Towards Autonomous Infrastructure Learning: Behavioral Pattern-Driven Optimization for Sustainable and Socially Responsible ML Infrastructure

Andrew Espira

Saint Peter's University

Research Assistant Proposal – Data Science Institute

Application Period: Fall 2025

1. Introduction

Modern machine learning (ML) training clusters frequently experience resource underutilization due to inflexible reservation policies, excessive graphics processing unit (GPU) allocation for long-running jobs, and inefficient job scheduling. These inefficiencies increase operational costs, reduce overall system performance, and limit access for research groups with fewer resources.

While prior systems such as Gandiva, Tiresias, Optimus, and Pollux have introduced innovations in scheduling and fairness, they operate reactively without leveraging behavioral patterns for predictive optimization (Xiao et al., 2018; Gu et al., 2019; Peng et al., 2018; Qiao et al., 2021).

This proposal explores autonomous infrastructure learning using behavioral pattern recognition to optimize ML clusters. It applies event-sequence analysis to identify and predict resource inefficiencies, supporting sustainability and social responsibility.

2. Problem Statement

Large-scale ML training environments suffer from three interrelated issues:

- Resource Hoarding – Jobs often reserve more ML cluster resources, such as GPUs, than required. This leads to unused GPU time and reduces overall cluster resource efficiency.
- Queue Spiraling – When job scheduling fails to prioritize workloads effectively, jobs can block each other, resulting in longer wait times for everyone. This reduces the number of jobs that finish and can make the system seem unfair.
- Accessibility Barriers – When ML cluster resources are not allocated efficiently, the overall computation cost increases. This creates barriers for smaller organizations or those with limited budgets to access the cluster.

Grounded in these challenges, the key research questions are:

1. How can behavioral pattern recognition identify resource waste patterns in ML cluster environments?
2. What event-sequence analysis methods (including but not limited to CRE-inspired approaches) can predict resource inefficiencies?
3. Can proof-of-concept autonomous recommendations improve resource allocation without manual intervention?
4. How might such improvements translate to sustainability and social responsibility outcomes?

3. Proposed Approach

The framework will investigate three integrated components:

Behavioral Pattern Detection Module

- Explore event-sequence analysis methods for identifying resource waste patterns
- Investigate several techniques, including methods inspired by Common Reliability Enumerations (CRE), approaches for detecting anomalies in time series data, and the use of statistical tools to identify common patterns.
- Evaluate Graph Neural Networks for topology-aware cluster analysis (Hamilton et al., 2017)

- Analyze historical traces from production clusters to validate the feasibility of pattern detection. Access to production traces will be safeguarded by appropriate data privacy and compliance measures, ensuring that all ethical considerations are addressed. This will involve anonymizing data as necessary and adhering to university guidelines on data handling and protection.

Proof-of-Concept Optimization Engine

- Develop basic autonomous recommendation system based on detected patterns
- Implement simple pattern-to-action mapping for scheduling suggestions
- Test feasibility of real-time pattern recognition using eBPF instrumentation
- Validate concept with controlled experiments on resource allocation improvements

Impact Assessment Framework

- Quantify potential ML cluster resource efficiency improvements using pattern-based optimization techniques.
- Analyze cost reduction implications for resource-constrained institutions
- Develop methodology for measuring sustainability and social impact outcomes
- Create reproducible evaluation framework for community validation

4. Related Work

Existing ML Schedulers:

- Gandiva (OSDI'18): GPU sharing and context-switching for efficiency improvement (Xiao et al., 2018).
- Tiresias (NSDI'19): Fair scheduling policies for distributed deep learning (Gu et al., 2019).
- Optimus (EuroSys'18): Resource prediction and dynamic scaling for ML workloads (Peng et al., 2018).
- Pollux (OSDI'21): Co-adaptive scheduling optimizing goodput with cost reductions (Qiao et al., 2021).

Research Gap: Current ML infrastructure schedulers address optimization challenges through specialized, component-level approaches. Optimus focuses on individual job performance prediction through convergence modeling (Peng et al., 2018), Pollux optimizes per-job resource allocation using adaptive goodput analysis (Qiao et al., 2021), and Tiresias ensures fair scheduling through execution-time estimation (Gu et al., 2019). However, these systems operate in isolation, each targeting specific optimization objectives without systematic integration of behavioral patterns across multiple infrastructure layers.

