AN EFFICIENT CODING FRAMEWORK FOR WIRELESS DISTRIBUTED COMPUTING

G.Divya[1], E.Kavitha[2], U.Sundhar[3]

[1, 2] Final Year Students, [3] Assistant Professor,CSE
Thiruvalluvar College of Engineering & Technology, Vandavasi. Tamil Nadu.
divyagad10@gmail.com[1], kavithaev1997@gmail.com[2], sundhars@gmail.com[3]

ABSTRACT - In Wireless distributed computing system, in which multiple mobile users, connected wirelessly through an access point, collaborate to perform a computation task. In particular, users communicate with each other via the access point to exchange their locally computed intermediate computation results, which is known as data shuffling. A unified coded framework for distributed computing with straggling servers, by introducing a tradeoff between "latency of computation" and "load of communication" for some linear computation tasks has been proposed. We show that the coded scheme of that repeats the intermediate computations to create coded multicasting opportunities to reduce communication load, and the coded scheme of that generates redundant intermediate computations to combat against straggling servers can be viewed as special instances of the proposed framework, by considering two extremes of this tradeoff: minimizing either the load of communication or the latency of computation individually. Furthermore, the latency load tradeoff achieved by the proposed coded framework allows to systematically operating at any point on that tradeoff to perform distributed computing tasks. We also prove an information theoretic lower bound on the latency-load tradeoff, which is shown to be within a constant multiplicative gap from the achieved tradeoff at the two end points.

1. INTRODUCTION

A distributed computer system consists of multiple software components that are on multiple computers, but run as a single system. The computers that are in a distributed system can be physically close together and connected by a local network, or they can be geographically distant and connected by a wide area network. A distributed system can consist of any number of possible configurations, such as mainframes, personal computers, workstations, minicomputers, and so on. The goal of distributed computing is to make such a network work as a single computer.

Distributed computing systems can run on hardware that is provided by many vendors, and can use a variety of standards-based software components. Such systems are independent of the underlying software. They can run on various operating systems, and can use various communications protocols. Some hardware might use UNIX or Linux as the operating system, while other hardware might use Windows operating systems. For inter machine communications, this hardware can use SNA or TCP/IP on Ethernet or Token Ring.

Wireless Computing Definition: Wireless USB (WUSB) is a form of **Universal Serial Bus** (USB) technology that uses **radio-frequency** (RF) links rather than cables to provide the interfaces between a computer and peripherals, such as monitors, printers, external drives, head sets, MP3 players and digital cameras.

Distributed computing find its application in the following area

- A Communication Efficient distributed learning framework for smart environments
- gRPC: A Framework for Efficient Service Architectures
- Prediction or Not? An Energy-Efficient Framework for Clustering-Based Data Collection in Wireless Sensor Networks
- Internet of Things (IoT): A Vision, Architectural Elements, and Future Directions
- An energy-efficient distributed in-network caching scheme for green content-centric networks
- Robust wireless video multicast based on a distributed source coding approach
- Power efficient data gathering and aggregation in wireless sensor networks

2. REVIEW OF LITERATURE SURVEY

In this paper," Poster abstract: A scalable coded computing framework for edge-facilitated wireless distributed computing [1]". They propose a scalable coded distributed computing framework for wireless distributed computing over a cluster of mobile users, in which the data shuffling across users are performed through an access point at the edge of the network. The proposed framework achieves a constant shuffling load that is independent of the number of participating users. The key idea is to utilize a particular repetitive structure of computation assignments at the users, in order to provide coding opportunities that reduce the shuffling load by a factor that grows linearly with the number of users. Data shuffling method exchange data through locally computed and stored in intermediated computers. It reduces the uplink communication bandwidth from user to access point and downlink communication bandwidth from access point to user this optimality in a decentralized environment is adopted in our project.

Current mobile applications treat the end-user device as a "thin client," with all of the heavy computations being offloaded to an infrastructure cloud. However, the computational capabilities of mobile devices are constantly improving, and it is worthwhile considering whether an edge-cloud that consists purely of mobile devices (operating effectively as "thick clients") can perform as well as, or even better than, an infrastructure cloud. In this paper, they study the trade-offs between offloading computation to an infrastructure cloud versus retaining the computation within a mobile edge-cloud. To this end, they develop and run two classes of applications on both types of clouds, and they analyze the performance of the two clouds in terms of the time taken to run the application, along with the total amount of battery power consumed in both cases. Our results indicate that there are indeed classes of applications where an edge-cloud can outperform an infrastructure cloud in terms of both latency and battery power. From this paper," The case for mobile edge-clouds [2]" an enhanced infrastructure for cloud environment is given which saves battery power and avoid latency.

Mobile device like a smart phone is becoming one of main information processing devices for users these days. Using it, a user not only receives and makes calls, but also performs information tasks. However, a mobile device is still resource constrained, and some applications, especially work related ones, usually demand more resources than a mobile device can afford. To alleviate this, a mobile device should get resources from an external source. One of such sources is cloud computing platforms. Nevertheless an access to these platforms is not always guaranteed to be available and/or is too expensive to access them. They envision a way to overcome this issue by creating a virtual cloud computing platform using mobile phones. They argue that due to the pervasiveness of mobile phones and the enhancement in their capabilities this idea is feasible. In this paper," A virtual cloud computing provider for mobile devices [3]" a virtual cloud computing platform with mobile phone is adopted which minimizes the resource constraint. This is adopted for effective utilization of resources.

