# Ameba: A High-performance and Energy-efficient Online Video Retrieval System

Jin Yang *, Jianmin Pang, Jintao Yu
State Key Laboratory of Mathematical Engineering and Advanced Computing
Zhengzhou, China
* ysire@163.com

Wei Cao
State Key Laboratory of ASIC and System
Fudan University
Shanghai, China

***Abstract*—This paper describes a high-performance and energy-efficient online video retrieval system called Ameba. Ameba contains a number of custom reconfigurable nodes which are grouped by a novel architecture. The system aims to address the performance and energy efficiency issues for large scale dynamic concurrent query requests. The openSURF approach and a Hamming distance based matching algorithm were implemented and improved on FPGA to increase the performance and energy efficiency. A predictive algorithm is proposed to forecast the trends of online query requests. This scalability which is obtained from the dynamic reconfiguration, makes the system have ability to improve its energy efficiency with the guarantee of performance. The simulation experiments are conducted with a considerable library which consists of 4650 videos with a combined length of more than 3000 hours. The comparative results demonstrate that Ameba has performance and energy efficiency advantages when facing large scale online video retrieval requests.**

***Keywords-video retrieval; energy efficiency; reconfiguration; green computing;***

## I. INTRODUCTION

With the development of We-Medias and video sharing websites, e.g. YouTube and YouKu, multimedia resources, particularly online videos are increasingly becoming the "biggest big data" on the Internet. Online video retrieval, especially online content-based video retrieval is becoming more and more necessary to the web users. However, it is not easy to retrieve a given video from the vast amount of big data with the limits of performance (also known as precision, recall, response time) and energy efficiency. Most existing video retrieval systems pay much attention to handle various photometric or geometric transformations, but difficult to handle the Web scale video database and return the search results in real time [8].

The IDC, where the retrieval system is commonly deployed, often connects hundreds to thousands of commodity CPU based computers in networks to build a cluster system for saving costs [2]. However, there are also some disadvantages: a) the computing nodes making up of the cluster have very low reconfigurable capacity. When in the low load state for a long time, the node cannot change its composition, such as disabling a running main memory slot to bring the node's power down. b) The whole cluster also cannot scale itself rapidly according to the change of the system's total load. The load of online video retrieval system changes periodically with the number of people surfing the Internet, while the video retrieval system supplies nearly the same capability and cost all over the running time.

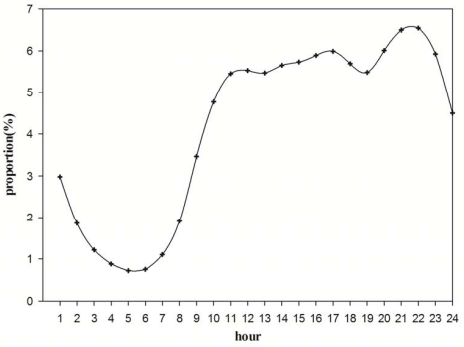


Figure 1. Distribution of web users in a day

This paper describes an online video retrieval system called Ameba which has both high energy-efficiency and enough computing performance, intending to address the two issues above mentioned by the following approaches. Firstly, we build a more reconfigurable node which consists of 4 FPGAs and 1 low-power CPU. The CPU is used only for starting the FPGAs, loading bit streams to FPGAs and powering on/off them, playing quite different roles with the other heterogeneous systems. FPGA's dynamic partial reconfiguration ensures the node can adjust most elements' states to make node's power and performance match the workload. Secondly, in order to scale the system well and truly, a local regression algorithm is adopted to predict the system's load. Figure 1 shows a fitting curve illustrating the statistics of the traffic of over 1.5 million websites in the year 2013 [1].

## II. THE AMEBA SYSTEM

### A. Architecture

There are three kinds of commonly used computing devices alternatively when we build a commodity cluster. These are CPU, GPU and FPGA. Each of them has own different characteristics [3]. The Ameba system, a video retrieval system,

[1] The data is sourced from tongji.baidu.com, which is the biggest analysis platform for Chinese website.

