Proposal of an Economical and Feasible Distributed SOS System for Mobile Users to Generate High Quality Answers

S.Mineesha[1]; Y.Sandeep[2]& Prof.Dr.G.Manoj Someswar[3]

[1]M.Tech.(CSE) from Sai Sudhir Institute of Engineering & Technology, Affiliated to JNTUH, Hyderabad, Telangana, India

[2]M.Tech. (CSE), Associate Professor & HOD, Department of CSE, Sai Sudhir Institute of Engineering & Technology, Affiliated to JNTUH, Hyderabad, Telangana, India

[3]B.Tech., M.S.(USA), M.C.A., Ph.D., Principal & Professor, Department Of CSE, Anwar-ul-uloom College of Engineering & Technology, Affiliated to JNTUH, Vikarabad, Telangana, India

ABSTRACT:

Recently, emerging research efforts have been focused on question and answer (Q&A) systems based on social networks. The social-based Q&A systems can answer non-factual questions, which cannot be easily resolved by web search engines. These systems either rely on a centralized server for identifying friends based on social information or broadcast a user's questions to all of its friends. Mobile Q&A systems, where mobile nodes access the Q&A systems through Internet, are very promising considering the rapid increase of mobile users and the convenience of practical use. However, such systems cannot directly use the previous centralized methods or broadcasting methods, which generate high cost of mobile Internet access, node overload, and high server bandwidth cost with the tremendous number of mobile users. We propose a distributed Social-based mobile Q&A System (SOS) with low overhead and system cost as well as quick response to question askers. SOS enables mobile users to forward questions to potential answerers in their friend lists in a decentralized manner for a number of hops before resorting to the server. It leverages lightweight knowledge engineering techniques to accurately identify friends who are able to and willing to answer questions, thus reducing the search and computation costs of mobile nodes. The trace-driven simulation results show that SOS can achieve a high query precision and recall rate, a short response latency and low overhead. We have also deployed a pilot version of SOS for use in a small group in our Institute. The feedback from the users shows that SOS can provide high-quality answers.

Keywords: People Helping One Another Know Stuff (PHOAKS); Social Sim Rank (SSR); Social Page Rank (SPR); Social Sim Rank(SSR); Social-based mobile Q&A System (SOS); Profile-Based Personalizatio*n*

INTRODUCTION

Distributed computing is a field of computer science that studies distributed systems. A distributed system is a software system in which components located on networked computers communicate and coordinate their actions by passing messages. The components interact with each other in order to achieve a common goal. There are many alternatives for the message passing mechanism, including RPC-like connectors and message queues.[1] Three significant characteristics of distributed systems are: concurrency of components, lack of a global clock, and independent failure of components. An important goal and challenge of distributed systems is location transparency. Examples of distributed systems vary from SOA-based systems to massively multiplayer online games to peer-to-peer applications.

A computer program that runs in a distributed system is called a distributed program, and distributed programming is the process of writing such programs. Distributed computing also refers to the use of distributed systems to solve computational problems. In

distributed computing, a problem is divided into many tasks, each of which is solved by one or more computers, which communicate with each other by message passing.[2]

The word *distributed* in terms such as "distributed system", "distributed programming", and "distributed algorithm" originally referred to computer networks where individual computers were physically distributed within some geographical area. The terms are nowadays used in a much wider sense, even referring to autonomous processes that run on the same physical computer and interact with each other by message passing.[3] While there is no single definition of a distributed system, the following defining properties are commonly used:

- There are several autonomous computational entities, each of which has its own local memory.
- The entities communicate with each other by message passing.

In this article, the computational entities are called *computers* or *nodes*.

A distributed system may have a common goal, such as solving a large computational problem.] Alternatively, each computer may have its own user with individual needs, and the purpose of the distributed system is to coordinate the use of shared resources or provide communication services to the users.[4]

Other typical properties of distributed systems include the following:

- The system has to tolerate failures in individual computers.
- The structure of the system (network topology, network latency, number of computers) is not known in advance, the system may consist of different kinds of computers and network links, and the system may change during the execution of a distributed program.

