Future Generation Computer Systems 55 (2016) 87–100

Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

Toward integrity assurance of outsourced computing — a game theoretic perspective

Yongzhi Wang [a,b,*], Jinpeng Wei [c], Shaolei Ren [d], Yulong Shen [a]

[a] School of Computer Science and Technology, Xidian University, 2 South Taibai Road, Xi'an, Shaanxi, PR China
[b] Florida International University, ECS 254 11020 SW 8th Street, FL, USA
[c] Florida International University, ECS 389 11020 SW 8th Street, FL, USA
[d] University of California, Riverside, CA, USA

HIGHLIGHTS

- Our method can ensure high result integrity in outsourced computing systems.
- Our algorithm can guarantee the highest result integrity under system restrictions.
- We proved the correctness of the proposed algorithms.
- We performed experiments to show the effectiveness of the proposed algorithms.

ARTICLE INFO

Article history:
Received 25 March 2015
Received in revised form
12 July 2015
Accepted 19 August 2015
Available online 3 September 2015

Keywords:
Outsourced computing
Game theory
Integrity assurance
Task scheduling

ABSTRACT

Outsourced computing is gaining popularity in recent years. However, due to the existence of malicious workers in the open outsourced environment, offering high accuracy computing services is critical and challenging. A practical solution for this class of problems is to replicate outsourced tasks and compare the replicated task results, or to verify task results by the outsourcer herself. However, since most outsourced computing services are not free, the portion of tasks to be replicated or verified is restricted by the outsourcer's budget. In this paper, we propose *Integrity Assurance Outsourced Computing* (IAOC) system, which employs probabilistic task replication, probabilistic task verification and credit management techniques to offer a high accuracy guarantee for the generalized outsourced computing jobs. Based on IAOC system, we perform theoretical analysis and model the behaviors of IAOC system and the attacker as a two-player zero sum game. We propose two algorithms, *Interactive Gradient Descent* (IGD) algorithm and *Tiered Interactive Gradient Descent* (TIGD) algorithm that can find the optimal parameter settings under user's accuracy requirement, without or with considering user's budget requirement. We prove that the parameter setting generated by IGD/TIGD algorithm form a Nash Equilibrium, and also suggests an accuracy lower bound. Our experiments show that even in the most severe situation, where the malicious workers dominate the outsourced computing environment, our algorithm is able to find the parameter settings satisfying user's budget and accuracy requirement.

1. Introduction

Outsourced computing is gaining popularity in recent years. The essential reason behind such a booming technical trend is the challenge of ever increasing computation scale and difficulty.

The increase of computing scale is substantiated by the computations whose input data sizes are significantly large, known as "big data". The solution to such a challenge is to adopt parallel computation paradigm, such as MapReduce [1]. The parallel computation framework is usually running as a cluster, which consists of multiple worker nodes. During computation, the input is split into multiple chunks, each of which is assigned to a worker node and processed as a task. Such a framework could easily scale up by adding more computation nodes to the cluster if the input size increases. However, since the customer does not want or cannot afford of investing a computation cluster, she usually

* Corresponding author at: School of Computer Science and Technology, Xidian University, 2 South Taibai Road, Xi'an, Shaanxi, PR China.
E-mail addresses: yongzhiwang@icloud.com (Y. Wang), weijp@cs.fiu.edu (J. Wei), sren@ece.ucr.edu (S. Ren), ylshen@mail.xidian.edu.cn (Y. Shen).

http://dx.doi.org/10.1016/j.future.2015.08.010
0167-739X/

outsources her computation to a computation service provider (e.g., cloud vendor[1] or grid computing network [2,3]), on which she can create a cluster swiftly and on demand, while saving the costs of infrastructure set up and maintenance.

The challenge of computation difficulty is substantiated by the fact that computers cannot replace humans on tasks that require human intelligent, such as recognizing images, finding bushiness information on the Internet, or solving Captchas. In most cases, such a class of computation is not difficult for human being. Yet they are tedious and require a lot of human effort. Crowdsourcing such as Mechanical Turk[2] offers an option to address such a challenge. In this computing paradigm, the crowdsourcing company categorizes similar tasks into batches, and assigns tasks to humans on the internet to solve. The crowdsourcing company receives the task results from the workers, verifies the correctness of their results and pays workers based on the correctness of their task results.

Although outsourced computations can have a variable of different formats, they share the following similar characteristics.

