Creating Efficient Algorithms for Byzantine Agreement

Benjamin Sommer, University of Illinois at Urbana-Champaign
ADVISOR: Professor Nitin Vaidya

Background

As the number of services being provided online rapidly expands, distributed applications need protection against the arbitrary failure of any piece of their networks. These failures can take the form of computer crashes, buggy code, or malicious attack and are called Byzantine failures. Two previously untested algorithms, GLNV and Digest, are presented below to guarantee the accuracy and efficiency of data transfer in the presence of Byzantine failures.

Algorithm Explanations

Basic:
The standard algorithm in which the source sends the data to all other servers, which forward their copies to everyone else.

GLNV:
An untested algorithm where the source sends out a different piece of the data to each of the servers plus a check sum to verify integrity. Now peer servers need only forward their part of the data.

Digest:
In the second untested algorithm, like Basic, the source sends the full data to all other servers. But then each server computes a hash of the data and forwards that to its peers.

Fundamental Questions

• Will the untested GLNV and Digest algorithms outperform the standard Byzantine agreement algorithm?
• If one or more do outperform the current standard, do they do so in every situation or only under certain constraints?

Research Plan

• Gain an understanding of how each of the algorithms works
• Create an implementation of each of the three algorithms
• Devise and execute a number of tests to determine each algorithm’s characteristics and properties

Research Results

Both of the new algorithms, GLNV and Digest, add a small non-data overhead in exchange for excellent scaling. As seen from the graph below, the results are quite significant: A speed-up of over 3.5 times on files of size 1 MB.

Basic

GLNV

Digest

GLNV and Digest Packet Size (bytes)

0 500000 1000000 1500000 2000000 2500000

Time Elapsed (s)

0 1 2 3 4 5 6 7

L=2MB

GLNV and Digest Packet Size (bytes)