Multimedia Matching as a Service: Technical Challenges and Blueprints

Jong Hun Choi, Ki-Woong Park, Sung Kyu Park, and Kyu Ho Park
Computer Enfineering Research Laboratory
Korea Advanced Institute of Science and Technology
{jhchoi, woongbak, skpark}@core.kaist.ac.kr, and kpark@ee.kaist.ac.kr

Abstract

With the advancement of digitalization and the availability of communication networks, multimedia matching services such as Picasa, Pudding app for iPhone, and Google Goggles have become increasingly popular services. To realize the services, many vendors constructed their system based on their own ways, which prevents people from taking advantage of high-quality multimedia matching services. The main problem is that they have been unable to share a well-defined multimedia matching library and united multimedia database. In this paper, we introduce the concept of "multimedia matching as a service (MaaS)" for the open and high-quality matching service. In an attempt to adopt the multimedia matching technologies into the cloud system, we have thoroughly investigated the related technology and challenging issues. Consequently, we have reported on our experience on deriving the blueprint to realize the MaaS in cloud computing system.

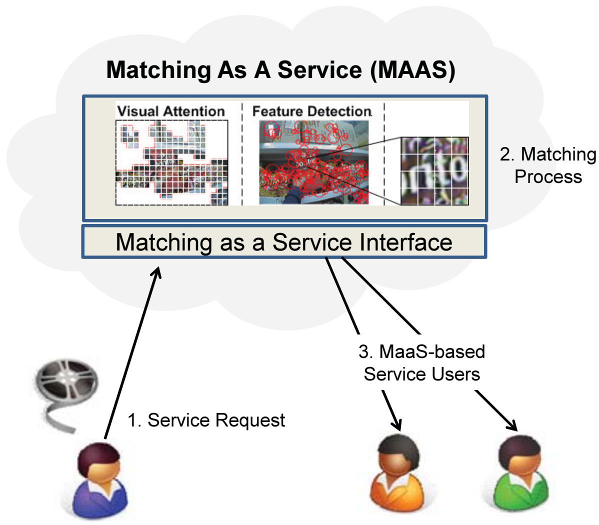


Figure 1. Overall Architecture of Multimedia Matching as a Service: matching cloud node equipped with Bone-V4 [1]

1 Matching Industry and Motivations

By the increasing prevalence of Smartphone such as iPhone and Google Android, people can use a camera with Internet-enabled environment anywhere. Moreover, using Internet broadcast and streaming service, such as YouTube and USTREAM TV, video contents are not only produced by major broadcasting companies, but also private contents producers. A number of multimedia contents are produced and stored at network database so that multimedia matching has become an important technology. Existing major multimedia searching methods highly depend on attached text information. They need inconvenient jobs, and all producers need to integrate tag information with their contents. If users want to find information related with their target multimedia data, the users need to input keywords of that. It hampers the usability of the services.

With the advancement of digitalization and the availability of communication networks, multimedia matching services such as Picasa [2], Pudding app [3] for IPhone, and Google Goggles [4] have become increasingly popular services. To realize the services, many vendors constructed their system based on their own ways, which prevents people from taking advantage of a high-quality multimedia matching services. The main problem is that they have been unable to share a well-defined multimedia matching library and united multimedia database.

In this paper, we introduce the concept of "multimedia matching as a service (MaaS)" for the open and high-quality matching service as shown in Fig. 1. If we provide accelerated cloud service node and well-defined service platform which are designed and worked for multimedia matching, it can solve above problems. MaaS users can exploit high-quality multimedia matching services, shared multimedia databases about service's results, and a well-defined multimedia matching library. In addition, this type of service can act as a core component in various regions which require high-speed pixel analyzing. Not just for service vendors, governments and laboratory can deploy the multimedia matching technology for their public interests and researches. Widely deployed surveillance cameras' and satellite cameras' video stream can be analyzed with this service in a scalable way. Examples include a prevention of crime, missile tracking, observing red enemy, predict a nature disaster, and etc.

