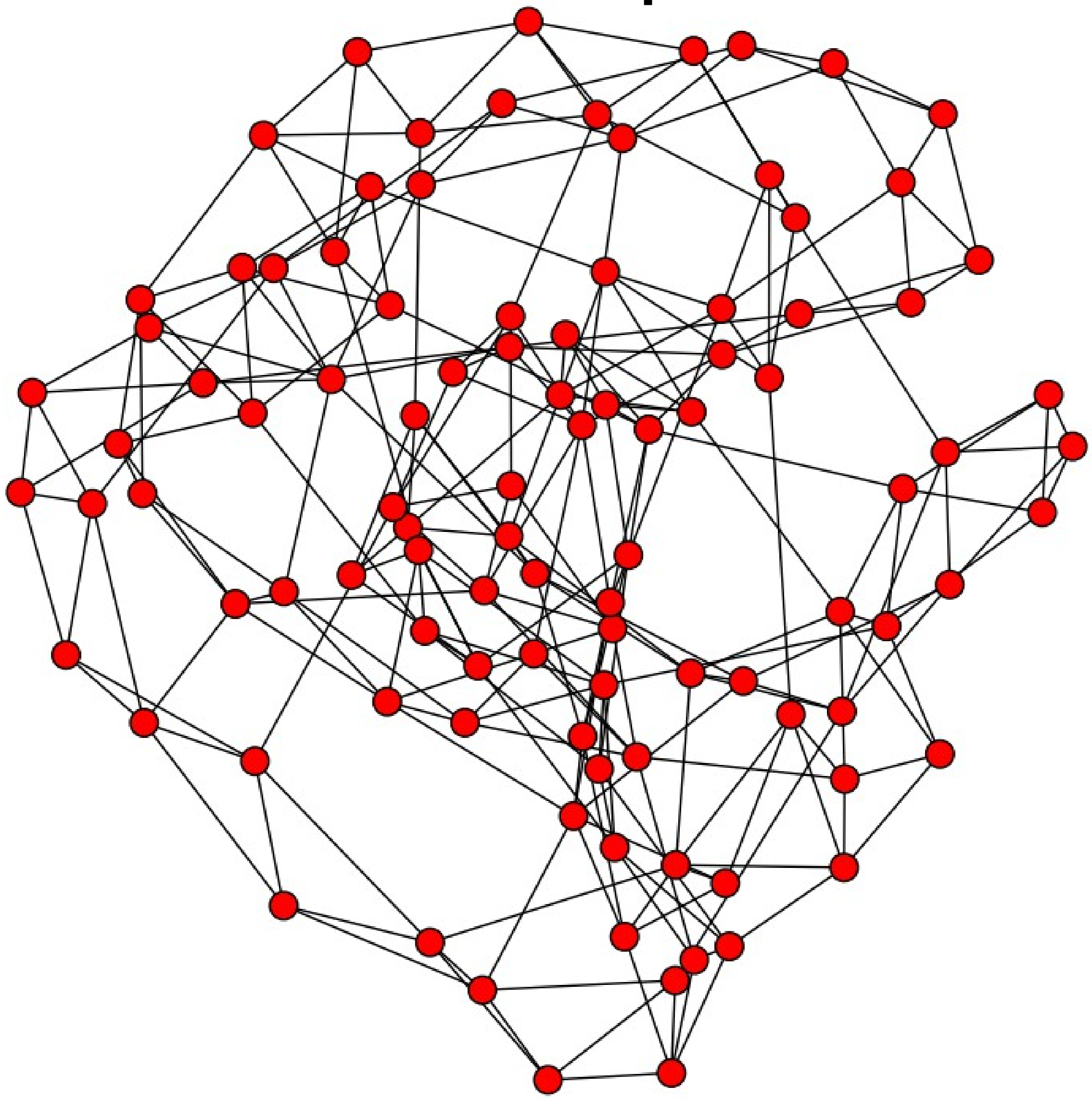
# Progress and Results (Design)

- Created a toy social network
  - Contains mock users with enough properties to test out the effectiveness of Jeeves as a tool in real-world situations
- Loaded *n* users and performed simple tests on each of them

# Network Generation

- We used NetworkX, a Python Package to generate network graphs
  - Developed by Los Alamos National Laboratory
  - Chose it because it is written in python, an easy language for rapid development
- Used Newman-Watts-Strogatz algorithm for generating realistic social networks
- Generated graph with 100 nodes and ~5 edges per node

# Our Graph



# Are the Graphs Realistic?

- We looked at different graphs metrics to ensure out graphs were realistic.
  - Directedness: Not directed
  - Degree Distribution (Scale-free network)
  - Clustering Coefficient: C = 0.343
  - Small-World Phenomenon
    - Diameter = 7 (n = 100)

# Progress and Results (Testing)

- Aimed at designing a semi-realistic infrastructure to confirm hypotheses about constraint complexity and performance
  - Created testing daemon to allow for easily running a variety of tests
- ~3/4 second to solve levels with non-trivial policies (not constant)
- 606 constraints loaded each time
  - Jeeves creates many constraints to represent the heap as Z3 does not support advanced datatypes

# Future Plans

- Create daemon to run tests against
  - Self sufficient - runs on server where we can connect to it and tell it to execute tests
- Add weighted distribution of query activities for more realistic tests
  - Periodically, each user would randomly pick a new query activity from table of typical user queries (see next page)
- Perform tests on larger social networks (500,1000,... users)

# Distribution of Queries in Social Networks (Simplified)

| | |
|---|---|
| Universal search | 4.5% |
| Browse profiles | 38.5% |
| Browse home page | 36.3% |
| List of friends | 12.3% |
| Browse friend updates | 3.0% |
| Browse member communities | 2.8% |
| Profile editing | 2.6% |

# Future Plans (contd…)

- Test corner cases
- Compare the efficiency of the Jeeves model to other models (also run networks with a size more comparable to that of a small real social network)
- Extend tests to more realistically model a social network
  - Simulating typical user actions on a social network
  - Time these actions to resemble user activity on a normal social network
  - Add more complex rules and queries

# Conclusion

- Currently, queries are just tests to see the relationship between expected scaling and actual scaling
- We perform these queries on a simulated social network
- Currently, Jeeves is too slow to manage a social network's data privacy
  - Jeeves is currently unoptimized
  - Needs to balance its usage of the Z3 solver (slow, but can handle complex policies) with evaluating policies in Scala (faster, but can't handle as complex policies)

# Acknowledgements

- Our mentor, Jean Yang, for guiding us through the project
- Jean Yang and Kuat Yessenov for developing the Jeeves library
- Dr. Khovanova and other PRIMES staff for organizing the PRIMES program
- Our parents for providing transportation to and from weekly meetings