Forensics-as-a-Service (FaaS): Computer Forensic Workflow Management and Processing Using Cloud

Yuanfeng Wen, Xiaoxi Man, Khoa Le and Weidong Shi
Department of Computer Science
University of Houston
Houston, Texas 77204-3010
e-mail: {wyf, xman, ktle, larryshi}@cs.uh.edu

***Abstract*—Digital forensics is a critical technology for obtaining evidences in crime investigation. Nowadays, the overwhelming magnitude of data and the lack of easy-to-deploy software are among the major obstacles in the field of digital forensics. Cloud computing, which is designed to support large scale data processing on commodity hardware, provides a solution. However, to support forensic examination efficiently using cloud, one has to overcome many challenges such as lack of understanding and experiences on configuring and using digital forensic analytic tools by the investigators, and lack of interoperability among the forensic data processing software. To address these challenges and to leverage the emerging trends of service based computing, we proposed and experimented with a domain specific cloud environment for supporting forensic applications. We designed a cloud based framework for dealing with large volume of forensic data, sharing interoperable forensic software, and providing tools for forensic investigators to create and customize forensic data processing workflows. The experimental results show that the proposed approaches can significantly reduce forensic data analysis time by parallelizing the workload. The overhead for the investigators to design and configure complex forensic workflows is greatly minimized. The proposed workflow management solution can save up to 87% of analysis time in the tested scenarios.**

Keywords*—*cloud computing; digital forensics

I. Introduction

Digital forensics is a technology to collect, examine, analyze, but still preserve the integrity of the data in modern high-tech crimes [1]. Digital forensics were conventionally used in physical hardware analysis, such as hard-disk, flash drives. As the ever increasing computing and storage needs arising in the Internet age, investigators in the public and private sectors are facing the same growing challenge when dealing with computer forensics [2], which is to examine an increasing number of digital devices (e.g., GPS gadgets, smartphones, routers, embedded devices, SD cards), each containing an immense volume of data, in a timely manner and with limited resources. At the same time, with proliferation of low cost and easy-to-access anti-forensic techniques (sometimes open source as well), offenders are becoming increasingly sophisticated and skillful at concealing information.

Computer forensic investigators and examiners are confronted with the problems of, (i) unacceptable backlog of information waiting for examination; (ii) miss of critical time window to follow the leads due to slowness of computer forensic examination; (iii) lack of understanding of the computer forensics and consequent incapability by the detectives to take advantages of digital forensic techniques to advance investigations; and (iv) overlook of relevant data and waste of resources due to lack of understanding of crime investigations by the forensic examiners.

The cloud computing model provides ideal opportunities to solve these problems. Cloud computing is a rapidly evolving information technology that is gaining remarkable success in recent years. It uses a shared pool of virtualized and configurable computing resources (both hardware and software) over a network to deliver services, such as to host and analyze large datasets immediately. These resources and services can be rapidly provisioned and released with minimal management effort or service provider interaction. Cloud computing is almost everywhere. Governments, research institutes, and industry leaders are quickly adopting the cloud computing model to solve the increasing computing and storage demands. This trend has significant implications for digital forensic investigations.

However, current forensic research related to the cloud is mainly focused on the stage of data collection (e.g., [3]). The examination and analysis on the data are still performed on local machines instead of in the cloud. Extending the services to the cloud often calls for the external assistance and professional software/applications. Researchers have made efforts to build a forensic cloud. Sleuth-Hadoop [4] tries to integrate different forensic analysis tools into the cloud. However, Sleuth-Hadoop doesn't have the flexibility for the investigators to build and customize the desired analysis workflow for specific forensic datasets.

The main contribution of our work is to fill the gaps. We propose a domain specific cloud environment for forensic applications. We designed a cloud infrastructure framework for dealing with large forensic datasets, sharing forensic software, and providing a way for the investigators to build workflows using a common interface. We proposed a schema-based forensic analysis workflow framework. The framework allows the forensic investigators to define their requirements in XML configuration files. Supported with a collection of forensic applications, the framework can select the appropriate applications, generate the corresponding map-reduce drivers, and set up the workflow in the cloud, automatically for the users.

The rest of this paper is organized as follows. Section II presents the system design of the forensic cloud. Section III shows the experimental results. Related works are discussed in Section IV. Section V concludes the paper.

II. Background

Four categories of cloud computing are defined by NIST (National Institute of Standards and Technology) [5], i.e., private cloud, community cloud, public cloud, and hybrid cloud. Currently, most research focuses on the community cloud and public cloud.

