A Community Platform for Research on Pricing and Distributed Machine Learning

Xuanzhe Li†, Samuel Gomena†, Logan Ballard†, Juntao Li‡, Ehsan Aryafar†, and Carlee Joe-Wong‡
†Computer Science Department, Portland State University, Portland, OR, 97201
‡Electrical and Computer Engineering Department, Carnegie Mellon University, Moffett Field, CA 94035

***Abstract*—Data generated by increasingly pervasive and intelligent devices has led to an explosion in the use of machine learning (ML) and artificial intelligence, with ever more complex models trained to support applications in fields as diverse as healthcare, finance, and robotics. In order to train these models in a reasonable amount of time, the training is often distributed among multiple machines. However, paying for these machines (either by constructing a local cloud infrastructure or renting machines through an external provider such as Amazon AWS) is very costly. We propose to reduce these costs by creating a marketplace of computing resources designed to support distributed machine learning algorithms. Through our marketplace (coined "DeepMarket"), users can lend their spare computing resources (when not needed) or augment their resources with available DeepMarket machines to train their ML models. Such a marketplace directly provides several benefits for two groups of researchers: (i) ML researchers would be able to train their models with much reduced cost, and (ii) network economics researchers would be able to experiment with different compute pricing mechanisms. The focus of this Demo is to introduce the audience to DeepMarket and its user interface (named "PLUTO"). In particular, we will bring a few laptops with pre-installed PLUTO applications so that users can see how they can create an account on DeepMarket servers, lend their resource, borrow available resources, submit ML jobs, and retrieve the results. Our overall goal is to encourage the conference audience to install PLUTO on their own machines and create a user and developer community around DeepMarket.**

I. Introduction

Deep learning has recently attracted a lot of attention and has been successfully applied in many diverse areas such as natural language processing, gaming, computer vision, and security. Developing deep learning models typically requires training a huge amount of data over multiple machines. Purchasing machines outright, however, can require significant upfront investment cost that is not justified by the intermittent use that many researchers require. Renting resources from cloud providers like Amazon AWS or Microsoft Azure is also expensive since cloud providers need to construct and maintain cloud (*i.e.,* data center) infrastructure with significant operation and maintenance cost. For example, training a reasonable language model can take multiple weeks on 10 or more GPUs, which costs at least $1000 over three weeks even with discounted spot prices [1]. Several recent works have attempted to intelligently exploit various types of cloud pricing to reduce their cost [2], but they are still fundamentally dependent on cloud providers' offering low-cost options.

One possible solution to reduce these costs is to introduce a *marketplace* for computing resources in which users can lend each other resources when they are idle, similar to Uber's or Airbnb's sharing platforms. Such a marketplace reduces upfront investment costs by allowing users to purchase fewer resources outright and supplement them with others' machines, and eliminates the need to pay a cloud provider. While prior work on volunteer computing has proposed similar ideas [3], such a computing marketplace is particularly appropriate for training deep learning models, which can easily be done distributedly. Algorithms to train these models can be easily adapted to run on intermittently available heterogeneous resources [4], e.g., both CPU and GPU servers, allowing a wide range of users to participate in the computing marketplace.

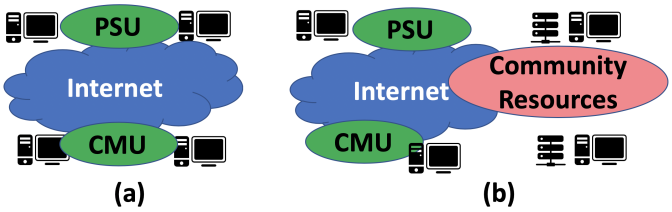

Fig. 1. (a) DeepMarket existing infrastructure is composed of GPU and CPU resources that are dynamically lent and borrowed by users across the campuses of PSU and CMU, (b) Our goal is to create a community of users around DeepMarket that contributes to all aspects of the project such as its development, maintenance, and management.

