SmartP2P: A Multiobjective Framework for Finding Social Content in P2P Smartphone Networks

Andreas Konstantinidis*†, Christos Aplitsiotis* and Demetrios Zeinalipour-Yazti*
*Department of Computer Science, University of Cyprus, Nicosia, Cyprus
†Department of Computer Science and Engineering, Frederick University, Nicosia, Cyprus

***Abstract*—In this demonstration paper, we present a novel framework for searching objects (e.g., images, videos, etc.) captured by the users in a mobile social community. Our framework, is founded on an in-situ data storage model, where captured objects remain local on their owners smartphones and searches then take place over a novel lookup structure we compute dynamically. Initially, a query user invokes a search to find an object of interest. Our structure concurrently optimizes several conflicting objectives (i.e., it minimizes energy consumption, minimizes search delay and maximizes query recall), using a Multi-Objective Optimization approach and calculates a diverse set of high quality non-dominated Query Routing Trees (QRTs), in a single run. The optimal set is then forwarded to the query user (decision maker) to select a particular QRT to be searched based on instant requirements and preferences.**

To demonstrate the capabilities of SmartP2P during the conference, we will utilize our cloud of smartphone devices, i.e. the SmartLab testbed composed of 40+ Android smartphones and tablets, as well as mobility and social patterns derived by Microsofts Geolife project, DBLP and Pics n Trails. We will allow the attendees to use a real SmartLab Android device to query our local Smartphone Network using any of the four algorithmic choices provided by the SmartP2P framework. The query device will then be provided with the optimal QRTs and the attendees will be able to visually decide the optimal QRT to be searched. A P2P search on the Smartphone Network will follow making available to the query user the desired objects of interest, in an optimal manner. The conference attendees will be able to appreciate how social content can be efficiently shared with other attendees within close proximity without revealing their personal content to a centralized authority.

I. Introduction

The widespread deployment of smartphone devices and the advent of social networks have brought a revolution in social-oriented applications and services for mobile phones. There is already a proliferation of innovative applications [1], [2] founded on the concept of a smartphone network[1]. For example, Google Latitude [3] enables users to track the places they and their social network have visited. The given service already reports over 3M enrolled users and over 1M active users, despite the controversial privacy concerns. Similarly, mobile social networking applications like Foursquare, Gowalla and Loopt enjoy enormous success in the Smartphone community and academic efforts in this direction are also underway [4].

Currently, the bulk of social networking services, designed for smartphone communities, rely on centralized or cloud-like architectures. In particular, in order to enable content sharing and community search, the smartphone clients upload their captured objects (e.g., images uploaded to Twitter, video traces uploaded to Youtube, etc.) to a central entity that subsequently takes care of the content organization and dissemination tasks, violating the following crucial constraints:

i. **Data-Disclosure Constraints:** Continuously disclosing user-captured objects to a central entity might compromise user privacy in very serious ways[2].
ii. **Energy Constraints:** Smartphones have expensive communication mediums, thus by continuously transferring massive amounts of data to a query processor, through WiFi/3G/4G connections, can both deplete the precious smartphone battery faster, increase query response times, but can also quickly degrade the network health.

In this paper, we present techniques to enable smartphone users keep their data *in-situ*, for data-disclosure and performance reasons, offering at the same time high performance search capabilities over other user's data in the social community. When a user invokes a search to find an object of interest, e.g., *"Pictures of street artists performing in Manhattan"*, the user first downloads a *Query Routing Tree (QRT)* $\mathcal{X}$ from a server, where $\mathcal{X}$ is tuned to optimize several objectives concurrently during searches in a smartphone network. In particular, the QRTs proposed in this work (i) minimize energy consumption during search; (ii) minimize the query response time in conducting the search; and (iii) maximize the recall rate of the user query in a *Multi-objective Optimization (MOO)* manner. This is because there is no single solution that can optimize all objectives at the same time due to their conflicting correlation, but a near-optimal set of non-dominated solutions, commonly known as the *Pareto Front (PF),* to provide to the decision maker. In order to do that we utilize several *Multi-Objective Evolutionary Algorithms (MOEAs)* that have been shown effective in obtaining a set of optimal solutions in a single run [6].

We start out this paper with a high level description of the system model and the problem formulation as well as the search algorithms behind the SmartP2P. We then present the SmartP2P framework and the real Android implementation. Finally, we present our demonstration setting and plan.

