Digital Forensics Compute Cluster (DFORC2) – A New High Speed Distributed Computing Capability for Digital Forensics

Daniel Gonzales, Zev Winkelman, Trung Tran, Ricardo Sanchez, Dulani Woods and John Hollywood
RAND Center for Justice, Infrastructure, and Environment, RAND Corporation
Arlington, Virginia, 20815, U.S.A.

ABSTRACT

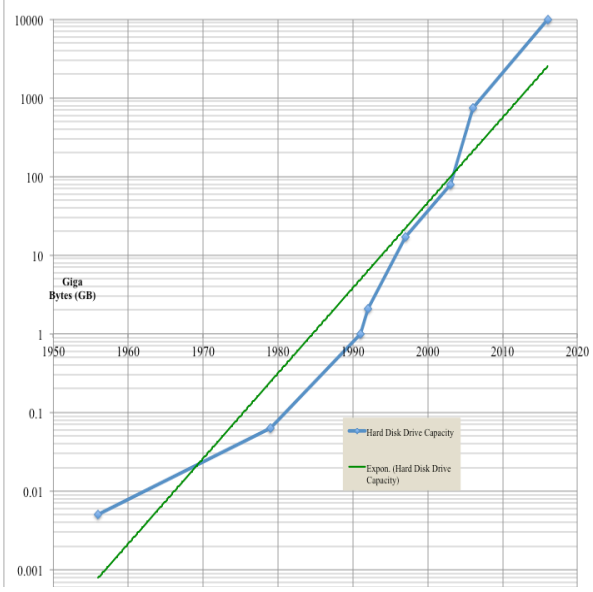
We have developed a new distributed computing capability, Digital Forensics Compute Cluster (DFORC2) to speed up the ingestion and processing of digital evidence. DFORC2 parallelizes evidence ingestion and file processing steps. It can be run on a standalone server or in the Amazon Web Services (AWS) cloud. When running in a cloud computing environment, its cluster resources can be dynamically scaled up or down using Kubernetes. DFORC2 is an open source project that uses Autopsy, Apache Spark and Kafka, and other open source software packages. It extends Autopsy's forensics capabilities to compute clusters and cloud architectures, so key digital forensics tasks can be accomplished simultaneously by a scalable array of cluster compute nodes. In this paper we compare the performance of a DFORC2 with a standalone version of Autopsy for evidentiary hard drives of different sizes.

Keywords: Digital Forensics, Cybersecurity, Cloud Computing, Spark, Kafka, Kubernetes.

1. INTRODUCTION

Law enforcement agencies (LEAs) and private investigators face an increasing backlog of evidence that must be analyzed to conduct thorough criminal investigations in a wide range of cases. Figure 1 indicates how hard disk drive (HDD) storage capacity has grown over time from the 1950s to 2016.

Figure 1 –HDD Storage Capacity Growth

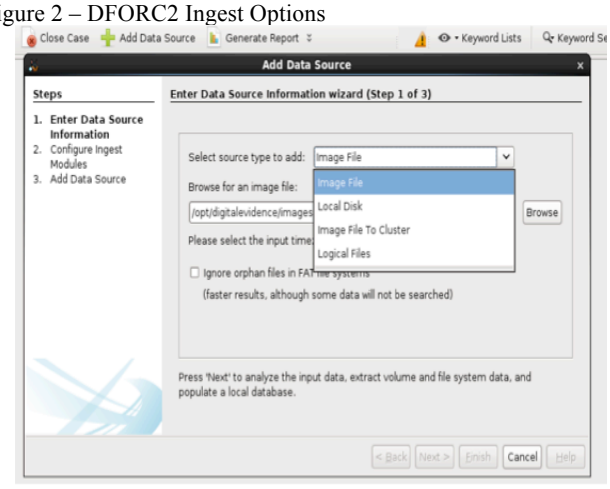


Sources: [1], [2], [3].

Cyber security breach, financial fraud, general criminal, and sexual contraband investigations increasingly require the use of digital forensics. In such investigations it is not uncommon to encounter HDDs with 1 to 2 TB of capacity. Even though HDD capacity growth has slowed from the torrid pace of the early 1990s, HDD capacity is still growing at an exponential rate. In the fourth quarter of 2016 it was possible for consumers to purchase a 10 TB, indicating that LEA investigation backlogs will continue to grow in the future [3]. Furthermore, the situation is even more challenging than shown in Figure 1, as solid state drives (SSDs) can provide even more storage than spinning disk HDDs. Seagate claimed in August 2016 their Barracuda SSD was the largest consumer storage device available with a capacity of 60 TB [3].

