MLSYSOPT ASSIGNMENT 1- [A1] [A2][A3] (Literature Exploration and Formulation) **Group: 43**

ML SYS OPTI: High-Performance Machine Learning: A Study of Parallel and Distributed Computation

Name	BITS ID	Contribution
SUBHRANSU MISHRA	2023AC05489	100%
AGHAV SAYALI SAKHARM	2023AC05435	100%
LAKSHMISRINIVAS PERAKAM	2023AC05540	100%
SHAILESH KUMAR SINGH	2023AC05475	100%
SATISH KUMAR DUMPETI	2023AC05885	100%

[A1] Literature Survey: IDEA Proposals

As part of this project, we have started exploring various ML sub domains and concepts that has been idealized using algorithms. Below are three possible approaches, covering supervised, unsupervised, and reinforcement learning.

1. **Parallelizing Ensemble Learning (Supervised ML)**
   - Focus on Random Forest or Gradient Boosting, where individual models can be trained independently in parallel.
   - Implement a parallel approach to reduce training time while maintaining accuracy.
2. **Parallelizing Deep Reinforcement Learning (DRL)**
   - Implement a parallelized DRL framework where multiple agents train simultaneously.
   - Improve training speed and policy learning through distributed execution.

Based on the feedback received after the first discussion, our team will implement the idea 2 first and if we are not able to secure necessary infrastructure, we will fall back to idea 1.

Research Papers / Journals Explored:

Paper1	***Parallelizing Ensemble Learning - (Supervised ML)***	*Access Link*
Paper2	***Parallelizing Deep Reinforcement Learning - (DRL)***	*Access Link*

[A2/A3] Problem Formulation & Initial Design Phase:

Academic Objective

The goal of this study is to explore the parallelization and/or distribution of machine learning algorithms to improve computational efficiency while maintaining or enhancing accuracy. The focus will be on optimizing training speed, reducing response time, and managing communication costs. This will be evaluated across three different learning paradigms—supervised learning, unsupervised learning, and reinforcement learning.

1. Parallelizing Ensemble Learning (Supervised ML)

Problem Statement:
Ensemble methods like Random Forest and Gradient Boosting are widely used in supervised learning due to their robustness and high accuracy. However, training these models can be computationally expensive, especially when working with large datasets. The primary challenge is to optimize training by parallelizing the execution of individual trees or boosting iterations across multiple processors.

Parallelization Approach:

- Each decision tree in the Random Forest can be built independently in parallel.
- Boosting algorithms, which are sequential by nature, can be parallelized by distributing data across multiple cores or using gradient-based approximations.
- Reduce inter-process communication overhead by maintaining local copies of dataset partitions and aggregating results periodically.

Performance Expectations:

- **Speedup:** Reduction in training time by distributing tree construction across multiple processors.
- **Communication Cost:** Low, as tree construction can be handled independently, with final model aggregation requiring minimal synchronization.
- **Response Time:** Faster model training allows for quicker deployment in real-world applications.

Potential Tech Stack & Methods:

- **Languages & Libraries:** Python, NumPy, scikit-learn, Dask, Spark MLlib, XGBoost
- **Parallelization Techniques:** Multi-threading (Python's joblib), multiprocessing, distributed processing with Apache Spark **(Optional If Configurable in Local PC)**
- **Computing Platforms:** Multi-core CPUs, GPU clusters, distributed computing on cloud platforms (AWS, GCP, Azure)

2. Parallelizing Deep Reinforcement Learning (Reinforcement Learning)

Problem Statement:
Deep Reinforcement Learning (DRL) is computationally expensive due to the need for extensive environment interactions and policy updates. Training a DRL agent in a sequential manner can be slow and inefficient, making real-time decision-making difficult. The challenge is to accelerate learning by parallelizing agent-environment interactions and experience replay mechanisms.

Parallelization Approach:

- Use multiple agents exploring the environment simultaneously, sharing experiences to improve policy updates.
- Parallelize the neural network updates by distributing batches across GPUs.
- Implement asynchronous experience replay to reduce latency.

Performance Expectations:

- **Speedup:** Faster policy updates by collecting experiences in parallel.
- **Communication Cost:** High, as agents must synchronize learned policies periodically.
- **Response Time:** Reduced training time, making real-time decision-making feasible.

Potential Tech Stack & Methods

- **Languages & Libraries:** Python, PyTorch, TensorFlow, OpenAI Gym, RLlib (Ray)
- **Parallelization Techniques:** Actor-Critic architectures (A3C, PPO), distributed rollout workers, GPU acceleration (CUDA, TensorFlow-Distributed)
- **Computing Platforms:** GPU clusters, cloud-based reinforcement learning (AWS SageMaker RL, Google TPU) → Very Complex with TPU and GPU to get

Conclusion:

After exploring different parallelization techniques across supervised, unsupervised, and reinforcement learning, it is evident that the first approach—**parallelizing ensemble learning (Random Forest or Gradient Boosting)**—is the most practical choice for this project as of now.
This method can be efficiently implemented on a **local multi-core PC**, making it accessible and manageable without the need for complex hardware configurations. In contrast, **K-Means clustering and Deep Reinforcement Learning require GPUs or TPUs**, which involve additional setup challenges and infrastructure requirements.

As it was suggested that K Means could be a general problem, hence in this project we are not going to consider the same. So the implementation will focus on DRL and if we fail to obtain TPU or GPU support, we will revert back to Random Forest based Algo boosting.

References:

Parallelizing Ensemble Learning	Ghimire, Ashutosh, and Fathi Amsaad. "A Parallel Approach to Enhance the Performance of Supervised Machine Learning Realized in a Multicore Environment." *Machine Learning & Knowledge Extraction* 6, no. 3 (2024).
Parallelizing Deep Reinforcement Learning	Kwok, Jacky, Marten Lohstroh, and Edward A. Lee. "Efficient parallel reinforcement learning framework using the reactor model." In *Proceedings of the 36th ACM Symposium on Parallelism in Algorithms and Architectures*, pp. 41-51. 2024.