- Each computer has only a limited, incomplete view of the system. Each computer may know only one part of the input.[5]

Distributed systems are groups of networked computers, which have the same goal for their work. The terms "concurrent computing", "parallel computing", and "distributed computing" have a lot of overlap, and no clear distinction exists between them. The same system may be characterized both as "parallel" and "distributed"; the processors in a typical distributed system run concurrently in parallel. Parallel computing may be seen as a particular tightly coupled form of distributed computing, and distributed computing may be seen as a loosely coupled form of parallel computing. Nevertheless, it is possible to roughly classify concurrent systems as "parallel" or "distributed" using the following criteria:

- In parallel computing, all processors may have access to a shared memory to exchange information between processors.
- In distributed computing, each processor has its own private memory (distributed memory). Information is exchanged by passing messages between the processors.[6]
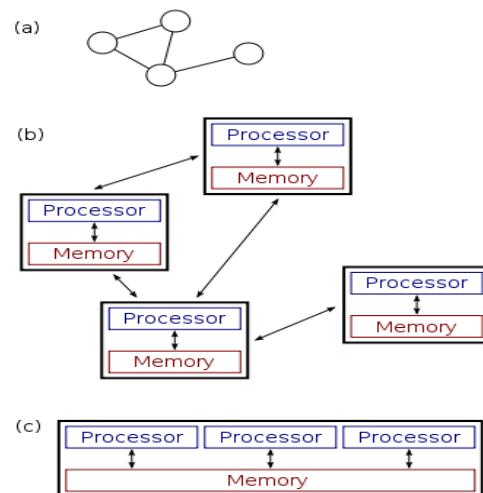

Figure 1: Differences between Distributed & Parallel Computing

The above figure 1 illustrates the difference between distributed and parallel systems. Figure (a) is a schematic view of a typical distributed system; as usual, the system is represented as a network topology in which each node is a computer and each line connecting the nodes is a communication link. Figure (b) shows the same distributed system in more detail: each computer has its own local memory, and information can be[7] exchanged only by passing messages from one node to another by using the available communication links. Figure (c) shows a parallel system in which each processor has a direct access to a shared memory.

The situation is further complicated by the traditional uses of the terms parallel and distributed *algorithm* that do not quite match the above definitions of parallel and distributed *systems*; see the section Theoretical foundations below for more detailed discussion. Nevertheless, as a rule of thumb, high-performance parallel computation in a shared-memory multiprocessor uses parallel algorithms while the coordination of a large-scale distributed system uses distributed algorithms.[8]

Wikipedia defines a social network service as a service which "focuses on the building and verifying of online social networks for communities of people who share interests and activities, or who are interested in exploring the interests and activities of others, and which necessitates the use of software."

A report published by OCLC provides the following definition of social networking sites: "Web sites primarily designed to facilitate interaction between users who share interests, attitudes and activities, such as Facebook, Mixi and MySpace."

Social networks can provide a range of benefits to members of an organization:

Support for learning: Social networks can enhance informal learning and support social connections within groups of learners and with those involved in the support of learning.

Support for members of an organization: Social networks can potentially be used my all members of an organisation, and not just those involved in working with students. Social networks can help the development of communities of practice. [9]

Engaging with others: Passive use of social networks can provide valuable business intelligence and feedback on institutional services (although this may give rise to ethical concerns).

Ease of access to information and applications: The ease of use of many social networking services can provide benefits to users by simplifying access to other tools and applications. The Facebook Platform provides an example of how a social networking service can be used as an environment for other tools.

Common interface: A possible benefit of social networks may be the common interface which spans work / social boundaries. Since such services are often used in a personal capacity the interface and the way the service works may be familiar, thus minimising training and support needed to exploit the services in a professional context. This can, however, also be a barrier to those who wish to have strict boundaries between work and social activities.[10]

Examples of Social Networking Services

Examples of popular social networking services include:

Facebook: Facebook is a social networking Web site that allows people to communicate with their friends and exchange information. In May 2007 Facebook launched the Facebook Platform which provides a framework for developers to create applications that interact with core Facebook features

MySpace: MySpace is a social networking Web site offering an interactive, user-submitted network of friends, personal profiles, blogs and groups,

commonly used for sharing photos, music and videos..

Ning: An online platform for creating social Web sites and social networks aimed at users who want to create networks around specific interests or have limited technical skills.

Twitter: Twitter is an example of a micro-blogging service. Twitter can be used in a variety of ways including sharing brief information with users and providing support for one's peers.

