Report
An application oriented study of offloading while choosing between available Network (3G, 4G and WLAN)

Aditya Khune

October 20, 2015

Contents

Abstract

Offloading has been widely considered for saving Energy and increasing responsiveness of the mobile devices. We have surveyed various applications which are likely to benefit from Offloading as suggested by important publications. Various type of applications mentioned in the important publications are as follows: matrix calculations, natural language translators, speech recognizers, optical character recognizers, image processors, image search, online games, video processing and editing, Web-browsers, navigation, face recognition, augmented reality, etc. These applications consume large mobile battery, memory, and computational resources. Out of those we have listed out 5 applications for experimentations as follows:

1. Matrix Multiplication
2. Internet Browsers
3. Zipper
4. Voice Recognition
5. Torrents

We have done energy analysis and response time analysis of all the above smartphone applications, we have compared the results obtained with the help of available network 3G, 4G and WiFi. In the end of this report I have listed out my findings based on the results obtained with all the experimentations. To decrease the interference of the screen while doing energy analysis we run the applications with minimum brightness. Power consumption is measured by monsoon power analysis tool.

Smartphone handsets used:

- Samsung S3
- LG G3

Network:

- AT & T's 3G, 4G (HSPA+) Network
- Comcast's WiFi Network

Other Tools:

- Monsoon Power Measurement Tool
- Android Device Bridge (ADB)
- Amazon Web Services (AWS)
- AWS Command Line Interface (CLI)

Experimental Setup and Procedure for plotting the plots:

We have run each experiment 10 times on each handsets mentioned above, and then averaged out the readings obtained. The lower and higher limit of error bars used in our plots is what we obtained by averaging out the readings on the handsets, and what we are showing in the plots is one sample of the readings obtained. Although there isn's a particular standard used in our plots but generally the lower bar energy wise is for LG G3 and higher side of the error bars is for Samsung S3, and vice a versa for Response time plots. All the experiments are done using 3G, 4G and WiFi networks separately in order to understand the effect of choosing the right network while offloading the tasks and data onto cloud.

1 Matrix Operations

We have chosen an android app which does matrix operation because most of smartphone applications which include image processing need a processing of large matrices.

This application calculates values of an Inverse Matrix. Figure 1.1 we can see the battery consumption of smartphone increases manyfolds as the size of Matrix increases largely because there increase in CPU's energy consumption as number of floating point operations increase. This application calculates Matrix inverse using Adjoint Method. As we can see in the Figure 1.1 Offloading the processing for matrix calculation on Cloud saves energy as the matrix size increases, but for small matrix operations (i.e. 3X3 and 4X4) the local processing is suitable as it saves both energy and time.

