CHTC

UW CS

HTC26

Dynamic Job Lease Optimization in HTCondor

Khyathi Vagolu

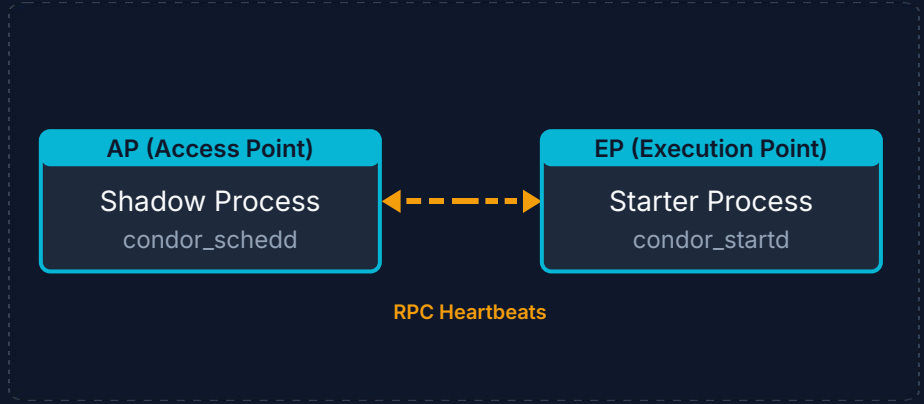
Collaborators:
Prof. Josiah Hanna (UW-Madison)
Todd Tannenbaum, Greg Thain (CHTC Team)

WISCONSIN
UNIVERSITY OF WISCONSIN–MADISON

CHTC Center For High Throughput Computing

Motivation: What happens under the hood?

- AP (Access Point) runs `condor_schedd` which spawns the Shadow process.
- EP (Execution Point) runs `condor_startd` which spawns the Starter process.
- Heartbeats over the network pipe keep the job lease alive and active.
- When the connection breaks, HTCondor enters a reconnection window (static, 40 minutes).



Problem Statement: Job Lease Timeout

"Why is my job suddenly idle or terminated?" - CHTC User

The Challenge

- Common pain: Unexpected terminations or long idle states
- Massive scale: Thousands of diverse jobs
- Network disruptions are a constant at scale
- Current solution: 40-min static lease timeout

The Objective

Minimise the number of jobs terminated or CPU cycles wasted

Can we do better than a static lease timeout?

The Trade-off: Idle Time vs. Job Failures

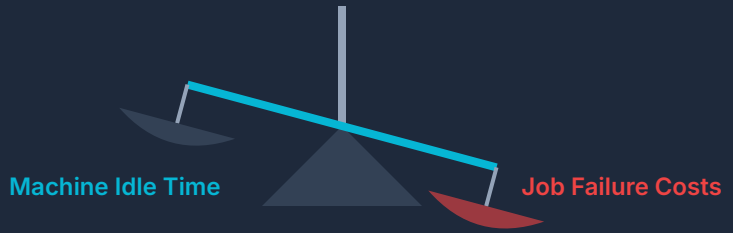
Operational Penalties

Over-Waiting: Slots tied up for machines already dead.

Under-Waiting: Killing 10-hour job for 5-sec transient blip.

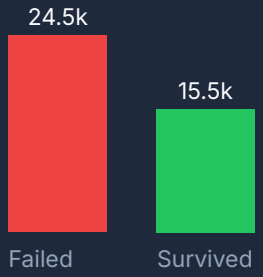
```
Static Baseline (T) = 40 minutes
```

Resource & Financial Impact

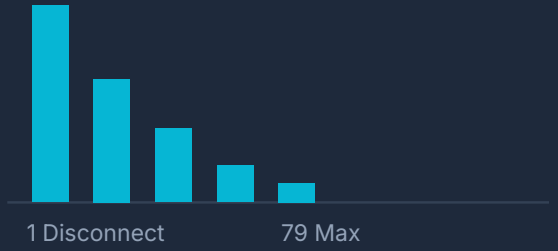


Understanding the scale from real Shadow Reconnect Logs

Job Outcomes (Reconnect Count)



Frequency of Disconnects



Work Done Before Disconnect (min)