Consequently, investigators require faster digital forensics tools. RAND was funded by the National Institute of Justice (NIJ) to develop a distributed computing capability that can accelerate digital forensics analysis.[1] To accomplish this objective RAND developed DFORC2, which is designed to provide LEAs and other users with a cost-effective and efficient digital forensics analysis capability. DFORC2 combines data ingest and file analysis steps so they can be run in parallel instead of serial fashion. We have done this by modifying Autopsy [4], established open source digital forensics tool suite, so it can run on stream processing and compute nodes in compute cluster. It is designed to run on a standalone server or in the Amazon Web Services (AWS) cloud. DFORC2 on AWS is designed to reduce infrastructure costs, as it will stand up cloud computing infrastructure only when it is needed.

Figure 2 – DFORC2 Ingest Options



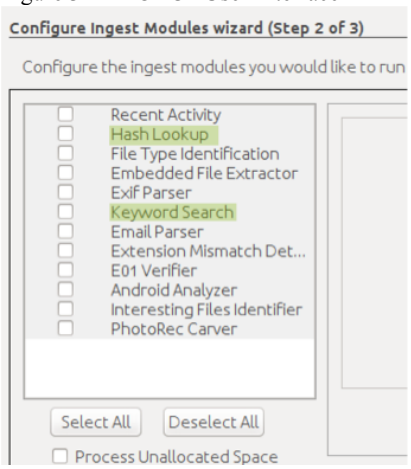
[1] This project was supported by NIJ Award No. 2014-IJ-CX-K102, awarded by the National Institute of Justice, Office of Justice Programs, U.S. Department of Justice.

DFORC2 is designed to hide complexity from the user, so it uses the Autopsy User Interface (UI), as shown in Figure 2.

Users familiar with Autopsy will easily be able to use DFORC2. To start a forensics analysis using DFORC2, one selects the target where investigation artifacts will be stored. When the user selects "Image file to Cluster," DFORC2 pipelines are activated and used to ingest and process the subject HDD image.

After the image target is selected the user uses the standard Autopsy UI to select which Autopsy forensics modules to run in the ingest and processing pipelines, as shown in Figure 3.

Figure 3 – DFORC2 User Interface



2. DFORC2 SYSTEM ARCHITECTURE

The DFORC2 software architecture is shown in Figures 4 and 5. The analysis process starts with ingestion of the subject drive, which is done with dc3dd and the Spark cluster.

Figure 4 – DFORC2 Spark Cluster

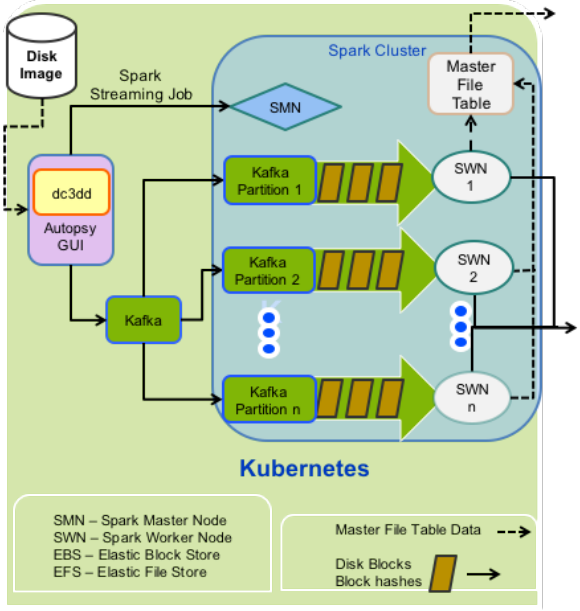
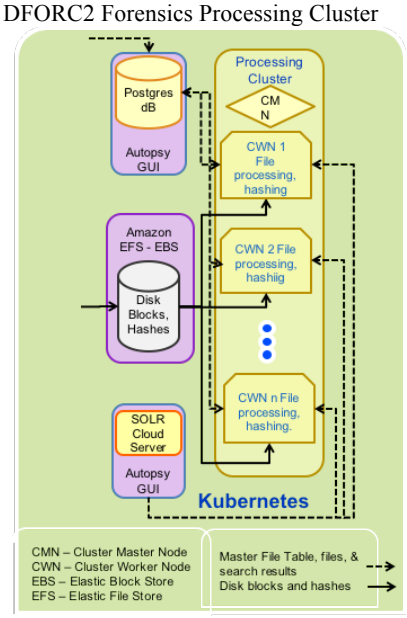