Results

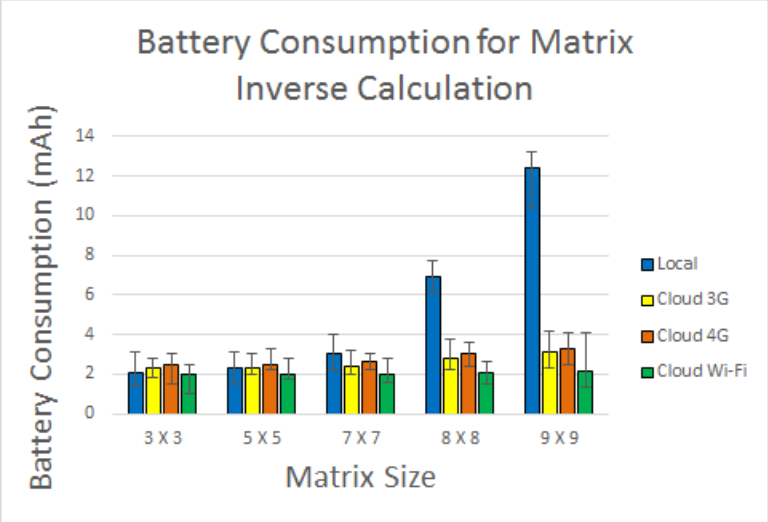

Figure 1.1: Battery Consumption for Matrix Inverse Calculation

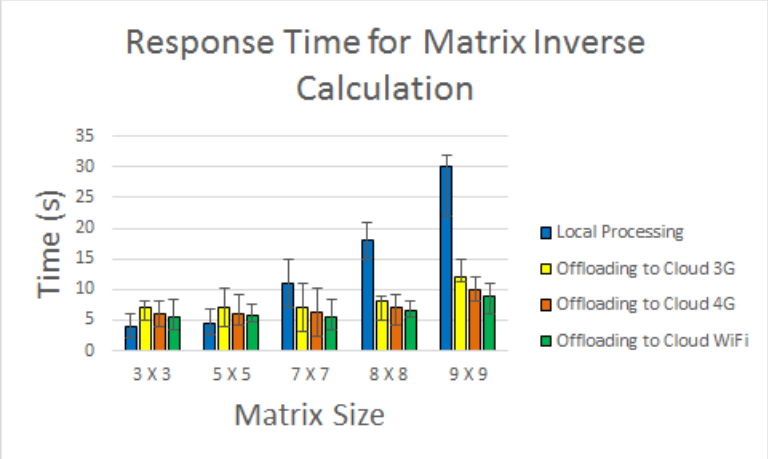

Figure 1.2: Response Time for Matrix Inverse Calculation

2 Internet Browsers

Cloud based Internet Browsers were introduced in order to overcome the processing and energy limitations of mobile devices. Already there are a number of cloud-based mobile web browsers that are available in the industry e.g. Amazon Silk [1], Opera Mini [2], Chrome beta [3] etc. Let us understand more about these browsers first. Cloud-based Web browsers([1], [3], [2], [4]) use a split architecture where processing of a Mobile web browser is offloaded to cloud partially, it involves cloud support for most browsing functionalities such as execution of JavaScript (JS), image transcoding and compression, parsing and rendering web pages. Prior research in this area such as [5] shows that CB does not provide clear benefits over Local or device-based browser (e.g. Local Processing) either in energy or download time. Offloading JS to the cloud is not always beneficial, especially when user interactivity is involved [5].

We have chosen one of the commercially available Cloud based mobile browser(puffin) and also a Local browser(Firefox) for our experiments. In Figure 2.1 and 2.2 we have plotted the smartphone readings that we have obtained by measuring data transfer and response time required by these browsers for following websites: 1. www.yahoo.com, 2. www.wikipedia.org, 3. www.amazon.com, 4. www.google.com, 5. www.facebook.com.

We have obtained our readings for a data range starting as low as 150 Kib to a session involving 5 MBs of data transfer to load the webpages. We have observed here that Cloud based web browsers are faster but expensive in terms of energy consumption. For small data transfers it is always suitable to use Local web browser to save both time and battery consumption. For a normal user overall data transfer during the browsing session does not go beyond 5-6 MBs for single session, which means we always will have small data transfers to the cloud and Local browsers show better results for those cases and that's why Cloud based web-browsers aren't very popular.

