ER IPT HACK_2023
Track 2: Calculations

YAV-SSV Team Approach

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
1898

Фізико-технічний інститут
КПІ ім. Ігоря Сікорського

ER IPT HACK

12-21
MAY
2023

SWIFTT

Infosoft

Task

HEP-Benchmarks
Group ID: 19914

- The focus is on resource-intensive tasks performed by CERN for processing Large Hadron Collider experiment results.
- Investigate measurement the performance of these computational tasks.
- Investigate measurement of energy efficiency of these computational tasks

Overview -
Energy Consumption Measurement

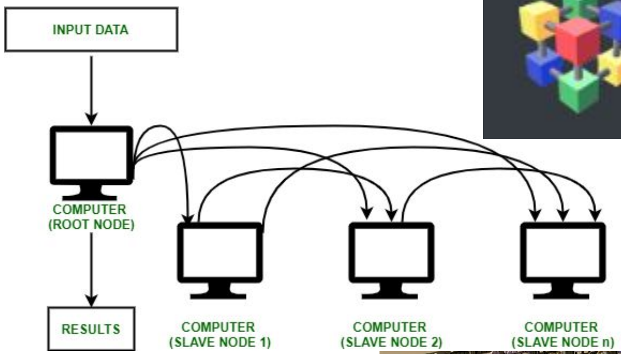
CPU consumes energy

OS reads sensors

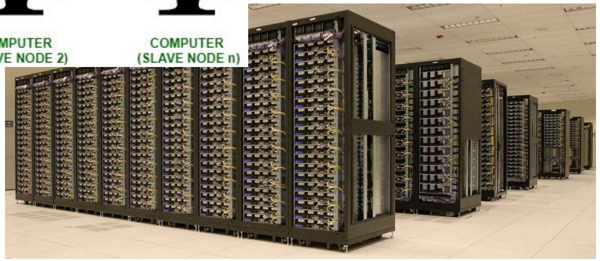
```
Intel RAPL
Powerstat
PowerTOP
perf
Likwid
```

Software provides access to the measurements

Overview - **WLCG**



WLCG
Worldwide LHC Computing Grid



- global collaboration of around 170 computing centres
- more than 40 countries
- distribute and analyse the ~200 Petabytes of data from Large Hadron Collider every year
- More: wlcg.web.cern.ch

Overview - **WLCG Workload**

HEP Benchmark Suite
gitlab.cern.ch/hep-benchmarks/hep-benchmark-suite

Benchmarking

- Run workloads representative of the production applications running on WLCG
- Enable performance studies on heterogeneous hardware

Production

- Madgraph5_aMC@NLO
- Madgraph4gpu
- github.com/mg5amcnlo
- github.com/madgraph5

Hardware

HEPScore Benchmark

- Replacement of proprietary HS06
- Consists of several CPU-intensive workloads
- Executes workload in containers (Singularity/Docker)
- Each workload score is normalised to the score of the reference server, in order to make it a dimensionless factor

More info:

- [Source code repository]
- [HEPScore the HEP-specific benchmark]


```
hepscore_benchmark:
  benchmarks:
    cms-reco-bmk:
      results_file: cms-reco_summary.json
      ref_scores:
        reco: 2.196
      weight: 1.0
      version: v2.1
      args:
        threads: 4
        events: 50
    lhcb-gen-sim-bmk:
      results_file: lhcb-gen-sim_summary.json
      ref_scores:
        gen-sim: 90.29
      weight: 1.0
      version: v2.1
      args:
        threads: 1
        events: 5
  settings:
    name: TestBenchmark
    reference_machine: "CPU Intel(R) Xeon(R) CPU E5-2630 v3 @ 2.40GHz"
    registry: docker://gitlab-registry.cern.ch/hep-benchmarks/hep-workloads
    method: geometric_mean
    repetitions: 3
    scaling: 355
    container_exec: singularity
```

Challenges

- **Slow** computation

alice-digi-reco-core-run3-bmk You can view the YAML HEPscore configur

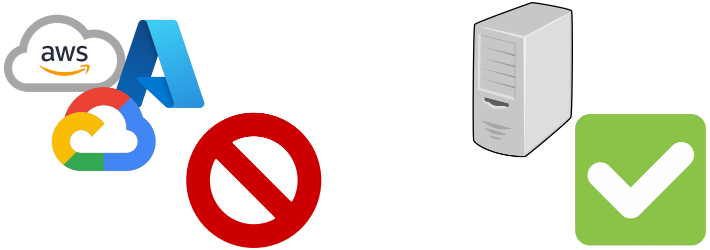
The benchmark will take 5+ hours to execute on modern hardware.

