Towards Quality Aware Collaborative Video Analytic Cloud

JongHyuk Lee, Tao Feng, Weidong Shi,
Apurva Bedagkar-Gala, Shishir K. Shah
Dept. of Computer Science, University of Houston
Houston, TX 77004, USA
Email: jonghyuk.lee@daum.net

Hanako Yoshida
Developmental Psychology
University of Houston
Houston, TX 77004, USA
Email: yoshida@uh.edu

***Abstract*—As cloud diversifies into different application fields, understanding and characterizing the specific workloads and application requirements play important roles in the design of efficient cloud infrastructure and system software support. Video analytic is a rapidly advancing field and it is widely used in many application domains (i.e., health, medical care, surveillance, and defense). To support video analytic applications efficiently in cloud, one has to overcome many challenges such as lack of understanding of the relationship and tradeoff between analytic performance metrics and resource requirements. Furthermore, cloud computing has grown from the early model of resource sharing to data sharing and workflow sharing. To address the challenges and to leverage emerging trends, we propose and experiment with a domain specific cloud environment for video analytic applications. We design a cloud infrastructure framework for sharing video data, analytic software, and workflow. In addition, we create a video analytic quality aware resource plan model to guarantee users QoS and optimize usage of resources based on predictive knowledge of video analytic softwares performance metrics and a resource planning model that optimizes the overall analytic service quality under users constraints (i.e., time and cost). The predictive knowledge is represented as input and analytic software specific predictors. The experimental results show that the video analytic quality aware resource planning model can balance the tradeoff between analytic quality and resource requirements, and achieve optimal or near-optimal planning for video analytic workloads with constraints in a resource shared environment. Simulation studies show that resource planning results using ground truth and video analytic performance predictions are very similar, which indicates that our analytic quality/resource predictors are very accurate.**

***Keywords*-Cloud Computing, Video Analytic, Quality Prediction, Planning**

I. Introduction

Cloud computing is emerging as a viable alternative to premise-based deployment of hardware and software solutions. The economies of scale and elasticity that cloud computing offers in an increasingly dynamic and competitive computing climate has garnered rapid adoption in the last few years and as a consequence is quickly altering the practice of how data are stored and processed across business, government, education, and research communities. Cloud computing as a disruptive technology, derives its power in part from recent advances in virtualization technology. Virtualization turns traditional software into appliances and allows them to be deployed and delivered as services in ways that are both massively scalable and elastic. To-date, cloud computing has been primarily concerned with delivering the software and application services required by the conventional business enterprise, such as web servers, database transaction processing, business applications, email services, etc.

As cloud diversifies into different application fields such as science computing and other types of data intensive computing services, new challenges and requirements arise. One such field is that of massive video analytic computing. Driven by the availability and wide adoption of low cost imagers (e.g., webcam, surveillance cameras, medical image sensors, smartphones), the demand for shared and public cloud infrastructure for video analytic applications is growing rapidly. According to studies, internet video will generate over 18 exabytes of data per month in 2013 [1]. Another example, in 2008, there were 4.2 million CCTV cameras in Britain [2]. Efficiently storing, querying, analyzing, and processing these video data presents one of the grand challenges to both the computing industry and research community. Consequently, cloud infrastructures and middlewares tailored for workload from specific domains become a trend in cloud to address the data storage and processing requirements from specific area of applications.

Additionally, cloud computing has grown from the early model of resource sharing to data sharing and workflow sharing. To address the new challenges in domain specific application scenarios and leverage the emerging trends of cloud, we proposed and experimented with a domain specific cloud environment for video analytic applications. The objective of video analytic cloud is to accelerate development and deployment of video analytic applications by allowing engineers, business operators, and scientists to share infrastructure, data, and analytic software.

We designed an infrastructure framework that can enable many new capabilities for practitioners, researchers, and engineers who are involved in different aspects of video analytic. The framework enables an array of new services that are built on top of the concept of domain specific cloud infrastructure that features heterogeneous accelerator environment, video analytic quality aware resource planning, video data sharing, and workflow sharing.

The main contributions of our work include: (i) selection of video analytic software and workflow adaptive to dataset, cloud resources, and user requirements; (ii) video analytic quality aware cloud resource planning; and (iii) a collaborative video analytic cloud environment for sharing data, analytic software, and workflow. To the best of our knowledge, no other studies have proposed a similar solution that integrates knowledge of video analytic software metrics (i.e., quality, resources) with cloud resource management to adaptively combine video analytic workflow placement with cloud resource planning for achieving the best analytic quality under constraints.

Here is the structure of this paper. Section 2 describes background and motivation. Section 3 presents design of our solution. Details of the video analytic aware plan model are provided in Section 4. Section 5 delves into video analytic performance prediction model. Evaluation results and analysis are explained in Section 6. Some related works are discussed in Section 7, and the final conclusions of the paper are presented in Section 8.

II. Background

As cloud continues to evolve responding to highly dynamic and constantly changing service landscape, several trends are emerging. One of them is cloud infrastructure and system support tailored to support specific application domains (i.e., video analytic, data mining, video streaming,

978-0-7695-4755-8/12 $26.00 © 2012 IEEE
DOI 10.1109/CLOUD.2012.141

scientific high performance computing). There are certain advantages of adopting domain specific cloud. These include improved efficiency and the opportunities of intertwining domain knowledge with cloud resource planning and management. The second trend is that, in addition to infrastructure and computing resource sharing, in domain specific cloud, data and workflow from multiple tenants can be shared. Such an open and collaborative environment will bring new capabilities to boost cloud customers' efficiency and productivity to a new level.