As the first step to realize the MaaS in cloud computing system, we have thoroughly investigated the related technology and challenging issue. Consequently, we have reported our experience on deriving challenging issues and the blueprint to realize the MaaS in cloud computing system.

The remainder of the paper is organized as follows: In Section 3, we present the challenges and blueprints to realize the matching as a service in cloud computing system. In Section 5, we present the related work of this study. Finally, in Section 6, conclusion and further works are made about the feasibility of the matching as a service system.

2 Deriving Challenges and Blueprint

Despite the promising technology of digitalization, a number of challenges are identified. By investigating the background technologies from matching technologies to cloud system, we identified five challenges to realize the system as follows:

- How to get the highest possible performance about pixel analyzing
- Management of computing and I/O resource
- Modulation of diverse matching algorithms
- Construction of a well-organized image database set
- Service interface of MaaS

Among the above challenging issues, how to design the highest possible performance to realize the MaaS platform is the most challenging issues. General-purpose GPU, many-core CPU and image recognition chip [1] can be helpful resources. Finding optimized scheduling resources is important. Making some partitions active and other partitions idle is not good for performance. For variety requirements of vendors, we have to offer their desired matching algorithms. Therefore, it is desired to provide a pluggable matching interface which is capable of changing the internal matching algorithm. We must provide sharing multimedia databases, it means that we must control super large-scale data. It will need well-categorized database. Last of all, service interface must be well-defined to make answers to the

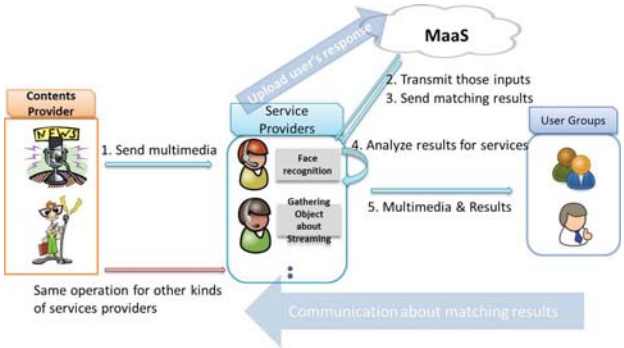


Figure 2. Outlook of MaaS's System

following questions: which part should be offered to service producers; how the contents can be linked between the producer and consumer.

3 Challenge to Realize Matching as a Service

3.1 How to get high performance about pixel analyzing

Many-core CPUs, general purpose GPUs and parallelized pixel analyzing algorithm can be a solution. Maximizing GPU's processing, minimizing main memory accesses, and reasonable number of cores working for pixel algorithm is important stage. Furthermore, specially designed chip for image recognition will enhance machine's performance, using SIFT algorithm [5]. It will be connected with a multimedia matching machine via PCI-E. The multimedia matching chip is worked similar with cell processors. It repeatedly executes SIFT algorithms to input pictures.

Parallelism in many-core CPUs, general purpose GPUs on graphic cards, and recognition chip equipped on our add-on board boost performance.

3.2 Manage computer's resource

Accelerated and optimized computation power does not have the same meaning with fastest 'multimedia matching cloud node'. Network I/O throughput, inner-bus bandwidth and memory's size can be a serious threshold wall. We must consider about how to optimize each components' throughput. Receiving input data and uploading target database's data must occur at the same time. Matching computation must be occurred with receiving next inputs. If matching computation exceeds a threshold wall, we must use a double-buffering technique for input and output data. In other cases, if network I/O or inner-bus's performance is a contention point, inputs and uploaded database's data are distributed to several matching cloud nodes.

