Distribution Network Distributed Resources Application Framework and Key Technologies Based on Cloud-Edge-Device Collaboration

Topic:

The paper discusses the application framework and key technologies for distributed resources in distribution networks based on cloud-edge-device collaboration. It emphasizes the need to adapt to new energy sources like photovoltaic, energy storage, and electric vehicles in the power grid using this collaborative approach.

Problem

The increased integration of distributed resources such as electric vehicles, distributed photovoltaic systems, and energy storage devices into power distribution networks presents significant challenges in data management. The traditional centralized cloud computing approach lacks the real-time processing capability to handle the massive and complex data generated by these resources effectively. The need for real-time responsiveness, secure data interaction, and efficient regulation of distributed resources necessitates a new approach.

Method

To address these challenges, the authors propose a framework based on cloud-edge-device collaboration. The framework leverages the computational power of the cloud, the flexibility of the edge layer, and the real-time responsiveness of the device layer to enhance data storage, processing, and application. The cloud layer handles large-scale data analysis and optimization, the edge layer processes regional data in real time, and the device layer collects and transmits data from distributed resources.

Results

The proposed framework enables two-way interaction between the distribution network and distributed resources, supporting hierarchical and distributed data storage and processing. The cloud-edge-device collaboration improves the efficiency of data mining at multiple levels and facilitates secure, timely, and reliable data exchange. Key technologies for the framework include intelligent Internet of Things mechanisms, local communication technologies, plug-and-play access for devices, accurate computation methods for regulation capacity, data modeling based on standards, and coordinated control of distributed resources.

Implications

Implementing this collaborative framework can significantly enhance the management and optimization of distributed energy resources in power distribution networks. The framework supports lean management, enhances service quality, and ensures secure data exchange, which is crucial for the advancement of the power Internet of Things. It can help achieve efficient regulation of distributed resources and contribute to the economic benefits of power systems.

Conclusion

The cloud-edge-device collaborative framework is presented as a robust solution for handling the challenges of distributed resource data management in distribution networks. The framework integrates the benefits of edge and cloud computing, facilitating multi-level data value extraction and business collaboration. It proposes a pathway to improve the safety, efficiency, and economic benefits of distributed resource management while addressing the limitations of traditional centralized cloud-based systems.

Towards Analyzing the Performance of Hybrid Edge-Cloud Processing

Topic:
This paper analyzes the performance of cloud-only, edge-only, and hybrid edge-cloud processing for big data analytics. It focuses on the performance trade-offs and develops a model to assess when and how processing should be distributed between the cloud and the edge.

Problem:
Organizations that operate in geographically distributed areas often use cloud computing for data collection and processing. However, the growth of edge computing offers opportunities to process data closer to its source, reducing network pressure and improving response times. A key challenge is deciding how much processing to perform at the edge versus in the cloud to optimize execution times while considering fluctuating edge-cloud bandwidth.

Method:
The authors develop an analytical model validated through measurements on edge devices (Nvidia Jetson TK1 and TX1) and Amazon Web Services (AWS) cloud platforms. The model assesses the performance of edge-only, cloud-only, and hybrid edge-cloud setups. The parameters considered include application characteristics, edge-cloud transfer bandwidth, processing times, and data transfer times. The authors measure the performance of seven MapReduce applications across these setups to evaluate their model.

Results:
The findings reveal a diverse performance landscape with no definitive winner between cloud-only, edge-only, and hybrid processing. Key factors affecting performance are the edge-cloud bandwidth and application characteristics, particularly the ratio of input and output data sizes (selectivity). For certain applications with small data transfers and significant local processing, edge-only or hybrid processing can outperform cloud-only setups. Conversely, cloud-only processing can be advantageous for applications with large data sets or when cloud processing speeds significantly surpass those at the edge.

Implications:
The paper's analysis shows that hybrid edge-cloud setups can be advantageous in reducing transfer times and improving execution speed for specific types of applications. The model provides a practical tool for system designers to evaluate the best deployment strategy based on application needs and edge-cloud bandwidth. The insights are particularly useful for designing systems that require real-time processing and efficient resource management.

Conclusion:
The study concludes that application selectivity, cloud speedup, and the computation-to-communication ratio are critical parameters for determining the optimal processing strategy. The authors' model and measurements offer a comprehensive method to analyze these parameters, guiding decisions on whether edge, cloud, or hybrid setups will provide the best performance for given use cases.

Decentralized Resource Auctioning for Latency-Sensitive Edge Computing

Topic:
The paper proposes a decentralized resource management technique for latency-sensitive IoT applications that leverages a decentralized resource auctioning system to allocate computational resources at the edge, ensuring compliance with latency and privacy constraints.

Problem:
The centralized cloud-based IoT infrastructure faces challenges in meeting latency and privacy requirements due to high latency, central points of failure, and difficulties with data privacy. The growth of IoT and edge devices demands decentralized resource management techniques to utilize computational resources near the devices efficiently. However, existing methods struggle to allocate resources effectively while ensuring low latency and data privacy.

Method:
The proposed approach is inspired by an auction house mechanism where devices bid to execute parts of an IoT application. It includes two modules:

1. **Deployment Policy Module**: The IoT application tasks are advertised to nearby devices (bidders), and based on their availability, devices bid for tasks. A dispatcher then allocates tasks to the most suitable devices, ensuring latency and resource constraints are met.

2. **Bidding Policy Module**: Devices generate bids based on available resources, ensuring privacy preferences are respected. Multiple bidding strategies are used to enhance task allocation, focusing on maximizing resource use while meeting application requirements.