A. Video Analytic Domain Specific Cloud

Image processing and video analytic researchers are usually not trained professionals in specific application domains where their designed algorithms may be used. On the other hand, professionals and domain scientists (i.e., security agents, radiologists, healthcare practitioners) often have no rigorous background in advanced image and video processing techniques. The rapid advance of video analytic and image processing techniques has led to prolific availability of task specific algorithms. Nonetheless, their adoption in solving real-world applications has been limited. On the other hand, domain scientists suffer from lack of tools and solutions to address the problems they are attempting to solve.

Researchers and engineers in the past spent huge amount of efforts and resources to develop new video analytic solutions for a wide range of applications (i.e., surveillance, biomedical, education, defense). The common practice for disseminating the research results is through publication or to release them as open source software. However, those traditional approaches remain largely ineffective because of a variety of reasons. For example, it is not efficient and cost effective to ask each research team to re-implement the same algorithm based on the published literature. For open source software, it is often time consuming to learn how to use and configure a new software tool. Many times, non-video analytic experts find it extremely difficult to replicate the infrastructure and platforms that are required to run open source software properly. It is not uncommon that people found open source software incompatible with their development and deployment environment.

On the other hand, much attention has been spent on understanding enterprise types of applications and many solutions have been proposed in the past to integrate application domain knowledge with cloud resource management for achieving optimal trade-off between resource consumption and service quality. However, such efforts haven't been applied to the cloud with focus on video analytic workload. Moreover, analytic applications and workloads differ drastically from applications such as video streaming where the focuses are on video processing, understanding, knowledge extraction vs. content delivery.

B. Video Analytic Algorithm Sharing and Evaluation

The automated interpretation of images/videos to extract semantic information in a timely manner is crucial in numerous video analytic tasks. Over the years, a multitude of approaches and algorithms have been developed. However, most approaches are developed for a specific application and cannot be generalized for all dataset. In fact, no single algorithm can be considered good for all image/video dataset, nor are all algorithms good for a particular image/video dataset. Each algorithm's utility is limited by its specific characteristics that makes it applicable for particular kind of video/image inputs. The fundamental challenge in building intelligent systems is then to provide a generalized framework that is capable of choosing a suitable algorithm from many candidates given a particular video dataset. Our solution of tackling this challenge is to develop a generalized framework and a prototype system that is designed for optimal prediction of vision algorithms performance given the inputs image/video quality and its application to an optimal algorithm selection process. The infrastructure offers an open environment where video analytic software and workflow can be shared and evaluated via a public infrastructure.

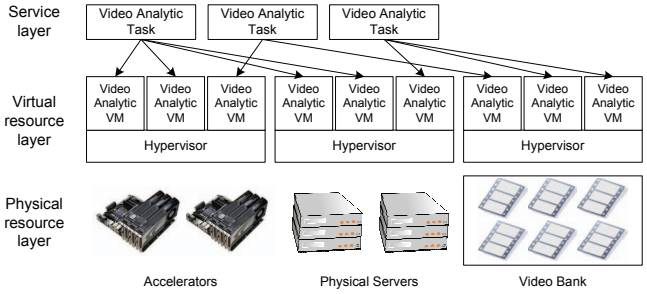


Figure 1. Targeted Cloud Infrastructure for Video Analytic Tasks

III. Design of Video Analytic Cloud

Video Analytic Quality Aware Cloud Infrastructure (VAQACI) aims to delivering the best video analytic processing performance and supporting engineers, scientists, and researchers in sharing video data, video analytic software, and workflow. It is composed of three layers: service layer, virtual resource layer, and physical resource layer as shown in Figure 1.

- Service layer: Service layer includes various services as shown in Figure 2. A user can create analytic tasks for analyzing video/image datasets and send the tasks to the analytic cloud. The user can specify constraints such as cost and time. According to analytic specific predictive knowledge, the cloud will choose the analytic software/workflow and allocate resources accordingly that can achieve the best possible analytic performance quality under the user specified constraints. The video analytic infrastructure can be shared by multiple simultaneously executed tasks;
- Virtual resource layer: Virtual resource layer contains several hypervisors such as Xen and KVM, VMs on top of it. Although VMs actually run over heterogeneous physical resources (e.g., CPU, memory, and network bandwidth), VMs having homogeneous capacities (e.g., computing, storage, and communication) could be provided to upper layer through this layer; and
- Physical resource layer: The physical layer includes physical devices such as accelerators, physical servers, and storage for video bank. A set of VMs can be allocated for serving a particular video analytic task. During task execution planning, it is plausible that VMs allocated for one task are spread across multiple physical servers, while each physical server hosts VMs belonging to different tasks.

A. Video Bank

Video data are complex, unstructured, and voluminous unlike conventional text data. For effective retrieval, video data need to be stored with an appropriate data structure [3] and be analyzed based on not only temporal and spatial but also multiple video streams [4]. In other hands, it is important to determine where and how to process and store video data and its information because analyzing the video data and extracting information from the data are both data-intensive and compute-intensive jobs. Although the quality

