



THE JOURNAL OF  
**DIGITAL FORENSICS,  
SECURITY AND LAW**

Journal of Digital Forensics,  
Security and Law

Volume 9 | Number 2

Article 12


2014

## Exploring Forensic Implications of the Fusion Drive

Shruti Gupta  
*Purdue University*

Marcus Rogers  
*Purdue University, rogersmk@purdue.edu*

Follow this and additional works at: <https://commons.erau.edu/jdfsl>

 Part of the [Computer Engineering Commons](#), [Computer Law Commons](#), [Electrical and Computer Engineering Commons](#), [Forensic Science and Technology Commons](#), and the [Information Security Commons](#)

### Recommended Citation

Gupta, Shruti and Rogers, Marcus (2014) "Exploring Forensic Implications of the Fusion Drive," *Journal of Digital Forensics, Security and Law*: Vol. 9 : No. 2 , Article 12.

DOI: <https://doi.org/10.15394/jdfsl.2014.1177>

Available at: <https://commons.erau.edu/jdfsl/vol9/iss2/12>

This Article is brought to you for free and open access by the Journals at Scholarly Commons. It has been accepted for inclusion in Journal of Digital Forensics, Security and Law by an authorized administrator of Scholarly Commons. For more information, please contact [commons@erau.edu](mailto:commons@erau.edu).



(c)ADFSL





This work is licensed under a Creative Commons Attribution 4.0 International License.

# EXPLORING FORENSIC IMPLICATIONS OF THE FUSION DRIVE

Shruti Gupta and Marcus Rogers

Purdue University

401 N. Grant St., IN 47906, Unites States

{shruti,rogersmk}@purdue.edu

## ABSTRACT

This paper explores the forensic implications of Apple's Fusion Drive. The Fusion Drive is an example of auto-tiered storage. It uses a combination of a flash drive and a magnetic drive. Data is moved between the drives automatically to maximize system performance. This is different from traditional caches because data is moved and not simply copied. The research included understanding the drive structure, populating the drive, and then accessing data in a controlled setting to observe data migration strategies. It was observed that all the data is first written to the flash drive with 4 GB of free space always maintained. If data on the magnetic drive is frequently accessed, it is promoted to the flash drive while demoting other information. Data is moved at a block-level and not a file-level. The Fusion Drive didn't alter the timestamps of files with data migration.

**Keywords:** file system, forensics, fusion drive, mac, digital forensics, computer forensics.

## 1. INTRODUCTION

A digital investigation can potentially involve any kind of operating system or file system. Mac computers from Apple are gaining prominence. As of 2012, they were the 3rd largest manufactured personal computers (AppleInsider, 2013). In the United States, the Mac OS X operating system represents 10% of the market share and globally it represents 7% (NetMarketShare, 2013). Digital investigations are based on many technical and non-technical judgments. If the environmental factors are different from the normal scenarios then it might lead to errors in judgment (Peron & Legary, 2005). Thus, it is important for the investigating officers to be well informed about systems that are different from the norm even in small ways. Therefore, even with PCs leading the market, there is a need to be technically sound with the functionality of Mac OS X operating systems to be able to retrieve all information from it.

Apple Inc. introduced the Fusion Drive in December 2012 with their Mac-mini and iMac models (Hutchinson, 2012). The Fusion Drive consists of a flash drive and a magnetic drive, which appears to be one single logical drive to the end-user. This can have many forensic implications related to the structure of the drive, the location of the files and the timestamps on the files. This paper explores the challenges that might be presented to forensic investigators as they face the Fusion Drive (FD). The paper first gives a background about the technical specifications of the fusion drive. It then explores other research studies that have similar goals. The next section describes the nature of the methodology that was used. The paper concludes with observations seen from the study and the conclusions that can be drawn from it.

## 2. BACKGROUND

In 2010, Apple filed a patent (Bazzani, 2010) describing a hybrid drive technology that online sources suggest implements the Fusion Drive (Purcher, 2012). The patent describes a storage device that combines flash memory with a magnetic drive. It also mentions that the address space might be dynamically allocated based on the nature of the usage activity on the drive and the nature of the applications being used. However, this is where the similarity with the FD ends. The patent claims that data storage is mainly allocated based on the environmental state of the drive. Example of change in the environmental state includes change in temperature, vibration of the drive or acceleration of the drive. This hasn't been mentioned in any literature related to the Fusion Drive. The patent also mentions that all data is written to the hard disk drive (HDD) first and then if there is a change in the state of the HDD, data can be written to the flash memory. This appears to be contrary to the functioning of the FD, which seems to write all data to the solid-state drive (SSD) first. Thus, while there are similar features, the authors do not believe that the patent filed in 2010 explicitly describes the Fusion Drive.