Policy Scorecard: Static T4 = 40 min

77.3%
Success Rate

Wasted Wait: 3,180 Hours
Work Lost / Failure: 85.9 Min
Disconnects / Job: 4.19

Theoretical Simulation

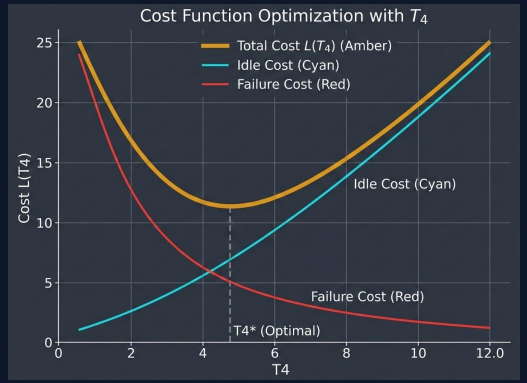
- Minimize Total Cost function: Minimize L(T)
- Variables: I(T) (Idle time), F(T) (Failure probability)

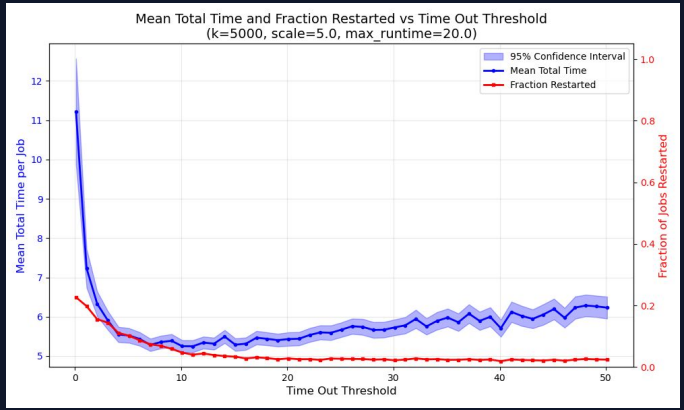
```
L(T) = C_idle * E[I(T)] + C_fail * E[F(T)]
```

Simulation parameters

```
K = 5000 jobs/batch Pareto
Fault Injection: Uniform 0-20
Max Retries: 10
Reconnections: 10 % permanent failure
Sweeping Bounds: [0.1, 50.0]
```

Simulation Data: Cost vs. Lease Duration





Machine Learning Techniques

Can we use Reinforcement Learning?

Learning the best action through trial and error by interacting with an environment, observing a cost, and adjusting a strategy.

Production Constraints

✕ Data Inefficiency: Standard RL needs 1000s of steps

✕ Risk: Cannot test arbitrary bad values on real jobs

The Sample-Efficient Path

Leveraging structural advantages of a single, bounded parameter.

No Neural Networks Needed

Bypass complex architectures for direct optimization.

