Scheduling and Placement for Low-Carbon Edge/Cloud Computing

March 17, 2022
LCSC Seminar Series

Dr. Lauritz Thamsen
UofG – SoCS – GLASS
https://lauritzthamsen.org
Outline

• Background
• Simulation of Energy Consumption
• Temporal Cloud Workload Shifting
• Forecast-Based Admission Control
• Outlook & Summary
Acknowledgement

• Work done with PhD students in Prof. Kao’s Distributed and Operating Systems group at TU Berlin

Philipp Wiesner
Dominik Scheinert
Kordian Gontarska
Ilja Behnke
Thorsten Wittkopp
Background
Data-Intensive Applications

- Search
- Business Intelligence
- Recommendations
- Scientific Data Analysis
- Log Analysis
- Telemedicine
- Mobility
- Industry 4.0
- Smart Water Networks
- Smart Homes
Diverse Computing Infrastructures

Heterogeneous and dynamic distributed computing environments from devices to data centers

Lower Latencies

More Resources
Research Questions

Given a job and objectives/constraints for its execution:

1. What resources to use for a job?

2. When and where to run the jobs?

3. How to set system configurations?
Sustainability Objectives

• Low energy consumption: Caching, local computing, utilizing resources fully (before scaling out more), energy-efficient languages and systems, …

• Low carbon emissions:
  • Grid energy carbon emissions are often different over time and locations → use low-carbon energy
  • On-site renewable energy sources → use all the green energy

• First requirement? Understanding energy demands…
LEAF: Energy Simulation

The “LEAF” Simulator

• Simulator for modeling the energy consumption of applications in cloud/fog/edge computing environments

• Design goals:
  • holistic but granular power modeling
  • realistic and dynamic compute environments
  • energy-aware online decision making
  • thousands of devices & applications

https://github.com/dos-group/leaf
Infrastructure Model

WAN (4G LTE access network)
- Latency: 80 ms
- Bandwidth Up: 50 Mbit/s
- Bandwidth Down: 100 Mbit/s
- Power Usage Up: 6 658 nJ/bit
- Power Usage Down: 20 572 nJ/bit
Power Modeling

Power (Watt)

\[ P_{\text{max}} \]

\[ P_{\text{static}} \]

Load (e.g. CPU load, bit/s, ...)

\[ C_{\text{idle}} \]

\[ C_{\text{max}} \]

\[ \sigma \]
Application Model

<table>
<thead>
<tr>
<th>WAN (4G LTE access network)</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Latency</td>
<td>80 ms</td>
</tr>
<tr>
<td>Bandwidth Up</td>
<td>50 Mbit/s</td>
</tr>
<tr>
<td>Bandwidth Down</td>
<td>100 Mbit/s</td>
</tr>
<tr>
<td>Power Usage Up</td>
<td>6 658 nJ/bit</td>
</tr>
<tr>
<td>Power Usage Down</td>
<td>20 572 nJ/bit</td>
</tr>
</tbody>
</table>

- **sink task**
  - (4,000 MIPS, 2GB RAM)

- **processing task**
  - (12,000 MIPS, 512MB RAM)

- **source task**
  - (3,000 MIPS, 128MB RAM)

Diagram showing network connections and task details.
Application Power Usage

WAN (4G LTE access network)
- Latency: 80 ms
- Bandwidth Up: 50 Mbit/s
- Bandwidth Down: 100 Mbit/s
- Power Usage Up: 6 658 nJ/bit
- Power Usage Down: 20 572 nJ/bit

sink task
(4,000 MIPS, 2GB RAM)
- 2.8 W

processing task
(12,000 MIPS, 512MB RAM)
- 7.6 W

source task
(3,000 MIPS, 128MB RAM)
- 2.4 W

Wi-Fi

Bluetooth

40 kbit/s
- 0.27 W

170 kbit/s
- 0.05 W
Event-Based Simulation

- Events read and update the infrastructure and application models
Experiment Setup

