IEEE International Symposium on Industrial Electronics (ISIE98), Pretoria, South Africa, July 7-10, 1998.

A Parallel Model for Multimedia Database on Cluster System Environment

Punpiti Piamsa-nga, Sanan Srakaew,
Nikitas A. Alexandridis, and George Blankenship
Department of Electrical Engineering and Computer Science
George Washington University, Washington D.C. 20052,
USA
{punpiti, srakaew, alexan, blankeng}@seas.gwu.edu

***Abstract* – In this paper, we propose a parallel model for content-based retrieval form a multimedia database using a heterogeneous cluster system. The proposed parallel unified model [6] was used to represent multimedia data. All types of multimedia data in the unified model are represented by k-dimensional signals. Each dimension of k-d data is separated into small blocks and then formed into a hierarchical multidimensional tree structure, called a k-tree. The parallel version of k-tree model was introduced in [7][8]. The previous experimental results show the huge reduction of retrieval time on a cluster of homogeneous workstations. In this paper, we extend our parallel model to a heterogeneous cluster system environment. We demonstrate the experimental results of using a parallel retrieval algorithm for the k-tree unified model on a cluster of heterogeneous system connected via a network. We use system characteristics to balance the loads of the processors. The experiments of the model with load balancing shows a significant reduction of retrieval time while maintaining the quality of perceptual results.**

I. INTRODUCTION

Multimedia databases have become more important since conventional databases cannot provide the necessary efficiency and performance. Multimedia databases encounter three major difficulties. First, the content is subjective information; that is, intelligence is required to characterize the data. The data recognition requires prior knowledge and special techniques in Computer Vision and Pattern Recognition; the problems are NP-hard. Second, if a method or processing technique is designed and developed for one type of data or feature, it usually not appropriate for others. For instance, a technique designed for indexing audio data may not be usable for image data; or, a technique developed for a color feature may not be useful for a texture feature in image and video data. Third, the usual huge size of multimedia data and the requirement for a similarity search affect the computation. A similarity search is desirable for a multimedia database system. If a picture of a house is used as a query to an image database, we expect to retrieve pictures that contain similar houses in them. The comparison is not pixel by pixel between a query and the records in a database; but rather, closeness to the query. Similarity matching requires the computation of the distance between a query and each record in the database; the best match is chosen from the data set with the smallest distances. To solve these three problems, we use a mathematical model to represent the features; a k-tree model to represent the data structures of the multimedia data; and exploit parallelism to reduce the retrieval time [7][8].

In this paper, color and texture are the features of interest; they represent the subjective information of the multimedia data. We use a normalization technique to generate the indices. The domain of a feature is reduced to a set of selected values from a universe of potential values for the feature. We use an identification number for each element in the reduced set [7]. When data is inserted into the system, it is converted to the selected domain. The feature is represented by a histogram. For color feature, a few colors are picked from the whole infinite universe of colors. A finite number indexes each color. The color feature of an image or a video is represented by a histogram using the indexed color. For texture feature, we selected a set of textures and assigned an identification number to each texture. The feature of a texture is represented by the histogram of texture identification, which is the same method that was used for the color feature. The comparison of two features is based upon the distance between the histograms that define the features.

To reduce the response time, we proposed a parallel model of a homogeneous system to perform a content-based multimedia retrieval. The experimental results were very positive in both qualitative and quantitative metrics. However, in the real world, we do not have dedicated machines that always have the same configurations. The homogeneous model may be not used efficiently enough in the real-life heterogeneous environment. In this paper, we investigate a parallel model for multimedia database retrieval using a heterogeneous cluster system. We use system characteristics, such as processor speed, to divide tasks among the processors in the systems. Our computer system environment is composed of Sun Sparc and Pentium-Linux machines, which are connected via a 10Mbit local area network. We evaluate the model by comparing the retrieval results with the previous homogeneous parallel retrieval. The experimental results show the heterogeneous model produces a significant reduction of the retrieval time of an image from a 30,000-record image database.

This paper is organized as follow. Section II describes our k-tree parallel model. Section III has the details of our cluster system environment and its heterogeneity. The experiment and its results are described in Section IV. The last section concludes our works and proposes future directions.

