

Low or No Cost Distributed Evolutionary Computation

Juan Julián Merelo

Abstract. From the era of big science we have back to the "do it yourself" era of science, where you don't have any money to buy clusters and subscribe to grids but still have algorithms that crave many computing nodes and need them for scalability. Fortunately, this coincides with the era of big data, cloud computing, and browsers including JavaScript virtual machines. This talk will concentrate on two different aspects of volunteer or freeriding computing: first, the pragmatic: where to find those resources, which can be used, what kind of support you have to give them; and then, the theoretical: how algorithms can be adapted to a environment in which nodes come and go, have different computing capabilities and operate in complete asynchrony of each other.

1 What Is the Point of Low or No Cost Evolutionary Algorithms

The world has computational resources in spades. Most of them do not belong to you or your lab. That does not mean you cannot use it. The problem is how.

Most theory in parallel computing has been devoted to predict and optimize the performance where the number of nodes, their connections, and the time every one is devoting to the computation is known in advance. However, even if Big Science is not over, the era of Citizen science has started (with SETI@home and then BOINC) and it offers a vast amount of computational resources to attract, if only you know how. But there is a challenge: knowing, or at least having a ballpark, of how your algorithm is going to perform in this uncertain environment, where none of the factors is known: neither the number of nodes, through how they are connected, to how long are they going to be focused on doing what you want them to.

Juan Julián Merelo

Dept. of Computer Architecture and Technology and CITIC-UGR, University of Granada, Spain
e-mail: jmerelo@geneura.ugr.es

Besides, since Amazon started selling EC2 several years ago, reliable and scalable computing resources are also available for a low price and on demand. Recently, Google has also refurbished its offering lowering their prices. This means that the conjunction of free or low-cost cloud computing engines, volunteer computing systems and the untapped capability of desktop systems can be used for creating massive, or at least potentially massive, distributed computing experiments.

2 Conclusion

In this talk we will offer our experience on using browser-based computing since 1999 [1] and other emerging paradigms, such as peer to peer based computing [2], mainly using evolutionary algorithms.

There are many challenges involved in using these resources: from adapting current algorithms so that they match this environment [3] to check which EA configuration works the best in it, or creating a framework that can use it easily [4]. But the main challenge is that asking people to contribute resources implies that you are opening your science to society and you have to give something in return: you have to adopt a set of best practices that have come to be known as Open Science, because “Give, and it shall be given unto you”, you will get as much back from society as you give to it opening your science and explaining it to the public. This, among other things, means that popularity will become directly performance of the metacomputer you create by attracting more users.

References

1. González, J., Merelo-Guervós, J.-J., Castillo, P.A., Rivas, V., Romero, G., Prieto, A.: Optimized web newspaper layout using simulated annealing. In: Mira, J., Sánchez-Andrés, J.V. (eds.) IWANN 1999. LNCS, vol. 1607, pp. 759–768. Springer, Heidelberg (1999)
2. Laredo, J.L.J., Eiben, A.E., van Steen, M., Guervós, J.J.M.: EvAg: a scalable peer-to-peer evolutionary algorithm. *Genetic Programming and Evolvable Machines* 11(2), 227–246 (2010)
3. Merelo, J.J., Castillo, P.A., Laredo, J.L.J., Mora, A., Prieto, A.: Asynchronous distributed genetic algorithms with JavaScript and JSON. In: WCCI 2008 Proceedings, pp. 1372–1379. IEEE Press (2008)
4. Guervós, J.J.M.: NodEO, a evolutionary algorithm library in Node. Technical report, GeNeura group (March 2014), <http://figshare.com/articles/nodeo/972892>