Design and Evaluation of a Benchmarking Service for AWS EC2 Hardware Metrics

Michał Jakub Szczepaniak

Technical University of Munich

TUM School of Computation, Information and Technology

Chair of Decentralized Information Systems

Munich, 19. November 2024

Motivation

	c7i.large	c7g.large	c7a.large	c6a.large
vCPUs	2	2	2	2
Architecture	x86	ARM	x86	x86
Manufacturer	Intel	AWS Graviton	AMD	AMD
Hyperthreading	Yes	No	No	Yes
Memory	4 GiB	4 GiB	4 GiB	4 GiB
Network	Up to 12.5 Gigabit	Up to 12.5 Gigabit	Up to 12.5 Gigabit	Up to 12.5 Gigabit

Motivation

- Cloud providers do not share information transparently
  - E.g. Network Bandwidth, vCPUs

- Benchmarking is required to discover true hardware characteristics
  - Benchmarking is difficult, expensive, and time-consuming

- The need for new service that characterizes:
  - Extensibility
  - Transparency

Motivation

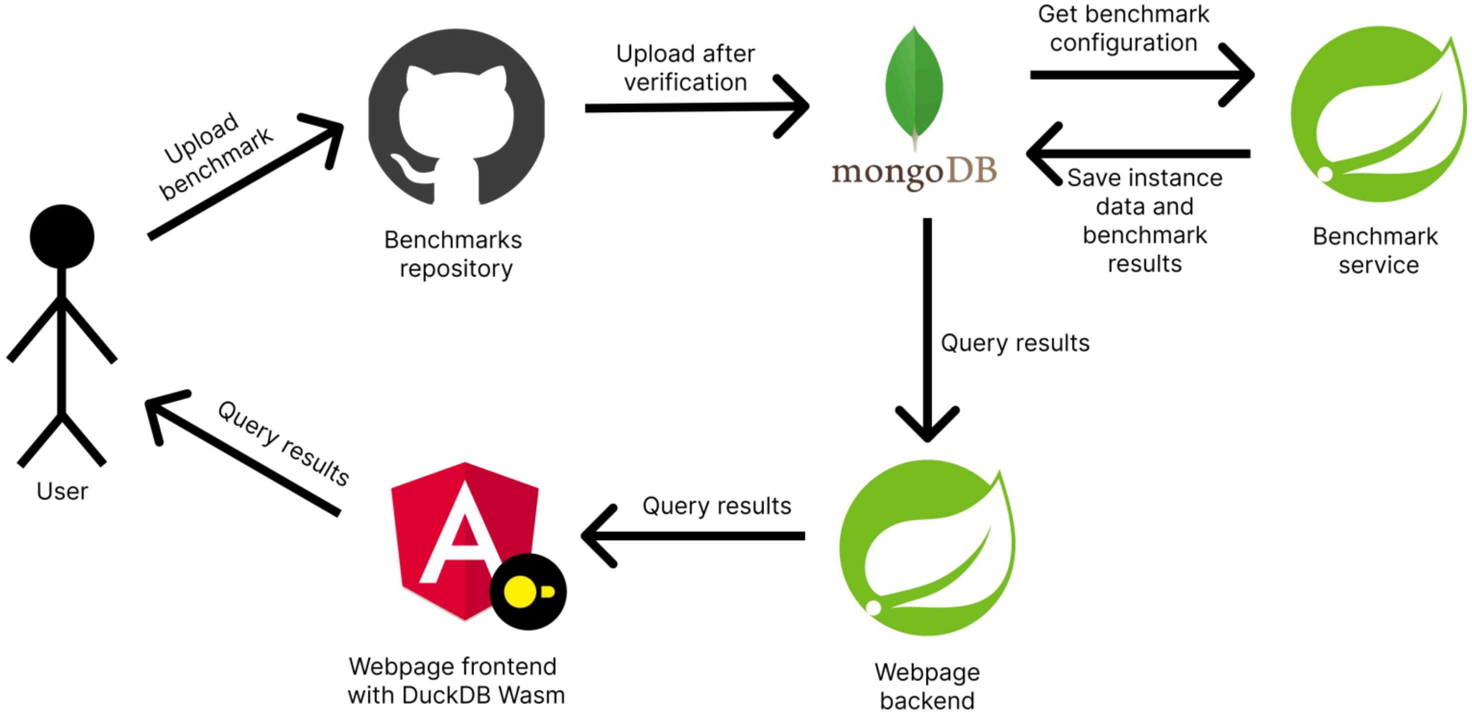
- Proposed solution: TUManyBenchmarks
  - Extensible
  - benchmarking service
  - for comparing EC2 instances
  - based on hardware metrics

