First Look

Bodo.ai: Faster, More Efficient Large-scale Python Data Analytics

Date: December 2021 **Author:** Kerry Dolan, Senior IT Validation Analyst

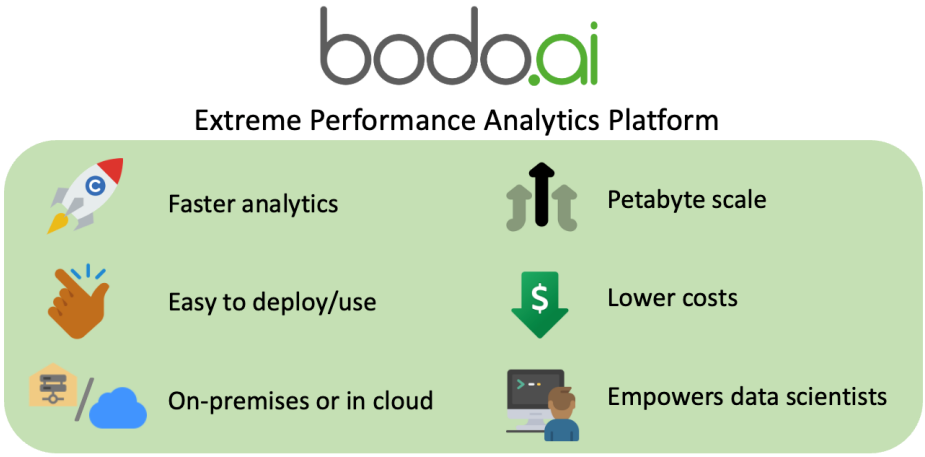
Data Analytics Challenges:

48% Of survey respondents cited ***"involving more people .. like developers and data science teams, to help broaden adoption by making access to analytics easier"*** on list of changes organizations expect to make to BI platforms over the next 12 months.[1]

33% Of survey respondents cited ***"improving data analytics for real-time business intelligence and customer insight"*** on list of business initiatives they expect to drive the most tech spending in their organizations in the next 12 months.[2]

To enable fast, efficient processing despite ever-growing data sets, engineers look for ways to speed data preparation, ETL, and featurization. While Python is a simple, interpreted language that is popular for analytics, scaling it is complex. Typically, a data scientist will prototype development in Python, but to use it in production at scale, high performance computing specialists must rewrite the code, test it, and validate it before deployment. This takes time, delays analysis, and increases costs of CPU and infrastructure.

Bodo.ai



Bodo simplifies and speeds high-performance data analytics tasks. Bodo is an extreme performance compute engine (available as software or SaaS) designed to parallelize Python analytics tasks such as data prep, ETL, and feature engineering. It can be used to optimize and parallelize standard Python libraries including Pandas, SciKit Learn and NumPy, and can be integrated with data sources such as cloud-based data lakes and data warehouses. Bodo searches Python code syntax and data sets for opportunities to parallelize, and creates MPI machine code for true parallel execution, scaling linearly across tens of thousands of CPUs. For large scale analytics, this can result in performance that is faster by orders of magnitude when compared with tools that use distributed computing with schedulers and wait times.

Benefits include:

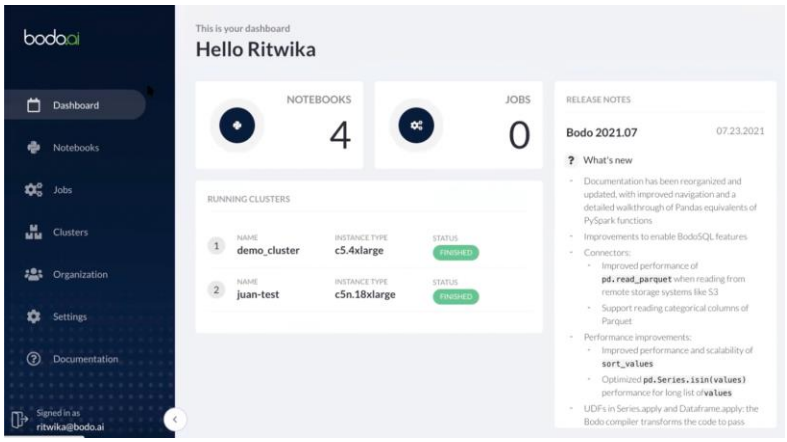
- *Fast, flexible, simple deployment.* The same easy-to-use Bodo compiler works on a multi-core laptop or on a cluster with thousands of cores.
- *Performance and scalability*. Bodo delivers automated parallel and sequential optimization of Python analytics code for petabyte scale workloads, increasing performance even on a single core. Bodo linearly scales performance with additional cores.
- *Reduced infrastructure costs*. Bodo gets the job done using less infrastructure, reducing CPU and infrastructure costs.
- *Empowered data teams.* Data scientists and data engineers can accomplish tasks using Python and SQL that were previously restricted to high performance computing experts using special languages and hardware.
- *Simple to learn*. Bodo uses a Python native API, so there is little new to learn.