II. THE K-TREE PARALLEL MODEL

A *k*-tree is a directed graph; each node has 2^k incoming edges and one outgoing edge with a balanced structure. [6] A *k*-tree is a *binary tree* for 1-dimensional data and a *quadtree* for 2-dimensional data. Exploiting a k-tree brings three main benefits. First, the k-tree holds the

information of spatio-temporal data on the tree structure itself. It reduces distance computation time to a comparison between two tree nodes. Second, a k-tree can accelerate multiresolution processing by calculating small, global information first and then large, local information when precise resolution is needed. Third, the data on a k-tree is unified since only the degree of the tree changes, while the processing algorithm and data structure remain invariant. Therefore, an algorithm for a particular type of feature can be reused for a feature of another media type.

Content-based retrieval of multidimensional signals is done by comparing features extracted from the input query with features extracted from every record in the database. The features of a multidimensional signal are subjective information. They are characteristics that are used to distinguish one signal from others. A 2-dimensional signal, such as an image, is characterized by features such as color, texture, and intensity. The basic algorithms for the searching of data in each of the different domains are quite similar. A matching search requires that the index key (defining feature) be unique and matched to the query. Exactly matched searching requires exhaustive comparisons that are inefficient and unsuitable for multidimensional signals; similarity searching is more appropriate. A similarity-search re-orders the database by distance between each record and the query; the result is selected from the ranking.

Multimedia data retrieval requires similarity searching; exact matching, which is used in conventional database, is not appropriate for this type of application. Similarity searching generally can be done in two steps; 1) finding distances between a query and all records in the database and then 2) sorting the distances and returning the results – the set of data items that have shortest distances. We also call this process "ranking." The details of regular weighted ranking are discussed in [7]. In Fig. 1, we show a parallel searching using multi-feature scheme. Prior to the search the database is distributed among processors. Each processor performs the comparison between the query and its portion of the database. The search results based on those features are sorted in parallel to create the final ranking. The pseudo code of parallel "ranking" is shown in Algorithm 1.

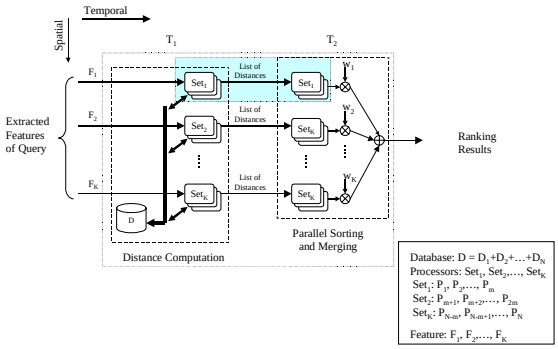


FIG. 1. PARALLEL MODEL MULTI-FEATURE SEARCHING.

ALGORITHM 1: DATAPARALLELSEARCH

DataParallelSearch(IN Query,
IN FeaturesInDatabase[N],
IN NumberOfProcessors,
OUT Record(Distances[N], DataNumber[N]))

1) **Begin**

2) Extract the interest *Feature* from the *Query.*

3) Distribute the data on to all processors in the system evenly.

4) *PortionSize = N / NumberOfProcessors.*

5) ***At all processors:*** **For** each *i* in 1..*PortionSize* of *FeatureInDatabase* **do**

6) Find the *Distances[i]* between *Feature* and *FeatureInDatabase[i]*

7) *DataNumber[N]=i*

8) **End For**

9) Parallel sort all the records of (*distance[N]*,*feature[N]*) in ascending order of *Distance[N]*

10) **End**

III. CLUSTER SYSTEM ENVIRONMENT

In this section, we give the details of our experimental platforms, evaluated multimedia database systems, and our proposed load-balancing algorithm for the k-tree model.

A. Computational platform

A cluster system environment plays an important role in a distributed multimedia database. A simple model of a distributed environment is a homogeneous system, where each workstation is identical and communication links between the workstations are also comparable. In such a system, data partitioning can be evenly subdivided. A static load-balancing scheme can be applied. A system environment is usually heterogeneous; the workstations are different in terms of computational power, storage space, hardware architecture, and so on. In this paper, we establish a uniform framework for content-based

multimedia data retrievalon a cluster system environment. In our experiment, one of the workstation serves as a host while others are assigned to be slave nodes. The system is composed of seven machines, which include two models of Sun workstations and Pentium/Linux PCs connected via local area network and use the Message Passing Interface (MPI) library as a message-passing interconnection mechanism [8].

B. Multimedia database system

In this paper, we use an image database which uses histogram-based features as indices. Two types of histogram-based feature (colors and textures) have been examined.