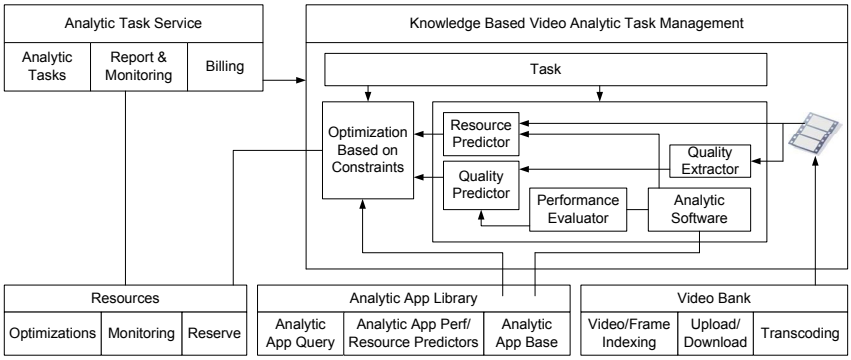


Figure 2. Video Analytic Quality Aware Cloud Services

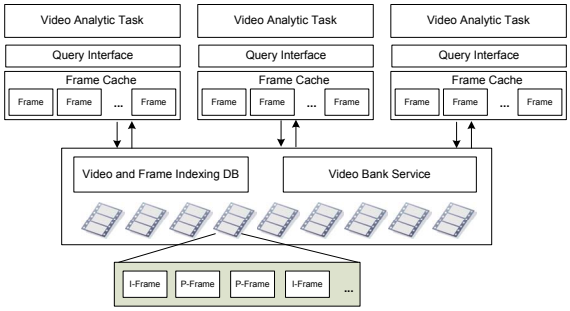


Figure 3. Video Bank

of video analysis results depends on the accuracy of domain knowledge and the kinds of processing algorithms, users demand faster response time of the result to detect unusual events and determine what to do. Therefore, if there is no room to improve the compute time, one can reduce the data transfer time between compute node and data node.

In cloud computing, MapReduce [5] is one of the most popular programming model, which uses a mechanism that a task and its data are co-located in the same node if possible. Currently, Hadoop, an open-source implementation inspired by the MapReduce and Google File System (GFS), is widely used for not only commercial applications (e.g, Facebook) but also scientific computations. HDFS (Hadoop Distributed File System), a Hadoop subproject, is a distributed file system designed to work on commodity hardware. Our video analytic cloud uses a video bank that comprises additional services required for efficient video data retrieval and analysis based on HDFS and database as shown in Figure 3.

Some main challenges of implementing a cloud wide video bank are storage and communication efficiencies. For reducing both storage and data transfer overhead, inter-frame compression should be used when images/video are stored. In inter-frame compression, a frame in a video is expressed in terms of one or more neighboring frames. However, an analytic task typically processes frame by frame on decompressed image inputs. When a video analytic task is distributed over multiple VMs and physical servers, it is inefficient to transfer and duplicate the same video file over multiple nodes. Our analytic cloud uses a two tier storage solution:

- Video frame and result indexing: Because video data files can be too big or too long to be processed by a single node, in our video analytic cloud, video datasets are stored over a distributed file system (HDFS) with searchable frame level indices. In addition, because the processed results are distributed in many nodes, one needs to index them for lookup. VMs for running an analytic task can retrieve frame segments from a video, have the frames decompressed and cached locally. The solution avoids frequent copies of large video files. The frame indices are kept in databases and can be queried;
- Upload and download: Input files divided from a video file are uploaded to several nodes. Output files in nodes are downloaded to the video bank or other nodes for succeeding computations and tasks automatically; and
- Transcoding: Some could wish to convert raw/compressed video data to another encoding standard due to its compressed format, bitrate, resolution, transfer time, etc.

B. Video Analytic Application Library

Video analytic software such as feature extraction, image processing algorithms, object tracking, human tracking, interaction tracking, intruder detection, etc. are created through collaborative processes involving many scientists and researchers. To expedite collaborations among them, it is necessary to reuse the analytic software and workflow. Image processing and video analytic researchers can distribute developed algorithms and software to an analytic software library, "analytic app store" where analytic workflow can be constructed using these software. Professionals, business owners, and scientists can on-demand create, invoke, and deploy analytic tasks using the analytic software and workflow stored in the library (analytic app library in Figure 2). Consequently, the infrastructure will accelerate dissemination and deployment of video analytic and image processing techniques. For each analytic task, selection of video analytic software and workflow is adapted according to the video data set, available resources, and constraints. The analytic library provides metadata for the stored analytic software and algorithms so they can be chosen for fulfilling analytic data processing tasks. Furthermore, for each analytic software, in addition to its purpose and usage context, the analytic app library stores specific quantitative models that can predict the analytic software's performance metrics under a given dataset. The kinds of performance metrics that can be predicted include, resources required for running the analytic software given a dataset, estimated completion time, and quality of the analysis results (i.e., accuracy, detection rate, and false alarm rate). Details of the predictive modeling are provided in Section V with tracking analytic as examples.

C. Video Analytic Tasks

A user can send requests of video analytic tasks to the cloud. For a simple task, the user can specify the kind of analytic processing that should be performed, the input

Table I
NOTATIONS USED IN THE MODEL

Notations	Description
T	A set of task ($T = \{t_1, t_2, ..., t_n\}$)
A_i	Pool of optional analytic software for task t_i ($A = \{a_{i1}, a_{i2}, a_{i3}, ...\}$)
p_i	Priority of task i
V	Video data set ($V = \{v_1, v_2, ..., v_n\}$)
U_i	A set of tasks for user i
Q_{Total}	Quality of all tasks
Q_{Ui}	Quality of analytic results for user i's tasks
Q_i	Quality of analytic results for task t_i
Q_{ij}	Quality of analytic results for task t_i using analytic software a_{ij}
R_{HW}	Total amount of resources that are available
R_{Total}	Total amount of resources used
R_{Ui}	Resources required for user i's tasks U_i
R_i	Resources required for task t_i
R_{ij}	Resources required when running task t_i using analytic software a_{ij}
S_{ij}	0-1 coefficient on whether analytic software a_{ij} is selected
C_{Ui}	Budget of user i
c_{Ui}	Cost constraints for user i's tasks
C_i	Real cost of executing task t_i
C_{ij}	Cost of executing task t_i using analytic software a_{ij}
D_{Ui}	Total execution time of user i's tasks
d_{Ui}	Time constraints for user i's tasks
D_i	Execution time of task t_i
D_{ij}	Execution time of task t_i using analytic software a_{ij}