978-1-4799-8688-0/15 $31.00 © 2015 IEEE
DOI 10.1109/BigMM.2015.72

whose most computational amount and complexity are comparing the query video's features with these in the video feature library one by one. For this job, the cluster formed by FPGAs is able to control itself finer-grainedly for energy consumption. Moreover, the dynamic partial reconfiguration function of the FPGA, is a more important factor that urges us to choose it.

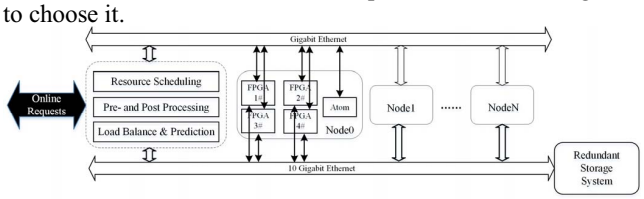


Figure 2. Architecture of the Ameba

According to [1], UNNS(Uniform Node Nonuniform System) and NNUS(Nonuniform Node Uniform Systems) are the two methods to group together the elements of a reconfigurable cluster. This work adopts an improved NNUS architecture, which has the flexible advantage of the UNNS from the logical point of view. Figure 2 presents an overview of the architecture of the Ameba system. The nodes which are nonuniform from 0 to N compose the whole cluster which is uniform, but the topologies of the networks determine that all FPGAs and all CPUs are separated into different network arrays, one for transferring computing data and another used for transferring control and management data. And grouping different types of PEs into different nodes is what the UNNS approach tries to do. In short, the improved NNUS architecture has both the low communication overhead of the NNUS and the finer-grained flexibility of the UNNS.

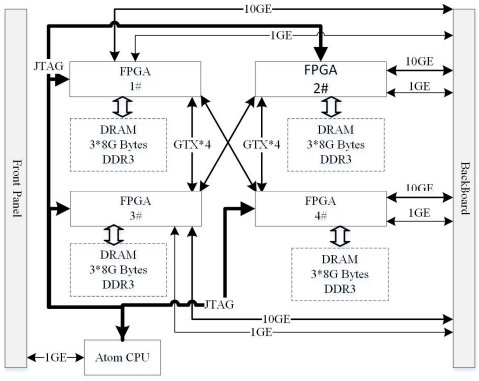


Figure 3. Node Architecture

## B. Hardware

Figure 3 provides an overall structure of each node. As it shows, there are four Xilinx Virtex-6 LX475T FPGAs and one Intel Atom CPU in a single node. Each node is a full scale server in 2U size with local disk storage attached to the Atom CPU. Every FPGA owns three 8G Bytes DDR3 memories. Data communications between FPGAs in one node are managed by four dedicated GTX*4 channels and those in different nodes via a 10-Gigabit Ethernet. The Gigabit Ethernet which connects all the FPGAs and CPUs is used to transfer the runtime state messages and command messages intended for the management and control of the PEs. The Atom CPU in each node is responsible for managing and controlling the four FPGAs in this node, and resources scheduling server manages all the Atom CPUs directly and other resources indirectly through the Atom CPU. The Atom CPU and the FPGAs in this node are also connected by a daisy chaining network which is dedicated to downloading bitstream files to the FPGAs.

The dedicated high speed channel between FPGAs intra-node and the large amount of distributed memory associated with each PE are two characteristics of the Ameba node. When the Ameba meets a heavy task that the videos to be retrieved are relevant to each other, the FPGA don't have no need to compete to request for reading the video features from the common feature library every time. The FPGAs in a node can exchange the feature data in a circular transferring chain to maximize the benefit of the data locality.