The Fusion Drive essentially implements the concept of auto-tiered storage, which is not new by any means. Tiered storage simply implies that data is hierarchically sorted by the system according to predetermined factors, to maximize system performance (Duplessie, 2004). In auto tiering, the user is not involved in the storage decisions. The concept of hierarchical storage management has always existed in the mainframe world. While it is still commonly used in the mainframe world, the concept of tiered storage is still new in personal computers (Duplessie, 2004).

However, tiering should not be confused with caching. Tiering and caching are similar concepts that have roots in the principle of locality of reference. It involves identifying a neighborhood in which a particular object operates, and optimizing the system performance within that neighborhood by manipulating the computation

of the system (Denning, 2005). However, unlike traditional SSD architectures, the Fusion Drive does not use the concept of caching (Shimpi, 2013). The total capacity of the FD is the combined sum of the capacity of the hard drive and the flash drive. This is the main difference between the FD and existing technologies like Intel's Smart Response Technology (SRT). Caching mechanisms like SRT algorithmically determine what things should be mirrored up from the HDD onto the SSD. In this caching scenario, the default location of the data is always the HDD with the SSD being used as a write cache. In caching, the main interaction is done with the hard drive and the solid-state drive is used to enhance performance.

In the Fusion Drive, the solid-state drive is the device that the user mainly interfaces and the magnetic drive is used to add to the storage capacity of the solidstate drive (Hutchinson, 2012). Also, while SSD caches mostly allocate data based on the frequency of read access, the FD appears to take write accesses into consideration while allocating storage for data. To sum it up, the technology at a higher level is very similar to other Original Equipment Manufacturers (OEM) that use hybrid drives in their PCs. The main difference lies in the capacity of the flash memory and the underlying software layer (Shimpi, 2013).

The Fusion Drive is available with capacities of 1 TB and 3 TB. The Mac mini or the iMac can have the 1 TB drive while the 3 TB can only be seen on the 27-inch iMac. As mentioned earlier, the FD combines a solid-state disk (128GB Samsung PM830) and a traditional hard disk (2.5 or 3.5) using the Core Storage technology. Core Storage is a logical volume manager that allows the operating system to treat multiple physical disks as a single volume and it was first seen in the Lion operating system (Shimpi, 2013). It aims to combine the speed of the flash memory with the capacity of the magnetic drive. Four main Core Storage calls are employed for the data-migration according to the activity shown by the command `fs_usage`. These are *RdChunkCS*, *WrChunkCS*, *RdBgMigrCs*, and *WrBgMigCS*. These calls are

made sequentially and form the crux of the auto-tiering happening on the drive. Data is moved in blocks of 128 KB between the drives. While the data structures that store usage data are not known, the drive is not using Hot File Clustering seen on earlier Macs that use solid state drives (Hutchinson, 2012). Later sections discuss more about the granularity of the data being moved.

The documentation provided by Apple mentions that only one additional partition could be added to the FD. This partition would be added to the HDD and it won't have any fusion capabilities. There is no way to add partitions on the SSD (*Apple Support*, 2013). The Recovery partition, which can be seen on operating systems succeeding OS X 10.7, is contained in the HDD. This has implications that the system can still boot in case the flash memory crashes.

### 3. PREVIOUS RESEARCH

The broad scope of the paper is to examine the forensic implications of auto-tiered storage that uses different types of storage devices. As mentioned earlier, auto-tiered storage has been commonly used in mainframes since the last decade but there isn't much literature that deals specifically with the forensic significance of auto-tiered storage or technical specifications of Fusion Drive operation. For lack of more related literature, the literature review addresses research studies that discuss the forensic implications of a new technology.

Beebe, Stacy, and Stuckey (2009) examined the forensic implications of the ZFS file system by Sun Microsystems. Just like the Fusion Drive by Apple, ZFS differed from the conventional systems in terms of functionality, disk layout, architecture etc. The researchers present an overview of the new file system and then discuss the implications it might have for the forensic community. Schuster (2008) conducted a study to examine the effect of the pool allocation strategies used by Microsoft Windows on forensic examination. To study this, the author designs an experiment in a controlled environment where snapshots of memory are taken over a period of time with new processes being in-

voked in a systematic manner. This allowed him to study the changes in the memory with each new process being invoked.