[1]We define a Smartphone Network as "*a set of smartphone devices that communicate in an unobtrusive manner, without explicit user interactions, in order to realize a collaborative or social task.*"

[2]"Google Apologizes for Buzz Privacy", David Coursey, PC World Business Center (online), Feb. 15th, 2010.

II. System Model and Problem Formulation

Overview of System Model: Let $\mathcal{C}$, denote a social networking service that maintains centrally the profiles $\mathcal{P} = \{p_1, p_2, ..., p_M\}$, for each of its M subscribed users (i.e., $\mathcal{U} = \{u_1, u_2, ..., u_M\}$). The profiles record basic user details, authentication credentials, the user interests (e.g., traveling, sports, music, etc.) and friendship relations that define the conceptual social network graph $\mathcal{G}$ among the M users. In our setting, a user u_i ($i \leq M$) uses a smartphone (or tablet) device to both perform its day-to-day activities but also to capture objects of interest at arbitrary moments (e.g., "take a picture of the Liberty Statue"). Each object o_{ik} might be tentatively *"tagged"* with GPS information and other user tags (e.g., *"lat: 40.689201355, long: -74.0447998047, tags: "Statue Liberty Ellis Island"*).

Connection Modalities: Each u_i features different Internet connection modalities that provide intermittent connectivity to $\mathcal{C}$ (e.g., WiFi, 2G/3G/4G). Each u_i also features peer-to-peer connection modalities that provide connectivity to nodes in spatial proximity (e.g., Bluetooth or Portable WiFi available in Android). We assume that when u_i is connected to $\mathcal{C}$, then $\mathcal{C}$ is aware of u_i's absolute location (e.g., GPS) or u_i's relative location (e.g., the cell-ids within u_i's range, WiFi RSS indicators within u_i's range or other means utilized for geo-location). Notice that each of the connection modalities comes at different energy and data transfer rate characteristics.

Search Techniques: Now let an arbitrary user u_j ($j \leq M$), be interested in answering a query[3] $\mathcal{Q}$ over its social neighborhood $\mathcal{G}'$ ($\mathcal{G}' \subseteq \mathcal{G}$). For instance, let $\mathcal{Q}$ be a depth-bounded breadth first search query over u_j's neighbors in the $\mathcal{G}$ graph (i.e., in $\mathcal{G}'$). This kind of conceptual query can be realized in the following manners:

1) *Centralized Search (CS):* This algorithm assumes that the multimedia objects and tags are all uploaded to $\mathcal{C}$ prior query execution. Once $\mathcal{Q}$ is posted, $\mathcal{C}$ can locally derive the answers (using its local tag database) and return the answers to u_j. .
2) *Distributed Random Search (DRS):* This algorithm assumes that the objects and tags are all stored in-situ (on their owner's smartphones). In order to realize the search task, a querying node u_j downloads from $\mathcal{C}$ the addresses (e.g., IP) of its first line neighboring nodes. u_j then contacts these nodes in order to conduct a search in a P2P fashion. Once some arbitrary node u_x receives $\mathcal{Q}$, it looks both at its local tags to identify an answer and forwards the request further to the network.

Although the DRS approach improves the data-disclosure drawback of the *CS* algorithm, it is quite inefficient during search. In particular, $\mathcal{Q}$ has to go over a random neighborhood rather than a neighborhood that is contextually related to the query. For instance, in our Liberty Statue query example, we would have preferred querying a friend living in lower Manhattan rather than a person living in California (as the former would have a higher probability of capturing the statue). Also, if u_j had two friends, u_x and u_y, both living in lower Manhattan, with u_x being in spatial proximity to u_j during the query (i.e., within a few meters), while u_y being far away, would have made u_x a better choice for posting the query (as u_x could have been queried through a local link such as bluetooth).

[3]Without loss of generality we assume simple keyword queries over tags

Problem Formulation: The *Multi-Objective Query Routing Tree (MO-QRT)* structure, proposed in this paper, improves the search operation of the DRS algorithm by optimizing the neighbor selection process. In particular, a node downloads from $\mathcal{C}$ a QRT $\mathcal{X}$ that is optimized according to the following formulation: *Given a social network of users $\mathcal{U}$, a list of active users $\mathcal{U}'$ and their coordinates, the profiles $\mathcal{P}$ of these users and a query $\mathcal{Q}$, posted by an arbitrary user u_j, $\mathcal{C}$ aims to optimize an $\mathcal{X}$ structure using the following* **objectives**:

Objective 1: *Minimize the total* Energy *consumption of $\mathcal{X}$*

$$Energy(\mathcal{X}) = min \sum_{\forall (u_a, u_b) \in \mathcal{X} (\mathcal{X} \subseteq \mathcal{U}')} e(u_a, u_b) \quad (1)$$

where, $e(u_a, u_b)$ denotes the energy consumption for transmitting one bit of data over the respective edge (WiFi, Bluetooth and 3G).