video dataset, and other constraints such as desired total execution time, maximum cost, etc. The cloud records each task submitted by a user using a XML formatted data structure. Multiple simple tasks can be chained to create a workflow that is composed of multiple tasks. The system represents a workflow using a task graph. After end users create tasks and workflows with specified data inputs, and submit them to the cloud, the analytic task management shown in Figure 2 will process the queued requests. Since each task can be completed by multiple analytic software with data dependent performance metrics, the video analytic task management will try to make optimal selection of analytic software and allocate resources accordingly with the objective of achieving the best analytic performance (result quality) for the input dataset while satisfying all the constraints provided by the user. This process is completed using the performance and resource predictive models that are specific to each implemented analytic software. The next section describes details of this optimization process. After planning, the analytic cloud allocates VMs based on the optimized resource plan and assigns processing workload to the allocated VMs. Then the allocated VMs execute the assigned video analytic tasks on video dataset retrieved from the video data bank.

IV. PLANNING MODEL BASED ON VIDEO ANALYTIC PREDICTIVE KNOWLEDGE

In this paper, we use the notations in Table I to define our model. According to our profiling analysis, the resources consumed by analytic tasks such as execution time and cost, are always proportional to the quality, which indicates that sufficient resources are needed to acquire the best quality. However in practice, given a deadline (d_{Ui}) and budget (c_{Ui}), we may not always achieve the best performance quality of result (Q_{Ui}). Different users may have different requirements and preferences, such as minimizing time or cost, or maximizing result quality based on their specific situations.

To optimize Q_{Ui} based on the deadline (d_{Ui}) and budget (c_{Ui}), we first make the following definitions:

Assume that user k's schedule consists of n independent video analytic tasks, $T_{Uk} = \{t_1, t_2, ..., t_n\}$. User k can set c_{Ui}, d_{Ui}, as constraints where c_{Ui} is desired cost and d_{Ui} is desired execution time. For a task t_i from T_{Uk}, $t_i = < v_i, A_i, p_i >$, where v_i is a video dataset, A_i is the set of applicable analytic software, p_i is a priority number. User i can assign priority number to each task or one priority number to all the tasks.

Resources required by task t_i using analytic software a_{ij}, can be described as $R_{ij} = f_R(t_i, a_{ij})$ where a_{ij} is the selected analytic software from set A_i. Similar to R_{ij}, result quality of task t_i using analytic software a_{ij} is $Q_{ij} = f_Q(t_i, a_{ij})$, where a_{ij} is the selected analytic software from A_i.

Resources required by task t_i can be described as, $R_i = f_R(t_i)$. When $|A_i|$ denotes the number of optional analytic software in A_i, the resources required by task t_i are:

$$R_i = \sum_{j=1}^{|A_i|} f_R(t_i, a_{ij}) * S_{ij}, \tag{1}$$

where S_{ij} is coefficient on whether analytic software a_{ij} is selected. Since one can only choose one analytic software from A_i for a given task. Now, for each task, we have:

$$\sum_{j=1}^{|A_i|} S_{ij} = 1, \tag{2}$$

Let Q_i be the analytic quality for task t_i, then $Q_i = f_Q(t_i)$, which can be expressed further in below:

$$Q_i = \sum_{j=1}^{|A_i|} f_Q(t_i, a_{ij}) * S_{ij}, \tag{3}$$

Let $C_{ij} = f_C(R_{ij})$ and $D_{ij} = f_D(R_{ij})$ respectively be the cost and time for task t_i when it is completed using analytic software a_{ij}. So one can get $C_i = \sum_{j=1}^{|A_i|} C_{ij} * S_{ij}$ and $D_i = \sum_{j=1}^{|A_i|} D_{ij} * S_{ij}$ respectively as the cost and time for task t_i.

For n independent tasks of user k, and for a set of selected analytic software $S = (S_{ij}), i = 1, ..., n$, j is from $|A_1|$, ..., $|A_i|$, total quality Q_{Uk} can be represented as:

$$\sum_{i=1}^{n} Q_i = \sum_{i=1}^{|T_{Uk}|} p_i * \sum_{j=1}^{|A_i|} f_Q(t_i, a_{ij}) * S_{ij}, \tag{4}$$

Whereas the total quality of all users are:

$$Q_{Total} = \sum_{i=1}^{|U_i|} Q_{Ui}, \tag{5}$$

The total resources used by user k's tasks, R_{Uk} can be represented as:

$$\sum_{i=1}^{n} R_i = \sum_{i=1}^{|T_{Uk}|} \sum_{j=1}^{|A_i|} f_R(t_i, a_{ij}) * S_{ij}, \tag{6}$$

Whereas the total resources cost of all users are:

$$R_{Total} = \sum_{k=1}^{|U_k|} R_{Uk}, \tag{7}$$

So, our objective is to maximize the total quality Q_{Total} under the constraints below:

- $R_{Total} \leq R_{HW}$, total resources required less than total resources available;
- For each user, $\sum_{i=1}^{|T_{Uk}|} C_i \leq c_{Uk}$, total cost required less than user k's cost constraints; and

- For each user, $\sum_{i=1}^{|T_{Uk}|} D_i \leq d_{Uk}$, total time required less than user k's time constraints.