Requirements

TUManyBenchmarks should:

- ➢ Allows the users to contribute benchmarks
- ➢ Automatically execute benchmarks
- ➢ Allows the users to access benchmark results

Architecture design


Upload benchmark
Benchmarks repository
Upload after verification
mongoDB
Get benchmark configuration
Save instance data and benchmark results
Benchmark service
Query results
User
Query results
Webpage frontend with DuckDB Wasm
Query results
Webpage backend

Benchmark upload

- The user uploads the benchmark to the GitHub repository as Pull Request
- It requires `configuration.yml`, which consists of three sections:
  - Configuration
  - Nodes
  - Plots
- Analysis of `configuration.yml` done by GitHub Actions and maintainer
- After approve, the configuration is uploaded to MongoDB by GitHub Actions

Benchmark execution

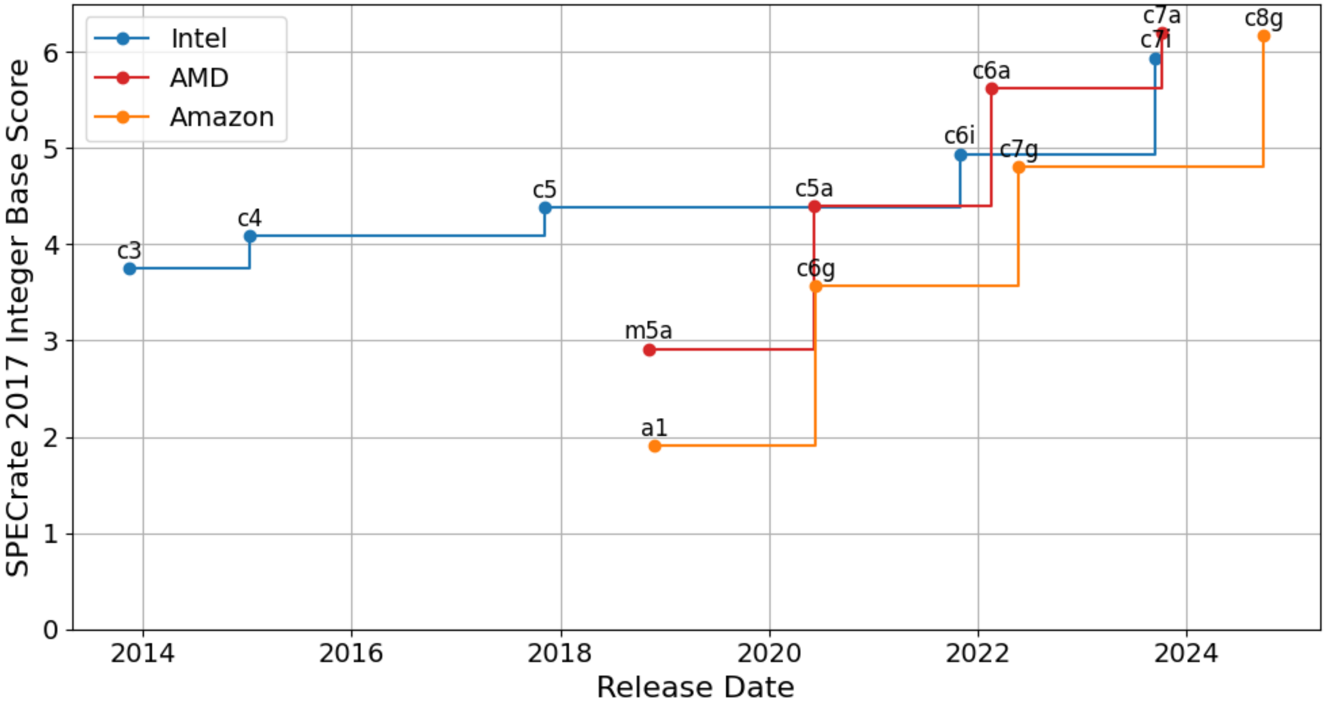
Benchmark service:

