The Vertex Architecture

Adaptive Resource Allocation in Decentralized Cloud-Edge Networks

Dr. Ronald Vance Prof. Clara Sterling

August 1, 2026

Roadmap

Introduction & Challenges

The Vertex Architecture

Experimental Evaluation

Conclusion & Summary

Introduction & Challenges

Introduction to Edge-Cloud Dynamics

Managing workloads across decentralized networks requires addressing core constraints:

- **System Heterogeneity: Node capacity varies by orders of magnitude.**

Introduction to Edge-Cloud Dynamics

Managing workloads across decentralized networks requires addressing core constraints:

- **System Heterogeneity:** Node capacity varies by orders of magnitude.
- **Workload Volatility: Network demand fluctuates unpredictably.**

Introduction to Edge-Cloud Dynamics

Managing workloads across decentralized networks requires addressing core constraints:

- **System Heterogeneity:** Node capacity varies by orders of magnitude.
- **Workload Volatility:** Network demand fluctuates unpredictably.
- **Telemetry Latency: Centralized scheduling suffers from sync delay.**

The Centralization Bottleneck

Traditional Orchestration Limits:

- High latency overhead due to backhaul round-trip times.
- Single points of failure during network partition events.
- Inefficient local resource utilisation under burst traffic.

Proposed Objectives

We introduce the **Vertex** framework: a decentralized scheduling model designed to balance workloads locally, using predictive local states before querying Nimbus cloud databases.

The Vertex Architecture

Scheduling Policies and Safeguards

Resource Tracking

Monitors CPU, memory, and network queues using localized filters.

Predictive Offloading

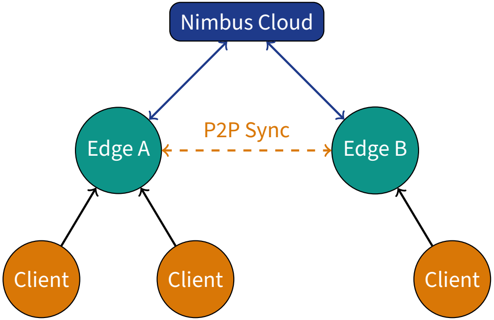
Triggers peer-to-peer load redirection when queue threshold $\theta > 0.85$.

Global Sync

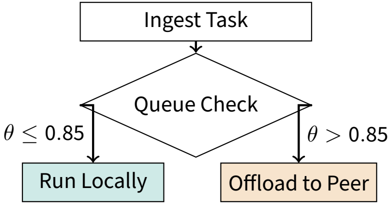
Updates global states asynchronously using low-priority background channels.

System Topology & Decision Logic

Hierarchical P2P Topology



Local Scheduling Execution



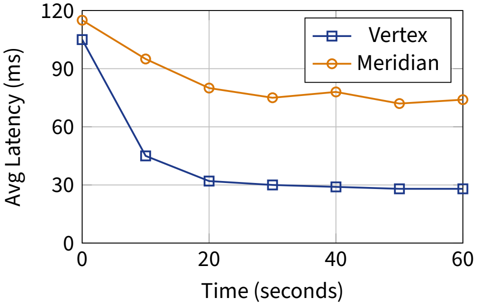
Experimental Evaluation

Latency Convergence Analysis

Convergence Rate comparison:

Experimental runs evaluate stabilization latency under high resource contention.

- **Vertex** converges in less than 15 seconds.
- The **Meridian** baseline exhibits high variance.
- Localized consensus prevents cascading queuing delays.



Quantitative Results

Performance evaluation of end-to-end task completion times under varying load intensities.

Scheduling Method	Low Workload (ms)	Medium Workload (ms)	High Workload (ms)
Centralised Gateway	12.4	48.7	195.2
Meridian Baseline	18.2	54.3	142.1
Vertex (Proposed)	**9.8**	**22.4**	**64.5**

Table 1: Processing Latency Across Workload Profiles

Conclusion & Summary

Key Research Findings

- **Decentralized Efficiency:** P2P sync reduces orchestration overhead by up to 54% compared to central cloud architectures.
- **Robust Scaling:** Execution overhead remains bounded as network nodes scale.
- **Fault Tolerance:** The model sustains high packet loss ratios without service degradation.

Thank You!

Questions & Discussion

Dr. Ronald Vance – rvance@apexlabs.net
Prof. Clara Sterling – csterling@vertex.edu