Enabling Elasticity on the Edge using Heterogeneous Gateways

Nabeel Nasir
nabeeln@virginia.edu

Bradford Campbell
bradjc@virginia.edu

Server Based Edge Computing: the Norm, but Has Its Drawbacks

Cloud
Expensive
Edge Server
Not elastic
Gateways
Requires IoT gateways
Devices
Centralized

Edge Gateways Becoming Increasingly Capable and Specialized

Substantially cheaper
Can match app requirements
Directly interact with IoT devices
Better deployment flexibility

Distribute Applications on Gateways Instead of a Server

Scale out by adding a new gateway
Schedule based on requirements

Middleware on Each Gateway to Enable This Vision

Gateway discovery for scalability
Supports IoT device handling
Encapsulates network topology
Distributed services to handle devices, applications, and data streams
Device interaction API hides underlying network complexity from apps
Provides remote management

Scheduler uses Requirements, Capabilities and Resource Usages

Optimize scheduling to:
- Minimize waiting time of tasks
- Maximize requirement satisfaction
- Minimize network traffic overhead
- Minimize no. of app transfers

Compared to other schedulers:
- Grid / Cluster Computing: Mostly homogeneous machines
- Real-time scheduling: Requires exec times and deadlines
- Function-as-a-Service Platforms: App’s requirements or machine’s capabilities not considered

Evaluation and Implementation of the Platform

Evaluation Plan
- Compare scheduler’s performance to other baselines
- Demonstrate feasibility by porting apps from platforms like cloudlet
  - Eg.: supporting AR, VR on the edge

- Implementation specifics:
  - Task requirements collected in JSON format
  - Execution environment: Node.js on Linux
  - Gateway capabilities: systeminformation npm module
  - Resource usage monitoring: mpstat, free, du, tegrastats etc.