Before beginning the extraction of features, all images are normalized, scaled down to 128x128 pixels. The color feature extraction is performed in two steps. The first step transforms the number of colors of the scaled images to a pre-selected 166-color set [5]. The second step stores the transformed image in a quad tree structure. Texture feature extraction requires three steps. The first step transforms the 64 blocks of 16x16 pixels in to 64 sets of wavelet data using a Quadratic Mirror Filter (QMF) (2 iterations, 7 sub-bands). [7][8] Each wavelet data produces seven subbands of means and variances; i.e. a 14-element vector. In the second step, the texture vectors are then compared to 162-reference textures from VisTex [7] in the known texture table to generate 64 texture indices representing textures for blocked data. The third step constructs and stores the texture features in a quad tree structure.

The steps of the quad tree generation are the same for both features. The transformed color images and texture-identification (texture-id) matrices are mapped onto the leaves of a quad tree structure. The leaves represent a single pixel of the normalized image. Histograms of the leaves, which share the same parent nodes, are summed and the results are stored at their parent nodes. The process continues iteratively for each level until the root has been reached.

C. Data Partitioning and Load Balancing

To exploit heterogeneous parallelism, we use task and machine characteristics to decide which processors the tasks should be allocated to. The heterogeneity in the task level is the differences of the searching into the feature indices and the heterogeneity of the machines includes processor power and communication speed. We can generalize the load distribution in a mathematical model as follows

Let W be the processing requirements; N be the number of processors; and S_i be the speed of processor i, then the work share of processor i (L_i) is defined by:

$$Li = W_i S_i / \sum_{k=1}^{N} S_k$$

In the experiment, we do not allow task and data migrations. Since the data size of the index table is very large, any feature-index transfers causes a significant increase in the computation time. We have measured the communication during periods of low system utilization. The average communication time per record is 1.05 second, which is just slightly less than the computation of the slowest machine in the system. Since the physical connection of the system uses a bus topology, the communication is always performed sequentially. Since the data movement would be less than optimal, we decided to avoid all data migrations. The network was used for gathering the results from each node to the host before summarizing final results.

IV. EXPERIMENTS AND RESULTS

The cluster environment consists of seven Unix-based machines; two Pentium/Linux-based PCs, two Sun Sparc-20 workstations, and three Sun Ultrasparc I workstations. One of Sun Ultrasparc is designated as the host. The MPI library is used as the interconnection mechanism. We use comparative computational power to classify workstation types. Table 1 shows the computational power ratios; the ratios are based on the results similarity searching a database of 500 images using single- and multi- feature storage algorithms. The computational power ratio of a Sun Ultrasparc I to a Pentium II PC to a Sun Sparc-20 is 3:2:1.8.

TABLE I: COMPUTATIONAL POWER RATIO OF WORKSTATIONS

		Sun Ultra-Sparc I	Pentium II	Sun Sparc 20
I/O time (seconds)	Color	1.05	1.37	1.54
	Texture	0.51	0.73	0.81
	Color& Texture	1.35	1.85	1.55
Computational time (seconds)	Color	0.15	0.35	0.45
	Texture	0.09	0.21	0.24
	Color& Texture	0.20	0.50	1.0
Computational power ratio		3	2	1.8
Average communica-tion time	1.05			

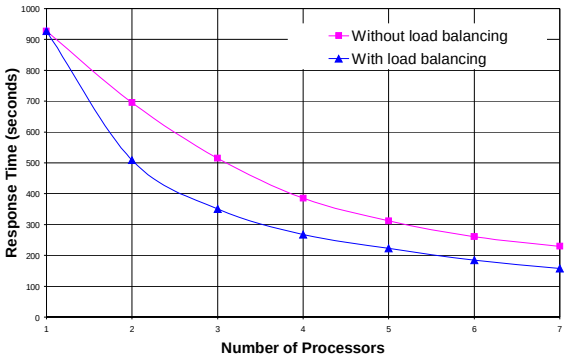
In this experiment, image retrieval is performed using two features; color and texture. The extracted features are derived from database images of 128x128 pixels; each is evenly divided into 64 blocks. The *quadtree* of the histograms for each image is made up of 3 levels; 64 leaf nodes. We perform two data partitioning schemes based on the database of 30,000 images. In the first scheme, the database of color and texture histograms is evenly divided among workstations, without considering the heterogeneity of cluster environment. In the second approach, data partitioning is based on the ratio of computational power of the workstations. Table 2 shows a data distribution profile as a function of the number of processors.