In the community cloud study, there are many solutions proposed for data sharing and collaborations. At Hewlett-Packard Labs, Erickson et al. [6] use a cloud-based platform to provide content-centered collaboration in the Fractal project. Social sharing of workflows are studied by Roure et al. [7]. Globus Online [8] focuses on data-movement functions to deal with new challenges brought by data-intensive, computational, and collaborative scientific research through cloud-based services. Compared with these studies, our work mainly concentrates on the workflow management in computer forensics and domain specific cloud infrastructure. Various kinds of other community cloud are also studied, e.g., volunteer cloud [9], [10], Nebula cloud [11], social cloud [12]. However, none of those is specifically designed for computer forensics. For domain specific applications, the one size fits all approach would not work because the specific characteristics and requirements from each application domain often demand customized solutions built on top of the cloud infrastructure.

In the public cloud, since users have different purposes to run their applications, studies mainly focus on the general-purpose resource management. For example, public cloud such as Amazon EC2[13] uses a scheduler in Xen hypervisor to schedule virtual machines. Song et al. [14] proposed a multi-tiered on-demand resource scheduling scheme to improve resource utilization and guarantee QoS in virtual machine based data centers.

One of the most popular programming models in the cloud is MapReduce [15], which is for distributed processing of large-scale data on clusters of commodity servers. Ananthanarayanan et al. [16] proposed an optimized cluster file system for MapReduce applications. They use metablock that is a consecutive set of blocks of a file that are allocated on the same disk instead of the traditional cluster file system. Apache Pig [17] is a platform for analyzing large data sets using MapReduce on the top of Hadoop.

Digital forensics are performed in four phases [2], i.e., collection, examination, analysis and reporting. The investigators will execute the following separately, 1) identifying, recording, acquiring data from possible sources, while preserving the integrity of the data; 2) processing the data with a combination of manual and automated methods, and extracting data of particular interest; 3) analyzing the results of the examination with legally justifiable methods and techniques to derive useful information; 4) describing the results of the analysis.

Forensic software provides many different kinds of tools to investigate suspicious servers, desktops, and personal digital devices such as cell phones, GPS navigators, PDAs, etc. The investigations mainly focus on discovering forensic evidence, and identifying suspicious files and activities. Bulk_extractor [18] can scan suspicious files and email and extract data from the disk images, files, and directories. Many comprehensive tools, such as FTK [19], OSForensics [20], Intella [21], etc., provide the investigation functions. However, they are stand-alone software running on local machines. Supports for inter-operations and large scale automated parallelization are poor, or almost none. Open Computer Forensics Architecture (OCFA) [22] is an automated system that can extract metadata from files, create indices for the target disk images and ultimately output a repository containing the files and indices for further examination. OCFA is able to work with other third part analysis software or data mining tools. The limitation of the OCFA is that it is not integrated with the cloud.

Sleuth Kit [23] has a cloud-based version, Sleuth Hadoop, which integrates several forensic software and enables them to run in the cloud. However, the analysis workflow is fixed in Sleuth Hadoop [4] without the capabilities to configure and construct workflow dynamically. It doesn't support collaborative software development and workflow management.

III. System Design

A. System Overview

The forensic cloud infrastructure aims to deliver the services that go beyond today's models of "software-as-a-service" and "infrastructure-as-a-service", with the goal of providing not only elastic computing resources for on-demand computer forensic data processing, but also an environment for intelligent forensic workflow management, customization, and collaboration.

The forensic cloud comprises two main layers: a service layer and a physical resource layer, as shown in Figure 1. The service layer has three major components, the forensic data manager, the forensic application manager and the forensic workflow manager. The physical layer is composed of physical devices such as accelerators, physical servers, and storage servers for supporting forensic data banks. A set of virtual machines can be allocated for serving a particular forensic data processing task.

B. Forensic Data Manager

Forensic data manager provides supports for uploading, storing, and retrieving the large-scale forensic data in the cloud. Forensic data are collected from diverse sources (e.g., disks, cellphones, embedded devices). With elastic storage resources provided by the cloud, forensic investigators can process, analyze, and archive forensic data with reduced cost, improved efficiencies, and increased productivity.