Note that this brief list of popular social networking services omits popular social sharing services such as Flickr and YouTube.[11]

Opportunities and Challenges

The popularity and ease of use of social networking services have excited institutions with their potential in a variety of areas. However effective use of social networking services poses a number of challenges for institutions including long-term sustainability of the services; user concerns over use of social tools in a work or study context; a variety of technical issues and legal issues such as copyright, privacy, accessibility; etc.

Institutions would be advised to consider carefully the implications before promoting significant use of such services.

LITERATURE SURVEY

PHOAKS (People Helping One Another Know Stuff) is an experimental system designed to help users locate information on the World Wide Web. [12] Using a collaborative filtering approach, it sifts through Usenet news messages and automatically identifies, keeps track of and redeploys recommendations of Web resources. PHOAKS differs from other recommender systems in that it is based on the principles of role specialization and reuse. Association for Computing Machinery Inc. A collaborative filtering system that recognizes and reuses recommendations. Finding relevant, high-quality information on the world-wide Web is a difficult problem. PHOAKS (People Helping One Another

Know Stuff) is an experimental system that addresses this problem through a collaborative filtering approach. PHOAKS works by automatically recognizing, tallying, and redistributing.

This research explores new ways to augment the search and discovery of relations between Web 2.0 entities using multiple types and sources of social information. Our goal is to allow the search for all object types such as documents, persons and tags, while retrieving related objects of all types. We implemented a social-search engine using a unified approach, where the search space is expanded to represent heterogeneous information objects that are interrelated by several relation types. Our solution is based on multifaceted search, which provides an efficient update mechanism for relations between objects, as well as efficient search over the heterogeneous data.[13]We describe a social search engine positioned within a large enterprise, applied over social data gathered from several Web 2.0 applications. We conducted a large user study with over 600 people to evaluate the contribution of social data for search. Our results demonstrate the high precision of social search results and confirm the strong relationship of users and tags to the topics retrieved.

This work investigates personalized social search based on the user's social relations -- search results are re-ranked according to their relations with individuals in the user's social network. We study the effectiveness of several social network types for personalization: (1) Familiarity-based network of people related to the user through explicit familiarity connection; (2) Similarity-based network of people "similar" to the user as reflected by their social activity; (3) Overall network that provides both relationship types. For comparison we also experiment with Topic-based personalization that is based on the user's related terms, aggregated from several social applications. We evaluate the contribution of the different personalization strategies by an off-line study and by a user survey within our organization. In the off-line study we apply bookmark-

based evaluation, suggested recently, that exploits data gathered from a social bookmarking system to evaluate personalized retrieval. In the on-line study we analyze the feedback of 240 employees exposed to the alternative personalization approaches. Our main results show that both in the off-line study and in the user survey social network based personalization significantly outperforms non-personalized social search. Additionally, as reflected by the user survey, all three SN-based strategies significantly outperform the Topic-based strategy.[14]

This paper explores the use of social annotations to improve web search. Nowadays, many services e.g., del.icio.us have been developed for web users to organize and share their favorite web pages on line by using social annotations. We observed that the social annotations can benefit the web search in two aspects: 1) the annotations are usually good summaries of corresponding web pages; 2) the count of annotations indicates the popularity of web pages. Two novel algorithms are proposed to incorporate these information into page ranking: 1) SocialSimRank (SSR) calculates the similarity between social annotations and web queries; 2) SocialPageRank (SPR) captures the popularity of web pages. Preliminary experimental results show that SSR can find the latent semantic association between queries and annotations, while SPR successfully measures the quality (popularity) of a web page from the web users perspective. We further empirically evaluate the proposed methods with 50 manually annotated queries and 3000 auto-generated queries, on a dataset consisting of 690,482 web pages with 2,879,614 different annotations. Experiments show that both SSR and SPR benefit the web search significantly. By incorporating both the SPR and SSR features, the quality of search results can be improved by as much as 14.80% and 25.02% compared with the original performance in MAP on two query sets respectively.