TABLE II: DATA DISTRIBUTION ACROSS WORKSTATIONS

# of proces -sors	# of image records distributed on each processor in thesystem						
	Ultra	P-II	Sparc	Ultra	P-II	Ultra	Sparc
1	30K	-	-	-	-	-	-
2	18K	12K	-	-	-	-	-
3	13.2K	8.8K	8K	-	-	-	-
4	9.2K	6.1K	5.5K	9.2K	-	-	-
5	7.6K	5K	4.5K	7.6K	5K	-	-
6	6.1K	4K	3.7K	6.1K	4K	6.1K	
7	5.4K	3.6K	3.3K	5.4K	3.6K	5.4K	3.3K

The results shown in Fig. 2(a) depicts the response time of the system as a function of the number of processors used to perform the ranking; the selection features are both color and texture. Fig. 2(b) shows as a function of the number of processors. As the number of processors used to perform the computation increases, the computation time decreases significantly. Moreover, the data partitioning based on heterogeneity information achieves a higher speedup than an even distribution approach. Fig. 3 depicts the top-twenty output images on the sorted list when both color and texture are used as selection features.

FIG. 2(A): RESPONSE TIMES



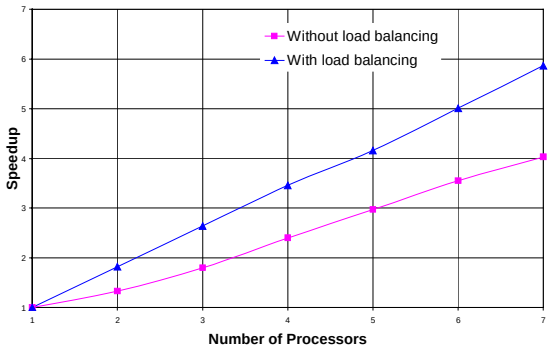


FIG. 2(B): SPEEDUPS

FIG. 3: PERCEPTION OUTPUT; THE SELECTION FEATURES ARECOLOR AND TEXTURE.

V. CONCLUSTIONS AND FUTURE DIRECTIONS

We introduced a parallel model for multimedia database content-based retrievals on a cluster of heterogeneous workstations. The model allows the extension of the system for the new types of data, new techniques, and new types of interest contents with less effort. The experimental results show that heterogeneous processing with load balancing can reduce retrieval time over a homogeneous approach. Data partitioning based on system heterogeneity achieves a better response time in comparison with uniform distribution of database over a cluster of workstations. Our future work will focus mainly on classification technique based on multi-resolution structure of the k-tree at different levels. Some load balancing and process migration techniques are also in our future work.

VI. REFERENCES

[1] P. Chalermwat, N. Alexandridis, P. Piamsa-nga, and M. O'Connell, Parallel image processing on heterogeneous computing network systems, *International Conference on Image Processing*, 1996.

[2] T. El-Ghazawi, P. Chalermwat, P. Piamsa-nga, A. Ozkaya, N. Speciale, and D. Wilson, PACET: PC-parallel architecture for cost-efficient telemetry processing, *IEEE Aerospace Conference*, 1998.

[3] V. Gudivada and V. Raghavan, "Special issue on content-based image retrieval systems," in *IEEE Computers*, Vol. 28, No. 9, September 1995.

[4] Z. Kemp, "Multimedia and spatial information systems," *IEEE Multimedia*, 2(4), 1995.

[5] J. R. Smith and S.-F. Chang, SaFe: "A General Framework for Integrated Spatial and Feature Image Search," *IEEE Workshop on Multimedia Signal Processing*, 1997.

[6] P. Piamsa-nga, N. Alexandridis, G. Blankenship, G. Papakonstantinou, P. Tsanakas, and S. Tzafestas, "A Unified Model for Multimedia Retrieval by Content," *International Conference on Computer and Their Application (CATA98)*, 1998.

[7] P. Piamsa-nga, N. Alexandridis, S. Srakaew, and G. Blankenship, "A parallel algorithm for multi-feature content-based multimedia retrieval," *Seventh International Conference on Intelligent Systems (ICIS98)*, Paris, France, July 1-3, 1998.

[8] S. Srakaew, N. A. Alexandridis, P. Piamsa-nga, and G. Blankenship, "A parallel model for multimedia retrieval based on multidimensional signal structure," in *International workshop on systems, signal and image processing (IWSSIP98)*, Zagreb, Croatia, June 3-5, 1998.