Our goal in this demo is to introduce DeepMarket, an open-source software framework that creates such a computing marketplace for training machine learning (ML) models. Users that lend resources to DeepMarket receive credits for doing so, which can be used to borrow resources from others in the future. In this way, researchers are incentivized to contribute their idle resources to DeepMarket; and DeepMarket automatically matches available resources to pending deep learning jobs, seamlessly executing the jobs over dispersed, heterogeneous machines. The demo will introduce the audience to three main components of DeepMarket: the user interface PLUTO; the service module, which tracks job and resource status; and the Executor module, which executes distributed training jobs. The demo will particularly showcase PLUTO's ability to lend computing resources in DeepMarket, borrow resources from others, submit ML jobs, and specify rental prices (or choose the default prices, which are optimized by DeepMarket).

DeepMarket is currently composed of GPU and CPU computing resources across the campuses of two universities:

Portland State University (PSU) and Carnegie Mellon (CMU). The platform is currently being used by machine learning students and researchers at these institutions to train models, and we plan to conduct economics experiments to determine the market-clearing price of compute resources under various conditions. In addition to introducing DeepMarket, a key goal of this demo is to encourage members of other universities to contribute to the project in order to create a community of users that use the platform for pricing and distributed ML and systems research. Users can contribute to DeepMarket by lending their idle resources, contributing to the software development, submitting ML jobs, or participating in the overall project development and management [5].

II. Related Work

There are existing infrastructures that provide low-cost access to massive computing resources such as Emulab [6], GENI [7], and OneLab [8]. However, these cloud testbeds do not provide easy to use abstractions and libraries for machine learning researchers or dynamic pricing capabilities to study network economics problems. Other computing infrastructure such as Akraino [9] and Steel [10] provide cloud services that are optimized for edge resources but do not provide access to a high number of computing resources or pricing capabilities.

In our prior work, we presented a preliminary demo ([11]) and performance results ([12]) of DeepMarket. Since then, we have updated the platform with several new functionalities including the following key capabilities: (i) the entire software stack is **sandboxed in Docker containers** [13]. For example, when a user lends a resource to DeepMarket, the application creates a Docker container which includes the necessary software images. Containerization provides an additional layer of security by isolating the DeepMarket job from the rest of the user machine and closing the container entirely once the job is completed. Docker containers can also operate on multiple operating systems, which increases the diversity of machines that can be used by DeepMarket. (ii) We have released a **new version of PLUTO** with several new functionalities. Users can now view data on the prices paid for all types of computing resources across all the users over the past week. This can assist users in selecting prices when lending or borrowing resources. We have also added a new tab that provides answers to frequently asked questions and a comment section such that a user can provide feedback on the platform.

III. DeepMarket Architecture

The DeepMarket architecture consists of three main components (modules) as shown in Figure 2. The first two modules (the executor module and the services module) reside on our servers at Portland State University (PSU). The third module (PLUTO) is an application that would be installed on the user machine. Our entire software stack (including the user module and backend services) is sandboxed in Docker containers for added security and cross-platform functionality.

Executor. This Module is responsible for data file management, scheduling of resources, and execution of distributed ML

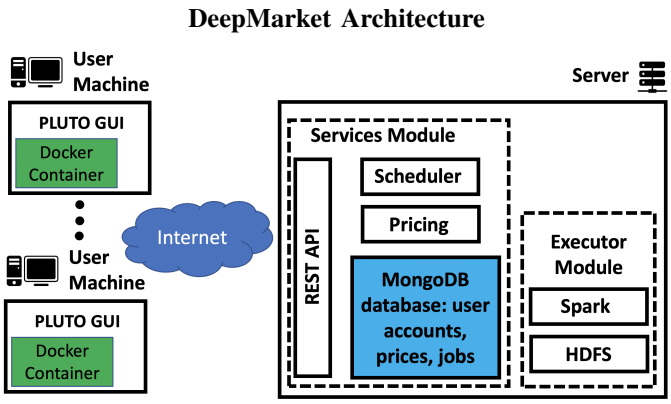

Fig. 2. DeepMarket architecture. The services and executor modules reside on servers at PSU and are responsible for job scheduling and execution, account creation and management, and storage of all data. Users use the PLUTO applications to submit jobs to DeepMarket, lend their resource, or borrow resources from DeepMarket. The PLUTO GUI resides on the user machines.