For a task i, $f_R(t_i, a_{ij})$ and $f_Q(t_i, a_{ij})$ can be estimated according to models in the next section. Based on the above analysis, we can transform the optimization problem into a 0-1 programming problem with S_{ij} as variables. Normally, a 0-1 programming problem is a NP problem, so we have to use branch-and-bound algorithm to acquire a local optimum. However, when the problem size grows too large beyond certain number of analytic tasks, the processing time may become unacceptable. We use algorithm below to manage the planning task so that the planning process can return within a time limit. The algorithm uses t and n as parameters. As an example, the time threshold t can be set as 1 minute or some other time and the task threshold n can be set as 100.

Algorithm 1 Pseudocode for Resource Planning

```
1: while do
2:   if time passed ≥ time threshold t or number of tasks
     ≥ tasks threshold n then
3:     Run planning software on at most n waiting tasks
       a time;
4:   end if
5: end while
```

V. Video Analytic Modeling

A. Algorithm Performance Prediction

Automated algorithm performance prediction can facilitate applications like optimal algorithm selection or optimal parameter selection that would maximize the probability of successfully completing a particular task. An algorithm's ability to provide desired output is dependent on the behavioral properties of the algorithm, which, in turn depends on the characteristics of the input that the algorithm receives. Thus, prediction of algorithm performance can be achieved by learning a relationship between an algorithm's behavior characteristics and the quality of the input [6], [7]. Performance metrics related to a task can be used as a measure of algorithm's behavioral properties. Objective data metrics including quality features can be used to measure properties of input data and possibly correlated to fulfilling a computational task. Within this context, a performance predictor can be considered an intelligent system that learns the joint model of the performance metrics and input quality metrics to predict the algorithm's response to the input. When the predictor encounters a new input, the knowledge captured during learning phase and the input quality are used to predict an algorithm's ability to succeed, without actual algorithm execution.

B. Tracking by Algorithm Selection

In this paper, as an example, let's consider a specific video analytic task, tracking objects in a given input video. It is well understood that different tracking algorithms perform differently on different datasets and in different environmental settings due to their inability to generalize across different real world scenarios. In order to maximize the probability of successful tracking in a given scenario, we apply a performance prediction based tracker selection technique in order to pick a tracker most suitable given an input video. Three object tracking algorithms, namely connected component tracker, mean shift tracker, and particle filter tracker, are used to realize the predictor and algorithm selector [8]. Each tracker builds on background model proposed in [9]. The connected component tracker uses a linear motion model designed for simple uniform motion and requires a reasonable frame rate to produce acceptable tracks. Mean-shift tracking has the ability to handle slightly more complex motion but still assumes uniform motion. Particle filter tracker employs a non-linear motion model that can better handle complex and slightly abrupt motions. Each of the algorithms also vary in their computational complexity. To assess the quality of an input video, an average measure of quality is computed over all frames. Each frame's quality is characterized by two objective image quality metrics: Signal Activity and Structural Similarity. The signal activity measure proposed in [10] quantifies image blurriness and structural similarity index metric (SSIM) proposed in [11] measures the amount of correlation between consecutive frames. Twelve videos (INRIA sequences) from the CAVIAR database [12] are used in this paper for algorithm selector design and evaluation. CAVIAR sequences are real world surveillance videos depicting various scenarios like people walking, browsing, meeting and fighting. Thus the videos exhibit a high degree of variability in the number of humans in the scene and the type of human motion observed ranging from simple linear trajectories to complex and abruptly changing movement patterns. Hence, the dataset is representative of attributes that would test the strengths and weaknesses of each of the three trackers. Systematic and objective performance evaluation of the tracker's characteristics proposed in [13] is used for performance characterization. We use F-measure [14] to quantify the tracker's correct responses and mistakes. Figure 4 shows the performance of the tracking algorithm with respect to the video quality measures.

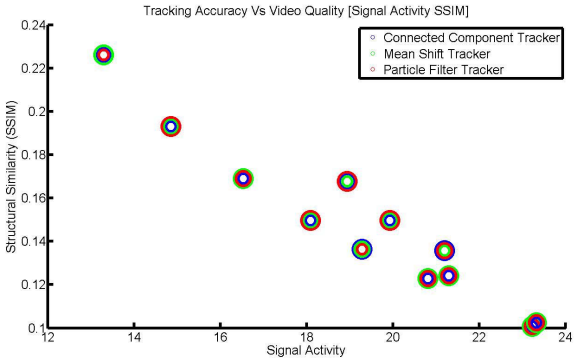


Figure 4. This figure shows the relationship between the video quality measures and tracker accuracy. The size of the circle indicates the tracking accuracy (larger the circle higher the accuracy).

A Neural Network [15] is trained and used as the classifier for performance prediction. Linear Discriminant Analysis (LDA) is a performed on video quality measure vector for subspace transformation to boost linear separability [15]. For each tracker, a predictor is trained. An optimal tracker selection system is simply based on having each of the three predictors estimate the performance accuracy for a given input video and then selecting the tracker with the highest predicted performance. Evaluation of the predictor and the algorithm selection system is based on leave-one-out cross validation across the video dataset and we found the selection accuracy to be 91.67%, i.e. the selector chooses the best tracker of the input video 11 out of 12 times. Figure 5 shows the true tracker rankings and the ranking given by the selector across all videos.

