

Mobile Virtual Storage using iATA Protocol with Disk Image Based Loop Device

Hee-Yuan Tan*, Bee-Lie Chai*, Hyotaek Lim**, Hoon-Jae Lee**

**Graduate School of Design and IT, Dongseo University, Sasang-Gu, Busan, 617-716 Korea*

***Div. of Computer & Information Engineering, Dongseo University, Sasang-Gu, Busan, 617-716 Korea*

heeyuan@dit.dongseo.ac.kr, beelie@dit.dongseo.ac.kr, htlim@dongseo.ac.kr, hjlee@dongseo.ac.kr

Abstract— With continuous growth in mobile technology and ubiquitous computing, the demand for more storage in mobile devices is also on the rise. Internet Advanced Technology Attachment (iATA), a virtual storage system that positioned itself as one of the feasible solutions to solve the mobile device's storage limitation problem. In existing iATA implementation, the storage space depends on the unused partition space on server and it is predefined in server, and does not provide any varying storage control, causing an inelastic storage management. In this paper, we proposed an improved solution to iATA mounting on disk image based loop device, alongside with designing a storage management mechanism for iATA system that allow us to increase the flexibility and variability on this virtual storage system.

Keywords— Virtual storage, iATA protocol, disk image, loop device, storage management

I. INTRODUCTION

In ubiquitous computing environment, mobile technology has played an important role in strengthening the mobility in collaboration, communication and information sharing. With growing popularity of mobile devices, there are more and more applications built for these devices in mind. There are numerous examples of existing mobile applications, such as SlingPlayer Mobile, Opera Mini, Google Maps for Mobile and some other mobile application which are widely being used by public. Increasing use of small portable devices and wireless network; enlarged mobile computation development; and increment of content; these are some of the factors that trigger the enormous increment in demand for mobile data storage to support users in accomplishing their task.

There are various means of storage for mobile devices, such as local storage and network storage [1]. However, most of the local storage mobile devices have limitation on storage space [2]. Take for example a Personal Digital Assistant (PDA); it allows user to access information from anywhere and anytime, but it might only have a few hundred megabytes of internal storage space or with a few gigabytes of memory cards attached to it. The storage limitation is a compromise between portability of the device against complete feature set found in regular computer [3]. One of

the ways to deal with data storage limitation on local storage is by using network storage, such as iATA virtual storage system.

Currently, the iATA disk or partition that is ready to be mounted is predetermined on server side via dedicated hard disk partition. For most users, the hard disk has been pre-partitioned to maximise all of the available capacity, leaving minimal or no unallocated capacity at all. User does not have the control to manage the disk or partition itself even there is plenty of storage space available in another partition. Deploying iATA solution in such scenario will require user to repartition their hard drive or installing another hard drive to run the existing iATA solution.

This paper contributes toward designing and implementing a disk management for iATA to enhance flexibility and usability of iATA via the introduction of disk image as a loop device. In this paper, an enhancement will be provided which enables users to determine on how much disk space is required, giving them the flexibility to name the created space, which can be useful to categorise them by different users, tasks, contents or purposes.

The following paper is organized as follow: Section II will give a brief overview of iATA protocol, while Section III will focus on description of proposed solution. Section IV will outline the implementation carried out. Performance evaluation of the proposed solution with comparisons to the existing implementation using dedicated hard disk storage will be show in Section V before we conclude on this paper in Section VI.

II. BACKGROUND

Advanced Technology Attachment (ATA) hard disk is one of the most popular storage choices in lower end system such as those targeting home user market or small and medium sized enterprises. Together with Internet Advanced Technology Attachment (iATA) protocol as a block level network storage protocol, allowing ATA device commands transported over the TCP/IP network, users can mount a remote disk space from a wireless connected mobile device client [4]. It provides users with a low cost, simple, and ubiquitous storage facility. iATA protocol has its inspiration from iSCSI and the design of the protocol mimics those of AOE [4][5]. Unlike iSCSI however, performance of iATA relies on Number of Sector per Command (NSPC) parameter [6].