Objective 2: *Minimize the* Time *overhead of $\mathcal{X}$*

$$Time(\mathcal{X}) = min(max_{(u_a, u_b) \in \mathcal{X}} t(u_a, u_b)) \quad (2)$$

where, $t(u_a, u_b)$ denotes the delay in transmitting one bit of data over the respective edge.

Objective 3: *Maximize the* Recall *rate of $\mathcal{X}$*

$$Recall(\mathcal{X}, \mathcal{Q}) = max(\frac{Relevant(\mathcal{Q}) \cap Retrieved(\mathcal{X}, \mathcal{Q})}{Relevant(\mathcal{Q})}) \quad (3)$$

where *Relevant($\mathcal{Q}$)* denotes the set of all objects in $\mathcal{U}'$ that are relevant to $\mathcal{Q}$, formally as: *Relevant($\mathcal{Q}$)* = $\bigcup_{\forall u_a \forall k (u_a \in \mathcal{U}')} (o_{ak})$), given that u_a's profile (denoted as p_a) contains terms found in $\mathcal{Q}$. On the other hand, *Retrieved($\mathcal{X}$,$\mathcal{Q}$)* denotes the set of objects that have been retrieved in response to $\mathcal{Q}$ over structure $\mathcal{X}$, formally as *Retrieved($\mathcal{X}$, $\mathcal{Q}$)* = $\bigcup_{\forall u_a \forall k (u_a \in \mathcal{X})} (o_{ak})$), again given that p_a contains terms found in $\mathcal{Q}$.

In a MOP, there is no single solution $\mathcal{X}$ that optimizes all objectives simultaneously, but a set of trade-off candidates. The set of trade-off solutions, commonly known as the Pareto Front (PF), is often defined in terms of Pareto Optimality [5].

III. SmartP2P Framework and Implementation

The SmartP2P general framework is composed of three phases as illustrated in Figure 1. Let an arbitrary user generate a query $\mathcal{Q}$ and forward it to the SmartP2P optimizer. In Phase 1, the optimizer is employed for finding a diverse and high-quality set of non-dominated smartphone QRTs (Pareto Front) that can facilitate the resolution of $\mathcal{Q}$. The Pareto Front is then forwarded to the query user (decision maker) that sets

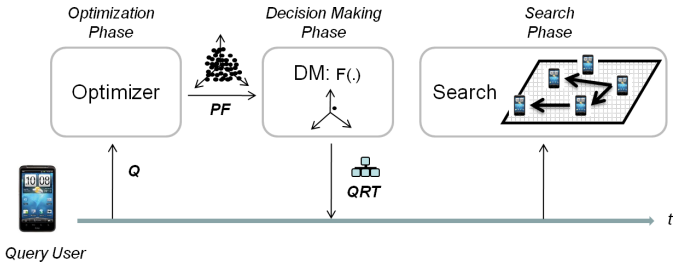

Fig. 1. The SmartP2P framework.

its preference with respect to the three objectives based on instant requirements. In Phase 2, the decision maker decides the QRT $\mathcal{X}^*$ that is closer to the user's choice an forwards it back to the query user. In Phase 3, the query user utilizes QRT $\mathcal{X}^*$ to search the peer-to-peer network and find objects of interest recorded by all users in $\mathcal{X}^*$ and related to query $\mathcal{Q}$. We have developed a prototype system that realizes the SmartP2P framework as described below.

The SmartP2P Optimizer: Figure 2 (center) shows the GUI through which a query, for example of the keyword "optimization", is formulated in order to find objects of interest. In the GUI, the group of algorithmic choices provided by the SmartP2P framework is shown. SmartP2P provides (a) two simple distributed choices, i.e. Random Search and Breadth-First Search, as well as (b) two MOO choices, i.e. the MOEA based on Decomposition (MOEA/D) and the Non-Dominated Sorting Genetic Algorithm II (NSGA-II). The user selects an algorithm and presses the "Go" button. Then the SmartP2P optimizer calculates a QRT in case (a) or a Pareto Front in case (b). In both cases the result is returned to the query user.