Fairbanks, Lee, and Owen III (2010) explored the forensic implications of the EXT4 file system. In typical fashion for these studies, they present an overview of the file system highlighting the changes from older systems. They then perform a controlled experiment where they populate the file system with known files and attempt to overwrite all the files with zeroed information. The authors observed that while it appears that all data has been over-written, traces of the test file can still be seen. They discuss the functionality of the file system that can be attributed to this anomaly and discuss the forensic implications of the findings. Hayes and Qureshi (2009) explore the forensic implications of the Vista operating system, which was new at the time. They outline the new features of the operating system and the forensic implications of the features. Specifically they discuss the changes to the NTFS file system, which includes modifications to the log files and security mechanisms like the inclusion of encryption and changes in the metadata.

Lastly, Garfinkel and Migletz (2009) published research related to the forensic significance of the new XML file formats. They explored the file structures related to the new document formats and present ways in which these aid digital investigation by contributing to better methods in data carving and data recovery.

### 4. METHODOLOGY

The aim of the research was to understand the changes in the forensic process brought about by using the Fusion Drive. There were three main goals of the research:

- Identify how the disk structure of the Fusion Drive differs from traditional drives
- Identify the data movement strategy between the drives
- Identify how the movement of the data between the two disks affects the modify-access-create (MAC) times.

The setup consisted of a 2012 iMac with a 3.1TB storage capacity on the Fusion Drive running Mac OS X version 10.8.2. The testing environment did not emulate a real life situation with a user freely accessing data and applications on the system. On the contrary, the test environment was extremely controlled while manipulating the movement of selected data. The corresponding disk activity was observed to understand the underlying storage allocation strategy. Schuster (2008) used a similar approach for his research.

#### 4.1 Identifying the Structure of the Disks

The first step was to analyze the available details of the drive on the system that could be used to distinguish between the two physical disks. As seen in Figure 1, the main storage information for the hard drive simply shows one drive while mentioning the presence of both the drives.

If we go to `Disk Utility` as seen in Figure 2, we see the device configuration of both the drives. This menu gives more details about the hardware configuration of both the drives including the drive interface type and the capacity of each drive. However, it does not provide information about current data usage on the drives or the names allocated to the individual drives. Finally, the `diskutil` command was used to identify the notations used for the individual drives. The output is discussed in the *Results* section.

#### 4.2 Analyzing the Movement of Data

A similar study to analyze the data movement strategy of the FD was published in a web log by Stein (2012). In the present study, the drive was populated with 200 GB in the form of 20 folders with each folder containing 100 files of 10 MB each. Random data was written using `dd` command with `/dev/urandom` as the input file. The drive initially had approximately 100 GB used. Disk activity was monitored using the `iostat` command, which shows rates at which data is being transferred to a particular disk.

As a next step, certain files were repeatedly accessed and the data migration activity was

observed. Files were also accessed at both block-level and file-level to estimate the granularity of the data being moved. (Solomon, Huebner, Bem, & Szeżynska, 2007) has used a similar approach to study persistence of data in memory. They made controlled changes to the system and took snapshots of the memory to analyze how the allocation changes with system changes that emulate a users usual behavior on the system.

## 5. RESULTS

This section outlines the results obtained from the experiment. It also presents some discussion about the importance of the results and significance of the same.

#### 5.1 Identifying the Structure of the Disks

Figure 3 shows the output of using the `diskutil` command in the terminal. We can see the notations that are used for the individual drives. In this particular drive, the flash drive is called `disk0` and the magnetic drive is called `disk1`. Again, we can see the individual disk capacities but we cannot see the individual disk utilization. The logical volume that combines `disk0` and `disk1` is called `disk2`. Many commercial disk analysis tools were employed to see how the drives are detected and all of them represented the disks as a single logical drive. A tool called *iStat Menus* showed the presence of multiple disks in the disk activity graph but any information about data storage was represented on a single logical volume.

#### 5.2 Analyzing the Movement of Data

As data was pushed to the logical drive, it was noticed that all the writes were directly made to the SDD till about 20 GB. After that the writes were directly going to the HDD. Since the size of the SDD is 20 GB, we can deduce that all writes go to the SDD first till the SDD is full and then the data continues to be written to the HDD. Figure 4 illustrates this, where the shift in disk activity between writing to the flashmemory and the hard drive is seen. The first three columns show the disk activity for the SDD and the next