- ➢ Recognize benchmarks to execute
- ➢ Recognize instances on which the benchmarks should run
- ➢ Set up infrastructure using AWS SDK
- ➢ Use SSH to communicate with benchmarks
- ➢ Optional: Prepare nodes using Ansible
- ➢ Execute benchmark
- ➢ Extract and save results as JSON

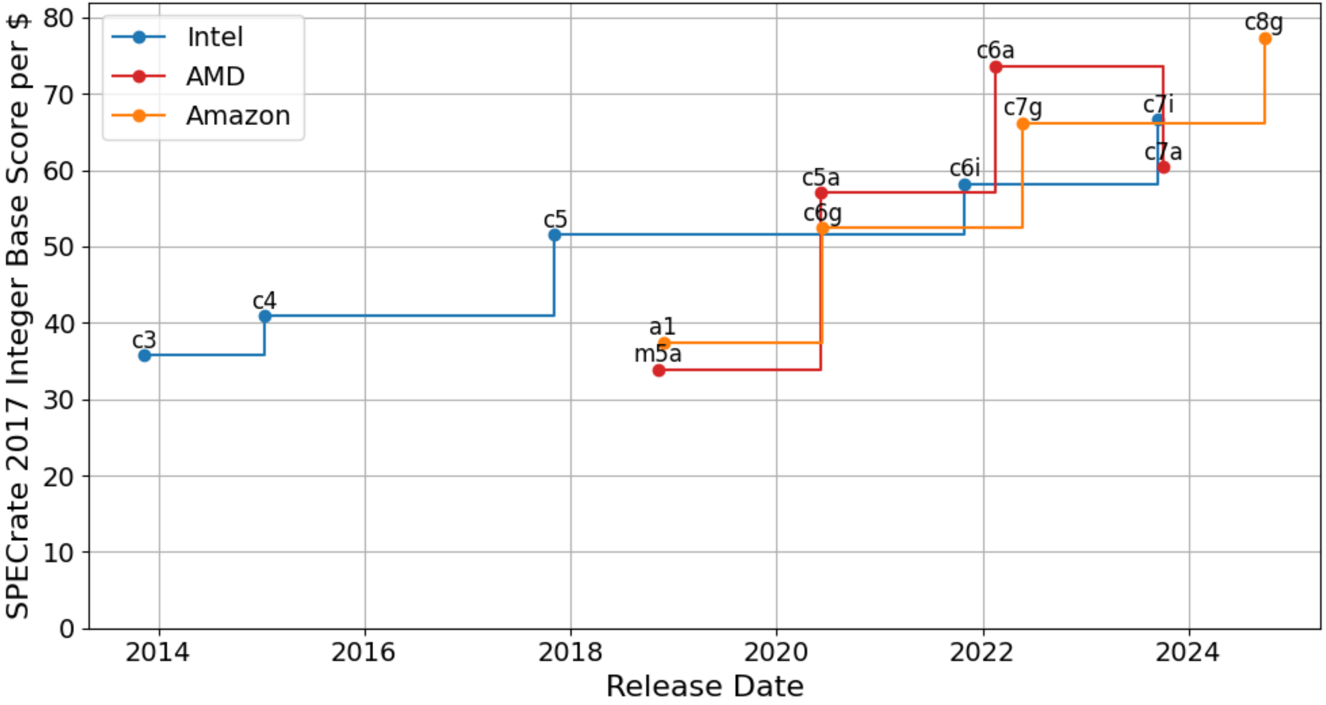
Results presentation

- Web application presents properties and benchmark results of EC2 instances
- The list view allows the users access the data using:
  - Set of simple filters
  - Query console with DuckDB Wasm
- The details view plots benchmark results to simplify analysis
- Webpage allows to compare benchmark results for different instances