Bayesian Optimization

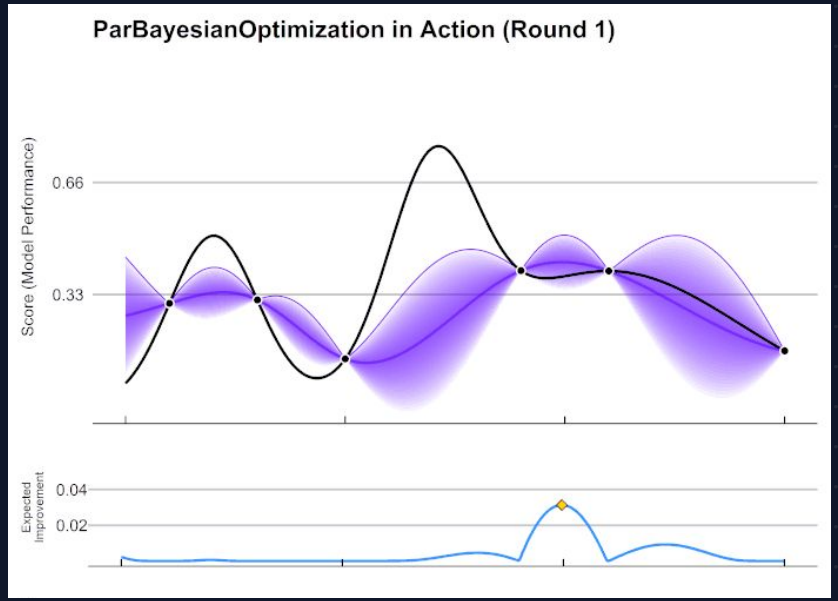
The True Optimum in a handful of trials

Bayesian Optimization: Modeling & Trade-offs

Bayesian Optimization uses Bayes' Theorem to combine prior beliefs with new data

- **Gaussian Process:** A probabilistic model mapping the complex loss landscape with predictive uncertainty.
- **Acquisition Function:** A decision rule that balances **Exploration** (searching high-variance areas) vs. **Exploitation** (refining known optima).
- **High Efficiency:** Reaches global optima in a handful of steps instead of thousands of brute-force trials.

$$\alpha(x) = E[\max(0, f(x*) - f(x))]$$



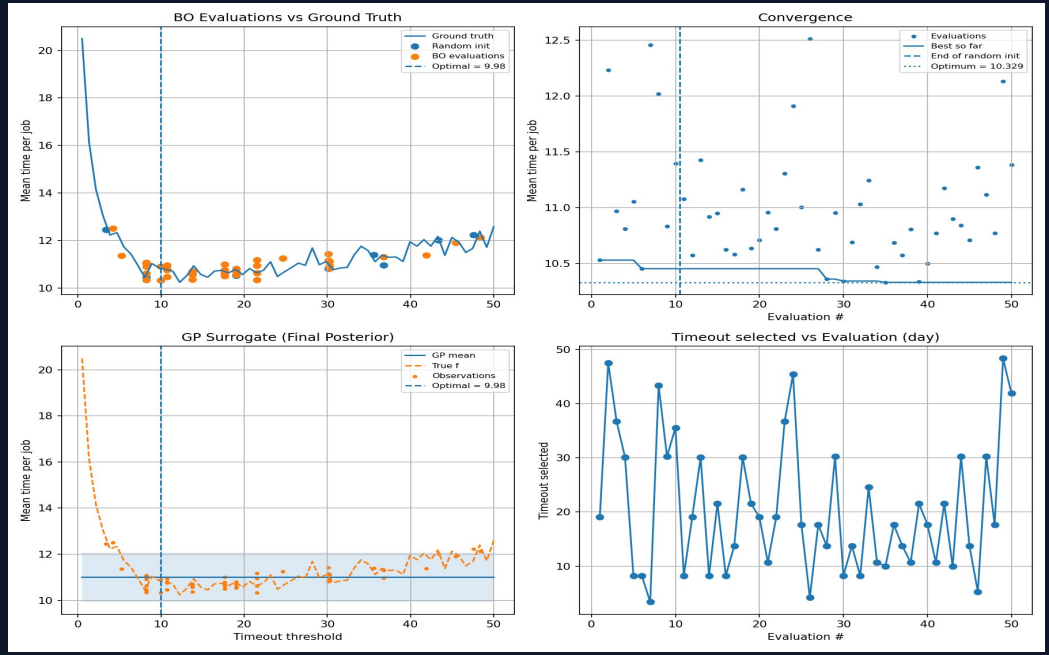
Simulation Results: Bayesian Optimization

BO Evaluations vs Ground Truth: BO focuses sampling near the true objective minimum

Convergence: After random initialization, the best value flatlines near the optimum (~9.98) within ~30 iterations

GP Surrogate: The model learns an approximation with uncertainty bands narrowing as samples accumulate

Exploration vs. Exploitation: Exploration settles into repeated selection of near-optimal values (~10) with occasional exploration



BO starts with random exploration, builds a surrogate model, and progressively focuses on the optimum i.e. finding the best timeout threshold in far fewer evaluations than a grid search.

Future Directions: Context-Aware Features & Adjustments

Real-System Deployment: Safely transitioning from simulation data to active learning on the live CHTC cluster (In Progress).

Context-Aware Adaptation: Engineering features to evaluate if optimal timeouts should depend on specific Job or Execution Point (EP) characteristics.

Continual Learning Framework: Moving beyond a static one-time fix to study how the optimal threshold shifts over time as cluster dynamics change.