Figure 4 shows the "front end" of DFORC2 which includes dc3dd and the Spark cluster. The disk image is read into the system using the ingest functionality of dc3dd [5]. dc3dd is called through the Autopsy UI as described above. dc3dd sends disk image blocks to the Kafka messaging service [6]. It hashes each block so DFORC2 can confirm ingested disk blocks have not been corrupted during ingestion processing. Apache Spark is used to manage the initial processing steps of the image ingest process into a high speed stream processing job [7]. Spark worker nodes pull blocks from one of several Kafka message queues, which are called Kafka partitions. Spark ingest nodes are paired with a Kafka partition to maximize performance, as shown in Figure 4. Data blocks are hashed before transfer to Spark by dc3dd and after receipt by each Spark node to ensure integrity. DFORC2 identifies "complete" files, reconstructs the master logical file file table, and stores this logical file metadata in the Postgres database shown in Figure 5, which is the same database the collaborative configuration of Autopsy uses to hold most investigation case metadata [8].

In DFORC2 Spark worker nodes only perform HDD block ingestion and master file table reconstruction tasks. A separate Processing Cluster, shown in Figure 5, runs a specially modified, "headless," version of Autopsy on each cluster worker node, so each worker node can independently perform digital forensics tasks (file hashing, digital key word searches, etc.) on logical files found on the HDD, as shown in Figure 5.

Figure 5 – DFORC2 Forensics Processing Cluster



Autopsy uses SOLR, an Apache open source enterprise search platform, to build search string indices [9]. After files are processed by the cluster worker nodes, they are sent to SOLR, which creates indices to be used in subsequent key word searches.

In contrast to the original standalone version of Autopsy, DFORC2 uses a scalable distributed storage architecture that an be accessed by the DFORC2 Spark and Processing Clusters. When Autopsy is run in a collaborative mode, where more than one PC is used in the investigation, and as a standalone application it stores investigation artifacts in a Postgres database, search string indices are stored in SOLR, and evidence disk blocks and hashes are stored in separate data store on the local file system. In a distributed computing system the local file system should not be used in order to prevent memory conflicts and high overhead data synchronization tasks and

onerous communications between compute cluster nodes. Therefore, the headless version of Autopsy used in DFORC2 has been modified so it stores disk blocks and hashes in the AWS Elastic File System (EFS) [10] or in AWS Elastic Block Store (EBS) [11]. When DFORC2 is used on a standalone server it reverts to the local file system.

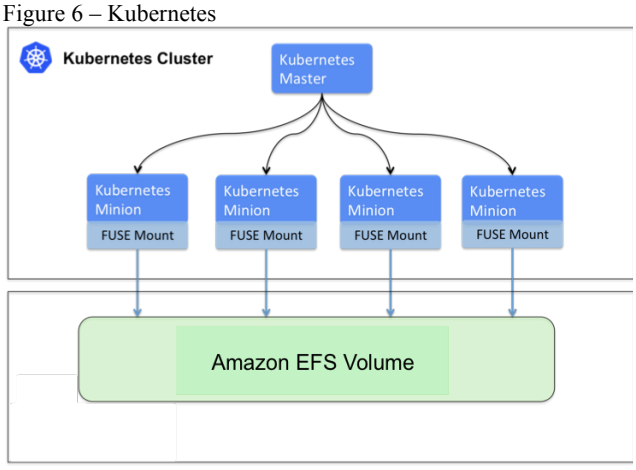
Amazon EFS is used because it provides a scalable and cost effective cloud storage solution for digital forensics processing. The size of the EFS volume is scaled up or down automatically by AWS so the user does not have to anticipate how much storage will be required prior to the investigation. In addition, the user does not have reserve a large storage volume that will be costly to reserve and lease. EFS is a cost effective storage solution, except in cases where the number of input/output operations (IOPS) exceeds a certain AWS performance

Finally, we note the number of Kafka partitions, Spark nodes and processing cluster nodes are all adjustable. When DFORC2 is run on a standalone server, the user will set the number of cluster nodes and Kafka partitions manually. When DFORC2 is run in the cloud these are set manually by the user at the start of an HDD forensic examination. Kubernetes also provides the capability to adjust the number of compute nodes dynamically during run time if any compute node runs out of memory or if it reaches the CPU processing limits of the node. These and other capabilities of Kubernetes are described in more detail below.