Collaborative research has been increasingly popular and important in academic circles. However, there is no open platform available for scholars or scientists to effectively discover potential collaborators. This paper discusses CollabSeer, an open system to recommend potential research collaborators for scholars and scientists. CollabSeer discovers collaborators based on the structure of the coauthor network and a user's research interests. Currently, three different network structure analysis methods that use vertex similarity are supported in CollabSeer: Jaccard similarity, cosine similarity, and our relation strength similarity measure. Users can also request a recommendation by selecting a topic of interest. The topic of interest list is determined by CollabSeer lexical analysis module, which analyzes the key phrases of previous publications. The CollabSeer system is highly modularized making it easy to add or replace the network analysis module or users' topic of interest analysis module. CollabSeer integrates the results of the two modules to recommend collaborators to users. Initial experimental results over a subset of the CiteSeerX database show that CollabSeer can efficiently discover prospective collaborators.[15]

SYSTEM STUDY
FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are

- ECONOMICAL FEASIBILITY
- TECHNICAL FEASIBILITY
- SOCIAL FEASIBILITY

ECONOMICAL FEASIBILITY

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

SOCIAL FEASIBILITY

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.
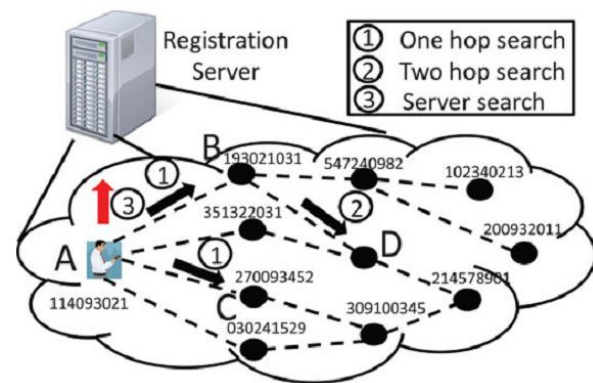
SYSTEM DESIGN

SYSTEM ARCHITECTURE:

Figure 2 :System Architecture

DATA FLOW DIAGRAM:

1. The DFD is also called as bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of input data to the system, various processing carried out on this data, and the output data is generated by this system.

2. The data flow diagram (DFD) is one of the most important modeling tools. It is used to model the system components. These components are the system process, the data used by the process, an external entity that interacts with the system and the information flows in the system.

3. DFD shows how the information moves through the system and how it is modified by a series of transformations. It is a graphical technique that depicts information flow and the transformations that are applied as data moves from input to output.

4. DFD is also known as bubble chart. A DFD may be used to represent a system at any level of abstraction. DFD may be partitioned into levels that represent increasing information flow and functional detail.
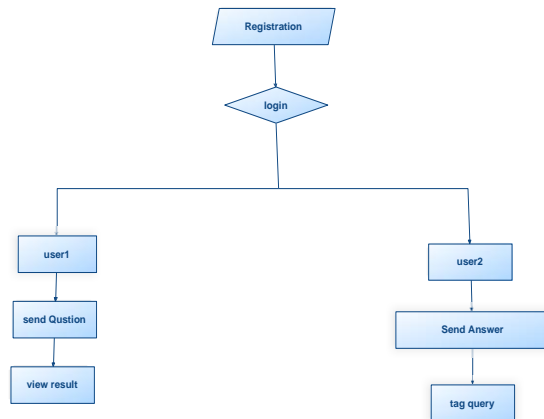
Figure3 : Data Flow Diagram

UML DIAGRAMS

UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group.

The goal is for UML to become a common language for creating models of object oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.

The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems.

The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems.

The UML is a very important part of developing objects oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

GOALS:

The Primary goals in the design of the UML are as follows:

1. Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.
2. Provide extendibility and specialization mechanisms to extend the core concepts.
3. Be independent of particular programming languages and development process.
4. Provide a formal basis for understanding the modeling language.
5. Encourage the growth of OO tools market.
6. Support higher level development concepts such as collaborations, frameworks, patterns and components.
7. Integrate best practices.

USE CASE DIAGRAM:

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.
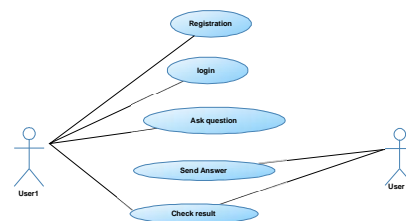

Figure 4 : Use Case Diagram

CLASS DIAGRAM:

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.
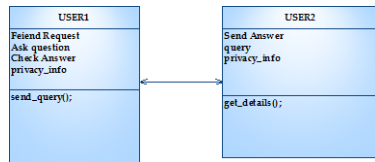
Figure 5: Class Diagram

SEQUENCE DIAGRAM:

