Context-aware Distributed Storage in Mobile Cloud Computing

Dong Han, Ye Yan, and Tao Shu
Department of Computer Science and Engineering, Oakland University, Rochester, MI, U.S.A.
{dhan, yyan, shu} @oakland.edu

Abstract

We improve data access efficiency in mobile cloud storage by exploring users' context information. Specifically, we optimize data reading and writing on mobile devices from/to distributed cloud storage according to network condition, user mobility pattern, and data access preference. We propose a Reed-Solomon erasure code based context-aware distributed storage system. A mixed integer linear programming (MILP) is proposed to optimize the efficiency of the system. The proposed method is verified through various simulation scenarios.

Introduction

Mobile cloud storage (MCS) allows users to access data at various locations using mobile devices such as smartphones and tablets, aiming at providing a reliable and robust user data storage.

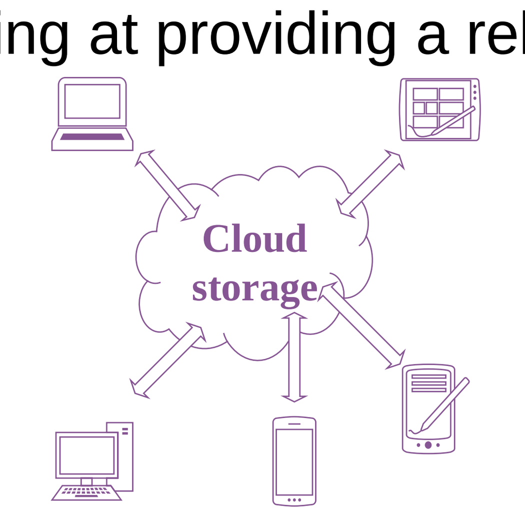

Fig. 1. Centralized cloud storage

Mainstream MCS systems are based on a centralized cloud storage architecture, as illustrated in Figure 1. An existing problem in a data center based MCS architecture is:

- The data storages are geographically far away from users, and thus inevitably incurs high communication overhead in delivering the data to mobile users.

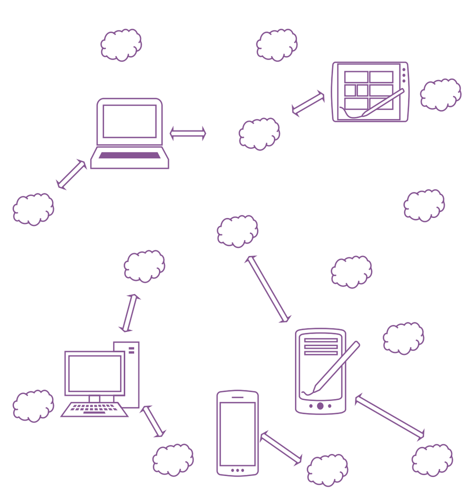

Fig. 2. Distributed cloud storage

To solve the issue in a centralized cloud storage, distributed MCS provides lots of widely deployed storage nodes to move data closer to users, as shown in Figure 2. However, to develop an efficient distributed MCS is challenging:

- Mobile users stay at different places with various time spans.
- Storing data on the node closest to the mobile user may not be the best choice due to network bottleneck.
- A user presents different data access preferences (intensity to read or write/update) at different places.

To solve the above problems, our contributions are summarized as:

- We developed a comprehensive context-aware analytical model which accounts three user context factors: network condition, user mobility pattern, and data access preference.
- Our proposed distributed storage takes the advantage of Reed-Solomon erasure code[1], which leads to a storage capacity gain over the legacy data duplication method.
- We established a mixed integer linear programming to minimize the average transmission time of the system.

System Model and Problem Formulation

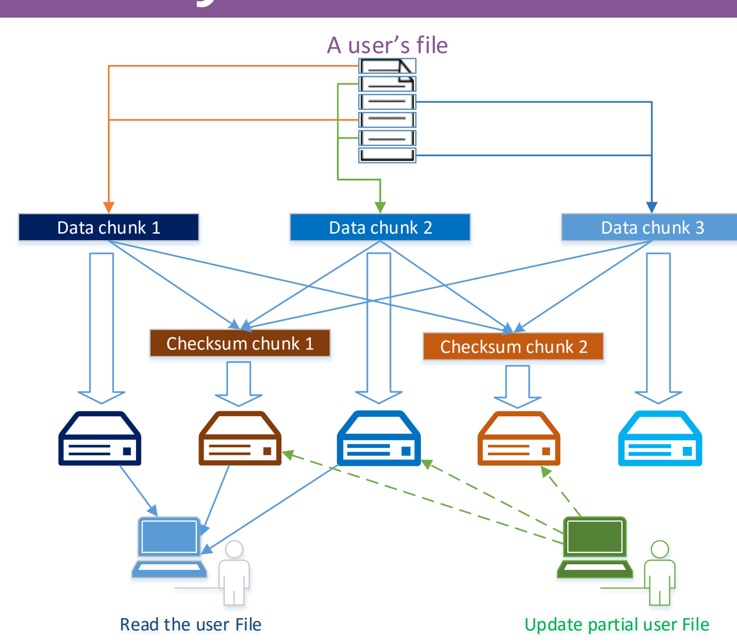

Fig. 3. An example of a file access in 3 out of 5 erasure coded distributed storage

Based on Reed-Solomon erasure code, a user file is encoded into k data chunks. Then r checksum chunks are generated by linear combinations of the k data chunks. These $m = k + r$ chunks are stored in m different storage servers. Data can be reconstructed by accessing any k out of the m servers. The storage model is shown as Figure 3.