Considering the scale of the data and the fact that most applications in the cloud use MapReduce [15] for parallelizing the applications and performing the analysis on the

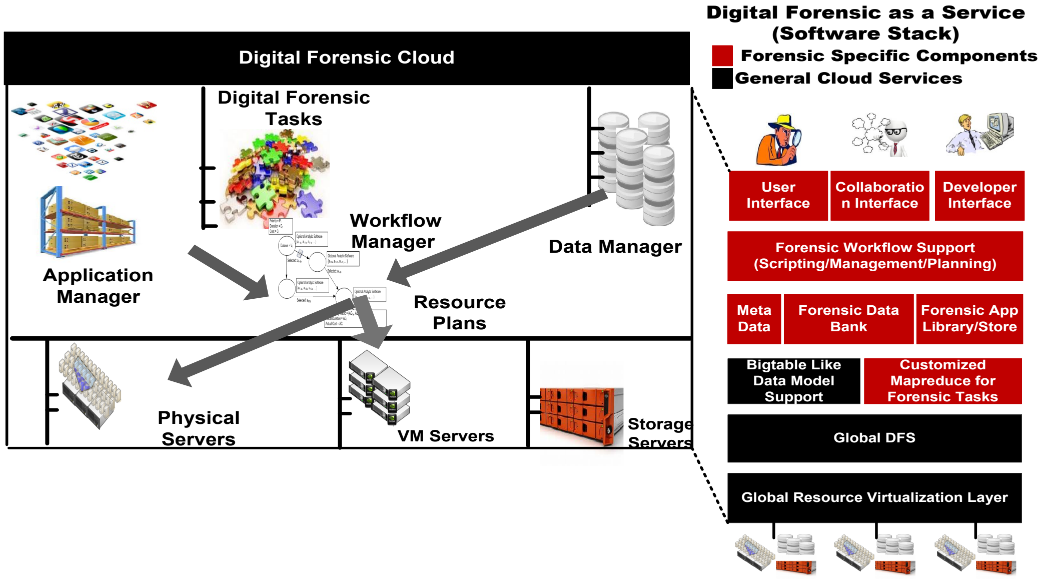

Digital Forensic Cloud
Digital Forensic Tasks
Application Manager
Workflow Manager
Data Manager
Resource Plans
Physical Servers
VM Servers
Storage Servers
Digital Forensic as a Service (Software Stack)
🟥 Forensic Specific Components
⬛ General Cloud Services
User Interface
Collaboration Interface
Developer Interface
Forensic Workflow Support (Scripting/Management/Planning)
Meta Data
Forensic Data Bank
Forensic App Library/Store
Bigtable Like Data Model Support
Customized Mapreduce for Forensic Tasks
Global DFS
Global Resource Virtualization Layer

Fig. 1. Forensic Cloud Overview and Software Stack

data, the data manager uses HDFS (Hadoop Distributed File System) [24] to store the data. HDFS is a distributed file system designed to work on commodity hardware maintained as a Hadoop subproject. HDFS stores all the data in blocks. The block size is usually 64MB or 128MB. HDFS works more efficiently if the single file size is larger than the block size, which, however, is not necessarily always the case for all the files in a target disk image. To avoid the small-file problem, the data manager organizes the files in HAR files or SequenceFile formats [25]. Creating a working copy, is managed by the forensic data manager as well. The forensic data manager also flattens all the directory information, which exports all the nested files into one folder. This can mitigate the anti-forensic (AF) approach called, "circular references". The "circular references" exploit uses symbolic links to point to a parent folder, which may make a search operation run for ever.

In addition, the data manager also maintains the metadata of the files in the HBase (an open-source, distributed, versioned, column-oriented store modeled after Google's Bigtable [26]). The metadata contains useful data for the files, for instance, the directory structure information before flatting, the hash values (MD5) of the files. The information is often used in analyzing the forensic data. For example, National Software Reference Library (NSRL) [27] provides a comprehensive database with the hash values for almost all the commercially available software. This provides a Reference Data Set (RDS) of information [27], which can be used as digital signatures of the known, good software applications. Therefore, by comparing the hash values of the files in a target disk with the database, the investigators can filter out all the uninterested files. This Known File Filter (KFF) operation can significantly reduce the sizes of the data that requires examination. All other similar metadata are calculated by the data manager and stored in the HBase. This is a default step when new files are uploaded to the forensic cloud and to be ingested.