iATA uses a client/server model, where the underlying iATA sits on the application layer of the TCP/IP protocol stack. Using reliable data transport of TCP, the iATA protocol data unit (PDU) consists of 2 main parts as shown in Figure 1. The first part contains a 128-bit Common Header Segment (CHS) which indicate the size and the type of data of the subsequent part of PDU [4]. The second part of PDU can either consists of a configuration/query message or the actual information on ATA command block. A configuration/query message functions as an exchange of parameters during beginning and the end of an iATA session, including login, discover and termination phases. The ATA command block on the other hand handles the data transfer (full feature phase) between client and server that contains an ATA command, 6 bytes Logical Block Address (LBA) and the actual data itself [5].

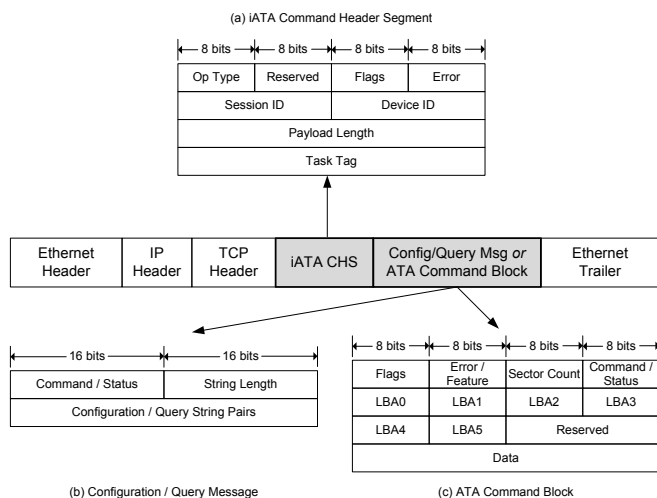


Figure 1. iATA Protocol Data Unit (PDU)

The mounted disk drive from the remote server looks as if it is a local storage in mobile devices. If the server is ready to accept remote connection, all the user needs to do is to mount the remote disk residing on the server to the mobile device. Users can freely copy, move and delete files inside the mounted disk, just like any regular file operation found in a typical system. The file operation is done via TCP/IP communication between the mobile device and the server. All these operations are transparent to the user.

Existing implementation of iATA at server side mounts an actual physical hard disk or a pre-determined partition as the option for virtual storage. An initial study has been carried out on the idea of partitioning the iATA disk as well as basic authentication by extending on the existing implementation of iATA [7].

III. PROPOSED ENHANCEMENT

The existing iATA implementation can be further enhanced to improve on the usability and management of the virtual storage. We will look into details on the specific implementation technique that can provide a complement to

the existing iATA implementation. Instead of mounting on an unused partition or hard disk drive, we proposed using disk image that can be mounted as a loop device.

Using disk image has several advantages compared to using disk partition directly. Firstly, using disk image, we can use any available space in the hard disk, without needing to create a dedicated partition from unallocated or unused portion of a hard disk. Secondly, backup and restore of disk image is easier compared to backing up or restore the whole partition. Thirdly, using a disk image, the underlying file system of the disk image need not be the same as the file system used in the host system. This is especially useful, if we need to support multiple different file systems that cater for different type of devices.

One extended benefit from using available disk space instead of dedicated partition is we can create and move around the disk image without limiting to a pre-determined location and storage space that we required. By using disk image, we have added disk image management enhancement into our proposed solution. We have identified four basic operations that needed to be performed in our proposed enhancement, namely listing of available disk image to be mounted, creating of new disk image, removing of existing disk image and activate available disk image. Figure 2 depicts the overall operation of how our proposed disk image management is going to work.

