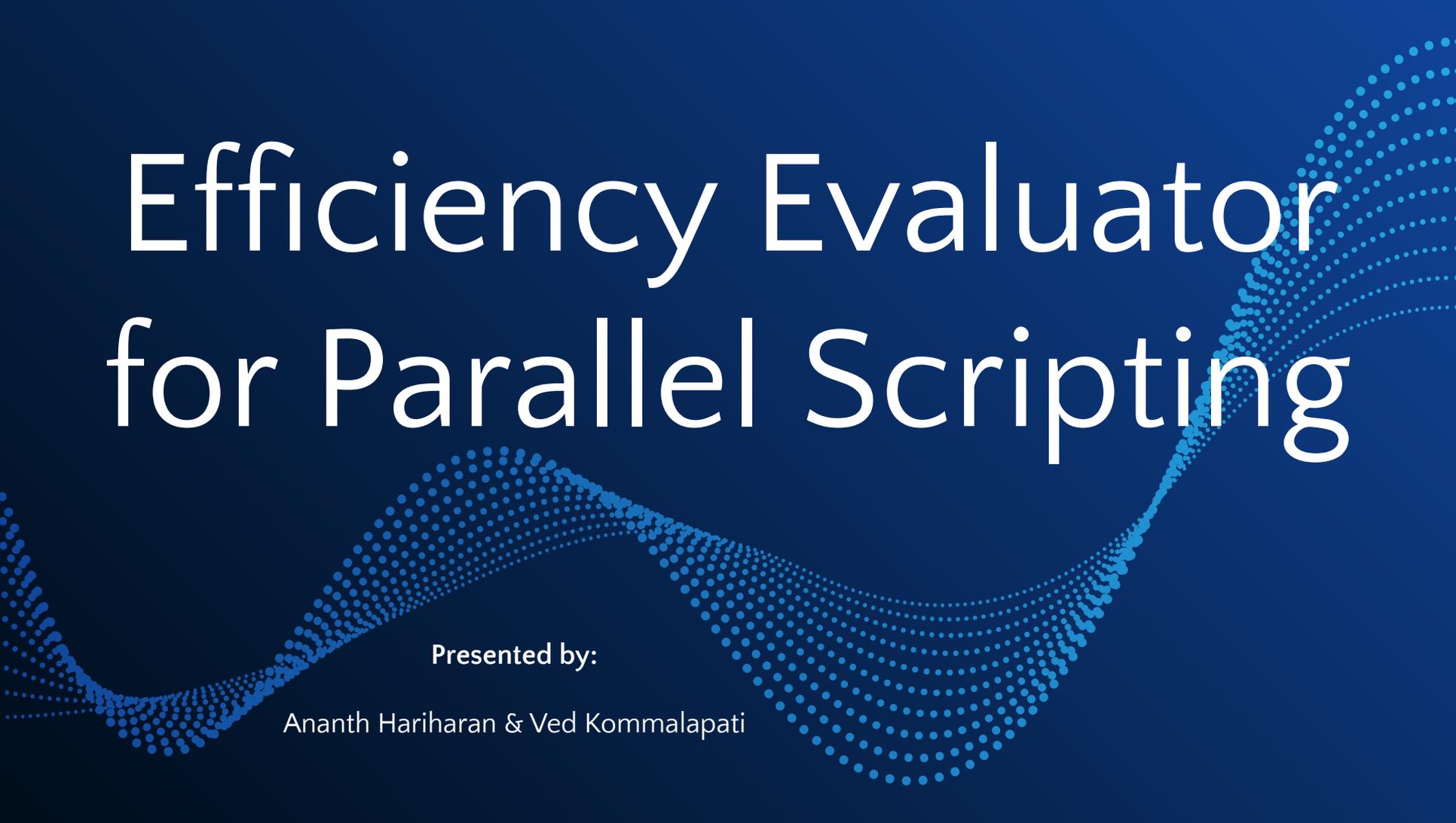


Efficiency Evaluator for Parallel Scripting

A decorative graphic consisting of multiple parallel, wavy lines of small blue dots, creating a sense of motion and depth. The dots are arranged in a way that suggests a wave or a series of overlapping paths, set against a dark blue background.

Presented by:

Ananth Hariharan & Ved Kommalapati

Who We Are

Our names are Ved Kommalapati and Ananth Hariharan, and we are 12th graders at Normal Community High School in Normal, Illinois

We had the privilege of being involved with a project using Parsl this summer under the guidance of Dr. Daniel Katz and Ben Clifford