Thank you!

Moving from hardcoded parameters to AI driven optimization.

Contact Information

Khyathi Vagolu: vagolu@wisc.edu

Josiah Hanna: jphanna@cs.wisc.edu

Todd Tannenbaum: tannenba@cs.wisc.edu

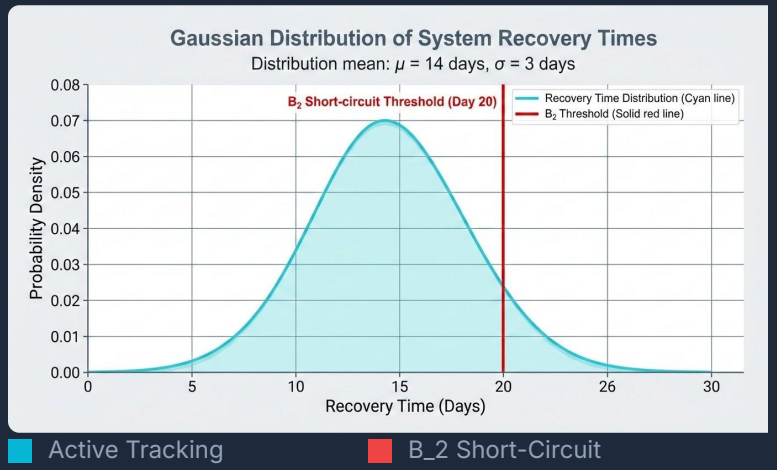
Greg Thain: gthain@cs.wisc.edu

Exploratory Data Analysis: Tracking and Short-Circuits

- B_2 identifies if the starter is definitively dead based on threshold breaches.
- Acts as a 'short-circuit' feature by halting recovery logic if terminal state is reached.
- Optimizes system resources by preventing redundant reconnection attempts.

```
T_1 (Start) = [Timestamp]
T_2 (Last Heartbeat) = [Timestamp]
T_3 (Reconnection) = [Timestamp]
B_2 (Critical Flag) = [Binary_State]
```

