How to Leverage Altruism (position paper)

Jean-Philippe Martin, Lorenzo Alvisi, et al. (TBD)

April 16, 2007

Abstract

When there is no central administrator to control the actions of nodes in a distributed system, the users may deviate for personal gain. The BAR model describes the three types of nodes in these environments: Byzantine nodes deviate arbitrarily, Altruistic nodes follow the protocol, and Rational nodes deviate for gain.

Previous BAR work used an equilibrium concept where both Altruistic and Rational nodes follow the protocol. These methods do not take advantage of the Altruistic nodes. In this paper we introduce a new equilibrium concept that allows these nodes to behave differently: we can then leverage Altruistic nodes when they are present.

1 Introduction

Fault-tolerance approaches that were appropriate for clusters of computers are not appropriate for *cooperative services*,[1] peer-to-peer systems where there is no central administrator controlling all of the nodes in the system. Freed from the oversight of a central administrator, the humans (or *users*) can interfere with the configuration of their computers or even replace the software in order to maximize their benefit or minimize their costs. This cause of deviation is not a concern for clusters of computers because, there, the person controlling the computers wants them to run the software without deviation.

[1] Also known as MAD, for "multiple administrative domains".

Cooperative services should not be modeled using the Byzantine model [6] because many problems of interest cannot be solved when all nodes are Byzantine (such as consensus in eventual synchrony [4] or with a fair scheduler [2]), yet in cooperative services it is conceivable that *every* node will deviate from the protocol because of the actions of the rational users that are controlling them. Cooperative services should not be described only in terms of rational utility-maximizing nodes either [5, 8, 10, 12], because although this approach handles rational behavior it is brittle in the face of Byzantine failures.

The BAR model [1, 3, 9] is best adapted to cooperative services. Its name comes from the three types of nodes that it models: Byzantine, Altruistic, and Rational. Byzantine nodes may deviate arbitrarily from the protocol. Altruistic nodes follow the protocol, and Rational nodes only deviate from the protocol if that increases the utility they receive.

Introducing Rational nodes may reduce performance, as Rational nodes may not be willing to take part in computation steps that contribute to system performance if the utility that they get from the computation is less than the cost itself. This "price of anarchy" has been measured for the virus inoculation game [11]. Existing protocols for the BAR model (e.g. [9, 7]) can tolerate both Byzantine and Rational behavior, but they force Altruistic and Rational nodes to behave identically, thus paying the full price for anarchy even when Altruistic nodes are present.

In this paper we show how to design protocols that leverage altruistic behavior, if present.

These protocols specify computation steps that Altruistic nodes will follow and Rational nodes may omit. As a result, protocol performance can be improved when nodes behave altruistically, even if the identity of the Altruistic nodes is not necessarily known.

2 BAR-Tolerant Protocols

In our setup, each node i is given a protocol σ_i for consideration. We call σ_i node i's *suggested protocol*; altruistic nodes will follow that protocol but we do not know a-priori what protocol the rational and Byzantine nodes will decide to follow. The protocol is repeated forever. Given a desired property P, our goal is to find a protocol σ that satisfies P when σ is given as the suggested protocol to each node in a cooperative service (or, more generally, to find a protocol profile $\vec{\sigma}$ with the same property; a protocol profile may assign a different protocol to each node).

The first step is to specify the beliefs of the rational nodes. In this paper we assume that all rational nodes believe that the other nodes may be altruistic, rational or Byzantine (but do not initially know which is which), and that there are at most f Byzantine nodes. We consider rational nodes that do do not collude.

The second step is to specify what value the rational nodes assign to events and possible future events. Events are valued using an utility function u that assigns a numerical score to an *outcome* of the protocol as seen by that particular node; the outcome is an execution trace, conceptually a history of the states that node was in. Outcomes with higher utility are preferred. Different nodes may have a different utility function, so we specify the set U of all utility functions that Rational nodes may have. For example some nodes may try to minimize their bandwidth costs while some other nodes may only care about their CPU costs. The value of possible future events is based on the utility function but it must also take into account the nodes' optimism or pessimism when comparing several possible future scenarios. We call this the *estimated utility function*. In our case we consider *risk-averse* nodes that consider the worst that may happen to them.