Kubernetes

The underlying foundation of DFORC2 is Kubernetes, which enables cluster size to dynamically scaled up or down [12]. Kubernetes is an open source project that is compatible with several major cloud computing environments, including AWS. It can orchestrate containerized applications and provides auto-scaling so the size of the Spark and processing clusters can be adjusted during run time based on demand. Without Kubernetes the number of Spark and processing cluster worker nodes would have to be fixed by the user prior to the start of a run. Without Kubernetes, the digital forensics analysts using DFORC2 would have to estimate the number of Spark and cluster worker nodes needed for a specific size hard disk and for a specific type of investigation. The number of cluster nodes needed could depend on many factors, which may be unknown to the analyst before the investigation. This limitation would likely require the analyst to overprovision the Spark and processing clusters to minimize timely processing of the evidence. Kubernetes solves this problem.

Figure 6 – Kubernetes



Kubernetes requires all applications to be placed in Docker containers. RAND has ported Autopsy and other components of DFORC2 to a standard set of Linux containers. These run as Kubernetes nodes or minions, which can then be linked to a centralized file system, as indicated in Figure 6.

3. STANDALONE AUTOPSY PERFORMANCE TESTS

We first tested a standalone version of Autopsy using a number of test HDD images.

Test Images

We used the following HDD images to evaluate the performance of Autopsy and DFORC2. These test images are widely used in the digital forensics academic community.

Table 1 – Test HDD Images

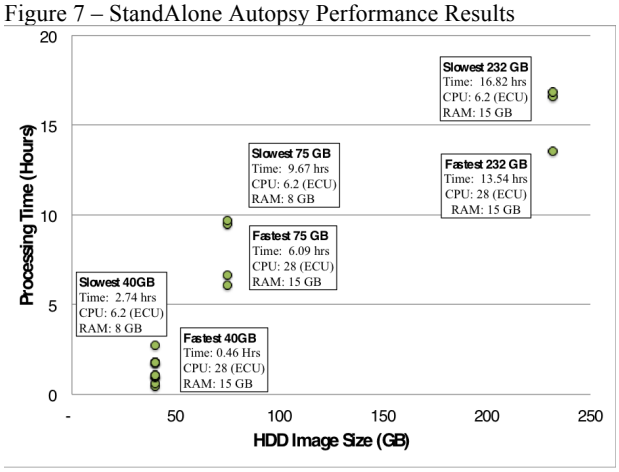
Image	Size
NPS DOMEX Users, 2009	40 GB
NPS 1weapondeletion, 2011	75 GB
NPS 2weapons, 2011	232 GB
NPS 2 TB, 2011	2 TB

Sources: [13], [14].

We tested the "off the shelf" standalone version of Autopsy using virtual machines (VMs) on AWS. Since the unmodified version of Autopsy is designed to run on Windows, we selected virtual machines with the Windows Server (2012 release 2) operating system. The test was designed to mirror the initial image and file processing capabilities of DFORC2 version 1, so the only Autopsy modules that were enabled were the "Keyword Search" and the "Hash Lookup." Through discussions with the Autopsy developers at Basis, we learned that these are the most computationally intensive modules. The version of Autopsy we used in testing was the most recent at the time, version 4.1.1. It was configured to run in standalone mode (versus collaborative mode). Standalone Autopsy test results are shown in Figure 7 for three of the four test HDD images listed in Table 1 (Test results for the NPS 2 TB image are not shown in the figure because of space limitations).

Figure 7 shows the slowest and fastest times Autopsy processed 40, 75, and 232 GB HDD images.

Figure 7 – StandAlone Autopsy Performance Results



Autopsy processing times depend on server resources. Autopsy was tested on a 6.2 ECU server and 8 GB RAM (a so-called m4.large AWS instance) and a larger server with 28 ECU and 15 GB RAM [15].

