Using Facebook as a Cloud Platform for Solving Numerical Optimization Problem

T.M. Rezwanul Islam
CIT department
Islamic university of Technology

S.M. Al Mahi
CIT department
Islamic university of Technology

Abu Sina Mohammad Raju Chowdhury
CIT department
Islamic university of Technology

ABSTRACT

In case of numerical optimization problem, the size of data sets and computational complexity is rising enormously. These numerical optimization problem generally has lots of local minima and Evolutionary Algorithms helps quickly to find the optimal solution in this exigent search space. Not only that but also computing the objective functions becomes a very complicated and time consuming task. For this reason distributed computing system is very helpful. That's why Facebook is used as a cloud platform. Facebook users voluntarily support this scientific work of finding the solution for Numerical Optimization problems by sharing their CPU cycle.

General Terms:

Evolutionary Algorithms, Distributed Computing

Keywords:

Numerical Optimization, Cloud, Sketching, Evolutionary Algorithms, Distributed Computingifx

1. INTRODUCTION

In mathematics and computer science, an optimization problem is the problem of finding the best solution from all feasible solutions. More formally, an optimization problem A is a quadruple (I, f, m, g), where,

—I is a set of instances.

—given an instance x $\in$ I , f(x) is the set of feasible solutions.

—given an instance x and a feasible solution y of, m(x,y) denotes the measure of y, which is usually a positive real.

—g is the goal function, and is either min or max.

The goal is then to find for some instance x an optimal solution, that is, a feasible solution y with,

$$m(x, y) = g\{m(x, y\prime)|y\prime \in f(x)\}$$

The high computational cost of complex engineering optimization problems has motivated the development of parallel optimization algorithms. As using a super computer appears very costly. So it was necessary to use volunteer grid. In volunteer grid mechanism, people help by sharing their CPU cycle time.
Now a days, Social networking has become an everyday part of many peoples lives and some social networking site has huge users. As for example Facebook has more than 750 million active users. Social networking has also extended beyond communication between friends, for instance, there are a large number of integrated applications and some organizations even use Facebook credentials. The structure of a social Network is basically a dynamic virtual organization with intrinsic trust relationships between friends. We used our friends idle CPU power to solve our optimization problems. Cloud environments typically provide low level abstractions of computation or storage. Computation and Storage clouds are often used to extend the capabilities of storage-limited devices such as phones and desktops, and provide transparent access to data from anywhere. There are a large number of commercial Cloud providers such as Amazon EC2/S3, Google App Engine, Microsoft Azure and also many smaller scale open clouds like Nimbus and Eucalyptus. These clouds provide access to scalable virtualized resources contributed by users are dynamically provisioned amongst a group of friends[9].
In this paper, we show the results and analysis for numerical optimization problems run in normal environment and in parallel environment by optimizing with the help of Niched Genetic Algorithm[10] and Differential Evolution. In the following sections we discuss about some related works (in section ii), Facebook applications (in section iii), Facebook as cloud platform (in section iv), optimization methods (in section iv), simulated environments (in section vi), results (in section vii).

2. RELATED WORKS

There are numerous example of social network and cloud computing association. Most applications use cloud platforms to host social networks. But our application uses the enormous users in the social networks who are willing to share their CPU and optimizes scientific problems. As instance, Facebook users can built cloud based applications hosted by Amazon Web Services.
There are similar efforts in the Grid community to leverage Social networking concepts, communities, and mechanisms. PolarGrid is one such example which collects Social data using theOpenSocial interface and relies on OpenID for identification. Different social networking functions are then in corporated and an application specific portal is developed.
Volunteer computing is a distributed computing model. In which users donate computing resources to a specific (academic) project. The first volunteer project was the Great Internet Mersenne Prime Search (www.mersenne.org). In 1996, however the term gained much exposure through the SETI@Home[11] projects in the late 90?s.These projects showed the massive computing power available through collaborative systems. One of the most relevant Volunteer computing efforts is Storage@Home[12]. which is used to backup and share huge data sets arising from scientific research. The focus of Volunteer computing has since shifted towards generic middle ware providing a distributed infrastructure independent of the type of computation, for example the Berkeley Open Infrastructure for Network Computing(BOINC)[13]. Most Volunteer platforms do not define SLAs, typically users are anonymous and are not accountable for their actions(they are rewarded with different incentives however). In a Social Cloud context this does not suffice as users need to have some level of accountability. A more realistic model for this type of open sharing is a credit based system in which users earn credits by contributing resources and then spend these credits when using other resources. This type of policy issued in systems such as PlanetLab[9].

