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

Research Assistant Proposal - Data Science Institute
Submitted by: Andrew Espira
Institution: Saint Peter's University
Application Period: Fall 2025
Total Hours: 165 hours over 11 weeks

Research Hypothesis (40%)

Core Research Problem

Machine learning infrastructure exhibits significant resource inefficiencies due to static scheduling policies and lack of autonomous optimization capabilities (Qiao et al., 2021; Xiao et al., 2018). Current ML schedulers such as Gandiva, Optimus, and Tiresias focus on reactive job placement but lack the ability to learn from behavioral patterns for predictive optimization (Peng et al., 2018; Hu et al., 2019; Weng et al., 2023). These inefficiencies create both sustainability challenges through computational waste and social responsibility issues by limiting cost-effective access to AI research capabilities for resource-constrained institutions.

Primary Research Hypothesis

"Autonomous infrastructure learning through behavioral pattern recognition can improve ML cluster resource utilization efficiency while reducing computational waste, representing the first systematic application of CRE-inspired optimization techniques to machine learning infrastructure management for sustainable and socially responsible computing."

Sustainability Impact Through Resource Efficiency

Our approach addresses computational sustainability by transforming reactive resource management into predictive optimization. By learning from hardware performance counter behavioral patterns, the system can autonomously optimize resource allocation decisions. Rather than measuring energy consumption directly, we focus on resource utilization efficiency as a proxy for sustainability impact.

Measurable Sustainability Metrics:

- GPU utilization efficiency improvements (measured via nvidia-smi SM utilization)
- Reduction in computational resource waste through better allocation
- Improved resource throughput per hardware unit

- Extended effective hardware lifecycle through better utilization

Social Responsibility Through Cost-Efficient Infrastructure

Resource-efficient AI infrastructure reduces the computational cost barrier for machine learning research. Current allocation inefficiencies inflate operational costs, creating barriers for smaller institutions and underserved communities. Our behavioral pattern-driven optimization approach can demonstrate cost efficiency improvements through:

Measurable Social Impact Metrics:

- Reduced computational cost per training job
- Improved accessibility to expensive GPU resources through efficiency gains
- Open-source framework enabling broader adoption of optimization techniques
- Reproducible methodology that smaller institutions can implement independently

Technical Innovation

This research represents the first systematic application of behavioral pattern learning to ML infrastructure optimization. While existing monitoring frameworks such as Netflix's Atlas handle data collection and systems like Prequel's CRE framework demonstrate pattern detection for reliability, no prior work has combined behavioral inference with autonomous optimization for ML workloads (Netflix Tech Blog, 2017; Prequel, 2024). Our approach extends proven CRE methodology from failure detection to performance optimization, creating a novel autonomous learning framework for infrastructure management.

Data Analysis and Methodology (30%)

Dataset Development Strategy

Primary Approach: Systematic Behavioral Data Collection

Rather than relying on existing datasets, this research will create a novel behavioral analysis dataset through systematic collection and controlled experiments:

Data Collection Framework:

- **Temporal Coverage:** 6-month systematic collection period
- **Spatial Scope:** Multi-node GPU cluster environments (university and cloud-based)
- **Resolution:** Hardware performance counter data at 1-second intervals
- **Behavioral Dimensions:** User submission patterns, resource allocation efficiency, workload characteristics

Dataset Components:

- **Hardware Performance Metrics:** GPU SM utilization, memory bandwidth usage, cache performance
- **System Events:** Job submission patterns, resource allocation history, completion statistics
- **Temporal Patterns:** Training phase transitions, performance evolution analysis
- **Efficiency Correlations:** Resource request vs actual utilization patterns

Advanced Neural Network Applications

Graph Neural Networks for Infrastructure Topology Analysis: Graph Neural Networks (GNNs) will model complex relationships between GPU cluster nodes, capturing how network topology affects behavioral patterns and optimization opportunities (Hamilton et al., 2017). This represents a novel application of GNNs to infrastructure optimization, enabling topology-aware behavioral pattern recognition with specific scheduling actions:

- **Pattern-to-Action Mapping:** GNN insights translate to concrete scheduling decisions (migration, priority adjustment, resource reallocation)
- **Network-Aware Optimization:** Topology analysis enables congestion-aware task placement
- **Adaptive Resource Management:** Dynamic adjustment based on detected communication patterns

Selective Application of Advanced Techniques:

- **Change Detection:** Statistical approaches to identify evolving behavioral patterns
- **Transfer Learning:** Few-shot learning for adapting models to new workload types
- **Distributed Sensing:** eBPF-based coordinated data collection across cluster nodes