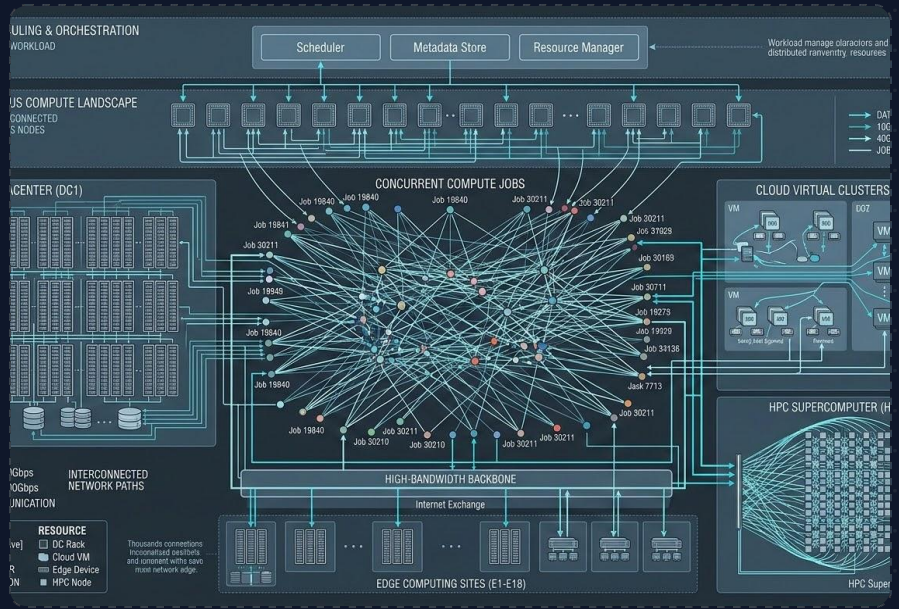
Recovery Mapping Distribution



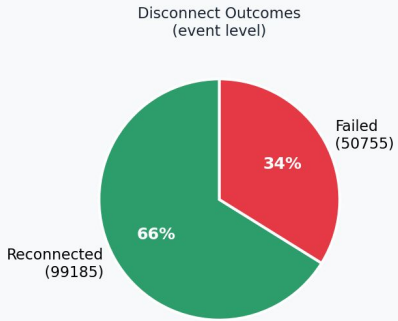
The Reality of Scale at CHTC

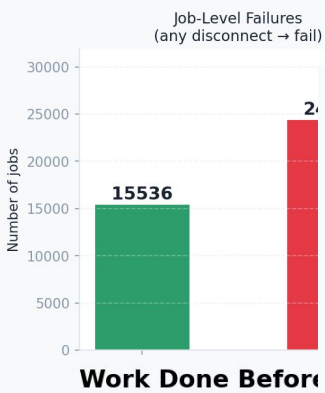
- CHTC processes thousands of daily jobs with high heterogeneity.
- Network Reality: Distinguish transient network blips from permanent hardware failures.

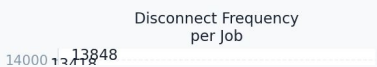
> *"Why is my job suddenly idle or terminated?"*
>
> - User Pain Point



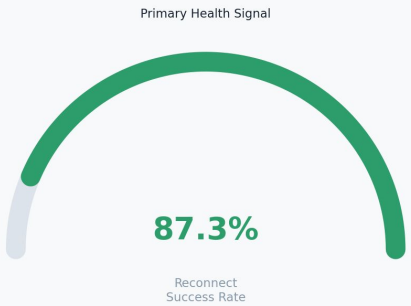
Overview — Disconnect Events (current T4 = 40 min)







Policy Score Card — Static T4 = 2400s (40 min)



Observable Metrics — Current Policy

Metric	Value	Interpretation
Reconnect success rate	87.3%	Primary: fraction of disconnects that self-healed
Wasted timeout wait	3180.67 h	n_fail × T4 = 4771 × 2400s
Mean work lost on failure	85.9 min	Avg T2-T1 at time of failure → CPU lost
Disconnects per job	4.19	Events / unique jobs; >1 = chronic instability
P(EP dead | fail)	46%	Fraction of fails where B2=True (EP confirmed dead)
Timeout efficiency	0.0000	reconnect_wait / T4 for successes (0 = instant ✓)
Needless restarts if T4=6	0	Successes cut off by smaller T4 (counterfactual)

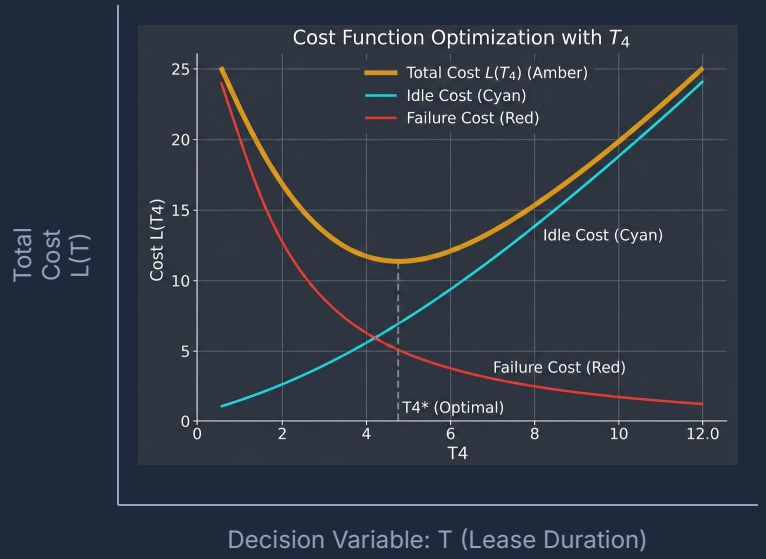
Theoretical Simulation

- Minimize Total Cost function: Minimize $L(T_4)$
- Variables: $I(T_4)$ (Idle time), $F(T_4)$ (Failure probability)

```
L(T) = C_idle * E[I(T)] + C_fail * E[F(T)]
```

```
K = 2000 | MAX_RUNTIME = 20s | SCALE = 5s
P_INF_RECONNECT = 10% | N_ITER = 40
M_MAX_RESTARTS = 10
T4_BOUNDS = [0.5s, 50.0s]
```