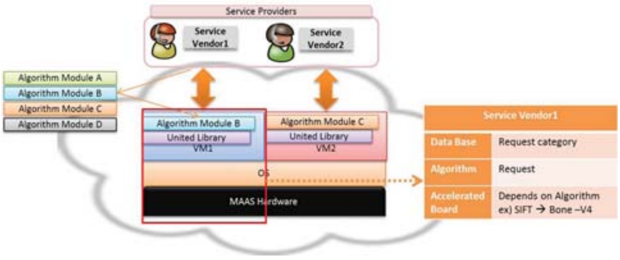


Figure 3. Modulation of Analyzing Algorithms for vendors' request

3.3 Modulation of Diverse Analyzing Algorithms

Users do not need to know about algorithms, they will choose among the services, depending on their types. Other words, service providers must have opportunity to select their needed algorithms. If we make an analyzing software package and operate this for whole vendors, it wastes computer resources. It means that, we need to modulate diverse analyzing algorithms and provide just vendors' needed algorithm. Also, this organization can make a sense with our add-on architecture, each modulated algorithms can be accelerated by our boards. Moreover, it can decrease database accesses. Each vendor's virtual machines do not need to access whole computation results by many kinds of analyzing algorithms, just access to selected algorithm's computation results.

3.4 Construction of Image database set

As previously mentioned, if computer do not have any information about images and videos, it cannot recognized those meaning of contents. For covering this weak point, first, when collecting images, we will use the ordinary method. Using text information, service machines do searching operations in web-page and gather images. Those images will be categorized by searching keywords. After this flow, uploaded images and videos will be matched with database's data and it will put in to the database by matching results. This method can make connections with text information, and visual features. [4] When users give a 'wallet' picture, results will be information about wallet and objects' pictures having similar look with 'wallet'.

3.5 Service interface for MaaS

Already mentioned, if computer do not have any information about images and videos, it cannot recognized those meaning of contents. For covering this weak point, first, when collect images, we will use the ordinary method. Using text information, service machines do searching operations in web-page and gather images. Those images will be categorized by searching keywords. After this flow, uploaded images and videos will be matched with database's data and it will put in to the database by matching results. This method can make connections with text information, and visual features. [4] When users give a 'wallet' picture, results will be information about wallet and objects' pictures having similar look with 'wallet'.

4 Experimental Results

For finding my MaaS' contention parts, I make an example service and analyze those experimental results. At the image matching area, analyzing video streams and finding matching images in DB is a heaviest job.

4.1 Example Service

First step, broadcasting servers broadcast video streams to my MaaS machine. Next step, the MaaS machine execute SIFT algorithm to video streams in real-time. Last step, MaaS machine tranmits matching results and video stream to a client computers.

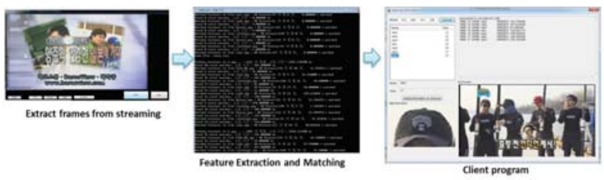
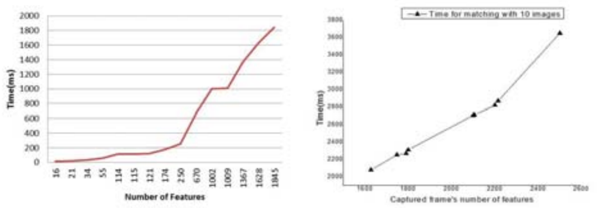


Figure 4. MaaS's Example service : Obeject Recognition at Video Streaming

4.2 Results and Possibilities

MaaS machine use a intel i5 CPU @ 2.67GHz's 1 core for compuation. At this environment, extacting features is consumed on average 1.8 sec, and matching with DB is consumed on average 1.7 sec. At this experiments, we used 10 images for data base. Our target system's DB will contaion at least 2 millions images. if we use the same method, matching with DB will need 30 minutes. Using parallel computation, many-core CPUs and GPUs can be a solution. However, above methods is not enough for real-time services, we must suggest how to decrease matching operations.



(a) Time for extract features (b) Time for mathcing with 10 images

Figure 5. Object recognition at Video streaming : Using i5's 1-core.

