A SURVEY ON VIDEO ANALYSIS IN SOCIAL MEDIA WITH WEB BASED MOBILE GRID COMPUTING

Mr.Aditya Gaikwad[1], Prof.Rokade M.[2]

[1] PG Student, SP'COE, DUMBARWADI (OTUR), SPPU, Maharashtra, India
[2]Associate Professor, SP'COE, DUMBARWADI (OTUR), SPPU, Maharashtra, India

ABSTRACT

In this project, we first survey the current situation of video processing on the edge for multimedia Internet-of-Things (M-IoT) systems in three typical scenarios, i.e., smart cities, satellite networks, and Internet-of-Vehicles. By summarizing a general model of the edge video processing, the importance of developing an edge computing platform is highlighted. Then, we give a method of implementing cooperative video processing on an edge computing platform based on light-weighted virtualization technologies. Performance evaluation is conducted and some insightful observations can be obtained. Moreover, we summarize challenges and opportunities of realizing effective edge video processing for M-IoT systems.

Keyword: - *M-IoT, SaW, MaaIP, SSCS***.**

---

1. INTRODUCTION

The social media paradigm has led to a significant rise in the volume of user generated content managed by social networks with millions of users accessing services, each of them often using multiple devices at the same time. Service providers aim to engage audience, eager for contents, by boosting the media relevance. To this end, a deep automatic tagging enables better matching of user interests with the content database and reveals underlying connection between items, such as applying face detection mechanisms or content based indexing to find related videos. Image analysis algorithms empower automatic retrieval of salience features but they also involve computing-intensive functions. Therefore, the processing requirements grow substantially when all the media items comprising the social network database are analyzed. Here, on the one hand big data challenges arise when social services have continuously increasing databases, while on the other hand more and more processing resources are required to analyses all the content. To deal with the aforementioned context, this paper introduces a new concept of Social at Work: SaW. It aims to complement a Web-based social media service with all the client devices, mostly mobiles that usually have underexploited resources while accessing the service. SaW proposes a Mobile as Infrastructure Provider (MaaIP) model, going beyond the Infrastructure as a Service (IaaS) model, and Creating a system related to Mobile Grid Computing Finally, the possibility to perform image processing tasks in parallel, such as feature extraction, segmentation, clustering and classification, eases to leap scalability. Due to the video stream nature, composed by individual frames, they can be easily split into independent tasks ready to be distributed. Beyond, the intrinsic presence of key frames in video coding makes easier navigation and selection of representative images. Servers can dispatch the tasks to users' devices where they are run in the background. These backgrounds Web browser applications must balance the mechanism to leverage all available computing resources while provide the best possible user experience. The social media paradigm has led to a significant rise in the volume of user generated content managed by social networks with millions of users accessing services, each of them often using multiple devices at the same time. Service providers aim to engage audience, eager for contents, by boosting the media relevance. To this end, a deeper automatic tagging enables better matching of user interests with the content database and reveals underlying connections between items, such as applying face detection mechanisms or content based indexing to find related videos. Image analysis algorithms empower automatic retrieval of salience features but they also involve computing-intensive functions. Therefore, the processing requirements grow substantially when all the media items comprising the social network database are analyzed. Here, on the one hand big data challenges arise when social services have continuously increasing databases, while on the other hand more and more processing resources are required to analyses all the content. Grid and Cloud technologies provide High Performance Computing systems that aim to satisfy these requirements. However, as pointed in, other under-explored alternatives could enhance the trade-o_ between infrastructure cost, elapsed time and energy saving. It would depend on the number of available processing nodes, the inherent characteristics of the tasks to be performed in parallel and the data volume. To deal with the aforementioned context, this paper introduces a new concept of Social at Work: SaW. It aims to complement a Web-based social media service with all the client devices, mostly mobiles that usually have underexploited resources while accessing the service. SaW proposes a Mobile as an Infrastructure Provider (MaaIP) model, going beyond the Infrastructure as a Service (IaaS) model, and creating a system related to Mobile Grid Computing concept with the available CPU and GPU resources of the different client devices to complement a virtualized cloud server, which provides the social media service. Inspired by the Mobile Grid Computing and the Mobile Cloud Computing (MCC) research fields over a social network mainly based on video content, SaW aims to bring together the huge pool of users permanently connected to media services in social networks and the ever increasing processing capabilities of most of their devices. As a consequence, service providers will embrace the community assets building a device centric grid to improve the social service by means of media analysis. Thus, SaW concept enables service provider to recruit spare CPU/GPU cycles of client devices into an active gear of the social platform, saving cloud resources to the server when the connected clients can perform those tasks.