Results

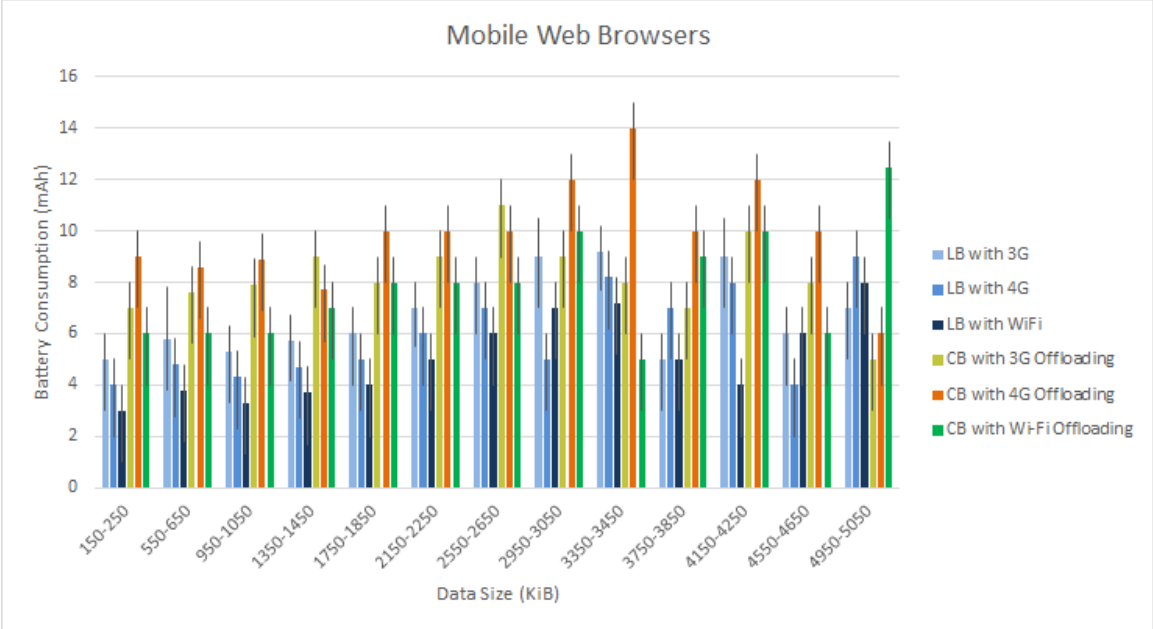

Figure 2.1: Battery Consumption for torrent downloading

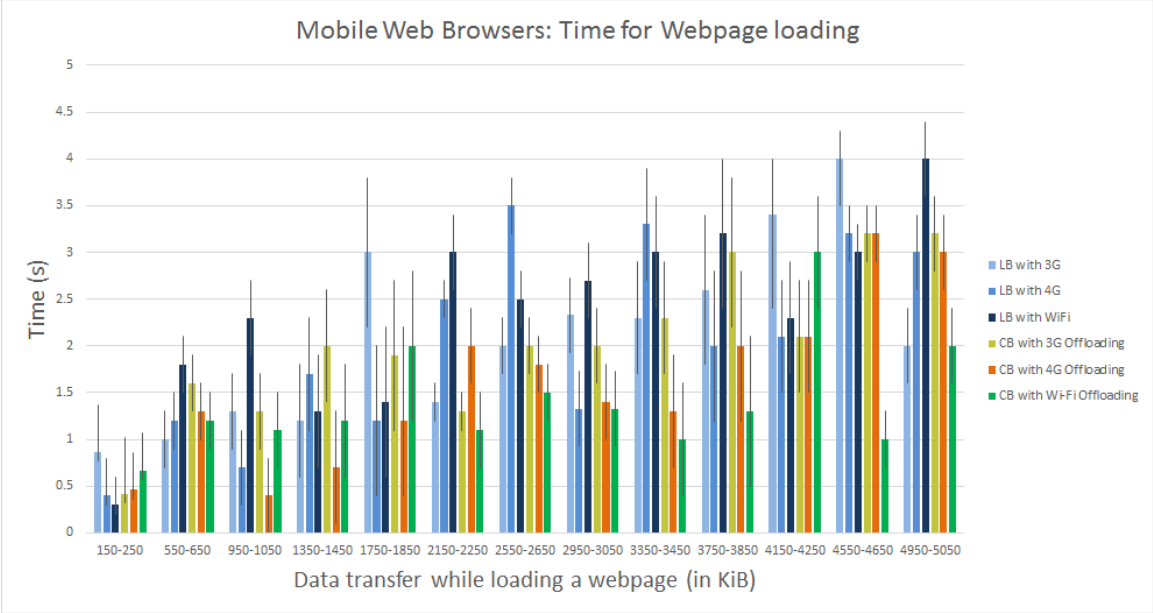

Figure 2.2: Response Time while torrent downloading

3 Zipper

Here the idea is the processing of zipping the files will be done either locally or on the cloud as directed by the Decision engines. The Zipper is an Android app that we used to compress the files locally. For Cloud based file comprssion we have used online zipping tools such as [6] and [7].

In figure 2.3 and figure 2.4 we have given a comparison of energy consumption and Response Time while doing Local Processing and Offloaded Processing with varying file sizes. For compressing files we have used pdf and word documents and also MP3 music files in equal size distribution.