[1] Source: ESG Research Report, *The Path to Data Leadership: Embracing Business Intelligence to Achieve Data-driven Success*, July 2021.
[2] Source: ESG Research Report, *2022 Technology Spending Intentions Survey*, November 2021.

ESG Demo Highlights

ESG viewed a remote demo of Bodo's managed cloud instance on a previously created AWS cluster using c5.4xlarge instances.

Ease of Use



- Bodo binaries are infrastrucure-agnostic, so Bodo works the same anywhere, with no additional code for scaling. This enables a data scientist to prototype code on a laptop, and then use that same code in a petabyte scale production cluster with no refactoring.
- Bodo code is human-readable, simplifying code writing and debugging.
- Bodo is easy to set up on-premises or in the cloud. A dashboard (hosted instance example at right) displays notebooks, jobs, and clusters, as well as settings, documentation, and release notes.
- The native Python APIs and intuitive GUI makes Bodo simple to use. We created a cluster with a few clicks by filling out a name and choosing instances, selecting the Bodo version, and setting the clusters to automatically shut down when the job was complete.

Performance Comparison

Bodo makes the greatest performance impact with tasks requiring significant computation such as user-defined functions and large joins. ESG viewed a data transform example using Bodo on one, 16, and 32 cores, and compared it to native Pandas code on one core without Bodo. First, we generated 50 million data points, set some values to NA, and wrote it to a Parquet file with a row group size of 100,000. Next, we wrote a data transform, which read the data into a dataframe, applied a custom Lambda function to the dataframe, and then wrote out to a Parquet file. Because this was a customized function within an "apply," Pandas could not use any optimized C libraries that are usually available to speed up compute in Pandas.

```
# Bodo on 1 core

import pandas as pd
import numpy as np
import bodo
import time

@bodo.jit
def f():
    df = pd.read_parquet("pd_example.pq")
    t0 = time.time()
    df["B"] = df.apply(lambda r: "NA" if pd.isna(r.A) else "P1" if r.A.
    df["C"] = df.A.dt.month
    print("Compute time:", time.time()-t0)
    df.to_parquet("bodo_output1.pq")
```

We ran the code with a single core using native Python and Pandas. Next, to use it with Bodo we simply added the "@bodo.jit" decorator to the code. To ensure an equivalent comparison of compute performance using clock time, we ran a compiled version of Bodo.

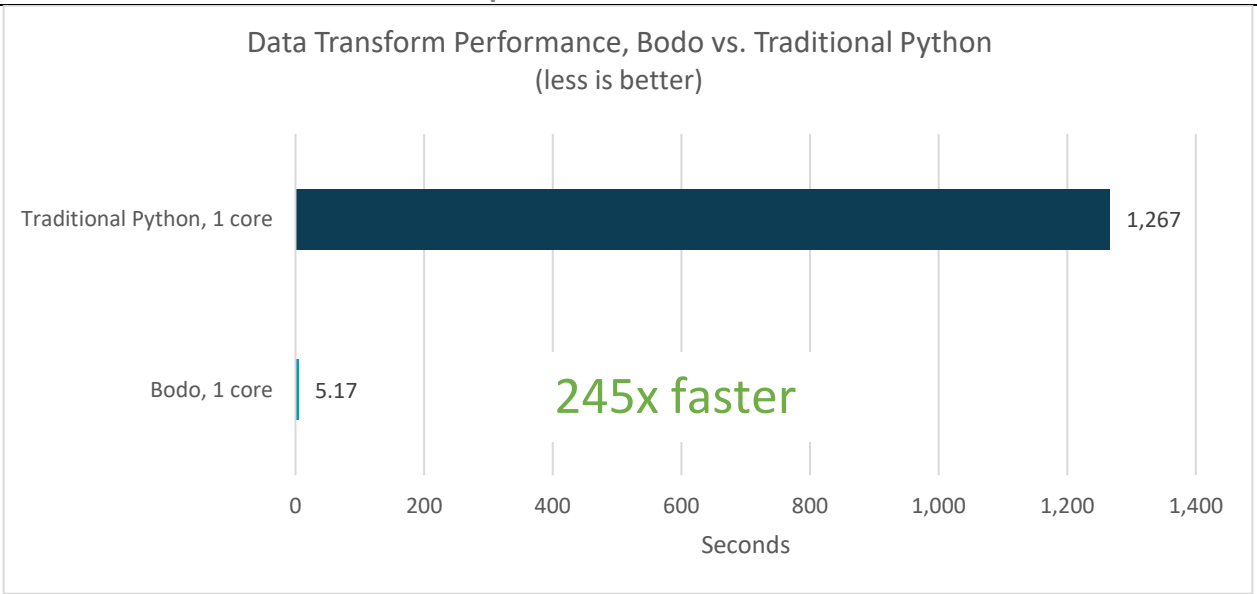
Bodo identified potential areas of parallelism in the function, automatically switched to a parallel version of the code, and executed operations on many data chunks at the same time.