- 24 hours of taxi traffic
- time steps of one second

|                  | Max load | $P_{static}$ | $\sigma$  
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Cloud Fog node</td>
<td>$\infty$</td>
<td>100 W</td>
<td>350 $\mu$W/MIPS</td>
</tr>
<tr>
<td>400 000 MIPS</td>
<td>100 W</td>
<td></td>
<td></td>
</tr>
<tr>
<td>WAN STL $\rightarrow$ Cloud</td>
<td>50 Mbit/s</td>
<td></td>
<td>6658 nJ/bit</td>
</tr>
<tr>
<td>WAN Cloud $\rightarrow$ STL</td>
<td>100 Mbit/s</td>
<td></td>
<td>20 572 nJ/bit</td>
</tr>
<tr>
<td>Wi-Fi Taxi $\rightarrow$ STL</td>
<td>1.3 Gbit/s</td>
<td></td>
<td>300 nJ/bit</td>
</tr>
<tr>
<td>Wi-Fi STL $\rightarrow$ STL</td>
<td>1.3 Gbit/s</td>
<td></td>
<td>100 nJ/bit</td>
</tr>
</tbody>
</table>

Two types of applications:
- CCTV: One for each traffic light
- V2I: One for each taxi
Experiment Results

![Graph showing experiment results]

- **Cloud only**: 62.23 kWh
- **Fog 1**: 43.44 kWh
- **Fog 2**: 33.02 kWh
- **Fog 3**: 27.60 kWh
- **Fog 4**: 23.83 kWh
- **Fog 5**: 24.09 kWh
- **Fog 6**: 26.34 kWh
- **Fog 6s**: 22.09 kWh

**Key**:
- **WAN**
- **Wi-Fi**
- **Cloud**
- **Fog dynamic**
- **Fog static**
Results Over Time

Infrastructure

Power Usage (Watt)

Time

Applications

Power Usage (Watt)

Time

cloud only

4 fog nodes

6 fog nodes with energy saving mechanism

- WAN
- Wi-Fi
- Cloud
- Fog
- Fog static

- CCTV
- V2I
Cloud Workload Shifting

Motivation

• Data centers are responsible for >1% of global energy consumption, with best case projections for 3% in 2030 [1, 2]

• Low-carbon objective: Compute when and where low-carbon energy is available
Changing Carbon Intensity

• What are the most promising times to shift work to?

• Carbon intensity: Amount of CO2 equivalent greenhouse gases emitted per kilowatt hour of energy
Shiftable Cloud Workloads

- Ad hoc vs. scheduled
- Temporal constraints
- Job runtimes
- Interruptibility
Experimental Evaluation

• Evaluation of two scenarios using the LEAF simulator:

  • Scenario 1 – Periodic Jobs: Nightly builds, integration tests, database backups, generation of business reports, …

  • Scenario 2 – Ad Hoc Jobs: ML training jobs, CI/CD, data analysis pipelines, scientific simulations, …
Scenario 1: Periodic Jobs

- Baseline: All jobs scheduled at 1 am in the night
- Increasing the window by +/- 1h to allow scheduling between
  - 00:00 to 3:00 (+/- 1h)
  - 23:00 to 4:00 (+/- 2h)
  - ...
  - 17:00 to 9:00 (+/- 8h)
Scenario 2: Large Ad Hoc Jobs

• Setting: Jobs arrive randomly during working hours (Mo - Fr, 9:00h - 17:00h)

• Baseline: Instant scheduling

• Investigate influence of
  • Deadlines
  • Interruptibility
Scenario 2: Overall Results

The bar chart shows the percentage of emissions saved in Germany, California, Great Britain, and France for Next Workday and Semi-Weekly scenarios. The chart compares non-interrupting and interrupting emissions.