Results:
The paper demonstrates that the proposed decentralized framework efficiently finds satisfiable solutions for resource allocation in edge computing environments. The framework uses Satisfiability Modulo Theories (SMT) to guarantee that if a valid allocation exists, it will be found, providing reliable task allocation under given constraints. Experimental results show that the approach can allocate resources effectively in edge environments with up to 20 nodes, finding solutions quickly (less than a second for smaller setups).

Implications:
The decentralized auctioning system enables efficient utilization of edge resources for latency-sensitive applications while preserving data privacy. It supports the deployment of IoT applications in real-time, near the data source, and eliminates the need for centralized control, making it highly suitable for dynamic and distributed IoT scenarios.

Conclusion:
The paper introduces a novel decentralized framework for latency-sensitive IoT applications that ensures privacy and meets latency constraints through a bidding mechanism. The solution is shown to be computationally efficient for realistic edge

network sizes, providing reliable task allocation. Future work aims to optimize the execution time further and expand the system's scalability to accommodate more nodes.

Comparison of Edge Computing Implementations: Fog Computing, Cloudlet and Mobile Edge Computing

Topic:
The paper provides a comparative analysis of three Edge Computing (EC) implementations: Fog Computing, Cloudlet Computing, and Mobile Edge Computing (MEC). It aims to identify key features and propose a decision framework to choose the optimal implementation for a given use case.

Problem:
The increase in intelligent and mobile devices, along with the rise of IoT, has created a demand for real-time processing, context-awareness, and support for mobility. Traditional cloud computing solutions face challenges in latency, bandwidth congestion, and environmental impact, making it essential to explore EC implementations to address these issues.

Method:
The paper examines each EC implementation—Fog Computing, MEC, and Cloudlet Computing—in terms of their architectures, node devices, software frameworks, and handling of user requests. It also compares these implementations based on factors like node location, context-awareness, proximity to end devices, access mechanisms, and inter-node communication. A decision tree is proposed to help select the appropriate EC implementation based on application-specific parameters like proximity, access mediums, context-awareness, power consumption, and computation time.

Results:
The comparative study shows that:

- **Fog Computing** offers more flexibility in node placement between end devices and the cloud but has limited computational capacity due to legacy device use.
- **MEC** provides high context-awareness and real-time data handling at mobile network base stations.
- **Cloudlet Computing** serves as a "data center in a box," offering high bandwidth and low latency through Wi-Fi connectivity but with limited context-awareness.

A decision tree framework is introduced, guiding the selection of an EC implementation based on specific use-case requirements such as logical/physical proximity and access needs.

Implications:
The choice of an EC implementation depends on the use case. For applications requiring high context-awareness and low latency (e.g., video streaming), MEC is preferred. Fog Computing is suitable for applications requiring support for a wide range of protocols and low power consumption. Cloudlets are advantageous for high bandwidth and one-hop access scenarios.

Conclusion:
While Fog Computing, MEC, and Cloudlet Computing share the same vision of Edge Computing, they differ in their architecture and features. There is a lack of standardization across these implementations, affecting their comparative evaluation. The proposed decision tree provides a pathway for selecting an optimal EC implementation based on specific application needs. Future work includes standardization and further development of the decision framework to enhance its comprehensiveness.

Amino - A Distributed Runtime for Applications Running Dynamically Across Device, Edge and Cloud

Topic:
The paper introduces "Amino," a distributed runtime system designed for developing and running applications dynamically across devices, edge nodes, and cloud servers. It focuses on enabling efficient code offloading and execution across these layers in dynamic, distributed environments.

Problem:
Most distributed applications are designed to run client components on devices and server components in the cloud, with a fixed design of where these components run. This rigid separation makes it difficult to optimize for real-time processing, availability, and resource constraints. The challenges of edge computing include hardware heterogeneity, network instability, high failure rates, and limited resources on edge devices.

Method:
Amino builds on the Sapphire framework to support distributed applications by introducing "Sapphire Objects" that can run across multiple environments (device, edge, cloud). The framework includes:

- A unified runtime layer across all platforms.
- Support for multiple programming languages using GraalVM.
- Deployment Managers (DMs) to enhance capabilities like caching, code offloading, and replication. The architecture of Amino allows application components to be

moved dynamically between devices, edge nodes, and cloud servers based on policies like reducing latency or conserving energy. The Kernel Server and Object Management Service (OMS) in Amino manage the distributed objects and facilitate cross-platform communication.

Results:

The authors present two applications to validate the Amino framework: a Go Game and License Plate Recognition. In both cases, Amino dynamically offloads processing to either edge or cloud resources based on real-time performance needs, resulting in improved speed and resource utilization. The Go Game application saw a 5x performance improvement when offloaded to the cloud, and License Plate Recognition optimized recognition speed depending on the complexity of the image data.

Implications:

Amino provides a flexible platform for edge-cloud distributed applications, enabling real-time decision-making for task placement across devices, edge nodes, and cloud servers. It simplifies the development process by abstracting away the complexity of distributed systems, allowing developers to focus on business logic while the framework handles dynamic code offloading, replication, and scalability.

Conclusion:

The paper concludes that Amino effectively bridges the gap between device, edge, and cloud environments for distributed application deployment. It supports multi-language development, offers dynamic code offloading, and chains multiple DMs to enhance object capabilities. Future work will focus on enhancing the offloading algorithm and developing additional DMs for optimal object placement in distributed systems.