Brief Announcement: Accurate Byzantine Agreement with Feedback

Vijay K. Garg,[*] John Bridgman and Bharath Balasubramanian
Department of Electrical and Computer Engineering
The University of Texas at Austin
Austin, TX 78712-1084, USA
garg@ece.utexas.edu, johnfb@mail.utexas.edu, balasubr@ece.utexas.edu

ABSTRACT

The Byzantine Agreement (BA) problem requires non-faulty processes to agree on a common value. In many applications, it is important that the processes agree on the *correct* value. In this paper, we present a problem called Accurate Byzantine Agreement with Feedback (ABAF) in which all processes receive common feedback from the environment indicating if the value they agreed upon was correct or not (accuracy). We present an algorithm that solves the ABAF problem based on a standard solution to the BA problem and a multiplicative method to maintain and update process weights indicative of how often they are correct. We make guarantees on the accuracy of the algorithm based on assumptions on the accuracy of the processes and the proportion of faulty and non-faulty processes in the system. For each iteration, if the weight of accurate processes is at least $3/4^{th}$ the weight of the non-faulty processes, the algorithm always decides on the correct value. When the non-faulty processes are accurate with probability greater than $1/2$, the algorithm decides on the correct value with very high probability after some initial number of mistakes. In fact, among n processes, if there exists even *one* process which is accurate for all iterations, the algorithm is wrong only $O(\log n)$ times for any large number of iterations of the algorithm.

Categories and Subject Descriptors

H.3.4 [**Systems and Software**]: Distributed Systems

General Terms

Algorithms, Reliability, Security

Keywords

Byzantine Agreement, Multiplicative Weight Update

1. INTRODUCTION

In real-world applications, processes in a distributed system may be compromised, leading to malicious or unpredictable behavior. The Byzantine Agreement (BA) problem [9, 7, 6, 3] requires all non-faulty processes to agree on a common value given that some of the processes may show arbitrary faulty or Byzantine behavior. In many applications, it is better for the system to agree on the correct value among the two binary values as specified by environmental feedback. We refer to this version of the BA problem as Accurate Byzantine Agreement with Feedback (ABAF). For example, suppose in a distributed control system a coordinated action needs to be taken (such as opening or closing a valve) depending upon the observations made by possibly faulty distributed processes. Depending upon the outcome of the action, the environment can provide feedback on whether the action taken was correct or not. In this paper, we give an algorithm, referred to as the ABAF algorithm that incorporates the external feedback for subsequent iterations of the algorithm. Our algorithm is based on two key components: a standard solution to the BA problem and a multiplicative weight update method. The concept of weighted majority and multiplicative weight update is used in many disciplines such as learning theory, game theory and linear programming [5, 8]. Typically, there are a set of experts and based on their views or predictions, a binary value needs to be chosen (such as the decision to buy or sell stocks in a stock market). Weights are assigned over some common distribution to these experts and a value is chosen according to the weighted majority. To improve predictions over time, the weight of each wrong expert is decreased by some constant proportion of its previous weight after every iteration. In this paper, we assume the presence of malicious Byzantine experts and apply this method to BA. We summarize our contributions in the following sections. The complete version of this paper can be found in [4].

2. ABAF PROBLEM

We consider a distributed system of processes with a completely connected topology. We assume a reliable, FIFO communication system in which there is a strict upper bound on message delivery (synchronous system). The processes may undergo Byzantine failures, i.e., fail in an arbitrary fashion; in particular, they may lie and collude with other failed processes to foil any algorithm. In the standard BA problem, all non-faulty processes must agree on a common value. The only requirement on the decided value is that it must be proposed by a non-faulty process (validity). We define the ABAF problem by replacing validity with the notion of accuracy.

Definition 1. (Accurate Byzantine Agreement with Feedback) Assume n processes, among which at most f Byzan-

[*] This research was supported in part by the NSF Grants CNS-0718990, CNS-0509024, Texas Education Board Grant 781, SRC Grant 2006-TJ-1426, and Cullen Trust for Higher Education Endowed Professorship.

PODC'11, June 6–8, 2011, San Jose, California, USA.
ACM 978-1-4503-0719-2/11/06.

tine faults can occur such that $n \geq 3f + 1$. Each of the non-faulty processes propose either 0 or 1. An algorithm that solves the Accurate Byzantine Agreement with Feedback (ABAF) problem, must satisfy these properties:

- Agreement: All non-faulty processes decide on the same value.
- Accuracy: All non-faulty processes decide on a value that is deemed correct by the environmental feedback.
- Termination: The algorithm terminates in a finite number of rounds.