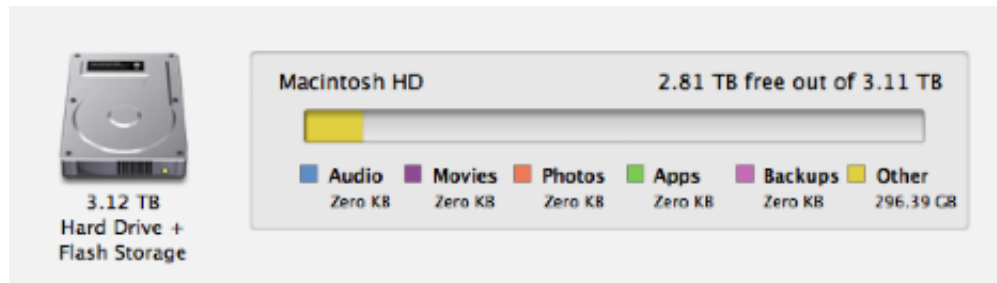


Figure 1 Device Storage Information Shown Under Storage Menu

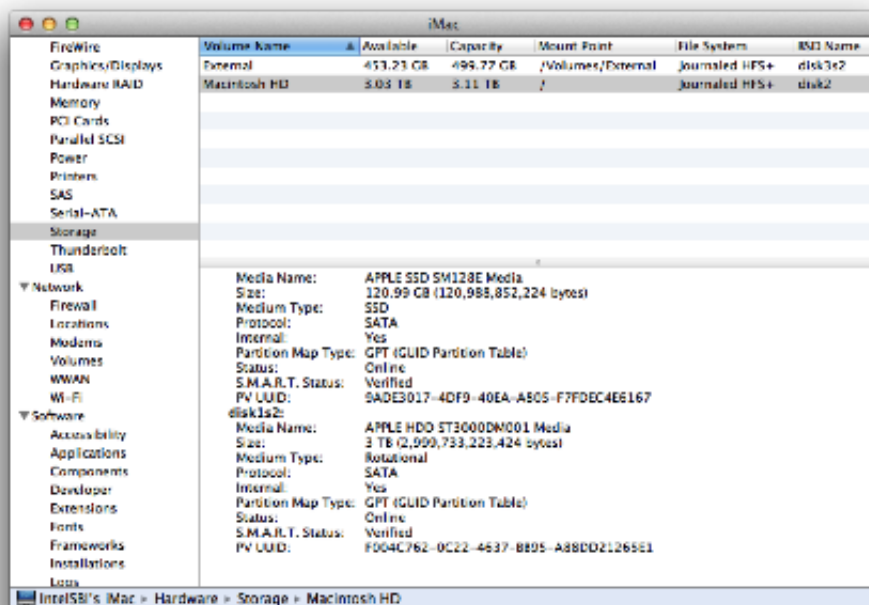


Figure 2 Device Storage Information Under the Disk Utility Menu

three columns show the disk activity for the HDD.

As soon as the last file was created, activity was seen on both drives (Figure 5) with data being read from disk0 (SSD) and being written to disk1 (HDD). The direction of data movement is estimated using the command `fs_usage`, which is indicated by the Core Storage read/write calls (Figure 6). This indicates data being transferred from the flash drive to the magnetic drive. This is done for around 4GB, which verifies the claims that the flash memory keeps a buffer area of 4 GB open for incoming files (Shimpi, 2013).

In the next step, to see how data is promoted

from the hard drive to the flash drive, we simulated a scenario where a user accesses certain files frequently. The files from folders 15-20 were assumed to be the files accessed frequently. The first megabyte from folders 15-20 was read continuously in a loop. This was done once all read activity had ceased from any previous steps. As mentioned earlier, each folder contained 10 GB of random data; so 50 GB of random data was being read continuously. As data is read from the files, it is first read from the HDD and then activity is seen both in the hard drive and the SSD. This is probably because as data is continuously read in a loop, it is promoted to the

```

dhcp-254-27:~ Intel$ diskutil cs list
CoreStorage logical volume groups (1 found)
|
+-- Logical Volume Group 689570F7-BF3C-4784-9173-5ACB7073B35B
-----
Name:          Macintosh HD
Size:          3120722075648 B (3.1 TB)
Free Space:    114688 B (114.7 KB)
|
+--< Physical Volume 9ADE3017-4DF9-40EA-A805-F7FDEC4E6167
-----
Index:        0
Disk:         disk0s2
Status:       Online
Size:         120988852224 B (121.0 GB)
|
+--< Physical Volume F004C762-0C22-4637-BB95-A88DD21265E1
-----
Index:        1
Disk:         disk1s2
Status:       Online
Size:         2999733223424 B (3.0 TB)
|
+--> Logical Volume Family 61F0ADD3-B47B-40D3-8B67-50E25C6D0004
-----
Encryption Status:  Unlocked
Encryption Type:    None
Conversion Status:  NoConversion
Conversion Direction: -none-
Has Encrypted Extents: No
Fully Secure:      No
Passphrase Required: No
|
+--> Logical Volume 23C5B119-3C82-49AC-B359-AC655A5529DF
-----
Disk:           disk2
Status:         Online
Size (Total):   3106191572992 B (3.1 TB)
Size (Converted): -none-
Revertible:     No
LV Name:        Macintosh HD
Volume Name:    Macintosh HD
Content Hint:   Apple_HFS
dhcp-254-27:~ Intel$ █