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.
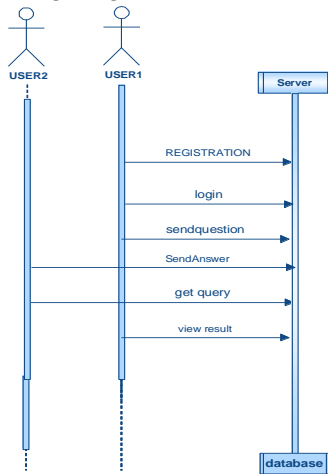

Figure6 : Sequence Diagram

ACTIVITY DIAGRAM:

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.
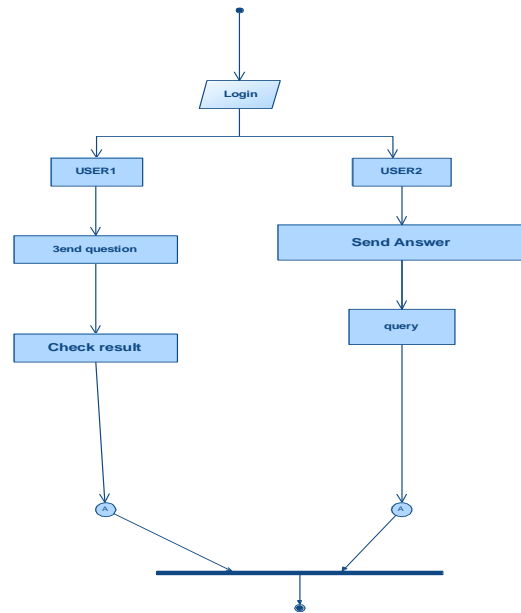

Figure 7 : Activity Diagram

INPUT DESIGN

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data in to a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy. Input Design considered the following things:

- ➢ What data should be given as input?
- ➢ How the data should be arranged or coded?
- ➢ The dialog to guide the operating personnel in providing input.
- ➢ Methods for preparing input validations and steps to follow when error occur.

OBJECTIVES

1. Input Design is the process of converting a user-oriented description of the input into a computer-based system. This design is important to avoid errors in the data input process and show the correct direction to the management for getting correct information from the computerized system.

2. It is achieved by creating user-friendly screens for the data entry to handle large volume of data. The goal of designing input is to make data entry easier and to be free from errors. The data entry screen is designed in such a way that all the data manipulates can be performed. It also provides record viewing facilities.

3. When the data is entered it will check for its validity. Data can be entered with the help of screens. Appropriate messages are provided as when needed so that the user will not be in maize of instant. Thus the objective of input design is to create an input layout that is easy to follow

OUTPUT DESIGN

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system's relationship to help user decision-making.

1. Designing computer output should proceed in an organized, well thought out manner; the right output must be developed while ensuring that each output element is designed so that people will find the system can use easily and effectively. When analysis design computer output, they should Identify the specific output that is needed to meet the requirements.

2. Select methods for presenting information.

3. Create document, report, or other formats that contain information produced by the system.

The output form of an information system should accomplish one or more of the following objectives.

- ❖ Convey information about past activities, current status or projections of the
- ❖ Future.
- ❖ Signal important events, opportunities, problems, or warnings.
- ❖ Trigger an action.
- ❖ Confirm an action.

SYSTEM ANALYSIS

EXISTING SYSTEM:

The search engines perform well in answering factual queries for information already in a database, they are not suitable for non-factual queries that are more subjective, relative and multi-dimensional, especially when the information is not in the database. One method to solve this problem is to forward the non-factual queries to humans, which are the most "intelligent machines "that are capable of parsing, interpreting and answering the queries, provided they are familiar with the queries. Accordingly, a number of expertise location systems have been proposed to search experts in social networks or Internet aided by a centralized search engine. To enhance the asker satisfaction on the Q&A sites, recently, emerging research efforts have been focused on social network based Q&A systems in which users post and answer questions through social network maintained in a centralized server. As the answerers in the social network know the backgrounds and preference of the askers, they are willing and able to provide more tailored and personalized answers to the askers. The social-based Q&A systems can be classified into two categories: broadcasting-based and centralized. The broadcasting-based systems broadcast the questions of a user to all of the user's friends. In the centralized systems, since the centralized server constructs and maintains the social network of each user, it searches the potential answerers for a given question from the asker's friends, friends of friends and so on.

DISADVANTAGES OF EXISTING SYSTEM:

1. Broadcasting and centralized methods are not suitable to the mobile environment, where each mobile node has limited resource.