NOTE: ~20 GB of free disk space in your Singularity or Docker cache are

- **Unsterility** of the environment

- **Hardware** dependence - inability to parallelize tests, not obvious comparison between CPU and GPU

- Limited use of **VM** due to lack of access to the physical CPU



Experiment workflow

Setup:

- Stick to a single benchmark task (belle2-gen-sim-reco-bmk)
- Run locally (AMD Ryzen 5 5500U, 6 cores, 12 threads)
- Linux, perf (power/energy-pkg), cpupower, docker

Workflow:

1. Limit upper CPU frequency
2. Run `perf` and benchmark workload
3. Save results

Repeat few times

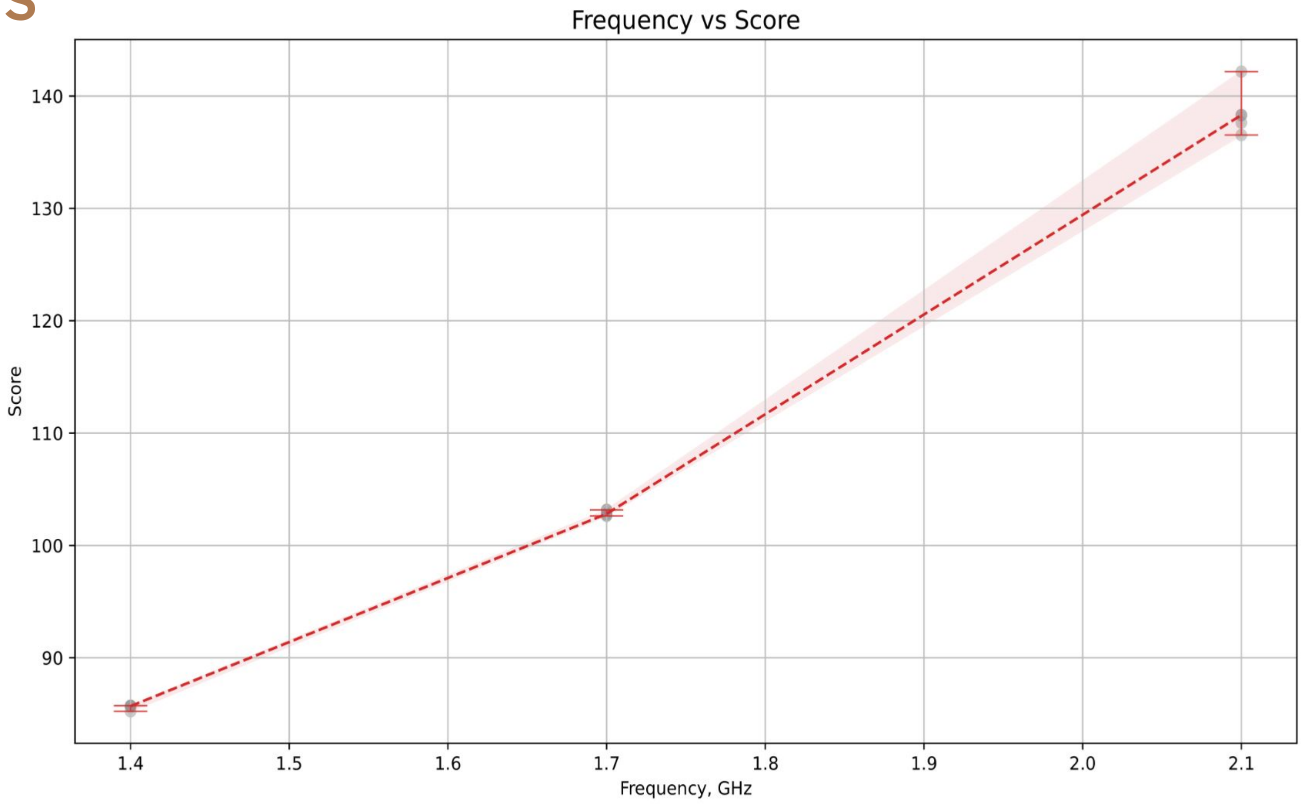
For each available cpu frequency
(1.4 GHz, 1.7 GHz, 2.1 GHz)

