Workload Allocation Scheme for Mobile Social Network

Fiverose P A[1] Meera V[2] Priyanka[3] Athulya O K[4]
[1,2,3,4]Department of Bachelor of Technology in Computer Science
[1,2,3,4]Nehru College of Engineering and Research Centre Thrissur, India

Abstract— There exists many idle resources in the network which can be used to accomplish huge tasks. In this paper, we focus on those idle computational resources especially in mobile social networks (MSNs). Based on the unique features of MSN, open-crowdsourcing is introduced. In the proposed system, a large work is divided into smaller modules, and is distributed among the nodes. Still the partition of work segment is a critical problem. The lack of global information, leads to unbalanced workload distribution on participating nodes. In this paper, we find that, for a work at different processing stages, one should adopt distinct workload exchanging schemes. Based on this observation, we propose an adaptive workload allocation scheme. This approach improves the system throughput and reduces the completion time of the task. In addition to traditional scheduling parameters, we consider a human's rejection, contact delay, and social similarity.

Key words: Workload Allocation Scheme, Mobile Social Network

I. INTRODUCTION

Many resources in the world such as machines' computational ability and human intelligence remain idle. In order to make use of these idle resources, several crowdsourcing projects have been implemented, such as Boinc[1] and Amazon MTurk[2]. In these projects, a task owner uploads a work onto a server in advance, and individuals who are interested in the work can participate in it when they are free. The existing crowdsourcing platforms lack an advertising mechanism to invite the workers. It is hard to get workers for some task since most of the individuals focus on works that provide large payments. It is also difficult to inform the offline workers about the arrival of new tasks. Since the existing system is centralized and platform specific, it is not flexible. To overcome these constraints a distributed and self-organized crowdsourcing scheme, called open-crowdsourcing is introduced. In the open-crowdsourcing scheme, the existing workers themselves can invite others who are interested and capable of doing the particular work. Instead of selecting work by the client themselves, our system directly send works to the clients based on their potential. Existing crowdsourcing system can be extended using this allocation scheme.

II. EXISTING SYSTEM

Due to the lack of real time global information it is impossible for nodes to provide an optimal allocation of work segments in the current system that is being used. There arises the issue of unbalanced workload distribution of workload among the participating nodes. Because the node having less speed may be accumulated with heavy workload while those having high speed is given by low workload.

III. PROPOSED SYSTEM

The proposed system can provide an optimal allocation of work segments by completing large work within a short span of time. It makes full use of idle resources and hence the system throughput can be significantly improved.

IV. IMPLEMENTATION

The proposed system can be implemented using java with jdk version 1.6 as front end and My SQL as back end. The database is connected with the front end using JDBC.

Open-crowdsourcing system contains a group of connected clients monitored by a server. The task owner uploads a work at the server site. The server distributes the work among the clients in the network. At regular intervals client updates its progression of work. Once the work is completed the client returns the final result to the server which is deployed to the task owner.

Server performs functions such as uploading files, space allocation, verification, security. At the initial stage server distributes the work equally among all the clients. As the work progress the server monitors the system parameters such as computing speed, average accepting rate, average inter-contacting time etc. of each client nodes. Based on these parameters the work is re-allocated among the clients. Server provides the necessary resources for the clients. This scheme provides security by allocating distinct memory space to each and every participating clients such that the clients can access only the resources in their reserved space. Client performs functions as file transfer, sending messages, updating files etc. Client can choose any work among those which are allocated by server. At regular intervals client updates its progression at the server site. Client can communicate with other clients and server via text message. Also, client exchanges the system parameters with other clients. Based on these parameters client can share the work with other nodes.

A. Peer Status Based Finishing Time Estimation

Using this algorithm, each node can estimate the work's finishing time of neighbouring node using pair-wise contact. The estimation of finishing time may not be accurate since each node can share its work with others at any time and hence the recorded carrying workload of a neighbour node might be changed at the time of estimation. By this approach, the resulting completion times of the two virtual nodes are still equal to those even after the work is shared. Thereby each node can estimate the system parameters of neighbouring nodes. Within each node's one-hop neighbourhood, if workload adjustment happens during the scenario, that some neighbours have finished their workloads while others are still working, computing resources will be wasted. The approximate value of the wasted computing resources is taken and using this, a node can estimate his finishing time by first summing it up with

the real workloads, and then dividing the result by the summation of weighted speeds.

B. Computing Ability Based Workload Allocation

Using this algorithm, each node can attain a potential rating, which indicates a node's future computing ability, by recruiting other nodes. We assume that each node is associated with a local computing speed and a potential speed. The links in our model are associated with both accepting rate and random contacting delay. To resolve this problem, we design a special weight on each link, which can successfully integrate the number of friends, contacting delays, and accepting decisions together. When abilities of direct contactors are obtained, a pair of encountering nodes can reallocate their workload based on their potential abilities.

C. Adaptive Scheme

At initial stage, work is allocated to all the nodes in an uniform manner. We combine the advantage of both PSFT and Computing Ability based Workload Allocation to take decision on workload allocation. As the work progress, by analysing the result of above specified algorithms, we prioritize the nodes for further allocation of work. The reallocation is done in such a way that every node completes the work at same time. Thus this scheme avoids the unbalanced distribution of work.

V. RESULT AND DISCUSSION

Fig. 1: Client A Form

Fig.1 shows Client A form. Here a client can select his/her work. Once the work is done, client can update his/her work, which will be informed at server side also. A client can share his/her work with other clients if needed. There is also provision for a client to communicate with server as well as with other clients.

Fig. 2: Server Form

Fig.2 shows Server form. Server schedules the work to clients

Server can monitor each and every client. Server can communicate with clients using message option.

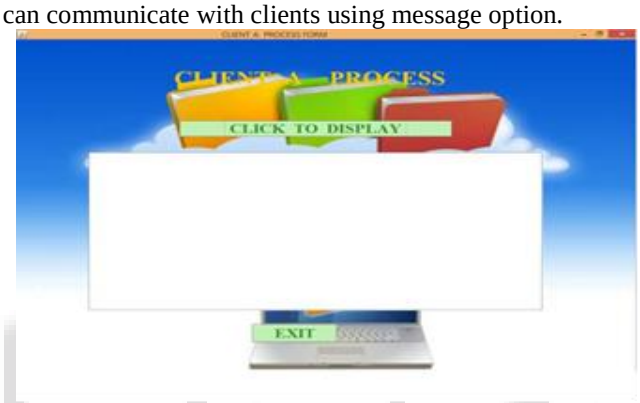

Fig. 3: Client A Process Form

Fig.3 shows Client a Process form. Server can visualize the performance of Client A using this form.

VI. CONCLUSION

Crowdsourcing is defined as the process of obtaining needed services, ideas, or content by soliciting contributions from a large group of people and especially from an online community, rather than from traditional employees or suppliers. This paper proposes new crowdsourcing system called Open-crowdsourcing. The proposed system is a self-organized and distributed system. The open crowdsourcing consider the problem of workload portioning and allocation among users. The proposed scheme is applicable to the system that contains multiple coexisting works. The developed adaptive workload allocation scheme results in optimal workload allocation and enhanced system throughput.

VII. FUTURE SCOPE

The developed system can be further enhanced by making it compatible to all types of networks and providing more security mechanisms.

REFERENCES

[1] (2014) [Online]. Available: boinc.berkeley.edu/, 2014.
[2] (2014)[Online].Available:www.mturk.com/mturk/,2014
.