3. FACEBOOK APPLICATIONS

Facebook launched the Facebook Platform on May 24, 2007, providing a framework for software developers to create applications that interact with core Facebook features[14]. A markup language called Facebook Markup Language was introduced simultaneously; it is used to customize the ”look and feel” of applications that developers create. Facebook JavaScript (FBJS) is parsed when a page is loaded to create a virtual application.
Usually Facebook applications are hosted independently. User retrieves the Facebook canvas url and the user can map this canvas url to a specific url which is hosted remotely. When a page is requested by the user through the Facebook Canvas URL (http://apps.facebook.com/numerical-opt/) the Facebook server forwards the request to the defined callbackURL. The application creates a page based on the request and returns it to Facebook. At this point the page is parsed and Face-book specific content is added according to the FBML page instructions. The final page is then returned to the user[9].

4. GLOBAL OPTIMIZATION METHOD

Evolutionary optimization is probably the most active and productive area in Evolutionary computation measured by the number of papers published and the number of successful applications reported. Although neither evolutionary programming nor genetic algorithms were first proposed as optimization algorithms, people had quickly realized they could adapt these algorithms to carry out combinatorial and function optimization algorithms, people had quickly realized they could adapt these algorithms to carry out combinatorial and function optimization[1]. An EA uses some mechanisms encouraged by biological evolution: reproduction, mutation, recombination, and selection. Candidate solution to the optimization problem play the role of individuals in a population, and the fitness function determines the environment within which the solutions ’live’. Evolution of the population then takes place after the repeated application of the above operators.

4.1 Evolution Program

The evolution program is a probabilistic algorithm which maintains a population of individuals, $P(t) = \{x^t{}_1, ...x^t{}_n\}$ for iteration t. Each individual represents a potential solution to the problem at hand, and, in any evolution program, is implemented as some data structure S. Each solution is evaluated to give some measure of its ’fitness’. Then a new population (iteration t + 1) is formed by selecting the more fit individuals. Some members of the new population undergo transformations by means of genetic operators to form solutions. There are many transformations (mutation type), which create new individuals by small change in a single individual ($m_i : S \rightarrow S$), and higher order transformations (crossover type), which create new individuals by combining parts from several (two or more) individuals ($c_j : S....S \rightarrow S$). After some number of generations the program converges-it is hoped that the best individual represents a near-optimum (reasonable) solution[2].

4.2 Differential Evolution

Differential Evolution is a Stochastic Direct Search and Global Optimization algorithm, and is an instance of an Evolutionary Algorithm from the field of Evolutionary Computation. It is related to sibling Evolutionary Algorithms such as the Genetic Algorithm, Evolutionary Programming, and Evolution Strategies, and has some similarities with Particle Swarm Optimization.
The Differential Evolution algorithm involves maintaining a population of candidate solutions subjected to iterations of recombination, evaluation, and selection. The recombination approach involves the creation of new candidate solution components based on the weighted difference between two randomly selected population members added to a third population member. This perturbs population members relative to the spread of the broader population. In conjunction with selection, the perturbation effect self-organizes the sampling of the problem space, bounding it to known areas of interest.
Differential Evolution has a specialized nomenclature that describes the adopted configuration. This takes the form of DE/x/y/z, where x represents the solution to be perturbed (such a random or best). The y signifies the number of difference vectors used in the perturbation of x, where a difference vectors is the difference between two randomly selected although distinct members of the population. Finally, z signifies the recombination operator performed such as bin for binomial and exp for exponential[2].

5. DISTRIBUTED EVOLUTIONARY ALGORITHMS

There have been many different approaches for making EAs work on different distributed computing systems. Here, popular sequential method has followed.