These first two steps are common with previous methods for writing BAR protocols [1, 3, 9]. The next two steps differ in in way that allow us to leverage Altruistic nodes.

The third step is to pick a protocol profile $\vec{\sigma}$ and a rational envelope $\vec{\Upsilon}$. σ_i is the suggested protocol for node i, and Υ_i is a set of protocols ($\sigma_i \in \Upsilon_i$). The intention is that if i is Altruistic it follows σ_i and if it is Rational then it follows some protocol in Υ_i. The pair $(\vec{\sigma}, \vec{\Upsilon})$ must satisfy the *BAR Equilibrium condition* that indicates that nodes perceive no benefit from deviating outside of the envelope (as opposed to inside), regardless of what they may have learned about other nodes being Byzantine (K_j).

$$\forall j, \forall \hat{u}_j \in \hat{U}, \forall K_j \in \mathcal{K}, \forall \sigma'_j \in \Sigma, \exists \sigma''_j \in \Upsilon_j : \\ \hat{u}_j(\vec{\sigma} \ominus \sigma'_j, K_j) \leq \hat{u}_j(\vec{\sigma} \ominus \sigma''_j, K_j)$$

The final step is to check whether the desired property P holds despite deviations of the Rational nodes within the rational envelope. If this is the case then we can say that the protocol is BAR-Tolerant (BART). Unlike IC-BFT protocols, BART protocols can leverage altruism since $\vec{\sigma}$ can include actions that only Altruistic nodes would follow.

3 Conclusion

The BAR model describes cooperative services. Previous work shows how to built protocols called IC-BFT that can tolerate Byzantine and Rational failures. We have shown how to build BART protocols that can achieve higher performance by leveraging Altruistic nodes, if present. If there is no Altruistic node then BART protocols degrade gracefully to an IC-BFT protocol.

References

[1] A. S. Aiyer, L. Alvisi, A. Clement, M. Dahlin, J.-P. Martin, and C. Porth. BAR fault tolerance for cooperative services. In *Proc. 20th SOSP*, pages 45–58, Oct. 2005.

[2] G. Bracha and S. Toueg. Asynchronous consensus and broadcast protocols. *J. ACM*, 32(4):824–840, 1985.

[3] A. Clement, J. Napper, H. C. Li, J.-P. Martin, L. Alvisi, and M. Dahlin. Theory of BAR games. In submission to PODC, 2007.

[4] C. Dwork, N. Lynch, and L. Stockmeyer. Consensus in the presence of partial synchrony. *J. ACM*, 35(2):288–323, 1988.

[5] G. Hardin. The tragedy of the commons. *Science*, 162:1243–1248, 1968.

[6] L. Lamport, R. Shostak, and M. Pease. The Byzantine generals problem. *ACM Trans. Program. Lang. Syst.*, 4(3):382–401, 1982.

[7] H. C. Li, A. Clement, E. Wong, J. Napper, I. Roy, L. Alvisi, and M. Dahlin. BAR gossip. In *Proc. 7th OSDI*, pages 191–204, Nov. 2006.

[8] W. Lloyd. *Two Lectures on the Checks to Population*. Oxford University Press, 1833.

[9] J.-P. Martin. *Byzantine Fault-Tolerance and Beyond*. PhD thesis, The University of Texas at Austin, Dec. 2006.

[10] A. Mas-Colell, M. D. Whinston, and J. R. Green. *Microeconomic Theory*. Oxford University Press, 1995.

[11] T. Moscibroda, S. Schmid, and R. Wattenhofer. When selfish meets evil: Byzantine players in a virus inoculation game. In *Proc. 25th PODC*, pages 35–44, July 2006.

[12] D. C. Parkes. *Iterative Combinatorial Auctions: Achieving Economic and Computational Efficiency*. PhD thesis, Department of Computer and Information Science, University of Pennsylvania, May 2001.