```

Figure 3 Output of the `diskutil` Command Showing Both Drives

flash drive. The activity again moves back to SSD, which indicates that the promoted data is being read from the flash drive now. These three transitions can be seen in Figure 7.

After the data-reading loop is stopped, there is activity between the flash memory and the HDD where data is written to the HDD (Figure 8). This is the FD demoting other data to free the 4 GB space on the flash drive.

The data-reading step is repeated again. However, this time the whole file is read instead of simply the first megabyte from each file. It can be seen in Figure 9 that the first MB is taken from the SSD and the rest is read from the HDD. This shows us two interesting features. First that the data previously read was indeed moved to the SSD (in the form of 1 MB from each file) and secondly that the data is moved



disk0			disk1			cpu			load average		
KB/t	tps	MB/s	KB/t	tps	MB/s	us	sy	id	1m	5m	15m
4.00	1	0.00	1024.00	145	144.81	1	6	93	1.48	0.91	0.70
4.00	1	0.00	1024.00	104	103.86	1	4	95	1.48	0.91	0.70
4.00	1	0.00	1024.00	148	147.79	1	6	93	1.48	0.91	0.70
4.00	1	0.00	1024.00	87	86.89	1	4	95	1.48	0.91	0.70
4.00	1	0.00	1004.18	154	150.82	1	6	93	1.44	0.91	0.70

Figure 4 Output of the iostat Command Showing Transfer of Activity from the Flash Drive (disk0) to the Magnetic Drive (disk1)

disk0			disk1			cpu			load average		
KB/t	tps	MB/s	KB/t	tps	MB/s	us	sy	id	1m	5m	15m
127.61	622	77.54	128.00	621	77.65	2	3	95	3.00	2.46	1.73
127.63	647	80.58	128.00	645	80.57	2	3	94	3.00	2.46	1.73
118.91	672	78.03	128.00	620	77.51	3	4	93	3.00	2.46	1.73
117.40	882	101.09	128.00	804	100.48	3	4	93	3.00	2.46	1.73

Figure 5 Disk Activity Seen on Both Drives When Data is Being Transferred from the Flash Drive to the Magnetic Drive to Free 4 GB

18:21:04.094875	RdChunkCS	D=0x04f66100	B=0x20000	/dev/disk0s2	0.002371	W	kernel_task.5041433
18:21:04.094890	RdBgMigrCS	D=0x00cea580	B=0x20000	/dev/CS	0.002392	W	kernel_task.5041433
18:21:04.095485	WrChunkCS	D=0xf8192c00	B=0x20000	/dev/disk1s2	0.000544	W	kernel_task.5041433
18:21:04.095498	WrBgMigrCS	D=0x00cea580	B=0x20000	/dev/CS	0.000563	W	kernel_task.5041433
18:21:04.096338	RdChunkCS	D=0x04f66200	B=0x20000	/dev/disk0s2	0.000897	W	kernel_task.5041433
18:21:04.096356	RdBgMigrCS	D=0x00cea5a0	B=0x20000	/dev/CS	0.000828	W	kernel_task.5041433
18:21:04.096793	WrChunkCS	D=0xf8192d00	B=0x20000	/dev/disk1s2	0.000403	W	kernel_task.5041433
18:21:04.096804	WrBgMigrCS	D=0x00cea5a0	B=0x20000	/dev/CS	0.000418	W	kernel_task.5041433
18:21:04.097120	RdChunkCS	D=0x04f66300	B=0x20000	/dev/disk0s2	0.000291	W	kernel_task.5041433
18:21:04.097137	RdBgMigrCS	D=0x00cea5c0	B=0x20000	/dev/CS	0.000311	W	kernel_task.5041433
18:21:04.097500	WrChunkCS	D=0xf8192e00	B=0x20000	/dev/disk1s2	0.000405	W	kernel_task.5041433
18:21:04.097500	WrBgMigrCS	D=0x00cea5c0	B=0x20000	/dev/CS	0.000420	W	kernel_task.5041433
18:21:04.099166	RdChunkCS	D=0x04f66400	B=0x20000	/dev/disk0s2	0.001549	W	kernel_task.5041433
18:21:04.099199	RdBgMigrCS	D=0x00cea5e0	B=0x20000	/dev/CS	0.001586	W	kernel_task.5041433
18:21:04.099633	WrChunkCS	D=0xf8192f00	B=0x20000	/dev/disk1s2	0.000399	W	kernel_task.5041433
18:21:04.099667	WrBgMigrCS	D=0x00cea5e0	B=0x20000	/dev/CS	0.000439	W	kernel_task.5041433
18:21:04.099972	select	S=0		/dev/CS	0.000005	W	Install Adobe FL.4207452
18:21:04.101573	RdChunkCS	D=0x04f66500	B=0x20000	/dev/disk0s2	0.001878	W	kernel_task.5041433
18:21:04.101614	RdBgMigrCS	D=0x00cea600	B=0x20000	/dev/CS	0.001922	W	kernel_task.5041433
18:21:04.102061	WrChunkCS	D=0xf8193000	B=0x20000	/dev/disk1s2	0.000411	W	kernel_task.5041433
18:21:04.102095	WrBgMigrCS	D=0x00cea600	B=0x20000	/dev/CS	0.000450	W	kernel_task.5041433
18:21:04.103092	RdChunkCS	D=0x04f66600	B=0x20000	/dev/disk0s2	0.000970	W	kernel_task.5041433