The Autopsy test results shown in Figure 7 do not include the time required to ingest the subject image. Autopsy does not have a built in image ingestion capability. Another tool like dc3dd is required to ingest the image from the subject HDD in forensically sound manner. Total HDD image processing times when using dc3dd and Autopsy together will be the sum of the dc3dd image ingestion and Autopsy file processing times. RAND ingested the test images using dc3dd. dc3dd image ingestion times for 40, 75, and 253 GB size images are shown in Table 2.

Table 2 – dc3dd HDD Image Ingestion Times

Ingestion Time (minutes)	Image Size (GB)
22	40
42	75
142	232
1253	2048

Standalone Autopsy tests were conducted in the AWS cloud, and that the evidence hard disk was not physically attached to a server running Autopsy using a write-blocker.
The test HDD image was resident on the same server used by the standalone Autopsy application. Any additional communications delays encountered in an actual HDD image ingestion are not included. The reader will note that such communications time delays can reduce HDD forensics ingestion and analysis processing speeds substantially, regardless of whether DFORC2 or a standalone version of Autopsy is used. Consideration of such time delays is beyond the scope of the investigation described in this paper but will be subject of future research.

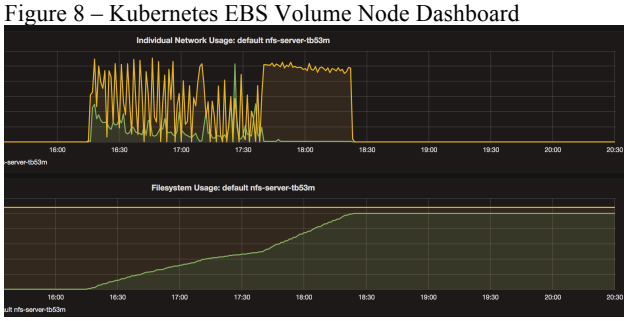
4. DFORC2 PERFORMANCE TESTS

DFORC2 was also tested in the AWS cloud. The DFORC2 test set up was similar to that used for standalone Autopsy, except the test HDD image was placed on a separate server. DFORC2, which was instantiated in a separate server cluster in the same AWS availability zone, ingested the image from this server.

Figure 8 is a Kubernetes dashboard screen shot which shows the HDD image block ingestion traffic into the EFS volume on the DFORC2 EBS server. This traffic is shown in yellow in the top pane of the figure. Shown in green in the top pane is communication traffic to the EBS volume from the processing cluster worker nodes that perform file processing simultaneously while image ingestion is on going. Processing cluster worker node traffic is much lower but continues after ingestion is completed (the results shown in the figure are for the 232 GB image case).

The green line in bottom pane of Figure 8 shows how quickly the EBS volume is filled by the Spark streaming process with HDD image blocks. When a dedicated dc3dd server performed the ingestion process, it completed ingestion in 2 hours and 22 minutes). For DFORC2 the image ingestion process is finished in a little over 4 hours for the 232 GB image. However, it should be noted that only 5 Spark nodes were used in the DFORC2 tests reported here. We believe DFORC2 image ingestion times can be reduced with a larger Spark cluster.

Figure 8 – Kubernetes EBS Volume Node Dashboard



DFORC2 performance results are shown in Table 3. As noted above DFORC2 was configured with 5 Spark nodes in all test cases. The number of worker nodes was varied between 10 and 35 nodes. All nodes or minions in the Kubernetes cluster used in these tests were standard AWS M4.large instances.

