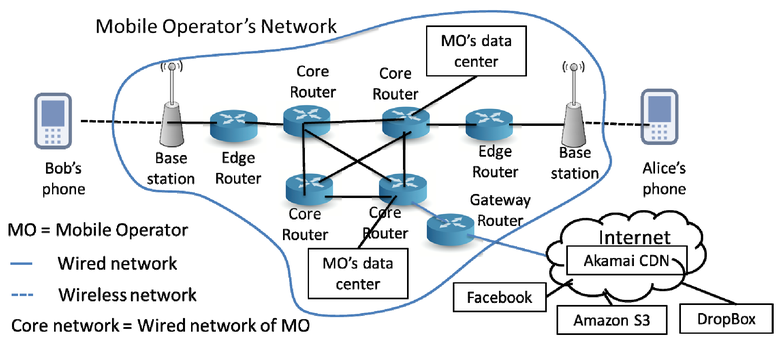
1st DFG/GENI Doctoral Consortium, San Juan, PR March 13th-15th, 2011

Infinity: In-Network Storage for Mobile Devices

Yudong Gao[1], Z. Morley Mao[1], Pradeep Padala[2], Ulas C. Kozat[2]
[1]University of Michigan, [2]DOCOMO USA Labs

Abstract

Data accessed on the mobile devices is exploding, but current mobile applications do a poor job in conserving energy while ensuring good performance, to satisfy the rapid increase in the frequency and volume of data access. The cloud services accessed by these applications consider neither the role of mobile operator's network nor the mobile device state, leading to poor application performance and wastage of network resources. Today's mobile operator (MO)'s networks are no longer "dumb pipes" but are connected to data centers with large amount of resources. We argue that this disruptive change makes MO's network increasingly resemble a cloud computing infrastructure. We propose a storage platform called **Infinity** that can be used by service providers to effectively exploit the mobile operator's network, while saving energy on the mobile devices.



MO's network resembles a cloud computing Infrastructure.

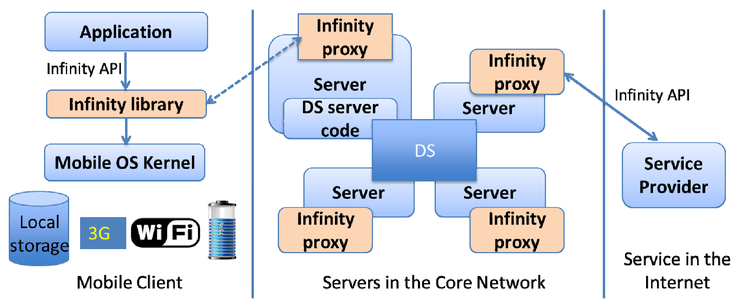
Research Objectives

Infinity's benefits are three-fold:

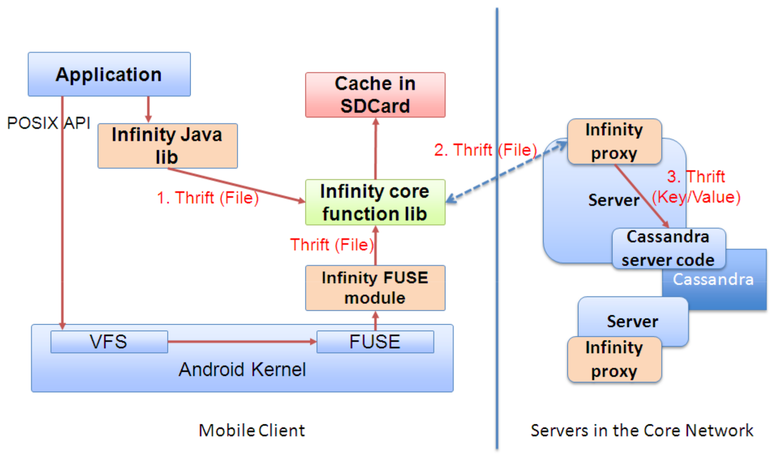
1. For **end-users**, we expect better battery life-time and performance;
2. For **service providers** such as Facebook and DropBox, they can readily use MO's infrastructure without developing their own storage infrastructure, gaining better performance and improved cost-effectiveness.
3. For **mobile operators**, the reduced traffic in MO's network can lower operational and capital costs.

Our key contribution is to develop a platform that solves the following challenges:

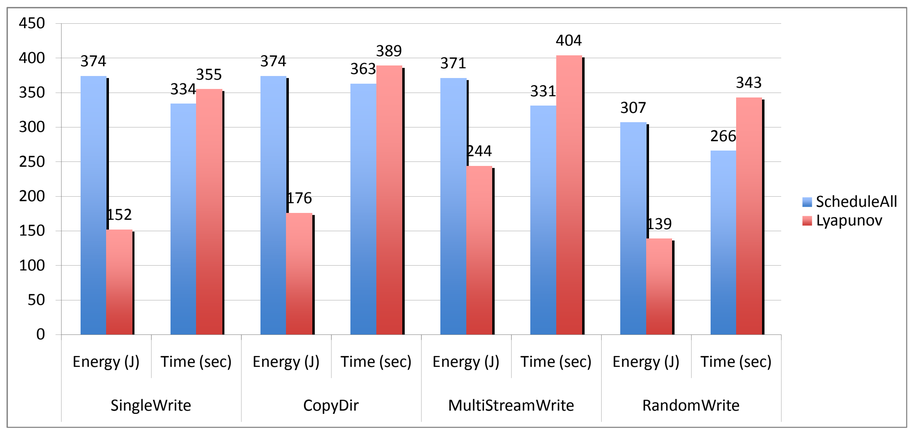
1) How to provide efficient access to seemingly infinite storage to mobile devices, while saving energy? (device efficiency).
2) How to build a storage platform that allows both users and service providers to exploit the core network's cloud resources distributed over wide-area network? (network efficiency).



Infinity's Architecture.



Client-side and back-end architecture.



Uploading scheduling results using 4 micro-benchmarks in FileBench. Infinity uses the Lypunov framework to schedule upload data, achieving 50% of energy saving while increasing the delay by 16% on average.

Infinity Architecture

The **client-side library** achieves device efficiency by scheduling data sending and receiving based on network and device conditions. Network scheduling decisions on downloading/pre-fetching and uploading data are based on monitored state of network interfaces (e.g., WiFi and 3G), local storage, and battery status. Caching in memory and on local storage is supported to handle disconnected operation with good performance. The library provides both Java APIs and file system interface.

The distributed storage hides implementation details of distribution and replication to provide scalable transparent infinite storage to mobile devices. Key-value storage is used for its simple get/put interface and built-in support for multi-datacenter deployment. We chose *Cassandra* since it is actively used in production systems.

Current State

We have a client-side prototype running on Android. Evaluation result show that its upload scheduling can achieve 50% of energy saving while increasing delay by less than 16%.

Our on-going work is on the distributed storage. The challenges and our solutions are:

1. How to support file system paradigm with object-level get/put interface? Our design is to map one block of a file to a single object to enable update and replication control at the per-block unit.

2. How to push data close to the user. Our design is a customized object replication strategy that controls the placement of an object when it is created, according to the user's access pattern.

Use of Glab/GENI Infrastructure

The back-end storage is designed to be deployed in multi-datacenter environment, which is simulated using the resources and emulation support provided by ProtoGENI.

We plan to deploy a location-aware object replication strategy, which replicates the objects to the server nodes close to the user according to the user's mobility, to reduce the access latency and core network traffic.

Current and Proposed Publications

Workshop paper submission to HotOS'11.