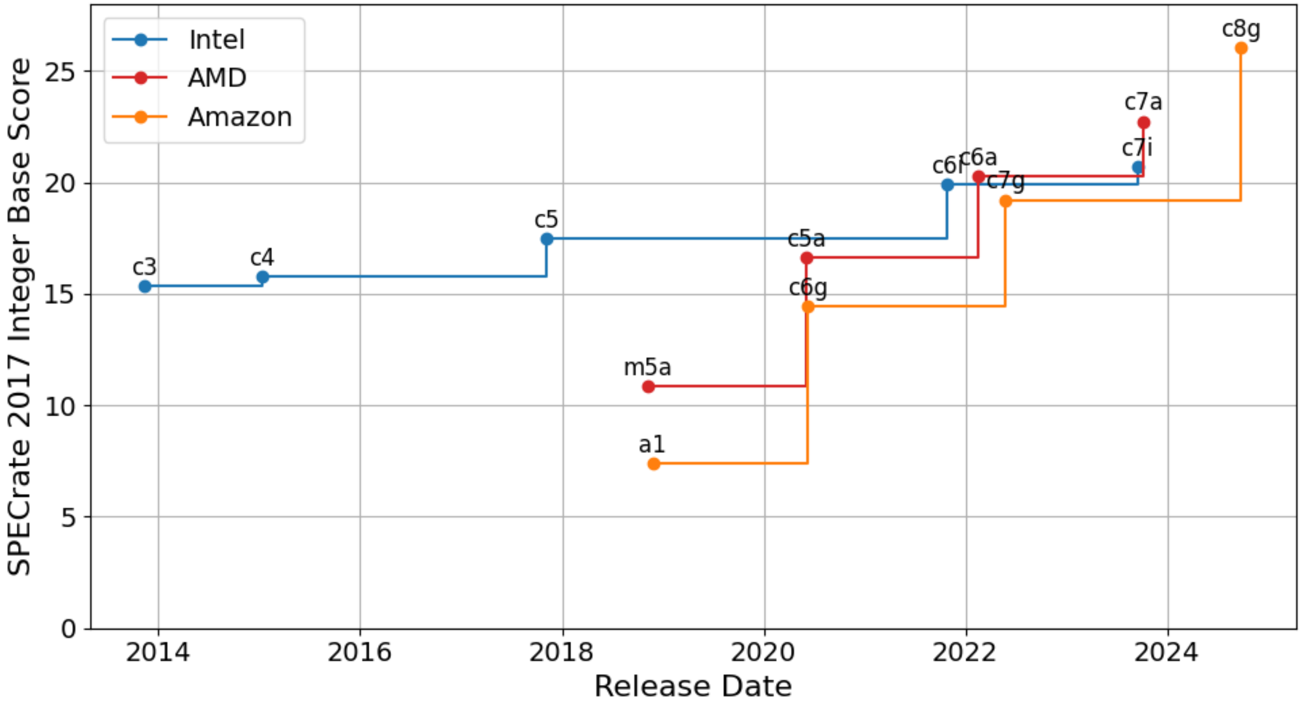
SPEC CPU – Single-core processing



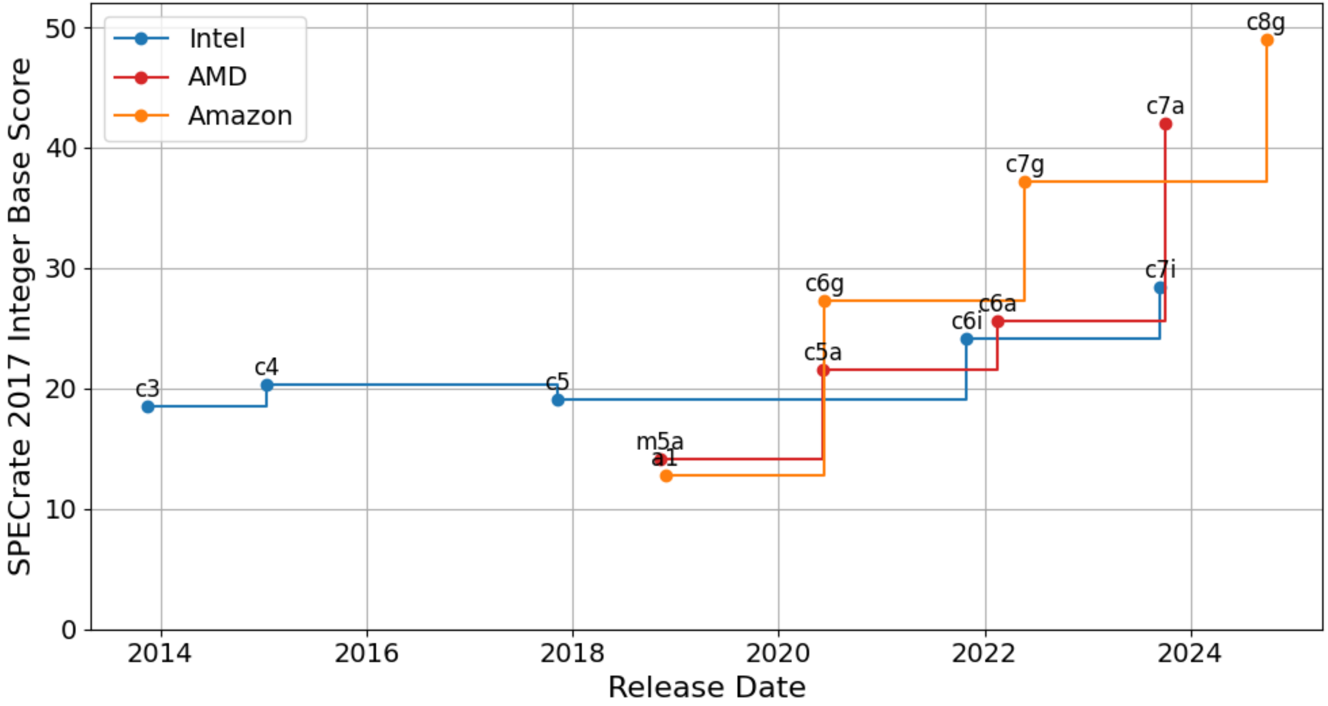
SPEC CPU – Single-core processing cost efficiency



