Hadepot: A Repository of Big-Data Applications

Magdalena Balazinska
Dept. of Computer Science and Engineering
University of Washington

The development of new systems for big-data applications requires clear benchmarks for evaluating and comparing these systems. There are two possible ways for developing these benchmarks: creating synthetic benchmark applications, similar to the TPC series [4], or using real applications and real datasets. We advocate the latter approach and discuss some of the opportunities and challenges.

Opportunity 1: Following Evolving Requirements

The amount of data that our society is able to generate is growing at an unprecedented scale and rate. Researchers estimate the size of the Web to be over a trillion Webpages. The size of Twitter has already exceeded the Library of Congress. The next generation of telescopic sky surveys such as the Large Synoptic Survey Telescope (LSST) [3] will generate 10s to 100s of petabytes a year of imagery and derived data. The Earth Microbiome Project [1] expects to produce 2.4 petabases in their metagenomics effort.

Because the datasets are growing so quickly, the requirements and needs for big-data applications are also changing. For this reason, we argue that using real data and real applications for benchmarking big-data tools creates the opportunity to always stay in tune with the needs of real users and real applications.

Opportunity 2: Grand Challenges

Ideally, if we try to leverage real applications for benchmarking, we may be able to identify some exciting grand challenges for big-data systems that, if solved, would enable new discoveries in domain sciences or would impact society in some other interesting way. This could be a lot more interesting than focusing on synthetic grand challenges such as quickly sorting a very large amount of data.

There are several challenges, however, with using real applications for system benchmarking.

Challenge 1: Collecting Representative Applications

The first challenge when using real applications and real data is to ensure that a large number of diverse and representative applications serve as benchmarks. It would be easy to fall into the trap of over-specializing all benchmarks to a small number of applications. It is also important to keep adding new applications to avoid having a repository of old and no-longer relevant applications.

To address both challenges, we propose to create online repositories of big-data applications. In fact, at the University of Washington, we started the *Hadepot* repository [2] where we are collecting MapReduce applications for exactly this purpose. A few community-wide repositories would help focus the efforts to build high-quality collections of applications.

We find that there are, however, two challenges in growing such repositories:

- *Incentivizing Contributions*: We need a way to incentivize users of big-data tools to contribute their applications. This requires tools for anonymizing data when necessary and streamlining the application and data contribution processes. We argue, however, that we need even more incentives because packaging an application and preparing documentation that is sufficiently detailed to enable the reuse of the application will still require some effort. One option is to partner with a cloud provider and enable application contributors to make money whenever someone uses the cloud to run their application. The application authors could use this money as cloud credits to push their own research forward.

- *Maintaining Contributions*: A second important challenge of maintaining a repository of benchmark applications is their maintenance. Clearly, the data and the algorithm descriptions must be copied to some shared cloud service where they can be archived. Indeed, relying on authors to maintain their contributions would certainly lead to broken links and would impose too much of a burden on contributors. On the other hand, hosting all such datasets in a public cloud has a cost. In some cases, cloud providers are willing to wave that cost if applications generate sufficient traffic. However, some alternate mechanism for sustaining a community repository of benchmark applications could be helpful.

Challenge 2: Cataloging Applications

Different big-data applications have different requirements and different characteristics. We posit that for a repository of applications to be truly useful for system benchmarking, the applications must be described and annotated in a way that enables the quick identification of groups of applications that are best suited for a given benchmark goal.

For example, applications have different high-level requirements:

- *OLTP or OLAP*: Some big-data applications are transaction-oriented while others focus on analytics.
- *Simple but large data:* Some applications need to process data that is structured and clean but massive in size.
- *Complex data:* Other applications need to process large-scale unstructured data or data with complex structure (*e.g.*, graphs).
- *Changing data:* A third type of applications need to process data that is continuously changing, perhaps even streaming.

Furthermore, when they are executing, applications have different properties: some applications exhibit significant load imbalance when executed in parallel. Some applications are IO intensive while others are CPU or network bandwidth intensive. There are many other important properties that will affect the big-data tools being benchmarked.

We posit that it is important to catalog all such properties as they are discovered over time because clearly listing these properties will enable system developers to select test applications based on their properties. For example, if a researcher develops a new scheduling algorithm, perhaps that researcher will be interested in testing her algorithm with a variety of applications that exhibit odd resource utilization patterns. An important challenge will be to automate the discovery and cataloging of such properties.

Challenge 3: Anonymizing Data

Finally, it is clearly critical to develop techniques for anonymizing submitted data and a large body of work exists in this area. Anonymization is a difficult problem in general. In the case of tool benchmarking, however, a great benefit is that system developers are not actually doing science with the applications and their data. They simply want to test the performance of their systems. As a result, even high degrees of anonymization and data perturbation should be acceptable, as long as the key application properties as described above are preserved.

Summary

Overall, we are in the camp that advocates the use of real applications for benchmarking purposes and see the need for well-organized, curated, and easily accessible repositories of such applications.

At the University of Washington, we started an effort to build such a repository of benchmark applications. We call our repository Hadepot [2] because our initial focus is on MapReduce applications.

References

[1] The Earth Microbiome Project. `http://www.earthmicrobiome.org/`.

[2] Hadepot: Repository of MapReduce applications. `http://nuage.cs.washington.edu/repository.php`.

[3] Large Synoptic Survey Telescope. `http://www.lsst.org/`.

[4] Transaction processing performance council. `http://www.tpc.org/`.