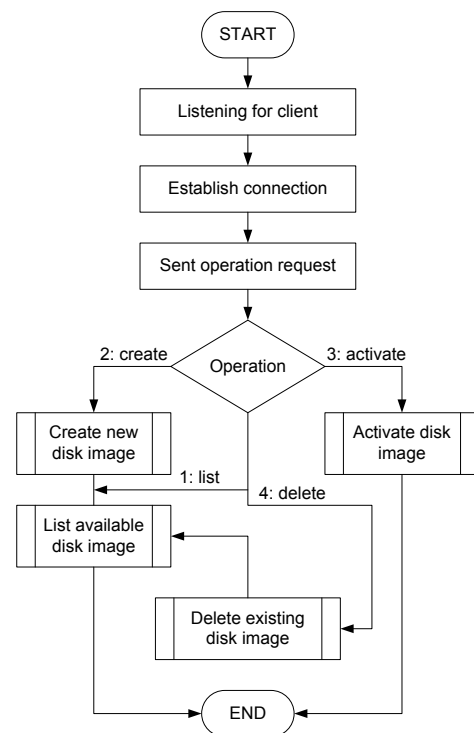


Figure 2. Overall operation process flow

Managing the disk image residing on the server involves establishing connection to the server that is waiting for operations from client. User authentication can be added if security requirement is needed. Once a connection is established, any of the above mentioned disk management operations can be sent alongside with parameters or arguments that go along with the operation.

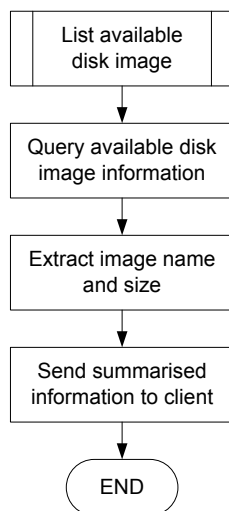


Figure 3. Disk image listing process flow

Figure 3 shows the list operation flow. Listing of disk image queries all the disk images that is available to be used. Operation listing will be carried out after a creation or a deletion of disk image. It can also be run independently for showing available disk image ready to be mounted. Useful information such as disk image name and total defined size can be gathered by querying directly from the server and information is then formatted based on requirement.

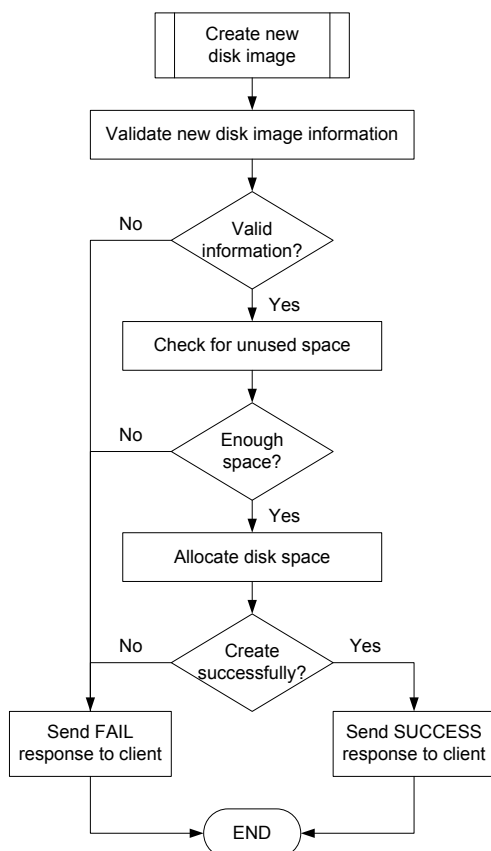


Figure 4. Disk image creation process flow

A new disk image can be created using the flow illustrated in Figure 4. All parameters or arguments of disk image creation need to be validated before creation, as part of error handling procedure. Once validation has been carried out, a check for unused disk space in server will be performed and if the required disk space is available, storage space will be allocated for the new disk image. A response on the operation is then returned for subsequent process to be executed.

A corresponding operation to disk image is removal of disk image. As shown in Figure 5, operation of removal or deletion of disk image need a prior validation, just as how creation of disk image is carried out, as part of a error handling procedure. What is different from disk image creation however, a removal of disk image operation requires an extra checking of whether the requested or the targeted disk image pending for deletion has been activated or not.

If the requested disk image has been activated, the targeted disk image is first deactivated from the available disk ready to be mounted. This ensures that the existing iATA listener does not mount to a non-existence or deleted disk image, raising an error if the listener were to attempt to do so. It is one of the measures to ensure that the iATA is always in a stable state and does not create extra unexpected error handling to iATA listener.