3.1 Results

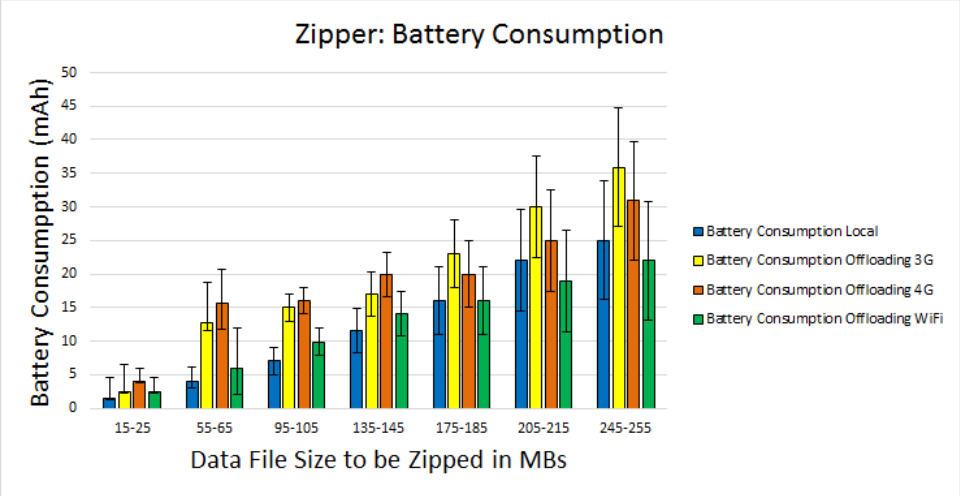

Figure 3.1: Battery Consumption by while file zipping

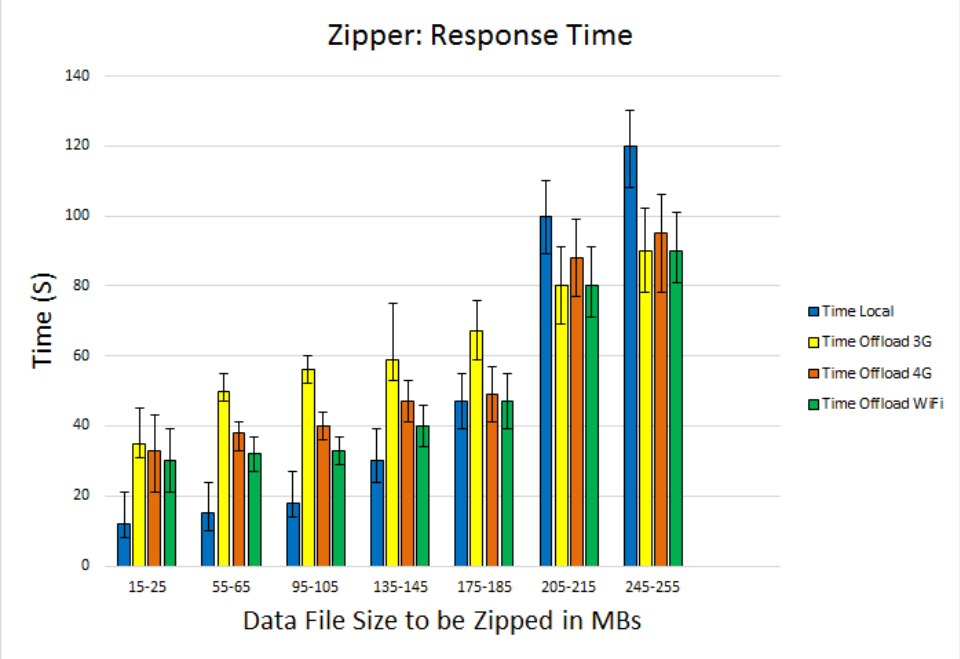

Figure 3.2: Response Time while file zipping

4 Voice Recognition and Translation App

Google translate is one of the app which uses cloud to do the voice recognition and translation. It also has an offline translation mode which does local processing on the device with small a Neural Network.

In figure 4.1 we can see the energy consumption of this app on our devices for a range of words. We have done our experimentations on our handsets using 3G, 4G and WiFi networks for recognizing and translating 20-140 words from English to Marathi translations.