- **Next Workday**
  - Non-Interrupting
  - Interrupting

- **Semi-Weekly**
  - Non-Interrupting
  - Interrupting
Admission Control

Problem Setting

• Assumptions:
  • Access to renewable energy sources, with not all the energy being used always, yet also no energy storage
  • Resource constrained compute nodes running a high-priority, time-critical base load, but over time also spare resources

• Goal: Utilize excess energy by computing low-priority, delay-tolerant workloads
Renewable Excess Power

- In some settings local demand does temporarily not cover all produced power, yet is also not put into storage or grids.
The “Cucumber” Concept

Resources
- renewable energy source
- other power consumers
- compute node

Forecasting
- power production forecast
- power consumption forecast
- resource constraints
- load forecast
- excess power forecast
- green excess capacity forecast
- updated periodically
- $\alpha = 0.5$

Admission Control
- new request
- workloads queue
- if workload would get accepted: can we still meet all deadlines using only green excess capacity?
- reject no
- yes

March 2022 LCSC Seminar Series – Low-Carbon Edge/Cloud Computing 33
Admission Control

- For each incoming job, Cucumber checks if it can be computed using excess energy only
- Through probabilistic forecasts, admission can be tuned towards
  - conservative (low acceptance rate, low grid power usage) or
  - optimistic results (vice versa)
Evaluation Results

<table>
<thead>
<tr>
<th>ML Training</th>
<th>Berlin</th>
<th>Mexico City</th>
<th>Cape Town</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline #1</td>
<td>100% (1.8% green)</td>
<td>100% (27.1% green)</td>
<td>100% (32.4% green)</td>
</tr>
<tr>
<td>Baseline #2</td>
<td>1.8% (100% green)</td>
<td>34.0% (100% green)</td>
<td>41.0% (100% green)</td>
</tr>
<tr>
<td>Conservative</td>
<td>1.9% (67.7% green)</td>
<td>29.9% (99.5% green)</td>
<td>40.3% (100% green)</td>
</tr>
<tr>
<td>Expected</td>
<td>5.4% (34.4% green)</td>
<td>36.9% (93.6% green)</td>
<td>42.1% (97.4% green)</td>
</tr>
<tr>
<td>Optimistic</td>
<td>18.6% (10.1% green)</td>
<td>39.8% (87.2% green)</td>
<td>45.0% (91.5% green)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Edge Computing</th>
<th>Berlin</th>
<th>Mexico City</th>
<th>Cape Town</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline #1</td>
<td>99.5% (0.5% green)</td>
<td>99.5% (24.0% green)</td>
<td>99.5% (29.4% green)</td>
</tr>
<tr>
<td>Baseline #2</td>
<td>0.3% (100% green)</td>
<td>26.5% (100% green)</td>
<td>32.4% (100% green)</td>
</tr>
<tr>
<td>Conservative</td>
<td>0.1% (100% green)</td>
<td>17.1% (100% green)</td>
<td>29.2% (100% green)</td>
</tr>
<tr>
<td>Expected</td>
<td>1.1% (34.2% green)</td>
<td>27.4% (97.2% green)</td>
<td>32.5% (99.4% green)</td>
</tr>
<tr>
<td>Optimistic</td>
<td>8.6% (5.4% green)</td>
<td>32.7% (81.8% green)</td>
<td>35.2% (92.1% green)</td>
</tr>
</tbody>
</table>
Outlook & Summary
Ideas for Next Steps

"Truer-to-life" experiments:
• More realistic and interesting simulations?
• Experiments with real hardware and systems?

Decentralized resource management:
• What happens beyond one node / datacenter?
• How should nodes negotiate workloads?

Continuous feedback for developers:
• How can CI/CD tools also report the energy consumption and emissions of applications?
Summary

• Taking carbon emission into account when managing edge/cloud computing workloads could reduce emissions

• Simulation, forecasting, and optimization methods will be valuable tools

• Not as clear how to get this into real services, systems, and developer tools though

• Interesting research into distributed systems, resource management, and also software engineering ahead
References