## C. Software

There are mainly four parts of the software in the Ameba system: the load query and balance modules aiming to query every running FPGA for its load state and balance the load among the FPGAs; the load prediction and reconfiguration modules that is used to predict the total system load according to the history of online requests and reconfigure the system when needed on the basis of the prediction; the video retrieval module used to retrieve the video feature in the very large scale of video feature library, which is the place spending most of time in the whole process; the video preprocess and postprocess modules which extract a series of interest features from the request video and summarize the retrieval results to return to the requester. Seeing from the system level, the video retrieval module runs on the FPGAs in each node, and other programs are targeted on the Atom CPUs and a Xeon X5650 6-core CPU in an IBM X3650 server. Besides, every Atom CPU and the IBM server run a copy of Linux Operating System. The IBM server can control each node by a management system based on the Open IPMI and the BMC module. Except the video retrieval module, all other three modules above are CPU execution built using the standard GCC complier.

The method of the Ameba system adopted to process the video retrieval is called content-based copy retrieval (CBCR) which aims at finding all the modified versions or the previous versions of a given candidate object [4]. A standard CBCR always contains three primary steps: a local features extraction, an approximate similarity search technique, and a post-process step based on a registration algorithm and a vote. Since mainly focusing on the green computing of the Ameba, this work won't describe the details of these steps, and only show some principle points as follow. Among the three steps, the first two cost most of time the whole process spends, and hence, we implement both of them on FPGA intending to reduce the execution time. The features extraction algorithm is based on the openSURF [5] library which is an open source implemented in C++ language. The details about the openSURF implementation on FPGA are shown on [6]. This work used a binary value vector and the Hamming distance [9, 10] to calculate the approximate similarity between two documents. If the Hamming distance between the two bit-sequence is smaller than a specified threshold, the technique regard them matching successfully. To calculate a global similarity and decide if the more similar documents are copies of the candidate document, a registration plus vote strategy is

exploited to perform that [4]. The post-process step is designed running on the X5650 CPU.

**Algorithm 1 Predictive Algorithm**

```
//zeros(M,N): Get a M*N Zero matrix
//length(x): Get length of x
//abs(x): Get absolute value of x

Input: x=[x1,x2,...,xn,xnext]   //time sequence
       y=[y1,y2,...,yn]         //history information sequence
Initialize: rmax = num          //domain radius
            x_f, f equals empty arrays
            delta=(xnext-x1)/num

for i=0:num
    x_val=x1+i*delta;
    x_f=[x_f, x_val];
    A=zeros(2,2);
    B=[];
    for j=1:length(x)
        s=abs(x(j)-x_val)/ rmax;
        if s<=0.5
            w= 2/3-4*s^2+4*s^3;
        elseif s<=1
            w = 4/3-4*s+4*s^2-4*s^3/3;
        else
            w=0;
        endif
        A = A + w*[1;x(j)]*[1,x(j)];
        B = [B,w*[1;x(j)]];
    endfor
  f =[f,[1,x_val]*inv(A)*B*y'];
endfor

Output: ynext=f[length(f)-1]
```

The predictive algorithm called Moving Least Squares (MLS) [7] is implemented in algorithm 1, where *w* represents the weight function which is a cubic spline function. If define $r_{\max}$ as the radius of the domain and $s' = \| x - x_i \|$, then $s = s'/r_{\max}$. One important thing to note about the algorithm is that the matrix $A$ should be nonsingular. To achieve this, $r_{\max}$ should be big enough. However, too big value of $r_{\max}$ will reduce the locality of the function $f$. It is suitable to choose a reasonable $r_{\max}$ according to the specific problem.

## III. EXPERIMENTS

### A. Experiment Setup

Experiments are simulated in the Ameba system which contains 10 nodes as mentioned above. The dataset is built by ourselves. It consists of 1027 high quality videos and 3623 low quality videos (totally 4650 videos) with a combined length of more than 3000 hours. All of them are downloaded from YouKu, TuDou and other free download websites. There are some particular demands here. First, our experiments need the duplicate transformations done by the real users. Another is that we need a more scalable dataset to meet the requirement of the big data.