5 Related Works

Feature extraction is commonly used in image recognition. Especially in this area, David Lowe's Scale Invariant Feature Transform (SIFT) algorithm is repeatedly mentioned [5]. A number of engineers try to accelerate SIFT. Previous works on accelerating SIFT with many-core CPU, Zhang, Chen, et. al. published a paper. They optimized OpenMP's parallelization, cache works, and SIMD operations [6]. Many authors explored accelerating SIFT with GPUs. Warn , Emeneker, et. al. accelerated and implemented an SIFT using OpenMP parallelization and GPU execution [7]. Heymann, Muller, et. al. optimized and implemented SIFT on General-purpose GPU. With this method, feature-tracking in real-time can be possible only with GPUs and meanwhile the CPU can do other tasks [8]. Moreover, other authors explored increasing SIFT's operation speed with a well-designed chip. Lee, Park, et. al. published SIFT processors. Using a heterogeneous many-core object recognition process, object recognition in real-time can be possible, when processing 30 fps 640*480 video [1]. In image categorization region, Wu, Jiang, at. al. suggested "Vicept" [9]. They propose a method to interpret an image into its semantic words.

6 Conclusion

In an attempt to adopt the multimedia matching technologies into the cloud system, we have thoroughly investigated the related technology and challenging issues. Consequently, we have reported on our experience on deriving the blueprint to realize the MaaS in cloud computing system. By investigating the background technologies from matching technologies to cloud system, we identified five challenges to realize the system as follows:

- How to get high performance about pixel analyzing
- Manage computer's resource
- Modulation of Diverse Analyzing Algorithms
- Construction of Image database set
- Service interface for MaaS

As the next step of this study, we plan to expand our experiences and continue to work on going forward to realize the multimedia matching as a service.

7 ACKNOWLEDGMENT

The work presented in this paper was supported by MKE (Ministry of Knowledge Economy, Republic of Korea), Project No. 10035231-2010-01.

References

[1] S. Lee, J. Oh, J. Park, J. Kwon, M. Kim, and H.-J. Yoo, "A 345 mw heterogeneous many-core processor with an intelligent inference engine for robust object recognition," *Solid-State Circuits, IEEE Journal of*, vol. 46, no. 1, pp. 42 –51, 2011.

[2] Google, "Picasa. photo tagging,categorizing service," Google., http://picasa.google.com/, April 2011. [Online]. Available: http://picasa.google.com/

[3] KT, "Pudding," Apple., http://itunes.apple.com/us/app/id379411152, April 2011. [Online]. Available: http://itunes.apple.com/us/app/id379411152

[4] Google, "Google goggle," Google., http://www.google.com/mobile /goggles/, April 2011. [Online]. Available: http://www.google.com/mobile/goggles/

[5] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vision*, vol. 60, pp. 91–110, November 2004. [Online]. Available: http://portal.acm.org/citation.cfm?id=993451.996342

[6] Q. Zhang, Y. Chen, Y. Zhang, and Y. Xu, "Sift implementation and optimization for multi-core systems," in *Parallel and Distributed Processing, 2008. IPDPS 2008. IEEE International Symposium on*, 2008, pp. 1 –8.

[7] S. Warn, W. Emeneker, J. Cothren, and A. Apon, "Accelerating sift on parallel architectures," in *Cluster Computing and Workshops, 2009. CLUSTER '09. IEEE International Conference on*, 312009-sept.4 2009, pp. 1 –4.

[8] S. Heymann, B. Frohlich, F. Medien, K. Muller, and T. Wiegand, "Sift implementation and optimization for general-purpose gpu," in *In WSCG 07*, 2007.

[9] Z. Wu, S. Jiang, L. Li, P. Cui, Q. Huang, and W. Gao, "Vicept: link visual features to concepts for large-scale image understanding," in *Proceedings of the international conference on Multimedia*, ser. MM '10. New York, NY, USA: ACM, 2010, pp. 711–714. [Online]. Available: http://doi.acm.org/10.1145/1873951.1874059