While these approaches demonstrate the value of predictive techniques for specific use cases, they lack a unified framework that comprehensively analyzes behavioral patterns spanning user interaction patterns, application resource consumption behaviors, and system-wide resource allocation dynamics. The absence of such holistic behavioral analysis creates optimization blind spots where individual component improvements may inadvertently create system-wide inefficiencies. For instance, resource hoarding can lead to a phenomenon known as queue spiraling, where excess reservations by individual jobs accumulate, causing significant delays across the system and unfairly prioritizing jobs of larger organizations over smaller ones. Furthermore, job-level optimizations that focus narrowly on individual tasks might lead to cluster fragmentation, where granular changes disrupt the overall cluster harmony, resulting in poor resource utilization. This research investigates whether event-sequence analysis methods can bridge the gap by providing systematic recognition of behavioral patterns across multiple infrastructure layers, enabling predictive optimization that prevents system-wide resource inefficiencies before they manifest as performance degradation.

Related Methodologies:
- Netflix Atlas demonstrates scalable time-series data collection for operational intelligence, handling billions of metrics for near real-time pattern analysis (Netflix Tech Blog, 2017).
- Event-sequence analysis approaches, such as those used in Common Reliability Enumerations (CREs), have shown effectiveness in systematically detecting infrastructure problems through temporal correlation patterns (Prequel, 2024).
- Recent work in GPU scheduling has identified optimization opportunities through systematic resource management approaches, such as addressing fragmentation in GPU sharing workloads (Weng et al., 2023).

5. Expected Contributions

Technical Contributions:

- Feasibility study of behavioral pattern recognition applied to ML infrastructure optimization
- Comparative analysis of event-sequence detection methods for resource waste identification
- Proof-of-concept framework demonstrating autonomous optimization recommendations
- Open-source prototype enabling community validation and extension

Academic Impact:

- Workshop paper documenting methodology and feasibility findings
- Reproducible research framework for future development
- Novel research direction at intersection of pattern recognition and infrastructure optimization

Sustainability and Social Impact:

- Quantified resource efficiency improvements through behavioral optimization
- Cost reduction analysis demonstrating improved accessibility for underserved institutions
- Framework enabling broader adoption of optimization techniques

6. Evaluation Plan

Datasets:

- Public traces (Alibaba PAI, Google Cluster traces, MLCommons benchmarks)
- Systematic data collection from university GPU clusters
- Controlled experimental environments for validation

Metrics:

- Cluster utilization efficiency, job completion time, queue wait time
- Resource waste reduction, cost per training job
- Accessibility improvements for diverse institution types

Baselines:

- Comparison against Gandiva, Tiresias, Optimus scheduling policies
- Evaluation against static allocation and reactive scheduling approaches

Methodology:

- Trace-driven simulations followed by controlled testbed validation
- Statistical significance testing for performance improvements

7. Implementation Timeline (11 weeks, 165 hours)

Weeks 1–3 (45 hours): Foundation and Pattern Analysis

- Literature synthesis on event-sequence analysis methods
- Dataset preparation from existing cluster traces
- Initial feasibility study of pattern detection approaches

Weeks 4–6 (45 hours): Behavioral Pattern Detection Development

- Implementation of multiple event-sequence analysis methods

- Comparative evaluation of pattern recognition approaches
- Development of basic pattern classification algorithms

Weeks 7–9 (45 hours): Proof-of-Concept Optimization

- Simple pattern-to-recommendation mapping prototype
- Validation experiments on controlled workloads
- Performance impact assessment of recommendations

Weeks 10–11 (30 hours): Documentation and Analysis

- Academic presentation preparation
- Feasibility findings documentation
- Future research direction identification

To define the standards for success, the next section establishes clear criteria and a validation framework for both minimum and extended outcomes.

Technical Feasibility Demonstration:

- Successfully implement detection on at least 3 ML workload types
- Achieve >70% accuracy in identifying resource hoarding patterns from historical traces
- Demonstrate proof-of-concept recommendations with measurable impact (>15% utilization improvement)

Academic Contribution:

- Workshop paper acceptance or submission-ready manuscript for systems conference
- Open-source prototype with documented methodology enabling community replication
- Validated experimental framework for pattern-based optimization research

Impact Assessment:

- Quantified resource efficiency improvements with statistical significance ($p < 0.05$)
- Cost reduction analysis showing potential democratization benefits for resource-constrained institutions
- Sustainability impact methodology with concrete efficiency metrics

Extended Success (Foundation for broader work)

Research Foundation for Future Development:

- Scalable architecture supporting multiple optimization algorithms
- Community adoption metrics (GitHub stars, forks, citations)
- Industry validation through pilot deployments or collaborations
- Grant funding secured for extended research

Failure Mitigation and Alternative Outcomes

If Pattern Detection Accuracy is Low (<70%):

- Document negative results as valuable contribution (what doesn’t work and why)
- Focus on methodology development and experimental framework validation
- Pivot to comparative analysis of different pattern detection approaches

