Brief Announcement: Algorithmic Mechanisms for Internet-Based Computing under Unreliable Communication*

Evgenia Christoforou[1], Antonio Fernández Anta[2,3], Chryssis Georgiou[1], and Miguel A. Mosteiro[4,3]

[1] University of Cyprus, Cyprus
{cs05ec1,chryssis}@cs.ucy.ac.cy
[2] Institute IMDEA Networks, Spain
antonio.fernandez@imdea.org
[3] Universidad Rey Juan Carlos, Spain
[4] Rutgers University, USA
mosteiro@cs.rutgers.edu

Abstract. This work, using a game-theoretic approach, considers Internet-based computations, where a master processor assigns, over the Internet, a computational task to a set of untrusted worker processors, and collects their responses. In particular, we consider a framework where altruistic, malicious, and rational workers co-exist, the communication between the master and the workers is not reliable, and that workers could be unavailable. Within this framework, we design algorithmic mechanisms that provide appropriate incentives to rational workers to act correctly, despite the malicious' workers actions and the unreliability of the network.

1 Motivation and Prior Work

In [1], an Internet-based master-worker framework was considered where a master processor assigns, over the Internet, a computational task to a set of untrusted worker processors and collects their responses. Three type of workers were assumed: *altruistic, malicious,* and *rational.* Altruistic workers always compute and return the correct result of the task, malicious workers always return an incorrect result, and rational (selfish) workers act based on their self interest. In other words, the altruistic and malicious workers have a predefined behavior: the first are *honest* and the latter are *cheaters* (they do not care about their utilities). Rational workers decide to be honest or to cheat based on which strategy would increase their utility. Under this framework, a game-theoretic mechanism

* This work is supported in part by the Cyprus Research Promotion Foundation grant ΤΠΕ/ΠΛΗΡΟ/0609(ΒΕ)/05, Comunidad de Madrid grant S2009TIC-1692, Spanish MICINN grant TIN2008–06735-C02-01, and the National Science Foundation grant CCF-0937829.

D. Peleg (Ed.): DISC 2011, LNCS 6950, pp. 147–149, 2011.

was designed that provided necessary incentives to the rational workers to compute and report the correct task result despite the malicious workers' actions. The design objective of the mechanism is for the master to force a desired *Nash Equilibrium* (NE), i.e., a strategy choice by each rational worker such that none of them has incentive to change it. That NE is the one in which the master achieves a desired probability of obtaining the correct task result, while maximizing its benefit. The utility of the mechanism was demonstrated by applying it to two paradigmatic applications: a SETI-like volunteer computing system and a contractor-based system, such as Amazon's mechanical turk. This work has not considered the possibility of network unreliability, which is a factor that cannot be ignored in Internet-based computations [2].

2 Contributions

This work extends the master-worker framework of [1] by additionally considering the possibility that the communication between the master and the workers is not reliable. That is, we consider the possibility that messages exchanged may get lost or arrive late. This communication uncertainty can either be due to communication-related failures or due to workers being slow in processing messages (or even crashing while doing so). For instance, Heien at al. [2] have found that in BOINC only around 5% of the workers are available more than 80% of the time, and that half of the workers are available less than 40% of the time. This fact, combined with the length of the computation incurred by a task [3], justifies the interest of considering in the Internet-based master-worker framework the possibility of workers not replying. In order to introduce this possibility in the framework, we consider that there is some positive probability that the master does not receive a reply from a given worker. Since it is now possible for a worker's reply not to reach the master, we additionally extend the framework of [1] by allowing workers to abstain from the computation. Imagine the situation where a rational worker decides to compute and truthfully return the task result but its reply is not received by the master. In this case the master provides no reward to the worker, while the worker has incurred the cost of performing the task. Hence, it is only natural to provide to the workers the choice of not replying, especially when the reliability of the network is low. This makes the task of the master even more challenging, as it needs to provide the necessary incentives to encourage rational workers to reply and do so truthfully, even in the presence of low network reliability.

Within this extended framework, we develop and analyze two game-theoretic mechanisms, a time-based mechanism and a reply-based one, that provide the necessary incentives for the rational workers to truthfully compute and return the task result, despite the malicious workers' actions and the network unreliability. Furthermore, we apply our mechanisms to two realistic settings: SETI-like volunteer computing applications and contractor-based applications such as Amazon's mechanical turk. *Full details can be found in [4].*

References

1. Fernández Anta, A., Georgiou, C., Mosteiro, M.A.: Algorithmic Mechanisms for Internet-based Master-Worker Computing with Untrusted and Selfish Workers. In: Proc. of IPDPS 2010, pp. 378–388 (2010)
2. Heien, E.M., Anderson, D.P., Hagihara, K.: Computing low latency batches with unreliable workers in volunteer computing environments. Journal of Grid Computing 7, 501–518 (2009)
3. Kondo, D., Araujo, F., Malecot, P., Domingues, P., Silva, L., Fedak, G., Cappello, F.: Characterizing result errors in internet desktop grids. In: Kermarrec, A.-M., Bougé, L., Priol, T. (eds.) Euro-Par 2007. LNCS, vol. 4641, pp. 361–371. Springer, Heidelberg (2007)
4. Technical Report of this work, `http://www.cs.ucy.ac.cy/ric/MARfailTR.pdf`