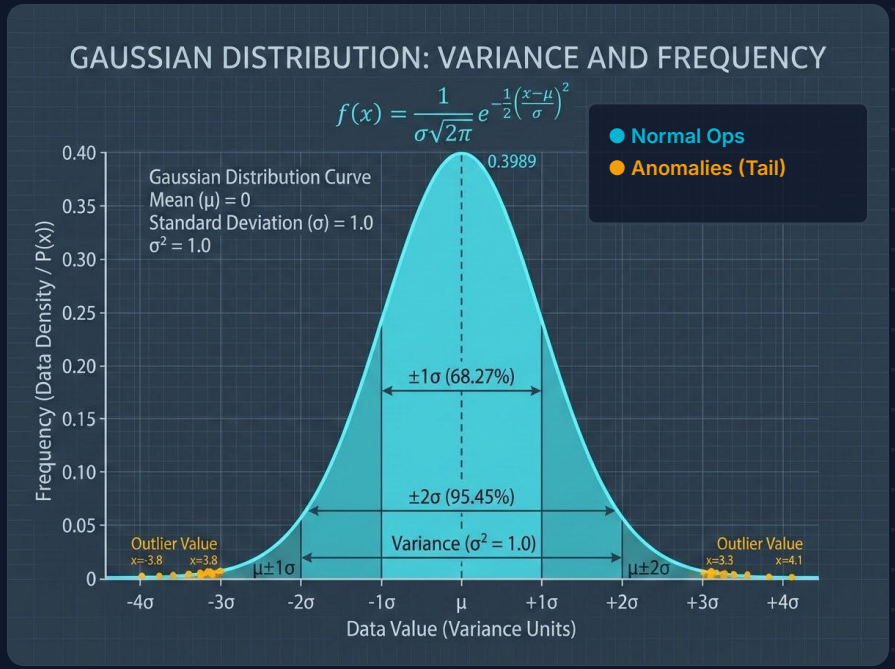
Simulation Data: Cost vs. Lease Duration



Gaussian Distribution: Modeling Infrastructure Data

- Modeling infrastructure data blips and failures using a Gaussian distribution.
- Capturing variance and frequency of network anomalies within distributed systems.
- Statistical modeling enables predictive resource allocation and lease optimization.

```
f(x | μ, σ²) = [1/√(2πσ²)] * e^[-(x-μ)² / 2σ²]
```



SGD to Model Timeout Parameter

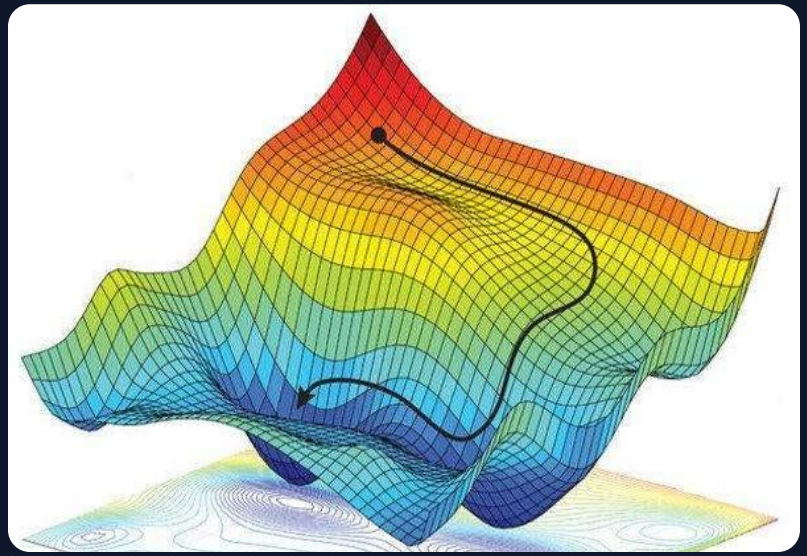
Standard SGD

- Requires differentiable gradients
- Assumes smooth landscapes

GAP

Infrastructure Reality

- Non-differentiable functions
- Noisy & black-box data



1. Compute Gradient
Find local loss slope

2. Update Params
Step toward minimum

Infrastructure Failure
Gradients cannot be computed on discrete, noisy data.