Probabilistic Byzantine Tolerance for Cloud Computing

Proposition de stage dans le cadre du LABEX SMART

Encadrant - Laboratoire d’accueil
Olivier Marin (MdC) - Laboratoire d’Informatique de Paris 6

Date et durée prévues
6 mois à partir du 1er février jusqu’au 31 juillet

Demande financière
2616 euros (6 mois à 436 euros)

Context
Heavy computational tasks are often performed these days in cloud computing environments by splitting the computation into multiple sub-tasks using platforms such as Apache Hadoop, Spark, etc. Commonly, these platforms are structured around a scheduler that disseminates the computation sub-tasks onto available worker nodes responsible for computing the results of these sub-tasks and returning their results. Obviously, this involves an implicit assumption that scheduling and routing sub-tasks to worker nodes is considerably cheaper than calculating them, as otherwise it would not make sense to do so.

When trying to make such systems resilient to Byzantine failures, one is faced with the large replication costs of masking Byzantine failures. Specifically, under the assumption that there could be up to $f$ Byzantine nodes in the system, each computation sub-task must be executed by $3f+1$ nodes in naive application of traditional Byzantine Fault Tolerance (BFT) approaches. Even more sophisticated mechanisms that distinguish between execution and validation require between $f + 1$ [1] to $2f + 1$ [2]. As an example, if 10 nodes might be Byzantine, this means that each computation task needs to be executed on 11 or 21 nodes, as the case may be.

The principle of elasticity in Cloud computing is to adapt resource provisioning as a means to optimize the tradeoff between cost and performance. Conversely tolerating Byzantine failures induces incompressible costs, and for traditional BFT approaches these costs quickly explode for in the context of Clouds [3] [4] [5]. Furhtermore, traditional BFT protocols require strong assumptions on the number of Byzantine nodes that exist in the system. In particular, the choice of a bound $f$ is generally arbitrary although it impacts strongly on the overhead.

Subject
In this internship we wish to explore an alternative design path, which trades provable correctness of each computation step with a probabilistic one in order to reduce the amount of resources required by the system.

Specifically, we assume that each node $j$ has a given probability $p_j$ of acting in a Byzantine manner during an arbitrary calculation of a computation sub-task. We define the
reputation $r_j$ of $j$ as $1 - p_j$. In addition, rather than requiring absolute masking of Byzantine failures, we only require that the probability of obtaining a correct answer to each computation sub-task remains above a given threshold. Consequently, each computation sub-task needs to be replicated only over the minimal number of worker nodes that would ensure meeting this probabilistic threshold.

As a motivating example, when a computation is sent to a group of worker nodes and only part of them generate an incorrect answer, the scheduler can send the same computation sub-task to additional nodes until the probability of obtaining the correct result is above the given threshold. If all replies are the same, an incorrect result will go undetected by the scheduler only if all the nodes are Byzantine. Let the probabilities of acting in a Byzantine manner be independent and identically distributed (IID) and equal to 0.1. Then we can ensure that such an undetected Byzantine failure will occur with a probability of at most 0.0001 by replicating the computation on 4 nodes only. The figure below shows an example where a task $T$ requires scheduling. Among its list of available workers, the Trusted Scheduler $(TT)$ provisions Worker 1 and Worker 2 since their joint probability of success is higher than $\Theta$, the application threshold. $TT$ also selects another node (Worker 3) which will be provisioned to bring the joint probability of success to an even higher value $\Theta'$ if Worker 1 and Worker 2 disagree on the result.

The goal of the internship is to explore resource allocation algorithms based on this principle, as we believe it bears great potential for reduced resource consumption compared to traditional approaches. This work falls within the scope of item 3, Virtualization and Clouds, in Program 4, Autonomic Distributed Environments for Mobility, of the SMART LABEX. This item targets highly dynamic distributed environments which induce critical issues in terms of resource consumption, performance, and reliability.

References