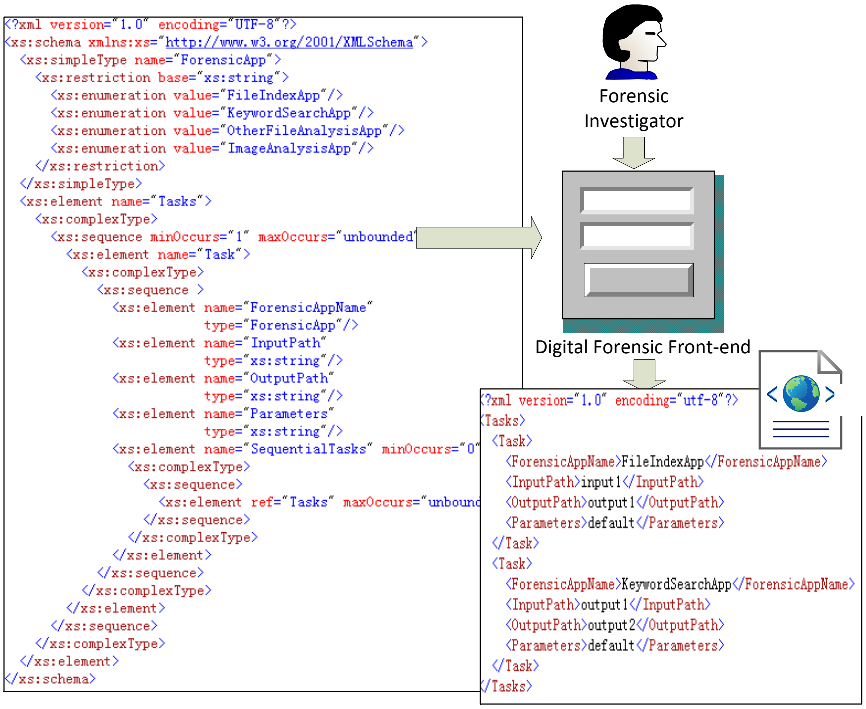
With the help of the universal management of the data, forensic analysis and data mining experts who develop software for forensic data processing only need to submit their software to the cloud.

C. *Forensic Application Manager*

Forensic applications and software such as files/emails search, image/videos analysis, etc. are created through collaborative processes involving many forensic experts and computer science researchers. To accelerate productivity and expedite collaborations among them, it is necessary to reuse the software and workflow. Forensic software vendors can distribute the developed algorithms and software to a software/app library, the "forensic app store" where forensic workflow can be constructed using these software. Forensic examiners and investigators can on-demand create, invoke, and deploy tasks using the forensic software and workflow stored in the library. Consequently, the infrastructure will accelerate dissemination and deployment of new forensic techniques.

All the applications in the "forensic app store" are tagged and categorized by the application manager. The application manager periodically generates an XML schema and metadata for all the available software. The schema is used to generate a user-friendly front-end web page (maintained by the workflow manager) and to validate the XML-based workflow configuration file.

An example schema file and xml configuration file are shown in Figure 2. In the schema file on the left of Figure 2, all the four applications available in the "app store" are listed. The digital forensic front-end web page can read the schema file and generate a drop-down list with these applications when a forensic investigator selects the applications. The investigators only need to click several buttons to generate an XML configuration file as shown on the right bottom of



```xml
<?xml version=“1.0“ encoding=“UTF-8“?>
<xs:schema xmlns:xs=“http://www.w3.org/2001/XMLSchema“>
  <xs:simpleType name=“ForensicApp“>
    <xs:restriction base=“xs:string“>
      <xs:enumeration value=“FileIndexApp“/>
      <xs:enumeration value=“KeywordSearchApp“/>
      <xs:enumeration value=“OtherFileAnalysisApp“/>
      <xs:enumeration value=“ImageAnalysisApp“/>
    </xs:restriction>
  </xs:simpleType>
  <xs:element name=“Tasks“>
    <xs:complexType>
      <xs:sequence minOccurs=“1“ maxOccurs=“unbounded“
        <xs:element name=“Task“>
          <xs:complexType>
            <xs:sequence >
              <xs:element name=“ForensicAppName“
                          type=“ForensicApp“/>
              <xs:element name=“InputPath“
                          type=“xs:string“/>
              <xs:element name=“OutputPath“
                          type=“xs:string“/>
              <xs:element name=“Parameters“
                          type=“xs:string“/>
              <xs:element name=“SequentialTasks“ minOccurs=“0“
                <xs:complexType>
                  <xs:sequence>
                    <xs:element ref=“Tasks“ maxOccurs=“unbound
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