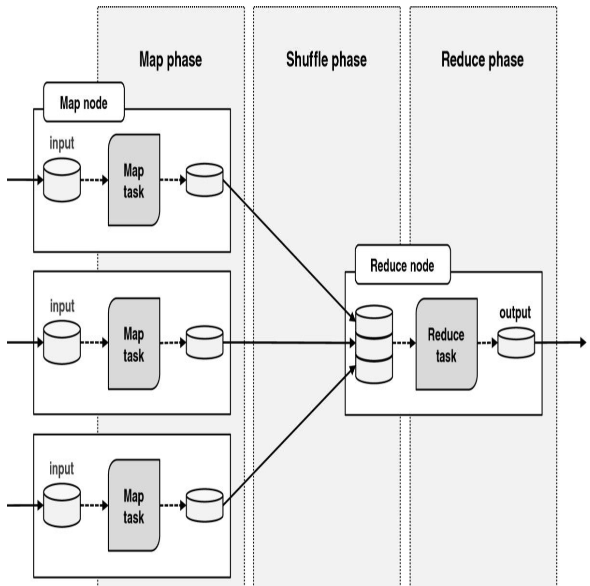

Fig: 1 Efficient Coding Framework

3.SYSTEM DESIGN

3.1 Existing System

In this work, a wireless distributed computing system, in which multiple mobile users, connected wirelessly through an access point, collaborate to perform a computation task. In particular, users communicate with each other via the access point to exchange their locally computed intermediate computation results, which is known as *data shuffling*. We propose a scalable framework for this system, in which the required communication bandwidth for data shuffling does not increase with the number of users in the network. The key idea is to utilize a particular repetitive pattern of placing the data set (thus a particular repetitive pattern of intermediate computations), in order to provide the coding opportunities at both the users and the access point, which reduce the required uplink communication bandwidth from users to the access point and the downlink communication bandwidth from access point to users by factors that grow linearly with the number of users. We also demonstrate that the proposed data set placement and coded shuffling schemes are optimal (i.e., achieve the minimum required shuffling load) for both a centralized setting and a decentralized setting, by developing tight information-theoretic lower bounds.

3.2 Proposed System

In wireless distributed computing system, in which multiple mobile users, connected wirelessly through an access point, collaborate to perform a computation task. In particular, users communicate with each other via the access point to exchange their locally computed intermediate computation results, which is known as *data shuffling .In this work* , a unified coded framework for distributed computing with straggling servers, by introducing a tradeoff between "latency of computation" and "load of communication" for some linear computation tasks has been proposed. We show that the coded scheme of that repeats the intermediate computations to create coded multicasting opportunities to reduce communication load, and the coded scheme of that generates redundant intermediate computations to combat against straggling servers can be viewed as special instances of the proposed framework, by considering two extremes of this tradeoff: minimizing either the load of communication or the latency of computation individually. Furthermore, the latency load tradeoff achieved by the proposed coded framework allows to systematically operate at any point on that tradeoff to perform distributed computing tasks. We also prove an information theoretic lower bound on the latency-load tradeoff, which is shown to be within a constant multiplicative gap from the achieved tradeoff at the two end points.

4. CONCLUSION

Thus in this work , An unified coded framework for distributed computing with straggling servers, by introducing a tradeoff between "latency of computation" and "load of communication" for some linear computation tasks has been proposed. We show that the coded scheme of that repeats the intermediate computations to create coded multicasting opportunities to reduce communication load, and the coded scheme of that generates redundant intermediate computations to combat against straggling servers can be viewed as special instances of the proposed framework, by considering two extremes of this tradeoff: minimizing either the load of communication or the latency of computation individually. Furthermore, the latency load tradeoff achieved by the proposed coded framework allows to systematically operating at any point on that tradeoff to perform distributed computing tasks. We also prove an information theoretic lower bound on the latency-load tradeoff, which is shown to be within a constant multiplicative gap from the achieved tradeoff at the two end points.

REFERENCES

[1]S. Li, Q. Yu, M. A. Maddah-Ali, and A. S. Avestimehr, "Poster abstract: A scalable coded computing framework for edge-facilitated wireless distributed computing," in *Proc. IEEE/ACM Symp. Edge Comput. (SEC)*, Oct. 2016, pp. 79–80.

[2]U. Drolia *et al.*, "The case for mobile edge-clouds," in *Proc. IEEE 10th Int. Conf. Ubiquitous Intell. Comput. (UIC)*, Dec. 2013, pp. 209–215.

[3]G. Huerta-Canepa and D. Lee, "A virtual cloud computing provider for mobile devices," in *Proc. 1st ACM Workshop Mobile Cloud Comput. Services Soc. Netw. Beyond*, 2010, Art. no. 1.

[4]M. Chowdhury, M. Zaharia, J. Ma, M. I. Jordan, and I. Stoica, "Manag-ing data transfers in computer clusters with orchestra," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 41, no. 4, pp. 98–109, Aug. 2011.

[5]S. Li, M. A. Maddah-Ali, and A. S. Avestimehr, "Fundamental tradeoff between computation and communication in distributed computing," in *Proc. IEEE ISIT*, Jul. 2016, pp. 1814–1818.

[6] J. Dean and S. Ghemawat, "MapReduce: Simplified data processing on large clusters," in *Proc. 6th USENIX (OSDI)*, Dec. 2004, p. 10.