programs. We use a combination of Apache Spark, TensorFlowOnSpark, and HDFS (Hadoop Distributed File System) to achieve these goals. HDFS distributes training data for the job(s) assigned to each worker machine in a fault-tolerant manner. Apache Spark along with TensorflowOnSpark enable distributed deep learning on a cluster of GPU and CPUs, and support all types of TensorFlow programs, including both asynchronous and synchronous training and inferencing.

Services. In this Module, we have developed a secure REST API Service that saves the job submission and available resources information submitted via PLUTO. We have also developed an optimized pricing algorithm to generate *suggested* prices for each time slot based on the previous jobs' execution time, current available resources and frequency of resources utilization; these prices would be designed to ensure that the "value" of users' contributed resources matches the resources they borrow from others. The price values along with historical price data would be shown in user interface.

PLUTO. This module is a simple and intuitive graphical user interface developed using PyQt5. It allows users to lend their spare computational resources and list them on DeepMarket, view their submitted jobs' statuses, and submit ML jobs (i.e., borrow computational resources from other DeepMarket users and run their jobs on them). Before submitting the job for execution, the user uploads the training data and source files to the Hadoop data file system (HDFS) and specifies any configuration requirements (e.g., minimum RAM or a minimum number of workers) for the workers that will run this job. PLUTO resides on users' machines.

IV. Demo Overview

This demo will allow the audience to run experiments on DeepMarket, and observe its pricing and distributed machine learning capabilities. We will bring four laptops with pre-installed PLUTO applications on them to represent user machines. Each user will be able to create a DeepMarket account on a laptop, add all or part of the laptop resources to the