From the listing of available disk image operation, user can pick which disk image to be activated for iATA listener to accept disk mounting request. Figure 6 outlines the process flow of such disk image activation. As the same with the other processes mentioned earlier, the chosen disk image parameters or arguments for activation are validated first before proceeding. If the selected disk image is yet to be activated, the server process will need to locate available and unused listening port number from the iATA listener itself. Once that is done, the iATA listener will be ready to wait for any client wishing to connect to the activated disk image.

The activation process can also be tweaked to support for switching between disk images. Instead of validating just the targeted disk image, the process would require to validate both the disk images before proceeding. For iATA setup that only allows a single disk image to be in active listening mode in any one time, the being-replaced disk image needs to be check from the iATA listener whether such disk image has been activated, and deactivates it to give way to the newly selected disk image to take over.

With these four operations in place, user residing on the client side can now have more flexibility to set the required disk image to be used and the intended storage space for the disk image. Users can have a better way to name and maintain the disk image according to their preference.

IV. EXPERIMENTAL IMPLEMENTATION

This section explains the implementation of the proposed solution using disk image to replace the existing disk partition approach. The relevant components and hardware that go into setting up the server and the client mobile device for experiment are explained, together with implementation considerations that go into it are discussed as well.

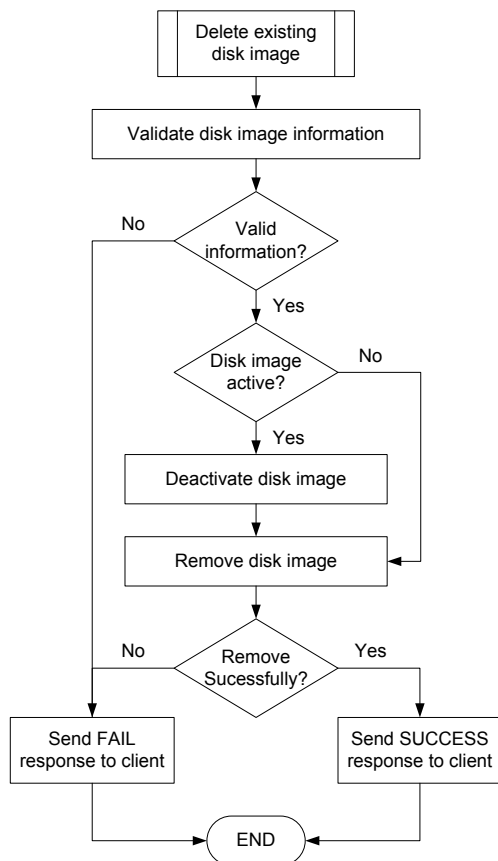


Figure 5. Disk image removal process flow

A. Environment Setup

An environment as shown in Figure 7 has been set up to test out our implementation of iATA enhancement, which we named it iATA Disk Manager. A couple of hardware devices are needed in our implementation such as Personal Computer (PC), Personal Data Assistant (PDA) and so on. A PDA (HP iPAQ hx2790b) running on Microsoft Windows Mobile 5 with .NET Compact Framework 3.5 installed, is used as the main mobile client for testing.

While on the server side, an Intel Pentium 4 2.0 GHz PC with 768 GB of RAM is used as our iATA server and is connected to Internet through TCP/IP network. iATA server is running on CentOS 5 Linux with Kernel 2.6.18. The iATA listener is not enabled when the system start up, in order to activate it via our mobile device.