VI. Evaluation

To evaluate the effectiveness of our solution, we used sample video datasets and analyzed the quality and performance under different constraints. The dataset used consists of 10

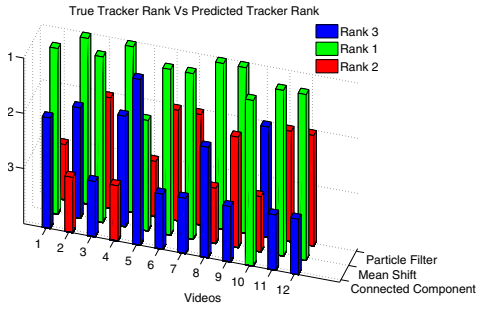


Figure 5. Comparison of True Tracker Ranking and Selector based Tracker Ranking. The y-axis depicts the true rank for each videos, z-axis shows respective trackers, and the color of each bar depicts the predictor based selection rank (green denotes rank-1, red denotes rank-2, and blue denotes rank-3 choice of the selector). The selected tracker to use for an input video is based on rank-1 selection.

Table II
THE ORIGINAL DATA COLLECTED

No.	Size	Signal activities	MSSIM
Video 1	940	12.506	0.27658
Video 2	1180	13.301	0.22609
Video 3	854	14.850	0.19317
Video 4	1228	16.532	0.16906
Video 5	1734	18.096	0.14965
Video 6	1596	19.284	0.13646
Video 7	710	20.808	0.12278
Video 8	448	22.046	0.10948
Video 9	658	23.227	0.10084
Video 10	748	20.335	0.27552

large videos of different sizes, signal activities, and MSSIM (mean structural similarity index matrix) as listed in Table II. For each video, there are three possible softwares (connected component tracker, mean shift tracker, and particle filter tracker) that can be selected for completing the analytic task.

For the purpose of evaluation, we collected execution time, peak memory requirement, cost, and result quality of the three tested algorithms for each of the ten videos as ground truth metric. The results are shown in Figure 6. The results suggest that mean shift tracker can achieve the highest quality for more than half of the tested videos. Particle filter consumes the largest amount of resources among the three and generates the highest quality results for four of the ten videos. Connected component tracker in general consumes the least amount of resources and has the lowest execution time but does not generate the highest quality result for any of the ten videos.

We simulated seven task settings and each consisted of analyzing the dataset comprising of ten different videos, $T = \{t_1, t_2, t_3, t_4, t_5, t_6, t_7, t_8, t_9, t_{10}\}$. We experimented with different constraint metrics for the seven settings. The constraints are shown in Table III, where Machine Hours means the execution time limit for one machine to complete a task setting. For example, Schedule 2's machine hour is 850 hours, which means that it should be finished by one machine in 850 hours or 85 machines in 10 hours. The experiment assumes a service charging policy in which using a 2 GB machine for one hour costs 5 cents. Similar to the quality predictor, we also used resource predictor to predict the resources consumed. For comparison, we computed the global optimal quality using both the quality/resource ground truth and the quality/resources predicted by the predictors.

As shown in Table III, the time constraints for Schedule 1 to Schedule 7 are from low to high. These different

Table III
SEVEN CONSTRAINT SETTINGS

No.	Machine hours
Schedule 1	800h
Schedule 2	850h
Schedule 3	900h
Schedule 4	950h
Schedule 5	1000h
Schedule 6	1050h
Schedule 7	1100h

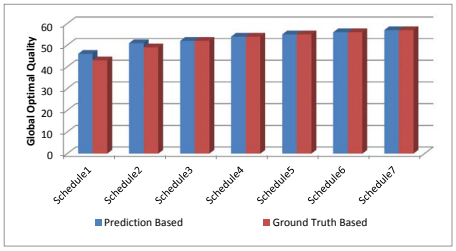


Figure 7. The Global Optimization Result of The Seven Schedules

constraints result in increase of global quality performance for the seven Schedules. The results in Figure 7 indicate that the quality of service is proportional to the amount of investment. In addition, Figure 7 shows quality performances for planning based on the ground truth and planning based on the predictors. One can clearly see that the video analytic quality results delivered by the planning based on the ground truth are very similar to the quality results based on the predictors.

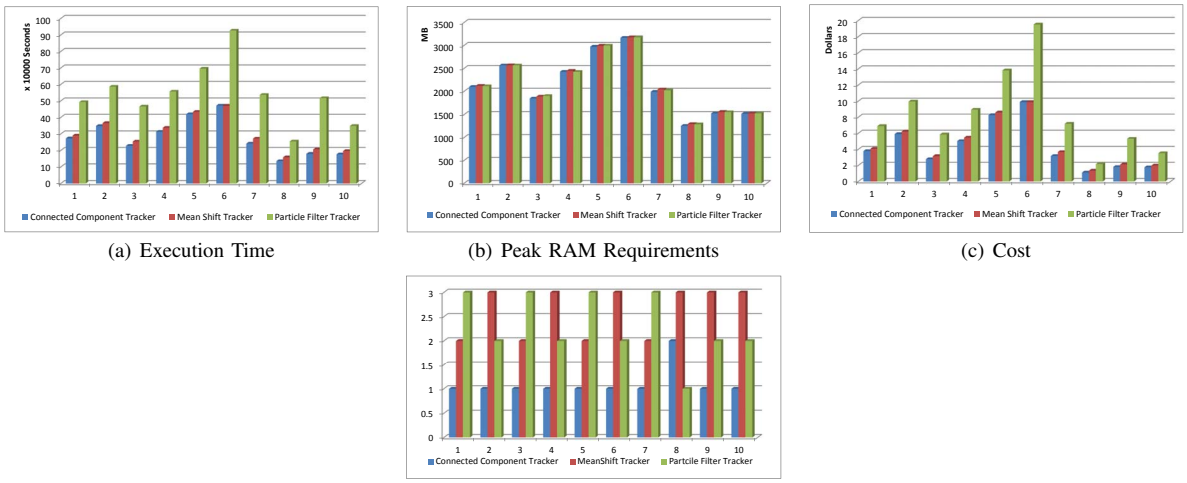
Furthermore, we can simulate task planning for multiple users. Assume that there are two users and each wants to process five of the above described sample videos. The task setting of user 1 is $T_1 = \{t_1, t_2, t_3, t_4, t_5\}$ and the task setting of user 2 is $T_2 = \{t_6, t_7, t_8, t_9, t_{10}\}$. The needs of the two users are different: User 1 wants the task to be completed using less machine hours, and the User 2 wants the highest quality. Our model can take into account these constraints and the results are shown in Table IV.