5.1 Sequential Approach

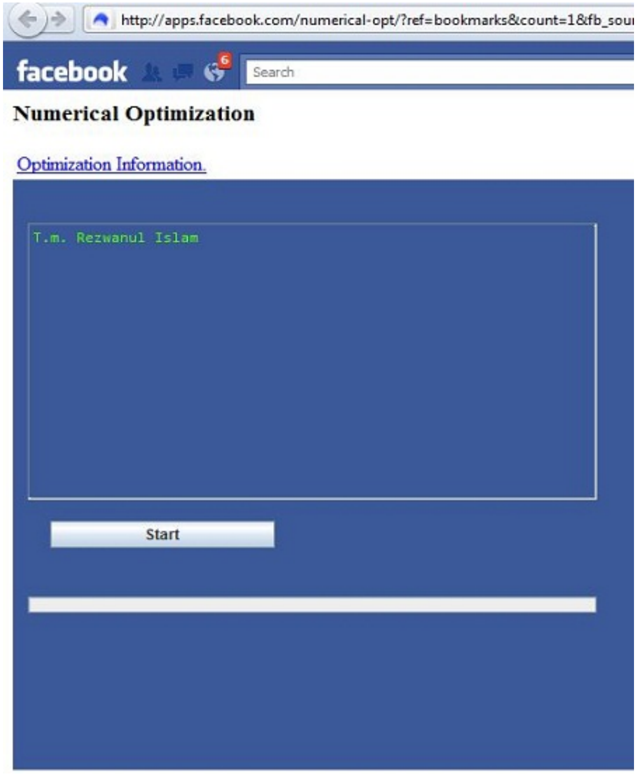
Distributed EAs typically use a single population strategy for generating new populations by evaluating each individual in parallel; repeating this process until the population has reached convergence criteria. It is possible to increase scalability past the population size by additionally evaluating the objective function in parallel. This type of approach is best suited to highly reliable and homogeneous computing nodes as found in clusters and supercomputers[8].

5.2 Asynchronous Approach

Single-population and a master-worker model is used in asynchronous distributed EAs. Here the generated population is kept in the server and workers evaluate individuals, using a work-stealing model. Asynchronous optimization generates new individuals based on a populations current state in response to requests for work and later inserts the results to the population when and if they are reported. As the heuristics for generating new individuals in the discussed EAs are randomized, this allows asynchronous evolutionary algorithms to generate as many unique individuals are required to satisfy all potential workers. There are also no dependencies between generated individuals, so if a worker fails and does not report the fitness of its individual, the search does not need to wait for that individual to be recalculated[8].

6. SIMULATED ENVIRONMENT: FACEBOOK AS A CLOUD PLATFORM

We have simulated an environment which is massively distributed. Here the computation power comes from Facebook use who are using our application. Users are contributing by running a java Applet . This java applet looks like (fig : 1).
In fact, there is a central population pool where the distributed computation result is stored, merged and redistributed to the current connected users. Distribution is done based on the Roulette Wheel Selection mechanism. User machine helps by performing EAs on the sub-population and improving the . Improved population is sent back to the population pool for merging. This process is continued until some convergence criteria is met. The process can be depicted as (fig : 2).


http://apps.facebook.com/numerical-opt/?ref=bookmarks&count=1&fb_sou
facebook
6
Search
Numerical Optimization
Optimization Information
Start

T.m. Rezwanul Islam

Fig. 1. Java Applet

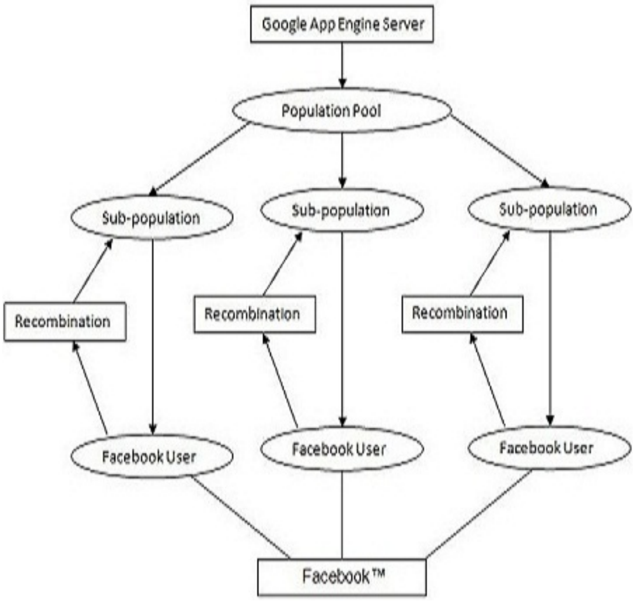


Fig. 2. Simulation process

7. RESULTS

For genetic algorithm, a mutation rate of .1 and crossover rate .8 was used. In Differential Evolution scale factor was randomly chosen between 0≈1 and recombination rate was .8. The Ackley, Griewank, Rastrigin and Schwefel problems were used as benchmark optimization problems which are described in other works[8].