2. Broadcasting to a large number of friends cannot guarantee the quality of the answers.

PROPOSED SYSTEM:

❋ In this research paper, we propose a distributed Social-based mobile Q&A System (SOS) with low node overhead and system cost as well as quick response to question askers. SOS is novel in that it achieves lightweight distributed answerer search, while still enabling a node to accurately identify its friends that can answer a question.

❋ We have also deployed a pilot version of SOS for use in a small group in Clemson University. The analytical results of the data from the real application show the highly satisfying Q&A service and high performance of SOS. SOS leverages the lightweight knowledge engineering techniques to transform users' social information and closeness, as well as questions to IDs, respectively, so that a node can locally and accurately identify its friends capable of answering a given question by mapping the question's ID with the social IDs. The node then forwards the question to the identified friends in a decentralized manner. After receiving a question, the users answer the questions if they can or forward the question to their friends. The question is forwarded along friend social links for a number of hops, and then to the server. The cornerstone of SOS is that a person usually issues a question that is closely related to his/her social life. As people sharing similar interests are likely to be clustered in the social network the social network can be regarded as social interest clusters intersecting with each other.

❋ By locally choosing the most potential answerers in a node's friend list, the queries can be finally forwarded to the social clusters that have answers for the question. As the answerers are socially close to the askers, they are more willing to answer the questions compared to strangers in the Q&A websites.

ADVANTAGES OF PROPOSED SYSTEM:

1. This avoiding the query congestion and high server bandwidth and maintenance cost problem.

2. Reducing the node overhead, traffic and mobile Internet access.

3. An asker identifies potential answerers from his/her friends based on their past answer quality and answering activeness to his/her questions.

SYSTEM TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

TYPES OF TESTS

Unit testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit

tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

Integration testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

Functional test

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

System Test

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

White Box Testing

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

Black Box Testing

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot "see" into it. The test provides inputs and responds to outputs without considering how the software works.

Unit Testing:

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

Test strategy and approach

Field testing will be performed manually and functional tests will be written in detail.

Test objectives

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.

Features to be tested
- Verify that the entries are of the correct format
- No duplicate entries should be allowed
- All links should take the user to the correct page.

Integration Testing

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

Acceptance Testing

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

<div align="center">

IMPLEMENTATION

</div>

MODULES:

1. Profile-Based Personalization.
2. Privacy Protection in PWS System.
3. Generalizing User Profile.
4. Online Decision.

MODULES DESCRIPTION:

Profile-Based Personalization:

This research paper introduces an approach to personalize digital multimedia content based on user profile information. For this, two main mechanisms were developed: a profile generator that automatically creates user profiles representing the user preferences, and a content-based recommendation algorithm that estimates the user's interest in unknown content by matching her profile to metadata descriptions of the content. Both features are integrated into a personalization system.

Privacy Protection in PWS System:

We propose a PWS framework called UPS that can generalize profiles in for each query according to user-specified privacy requirements. Two predictive metrics are proposed to evaluate the privacy breach risk and the query utility for hierarchical user profile. We develop two simple but effective generalization algorithms for user profiles allowing for query-level customization using our proposed metrics. We also provide an online prediction mechanism based on query utility for deciding whether to personalize a query in UPS. Extensive experiments demonstrate the efficiency and effectiveness of our framework.

Generalizing User Profile:

The generalization process has to meet specific prerequisites to handle the user profile. This is achieved by preprocessing the user profile. At first, the process initializes the user profile by taking the indicated parent user profile into account. The process adds the inherited properties to the properties of the local user profile. Thereafter the process loads the data for the foreground and the background of the map according to the described selection in the user profile.

Additionally, using references enables caching and is helpful when considering an implementation in a production environment. The reference to the user profile can be used as an identifier for already processed user profiles. It allows performing the customization process once, but reusing the result multiple times. However, it has to be made sure, that an update of the user profile is also propagated to the generalization process. This requires specific update strategies, which check after a specific timeout or a specific event, if the user profile has not changed yet. Additionally, as the generalization process involves remote data services, which might be updated frequently, the cached generalization results might

become outdated. Thus selecting a specific caching strategy requires careful analysis.

Online Decision:

The profile-based personalization contributes little or even reduces the search quality, while exposing the profile to a server would for sure risk the user's privacy. To address this problem, we develop an online mechanism to decide whether to personalize a query. The basic idea is straightforward. if a distinct query is identified during generalization, the entire runtime profiling will be aborted and the query will be sent to the server without a user profile.