2. LITURATURE SURVEY

A definition of the Internet-based computing models and focusing on the different topics addressed by distributed computing: the interoperability, the task distribution managing, the parallel processing capabilities and the different data structures.

2.1 Grid Computing

Grid Computing has been an important paradigm in distributed systems for the last two decades. Basically, a grid is a network system where computing tasks are distributed to use non-dedicated computing resources, which may include servers or client computers. The high potential of the nowadays abundant and frequently idle client hardware boosts the opportunistic and delay-tolerant use of client resources in the grid.

2.2 Mobile Grid Computing

Grid Computing is characterized by the heterogeneity of the resources in both amount and nature, by the sporadic availability, churn and unreliability of the devices, and by their anonymity and lack of trust. These issues are more relevant in Mobile Grid Computing (MGC), where computing resources include mobile devices with wireless communications, and therefore prone to disconnections and other eventualities.

2.3 Cloud Computing

More recently, Cloud Computing, a new paradigm of distributed computing where virtualized computing resources are provided on-demand, has experienced a dramatic growth. Now a day the cloud is a cost saving opportunity for many enterprises and many cloud vendors]. Amazon is a popular cloud service provider with solutions like Amazon Simple Storage Service S3 and the Elastic Cloud Computing EC2 as an interface to them. Eucalyptus is an open source cloud implementation on top of Amazon EC2. Being not tied to a specific hardware model, Cloud Computing enables an improved time-to-market for services achieving: a reduced infrastructure deployment time thanks to an increased service availability and reliability; rapid creation of additional service instances; and cloud interoperability, which lets professionals deploy a service on multiple clouds. Thus, cloud computing provides theoretically unlimited scalability and optimized service performance. Since the costs of cloud solutions are a key factor, new models are required to _t better with specific applications, infrastructure environments and business contexts. These new models are classified in three, according to the different virtualization layers: Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS).

3. SaW Architecture

The deployed SaW solution works over client-server architecture (see Figure 1). It improves the architecture presented on towards a hardware accelerated approach, considering all the new aspects introduced by usage of GPU resources within SaW concept. On the server-side there is a SaW Scalable Cloud Server (SSCS) which manages server resources in order to provide a consistent, scalable and a single service front-end to the clients. It deals with balancing the load through the different available servers. The SaW client-side is completely Web browser oriented. Hence, emerging technologies such as HTML5, JavaScript, WebGL or WebCL play a crucial role by providing interoperability to cope with hardware and software heterogeneity. Image Processing Manager: This core layer provides the Web application an API to perform the management of the image processing scripts on the client side. It runs image analysis tasks in the background on top of the JS Injection Manager and using the CPU and GPU processing libraries.