If Autonomous Optimization Shows Limited Impact (<15% improvement):

- Emphasize feasibility study contribution and framework development

- Document limitations and requirements for production deployment
- Focus on sustainability and social impact analysis rather than performance claims

9. Resource Requirements

Computing Infrastructure:

- Google Cloud ($300 free credits), AWS SageMaker Studio Lab (free tier)
- NVIDIA Academic Grant Program application for extended resources
- MLCommons benchmark datasets for reproducible validation

Software Development:

- eBPF development framework (BCC, libbpf) for kernel instrumentation
- Python ML ecosystem (scikit-learn, PyTorch) for behavioral analysis
- OpenTelemetry integration for monitoring extension
- Kubernetes cluster for controlled experimental environments

Expected Budget:

- Cloud computing: $200–$500 for extended experiments beyond free tiers
- Development tools: $0 (open source ecosystem)
- Total investment: <$500 for comprehensive 11-week research validation

10. Conclusion

This proposal presents a systematic approach to ML cluster optimization through behavioral pattern recognition and autonomous resource management. Through event-sequence analysis, the framework aims to address resource hoarding and queue spiraling while supporting sustainability and social responsibility. The anticipated outcomes are improved cluster efficiency, reduced costs, and expanded accessibility for a wide range of research communities.

The 11-week research period will provide a strong foundation for this emerging area. Success criteria and fallback strategies will ensure academic value regardless of technical outcomes. This project presents a systematic exploration of autonomous infrastructure learning in ML environments, seeking to advance sustainable and equitable computing. Anticipated challenges include the complexity of integrating autonomous systems with existing infrastructures and the difficulty in accurately modeling and predicting behavioral patterns across diverse datasets. To address these, we will employ iterative testing and simulation and collaborate with industry partners to ensure practical applicability.

References

Barroso, L. A., & Hölzle, U. (2007). The case for energy-proportional computing. *Computer, 40*(12), 33–37. https://doi.org/10.1109/MC.2007.443

Bronstein, M. M., Bruna, J., LeCun, Y., Szlam, A., & Vandergheynst, P. (2017). Geometric deep learning: Going beyond Euclidean data. *IEEE Signal Processing Magazine, 34*(4), 18–42. https://doi.org/10.1109/MSP.2017.2693418

Chen, T., Xu, B., Zhang, C., & Guestrin, C. (2016). Training deep nets with sublinear memory cost. *arXiv preprint arXiv:1604.06174*. https://doi.org/10.48550/arXiv.1604.06174

Dean, J., Corrado, G., Monga, R., Chen, K., Devin, M., Mao, M., Ranzato, M., Senior, A., Tucker, P., Yang, K., Le, Q. V., & Ng, A. Y. (2012). Large scale distributed deep networks. In *Advances in Neural Information Processing Systems 25 (NIPS 2012)* (pp. 1223–1231).

Ghodsi, A., Zaharia, M., Hindman, B., Konwinski, A., Shenker, S., & Stoica, I. (2011). Dominant resource fairness: Fair allocation of multiple resource types. In *Proceedings of the 8th USENIX Symposium on Networked Systems Design and Implementation (NSDI '11)* (pp. 323–336).

Grandl, R., Ananthanarayanan, G., Kandula, S., Rao, S., & Akella, A. (2014). Multi-resource packing for cluster schedulers. *ACM SIGCOMM Computer Communication Review, 44*(4), 455–466.

Greenberg, A., Hamilton, J. R., Jain, N., Kandula, S., Kim, C., Lahiri, P., Maltz, D. A., Patel, P., & Sengupta, S. (2009). VL2: A scalable and flexible data center network. *ACM SIGCOMM Computer Communication Review, 39*(4), 51–62.

Gu, J., Chowdhury, M., Shin, K. G., Zhu, Y., Jeon, M., Qian, J., Liu, H., & Guo, C. (2019). Tiresias: A GPU cluster manager for distributed deep learning. In *Proceedings of the 16th USENIX Symposium on Networked Systems Design and Implementation (NSDI '19)* (pp. 485–500).

Hamilton, W. L., Ying, R., & Leskovec, J. (2017). Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems 30 (NIPS 2017)* (pp. 1024–1034).

Hindman, B., Konwinski, A., Zaharia, M., Ghodsi, A., Joseph, A. D., Katz, R., Shenker, S., & Stoica, I. (2011). Mesos: A platform for fine-grained resource sharing in the data center. In *Proceedings of the 8th USENIX Symposium on Networked Systems Design and Implementation (NSDI '11)* (pp. 295–308).

