Comparative Evaluation of Approximate Byzantine Vector Consensus Algorithms

Shiyang Cheng, Kyle Sung
shiyangcheng@utexas.edu, kyle@utexas.edu

***Abstract*—This is an abstract.**

I. Introduction

This paper describe two approximate multidimension concensus algorithms in distributed system. These algorithms are different from traditional concensus algorithms. They are meant to resolve the Byzantine problem which distributed system contains arbitrary failures. In traditional Byzantine problem: there are n processes in the system, several of them are faulty processes which can be considered generate any possible output in the system. Each non-faulty process will propose one value and then they will get one concensus value which need to meet several conditions:

- Termination: Every correct process eventually delivers some message
- Agreement: If a correct process delivers value m, then all correct processes deliver m
- Nontriviality: Both values should be possible outcomes. This property eliminates the protocals that returns a fixed value independent of the initial input

In multidimensional system, all the processes will propose one vector of values, and all non-faulty processes will get concensus on the n-dimensional value. The multidimensional input which is d-dimensional vector can be considered as a point in d-dimensinal Euclidean space with $d > 0$. In the multidimentional Byzantine Concensus problem, the out come of each process should also be identical. And the output value need to be in the convex hull of the non-faulty processes' input in the d-dimensinal Euclidean space.

To solve this problem, reasearchers also propose another problem named Byzantine Approximate Agreement problem. This problem also defined the out come of non-faulty processes will be in the convex hull. The outputs shold be within the Euclidean distance ϵ of each other.

In these problem, the algorithms defined in a model include following property:

- All message will be eventually delivered
- Any two processes is connected to each other
- The communication channel is FIFO channel
- The processes can identify the sender by the sender ID in the message

From Vaidya and Garg's obversation, simply performing scalar consensus on each dimension of the input vectors independently does not solve the vector consensus problem. In particular, even if validity condition for scalar consensus is satisfied for each dimension of the vector separately, the above validity condition of vector consensus may not necessarily be satisfied. For instance, suppose that there are four processes, with one faulty process[2].

A. Multidimensional Byzantine Concensus

For synchronous system, the algorithms will run in round by round. In each round, processes will send messages and receive messages which were sent in this round.

A protocal solving the Multidimensional Byzantine Concensus problem need to satisfy following conditions[1]:

- Agreement. The output vector at all the non-faulty processes must be identical.
- Validity. The output vector at all non-faulty processes must be in the convex hull of the non-faulty inputs.
- Termination. Each non-faulty process must terminate within a finite amount of time.

This is known that $n > 3f$ is necessary and sufficient to solve the scalar consensus, under the condition that the commnunication model is a complete graph.

B. Multidimensional Byzantine Approximate Agreement

In asynchronous systems, the message deliver time is not guarenteed. The message may take unbound time to deliver. Also, there is not disjoint round in the algorithmes. It is not possible to identify a process is faulty or slow[attiya2004distributed]. And it is well-known that asynchronous scalar consensus is impossible in the presence of even a single crash failure[**?**]. Here we discuss the algorithms are also under the same condition, but the input and output switched to a vector values.

A protocol satisfying these conditions could be considered solving the Multidimensional Byzantine Approximate Agreement problem:

- Agreement. The output vectors of non-faulty processes should be within Euclidean distance $\epsilon > 0$, a constant defined a priori.
- Validity. The output vector at all non-faulty processors must be inside the convex hull of the input inputs.
- Termination. Each non-faulty process must terminate within a finite amount of time.

II. Asynchronous communication primitives

A. Reliable Broadcast

Reliable broadcase

B. *Witness Technique*

Witness Technique

III. Safe Area

A. *Introduction*

Introduction to safe area.

B. *Algorithms*

Algorithms.

IV. Sufficient Condition for Multidimensional Approximate Agreement

A. *AAD*

B. *MH*

C. *VG*

V. Comparasion

A. *Time complexity*

Time complexity.

B. *Running Time*

Running time.

VI. Conclusion

This is conclusion

References

[1] M. J. Fischer, N. A. Lynch, and M. S. Paterson. Impossibility of distributed consensus with one faulty process. *Journal of the ACM (JACM)*, 32(2):374–382, 1985.

[2] H. Mendes, M. Herlihy, N. Vaidya, and V. K. Garg. Multidimensional agreement in byzantine systems. *Distributed Computing*, 28(6):423–441, 2015.

[3] N. H. Vaidya and V. K. Garg. Byzantine vector consensus in complete graphs. In *Proceedings of the 2013 ACM symposium on Principles of distributed computing*, pages 65–73. ACM, 2013.