```xml
<?xml version=“1.0“ encoding=“utf-8“?>
<Tasks>
  <Task>
    <ForensicAppName>FileIndexApp</ForensicAppName>
    <InputPath>input1</InputPath>
    <OutputPath>output1</OutputPath>
    <Parameters>default</Parameters>
  </Task>
  <Task>
    <ForensicAppName>KeywordSearchApp</ForensicAppName>
    <InputPath>output1</InputPath>
    <OutputPath>output2</OutputPath>
    <Parameters>default</Parameters>
  </Task>
</Tasks>
```

Fig. 2. An example of the schema XML generated by application manager and the XML workflow configuration file generated by the workflow manager. The file on the left is the schema XML listing all the four applications and the desired structure of the work configuration file; the file on the right bottom is the XML configuration file with two tasks.

Figure 2. This configure file is used by the workflow manger to generate MapReduce drivers and workflow assembly. For more advanced investigators, they can directly write the XML configuration file and use the schema to valid the file. This will reduce the chances of creating an invalid file. In reality, there could be more categories than the example provided.

The application manager provides a set of default categories of the applications, including FileIndexApp, KeywordSearch App, ImageAnalysis App, etc. Users can also add customized tags and categories into the cloud when uploading the new applications. In addition, more tags and supplementary categories could be created by users. Users are allowed and encouraged to rate the applications after using. The ratings are further used for the application recommendation. The applications are sorted from the highest rating to the lowest in the generated XML file. Therefore, highly qualified applications will be presented to users at the top of the candidate application list. The user ratings are the key criteria to evaluate the applications.

The application manager also provides recommendations. Currently, it is community oriented. Each application will be rated by all the users who have tried it. When the application manager generates the schema file, the rating information will be included. Therefore, when users select the application, they are aware of the information that can be used to evaluate the candidate applications.

D. Forensic Workflow Manager

Forensic investigators can send data processing jobs to the cloud. For example, an investigator can specify, the objectives of data processing, the input dataset (stored in the cloud using forensic data manager), and other constraints. The cloud can create a workflow by decomposing the user's request into multiple processing steps. The workflow manager is responsible for setting up, optimizing, executing and reporting the workflow.

1) Workflow Setup: The workflow manager represents a workflow using an XML configuration file. The structure of this XML file is defined in the schema file generated by the application manager. Generally, the schema file contains two kinds of information. One is for all the available applications or software in the “application store”, which are defined in a simple type (xs:simpleType) or a complex type (xs:complexType); the other is the root element structure, called “tasks”. The “tasks” may contain one or more “tasks”, each of which needs the application name, input path, output path, and parameters for execution. All the tasks on the same level are independent and can be executed in parallel. If a user would like to define the dependency between two tasks, the second task should be configured as a “sequential task” of the first task. Figure 2 shows an example. Complex workflows can be also described by assigning the subtasks, which can be recursively built with arbitrary levels of dependencies.To facilitate the procedure of setting up a forensic workflow, the workflow manager uses the schema file to generate a user-

Copyright (c) IARIA, 2013. ISBN: 978-1-61208-271-4

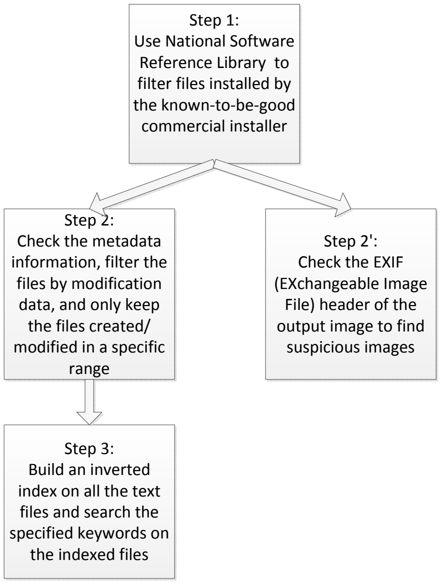

Step 1:
Use National Software Reference Library to filter files installed by the known-to-be-good commercial installer
Step 2:
Check the metadata information, filter the files by modification data, and only keep the files created/ modified in a specific range
Step 2':
Check the EXIF (EXchangeable Image File) header of the output image to find suspicious images
