THE BIG PICTURE

User -> Ask a machine

Ranks the Content

Index

Social Graph

Ranks the Users

Response

Ask a human
Zing: User Behavior Analysis

User interests: *networking*

- Find old friends
- Find friend with mutual interests
- Find game play friends
- Find friend within group
- Find friend around a location

Problems

Result is *not relevant* to user's network.

Browsing *thousands of results* is a pain.
Solution

Social Search + Faceted Navigation
Zing Friend Finder

Showing 342 results for "Kien"

First prototype
BIG DATA

26  avg friend/user

30,000  avg friends of friends/user

800,000  maximum friends of friends/user

17,000,000  users in index*

* test index
STRICT SLA

Worst case: 1s (1000ms) / request.
CHALLENGES

Check relationship

Ranking user's affinity

ONLINE
CHALLENGES

Example:
User A executes a query: "Nam" => 500,000 hits
Social search: check each hit's relationship with user A.

$$500,000 \times 0.1ms = 50s!!$$

BOTTLENECK!
Built a **fast, memory-efficient** graph database.

Built a **social facet handler**.

**Trade off** results: use bitset / filter.
SYSTEM MODEL

1. Query
2. Search through Lucene index
3. Retrieve hits from index
4. Retrieve relationship between each hit and user A.
5. Classify hits by relationship and user profiles
6. Render results

GraphDB server

Faceted search engine

UserProfile index

City facet handler
Social facet handler
Group facet handler

User A
PERFORMANCE EVALUATION

1. User's query
   - Simple query: "Kiên"
     - 200-300ms => FAST.
   - Complex query: "Phạm OR Kiên"
     - More hits => slow.

2. Facet
   - All facets is selected => slow.

3. User's network
   - More friends => more hits => slow.
EXPERIMENTAL RESULTS

Query execution time

Execution time (seconds)

single name query  full name query

min  avg  max
DISCUSSIONS