Results

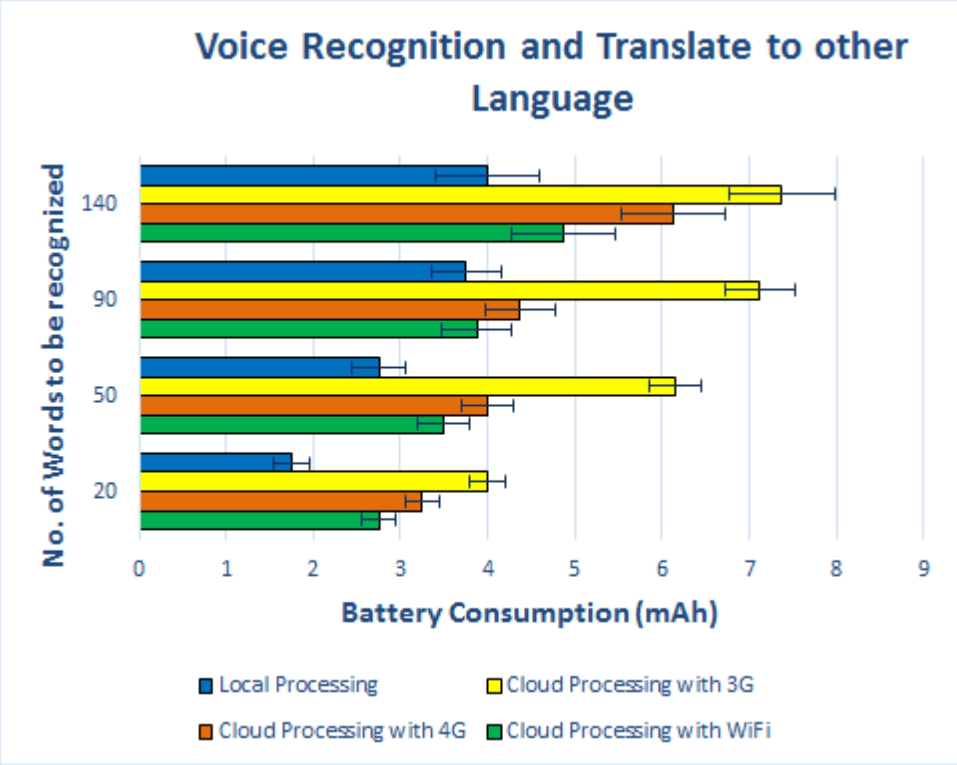

Figure 4.1: Battery Consumption while Voice Recognition and Translation App

5 Torrents

In this strategy the cloud servers are used as a BitTorrent client to download torrent pieces on behalf of a mobile handheld device. While the cloud server downloading the torrent pieces, the mobile handheld device switch to sleep mode until the cloud finishes the torrent processes and upload the torrent file in one shot to the handheld device. This strategy saves energy of smartphones because downloading torrent pieces from torrent peers consumes more energy than downloading a one burst of pieces from the cloud. Similar strategy is proposed by Kelenyi et al. in [8]