Step 3:
Build an inverted index on all the text files and search the specified keywords on the indexed files

Fig. 3. A Workflow Example Constructed by the Workflow Manager

friendly web portal, which allows forensic investigators to design the workflow and select the desired applications. After designing the workflow, the frontend will pass the workflow to the backend engine. This engine will generate an XML configure file and further generate the Map-Reduce drivers for each step and the necessary synchronization codes (if multiple steps are involved in the workflow) automatically for the forensic investigators. The fewer lines of codes to write, the less chance to generate errors.

2) Workflow Recommendation: Since each step could be completed by multiple candidate software with data dependent performance metrics, the workflow manager will try to make optimal selection/recommendaton of software/workflow and allocate resources accordingly with the objective of achieving the best performance (result quality) for the input dataset with the help of user ratings and the pre-defined workflows. For example, the workflow manager recommends building indices before keyword search. Another example is that by default, the workflow manager will select the National Software Reference Library (NSRL) to filter out the typical contents created by the commercial installer, such as dll, exe, static data. The recommendations are based on the user ratings and evaluation. An example is shown in Figure 3.

3) Workflow Execution: To execute the workflow, the workflow manager allocates processing resources such as elastic machine hours based on an optimized resource plan and assigns workload to the allocated resources using the MapReduce model customized for data intensive forensic computations. Then, the allocated resources execute the assigned tasks on datasets retrieved from the cloud forensic data banks administrated by the data manager. The workflow manager will direct the workflow execution and track the status of each task in the workflow.

4) Workflow Report: Finally, after finishing the workflow, the workflow manager will generate a report to the users. In addition, the workflow manager also stores the status and report in its own database.

IV. Evaluation

In this section, we present the results of a comprehensive evaluation of our system.

A. Experimental Setup

During our evaluation, we deployed a forensic cloud as described earlier using the Amazon' Elastic Compute Cloud(EC2) service. The deployment uses Medium Level-1 (M1) EC2 instances. According to Amazon, these are 64-bit instances with 3.75 GB of memory, 410GB of harddisk and one virtual core containing two EC2 compute units (ECU). One ECU is equivalent to a 1.0-1.2 GHz Xeon processor. The forensic cloud infrastructure is based on Hadoop 0.20 and HBase 0.20, which is managed by Cloudear Manager [28]. The data from a volunteer's hard drive image was uploaded to the forensic cloud. Notice that, the uploading time is not counted and evaluated in the following experiments. This is because, as mentioned previously, the data used are collected from different sources in a distributed way using the cloud as well. We simplified the process by uploading a dedicated image disk for studying purpose. Therefore, the uploading time is not considered.

B. Experimental Results

First, we compared the system outputs and analyzed the performance using the same disk image dataset, which is a working disk image from volunteer users. Figure 4 shows the forensic analysis time on the target image. The image size is 160GB. It shrinks to 10GB after applying the filer operations mentioned in the previous sections. The number of nodes used in the experiment increases from 1 to 10. With more nodes involved, the analysis time is reduced from 21 minutes to only 6 minutes, i.e., 71% of analysis time is saved. However, given a fixed size of test data, the analysis speed can't be further accelerated by adding more nodes. As shown in Figure 4, the forensic cloud with more than 8 nodes has almost the same performance. This is because when more nodes are involved, some of the MapReduce tasks are not executed at the same machine where the data are stored. Copying data between nodes cuts down the benefits. Figure 5 shows the percentage of the MapReduce tasks running locally. The percentage drops from 100% to 40% when the number of nodes changes from 2 to 10. This explains why the speedup of analysis time is only 3. However, when the size of data to be analyzed keeps increasing, more time can be saved, because more data blocks can be processed locally. As shown in Figure 6, when the size of the data increases by 200%, i.e., the size is tripled, the analysis time only increases by 100%. This gives us the clue that the forensic cloud can save more time when dealing with large amount of data.