Table 3 – DFORC2 Performance Results

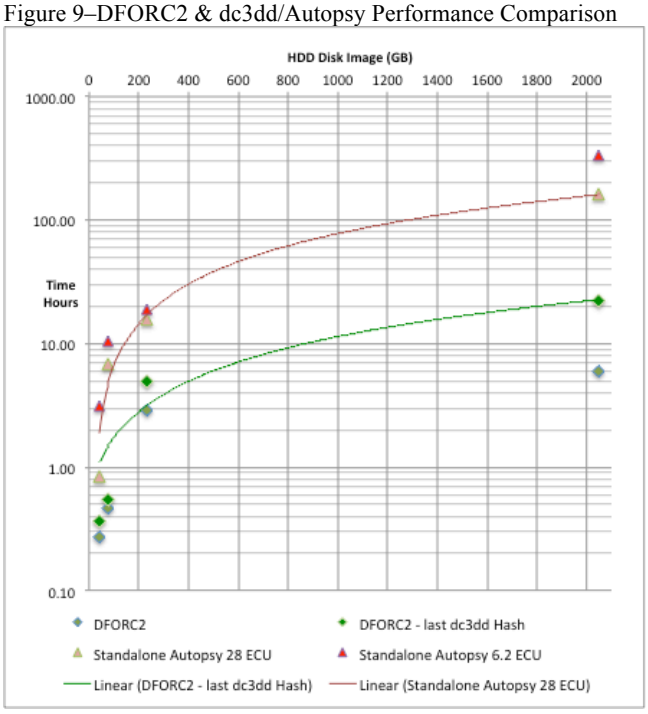
Ingestion & Processing Time (hours:min:sec)	Processing Cluster Nodes	Spark/ Kafka Nodes	HDD Image Size (GB)
0:22:00	20	5/5	40
0:33:00	20	5/5	75
4:56:00	35	5/5	232
22:30:05	35	10/10	2048

For the 40 GB image case, DFORC2 completes both ingestion and file processing in a little over 16 minutes. With 10 process cluster worker nodes DFORC2 completes processing a 75 GB image in about an 1 hour and 15 minutes. On the other hand, if DFORC2 is configured with 20 process cluster worker nodes, it completes processing of a 75 GB image in a little over 29 minutes, or in less than half the time. In contrast, dc3dd and stand alone Autopsy would require almost 7 hours to complete ingestion and processing of a 75 GB HDD image.

Table 3 shows that it took DFORC2 about 11 hours and 16 minutes to process a 232 GB image if it had access to 66 worker nodes. We believe DFORC2 performance will improve significantly for larger drive sizes when DFORC2 has access to more process cluster nodes and at least 10 Spark nodes.

Figure 9 compares the performance of DFORC2 to that of dc3dd and stand alone Autopsy. The results shown include image ingestion and file processing steps for both systems. As mentioned above DFORC2 performs image ingestion simultaneously with file processing. By combining these steps the overall investigation process can be sped up considerably, as shown in the figure.

Figure 9–DFORC2 & dc3dd/Autopsy Performance Comparison



DFORC2 completes image ingestion and file processing tasks much faster than the standalone Autopsy server. DFORC2 cuts the time in half for processing a 232 DG image, and is fourteen times faster than standalone Autopsy when a 2 TB HDD image is ingested and processed.

5. USING DFORC2 in COMPUTING CLOUDs

To simplify the test environment and manage costs we tested both Autopsy and DFORC2 in the AWS cloud. Many LEAs do not use cloud computing services to support digital forensics investigations because they lack tools that work effectively in the cloud, but also because of legal concerns as to whether the chain of custody for digital evidence can be demonstrated and maintained effectively in computing clouds. While this subject is beyond the scope of this paper, it is a subject that RAND is examining for DFORC2 (the results of this investigation will be reported in a separate paper).

DFORC2 has been designed so it can be run on a stand alone server, although in this deployment case, the number of Spark and processing cluster worker nodes will be limited (test results for DFORC2 on a stand alone server will be reported in a separate paper).

DFORC2 offers the most significant performance improvement for digital forensics when it is used as a cloud based application. There are four ways DFORC2 can be used in the cloud. These along with the traditional standalone server deployment approach are shown in Table 4.

The traditional approach for using DFORC2 is shown in the second of Table 4. In this case the HDD image would be ingested locally on a single machine or server. In the traditional approach, DFORC2 cannot be run as a distributed application, but with a multi-core server it would be run with its Spark streaming ingest capability. So, in this case, it is possible that DFORC2 could provide some performance improvement over using the combination of dc3dd and Autopsy for the same investigation. DFORC2 would also provide the advantage that data acquisition and analysis can be done at the same time, as the analyst would not have to wait several hours or days for the image ingestion job to complete.

Table 4 –DFORC2 Implementation Options

Option	Streaming	Distributed	Cloud
DFORC2 on premise single machine	Yes	No	No
DFORC2 on premise data center	Yes	Yes	No
DFORC2 on premise – remote data center	No	No	Yes
Ship HDDs to AWS DFORC2 processing	Yes	No	Yes

The second option shown in the table is to ingest the HDD image locally on premise at a private datacenter that was equipped with compute cluster. In this case DFORC2 could be loaded on the cluster and multiple Spark and processing cluster nodes could be instantiated, but the full functionality of Kubernetes (e.g., dynamic scaling) may not be available.

