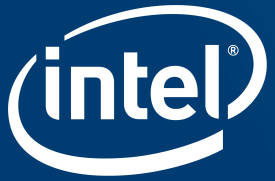


Compute Utilities

Michael McCool

Oct 15, 2020

W3C Joint WoT/Web and Networks Meeting

Problem: Limited Clients, Sensitive Data

Trends:

- Limited client tradeoffs
  - Performance
  - Power/thermals/runtime
- Compute as a utility
  - Cloud computing
  - Edge computing
- Browser as an application platform
  - Progressive web apps (PWAs)

Pain Points:

- Privacy and security
- Programming model
- Performance and power
- Thermal management
- Scalability and Flexibility
- Latency
- Mobility
- Offline functionality

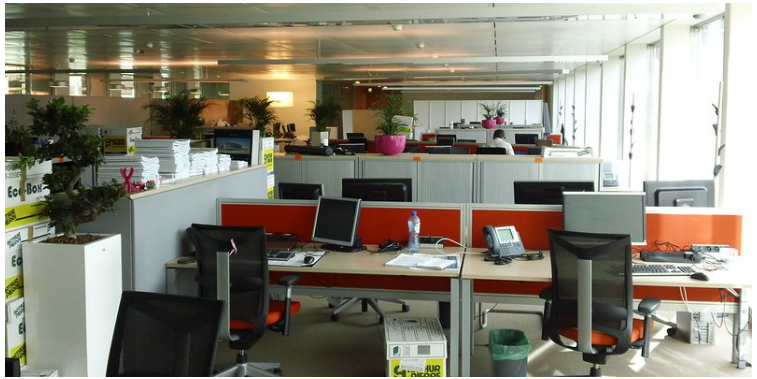
Use Case Domains: Private

Home

- Compute offload supporting work from home and home security
- User wants to orchestrate local devices and AI services
- ***Owner wants to ensure private data stays on-site***

Office

- Individual computers can be low-cost and lightweight
- Workers can offload work to local edge computers and private cloud
- ***Private business, customer, and employee data stays on-site***



Use Case Domains: Public

Retail

- Small business owners wanting to self-manage technology (1)
- Large retail franchises deploying applications for use on employees' own devices (BYOD context)
- **Manage private payment data**

City

- Cities want to develop third-party application ecosystem to best provide value to citizens (2,3)
- Ambient services supporting citizens
- Multivendor, avoid silos
- **Maintain privacy and transparency**

(1) https://www.conexxus.org/
(2) https://www.mckinsey.com/business-functions/digital-mckinsey/our-insights/the-internet-of-things-the-value-of-digitizing-the-physical-world
(3) https://machinaresearch.com/news/smart-cities-could-waste-usd341-billion-by-2025-on-non-standardized-iot-deployments/

Edge Compute/Web Hybrids Discussed in W&N

Edge Worker

- Extend web worker to offload work from browser to edge computer (Intel)
- Extend service worker to execute computations in the CDN (CloudFlare)

Distributed Browser

- Break browser into multiple processes, offload some to another computer

Mobile Worker

- Dynamically migrate running web worker to another computer

→ **All need "another computer" to offload to:**
How to find? How to decide to offload?

Edge Worker

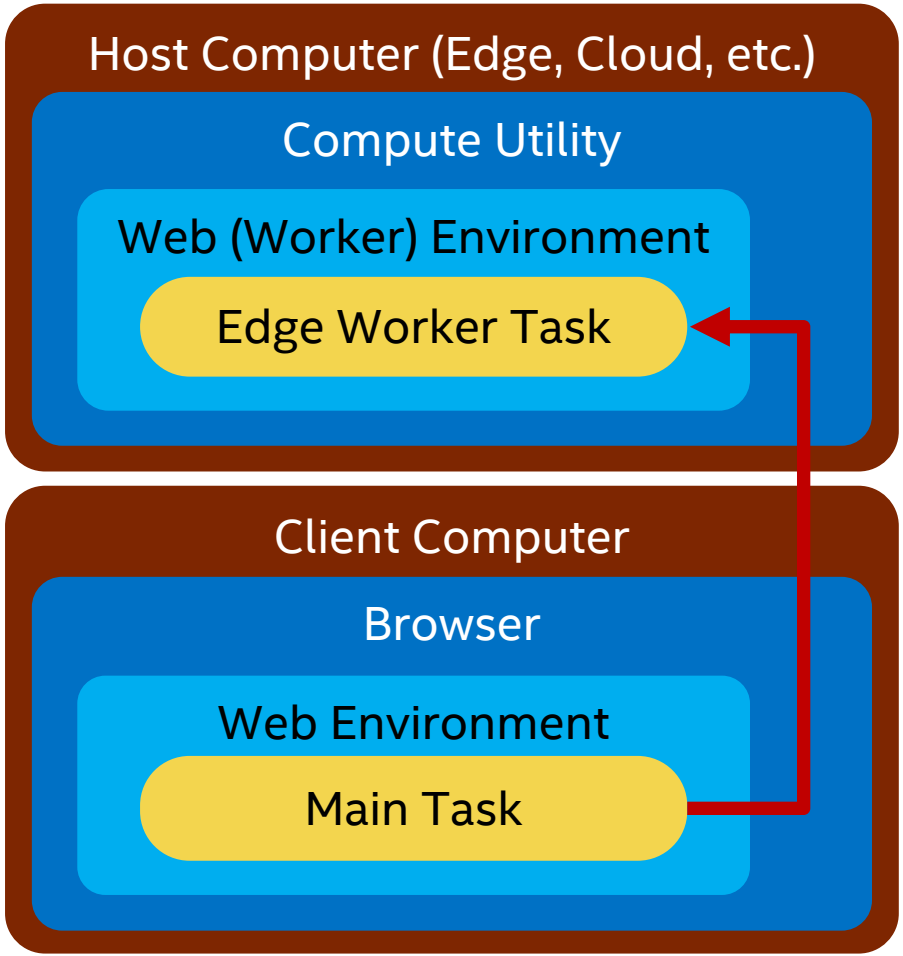
Proposal (W3C Web and Networks IG)

