A Priority-Based Carbon-Aware Optimised Framework for Energy-Efficient Task Offloading in Fog-Cloud

Abstract

The increasing energy demands of modern computational tasks necessitate efficient resource management strategies that prioritize sustainability. This paper proposes a carbon-aware task offloading framework that optimizes task allocation across computational nodes based on energy availability, latency, and carbon emissions. Each computational node possesses two energy sources: (1) renewable energy (e.g., solar, wind) with a limited supply and (2) gas-based energy, which is unlimited but results in higher carbon emissions. The proposed model prioritizes nodes utilizing renewable energy whenever possible; however, if renewable energy is insufficient or inefficient due to excessive latency and energy consumption, tasks are allocated to nodes powered by gas-based energy.

To achieve optimal task allocation, we introduce an optimization-based fitness function that evaluates candidate node assignments based on three factors: energy consumption, latency, and carbon emissions. The decision-making process seeks to (1) minimize overall energy consumption, (2) reduce processing and transmission latency, and (3) lower carbon emissions. The proposed framework incorporates a metaheuristic optimization algorithm, such as a Genetic Algorithm (GA) or Particle Swarm Optimization (PSO), to determine the most efficient allocation strategy dynamically.

Keywords

carbon emission, energy consumption, offloading, edge-cloud, metaheuristic

ACM Reference Format:

. 2024. A Priority-Based Carbon-Aware Optimised Framework for Energy-Efficient Task Offloading in Fog-Cloud. In . ACM, New York, NY, USA, 2 pages. https://doi.org/XXXXXXX.XXXXXXX

As computational workloads increase in complexity, optimizing energy usage has become a critical challenge. Traditional task offloading strategies focus on performance metrics such as execution time and system utilization; however, emerging sustainability concerns necessitate a carbon-aware approach that integrates renewable energy prioritization.

This research introduces a multi-objective task scheduling framework that considers:

Energy efficiency: Minimizing the overall power consumption of computational nodes.

Latency optimization: Reducing transmission and processing delays.

Carbon footprint reduction: Favoring nodes with renewable energy sources before utilizing fossil-fuel-based alternatives.

To implement this strategy, we propose a fitness function that quantifies the trade-offs between energy consumption, latency, and carbon emissions. The system uses an optimization algorithm to dynamically allocate tasks to nodes in a way that minimizes energy wastage and environmental impact.

1 System Model and Optimization Problem

1.1 Problem Definition

We consider a time series of task generation with deadlines that must be executed on fog nodes. Each fog node is equipped with two types of energy resources: renewable and Gas-based Energy. The availability of renewable energy varies across nodes and time. Our objective is to allocate tasks to fog nodes efficiently, prioritizing nodes powered by renewable energy whenever possible. If no suitable renewable energy node is available and we will miss the deadline, tasks will be assigned to nodes utilizing Gas-based energy to ensure execution.

1.2 Computational Node Energy Model

Each computational node i has two energy resources:

- **Renewable Energy** (R_i): A limited source (e.g., solar, wind) with a finite energy budget per time unit.
- **Gas-based Energy** (G_i): An unlimited source but with high carbon emissions.

Each task j has an energy requirement E_{ij} when processed on node i, where the energy source is either renewable or gas-based.

1.3 Optimization Constraints

$$\sum_j T_j \leq C_i, \quad \forall i$$

Each node can only process a limited number of tasks based on its computational capacity C_i.

$$P_{ij} = 0 \Rightarrow Q_{ij} = 1$$

If a node has no renewable energy available, it must rely on gas-based resources.

2 Optimization Model and Fitness Function

We adopt a metaheuristic optimization algorithm to dynamically determine the most efficient task allocation strategy. The chosen algorithm optimizes node selection based on our fitness function. The considered metaheuristic techniques include:

- Energy Consumption: Minimizing the overall energy usage.
- Carbon Emissions: Energy Type and Availability Prioritize renewable-powered nodes if energy is available. Reducing

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-1-4503-XXXX-X/18/06

https://doi.org/XXXXXXX.XXXXXXX

the carbon footprint by prioritizing renewable energy resources. If renewables are exhausted, select gas-powered nodes.
- Computational Resources Ensure the node has enough CPU/memory to handle the task.
- Task Deadline: Ensuring that tasks meet their deadline constraints to maintain system reliability and performance.

Mathematical Model: Define an objective function: Minimize

$$\sum_{i=1}^{N} C_i P_i + G_i Q_i$$

Ci = Carbon cost per unit task on node i
Pi = Probability of selecting renewable energy on node i
Gi = Carbon cost for gas-powered node i
Qi = Probability of selecting a gas-powered node

Ci + Qi = 1 (A task is either processed on renewable or gas) Pi is higher when renewable energy is available.

My idea is evolving into a multi-objective optimization problem, where you aim to minimize energy consumption, latency, and carbon emissions while maximizing the use of renewable energy. The approach will involve a fitness function that evaluates different task allocation decisions and a framework for implementing the scheduling strategy.

Each computational node has:

- **Renewable Energy Availability** (R_i): Limited (e.g., 10 kJ per node).
- **Gas Energy Availability** (G_i): Unlimited, but emits higher carbon emissions.
- **Latency** (L_{ij}): Time taken for node i to process task j.
- **Energy Consumption** (E_{ij}): Energy used when node i processes task j.
- **Carbon Emission** (C_{ij}): Emissions based on energy source.

To evaluate a task allocation decision, we define a fitness function that balances

- **Energy Efficiency**: Prefer nodes with lower energy consumption.
- **Latency Minimization**: Reduce delay when assigning tasks.
- **Carbon Emission Reduction**: Prioritize renewable energy over gas.

The fitness function is:

$$F(T) = \alpha \sum_{i,j} E_{ij} + \beta \sum_{i,j} D_{ij} + \gamma \sum_{i,j} C_{ij}$$

Where:

- E_{ij} = Energy consumed by node i for task j.
- D_{ij} =denotes the penalty for tasks missing deadlines j on node i.
- C_{ij} = Carbon emission when task j is executed on node i.
- α, β, γ = Weights for trade-offs between energy, latency, and emissions.

3 Optimization Algorithm for Task allocation

To solve the optimization problem, we employ a Genetic Algorithm (GA) due to its robustness in handling multi-objective constraints. The steps include:

1. **Initialization**: Tasks arrive in a time-series manner with specific execution requirements and deadlines.
2. **Resource Assessment:** The availability of renewable and non-renewable energy on fog nodes is monitored to ensure efficient task execution.
3. **Fitness Evaluation**: Compute $F(T)$ for each candidate solution.
4. **Selection**: Choose the best-performing solutions.
5. **Crossover & Mutation**: Generate new candidate solutions through genetic operations.
6. **Convergence Check**: If no further improvement is observed, select the best allocation and execute the tasks accordingly.

4 Experimental Setup and Evaluation

The proposed model will be evaluated through simulations using realistic workload scenarios. Performance will be compared against traditional energy-agnostic task scheduling algorithms. Metrics include:

- **Total energy consumption**
- **Average task execution latency**
- **Carbon footprint reduction**