Figure 6 Core Storage Calls Indicating the Direction of Data Movement

at blocklevel and not file level.

Between each step, the purge command was used to flush out the cache. It was seen that if data is being read from previously being stored on the cache then the FD does not promote the data from the HDD. In another words, cache contents are not taken into consideration for storage allocation. Lastly, the MAC times of

the files were observed and it was seen that as data is moved between disks, the MAC times do not show any anomalous behavior. It has already been established that everything in the Fusion Drive gets written to the flash drive first. Thus the created time remains the time at which data is first created on the flash drive.



disk0			disk1			cpu			load average		
KB/t	tps	MB/s	KB/t	tps	MB/s	us	sy	id	1m	5m	15m
133.00	8	1.04	336.24	134	43.94	4	3	94	0.61	0.51	0.52
11.69	13	0.15	290.59	148	41.96	4	3	93	0.80	0.55	0.54
12.50	8	0.10	338.82	136	44.95	4	3	93	0.80	0.55	0.54
0.00	0	0.00	341.33	132	43.96	4	3	93	0.80	0.55	0.54
8.00	4	0.03	336.12	131	42.94	4	3	93	0.80	0.55	0.54
15.20	5	0.07	289.93	166	46.98	4	3	93	0.80	0.55	0.54
0.00	0	0.00	341.33	129	42.94	4	3	93	0.98	0.59	0.55
64.98	179	11.35	187.66	118	21.61	10	13	77	0.98	0.59	0.55
96.04	596	55.92	128.00	437	54.68	12	16	72	0.98	0.59	0.55
49.36	537	25.90	128.00	180	22.47	12	16	73	0.98	0.59	0.55
128.00	143	17.85	128.00	143	17.85	12	15	73	0.98	0.59	0.55
44.74	493	21.55	128.00	150	18.73	12	16	73	1.30	0.66	0.58
75.62	597	44.10	127.63	338	42.08	12	16	72	1.30	0.66	0.58
33.64	573	18.84	128.00	119	14.86	12	16	72	1.30	0.66	0.58
0.00	0	0.00	0.00	0	0.00	12	15	73	1.30	0.66	0.58
36.86	14	0.50	0.00	0	0.00	12	15	74	1.30	0.66	0.58
15.06	17	0.25	128.00	1	0.12	11	15	74	1.44	0.70	0.59
44.00	12	0.51	0.00	0	0.00	12	15	73	1.44	0.70	0.59
0.00	0	0.00	0.00	0	0.00	12	15	74	1.44	0.70	0.59
22.80	117	2.60	0.00	0	0.00	12	16	72	1.44	0.70	0.59
disk0			disk1			cpu			load average		
KB/t	tps	MB/s	KB/t	tps	MB/s	us	sy	id	1m	5m	15m
28.75	118	3.31	0.00	0	0.00	12	16	72	1.44	0.70	0.59
32.32	215	6.78	0.00	0	0.00	23	18	59	1.40	0.71	0.60
41.05	164	6.57	0.00	0	0.00	13	16	71	1.40	0.71	0.60
28.55	174	4.85	60.00	2	0.12	17	28	55	1.40	0.71	0.60