Fig. 2. The SmartP2P Android GUI. The intro screen (left), the keyword search optimization with four algorithmic choices screen (center) and the resulted Pareto Front screen for decision making (right). The Pareto Front is forwarded from the optimizer and displayed to the user's GUI. The slide bar shows the user's preference with respect to the recall and time overhead objectives, if the decision making checkbox remains untick then the QRT with the minimum energy is automatically selected.

The SmartP2P Decision Maker: The decision maker is only enabled when the query user selects an algorithm from (b) to perform the search. In this case, the Pareto Front is forwarded and displayed to the query user as shown in Figure 2 (right) (note that the image can zoom in/out for better visualization). Then the query user makes use of the slide bar below the Pareto Front image to set a desired level of time and recall of the search to be initiated. Note that if the user does not choose a desired level of those two objectives, the solution with the minimum energy consumption is automatically chosen. By pressing the "Go" button, the decision maker finds the QRT that is closer to the user's choice and downloads it from the optimizer to the user's smartphone.

The SmartP2P Search: The query user initiates the search. The results of the search as well as the selected QRT are both displayed on the user's Smartphone as shown in Figure 3.

Fig. 3. The results retrieved after initiating a P2P search on the Smartphone network (left). The QRT selected by the query user and initiated to retrieve the objects of interest (right).

Our client-side software is developed around the SDK Tools r12 of Android 2.2 and its installation package (i.e. APK) has a size of 327KB. Our code is written in JAVA and consists of around 7500 lines of code. In particular our server-code (i.e. optimizer) uses 5000LOC and runs over JDK 6 and Ubuntu Linux, out smartphone code uses 1600 LOC plus 250 lines of XML elements. The server side also includes a Microsoft SQL server R2 of around 700 LOC and utilizes the JMATH-PLOT package for drawing the Pareto Front images.

IV. Demonstration Settings

For the actual demo at the conference we will use SmartLab.

A. SmartLab

Is an innovative programming cloud[4] of approximately 40+ Android smartphones and tablets, which is deployed at the University of Cyprus. Its intuitive web-based interface is easy to use and provides the ability to reserve and use Android devices for a desired amount of time. Users are able to reboot, list, transfer and remove files, change Android device settings by using the interactive Android Debugging (ADB) shell session. Additionally, registered users can upload and install executable APK files on their reserved Android devices simultaneously. The *SmartLab* users are also able to extract application data, output and results automatically from all reserved devices, take screenshots as well as watch the display of all reserved devices during runtime. Users are also granted access to log files for error and exception handling purposes.

[4]Available at: http://smartlab.cs.ucy.ac.cy

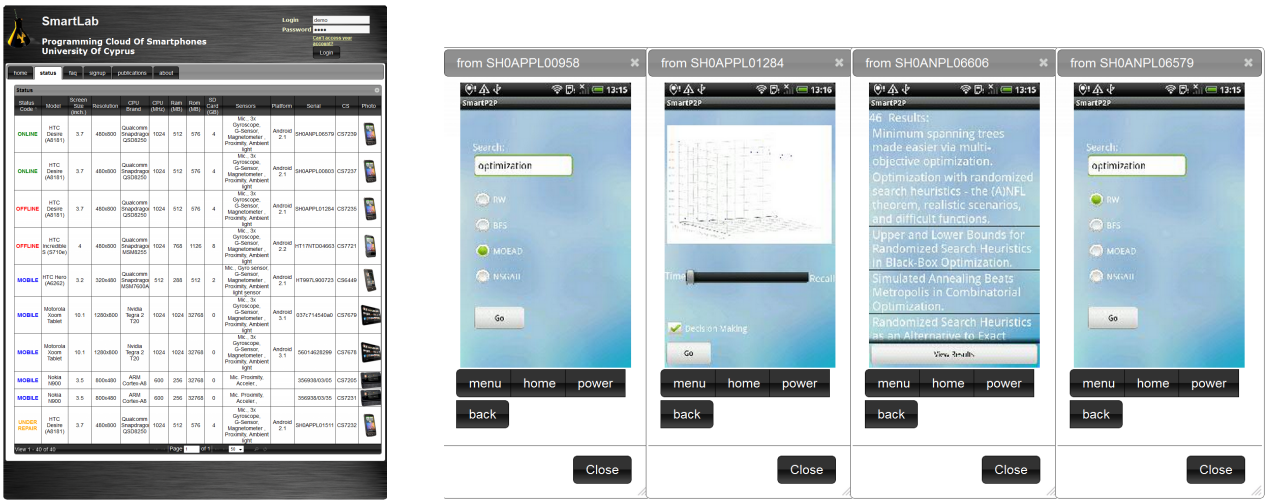

Fig. 4. The SmartLab cloud of Smartphone devices (left). A screenshot of the SmartP2P on SmartLab (right)

B. Demo plan

At the conference site, and prior demonstration, we will allow participants to register to the SmartLab and reserve devices. We will offer tutorials to the attendees of how to use SmartLab to manage Android devices. Then we will allow the users to select a data set as well as a mode, which follow, for running the SmartP2P on SmartLab.