RESULTS AND CONCLUSION

In this research paper, we present the design and implementation of a distributed Social-based mobile Q&A System (SOS). SOS is novel in that it achieves lightweight distributed answerer search, while still enables a node to accurately identify its friends that can answer a question. SOS uses the FOL representation and inference engine to derive the interests of questions, and interests of users based on user social information. A node considers both its friend's parsed interests and answer quality in determining the friend's similarity value, which measures both the capability and willingness of the friend to answer/forward a question. Compared to the centralized social network based Q&A systems that suffer from traffic congestions and high server bandwidth cost, SOS is a fully distributed system in which each node makes local decision on question forwarding. Compared to broadcasting, SOS generates much less overhead with its limited question forwarding hops. Since each user belongs to several social clusters, by locally selecting most potential answerers, the question is very likely to be forwarded to answerers that can provide answers. The low computation cost makes the system suitable for low-end mobile devices. We conducted extensive trace-driven simulations and implemented the system on iPod Touch/iPhone mobile devices. The results show that SOS can accurately identify answerers that are able to answer questions. Also, SOS earns high user satisfaction ratings on answering both factual and non-factual questions.

REFERENCES

[1] Google, http://www.google.com, 2013.

[2] Bing, http://www.bing.com, 2013.

[3] B.M. Evans and E.H. Chi, "An Elaborated Model of Social Search," J. Information Processing and Management, vol. 46, pp. 656-678, 2009.

[4] L. Terveen, W. Hill, B. Amento, D. McDonald, and J. Creter, "PHOAKS: A System for Sharing Recommendations," Comm. ACM, vol. 40, pp. 59-62, 1997.

[5] M.S. Ackerman, "Augmenting Organizational Memory: A Field Study of Answer Garden," ACM Trans. Information Systems, vol. 16, pp. 203-224, 1998.

[6] L.G. Terveen, P.G. Selfridge, and M.D. Long, "Living Design Memory: Framework, Implementation, Lessons Learned," Human-Computer Interaction, vol. 10, pp. 1-37, 1995.

[7] E. Amitay, D. Carmel, N. Har'El, S. Ofek Koifman, A. Soffer, S. Yogev, and N. Golbandi, "Social Search and Discovery Using a Unified Approach," Proc. 20th ACM Conf. Hypertext and Hypermedia (HT '09), 2009.

[8] D. Carmel, N. Zwerdling, I. Guy, S. Ofek-Koifman, N. Har'el, I. Ronen, E. Uziel, S. Yogev, and S. Chernov, "Personalized Social Search Based on the User's Social Network," Proc. 18th ACM Conf. Information and Knowledge Management (CIKM '09), 2009.

[9] S. Kolay and A. Dasdan, "The Value of Socially Tagged URLs for a Search Engine," Proc. 18th Int'l Conf. World Wide Web (WWW '09), 2009.

[10] S. Bao, G. Xue, X. Wu, Y. Yu, B. Fei, and Z. Su, "Optimizing Web Search Using Social Annotations," Proc. 16th Int'l Conf. World Wide Web (WWW '07), 2007.

[11] H.H. Chen, L. Gou, X. Zhang, and C.L. Giles, "CollabSeer: A Search Engine for Collaboration Discovery," Proc. 11th Ann. Int'l ACM/IEEE Joint Conf. Digital Libraries (JCDL '11), 2011.

[12] C.Y. Lin, N. Cao, S.X. Liu, S. Papadimitriou, J. Sun, and X. Yan, "Smallblue: Social Network Analysis for Expertise Search and Collective Intelligence," Proc. IEEE 25th Int'l Conf. In Data Engineering (ICDE '09), 2009.

[13] H. Kautz, B. Selman, and M. Shah, "Referral Web: Combining Social Networks and Collaborative Filtering," Comm. ACM, vol. 40, pp. 63-65, 1997.

[14] D.W. McDonald and M.S. Ackerman, "Expertise Recommender: A Flexible Recommendation System and Architecture," Proc. ACM Conf. Computer Supported Cooperative Work (CSCW '00), 2000.

[15] Yahoo answer, http://answers.yahoo.com, 2013.

Conference Chair: Prof.Dr.G.ManojSomeswar, Director General, Global Research Academy, Hyderabad, Telangana, India.
Papers presented in Conference can be accessed from www.edupediapublications.org/journals

P a g e | **329**