Our Objectives

The purpose of our project



Our Research Question

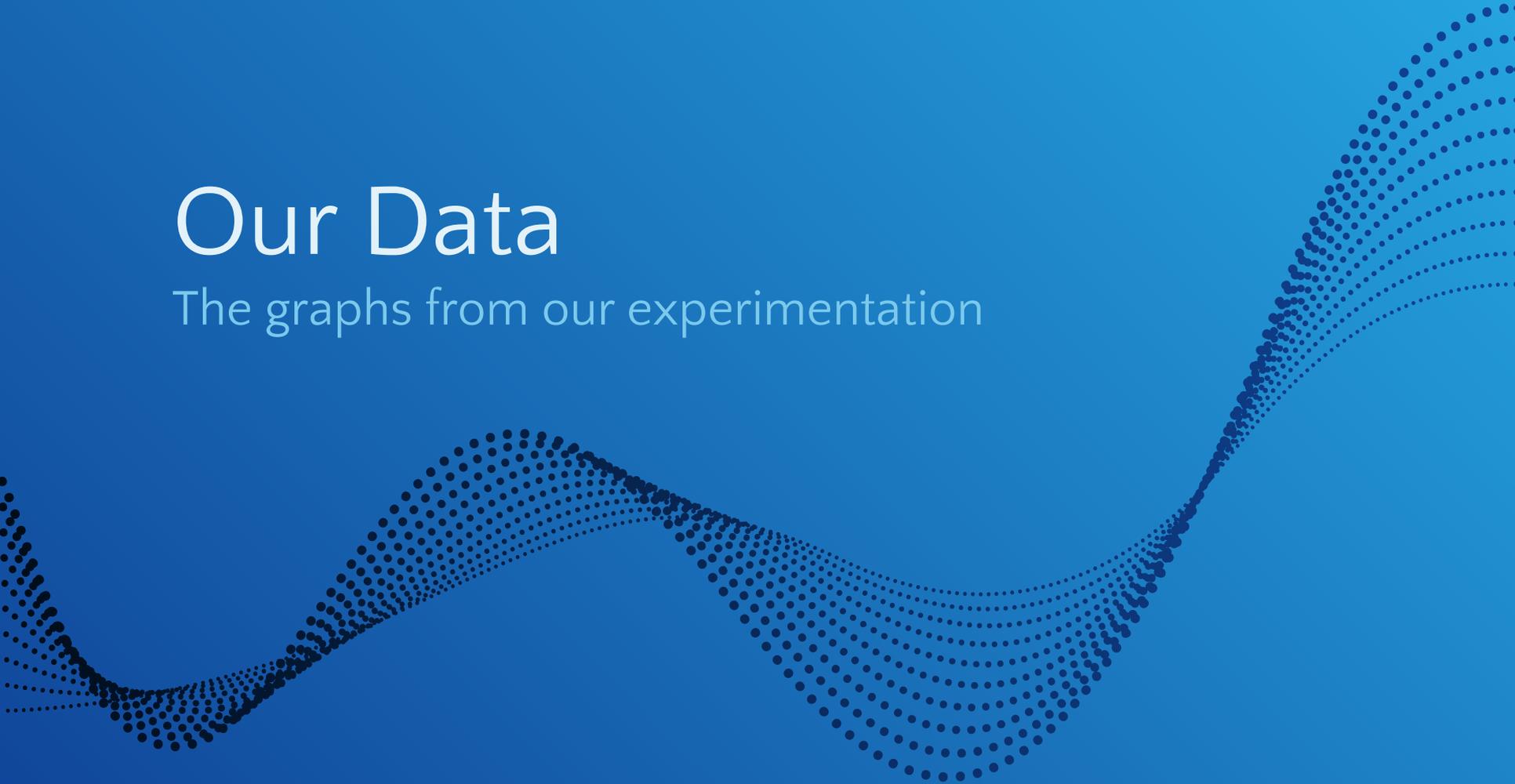
- How should workers and cost be balanced?
- What is the most optimal use of nodes for a given workflow?
- How can the use of time and resources be managed?

Our Solution

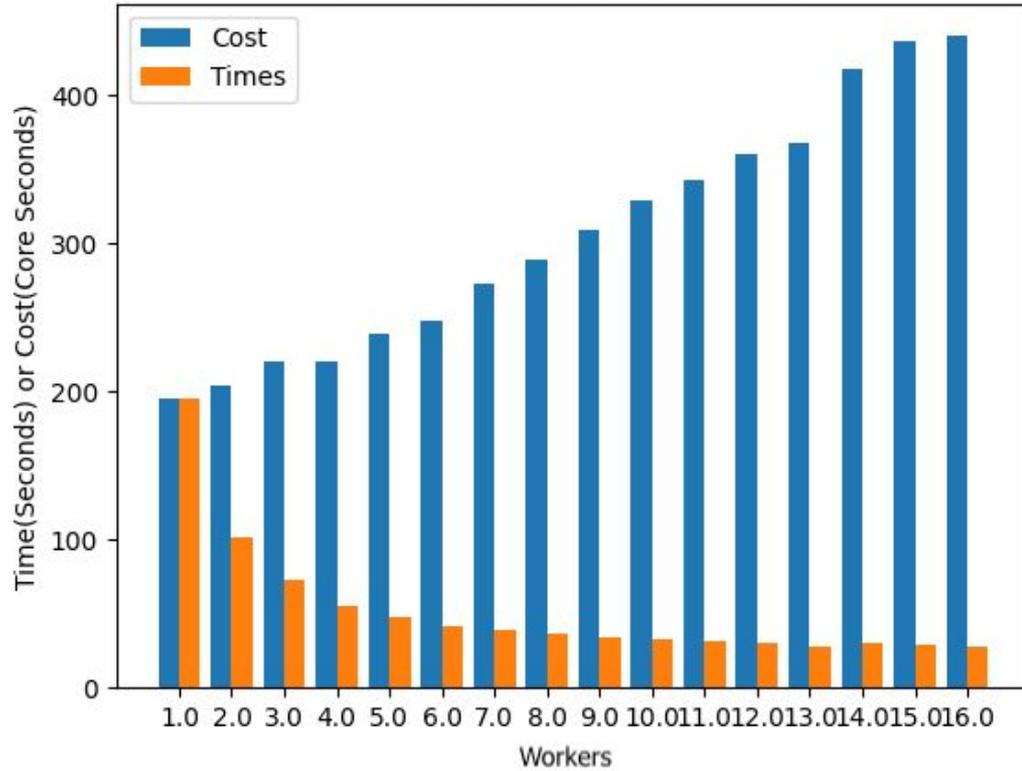
- Efficiency Evaluator for Parallel Scripting (EEPS)
- Determines cost in relation to time
- Outputs graphs with efficiency data
- Helps user find most optimal number of cores for their program

