#304 Using VDMS to Index and Search 100M Images

Email notification
Select to receive email on updates to reviews and comments.

PC conflicts
None

Other conflicts

Rejected

Submission (1.8MB) 27 May 2020 12:13:24pm PDT · 1f6de82e

Abstract

Data scientists spend most of their time dealing with data preparation, rather than doing what they know best: build machine learning models and algorithms to solve previously unsolvable problems. In this paper, we describe the Visual Data Management System (VDMS), and demonstrate how it can be used to simplify the data preparation process and consequently gain in efficiency simply because we are using a system designed for the job. To demonstrate this, we use one of the largest available public datasets (YFCC100M), with 99+ million images and videos, plus add-ons that include machine-generated tags and 4K dimensional feature vectors, for a total of ~13TB of data. VDMS differs from existing data management systems due to its focus on supporting machine learning and data analytics pipelines that rely on images, videos, and feature vectors, treating these as first class citizens. We demonstrate how VDMS outperforms well- known and widely used systems for data management by up to ~35x, with an average improvement of about 15x for our use-cases, and particularly at scale. At the same time, VDMS simplifies the process of data preparation and data access, and provides functionalities non-existent in alternative options.

Authors (blind)

L. Remis, C. Lacewell [details]

Appendices (Supplemental Material) (1.5MB)

Topics

	Nov	ExpMet	WriQua	OveMer	RevExp
Review #304A	1	3	2	1	3
Review #304B	1	2	2	1	3

You are an **author** of this submission.

Edit submission

Reviews in plain text

Review #304A

Paper summary

VDMS is a system that is built on top of existing open-source projects such as OpenCV and TileDB and able to search 100M images faster than MySQL.

In ML pipeline, many of the steps currently require understanding different software solutions that provide various functionalities that must be stitched together with a script. Many ad-hoc solutions also must combine databases and file systems to storage metadata and visual data.

VMDS simplifies the data preparation process and gain in efficiency as it creates one full system to do the job.

Strengths

Good motivation.

A system that is faster than MySQL, up to 35x and about 15x on average.

Significant Weaknesses

Almost no contribution to the systems community (VDMS looks like a simple layer built on top existing technologies such as OpenCV and TileDB, and mainly stores the metadata in persistent memory).

The paper is submitted to the wrong venue. Only 1 SOSP/OSDI reference. Submitting this to the database/visualization community.

The design is only 1.5 page, and evaluation already starts on page 4. The design is mainly about putting metadata as graphs in persistent memory and also building another layer on top of TileDB and OpenCV.

I suspect performance improvement really comes from TileDB and putting metadata in the persistent memory. There is no advanced discussion on index data structures, etc.

Comments for author

The weaknesses section above sums up my rating.

Thank you for submitting to OSDI, but I think you submit to a wrong venue. Maybe your work has a contribution, but definitely not to the hardcore systems community like OSDI.

How much of the performance improvement is due to TileDB vs. your layer?

If you build on top TileDB and OpenCV, please be considerate to describe what they are and their primary advantages. What's the diff between TileDB and MySQL?

Figure 1 and 2 seem similar, and they are too big. What does 4 drums in Figure 1 and 2 mean? Do you use a RAID of 4 disks?

How big is the YFCC100M dataset? (in bytes I mean).

Section 3.2.2---explain the speedup. Why do you improve by that much? What is it that makes your performance an order of magnitude better? What is the key? Is it because of TileDB, your in-memory metadata, or any other reasons.

28 references---need more in this era of open publications. Especially you tackle an important subject.

Novelty

1. Published before or openly commercialized

Experimental methodology

3. Average

Writing quality

2. Needs improvement

Overall merit

1. Reject (Serious problems, I'll argue against this paper)

Reviewer expertise

3. Knowledgeable

Review #304B

Paper summary

This paper presents VDMS, a visual (media) data management system to facilitate analytics and machine learning tasks using such data. It includes a persistent memory graph database that maintains the relationship between data objects and supports image/video storage plus a variety of operations. The authors evaluate VDMS' efficiency in database build time and storage space consumption, and assess its image search/retrieval performance.

Strengths

+ Efficient and scalable visual data management is important for many data analytics and machine learning projects today.

Significant Weaknesses

– Thin technical discussion and lack of research contributions

– No justification for chosen design choices, lack of description of objects

– No related work discussion, especially the relationship between this work and existing object DBMS systems

– Evaluation compares with MySQL, a system not designed for visual data management; no results on video data despite earlier discussion on video processing operations.

Comments for author

This paper addresses a real problem, as I believe that many data science or machine learning practitioners are indeed using inefficient, ad-hoc tools or scripts to set up their routine workflows analyzing visual data.

However, the paper reads more like a development report than a research paper. It is not clear what exactly are the research problems and what are the (new) approaches proposed by the authors. The authors did not specify any technical contributions and the technical part (between introduction and evaluation) is exactly 2 page long.

Even for development, the paper does not justify design/implementation choices. For example, why do you use persistent memory for maintaining object relations (metadata) here? For such visual databases, data are imported in batch and are used in read-only manners (as reflected in your own evaluation). What's the importance of data persistence that make persistent memory attractive? Also, how do you store/organize your object data, or are they simply files indexed by the graph database?

There is no related work discussion at all. You should especially discuss object databases (such as recent versions of Postgres).

The evaluation also lacks comparison with such alternative solutions. It compares with MySQL, which are not designed/optimized for media data. The MySQL database building process uses a relational schema defined by the authors and involves python processing. It's not clear how much such overhead the VDMS side needs. In addition, the graph database used in VDMS evaluation does not use persistent memory at all. Finally, why leave some experiments to the Appendix, when there is 2/3 page left in the main paper?

Novelty

1. Published before or openly commercialized

Experimental methodology

2. Poor

Writing quality

2. Needs improvement

Overall merit

1. Reject (Serious problems, I'll argue against this paper)

Reviewer expertise

3. Knowledgeable