In the following figures (fig : 3,4,5,6) we tried to show that optimization of different numerical functions. The dots in the figure

Name	Min(local)	Min(Cloud)	Cost Evaluation
Ackley	0.003801	0.0	38400
Rastrigin	0.0	0.0	48000
Griewank	0.011698	0.0	78000
Schwefel	418.982	0.0	NA

Name	Convergence(local)	Convergence(cloud)
Ackley	NA	Y
Rastrigin	Y	Y
Griewank	NA	Y
Schwefel	NA	Y

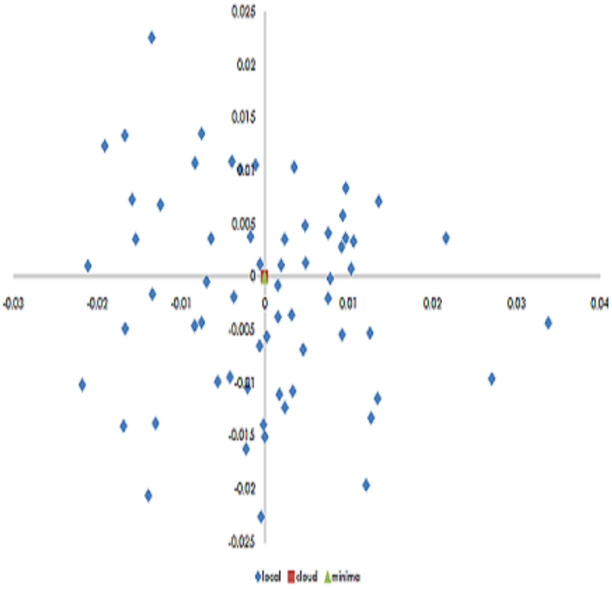


Fig. 3. Ackley function

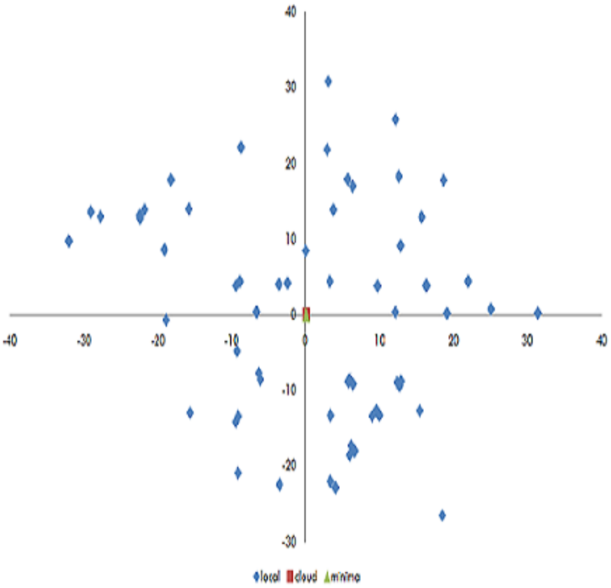


Fig. 4. Griewank function

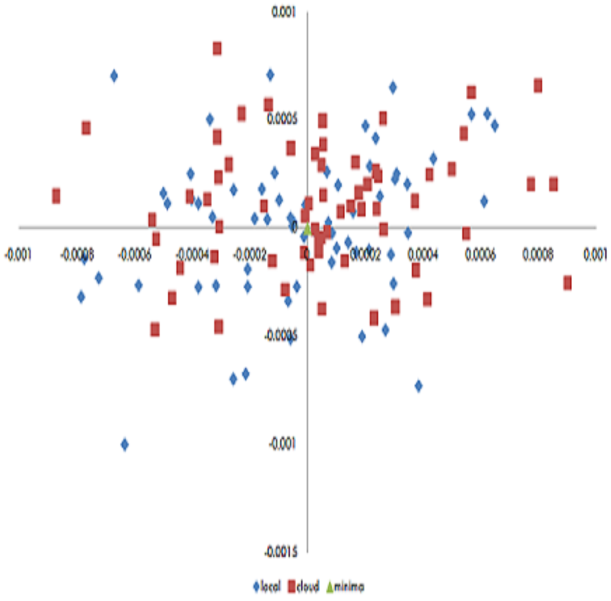


Fig. 5. Rastrigin function

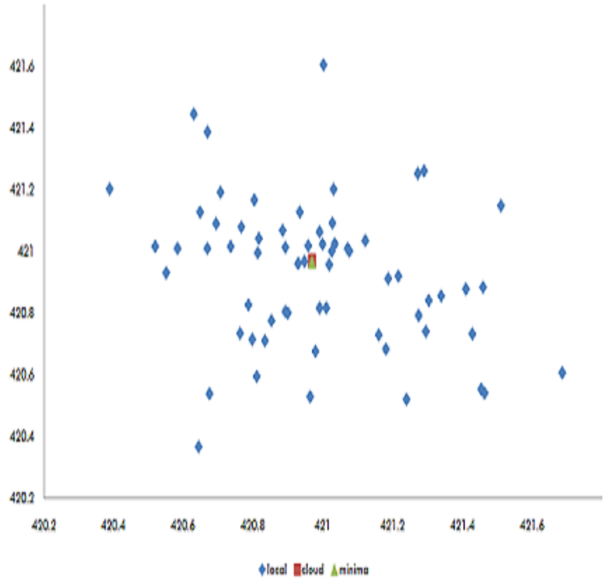


Fig. 6. Schwefel function

indicates the individuals after a certain number of evaluation of the function. Blue, red, magenta are for individuals resulted from the local machine, individuals resulted from Facebook application and global optimum respectively. We can see that Facebook application individuals are closer to global optima. As we expected our result is giving more way of solving numerical optimization problem.