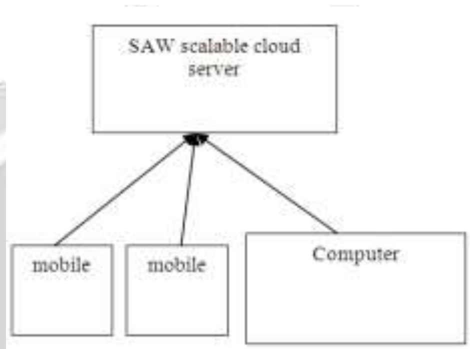

Fig -1: General Saw system architecture diagram

4. SYSTEM OVERVIEW

The deployed Saw solution works over client-server architecture. It improves the architecture presented on towards a hardware accelerated approach, considering all the new aspects introduced by usage of GPU resources within SaW concept. On the server-side there is a SaW Scalable Cloud Server (SSCS) which manages server resources in order to provide a consistent, scalable and a single service front-end to the clients. It deals with balancing the load through the different available servers. The SaW client-side is completely Web browser oriented. Hence, emerging technologies such as HTML5, JavaScript, WebGL or WebCL play a crucial role by providing interoperability to cope with hardware and software heterogeneity. Algorithms 1 and 2 provide an example of SaW with the client device benchmarking process and the SSCS workflow respectively. SSCS executes two concurrent tasks in Algorithm 2. First, by Enroll Thread, SSCS continuously performs a recruitment loop which orders the new devices connected to the social media service to self-assess their performance scores. Thus, SSCS enrolls the devices in the appropriate queue based on the reported type following Algorithm 1, which is executed in client devices remotely and provides, as an outcome, a normalized device type i according to the classification Second, by Task Distribution, SSCS matches the queued image processing tasks to suitable devices in terms of workload and elasticity factors. To this end, the image size and algorithm complexity are considered.

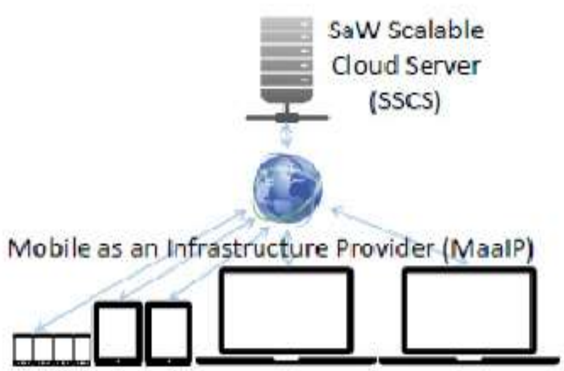

Fig -1: System Design

4.1 Algorithms

Algorithm 1 Device benchmark example

```
procedure BENCHMARK(d_id)                    ▷ assessed at each device
Input: d_id                       ▷ device ID from social media session
      b̂_d                               ▷ estimated bandwidth for device
      F̂_cd                         ▷ estimated CPU processing capability
      F̂_gd                         ▷ estimated GPU processing capability
      i ← getDeviceType(b̂_d, F̂_cd, F̂_gd)
      report i          ▷ send normalised device type to the SSCS
                          following the classification in Table II
```

Algorithm 2 SSCS workflow example

```
procedure ENROLLTHREAD( )                    ▷ SSCS recruitment loop
Data: qD                      ▷ N queues by type of available devices
    for each newSession d_id do                    ▷ new connection
        i ← Benchmark(d_id)                ▷ get type from the device
        queue(qD_i, d_id)                 ▷ queue device based on type

function TRANSFER(Σ_j, Ω_k, d_id, i)          ▷ send task to a device
                                                  of type i
Input: Σ_j                                    ▷ image to be processed
Input: Ω_k                   ▷ programmed image processing algorithm
Input: d_id                       ▷ device ID from social media session
Input: i                                     ▷ normalised device type
Data: qD_i                     ▷ queue with available devices of type i
    deliverTask (Σ_j, Ω_k) to d_id           ▷ deliver task to resource
    waitTTL(i)                  ▷ wait estimated TTL for device type i
    if error then
        return error_msg                                   ▷ error
    if timeout then
        dequeue(qD_i, d_id)              ▷ remove unresponsive device
        return error_msg                                 ▷ timeout
    return ok                                                 ▷ ok

function TASKDISTRIBUTION(Ω_k)               ▷ Tasks dispatching loop
Input: Ω_k                   ▷ programmed image processing algorithm
Data: qΣ                                               ▷ images queue
Data: qD             ▷ N queues with available devices based on type
    while !empty(qΣ) do                ▷ more images in the queue qΣ
        Σ_j ← dequeue(qΣ)                  ▷ next image to be processed
        z ← getTargetDevice(Σ_j, Ω_k)   ▷ target device type under
                                              elasticity factors
        for i = N to z do          ▷ start from more powerful devices
                                      N, stop at target devices z
            if !empty(qD_i) then
                d_id ← dequeue(qD_i)
                Transfer(Σ_j, Ω_k, d_id, i)  ▷ assign task to resource
                if !ok then
                    queue(qΣ, Σ_j)                   ▷ re-queue image
    return completed                                     ▷ completed

procedure MAINLOOP( )                                ▷ SSCS main loop
    EnrollThread()                                  ▷ recruitment loop
    while !empty(qΩ) do            ▷ more algorithms in the queue qΩ
        Ω_k ← dequeue(qΩ)                     ▷ next batch processing
        TaskDistribution(Ω_k)                ▷ tasks dispatching loop
```