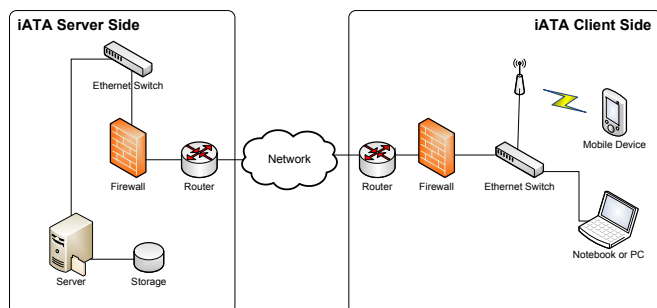


Figure 7. Experiment environment Setup

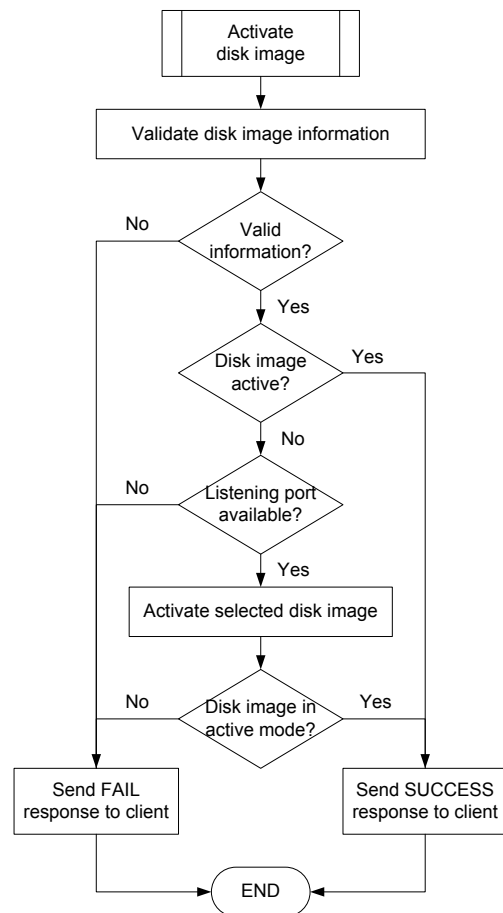


Figure 6. Disk image activation process flow

B. Implementation Consideration

Mobile devices have less computing resources compared to normal computer. In our implementation, we have factored in compact and lightweight consideration that requires less computation and processing. Request and response message are kept to the minimal with only the necessary data and parameters are included.

Target mobile devices in our environment supports different character sets from languages other than Latin script based languages. Both server and client have been added with Unicode support to allow user to specify non English disk image name.

Deployment of proposed solution leverages as many in-built command provided by Linux as possible for efficiency and compatibility reasons, and at the same time making it easily setup and maintain in various distribution of Linux, if similar deployment is installed on another machine.

C. Interface Design

The Graphical User Interfaces (GUI) in mobile device as shown in Figure 8 are developed using VB.NET via Microsoft Visual Studio 2008. The client interface implemented has all four operations mentioned in previous section together with the ability to do switching between disk images. File size in kilobytes are also shown along side with the name of disk image.

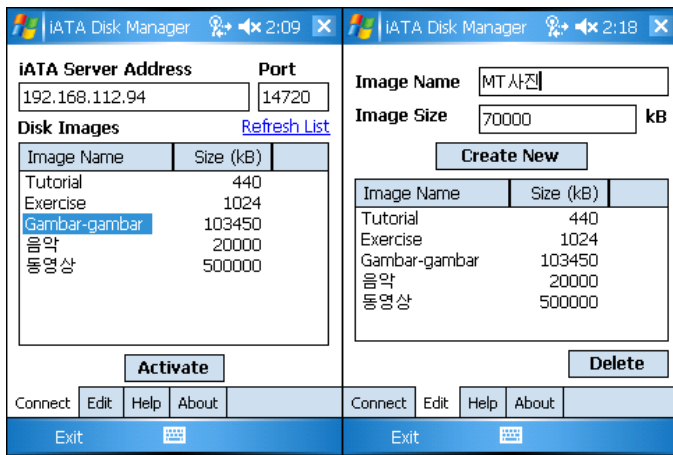


Figure 8. Graphical user interface

D. Testing

Within our test environment, we have tested the four operations deployed using Latin characters and Korean characters. Also in our test case, we have tested switching between disk images. The test cases are also split into two sets where one set running only the iATA Disk Manager and tested independently, while the other includes the original iATA listener which handles the actual disk mounting process.

The iATA listener can mount to the required disk image in all our test cases. Except for some slight waiting time required when switching between disk images, the remaining process performs within expectation.