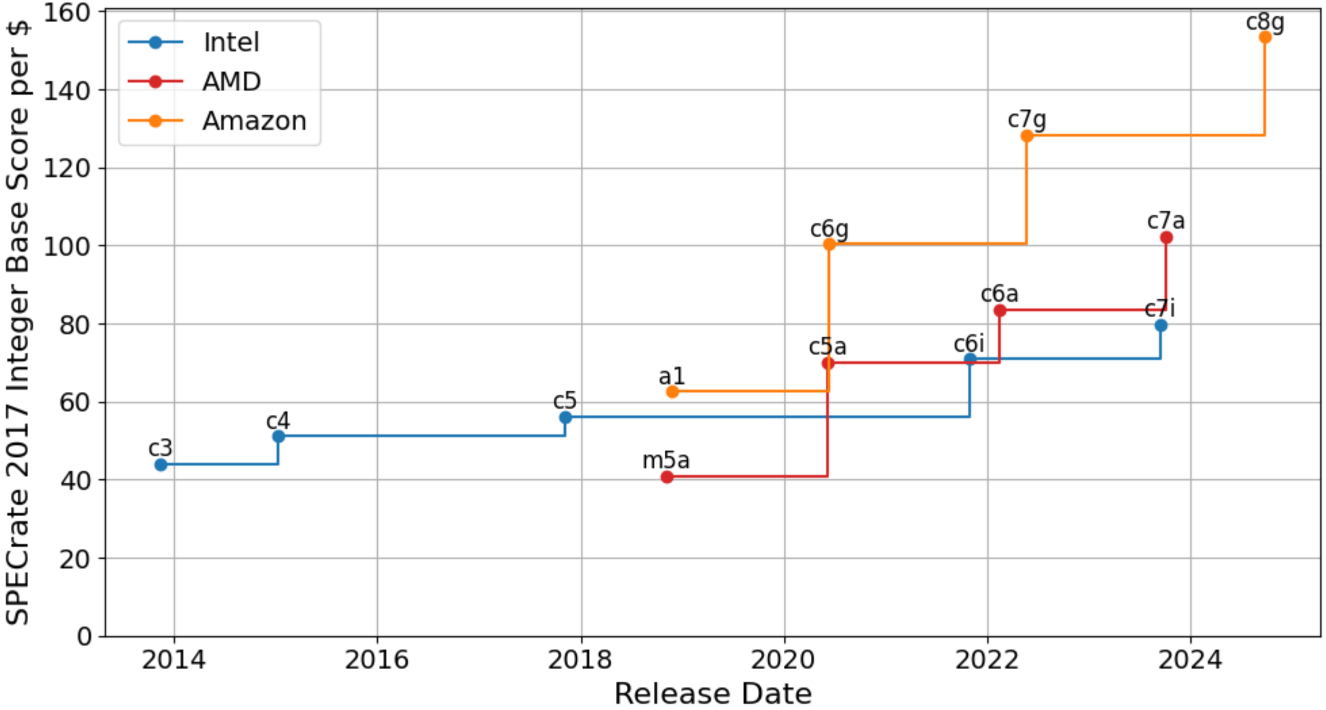
SPEC CPU – Multi-core processing (physical cores)