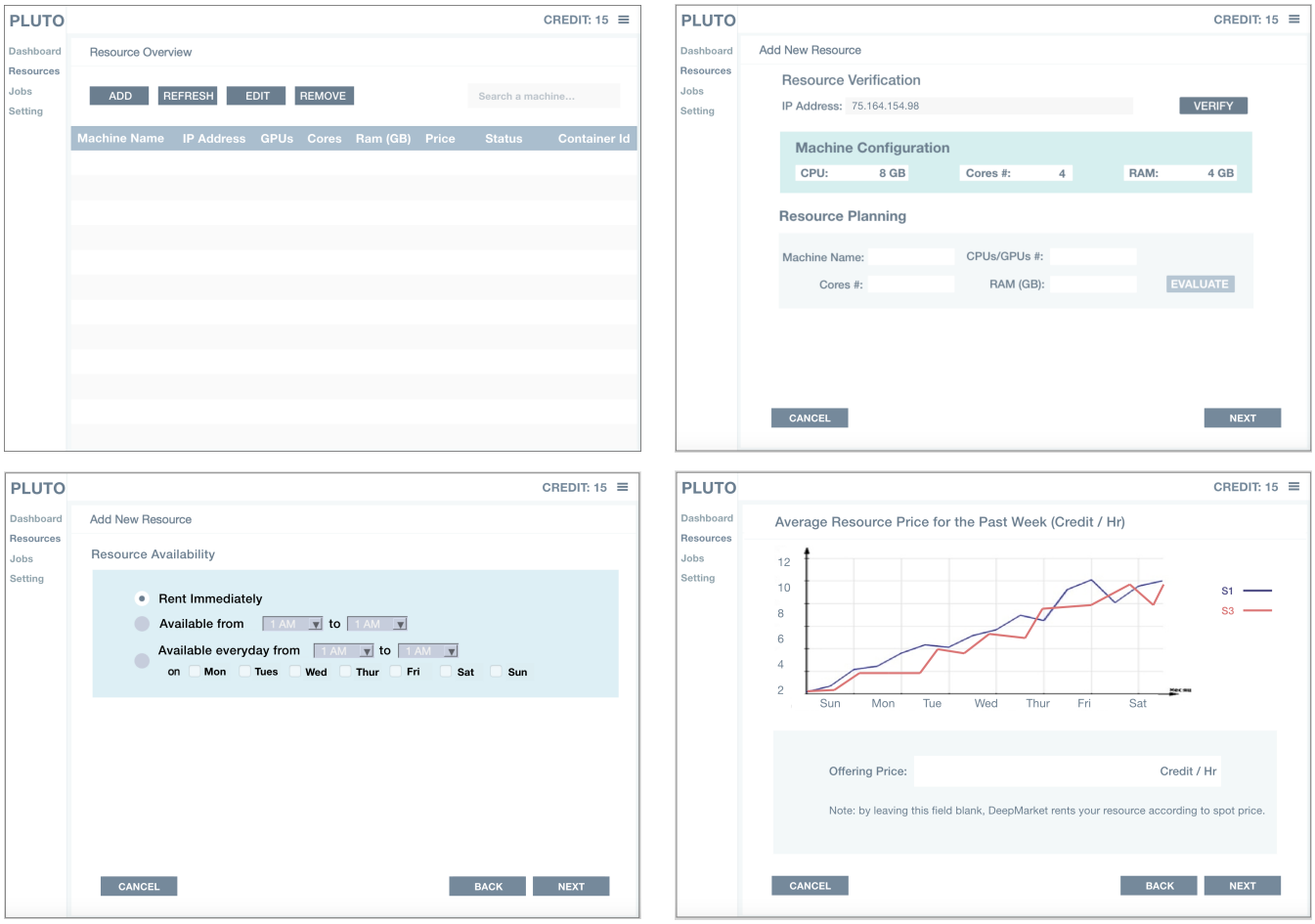

Fig. 3. Flow of adding a computing resource to DeepMarket: (a) the "Resource" tab allows a user add/remove a resource. The user can also edit the properties of a previously lent resource (e.g., its availability time), (b) the user specifies the portion of computing resource that will be lent to DeepMarket (e.g., 1 out of 2 available cores or 2 out of 4 GB of RAM), (c) the user specifies the timing availability of resource (e.g., every Monday and Wednesday from 10 PM to 5 AM), and (d) historical pricing data for two sample configurations (S1 and S3) is shown to the user to assist the user in choosing an appropriate price.

DeepMarket marketplace, or submit a ML job and retrieve its results via PLUTO. In addition, the demo will provide us with an opportunity to obtain early feedback from the audience (e.g., obtain inputs on desired features). We will also encourage the audience to start adopting DeepMarket in their own research, contribute computing resources, or take an active role in project development and management.

We next discuss some of the key features of PLUTO that will allow the audience members to experiment with it:

Resources Tab. Through PLUTO's resources tab (Figure 3), users can lend their current machines and add them to the pool of resources at DeepMarket. Machine IP address and configuration (e.g., its RAM, number of cores) will be automatically loaded on the application. PLUTO allows the user to lend all or part of the machine's available computing resources (e.g., half of the RAM and cores). As a result, users can continue to use the machine for personal use while lending a portion to DeepMarket. Next, the user specifies the availability of the resource. In our current implementation, the resource can be lent either immediately, at a specific day, or in a recurring manner (e.g., every Monday and Wednesday from 10 PM to 5 AM); the Spark backend automatically accounts for these constraints in scheduling jobs to machines. Finally, prior to resource addition, the user specifies the lending price. In order to assist the user in selecting an appropriate lending price, we show the average lending price for the past seven days for two configurations (S1 and S3). For example, S1 and S3 might represent Intel x3/x5/x7 configurations (or Amazon compute configurations). DeepMarket will only run a job on the user lent machine if it can pay the user at least the user specified price. User can also leave the price selection to DeepMarket (to automatically match lenders and borrowers with optimized pricing) by leaving the "Offering Price" blank.