Ten high quality videos and ten low quality videos are randomly selected and transformed by totally six factors of the TV logo, subtitle, resolution, brightness, contrast and aspect ratio. As a result, 120 videos are generated to prepare for evaluating the recall and precision. Also 100 near-duplicate videos from the 3623 low quality set and other 100 mixed videos combined with 120 videos (totally 320 videos) are prepared to test the performance and energy-efficiency. These 320 videos will be repeatedly sent many times to simulate the concurrent online query requests in Figure 1.

### B. Metrics

According to [4], recall and precision metrics are defined as:

$$Recall\ r_c = \frac{number\ of\ true\ positives}{total\ number\ of\ true}$$

$$Precision\ p_r = \frac{number\ of\ true\ positives}{total\ number\ of\ positives}$$

We use the Application Energy Efficiency (AEE) to measure how efficiently a system execute an application. The AEE is defined as:

$$AEE = total\ tasks\ processed / total\ energy\ consumed$$

### C. Result

#### 1) Recall and Precision evaluation

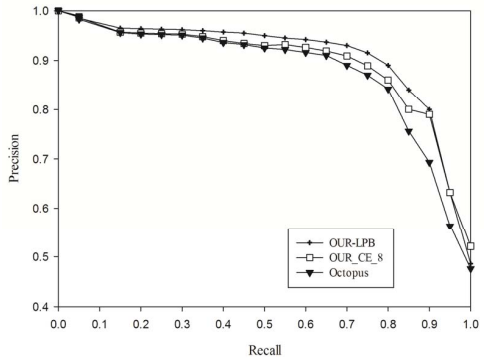


Figure 4. Averaged precision/recall graphs

The experiment is conducted on a single node with the 120 transformed videos mentioned above. The recall and precision ratios presented in Figure 4 are the average values of these 120 experimental results. Besides, Table I also shows a comparison result of our system with the methods in [8] from the MAP, time and energy. It can be seen that the Ameba system has

better performance than other methods in the aspect of response time. However, both the OUR_LBP and OUR_CE_8 have better PR curves than the Ameba. The main reason is that we implement the FPGA configuration with fixed point. Some truncation error may propagate and accumulate and lead to matching performance degradation. However, according the results, though the MAP of the Ameba is worse than other methods, the overall accuracy of retrieval is acceptable as an online video retrieval system.

TABLE I. COMPARISON ON TIME, MAP, PERF AND AAE

| Methods | MAP | Time(s) | Perf. (FPS) | AAE (FPS/W) |
|---|---|---|---|---|
| OUR_LPB | 0.953 | 3.7×10-3 | - | - |
| OUR_CE_8 | 0.950 | 3.6×10-3 | - | - |
| Ameba | 0.908 | 1.7×10-4 | 7403.83 | 32.8346 |

*2) Energy efficiency evaluation*

As far as we know, there are not energy efficiency evaluation for a video retrieval system in the previous works. We evaluate the energy efficiency of the Ameba system in two aspects. The first one is the energy efficiency in the dynamic reconfiguration mode comparison with itself in the static configuration mode. The Second one is the comparison with a cluster composed of 10 IBM X3650 servers in the dynamic mode. The programs executed on these servers use the same algorithm with FPGA.

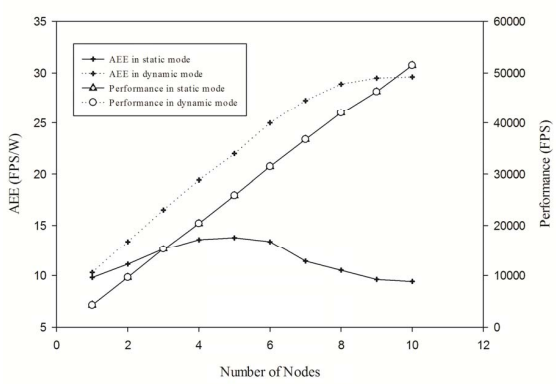


Figure 5. Comparison between dynamic and static

The first comparison is shown in Figure 5. It is can be seen that the energy efficiency in dynamic reconfiguration mode is far better than the efficiency in the static configuration mode. At the same time, the performance of both the modes is similar. That is to say, the Ameba system uses the less resource to gain the same performance with the scenario that all resource are used. In the current simulation of query requests, the Ameba system can achieve 3 times benefit compared with a static configuration system.