Data sets: We will allow the Demo users to select among a number of available mobility and social patterns:

Mobility i) *GeoLife* [7]: This real dataset by Microsoft Research Asia, includes 1,100 trajectories of a human moving in the city of Beijing over a life span of two years (2007-2009). The average length of each trajectory is $190,110 \pm 126,590$ points, while the maximum trajectory length is 699,600 points. Notice that 95% of the GeoLife dataset refers to a granularity of 1 sample every 2-5 seconds or every 5-10 meters.

Social i) *DBLP* [8]: This real dataset by the DBLP Computer Science Bibliography website, includes over 1.4 million publications in XML format. In particular, the dataset records the paper titles, paper urls, co-authors, links between papers and authors and other useful semantics. In order to map this dataset to our problem, we assume that each object is an author's paper. We also assume that each object is "tagged" by the keywords found in the paper title.

Mobility and Social ii) *Pics n Trails* [9], [10]: This is a real data set composed of GPS traces of users moving in Tokyo, Japan during the year 2007-2008 and a collection of photos, tagged by the location taken and a short description. In particular the data set is comprised by 4179 photos of sightseeing and events in Japan as well as trajectories with a granularity 1 sample every 10-15 seconds.

Modes: After the attendees get familiar with the SmartLab environment and select a mobility and social pattern they will be asked to choose a demo mode plan. The demo modes are classified based on the execution of the installation packages on the Android devices through the SmartLab environment. The application files can be executed either interactively with remote screens that mimic the clicks of the mouse as touches on the screen or by running Android Monkey-runner scripts (written in python). The latter scripts, can instruct the target devices to conduct pre-specified keystrokes automating in this manner any desired functionality, as opposed to manually entering this input on each and every device.

Based on the selected data sets and modes we will let the attendees to query the SmartLab smartphone network with any of the four algorithmic choices, perform decision making tasks by visually deciding near-optimal QRTs from Pareto Fronts, which will be obtained in real-time, initiate P2P searches and retrieve objects of interests (e.g. images, text etc.) from the smartphone network. Through our demo, the conference attendees will be able to appreciate how social content can be efficiently shared between smartphone users optimizing multiple conflicting user-oriented, network-oriented and system-oriented objectives, simultaneously, without disclosing their personal data to a central authority ensuring privacy.

REFERENCES

[1] Azizyan M., Constandache I., Choudhury R.-R., "SurroundSense: mobile phone localization via ambience fingerprinting," In *MobiCom'09*.
[2] Campbell A., Eisenman S., Lane N., Miluzzo E., and Peterson R., "People-centric urban sensing," In *WICON*, 2006.
[3] Google Latitude, 11/2010, http://www.google.com/latitude
[4] Sarigol E., Riva O., Alonso G., "A tuple space for social networking on mobile phones," In *ICDE*, 2010.
[5] Deb K., Pratap a., Agarwal S., Meyarivan T., "A Fast and Elitist Multiobjective Genetic Algorithm: NSGA II," *IEEE TEvC*, 2002.
[6] Zhang Q., Li H., "MOEA/D: A Multi-objective Evolutionary Algorithm Based on Decomposition," IEEE TEvC, 2007.
[7] Zheng Y., Liu L., Wang L., Xie X., "Learning transportation mode from raw gps data for geographic applications on the web," In *WWW'08*, 2008.
[8] DBLP Computer Science Bibliography, 2010.
[9] G. C. de Silva, T. Yamasaki, K. Aizawa, "Sketch-Based Spatial Queries for Retrieving Human Locomotion Patterns From Continuously Archived GPS Data," IEEE Trans. on Multimedia, vol.11, no.7, 2009.
[10] G. C. de Silva, K. Aizawa, "Retrieving Multimedia Travel Stories Using Location Data and Spatial Queries", In Proc. The 17th ACM International Conference on Multimedia, pp. 785-788, October 2009.