3. ABAF ALGORITHM

We propose an algorithm for the ABAF problem based on maintaining a common weight vector at all processes and updating this vector based on the feedback for each iteration. Initially, the weight of each process is a non-negative value proportional to the trust of the system on that process. If there is no prior information available, then the weights can simply be initialized to $1/n$. The algorithm has four steps. In step 1, the processes propose a value and exchange this value with each other to populate a vector V of all inputs. In step 2, a standard BA algorithm [1, 2] is used to ensure that each non-faulty process agrees on all the values in V. In step 3, processes determine the sum of weights of all processes that support value 0 or 1 in V. The value with the weighted majority is decided upon. Finally, in step 4, processes receive the common feedback from the environment to determine the correct value. If the value decided was incorrect, then the weights of the processes that proposed an incorrect value is reduced by a constant proportion ϵ $(0 < \epsilon < 1)$ of its previous weight (multiplicative update). We show in [4] that this algorithm guarantees agreement and termination. We summarize the accuracy results in the following section.

4. ACCURACY GUARANTEES

We make guarantees on the accuracy of the ABAF algorithm based on the accuracy of the processes in the system and the proportion of faulty and non-faulty processes. A process is called accurate for an iteration if it proposes the correct value for that particular iteration of the ABAF algorithm. We define fault ratio (r), to be the ratio of the total weight of the faulty processes to that of the non-faulty processes.

THEOREM 1. *(Deterministic Accuracy) For each iteration, if the total weight of the accurate processes is greater than* $(1/2 + d)$ *times the total weight of the non-faulty processes, and if the initial fault ratio of the system is less than* $2d$*, i.e.,* $(r_0 < 2d)$*, then the ABAF algorithm guarantees accuracy.*

THEOREM 2. *(Probabilistic Accuracy) Let all weights in the system be in* $[0, 1]$*. For any iteration of the ABAF algorithm, if the probability with which a non-faulty process proposes the correct value* β*, is greater than* $1/2 + d$ $(0 < d < 1/2)$ *and if the fault ratio of the system is less than* $2d$*, i.e.,* $(r < 2d)$*, then the ABAF algorithm guarantees accuracy with probability greater than* $1 - (\frac{e^{-\delta}}{(1-\delta)^{(1-\delta)}})^{\mu}$*, where* $\mu = p(1/2 + d)$ *and* $\delta = (2d - r)/(2d + 1)$*.*

THEOREM 3. *(At-Least-One Accuracy) If there exists at least one process such that it is inaccurate at most* b *out of* j *iterations of the ABAF algorithm, then the algorithm is inaccurate at most* $2(1 + \epsilon)b + (2/\epsilon) \log n$ *times.*

5. EXPERIMENTAL EVALUATION

The experimental evaluation compares three different update methods: "update-on-inaccuracy" (model used in the ABAF algorithm), in which the weights are updated only when the decided value is incorrect, "always-update", in which the weights are updated after every iteration, and "never-update", in which the weights are never updated. The last option reduces to standard Byzantine agreement. Our simulation uses two models for faulty processes. Model 1 uses a Byzantine process that will always propose the incorrect value. Model 2 uses a Byzantine process that proposes the correct value based on the percentage of its own weight to the weight of all processes. We compare the performance of the three update methods for all three accuracy models (deterministic, probabilistic, at-least-one). The results of all three experiments show that always-update performs very well with model 1 and very badly with model 2, never-update performs vice-versa, but update-on-inaccuracy, the model used in this paper, offers the best compromise for both models.

6. REFERENCES

[1] P. Berman, J. Garay, and K. Perry. Towards optimal distributed consensus. *Foundations of Computer Science, Annual IEEE Symposium on*, 0:410–415, 1989.

[2] P. Berman and J. A. Garay. Asymptotically optimal distributed consensus (extended abstract). In *Proceedings of the 16th International Colloquium on Automata, Languages and Programming*, pages 80–94, London, UK, 1989. Springer-Verlag.

[3] A. Clement, M. Marchetti, E. Wong, L. Alvisi, and M. Dahlin. Making byzantine fault tolerant systems tolerate byzantine faults. In *6th USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, Apr. 2009.

[4] V. K. Garg, J. Bridgman, and B. Balasubramanian. A report on accurate byzantine agreement with feedback. Technical Report TR-PDS-2011-001 `http://maple.ece.utexas.edu/TechReports/2011/TR-PDS-2011-001.pdf`, Parallel and Distributed Systems Laboratory, ECE Dept. University of Texas at Austin, 2011.

[5] S. Kale. *Efficient algorithms using the multiplicative weights update method.* PhD thesis, Princeton, NJ, USA, 2007. AAI3286120.

[6] V. King and J. Saia. Breaking the $O(n^2)$ bit barrier: scalable byzantine agreement with an adaptive adversary. In *Proceeding of the 29th ACM SIGACT-SIGOPS symposium on Principles of distributed computing*, PODC '10, pages 420–429, New York, NY, USA, 2010. ACM.

[7] L. Lamport, R. Shostak, and M. Pease. The byzantine generals problem. *ACM Trans. Program. Lang. Syst.*, 4:382–401, July 1982.

[8] N. Littlestone and M. K. Warmuth. The weighted majority algorithm. *Inf. Comput.*, 108:212–261, February 1994.

[9] M. Pease, R. Shostak, and L. Lamport. Reaching agreements in the presence of faults. *Journal of the ACM*, 27(2):228–234, Apr. 1980.