Another set of tests was conducted to evaluate the performance of data transfer for our proposed disk image based loop device in iATA will be explained in detail in the next section.

V. PERFORMANCE ANALYSIS

Performance of disk image based iATA was evaluated using a 2GB VFAT disk image. Various tests were conducted using Pocket Mechanic Professional version 2.90 storage benchmarking tool. Tests were all conducted using a PDA that was connected through a WiFi access point. Performance comparison between disk image based implementation against disk partition implementation will be discussed at the end of this section.

A. iATA Data Reading Result

Ten different files with size ranging from 4KB to 2MB were created by the tools to test on the throughput using both buffered read and unbuffered read. Figure 9 shows the unbuffered file system read test. Throughput for unbuffered read peaked for file size of 32KB and drop substantially for file size that is larger thereafter.

Figure 10 show a summarized result for a buffered read. With buffer, the file system read has gained significant throughput. In buffered read, the maximum throughput is obtained when a 64KB file was read. As file size is getting larger, the throughput has dropped significantly.

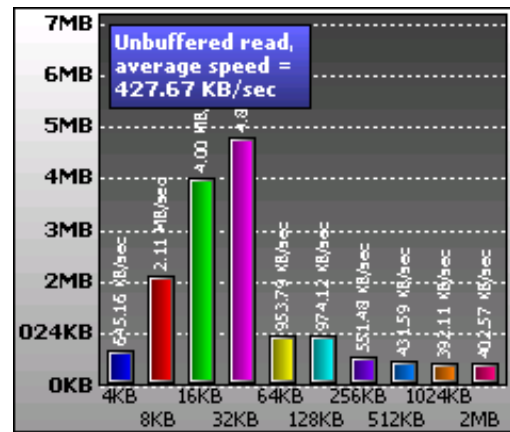


Figure 9. Unbuffered file system read

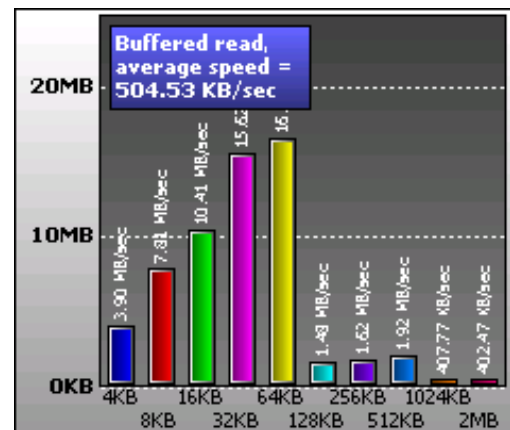


Figure 10. Buffered file system read

B. iATA Data Write Result

Test for file system write for both buffered and unbuffered was carried out using the same file range of 2KB to 2MB. Figure 11 shows the unbuffered write performance, where the throughput of varying file sizes falls in the range of 200KB/s and 400KB/s, with the peak occurs at file size of 16KB and a low at 4KB.