Purpose:

- Run compute-intensive tasks elsewhere on the network, improving performance
- Provide persistent (always-on) response

Implementation:

- Work moved to a "compute utility" hostable on a variety of platforms
- Lifetime tied to installation of persistent web app; event-driven execution
- May respond even when web app closed



Key Issues for Standardization

- **Discovery of “compute utility” or compute services**
  - Both local (local area network) and remote (on the internet)
- **Compute utility network API**
  - To load workload into a compute node and to perform migration
- **Metadata and metrics**
  - Capabilities, performance, latency, etc.
- **Packaging of compute workloads**
  - Options include container images, scripts, and WASM
- ***Browser API (edge workers)***
  - Based on web worker API

Packaging/Runtime Choices

Container Images

- Non-browser runtime, e.g. Docker, Kubernetes, etc.
- Both docker and kata containers allow GPU access
- High performance

WASM Modules

- Needs browser runtime or WASI runtime
- Support for acceleration still in design phase or experimental

Scripts

- Send script to remote lightweight execution environment (w/ webGPU)

Relationship to WoT

- **Discovery of “compute utility” or compute services**
  - Can use WoT Discovery
- **Compute utility network API**
  - Can describe with WoT Thing Description
- **Metadata and metrics**
  - Can be provided via JSON in Thing Description
  - WoT Discovery to support JSONPath/XPath/SPARQL/geo queries
- **Packaging of compute workloads**
  - Scripts can use WoT Scripting API (orchestration use case)

Metadata

- Compute utilities should provide metadata about capabilities, performance, and resources
- Metadata about network is also needed to determine QoS (latency, BW, etc).
- Workloads need to have metadata about their requirements
- The client needs to decide to offload a workload based on this metadata

Issues:

- We may not be able to expose the metadata directly to the client application code (privacy issues; want to avoid fingerprinting). Rather the client should support a (configurable, automated) "decision process".
- Compute utilities and networks may lie about their capabilities. May need a reputational scoring system to identify untrustworthy compute utilities.

Metadata

Workload

Network QoS

- Maximum latency
- Minimum bandwidth
- Minimum network reliability (opt)

Compute QoS (predictive/adaptive)

- Minimum memory size
- Performance/load type/benchmarkl
- (CPU) + Accelerator technology

Compute Utility

- Memory size
- Accelerator technology
- Performance/load type/benchmark
- Variants (compatible, but with variable performance

Network (LPP)

- Latency
- Bandwidth
- Reliability

Offload Decision Rule

1. Feasible options exist
    1. Meet minimum QoS requirements
    2. Meet minimum reliability requirements
2. Satisfies agent settings
    1. Allow remote offload
    2. Extra performance/reliability requirements (e.g. at least 10x speedup/power savings)
3. Improve some metric by some minimum amounts
    1. Performance
    2. Power reduction

Summary

All proposals for edge computing so far need

- A target to offload to with "good" properties relative to client
- Need to decide whether offload is beneficial
- Decision requires metrics on performance, connectivity, etc.

Compute Utility could be defined that

- Is discoverable (via WoT Discovery, for example)
- Has standardized network interface (described by WoT TD, for example)
- Has standardized workload packaging (using scripts and including WebGPU and WoT Scripting API, for example)