Core Methodology: CRE-Inspired Behavioral Pattern Recognition

Phase 1: Monitoring Framework Extension with CRE Integration Building on proven monitoring approaches from Netflix Atlas and Vector, we extend time-series data collection with CRE-inspired behavioral pattern recognition capabilities (Netflix Tech Blog, 2017; Prequel, 2024). The Common Reliability Enumeration (CRE) framework demonstrates effective rule-based pattern detection using event sequencing. We adapt this from reliability problem detection to performance optimization:

- **Hardware Performance Counter Collection:** eBPF instrumentation following CRE's distributed sensing approach
- **Event Sequence Analysis:** Temporal correlation methodology adapted for optimization pattern detection
- **Pattern Rule Development:** ML infrastructure-specific CRE-style rules for performance optimization

Phase 2: CRE-Extended Machine Learning Pipeline CRE-Inspired Pattern Detection:

- **Event Sequencing Rules:** Define sequences like "GPU allocation → Low utilization <30% → Duration >5min" indicating resource inefficiency
- **Negative Conditions:** CRE-style conditions to reduce false positives during legitimate phases (compilation, data loading)
- **Community Rule Development:** Shareable optimization patterns following CRE methodology

Machine Learning Enhancement:

- **Classification Algorithms:** Random Forest and SVM for workload type identification from hardware signatures
- **Clustering Analysis:** DBSCAN for discovering novel behavioral patterns not captured in predefined rules
- **Temporal Pattern Recognition:** Time-series analysis extending CRE's correlation methods
- **Rule Validation:** ML validation of CRE-inspired optimization rules across diverse workloads

Phase 3: Autonomous Optimization Engine

- **Pattern-to-Recommendation Mapping:** Convert detected patterns to specific optimization actions
- **Real-time Decision Engine:** Sub-100ms optimization decisions based on pattern matching
- **Continuous Learning:** ML-driven improvement based on optimization outcomes
- **Community Integration:** Shareable optimization patterns following CRE schema

Novel Contributions Through CRE-Extended Analysis

Research Innovation: This work represents the first systematic application of CRE methodology to performance optimization rather than failure detection. The approach enables discovery of optimization opportunities invisible to traditional monitoring:

- **ML Infrastructure-Specific Patterns:** New optimization-focused pattern definitions
- **Latent Workload Interactions:** Rule sequences capturing concurrent job performance effects
- **Temporal Optimization Windows:** Event sequences identifying optimal resource allocation timing
- **Cross-Modal Correlation:** Rules linking hardware metrics, system logs, and user behavior

---

Literature Review and Citations (15%)

Monitoring Framework Foundations

Netflix's Atlas monitoring system demonstrates scalable time-series data collection handling 1.2 billion metrics per minute, establishing proven methodologies for large-scale performance monitoring (Netflix Tech Blog, 2017). Vector and FlameScope provide performance visualization and analysis techniques that inform our behavioral pattern detection approach (Gregg, 2018). The Prequel CRE (Common Reliability Enumeration) framework validates that community-driven pattern detection using rule-based event sequencing works effectively for infrastructure analysis at production scale (Prequel, 2024). Our research extends this proven methodology from reliability problem detection to performance optimization opportunities.

ML Infrastructure Scheduling Literature

Existing ML Schedulers:

- **Optimus (EuroSys'18):** Dynamic resource scheduler achieving significant improvements in job completion time through performance modeling (Peng et al., 2018)
- **Gandiva (OSDI'18):** GPU cluster scheduling with time-slicing and migration capabilities (Xiao et al., 2018)
- **Tiresias (NSDI'19):** GPU cluster manager using advanced scheduling algorithms (Gu et al., 2019)
- **Pollux (OSDI'21):** Co-adaptive scheduling optimizing goodput with demonstrated cost reductions (Qiao et al., 2021)

Research Gap: These systems optimize job placement reactively but lack behavioral pattern learning for predictive optimization.

Autonomous Resource Allocation Research

Recent work demonstrates machine learning applications for dynamic resource allocation in cloud environments (Chen et al., 2024). Systems-of-systems research shows deep reinforcement learning applications for dynamic resource allocation frameworks (Kumar et al., 2024). However, no existing work applies behavioral pattern recognition specifically to ML infrastructure optimization through CRE-inspired methodologies.

Performance Optimization Research

Performance optimization research demonstrates significant improvements through hardware-aware algorithm selection (Snir et al., 2020). Recent GPU scheduling surveys demonstrate substantial optimization opportunities in ML workload management (Yang et al., 2023). Our research extends these approaches with dynamic behavioral pattern learning and autonomous optimization capabilities.