The third option shown in the table is to acquire the HDD image on premise and to stream the drive disk blocks into the Spark cluster over a high-speed communications link to a remote data center or cloud where DFORC2 would run. In this case the HDD image would be acquired locally, but ingested remotely (e.g., in the cloud). Spark is designed to support this approach, but it should be noted that good communications link with a bandwidth at least as high as dc3dd ingest speed is needed to make this approach practical. RAND has measured dc3dd ingest speed over USB 2.0 and determined it to be approximately 15 Mbps, which is therefore maximum ingestion speed for HDD images using common computing hardware.

The fourth option is to ship HDD evidence to a cloud service provider for remote acquisition, ingestion, and file processing. This option is only desirable if a high-speed communications link to the remote cloud data center is not available. We plan to investigate feasibility of employing this option with AWS.

8. CONCLUSIONS

This paper describes DFORC2, a new distributed computing capability for accelerating ingestion and analysis of digital evidence in a forensically sound manner. DFORC2 is based on high-performance open source software we have integrated with an established open source digital forensics tool suite – Autopsy. DFORC2 provides an effective digital forensics capability that provides many of the same digital forensics functons that Autopsy does as a stand alone application.

DFORC2 can run on a high-performance compute cluster or in a computing cloud such as AWS. We evaluated the performance of DFORC2 as a cloud application and compared its performance to a standalone version of Autopsy. The results demonstrate DFORC2 is faster by at least an order of magnitude than standalone Autopsy. RAND is investigating the DFORC2 chain of custody. In a future publication we plan to show that a robust chain of custody can be established for DFORC2.

9. REFERENCES

[1] S. J. Vaughan-Nichols, "Hard drive technology reaches a turning point," *Computer*, vol. 36, no. 12, pp. 21–23, 2003.

[2] "Timeline: 50 Years of Hard Drives," *PCWorld*, 13-Sep-2006. [Online]. Available: http://www.pcworld.com/article/127105/article.html. [Accessed: 04-Apr-2017].

[3] "Seagate's 10TB Barracuda Pro is the world's largest consumer hard drive," *PCWorld*, 19-Jul-2016. [Online]. Available: http://www.pcworld.com/article/3096292/storage/seagates-10tb-barracuda-pro-is-the-worlds-largest-consumer-hard-drive.html. [Accessed: 04-Apr-2017].

[4] "Autopsy." [Online]. Available: http://www.sleuthkit.org/autopsy/. [Accessed: 28-Jan-2016].

[5] "dc3dd download | SourceForge.net." [Online]. Available: http://sourceforge.net/projects/dc3dd/. [Accessed: 27-Jan-2016].

[6] "Apache Kafka." [Online]. Available: http://kafka.apache.org/index.html. [Accessed: 09-Jun-2015].

[7] "Apache Spark™ - Lightning-Fast Cluster Computing." [Online]. Available: https://spark.apache.org/. [Accessed: 09-Jun-2015].

[8] "PostgreSQL: The world's most advanced open source database." [Online]. Available: http://www.postgresql.org/. [Accessed: 28-Jan-2016].

[9] "Apache Solr -." [Online]. Available: http://lucene.apache.org/solr/. [Accessed: 05-Apr-2017].

[10] "Amazon EFS Performance - Amazon Elastic File System." [Online]. Available: http://docs.aws.amazon.com/efs/latest/ug/performance.html. [Accessed: 30-Jan-2017].

[11] "Amazon Elastic Block Store (EBS) – Block Storage for EC2," *Amazon Web Services, Inc.* [Online]. Available: //aws.amazon.com/ebs/. [Accessed: 30-Jan-2017].

[12] "Kubernetes," *Kubernetes*. [Online]. Available: http://kubernetes.io/. [Accessed: 30-Jan-2017].

[13] "Digital Corpora." .

[14] "The CFReDS Project." [Online]. Available: https://www.cfreds.nist.gov/. [Accessed: 05-Apr-2017].

[15] "Amazon EC2 FAQs - Amazon Web Services," *Amazon Web Services, Inc.* [Online]. Available: //aws.amazon.com/ec2/faqs/. [Accessed: 05-Apr-2017].

.