5. CONCLUSIONS

In this report we have introduced the concept of Social at Work, SaW, which aims to complement a Web-based social media service with all the idle devices, mostly mobiles, that usually have underexploited resources while accessing the service. SaW proposes a Mobile as an Infrastructure Provider (MaaIP) model, creating a system related to Mobile Grid Computing concept with the available CPU and GPU resources of the different client devices, to complement a virtualized cloud server providing the social media service. Aimed to achieve enhanced and automatic media tagging over social media datasets, SaW fosters background dispatching of media analysis over connected clients, providing a high elasticity and dealing with the availability of the resource Related to the spontaneous presence of users. The computing tasks are embedded in the foreground social Content without draining the users bandwidth or affecting to the perceived Quality of Experience. In harmony with the Presented scenario, delay-tolerant background tasks enable the SaW approach to exchange time-for-resources or time-for money. This means that mobile devices, instead of being as resource intensive as servers, can dedicate the sufficient time to perform the task, preserving according to their capabilities, and saving cloud costs to service provider

6. ACKNOWLEDGEMENT

I want to thanks to my guide Prof.Rokade for her esteemed guidance and encouragement, especially through difficult times. His suggestions broaden my vision and guided me to succeed in this work. I am also very grateful for his guidance and comments while designing part of my research paper and learnt many things under his leadership.

7. REFERENCES

[1]. Mikel Zorrilla, Juli´an Fl´orez, Alberto Lafuente, Angel Martin, Jon Montalban, Igor G. Olaizola, Member, IEEE, and Inigo Tamayo SaW: Video Analysis in Social Media with Web-based Mobile Grid Computing DOI 10.1109/TMC.2017.2766623, IEEE Transactions on Mobile Computing

[2]. Botan I et al. (2010) SECRET: a model for analysis of the execution semantics of stream processing systems. Proc VLDB Endow 3(1–2):232–243

[3]. Salathé M et al. (2012) Digital epidemiology. PLoSComputBiol 8(7):1–5

[4]. Bollen J, Mao H, Zeng X (2011) Twitter mood predicts the stock market. J ComputSci 2(3):1–8

[5]. Chandramouli B et al (2010) Data stream management systems for computational finance. IEEE Comput 43(12):45–52

[6].Chandrasekar C, Kowsalya N (2011) Implementation of MapReduce Algorithm and Nutch Distributed File System in Nutch. Int J ComputAppl 1:6–11

[7].Cioffi-Revilla C (2010) Computational social science. Wiley Interdiscip Rev Comput Statistics 2(3):259–271

[8].Galas M, Brown D, Treleaven P (2012) A computational social science environment for financial/economic experiments. In: Proceedings of the Computational Social Science Society of the Americas, vol 1, pp 1–13

[9].Hebrail G (2008) Data stream management and mining. In: Fogelman-Soulié F, Perrotta D, Piskorski J, Steinberger R (eds) Mining Massive Data Sets for Security. IOS Press, pp 89–102

[10].HirudkarAM,Sherekar SS (2013) Comparative analysis of data mining tools and techniques for evaluating performance of database system. Int J ComputSciAppl 6(2):232–237

[11].Kaplan AM (2012) If you love something, let it go mobile: mobile marketing and mobile social media 4x4. Bus Horiz 55(2):129–139

[12].Kaplan AM, Haenlein M (2010) Users of the world, unite! the challenges and opportunities of social media. Bus Horiz 53(1):59–68.