Scaling Data Science does not mean Scaling Machines

Devin Petersohn
University of California
devin.petersohn@berkeley.edu

There is an increasing need to support data science at scale, but scale is not the only requirement for scaling data science. Many modern data science frameworks focus solely on how well their implementation scales to many machines, while ignoring the system's usability. Moreover, some systems focus on reducing compute cost for the user. Compute and cloud monetary cost is yet another easy-to-measure metric that can demonstrate an impact for the end user. We argue that these metrics alone do not accurately reflect the impact on a data scientist's productivity or cost. The narrow focus on scale and cloud compute cost has led to a landscape of tools that look and feel unfamiliar to the data scientists who are supposed to use them.

Scaling to more machines does not necessarily translate to improved productivity for the data scientist; being able to scale to 10x the data will likely not yield 10x the insights or yield insights that are 10x more valuable, so while scalability metrics do matter in showing the capabilities of a system, they do not correspond 1:1 to realized value to the data scientist. Similarly, decreasing costs in the cloud does not necessarily save money for an organization; compute time is significantly cheaper than human time, and strictly measuring compute cost does not include the cost of the human's time to setup the new system. In a typical data science workflow, the data scientist is often the bottleneck. Data scientists need time to interpret results, and they spend a significant portion of their time cleaning and trying to understand their data.

Data science would not exist without data scientists, and a data scientist's primary goal is to extract value from data. When designing and building new systems, it can be tempting to place new requirements on the user in the interest of decreasing an implementation's complexity or faster development time. For example, many implementations request partitioning hints from the user. From the perspective of the project, this may seem like a good idea: the creators of the project are likely distributed systems experts and therefore want these knobs to tune performance. However, data scientists are generally not distributed computing experts, rather they are experts in their scientific or business domain. Placing new distributed computing expertise requirements on data scientists offloads too much of the responsibility onto them: they are now responsible for both writing their queries and for how efficiently they run. Data scientists typically do not, and should not, worry about where or how their workload runs, nor should they have to rewrite their queries to run on different sizes of data. The cognitive overload of new requirements, APIs, and knowledge takes away from the data scientist's ability to extract value from data. All of this is done in the interest of machine time, so we are in effect trading human time for machine time.

When an organization evaluates a new tool, it can be difficult to measure the expected value that the new tool will provide. There are certainly challenges to benchmarking how long it takes a data scientist to learn a new system's API and how much time a new system saves the data scientist or how much more productive a data scientist is when using a new system. We argue that an unbiased benchmark that measures data scientist "productivity" would more accurately measure the impact of adopting a new system. One potential measure of productivity is evaluating how long the data scientist spends synchronously waiting on the system. In interactive and exploratory data analysis, the data scientist spends some portion of their time thinking or typing new commands, and in these cases the data scientist is "working" and the CPU is idle. When a data scientist submits a query and waits on the result to decide what to do next, the data scientist is idle while the CPU is "working". Providing faster results to the user will enable them to make decisions faster, potentially resulting in more queries in a work day, which can translate to more insights. There should also be some measure of how difficult it is to adopt a new system, or even if the new system can do everything needed by the data scientists who will use it.

As a first step toward scaling the capabilities of data scientists, we built Modin (github.com/modin-project/modin). Modin was designed to enable data scientists to use the same Jupyter notebook while working on any environment, from a laptop to a large cluster. We started Modin as a drop-in replacement for pandas (github.com/pandas-dev/pandas), the most popular dataframe library in the Python programming language. Since its creation 2 years ago, Modin has accumulated over 5,500 GitHub stars and over 500,000 downloads, which shows both a desire and need for systems which optimize for the data scientist's time. We hope our work can bridge the gaps that exist between industry, open-source, and academia.

11th Annual Conference on Innovative Data Systems Research (CIDR '21)
January 10-13, 2021, Chaminade, USA.