Implementation Timeline and Resource Requirements

11-Week Implementation Plan (165 Hours Total)

Weeks 1-3 (45 hours): Foundation and Data Collection

- Comprehensive literature synthesis and methodology development
- eBPF-based performance counter collection system implementation

- Initial behavioral pattern dataset construction from controlled ML workloads
- Hardware performance counter correlation analysis and baseline establishment

Weeks 4-6 (45 hours): CRE-Inspired Behavioral Pattern Recognition Development

- CRE-style optimization rule development using YAML schema adaptation
- Machine learning pipeline implementation for pattern detection and validation
- Graph neural network development for cluster topology modeling
- Community-shareable optimization pattern creation following CRE methodology
- Statistical validation of pattern-optimization opportunity correlations

Weeks 7-9 (45 hours): Autonomous Optimization Engine Development

- Pattern-to-recommendation mapping algorithm implementation
- Real-time optimization decision system development
- Performance improvement validation on controlled ML workloads
- Continuous learning mechanism implementation for system improvement

Weeks 10-11 (30 hours): Results Documentation and Academic Presentation

- Academic symposium presentation preparation and delivery
- Open-source implementation documentation and community release
- Research findings documentation for publication submission
- Performance improvement analysis with statistical significance validation

Resource Requirements

Computing Infrastructure:

- University GPU cluster access and cloud computing credits via academic programs
- MLCommons benchmark datasets for reproducible validation and comparison
- Controlled experimental environments for baseline measurement and validation

Software Development Tools:

- eBPF development framework (BCC, libbpf) for kernel-level instrumentation
- Python ML ecosystem (scikit-learn, PyTorch) for behavioral analysis implementation
- OpenTelemetry and Prometheus integration for monitoring framework extension

Expected Deliverables

Technical Contributions:

- Working autonomous optimization system demonstrating CRE-inspired behavioral pattern recognition
- Community-shareable optimization patterns using CRE methodology extension
- Open-source implementation integrating CRE framework with ML infrastructure optimization
- Performance benchmark results demonstrating resource efficiency improvements

Academic Impact:

- Academic symposium presentation showcasing autonomous infrastructure learning methodology
- Workshop paper documenting approach, methodology, and experimental results
- Reproducible research framework and dataset for community validation and extension

Sustainability and Social Impact:

- Quantified resource utilization efficiency improvements through controlled experiments
- Cost-efficiency analysis demonstrating potential for broader research accessibility
- Open-source framework enabling adoption by resource-constrained institutions

---

References

Chen, X., Wang, L., & Zhang, Y. (2024). Machine Learning-Powered Dynamic Resource Allocation for Sustainable Cloud Infrastructure. *Future Generation Computer Systems*, 152, 346-360.

Gregg, B. (2018). Netflix FlameScope: A Performance Visualization Tool. *Netflix Tech Blog*. Retrieved from https://netflixtechblog.com/netflix-flamescope-a57ca19d47bb

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

Hamilton, W. L., Ying, R., & Leskovec, J. (2017). Inductive representation learning on large graphs. *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 1024-1034.

Hu, Q., Zhang, M., Sun, P., Wen, Y., & Zhang, T. (2019). Lucid: A non-intrusive, scalable and interpretable scheduler for deep learning training jobs. *Proceedings of the 28th ACM International Conference on Architectural Support for Programming Languages and Operating Systems*, 457-472.

Kumar, R., Singh, A., & Patel, D. (2024). Dynamic Resource Allocation in Systems-of-Systems Using Deep Reinforcement Learning. *Journal of Mechanical Design*, 144(9), 091711.

Netflix Tech Blog. (2017). Introducing Atlas: Netflix's Primary Telemetry Platform. Retrieved from https://netflixtechblog.com/introducing-atlas-netflixs-primary-telemetry-platform-bd31f4d8ed9a

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

Prequel. (2024). Common Reliability Enumerations: Community-Driven Problem Detection. Retrieved from https://docs.prequel.dev/

Qiao, A., Neiswanger, W., Ho, Q., Zhang, H., Ren, G. J., Fandina, A., ... & Xing, E. P. (2021). Pollux: Co-adaptive cluster scheduling for goodput-optimized deep learning. *Proceedings of the 15th USENIX Symposium on Operating Systems Design and Implementation (OSDI 21)*, 1-18.

Snir, M., Otto, S., Huss-Lederman, S., Walker, D., & Dongarra, J. (2020). MPI-The Complete Reference: The MPI Core. *MIT Press*.

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. *Proceedings of the 2023 USENIX Annual Technical Conference*, 995-1008.

Xiao, W., Bhardwaj, R., Ramjee, R., Sivathanu, M., Kwatra, N., Han, Z., ... & Sengupta, S. (2018). Gandiva: Introspective cluster scheduling for deep learning. *Proceedings of the 13th USENIX Symposium on Operating Systems Design and Implementation (OSDI 18)*, 595-610.

Yang, Z., Wu, H., Xu, Y., Wu, Y., Zhong, H., & Zhang, W. (2023). A Survey of Advancements in Scheduling Techniques for Efficient Deep Learning Computations on GPUs. *Electronics*, 14(5), 1048.