(a) The computation job is usually significantly large, so that the job owner cannot process it locally. Therefore, such jobs usually are split into multiple tasks, and computed by computation service providers.
(b) The task number of a job is usually proportional to the input data size and is usually large. When the input data size increases, the job can scale up by increasing the task number.
(c) The outsourced computation is usually not free. The computation service provider usually charges the user based on the amount of computation performed (correctly).

However, most outsourced computing paradigms share the same vulnerability: *Since the tasks are outsourced to untrusted workers, if some workers are malicious, they could tamper the task result, and therefore affect the job result correctness.*

In this paper, we extend the existing solution in [4] and [5] to a generalized class of outsourced computing jobs, propose *Integrity Assurance Outsourced Computing* (*IAOC* for short) system. IAOC employs probabilistic task replication, probabilistic task verification and credit management techniques to offer high result accuracy assurance. We perform theoretical analysis on IAOC system and model the accuracy of outsourced computing based on the system parameters. We find that the outsourced computing accuracy is determined together by IAOC system and the participant malicious workers. Hence, we model the outsourced computing as a two-player zero sum game, where the IAOC system and the malicious workers can adjust their behaviors to maximize their benefits. We propose a novel algorithm called *Interactive Gradient Descent* algorithm (*IGD* for short) to search for the optimal behaviors (i.e., parameter setting) for IAOC system that satisfies user's accuracy requirement. We prove that the algorithm generated optimal behavior forms Nash Equilibrium. In other words, either the IAOC and the attacker do not have incentive to deviate from the algorithm suggested behavior. Therefore, the algorithm predicted accuracy is the highest lower bound.

Based on IGD algorithm, we propose a *Tiered Interactive Gradient Descent* (*TIGD*) algorithm, that also generates optimal behavior for IAOC system and the malicious worker, while considering user's outsource budget from multiple aspects (see Section 5.2). The generated optimal behavior satisfies user's budget requirement, meanwhile, guarantees the highest lower bound of the job accuracy.

We perform a set of experiments based on TIGD algorithm. The result shows that the algorithm is effective in finding optimal parameter setting even in the most severe situation where all the workers in the outsourced environment are malicious.

The rest of this paper is organized as follows. Section 2 defines the generalized outsourced computation system and the system assumptions. Section 3 presents the integrity assurance outsourced computing (IAOC) system on the generalized outsourced computing system. Section 4 builds a mathematical model of IAOC system to measure the accuracy of the job and performs a set of simulation based on this model. Section 5 presents the IGD and TIGD algorithm. Section 6 describes and analyzes the experiment result. Section 7 discusses related work, and Section 8 concludes the paper.

2. System definition and assumption

2.1. System definition

In this section, we define the generalized outsourced computing system. The entity who provides the outsourced computing service is called a *provider*. The entity who outsources its computation to the provider is called a *user*. A provider maintains a distributed system to perform the computation. The system consists of a *master* and many *workers*. The master controls the entire computation and responsible for assigning computation to workers. Each worker is the actual entity to perform the computation. The unit of computations outsourced by the user is a *job*. Since the outsourced job is usually significantly large, a job is usually split into multiple (maybe a significant number of) *tasks*, which are assigned to different workers to compute. As a reward, the provider charges the user according to the amount of computation she has performed for that user. The user can verify the task results returned by the provider. If the user finds that the task results are incorrect, she can reject the provider's results, but she has to show the proof to the provider that the results are incorrect. Therefore, the user only pays the provider the amount of computation that she accepts.

In some scenarios, the master accepts jobs from the user and hires workers to compute those jobs. For example, in crowdsourcing computing, the crowdsourcing company accepts jobs from the user, assigns tasks to workers on the Internet and pays workers on behalf of the user. The crowdsourcing company is trusted. Therefore, the crowdsourcing company can verify the task results for the user and determine whether to accept or reject the worker submitted results.

2.2. System assumption

In the outsourced computing system, we assume the master is trusted, but workers can be malicious due to the open environment. The malicious workers' goal is to insert incorrect results to the job without been detected. For example, in the crowdsourcing computing scenario, the worker can return trivial but incorrect answers to the user in order to earn more money. We assume the malicious workers are highly intelligent. For example, they can exchange the task information and coordinate with each other to cheat at the optimal time. For instance, if two tasks that compute the same input are assigned to two malicious workers simultaneously, they can return the same erroneous results (i.e., to collude) so that simply comparing the task result results cannot detect the error. We call such malicious workers *collusive* workers.

We assume that the task number in a job is big enough and all task results are equivalently important to the overall job result. In other words, if a small number of task results are incorrect and undetected, it will not undermine the overall job accuracy

[1] Amazon Web Services. http://aws.amazon.com.
[2] https://www.mturk.com.