Proceedings of the 20th Conference on Computer Science and Intelligence Systems (FedCSIS) pp. 1–2
DOI: 10.15439/2025F4345
ISSN 2300-5963 ACSIS, Vol. 43

# Active Inference in the Distributed Computing Continuum

Schahram Dustdar,
TU Wien, Austria

***Abstract*—Distributed applications now span sensors, edge nodes, fog clusters, and hyperscale clouds. Meeting service-level objectives across this "Distributed Computing Continuum" persistently fails when management is reactive, centralized, and blind to uncertainty. I argue for *predictive equilibrium* as the control objective and for a concrete diagnostic: the Kullback–Leibler divergence between a system's expected and observed causal behavior under perturbations, each modeled with a Bayesian network. This perspective draws from predictive regulation in neuroscience and the fluctuation–dissipation view of equilibrium in physics, and it sets the stage for *antifragility*—systems that get better because they were stressed, not despite it.**

## I. Introduction

THE practical difficulty of continuum computing is not merely scale or heterogeneity but *nonstationarity*: workloads, topologies, energy states, and network conditions drift faster than operators can retune thresholds. Threshold rules act only after damage is visible; they presume that the future resembles the past and that a central brain has enough signal to decide well for everyone. In real deployments, devices often act with partial information, and local decisions interact in nontrivial ways, which makes steady Service Level Objective (SLO) compliance brittle. A different organizing principle is needed—one that regulates for stability while learning in the face of novelty.

A helpful lens comes from biology. Bodies do not maintain temperature or glucose by passively waiting for deviations; they *predict* and prepare. **Homeostasis** is reactive stabilization; **allostasis** generalizes it to **predictive regulation**, adjusting internal states in anticipation of expected demands. The **Free Energy Principle** reframes this as minimizing prediction error so organisms preserve their form and functions. Transported to engineered systems, the analogue is clear: components should anticipate loads, not just recover from violations, and success should be judged by how well predictions align with reality when the world is nudged.

The second lens is physical. In many systems, the relationship between internal fluctuations and responses to small perturbations is codified by the **Fluctuation–Dissipation Theorem**; departures from that relationship mark distance from equilibrium. Recent studies show how FDT violations reveal nonequilibrium brain dynamics. For computing, the physics is an analogy rather than a derivation, but it is a useful one: if a deployment's internal model predicts how a nudge will propagate and the observed response matches, we are near equilibrium; if prediction and observation disagree, we are not.

Putting these lenses together yields a working definition: **predictive equilibrium** is the condition in which a continuum system's internal model accurately represents its behavior under perturbations, such that deviations remain small enough to meet goals. This is not a static point but an active property sustained through ongoing modeling, anticipation, and reorganization. It entails dynamic balance (local parts keeping global SLOs on track), continuous reconfiguration (structure in service of goals), and predictive consistency (forecasts match outcomes under realistic nudges). Equilibrium so defined is the platform upon which antifragility can emerge.

To operationalize the idea, represent the deployment as a **Bayesian network** whose nodes are salient metrics and whose directed edges encode probabilistic dependencies within and across tiers. Choose a small, actionable vocabulary—latency percentiles by tier, queue depths, utilization, link RTT/loss, battery state, SLO flags—so the model remains interpretable and fast to update. Then specify a **perturbation** with clear semantics (bandwidth throttle, dependency delay, CPU cap, synthetic burst). Apply it twice: once to the model to produce an **expected** network (updating the affected conditionals) and once to the live system or a faithful twin to produce an **observed** network. The **KL divergence** between these distributions is the "distance-from-equilibrium" signal. Small divergence indicates predictive alignment; a spike signals a breakdown that warrants model revision, policy change, or both.

Why privilege KL divergence over raw SLOs? A latency breach only states that a guarantee failed. KL divergence localizes *why* predictions broke: edges may rewire (emergent dependency), weights may shift (weaker coupling), or noise may increase (stochasticity rose), all of which are visible in the comparative structure and parameters of the two networks. Moreover, the same divergence that warns of impending SLO drift

doubles as a *learning* signal for improving the internal model—a property traditional thresholds lack. Conceptually, this aligns with active-inference accounts where action is chosen to balance pragmatic value and information gain; minimizing KL between expected and observed behavior is precisely minimizing that mismatch.