The objective function of the optimization is to minimize the expected global transmission time in data access of all users: $\underset{X,Y,Z}{\text{minimize}} \sum_{h \in U} T_h^{op}$.

The X, Y are matrices of decision variables denote selected storage nodes to store data chunks, and checksum chunks, respectively.

The Z is a matrix of decision variable for storage nodes chosen to provide read service. We use T_h^{op} to denote the expected transmission time of user h, where the h is in the set of users U.

To optimize the data access efficiency, the context-aware model is established by considering the 3 types of context information, as constraints of the objective function:

- Network condition: it is measured as bandwidths between place of interests(POIs) and storage nodes.
- User mobility pattern: We discover users' frequently visited POIs[2].
- Data accessing preferences: The probability of a user read/write files at each POI based on his/her long-term file operating history.

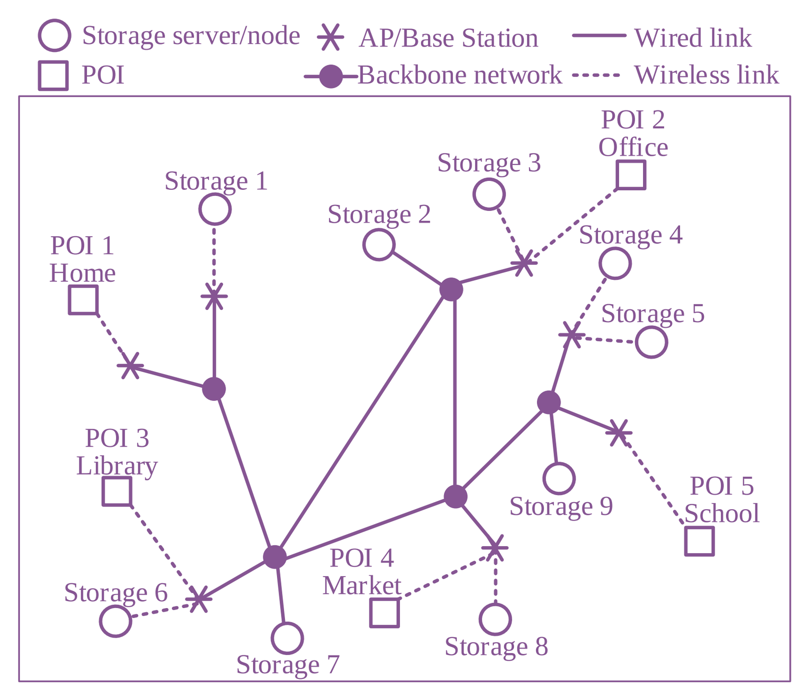

Fig. 4. A layout has 9 storage nodes and 5 places of interest (POIs). At any POI, a user can access network through one base station or AP. All base stations/APs are connected by a backbone network. All mobile devices and storage servers are connected to the nearest base station/AP.

Table 1. The context-aware constraints of 4 strategies. The CNT-OPT is our proposed approach.

Strategy abbr.	CNT-0	CNT-1	CNT-2	CNT-OPT
Network Condition	No	Yes	Yes	Yes
User Mobility Pattern	No	No	Yes	Yes
Data access preference	No	No	No	Yes

