Chapter 35
PIR:
A Domain Specific Language for Multimedia Information Retrieval

Xiaobing Huang
University of Wisconsin, USA

Tian Zhao
University of Wisconsin, USA

Yu Cao
University of Massachusetts, USA

ABSTRACT

Multimedia Information Retrieval (MIR) is a problem domain that includes programming tasks such as salient feature extraction, machine learning, indexing, and retrieval. There are a variety of implementations and algorithms for these tasks in different languages and frameworks, which are difficult to compose and reuse due to the interface and language incompatibility. Due to this low reusability, researchers often have to implement their experiments from scratch and the resulting programs cannot be easily adapted to parallel and distributed executions, which is important for handling large data sets. In this paper, we present Pipeline Information Retrieval (PIR), a Domain Specific Language (DSL) for multi-modal feature manipulation. The goal of PIR is to unify the MIR programming tasks by hiding the programming details under a flexible layer of domain specific interface. PIR optimizes the MIR tasks by compiling the DSL programs into pipeline graphs, which can be executed using a variety of strategies (e.g. sequential, parallel, or distributed execution). The authors evaluated the performance of PIR applications on single machine with multiple cores, local cluster, and Amazon Elastic Compute Cloud (EC2) platform. The result shows that the PIR programs can greatly help MIR researchers and developers perform fast prototyping on single machine environment and achieve nice scalability on distributed platforms.

1. INTRODUCTION

Multimedia Information Retrieval (MIR) (Datta, Joshi, Li, & Wang, 2008; Lew, 2012; Lew, Sebe, Djeraba, & Jain, 2006; Yoshitaka & Ichikawa, 1999) refers to the research endeavor that centers on searching knowledge from multimedia data. In the last decades, substantial progress has been made in different area of MIR research, such as multimedia feature extraction (Hu, Xie, Li, Zeng, & Maybank, 2011; Tuytelaars & Mikolajczyk, 2008), learning and semantics (Atrey, Hossain, El Saddik, & Kankanhalli, 2010; Clinchant, Ah-Pine, & Csurka, 2011; Wang & Hua, 2011), and high performance indexing and query (Moise, Shestakov, Gudmundsson, & Amsaleg, 2013; Scherp & Mezaris, 2013; Shestakov, Moise, Gudmundsson, & Amsaleg, 2013; Mohamed & Marchand Maillet, 2012). As shown by recent surveys (Datta et al., 2008; Lew, 2012; Lew et al., 2006; Yoshitaka & Ichikawa, 1999), since the year 2000, the MIR research efforts have grown tremendously in terms of the number of researchers and practitioners involved, as well as the research papers published. As a result of substantial progress of MIR research and applications, many related software packages, libraries, and systems have been developed and evaluated using a wide range of multimedia data. Some prominent examples include the GIFT (the GNU Image-Finding Tool) (CVML, 2007), FIRE (the Flexible Image Retrieval Engine) (Deselaers et al., 2010), Caliph & Emir (Lux, 2009), LIRE (Lucene Image Retrieval) (Savvas & Chatzichristofis, 2008), ImageTerrier and OpenIMAJ (Hare, Samangooei, Dupplaw, & Lewis, 2011). While significant progress in both MIR research and software development have been made, in practice, we have witnessed that code reuse and system composition for MIR research are still very difficult and the new system developed on top of existing MIR implementation are not optimized for efficiency and cannot be easily adapted for parallelization, which is essential for handling large multimedia data sets. In addition, there is often a steep learning curve for researchers to understand and appropriately use existing frameworks and packages that serve a wide range of MIR purposes before they can even write a single line of code. In fact, sometimes the learning cost is so high that researchers have to give up and turn to create their own software packages instead; such practices accumulatively worsen the current status. Moreover, a lot of components of the MIR software libraries are sequential programs that are designed to run on shared memory computer architectures. MIR experiments of large data sets are time consuming and resource intensive; they often take hours to days to complete and some may even fail after exhausting main memory.

Recent development of distributed computing platforms such as Cloud-based services provides great opportunity to reduce runtime cost of large MIR experiments. A Cloud based service such as Amazon EC2 allows users to distribute workload to many worker nodes; thereby reducing the runtime costs for data parallel applications. However, it is nontrivial to develop MIR applications and deploy them on distributed platforms. An example of this would be to implement a MIR application using a MapReduce framework such as Hadoop and then deploy the application on a distributed computing platform. This presents several challenges to the developers. Firstly, they are often required to have a good understanding of the distributed framework interface they are working with. They need to not only grasp the appropriate usages of different distributed programming artifacts but also understand how to deploy the application on the distributed computing environments that mostly demand platform-specific configurations, settings, and performance tuning. Secondly, they frequently need to adapt the sequential MIR libraries to run in distributed settings where subtle errors such as race conditions can arise. For example, if a user runs some parallel data processing task using several worker nodes while the task program calls a MIR library component that writes to a static variable, a race condition will form if subsequent

Related Content

Experiences with Software Architecture Analysis of Usability
Eelke Folmerand Jan Bosch (2008). *International Journal of Information Technology and Web Engineering (pp. 1-29).*
www.irma-international.org/article/experiences-software-architecture-analysis-usability/2648

Sentiment Analysis on Movie Reviews Dataset Using Support Vector Machines and Ensemble Learning
Razia Sulthana A., Jaithunbi A. K., Haritha Harikrishnanand Vijayakumar Varadarajan (2022).
International Journal of Information Technology and Web Engineering (pp. 1-23).
www.irma-international.org/article/sentiment-analysis-on-movie-reviews-dataset-using-support-vector-machines-and-ensemble-learning/311428

A Framework for Preserving the Privacy of Online Users Against XSS Worms on Online Social Network
Pooja Chaudhary, B. B. Guptaand Shashank Gupta (2019). *International Journal of Information Technology and Web Engineering (pp. 85-111).*
www.irma-international.org/article/a-framework-for-preserving-the-privacy-of-online-users-against-xss-worms-on-online-social-network/217696

Anomaly Detection and Quality Evaluation of Web Applications
May Haydar, Ghazwa Malak, Houari Sahraoui, Alexandre Petrenkoand Sergiy Boroday (2008). *Handbook of Research on Web Information Systems Quality (pp. 86-103).*
www.irma-international.org/chapter/anomaly-detection-quality-evaluation-web/21967

Dealing with Precise and Imprecise Temporal Data in Crisp Ontology
Fatma Ghorbel, Fayçal Hamdiand Elisabeth Métais (2020). *International Journal of Information Technology and Web Engineering (pp. 30-49).*
www.irma-international.org/article/dealing-with-precise-and-imprecise-temporal-data-in-crisp-ontology/252854