Figure 7 Disk Activity Showing Data Being Promoted from the Magnetic Drive to the Solid State Drive

Disk 0	KB/t	tps	MB/s	Disk 1	KB/t	tps	MB/s	us	sy	id	1m	5m	15m
0.00	0	0.00	0.00	0.00	0	0.00	0.00	1	1	98	0.86	0.93	0.93
0.00	0	0.00	0.00	0.00	0	0.00	0.00	2	1	98	0.86	0.93	0.93
0.00	0	0.00	0.00	0.00	0	0.00	0.00	1	1	98	0.86	0.93	0.93
137.70	233	31.29	78.67	24	1.84	2	2	96	0.79	0.92	0.92	0.92	
277.10	273	73.77	104.14	59	5.99	4	4	92	0.79	0.92	0.92	0.92	
251.62	388	95.27	95.47	53	4.94	4	4	92	0.79	0.92	0.92	0.92	
283.36	356	98.44	93.80	49	4.48	4	4	92	0.79	0.92	0.92	0.92	
160.46	1217	190.63	49.33	3	0.14	6	8	86	0.79	0.92	0.92	0.92	
0.00	0	0.00	0.00	0	0.00	10	12	78	0.81	0.92	0.92	0.92	
0.00	0	0.00	0.00	0	0.00	9	11	79	0.81	0.92	0.92	0.92	

Figure 8 Disk Activity Showing the Flash Drive Freeing up Space After Reading Stops

## 6. CONCLUSIONS

The biggest impact of the Fusion Drive on the forensic methodology would likely be the change in the imaging process. While the drive was not tested with existing Mac tools, it is mostly expected that imaging the Fusion Drive would need specialized tools that can recover data from tiered storage. Imaging of the Fusion Drive

would probably be similar to the imaging process used for RAID storage systems. Before the advent of the Fusion Drive, tiered storage was not commonly seen in home computers, and hence not a big concern for law enforcement (Duplessie, 2004). The large capacity of the drive (3.1 TB) will also present challenges to imaging the drive but this is not specific to the Fusion Drive. Increasing storage capacity is an

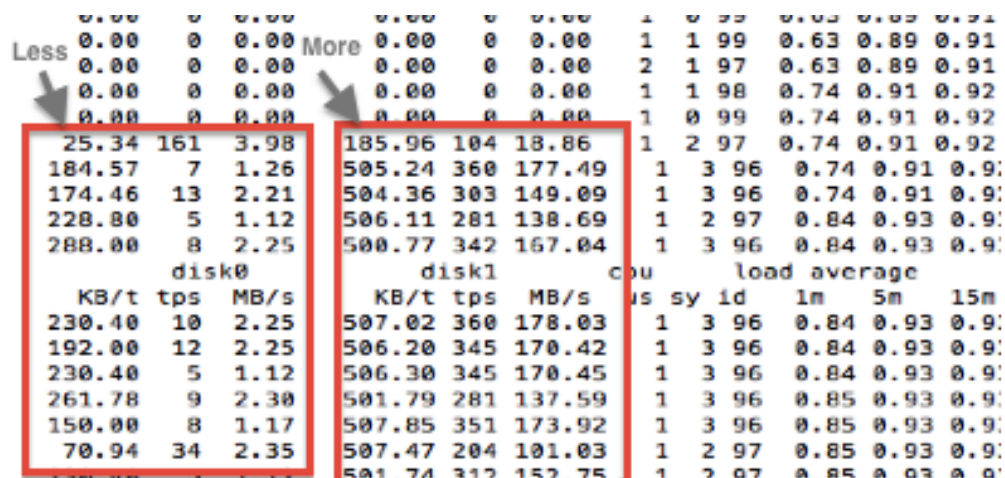


Figure 9 Disk Activity Showing 1 MB of File Read from SSD and Rest from Magnetic Drive

old problem that the forensic community has had to deal with.

The study also showed that multiple factors impact the allocation of data. Unlike with conventional SSD caches, the presence of data in the flash memory does not indicate that the data was frequently accessed. As seen by the manipulations, all data is written to the flash drive first. Data can be promoted or demoted based on the amount of space needed on the flash drive and cannot be explicitly related to the usage of the data. Thus, the location of the data does not provide much information about its usage. If data is moved to the HDD and then moved back to the SSD, it could be an indicator that the data was accessed frequently but this would be an estimate relative to other files. Also, with lack of documentation about data movement strategies, there might be other factors like the environment state mentioned in the patent (Bazzani, 2010) that determine the allocation of data. Therefore, at this stage it is unlikely that much forensic significance can be attributed to the location of data on either of the drives.