An architecture built around predictive equilibrium emphasizes **local modeling, safe perturbation, and hierarchical synthesis**. Edge devices maintain compact, incremental BNs over their local neighborhoods and expose an "expectation API" that answers *What should happen here if bandwidth drops by 20% for 30 seconds?* Fog nodes orchestrate small, isolated perturbation campaigns (canaries, synthetic traffic), fit the observed BN, and compute divergence, while also enacting reorganizations—operator placement, replication factors, routing. A cloud-level meta-controller aggregates divergence summaries and, crucially, modulates the exploration–exploitation balance. Here the physics analogy pays off again: use an **FDT-inspired signal** as a *modulator* on the active-inference drive, transiently biasing decisions toward information gain when prediction degrades, then annealing back as equilibrium returns.

Consider a city-scale video analytics deployment. A fog controller periodically reduces uplink bandwidth by fifteen percent for a handful of cameras. The expected BN predicts mild queue growth and a compensatory frame-rate dip, preserving tail latency. Instead, observations reveal that a co-located GPU tenant injects bursty contention, creating a new dependency from that tenant's utilization to the canary's latency. KL divergence jumps. The controller updates the model (adding the edge), raises isolation for GPU tenants in that cell, and adjusts placement to avoid co-scheduling bandwidth-sensitive pipelines with bursty neighbors. In the next probe, divergence falls and the new policy holds. The system did not merely "heal"; it learned a structural lesson and retained it—an instance of antifragility in miniature.

Antifragility is not a slogan but a design criterion: systems should improve *because* stress exposes mismatches. In Taleb's sense, antifragile entities gain from volatility. In our setting, the gain is a better causal model and a sharper policy encoded in the BN and the controller; perturbations become training data that increase predictive fidelity and coordination skill. Equilibrium supplies the safety rails—keeping the system within acceptable performance bands while it harvests information from controlled nudges. There are limits. This is a **conceptual** framework; practical feasibility and cost remain to be demonstrated at scale. Perturbations must be safe and ethically scoped. BN learning must be sparse, incremental, and bounded in overhead. Multi-objective trade-offs (latency vs. energy vs. privacy), partial observability, and compliance constraints complicate modeling and may require task-specific variables and priors. Yet these are engineering questions, not theoretical roadblocks. A sensible validation path is to start in a digital twin, calibrate divergence thresholds to the system's natural variability, and then graduate to canary slices in production.

In sum, predictive equilibrium reframes continuum operations as a dialogue between model and world. Neuroscience supplies the instinct—prepare rather than react—and physics offers a ruler for distance from equilibrium. With Bayesian networks to represent expectations and KL divergence to measure their failure modes, we obtain both an early-warning gauge and a learning gradient. That combination is the essence of antifragility in engineered systems: *stay stable enough to learn, and learn enough to become more stable the next time*. In our previous work [1-5] some of the aspects above are detailed.

## References


[1] S. Dustdar, V. Casamayor Pujol, and P. K. Donta, "On Distributed Computing Continuum Systems," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 4, pp. 4092–4105, Apr. 2023, doi: 10.1109/TKDE.2022.3142856.

[2] V. Casamayor Pujol, B. Sedlak, Y. Xu, P. K. Donta, and S. Dustdar, "DeepSLOs for the Computing Continuum," in *Proceedings of the 2024 Workshop on Advanced Tools, Programming Languages, and PLatforms for Implementing and Evaluating algorithms for Distributed systems*, in ApPLIED'24. New York, NY, USA: Association for Computing Machinery, Jun. 2024, pp. 1–10. doi: 10.1145/3663338.3663681.

[3] B. Sedlak, V. C. Pujol, P. K. Donta, and S. Dustdar, "Equilibrium in the Computing Continuum through Active Inference," *Future Gener. Comput. Syst.*, May 2024, doi: 10.1016/j.future.2024.05.056.

[4] V. Casamayor Pujol, B. Sedlak, P. K. Donta, and S. Dustdar, "On Causality in Distributed Continuum Systems," *IEEE Internet Comput.*, vol. 28, no. 2, pp. 57–64, Mar. 2024, doi: 10.1109/MIC.2023.3344248.

[5] V. C. Pujol, B. Sedlak, T. Salvatori, K. Friston, and S. Dustdar, "Distributed Intelligence in the Computing Continuum with Active Inference," May 30, 2025, *arXiv*: arXiv:2505.24618. doi: 10.48550/arXiv.2505.24618.