The planning process takes 30 KB on memory usage and uses less than 1 second time to complete. The simulation results show that our analytic quality and resource aware planning approach can balance the tradeoff between quality and resource requirements in a cloud environment and achieve optimal or near-optimal planning for video analytic workloads with constraints.

VII. RELATED WORK

NIST [16] classified cloud computing into four categories: private cloud, community cloud, public cloud, and hybrid cloud. In these categories, VMs are owned, managed, and operated in different ways. There are even various kinds of community cloud, including: volunteer cloud [17], [18], Nebula cloud [19], social cloud [20], collaboration cloud [21], [22], etc.

In community cloud, there are studies for sharing and collaboration. Erickson et al. [21] represented cloud-based platform and applied three approaches (tools to model content-centric processes effectively, policy-driven storage, and innovative application models) to content-centered collaboration via the Fractal project at Hewlett-Packard Labs because users look for more efficient ways to create, manage, distribute, archive, and repurpose contents generated by themselves and others. Foster [22] described Globus Online that focuses on data-movement function to overcome the



(a) Execution Time (b) Peak RAM Requirements (c) Cost

(d) Real Quality Result

Figure 6. Video Analytic Tradeoff Between Quality, Cost, and Resource Consumption

Table IV
PLANNING FOR TASKS FROM TWO USERS

No.	Prediction Based		Ground Truth Based	
	Machine hour	Quality	Machine hour	Quality
User 1	436h	9	436h	9
User 2	552h	24	603h	26

complexities related to data-intensive, computational, and collaborative scientific research through cloud-based services. Roure et al. [23] showed possibility for social sharing of workflows. Comparing with these work, we concentrate on domain specific cloud infrastructure for practitioners, researchers, and engineers who are involved in various aspects of video analytic processing.

In public cloud, resource management is studied for general-purpose because users have different purposes to run their applications. In general, public cloud such as Amazon EC2[24] does not provide resource management services for users. It just uses a scheduler in Xen hypervisor to schedule VMs. In such a traditional resource management, several nodes are managed by a single scheduler for each service. On the other hand, in the current VM-based data center, there are many VMs in a node (server) and each service is provided via several VMs located in different nodes. Therefore, it is imperative to manage the VMs at the global level because local optimization between services in a server could not always lead to global optimization between services in many servers. Song et al. [25] proposed a multi-tiered on-demand resource scheduling scheme to improve resource utilization and guarantee QoS in VM-based data center. They proposed three-tier correlative schedulers: application-level scheduler, local-level scheduler, and global-level scheduler. The global-level scheduler collects information from all local-level scheduler and returns information for adjusting resource overload back to the local-level scheduler. However, compared to our work, they consider only resource-level scheduling such as increasing or decreasing CPU and memory capacity. The types of workload considered in [25] are mainly web services. To optimize performance and utilization of domain specific applications, it is imperative to provide more customized environments (i.e, private cloud and community cloud). For this reason, Zynga has moved from Amazon Web Services (i.e., public cloud) to its private cloud to maximize the performance and reliability of its social game network [26].

One of the most popular programming model in the cloud is the MapReduce [5], which is for distributed processing of large-scale data on clusters of commodity servers. Hadoop is a software framework inspired by the MapReduce and Google File System (GFS). Because Hadoop does not depend on hardware to execute jobs, it schedules jobs in application-level and handles its failure directly. However a scheduler in Hadoop is not flexible to optimize performance of applications because it uses only a single queue for scheduling jobs. Tian et al. [27] proposed a Triple-Queue Scheduler based on MR-Predict mechanism to increase the utilization rate of both CPU and I/O bandwidth when executing heterogeneous workloads in MapReduce model. They used workload predict mechanism for classifying workloads into three categories (three different queues: CPU-bound, I/O-bound, and wait) based on their CPU and I/O utilization. Zaharia et al. [28] proposed a scheduling algorithm for speculative execution in heterogeneous resources and workloads, called Longest Approximate Time to End (LATE), different from a traditional Hadoop scheduling algorithm for homogeneous resources and workloads. The LATE is based on three principles: prioritizing tasks (workloads) to speculate, selecting fast nodes to run on, and capping speculative tasks to prevent thrashing. Chen and Schlosser [29] implemented three data-intensive and compute-intensive tasks: a large-scale machine learning computation, a physical simulation task, and a digital media processing task using MapReduce model. Based on their experiences, they also proposed additional features to enhance the MapReduce system for supporting wider varieties of applications: indexing, provenance tracking, flexible compositions of components, and optimization strategies. Comparing with these works, we consider an environment that a job could be processed using heterogeneous workflow (or algorithm) according to data and users' constraints.