The granularity level of the data movement suggests that parts of the same file can exist on both drives at the same time. Thus, if one of the drives gets corrupted, it might be difficult to reconstruct the data. However, all data is written to the flash drive first. This includes

all applications, personal files, cookies, etc. The flash drive has a capacity of around 120 GB. Thus, if the Fusion Drive shows less than 120 GB being used, it is possible that all the data is stored on the flash drive solely. The observations also show that the MAC times are not a concern with a Fusion Drive, as MAC times are not impacted by the data migration between the disks. The timestamps retained the value when they were first written to the flash drive and did not alter even when the files were transferred between disks.

Every new technology comes with new challenges for the forensic examiners. The Fusion Drive is not much different. The challenges associated with the Fusion Drive mostly seem related to imaging the drive. With more information being available about the data allocation algorithm, it might be possible to attribute forensic importance to the data location. Other than these observations, based on documentation currently available, the Fusion Drive does not seem to provide any advantages or disadvantages to forensic examination.

Future work in this area would involve imaging and indexing of the Fusion Drive with commonly used forensic tools compatible with Mac computers. Also, with frequent changes in technology, it would be interesting to note any modifications to the architecture and data migration algorithms used in Fusion Drives, and the rami-

fications of these changes.

## REFERENCES

- AppleInsider. (2013). *Apple's mac shipments decline in u.s. as pc market slows*. Retrieved 2013-03-11, from <http://appleinsider.com/articles/13/01/10/apples-mac-shipments-decline-in-us-as-pcmarket-slows>
- Apple support. (2013). Retrieved 2013-01-28, from {<http://support.apple.com/kb/HT5446>}
- Bazzani, K. (2010, March 24). *Hybrid-device storage based on environmental state*. Google Patents. (US Patent App. 12/730,838)
- Beebe, N. L., Stacy, S. D., & Stuckey, D. (2009). Digital forensic implications of zfs. *digital investigation*, 6, S99--S107.
- Denning, P. J. (2005). The locality principle. *Communications of the ACM*, 48(7), 19--24.
- Duplessie, S. (2004). A blueprint for tiered storage. [White Paper]. Retrieved from <http://images.apple.com/xserve/raid/pdf/Tiered\Storage\Whitepaper.pdf>
- Fairbanks, K. D., Lee, C. P., & Owen III, H. L. (2010). Forensic implications of ext4. In *Proceedings of the sixth annual workshop on cyber security and information intelligence research* (p. 22).
- Garfinkel, S. L., & Migletz, J. J. (2009). *New xml-based files: implications for forensics* (Tech. Rep.). DTIC Document.
- Hayes, D. R., & Qureshi, S. (2009). Implications of microsoft vista operating system for computer forensics investigations. In *Systems, applications and technology conference, 2009. lisat'09. ieee long island* (pp. 1--9).
- Hutchinson, L. (2012). *Achieving fusionwith a service training doc, ars tears open apples fusion drive*. Retrieved 2013-03-11, from <http://arstechnica.com/apple/2012/11/achieving-fusion-with-a-service-training-doc-ars-tears-open-apples-fusion-drive/>
- NetMarketShare. (2013). *Operating system market share*. Retrieved 2013-03-11, from <http://www.netmarketshare.com/operating-system-marketshare.aspx>
- Peron, C. S., & Legary, M. (2005). Digital anti-forensics: emerging trends in data transformation techniques. In *Proceedings of*.
- Purcher, J. (2012). *Apple fulfills patent with the imac's new fusion drive*. Retrieved 2013-02-15, from <http://www.patentlyapple.com/patentlyapple/2012/10/apple-fulfills-patent-with-the-imacs-new-fusion-drive.html>
- Schuster, A. (2008). The impact of microsoft windows pool allocation strategies on memory forensics. *digital investigation*, 5, S58--S64.
- Shimpi, A. (2013). *A month with apple's fusion drive*. Retrieved 2013-01-28, from <http://www.anandtech.com/show/6679/a-month-with-applesfusion-drive>
- Solomon, J., Huebner, E., Bem, D., & Szeżynska, M. (2007). User data persistence in physical memory. *Digital Investigation*, 4(2), 68--72.
- Stein, P. (2012). *Fusion drive on older macs? yes!* Retrieved 2013-03-28, from <http://jollyjinx.tumblr.com/>