In the second set of experiments, we compared the lines of codes (LoC) that is needed for the configuration with and without the workflow manager. Figure 7 shows how much effort could be saved in terms of LoC. Workflows with different sequential tasks are built up. Without workflow manager, to configure one workflow task, on average 40 LoCs

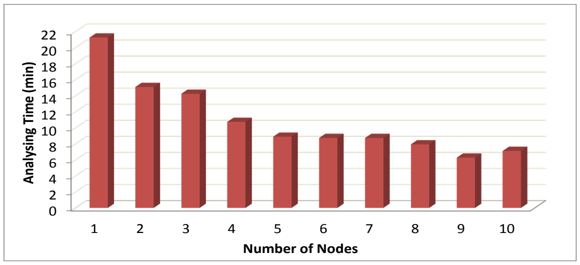


Fig. 4. Analysis Time under Different Number of Nodes in the Cloud

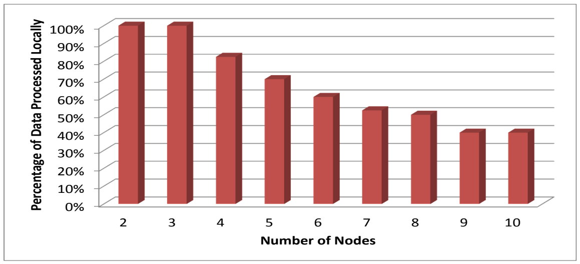


Fig. 5. Percentage of the MapReduce Tasks Processed Locally under Different Numbers of Nodes

are needed, but only 4 LoCs are actually required for the workflow XML file. The LoCs can be reduced by 90% when using the workflow manager to configure a forensic data processing task.

We further compared the performance with and without optimization performed by the workflow manager. We have ten similar tasks, i.e., searching for some keywords, in our experiments. The workflow can intelligently add an extra step of building indices before running all the ten tasks. As shown in Figure 8, the analysis time increases linearly with the number of tasks without the help of workflow management. With the workflow management and optimization, the total time is a little more than the time spent without the workflow management if there is one task executed. However, the total execution time increases slightly when more similar tasks are executed. This is because when the indices are built, further keyword search operations will be accelerated dramatically by the indices stored in the HBase.

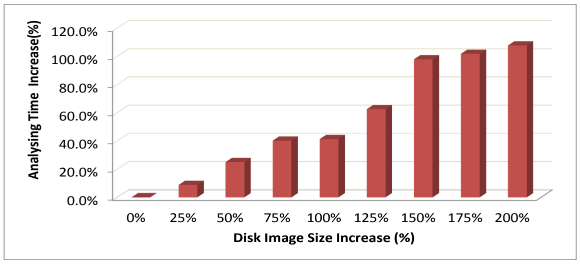


Fig. 6. Percentage of the Increased Analysis Time under Different Increased Image Sizes

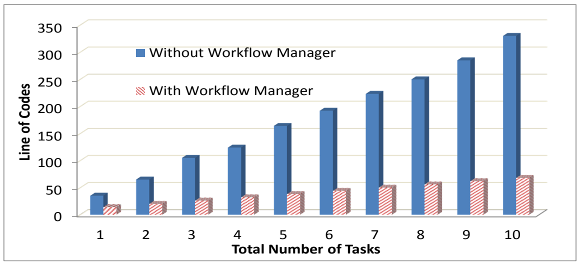


Fig. 7. Line of Codes Needed to Configure Different Tasks in a Workflow w/ and w/o the Dataflow Manager

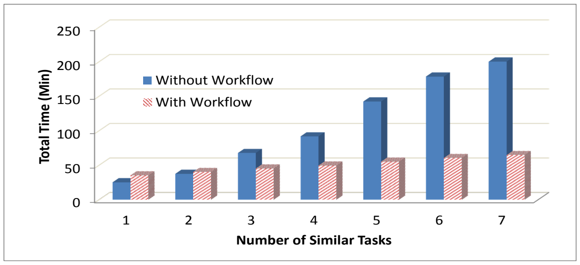


Fig. 8. Total Time Spent w/ and w/o the Workflow Manager's Optimization

V. Conclusions

We proposed and implemented a domain specific cloud environment for digital forensics. We designed a cloud based framework for supporting automated forensic workflow management and data processing. A schema-based forensic workflow framework is proposed. The experimental results show that using the proposed forensic cloud services can save at least 71% of the time with only 10 virtual machine nodes. Meanwhile, the lines of codes for specifying a workflow are also reduced to only 10% when using the proposed workflow management approach. For the investigators, it could be even easier by using the web-based portal, clicking buttons and selecting the desired applications from the dropdown lists. The automated and optimized workflow management approach can save 87% of the analysis time in the tested scenarios. The proposed framework provides valuable insights on designs of domain specific cloud environments using computer forensics as a target field. It demonstrates that, in addition to providing elastic computing resources, cloud can be used as an environment for workflow management and coordinated software development.

VI. Acknowledgement

We would like to thank the reviewers for their comments which significantly improved the paper. This research is partially supported by the Department of Homeland Security (DHS) under Award Number N66001-13-C-3002, and the National Science Foundation under Award Number CNS 1205708. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the opinions or policies of DHS or NSF.

References

[1] A. of Chief Police Officers, “Good practice guide for computer based electronic evidence,” ACPO, Tech. Rep.