Jobs Tab. PLUTO's "Jobs" tab (Figure 4) allows a user to borrow computing resources from DeepMarket and run distributed machine learning jobs. On the first screen, the user can add a new job or observe a summary of jobs that he/she has previously submitted. Here, the user can click on "Check Detail" to observe the detail of currently running and previously submitted jobs. In order to submit a new job, the user needs to specify two things: (i) *Job Detail:* here the user specifies the job name as well as the computing

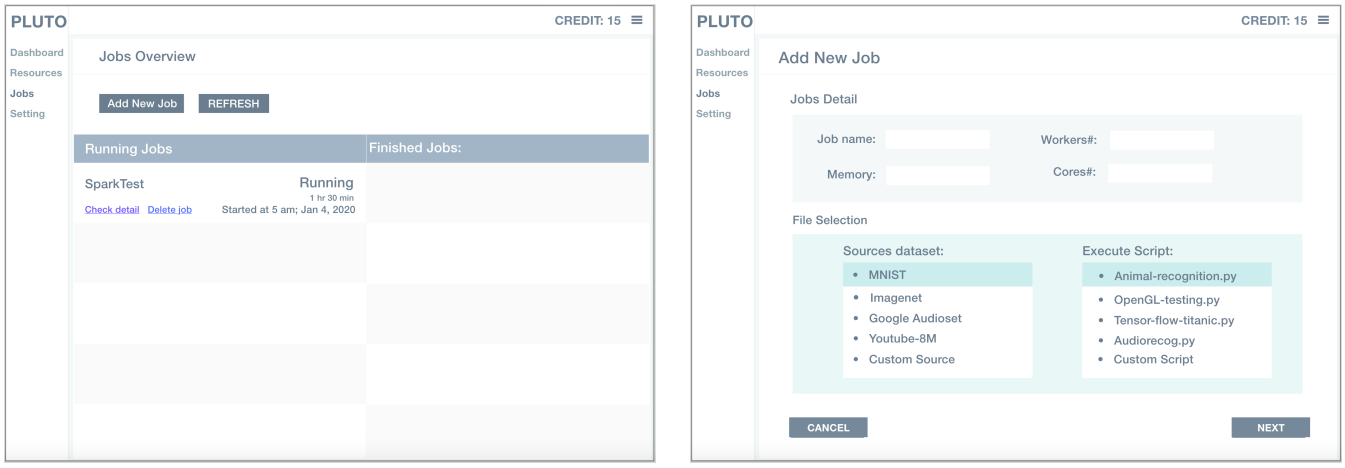

Fig. 4. Flow of job submission to DeepMarket: (a) the initial screen allows the user to add a new job, (b) user specifies the number of required resources (e.g., number of worker machines and their cores) prior to job submission. Further, the user specifies the training data and the executable script to be run on the data. In addition to custom source files and training data, our design allows for selection of standard datasets (e.g., MNIST) or sample executable scripts.

resources that are requested to run the job (e.g., the number of requested worker machines, their number of cores, and RAM). (ii) *Job data and scripts:* These are provided as HDFS paths to the training data files and the source file of the ML training algorithm. In our current implementation the user can select standardized files that we have already been uploaded to DeepMarket. These include datasets such as MNIST, Youtube-8M, and Kaggle-Titanic along with sample Python TensorFlow source codes that run on these datasets to train standardized ML models. Similar to the "Resources" tab, prior to job submission the user needs to specify the price (credits/hr) he/she is willing to pay. PLUTO recommends prices to user similar to the "Resources" tab (not shown here due to page limitations). A user job is only run if DeepMarket can find resources with similar or less cost.