Table II presents the result of the second comparison. It shows that although the server cluster is also scaled with the change of query requests, its energy efficiency can't exceed the Ameba. The reasons are based on the following aspects. Firstly, the performance and power of the CPU based server is much worse than the node based on FPGA. Secondly, it takes a longer time to reconfigure a CPU based server than a FPGA. As a result, the Ameba increases the energy efficiency by 47 times compared with a dynamic CPU based cluster.

TABLE II. COMPARISON BETWEEN AMEBA AND CLUSTER

| Number of Nodes | AEE of CPU Based Cluster (FPS/W) | AEE of Ameba (FPS/W) |
|---|---|---|
| 2 | 0.28 | 13.4 |
| 4 | 0.48 | 19.43 |
| 6 | 0.53 | 25.02 |
| 8 | 0.60 | 28.88 |
| 10 | 0.63 | 29.62 |

## IV. CONCLUSION

We describe an energy-efficient online video retrieval system built on the reconfigurable hardware in this paper. The system aims to supply the video retrieval service to the large scale Internet users. It is a content-based video retrieval system which implements the openSURF algorithm and matching algorithm based on Hamming distance on FPGA. The two time consuming phase of retrieval can be speedup by the parallelism of FPGA and the high-speed communication networks. A predictive algorithm based on the MLS method is used to forecast the trend of the query requests. The experimental results show that this video retrieval system has acceptable recall, precision and excellent response time and energy efficiency.

## ACKNOWLEDGMENT

This work is supported by National Nature Science Foundation of China (Grant No. 61472447), Key Project of Major Program of Shanghai Committee of Science and Technology under Grant (No. 13DZ1108800).

## REFERENCES

[1] T. El-Ghazawi, E. El-Araby, M. Huang, K. Gaj, V. Kindratenko, and D. Buell, "The promise of high-performance reconfigurable computing," IEEE Computer, vol. 41, pp. 69-76, 2008.

[2] K. H. Tsoi and W. Luk, "Axel: a heterogeneous cluster with FPGAs and GPUs," in Proceedings of the 18th annual ACM/SIGDA international symposium on Field programmable gate arrays, 2010, pp. 115-124.

[3] B. Betkaoui, D. B. Thomas, and W. Luk, "Comparing performance and energy efficiency of FPGAs and GPUs for high productivity computing," in Field-Programmable Technology (FPT), 2010 International Conference on, 2010, pp. 94-101.

[4] A. Joly, O. Buisson, and C. Frelicot, "Content-based copy retrieval using distortion-based probabilistic similarity search," Multimedia, IEEE Transactions on, vol. 9, pp. 293-306, 2007.

[5] C. Evans, "The opensurf computer vision library," 2012.

[6] X. Fan, C. Wu, W. Cao, X. Zhou, S. Wang, and L. Wang, "Implementation of high performance hardware architecture of OpenSURF algorithm on FPGA," in Field-Programmable Technology (FPT), 2013 International Conference on, 2013, pp. 152-159.

[7] P. Lancaster and K. Salkauskas, "Surfaces generated by moving least squares methods," Mathematics of computation, vol. 37, pp. 141-158, 1981.

[8] L. Shang, L. Yang, F. Wang, K.-P. Chan, and X.-S. Hua, "Real-time large scale near-duplicate web video retrieval," in Proceedings of the international conference on Multimedia, 2010, pp. 531-540.

[9] J. Oostveen, T. Kalker, and J. Haitsma, "Feature extraction and a database strategy for video fingerprinting," in Recent Advances in Visual Information Systems, ed: Springer, 2002, pp. 117-128.

[10] M. L. Miller, M. A. Rodriguez, and I. J. Cox, "Audio fingerprinting: nearest neighbor search in high dimensional binary spaces," Journal of VLSI signal processing systems for signal, image and video technology, vol. 41, pp. 285-291, 2005.