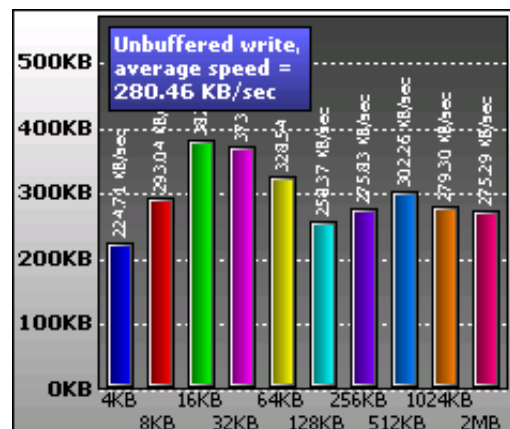


Figure 11. Unbuffered file system write

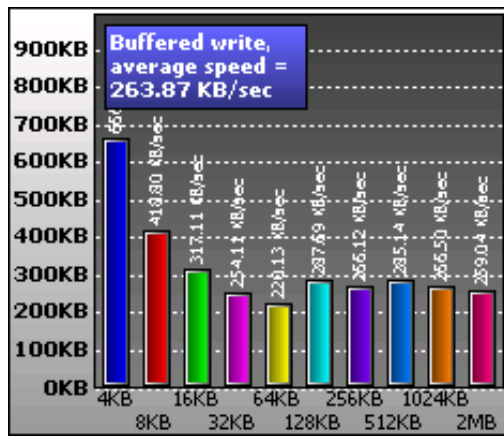


Figure 12. Buffered file system write

Figure 12 shows the buffered write for ten different file sizes. Except for files in 4KB and 8KB, the remaining file sizes show throughput of similar magnitude. The highest throughput occurs for file in 4KB giving one-third faster result from the closes 8KB file, and approximately double for the remaining file sizes.

C. Discussion

The same test was also repeated on existing disk partition implementation to measure the throughput achieved. Table 1 summarises the results obtained from previous disk image test and contrast it with disk partition test. Implementation using disk image gave comparable performance to a disk partition implementation. The file system general test shown in the last row of Table 1 measure the number of file system operation that can be executed per second.

Table 1. Performance result comparison

Test	Disk Image	Disk Partition
File system read test (unbuffered)	427.67 KB/s	426.30 KB/s
File system read test (buffered)	504.63 KB/s	520.11 KB/s
File system write test (unbuffered)	280.46 KB/s	267.27 KB/s
File system write test (buffered)	263.87 KB/s	233.08 KB/s
File system general test	31.78 op./s	30.67 op./s

VI. CONCLUSIONS

In this paper, we have proposed an enhancement to the existing iATA protocol implementation by using disk image on loop device instead of using disk partition. Mounting loop device has provided another level of abstraction to the existing iATA virtual storage system. This additional level of abstraction not only did not incur performance degradation in our experimental analysis, it has opened up opportunity for us to include disk management that is absent in existing iATA implementation.

We have identified four main operations for the proposed disk management enhancement, namely, listing, creation, removal and activation of disk image. The main advantage in this enhancement is users can decide on the total storage space needed for each disk image and allowing them to use more than just the single predefined storage space. Also, user can now deploy iATA virtual storage solution using available free space in their hard disk storage instead of repartitioning hard disk for dedicated partition support.

There are several future works which we have identified to further improve usability of our proposed solution. We plan to add user based disk management, in which individual user will have a different view from the other user. Also, with login credential in place, we plan to develop user level or group based disk sharing feature analogous to those file sharing features found in major operating systems with access control list.

REFERENCES

- [1] J. J. Barton, S. Zhai, and S. B. Cousins, "Mobile Phones Will Become The Primary Personal Computing Devices," in *Proc. IEEE WMCSA '06*, 2006, pp. 3-9.
- [2] B. Marsh, F. Douglass, and R. Cáceres, "Systems Issues in Mobile Computing", Technical Report MITL-TR-50-93, Matsushita Information Technology Laboratory, Princeton, NJ, 1993.
- [3] J. S. Pierce, H. E. Mahaney, "Opportunistic Annexing for Handheld Devices: Opportunities and Challenges," in *Proc. HCIC*, 2003.
- [4] C. M. Yeoh, Y. S. They, H. J. Lee, and H. Lim, "Design and Implementation of iATA on Windows CE Platform: An ATA-Based Virtual Storage System," in *Proc. WRI CMC '09*, 2009, pp. 85-89.
- [5] Y. S. They, C. M. Yeoh, H. J. Lee, and H. Lim, "Design and Implementation of ATA-based Virtual Storage System for Mobile Device," in *Proc. MUE '08*, 2008, pp. 490-495.
- [6] C. M. Yeoh, H. Lim, "Design and Parameter Optimization of Virtual Storage Protocol (iATA) for Mobile Devices," *Journal of the Korea Institute of Maritime Information & Communication Sciences*, vol. 13, No. 2 pp. 267-276, Feb. 2009.
- [7] S. Park, J. Chun, and H. Lim, "A study on Partition Allocation Techniques of iATA-based Virtual Storage," in *Proc. KIMICS Conference Spring '09*, 2009, pp. 68-71.