V. Future Work

We are currently working on adding the following capabilities to DeepMarket:

Networking Among Containers. Our current implementation of DeepMarket has a robust performance when all containers (and computing resources) belong to the same local network. While we can currently submit jobs to DeepMarket from anywhere with an Internet connection, the jobs run smoothly only when all resources are in the same local network. Increasing robustness of communication and networking among containers that belong to different local networks is an important feature that we are actively working to resolve.

Custom File Transfer. DeepMarket does not yet fully support custom source (e.g., training dataset) and script (e.g., execution file) transfer through PLUTO. Robust implementation of this functionality is still under development.

Server Robustness and Redundancy. Our current implementation uses only a single server at PSU, which can be a single point of failure when the machine is down, or access to PSU network is denied (e.g., in a DoS attack). Creating server redundancy so that server functionality can smoothly transition to a different machine (e.g., to a CMU server) is another functionality that we are working to add to DeepMarket.

VI. Acknowledgements

This research was supported by NSF grants CNS-1910517, CNS-1942305, CNS-1751075 and CNS-1909306, and ARO W911NF1910036. The authors would like to thank the reviewers and ICDCS demo chairs for their useful comments.

References

[1] Amazon EC2, "Amazon ec2 spot instances," https://aws.amazon.com/ec2/spot/, 2018.
[2] L. Zheng, C. Joe-Wong, C. W. Tan, M. Chiang, and X. Wang, "How to bid the cloud," in *ACM SIGCOMM Computer Communication Review*, 2015, vol. 45, pp. 71–84.
[3] E. Lavoie and L. Hendren, "Personal volunteer computing," 2018, https://arxiv.org/pdf/1804.01482.pdf.
[4] Xiaoxi Zhang, Jianyu Wang, Gauri Joshi, and Carlee Joe-Wong, "Machine learning on volatile instances," in *Proceedings of IEEE INFOCOM*, 2020.
[5] "DeepMarket Project Website," https://deepmarket.cs.pdx.edu/
[6] M. Hibler, Ro. Ricci, L. Stoller, J. Duerig, S. Guruprasad, T. Stack, and K. Webb J. Lepreau, "Large-scale virtualization in the emulab network testbed," in *USENIX Annual Technical Conference, Boston, MA*, 2008.
[7] M. Berman, J. S. Chase, L. Landweber, A. Nakao, M. Ott, D. Raychaudhuri, R. Ricci, and I. Seskar, "GENI: A federated testbed for innovative network experiments," in *Computer Networks, 61, pp.5-23*, 2014.
[8] S. Fdida, F. Timur, and M. Sophia, "Onelab: Developing future internet testbeds," in *European Conference on a Service-Based Internet. Springer, Berlin, Heidelberg*, 2010.
[9] "Akraino Edge Stack," The Linux Foundation, https://www.lfedge.org/projects/akraino.
[10] S. A. Noghabi, J. Kolb, P. Bodik, and E. Cuervo, "Steel: Simplified development and deployment of edge-cloud applications," in *10th USENIX Workshop on Hot Topics in Cloud Computing*, 2018.
[11] S. Yerabolu, S. Gomena, E. Aryafar, and C. Joe-Wong, "An edge computing marketplace for distributed machine learning," in *Proceedings of ACM SIGCOMM (SIGCOMM Posters and Demos)*, 2019.
[12] S. Yerabolu, S. Kim, S. Gomena, X. Li, R. Patel, S. Bhise, and E. Aryafar, "DeepMarket: An edge computing marketplace with distributed tensorflow execution capability," in *IEEE ECOFEC (ECOnomics of Fog, Edge, and Cloud Computing) Workshop*, 2019.
[13] "Docker Container System," https://www.docker.com/.