8. CONCLUSION

This work presents a simulation framework that depicts how performance is better in cloud network than in comparison with local machine run . Almost for every case, the local machine and cloud provides the accurate result but in case of cloud platform the performance is tuned greatly. Though in this paper we have used only four benchmark functions but we are looking forward to increase the number of functions for our simulation in future. We also looking forward to test the functions in broader spectrum as well in supercomputer but due to resource limitation it is impossible right now.

9. ACKNOWLEDGEMENT

We wish to acknowledge our profound indebtedness to Prof. Dr. M.A. Mottalib (Head, Department of Computer Science and Information Technology, IUT) and AKM Khaled Ahsan Talukder (Assistant Professor, Department of Computer Science and Information Technology). Their deep knowledge in the field of Numerical Optimization and Distribued computing influenced us to carry out this project up to this point. We wish to express our gratitude to the Islamic University of Technology (IUT) for providing us the environment and support.

10. REFERENCES

[1] R. Sarker, M. Mohammadian and X. Yao (eds.),”Evolutionary Optimization”, *INTERNATIONAL SERIES IN OPERATIONS RESEARCH AND MANAGEMENT SCIENCE* Volume: 48, Kluwer Academic, ISBN 0-7923-7654-4, January 2002, 432 pp.

[2] Zbigniew Michalewicz ”Genetic Algorithms + Data Structures = Evolution Programs” *3rd rev and extended edition, Springer-Verlag, Berlin, 1996Tavel*, P. 2007 Modeling and Simulation Design. AK Peters Ltd.

[3] K. Price and R. Storn, ”Differential Evolution: Numerical Optimization Made Easy”, *Dr. Dobb’s Journal*, 1997 Forman, G. 2003.

[4] K. V. Price, ”An introduction to differential evolution”, in *New Ideas in Optimization*, pages 79 - 108, McGraw-Hill Ltd., UK, 1999.

[5] K. V. Price and R. M. Storn and J. A. Lampinen, ”Differential evolution: A practical approach to global optimization”, *Springer*, 2005. ISBN 978-3-540-20950-8.

[6] R. Storn and K. Price, ”Differential Evolution: A Simple and Efficient Adaptive Scheme for Global Optimization over Continuous Spaces”, *Technical Report TR-95-012, International Computer Science Institute*, Berkeley, CA, 1995.

[7] R. Storn and K. Price, ”Differential evolution: A simple and efficient heuristic for global optimization over continuous spaces”, *Journal of Global Optimization*, 1997.

[8] Travis Desell. ”Asynchronous Global Optimization for Massive-Scale Computing” PhD thesis, *Rensselaer Polytechnic Institute*, December 2009.

[9] Kyle Chard, Simon Caton, Omer Rana, Kris Bubendorfer. ”Social Cloud : Cloud Computing in Social Networks”, *2010 IEEE 3rd International Conference on Cloud Computing*.

[10] Ole J.Mengshoel, David E. Goldberg. ”The Crowding approach to Niching in Genetic Algorithm”, *200X by the Massachusetts Institute of Technology*.

[11] DanWerthimer, JeffCobb, MattLebofsky, DavidAnderson, and EricKorpela. ”Seti@home?massively distributed computing for seti”. *Com-puting in Science and Engineering*,3(1):78 - 83,2001.

[12] A.L.Beberg and V.S.Pande. ”Storage@home: Petascale distributed storage”.*In Parallel and Distributed Processing Symposium,2007*. IPDPS2007. IEEE International, pages 1 - 6,2007.

[13] David P. Anderson .”Boinc : A system for public-resource computing and storage”, *In GRID 04: Proceedings of the 5th IEEE / ACM International Workshop on Grid Computing*,pages 410 ,Washington, DC, USA, 2004 .IEEE Computer Society.

[14] Kirkpatrick, David (2007-05-29). ”Facebook’s plan to hook up the world”. *CNN. Retrieved* 2008-03-05