A Scalable Framework for Wireless Distributed Computing

ABSTRACT

Recent years have witnessed a rapid growth of computationally intensive applications on mobile devices, such as mapping services, voice/image recognition, and augmented reality. The current trend for developing these applications is to offload computationally heavy tasks to a "cloud", which has greater computational resources. While this trend has its merits, there is also a critical need for enabling *wireless distributed computing*, in which computation is carried out using the computational resources of a cluster of wireless devices collaboratively. Wireless distributed computing eliminates, or at least de-emphasizes, the need for a core computing environment (i.e., the cloud), which is critical in several important applications, such as autonomous control and navigation for vehicles and drones, in which access to the cloud can be very limited. Also as a special case of the emerging "Fog computing architecture" it is expected to provide significant advantages to users by improving the response latency, increasing their computing capabilities, and enabling complex applications in machine learning, data analytics, and autonomous operation.

EXISTING SYSTEM

In Existing System, Without the help from a centralized cloud, the local computing capability of a wireless device is often limited by its local storage size. For example, for a mobile navigation application in which a smart car wants to compute the fastest route to its destination, over a huge dataset containing the map information and the traffic conditions over a period of time, the local storage size of an individual car is too small to store the entire dataset, and hence individual processing is not feasible. However, using 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*.

DIS ADVANTAGES

- Increasing the shuffling load.

- Individual processing is not feasible.

PROPOSED SYSTEM

In Proposed System, 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.

ADVANTAGES

- It improves the response latency, increasing their computing capabilities.
- Random linear coding at the access point has been utilized before in solving network coding problems.

SYSTEM REQUIREMENTS

H/W System Configuration:-

Processor	-	Pentium –III
RAM	-	256 MB (min)
Hard Disk	-	20 GB
Key Board	-	Standard Windows Keyboard
Mouse	-	Two or Three Button Mouse
Monitor	-	SVGA

S/W System Configuration:-

Operating System	:	Windows95/98/2000/XP
Application Server	:	Tomcat5.0/6.X
Front End	:	HTML, Jsp
Scripts	:	JavaScript.
Server side Script	:	Java Server Pages.
Database	:	MySQL 5.0
Database Connectivity	:	JDBC