Isard, M., Prabhakaran, V., Currey, J., Wieder, U., Talwar, K., & Goldberg, A. (2009). Quincy: Fair scheduling for distributed computing clusters. In *Proceedings of the 22nd ACM Symposium on Operating Systems Principles (SOSP '09)* (pp. 261–276).

Jeon, M., Venkataraman, S., Phanishayee, A., Qian, J., Xiao, W., & Yang, F. (2019). Analysis of large-scale multi-tenant GPU clusters for DNN training workloads. In *Proceedings of the 2019 USENIX Annual Technical Conference (USENIX ATC '19)* (pp. 947–960).

Kipf, T. N., & Welling, M. (2017). Semi-supervised classification with graph convolutional networks. In *Proceedings of the 5th International Conference on Learning Representations (ICLR '17)*.

Koomey, J., Berard, S., Sanchez, M., & Wong, H. (2011). Implications of historical trends in the electrical efficiency of computing. *IEEE Annals of the History of Computing, 33*(3), 46–54. https://doi.org/10.1109/MAHC.2010.28

Li, M., Andersen, D. G., Park, J. W., Smola, A. J., Ahmed, A., Josifovski, V., Long, J., Shekita, E. J., & Su, B. Y. (2014). Scaling distributed machine learning with the parameter server. In *Proceedings of the 11th USENIX Symposium on Operating Systems Design and Implementation (OSDI '14)* (pp. 583–598).

Netflix Tech Blog. (2017, March 2). Introducing Atlas: Netflix's primary telemetry platform. https://netflixtechblog.com/introducing-atlas-netflixs-primary-telemetry-platform-bd31f4d8ed9a

Parkes, D. C., Procaccia, A. D., & Shah, N. (2013). Beyond dominant resource fairness: Extensions, limitations, and indivisibilities. In *Proceedings of the 14th ACM Conference on Electronic Commerce (EC '13)* (pp. 808–825).

Peng, Y., Bao, Y., Chen, Y., Wu, C., & Guo, C. (2018). Optimus: An efficient dynamic resource scheduler for deep learning clusters. In *Proceedings of the 13th EuroSys Conference (EuroSys '18)* (pp. 1–14).

Prequel. (2024). Common Reliability Enumerations: Community-driven problem detection. https://docs.prequel.dev/

Qiao, A., Choe, S. K., Subramanya, S. J., Neiswanger, W., Ho, Q., Zhang, H., Ganger, G. R., & Xing, E. P. (2021). Pollux: Co-adaptive cluster scheduling for maximizing goodput in deep learning. In *Proceedings of the 15th USENIX Symposium on Operating Systems Design and Implementation (OSDI '21)* (pp. 1–18).

Schwarzkopf, M., Konwinski, A., Abd-El-Malek, M., & Wilkes, J. (2013). Omega: Flexible, scalable schedulers for large compute clusters. In *Proceedings of the 8th ACM European Conference on Computer Systems (EuroSys '13)* (pp. 351–364).

Strubell, E., Ganesh, A., & McCallum, A. (2019). Energy and policy considerations for deep learning in NLP. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL '19)* (pp. 3645–3650).

Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., & Bengio, Y. (2018). Graph attention networks. In *Proceedings of the 6th International Conference on Learning Representations (ICLR '18)*.

Verma, A., Pedrosa, L., Korupolu, M., Oppenheimer, D., Tune, E., & Wilkes, J. (2015). Large-scale cluster management at Google with Borg. In *Proceedings of the 10th European Conference on Computer Systems (EuroSys '15)* (pp. 1–17).

Weng, Q., Yang, L., Yu, Y., Wang, W., Tang, X., Yang, G., & Zhang, L. (2023). Beware of fragmentation: Scheduling GPU-sharing workloads with fragmentation gradient descent. In *Proceedings of the 2023 USENIX Annual Technical Conference (USENIX ATC '23)* (pp. 995–1008).

Xiao, W., Bhardwaj, R., Ramjee, R., Sivathanu, M., Kwatra, N., Han, Z., Patel, P., Peng, X., Zhao, H., Zhang, Q., Yang, F., & Zhou, L. (2018). Gandiva: Introspective cluster scheduling for deep learning. In *Proceedings of the 13th USENIX Symposium on Operating Systems Design and Implementation (OSDI '18)* (pp. 595–610).

Zaharia, M., Borthakur, D., Sen Sarma, J., Elmeleegy, K., Shenker, S., & Stoica, I. (2010). Delay scheduling: A simple technique for achieving locality and fairness in cluster scheduling. In *Proceedings of the 5th European Conference on Computer Systems (EuroSys '10)* (pp. 265–278).