As Figure 1 shows, for this demo ESG validated that:

- With traditional Python on a single core, the data transform task took 1266.93 seconds, or a little more than 21 minutes.

- The traditional Python code is slower because it must run the tasks individually for each of the 50 million entries.
- With Bodo on a single core, the task took 5.16 seconds.
  - Bodo provided sequential optimization, taking out the delays generated by interpreter overhead. It created an optimized binary and applied the Lambda to all 50 million entries in parallel.
- Using a single core, Bodo performed more than 245x faster than native python.

Figure 1. Data Transform Performance Comparison

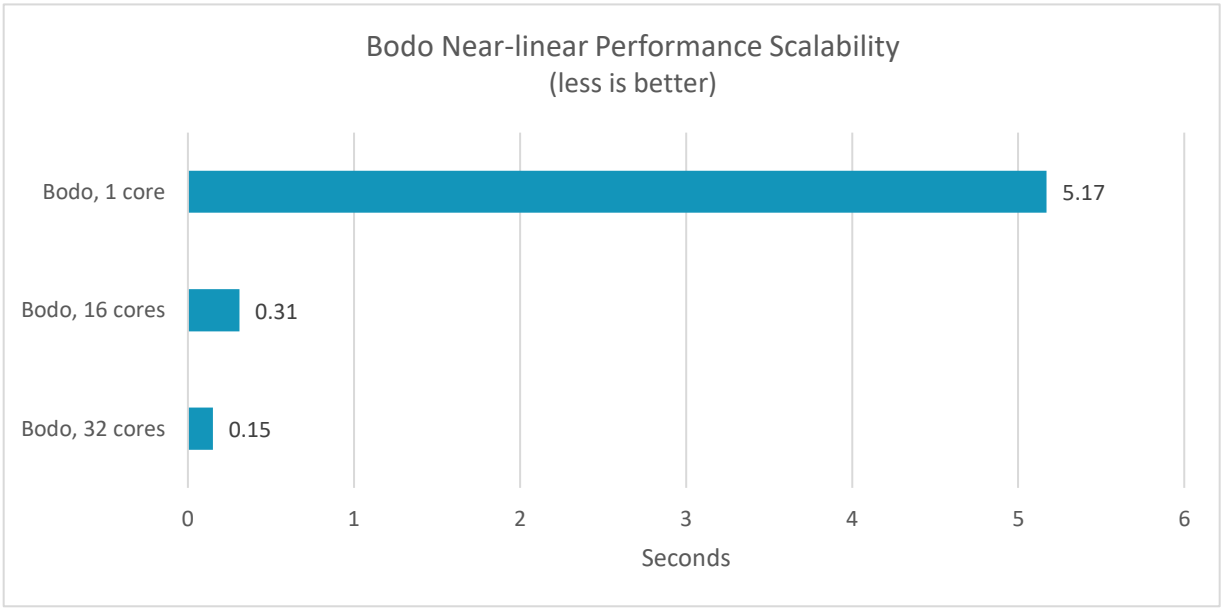


Source: Enterprise Strategy Group

As Figure 2 shows, ESG also validated Bodo's near-linear performance scalability when the number of cores increased.

- With 16 cores, Bodo's optimizations were multiplied by the additional compute capability. The task took 0.31 seconds.
- With 32 cores, the task took 0.15 seconds.

Figure 2. Bodo Near-linear Performance Scalability



Source: Enterprise Strategy Group

Another key metric is CPU efficiency and runtime. Bodo utilizes 100% of available compute resources, which other solutions do not. This speeds Bodo execution time and saves on infrastructure costs. When buying CPU compute time in the cloud, the faster tasks complete, the lower the cost. Bodo both speeds analytics processing and reduces computing costs.

Why This Matters

Growing data sets add to the complexity of data analytics. Organizations are often caught between two goals: the desire to use as much data as possible to gain the most knowledge, and the need to make analytics simpler, faster, and more efficient to speed time to insight. Accelerating data prep, ETL, and feature engineering can deliver faster answers that help generate better, faster business decisions.

Bodo.ai is a high-performance, inferential compiler for large-scale Python data analytics that generates true parallel execution to dramatically speed compute-heavy tasks such as large joins and user-defined functions. Because Bodo is a compiler that uses standard Python APIs, it is simple to learn and can empower developers, data scientists, and data engineers to execute tasks previously reserved for HPC specialists. This also eliminates the time-consuming code rewrites, testing, and validation required to scale analytics using other solutions. Bodo enables developers to put code straight into production.

ESG validated 245x faster performance for a data transform using Bodo, as well as near-linear performance scalability when increasing the number of cores. We also validated Bodo's maximum CPU efficiency, which can save time and money.

It should be noted that Bodo was not designed for all Python workloads. But for data analytics, ESG believes Bodo can deliver simple, automatic parallelization for faster, more-efficient analytics processing at extreme scale. Processing analytics faster by orders of magnitude could be a game changer for any organization.