Performance Evaluation

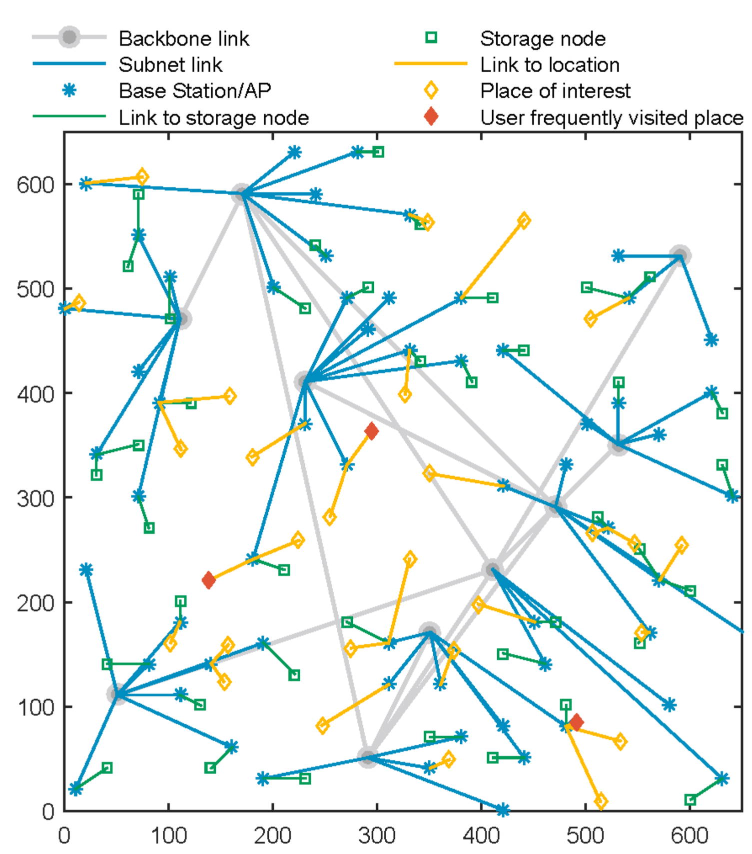

Fig. 5. A typical layout used in the simulation.

We randomly generate 100 different virtual layouts as Figure 5 for the purpose of simulation. We compared different strategies under different configurations of erasure code and context information:

- Figure 6 compares strategies by varying checksum chunks from 1 to 4.
- Figure 7 presents the performance under fixed ratio of checksum chunks to data chunks, while increasing the number of all chunks.
- Figure 8 shows the performance under different mobility patterns.

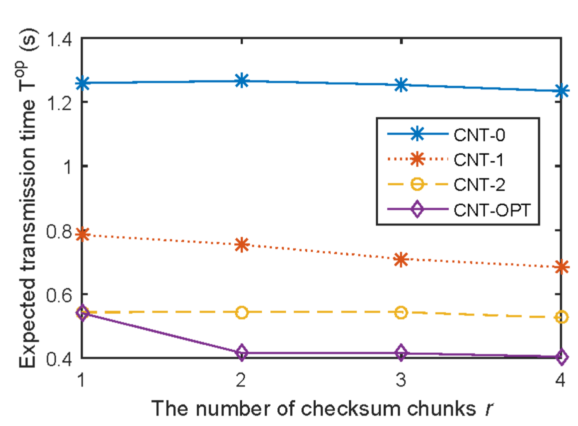

Fig. 6. The performance for different size of checksum chunks.

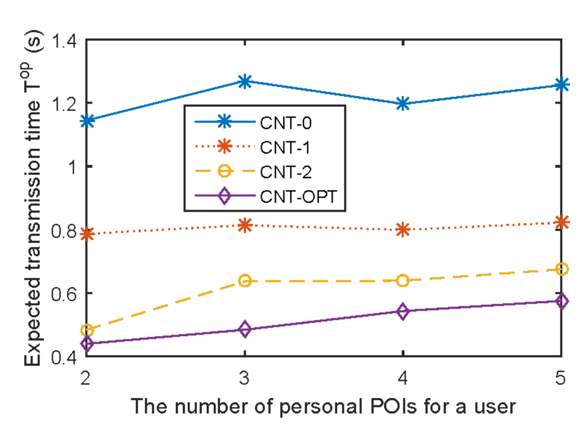

Fig. 7. The performance for the fixed redundancy ratio.

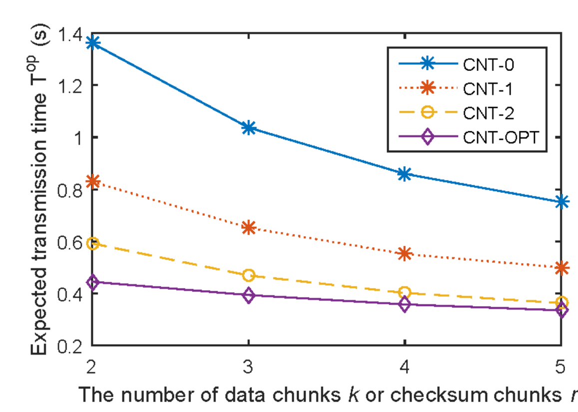

Fig. 8. The impact from the number of POIs.

Conclusions and Future Work

- The simulation results verified the efficiency of MILP as an optimal centralized strategy that improves the data access efficiency significantly in mobile cloud storage.
- In our future work, some other optimization methods based on distributed algorithms will be further studied, such as stable matching.

References

[1] J. S. Plank and Others, "A tutorial on Reed-Solomon coding for fault tolerance in RAID-like systems," Softw., Pract. Exper., Vol. 27, No. 9, pp. 995–1012, 1997.

[2] C. Zhou, N. Bhatnagar, S. Shekhar, and L. Terveen, "Mining personally important places from gps tracks," in Data Engineering Workshop, 2007 IEEE 23rd International Conference on. IEEE, 2007, pp. 517–526.