Our Data

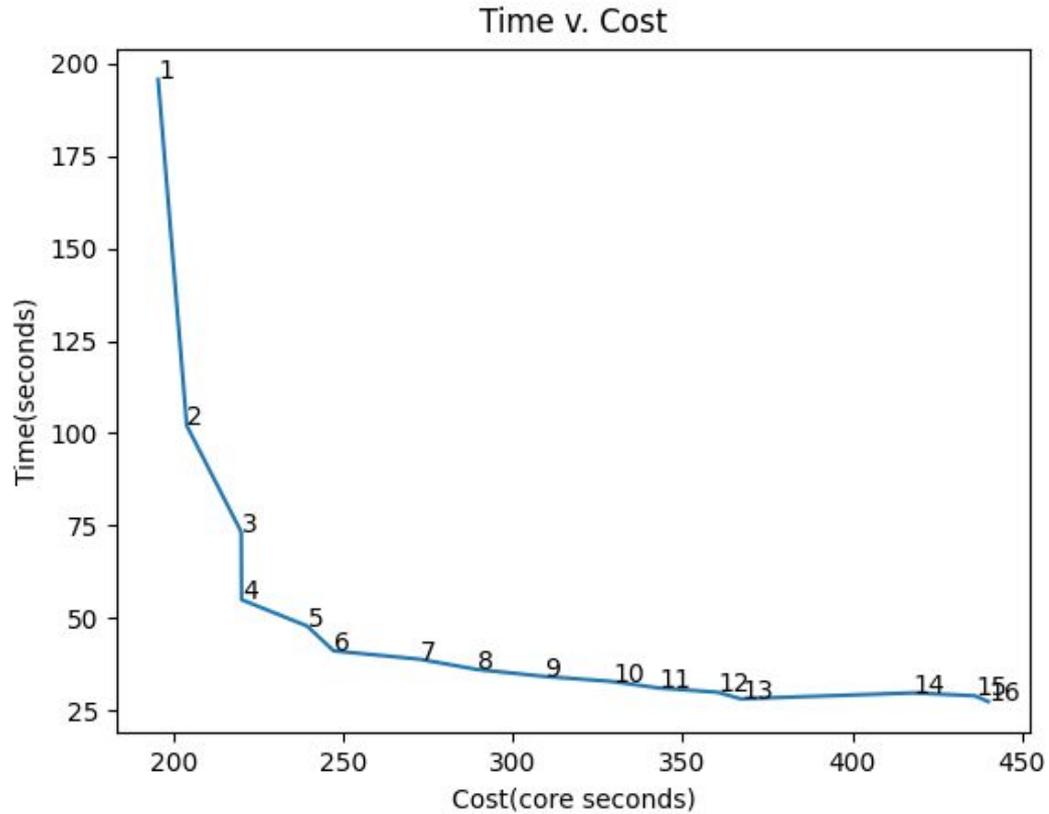
The graphs from our experimentation



Cost and Time v. Workers(Cores)



Illustrated effects of cores on cost and time



Trade-offs between shorter runtimes and cost

Our Guide

The process of using EEPS

Using EEPS



Input your code

Add your Parsl workflow to 'eeps.py' as directed in the documentation

View charted information

After choosing the number of workers you want to test, run `singleCpwTest.py`

Evaluate the data

EEPS will chart out data that gives you an idea of the runtimes and resource costs associated with your project

Future Improvements

- Modularization
- Database implementation
 - monitoring.db functionality
- More informational graphs
 - Enables user to input specific requirements
 - Output ideal workers/cores for workflow

Thank You!

[EEPS GitHub Repository](#)

[More Information](#)
[\(Parsl Blog Article\)](#)

Parsl & funcX Fest 2021