Several companies actually use cloud-based platform to analyze video and images collected from webcams, surveillance cameras, medical application and there are some studies about it. Yu et al. [4] proposed a distributed real-time video analytic system. They used UIMA (Unstructured Information Management Architecture) that supports infor-

mation data flow control engine (i.e., application form of directed acyclic graph) and multiple commodity databases. The system also used a task scheduler for data locality by pushing down computation modules into database engine to minimize data transfer cost, which is similar to MapReduce model. Comparing with these works, we provide adaptive video analytic software and workflow selection, video analytic quality aware cloud resource planning, and a collaboration environment for sharing data, analytic software, and workflow. Our planning optimization is based predicative models that can predict a video analytic software's performance metrics and resource usage for a given video dataset. To our knowledge, our approach is the first and the only one that incorporates such knowledge into resource and task planning in a shared video analytic cloud environment.

VIII. Conclusions

In this paper, we present a video analytic quality aware cloud infrastructure framework and resource planning model for delivering the best video processing performance under user defined constraints in a resource shared cloud environment. The framework supports engineers, scientists, and researchers for sharing video data, video analytic software, and workflow. In the framework, we design a video bank for effective data storage and retrieval, analytic application "store" for sharing analytic software among professionals, business owners, and scientists, and video analytic tasks for composing workflows. To maximize analytic performance quality under constraints such as time and cost, we propose a video analytic quality aware resource planning model and analytic software dependent quality/resource predictors that can pick the most suitable analytic software for user's tasks. Based on a machine learning approach, for any analytic software, the predictors predict the performance and resource metrics given an input video dataset without actually executing the analytic software. The predictive knowledge is incorporated into our task planning model. In our evaluation, we experimented with task settings under different constraints using real video datasets. The results showed that our quality aware model can balance the tradeoff between video analytic quality and resource requirements, and achieve optimal or near-optimal planning for video analytic workloads with constraints in a cloud environment. Furthermore, the planning results using ground truth and predictions of video analytic performance are very similar, which indicates that our analytic quality/resource predictors are very accurate.

References

[1] Cisco, "Cisco visual networking index: Forecast and methodology, 2010-2015," 2011.

[2] BBC, "Britain is 'surveillance society'," 2006.

[3] P. Sridharan and S. Raman, "Characteristics of video data for signal analysis," in *Proceedings of the3rd International Conference on Signal Processing*, vol. 2, Oct 1996, pp. 1254–1257.

[4] T. Yu, B. Zhou, Q. Li, R. Liu, W. Wang, and C. Chang, "The design of distributed real-time video analytic system," in *Proceedings of the first international workshop on Cloud data management*, ser. CloudDB '09. New York, NY, USA: ACM, 2009, pp. 49–52.

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

[6] A. Gala and S. Shah, "Joint modeling of algorithm behavior and image quality for algorithm performance prediction," in *Proceedings of the British Machine Vision Conference*, 2010, pp. 31.1–31.11.

[7] S. Shah, "Performance modeling and algorithm characterization for robust image segmentation," *International Journal of Computer Vision*, vol. 80, no. 1, pp. 92–103, 2008.

[8] D. A. Forsyth and J. Ponce, *Computer Vision: A Modern Approach*. Prentice Hall, August 2002.

[9] L. Li, W. Huang, I. Y. Gu, and Q. Tian, "Foreground object detection from videos containing complex background," in *MULTIMEDIA '03: Proceedings of the eleventh ACM international conference on Multimedia*, 2003.

[10] Z. Wang, H. R. Sheikh, and A. C. Bovik, "No-reference perceptual quality assessment of jpeg compressed images," in *Proceedings of IEEE 2002 International Conferencing on Image Processing*, 2002, pp. 22–25.

[11] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibilty to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, pp. 600–612, 2004.

[12] "Caviar: Context aware vision using image-based active recognition." 2004. [Online]. Available: http://homepages.inf.ed.ac.uk/rbf/CAVIAR/.

[13] K. Bernardin and R. Stiefelhagen, "Evaluating multiple object tracking performance: The clear mot metrics," *EURASIP Journal on Image and Video Processing*, 2008.

[14] C. J. van Rijsbergen, *Information Retrieval, 2nd edition*. Dept. of Computer Science, University of Glasgow, 1979.

[15] R. O. Duda, P. Hart, and D. Stork, *Pattern Classification (2nd Edition)*. Wiley-Interscience, November 2000.

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

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

[18] 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.

[19] 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.

[20] 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.

[21] 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, pp. 34–42, September 2009.

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

[23] 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, pp. 561 – 567, 2009.

[24] "Amazon EC2," http://aws.amazon.com/ec2/.

[25] 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.

[26] J. Niccolai, "Zynga makes dramatic shift from public cloud," http://www.itworld.com/cloud-computing/250384/zynga-makes-dramatic-shift-public-cloud, 2012.

[27] C. Tian, H. Zhou, Y. He, and L. Zha, "A dynamic mapreduce scheduler for heterogeneous workloads," in *Proceedings of the 2009 Eighth International Conference on Grid and Cooperative Computing*, ser. GCC '09. Washington, DC, USA: IEEE Computer Society, 2009, pp. 218–224.

[28] M. Zaharia, A. Konwinski, A. D. Joseph, R. Katz, and I. Stoica, "Improving mapreduce performance in heterogeneous environments," in *Proceedings of the 8th USENIX conference on Operating systems design and implementation*, ser. OSDI'08. Berkeley, CA, USA: USENIX Association, 2008, pp. 29–42.

[29] S. Chen and S. W. Schlosser, "Map-reduce meets wider varieties of applications," Intel Research Pittsburgh, Tech. Rep.