SPEC CPU – Multi-core processing (all cores)

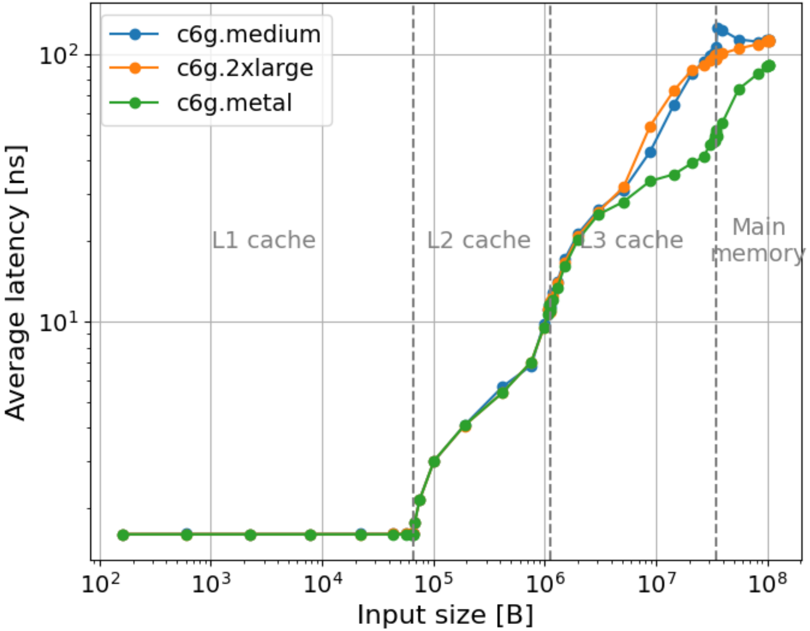


SPEC CPU – Multi-core processing cost efficiency

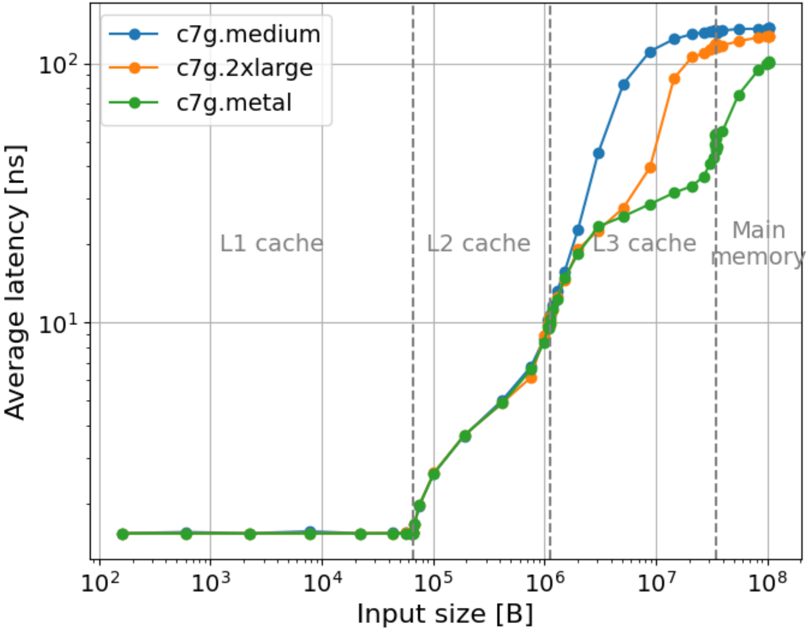


Cache Latency and CAT

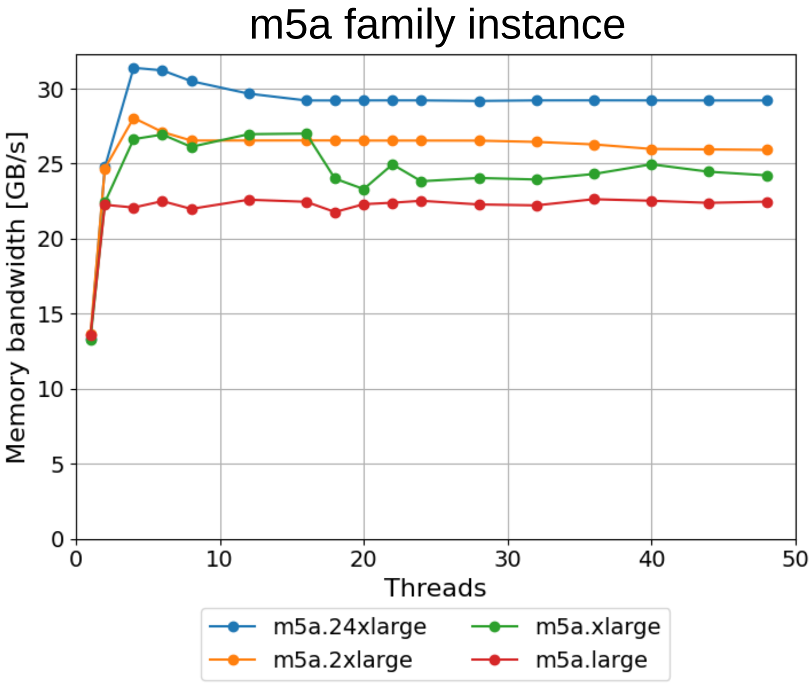
c6g family instances

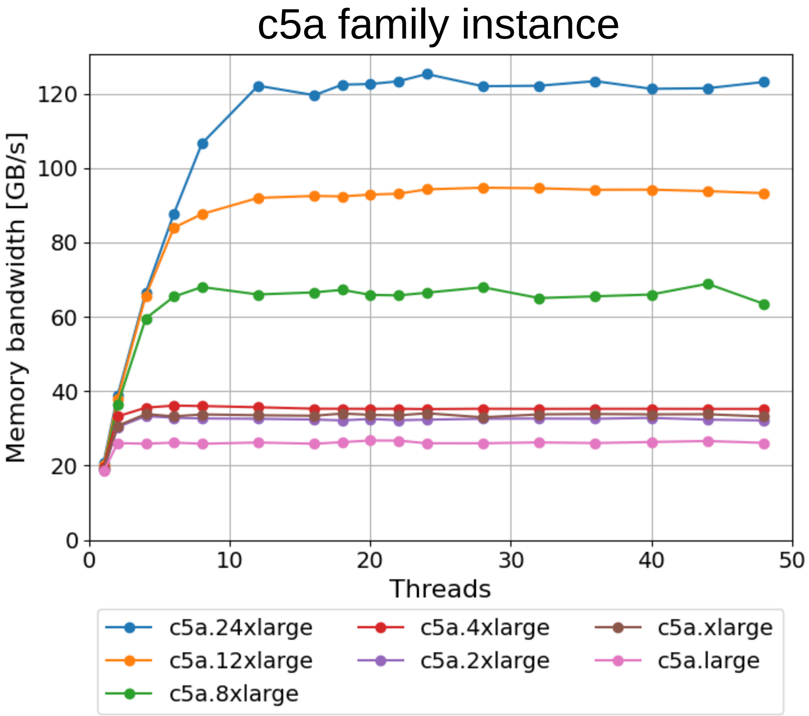


c7g family instances



Memory bandwidth and MBA





Comparison to other services

	TUMany-Benchmarks	Instance comparison service	Benchmark service	Automatic selection tools
Information about instance properties	+	+	+	+
Information about hardware metrics	+	-	+/-	+
Instance comparison	+	+	+	+
Instance comparison based on hardware metric	+	-	+/-	+
Instance comparison using SQL queries	+	-	-	-
Work with any use case	+	+	-	-
Allows for adding new hardware metrics	+	-	-	-

Live Demo

Summary

- Cloud providers do not provide transparent information about offered instances
- TUManyBenchmarks supports users in discovering real hardware metrics by automatic benchmark execution and simplifying data analysis
- TUManyBenchmark allows users, for example, to analyze the performance and cost efficiency of vCPUs over time or resource isolation
- TUManyBenchmark combines functionalities of instance comparison services, benchmark services, and automatic selection tools to provide unique user experience

Design and Evaluation of a Benchmarking Service for AWS EC2 Hardware Metrics

Michał Jakub Szczepaniak

Technical University of Munich

TUM School of Computation, Information and Technology

Chair of Decentralized Information Systems

Munich, 19. November 2024