[2] K. Kent, S. Chevalier, T. Grance, and H. Dang, “Guide to integrating forensic techniques into incident response,” National Institute of Standards and Technology, Tech. Rep.

[3] J. Dykstra and A. T. Sherman, “Acquiring forensic evidence from infrastructure-as-a-service cloud computing: Exploring and evaluating tools, trust, and techniques,” Digital Investigation, vol. 9, 2012, pp. S90–S98.

[4] “Sleuth Hadoop,” http://www.sleuthkit.org/tsk_hadoop/, retrieved April 2013.

[5] P. Mell and T. Grance, “The NIST definition of cloud computing,” http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf.

[6] J. Erickson, M. Rhodes, S. Spence, D. Banks, J. Rutherford, E. Simpson, G. Belrose, and R. Perry, “Content-centered collaboration spaces in the cloud,” IEEE Internet Computing, vol. 13, September 2009, pp. 34–42.

[7] D. D. Roure, C. Goble, and R. Stevens, “The design and realisation of the myexperiment virtual research environment for social sharing of workflows,” Future Generation Computer Systems, vol. 25, no. 5, 2009, pp. 561 – 567.

[8] I. Foster, “Globus online: Accelerating and democratizing science through cloud-based services,” Internet Computing, IEEE, vol. 15, no. 3, May-June 2011, pp. 70 –73.

[9] S. Caton and O. Rana, “Towards autonomic management for cloud services based upon volunteered resources,” Concurrency and Computation: Practice and Experience, 2011.

[10] S. Distefano, V. D. Cunsolo, A. Puliafito, and M. Scarpa, “Cloud@home: A new enhanced computing paradigm,” in Handbook of Cloud Computing, B. Furht and A. Escalante, Eds. Springer US, 2010, pp. 575–594.

[11] A. Chandra and J. Weissman, “Nebulas: using distributed voluntary resources to build clouds,” in Proceedings of the 2009 conference on Hot topics in cloud computing. USENIX Association, 2009.

[12] S. Xu and M. Yung, “Socialclouds: Concept, security architecture and some mechanisms,” in Trusted Systems, ser. Lecture Notes in Computer Science, L. Chen and M. Yung, Eds. Springer Berlin / Heidelberg, 2010, vol. 6163, pp. 104–128.

[13] “Amazon EC2,” http://aws.amazon.com/ec2/, retrieved April 2013.

[14] Y. Song, H. Wang, Y. Li, B. Feng, and Y. Sun, “Multi-tiered on-demand resource scheduling for vm-based data center,” in Proceedings of the 2009 9th IEEE/ACM International Symposium on Cluster Computing and the Grid, ser. CCGRID ’09. Washington, DC, USA: IEEE Computer Society, 2009, pp. 148–155.

[15] J. Dean and S. Ghemawat, “Mapreduce: simplified data processing on large clusters,” Commun. ACM, vol. 51, Jan. 2008, pp. 107–113.

[16] R. Ananthanarayanan, K. Gupta, P. Pandey, H. Pucha, P. Sarkar, M. Shah, and R. Tewari, “Cloud analytics: do we really need to reinvent the storage stack?” in Proceedings of the 2009 conference on Hot topics in cloud computing, ser. HotCloud’09. Berkeley, CA, USA: USENIX Association, 2009.

[17] “Apache Pig,” http://pig.apache.org//, retrieved April 2013.

[18] “Bulk Extractor,” https://github.com/simsong/bulk_extractor/wiki/Introducing-bulk_extractor, retrieved April 2013.

[19] “FTK (Forensics Toolkit),” http://www.accessdata.com/, retrieved April 2013.

[20] “OSForensics,” http://www.osforensics.com/, retrieved April 2013.

[21] “Intella,” http://www.vound-software.com/, retrieved April 2013.

[22] E. Huebner and S. Zanero, Open Source Software for Digital Forensics. Springer, 2010. [Online]. Available: http://books.google.com/books?id=2gl7k8PbIFYC

[23] “The Sleuth Kit,” http://www.sleuthkit.org/, retrieved April 2013.

[24] K. Shvachko, H. Kuang, S. Radia, and R. Chansler, “The hadoop distributed file system,” in Proceedings of the 2010 IEEE 26th Symposium on Mass Storage Systems and Technologies (MSST), ser. MSST ’10. Washington, DC, USA: IEEE Computer Society, 2010, pp. 1–10.

[25] “Apache Hadoop Wiki-Sequence File,” http://wiki.apache.org/hadoop/SequenceFile, retrieved April 2013.

[26] F. Chang, J. Dean, S. Ghemawat, W. C. Hsieh, D. A. Wallach, M. Burrows, T. Chandra, A. Fikes, and R. E. Gruber, “Bigtable: a distributed storage system for structured data,” in Proceedings of the 7th USENIX Symposium on Operating Systems Design and Implementation - Volume 7, ser. OSDI ’06. Berkeley, CA, USA: USENIX Association, 2006, pp. 15–15.

[27] “National Software Reference Library,” http://www.nsrl.nist.gov/, retrieved April 2013.

[28] “Cloudera,” http://www.cloudera/, retrieved April 2013.