Results

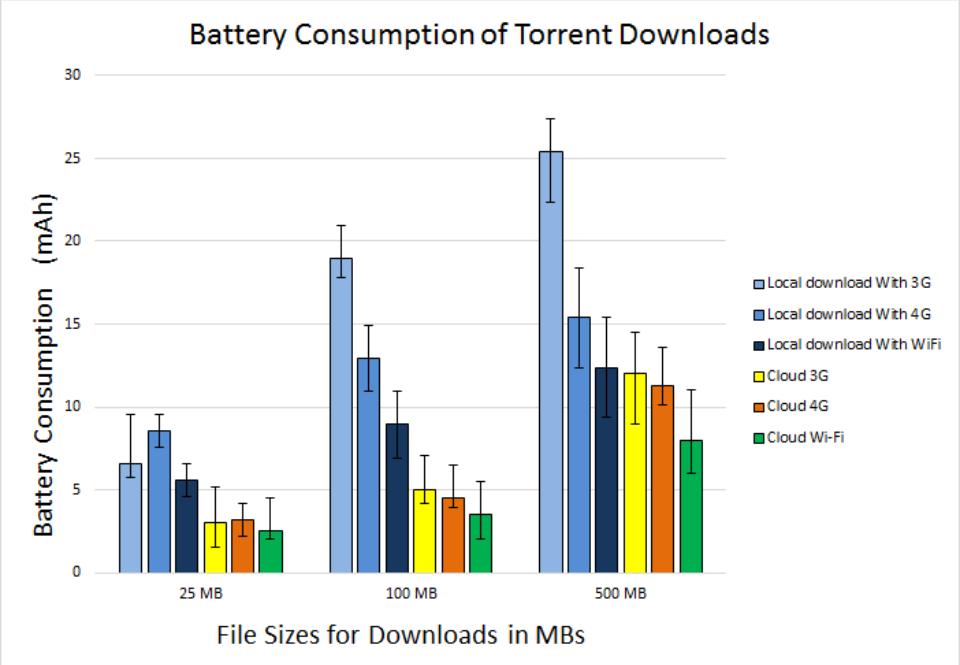

Figure 5.1: Battery Consumption for torrent downloading

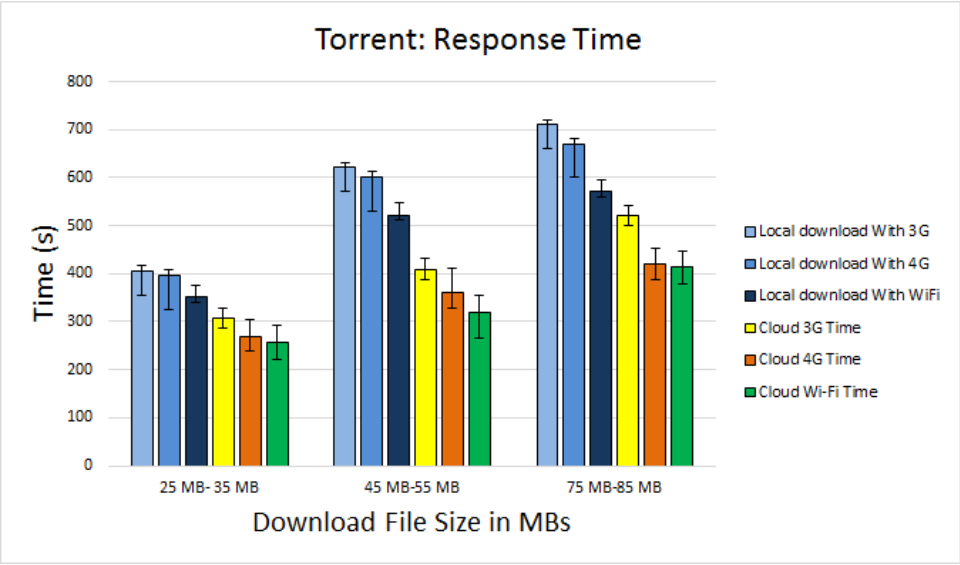

Figure 5.2: Response Time while torrent downloading

6 My Findings and Conclusion

Prior Related Works have shown that we can save energy by offloading the compute intensive tasks onto the cloud, the data transfer required must be low. But by careful observations of the results that we have obtained we have shown that what we should look for in an application is not only 'compute intensive' tasks but compute and 'data intensive tasks' in order to use offloading architecture more effectively. In fact our results show that for only data intensive applications like torrents (not much computing), the cloud computing can be even more useful. We think that the reason behind this is because most of the smartphones today are equipped with advanced processors like quad core CPUs which give them good capacity to deal with even compute intensive applications, but when it comes to data intensive computational tasks offloading to Cloud for computing seems a better option.

Bibliography

[1] "Amazon silk split browser architecture.." https://s3.amazonaws.com/awsdocs/AmazonSilk/latest/silk-dg.pdf.

[2] "Opera mini architecture and javascript.." http://dev.opera.com/articles/view/opera-mini-and-javascript/.

[3] "Data compression proxy in android chrome beta.." https://developers.google.com/chrome/mobile/docs/data-compression.

[4] X. S. Wang, H. Shen, and D. Wetherall, "Accelerating the mobile web with selective offloading," in *Proceedings of the second ACM SIGCOMM workshop on Mobile cloud computing*, pp. 45–50, ACM, 2013.

[5] A. Sivakumar, V. Gopalakrishnan, S. Lee, S. Rao, S. Sen, and O. Spatscheck, "Cloud is not a silver bullet: A case study of cloud-based mobile browsing," in *Proceedings of the 15th Workshop on Mobile Computing Systems and Applications*, p. 21, ACM, 2014.

[6] "ezyzip: The simple online zip tool." http://www.ezyzip.com/.

[7] "Online-conver.com." http://archive.online-convert.com/convert-to-zip.

[8] I. Kelényi and J. K. Nurminen, "Cloudtorrent-energy-efficient bittorrent content sharing for mobile devices via cloud services," in *Proceedings of the 7th IEEE on Consumer Communications and Networking Conference (CCNC)*, vol. 1, 2010.