Results

5 runs for each frequency

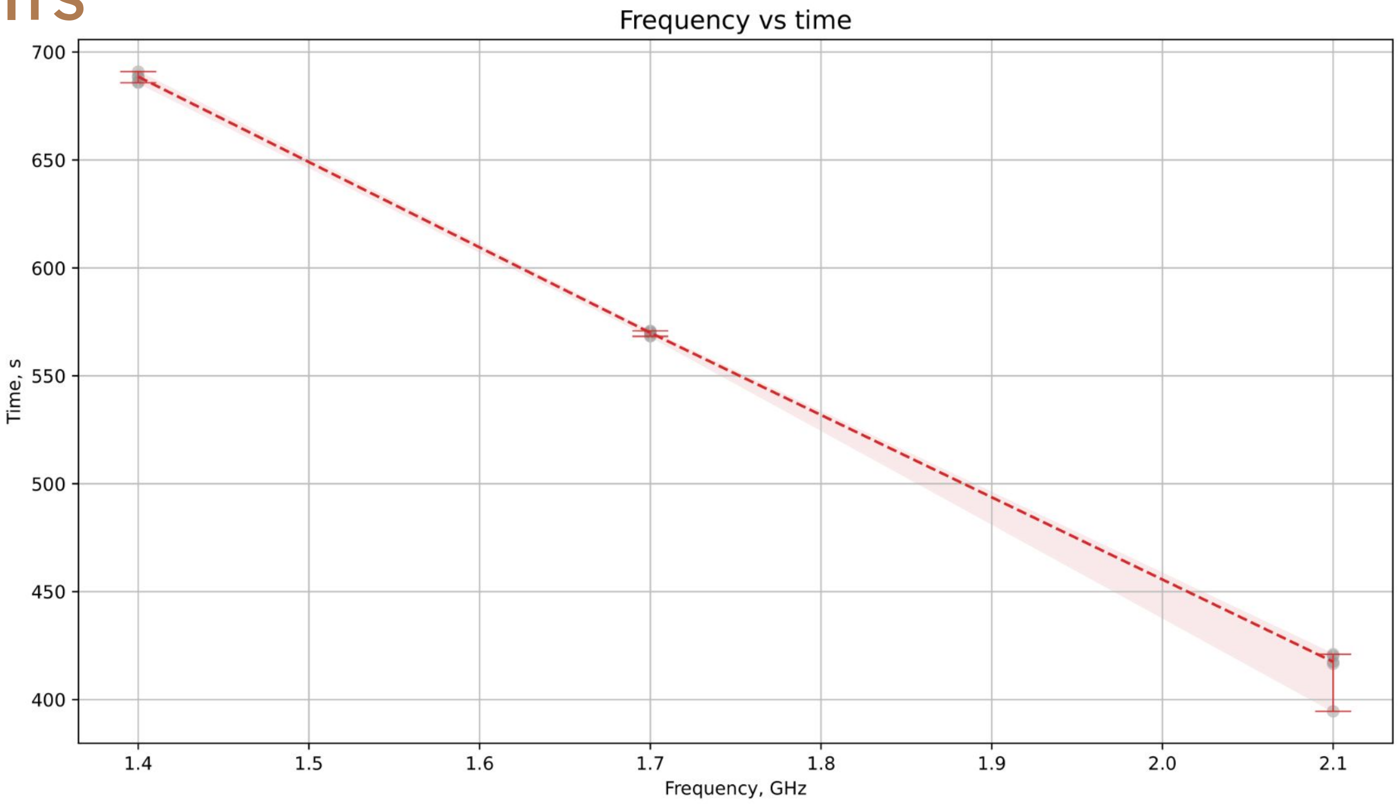
	1.4 GHz	1.7 GHz	2.1 GHz
Power consumption median, Joules	5927.65	5468.11	6267.85
Power consumption max, Joules	6023.65	5492.29	6960.44
Power consumption min, Joules	5907.67	5427.24	6210.68
Benchmark score median	85.697	102.808	138.308
Benchmark score max	85.7325	103.163	142.1775
Benchmark score min	85.2355	102.6305	136.533
Execution time median, s	688.5928424	569.9492953	417.5034929
Execution time max, s	690.952235	570.8386744	420.9916446
Execution time min, s	685.7868938	568.2845271	394.5543055
Score per joule median	0.014463152	0.018856345	0.021958646
Score per joule max	0.014506057	0.018910256	0.022269381
Score per joule min	0.014150142	0.018740637	0.02042651

Results



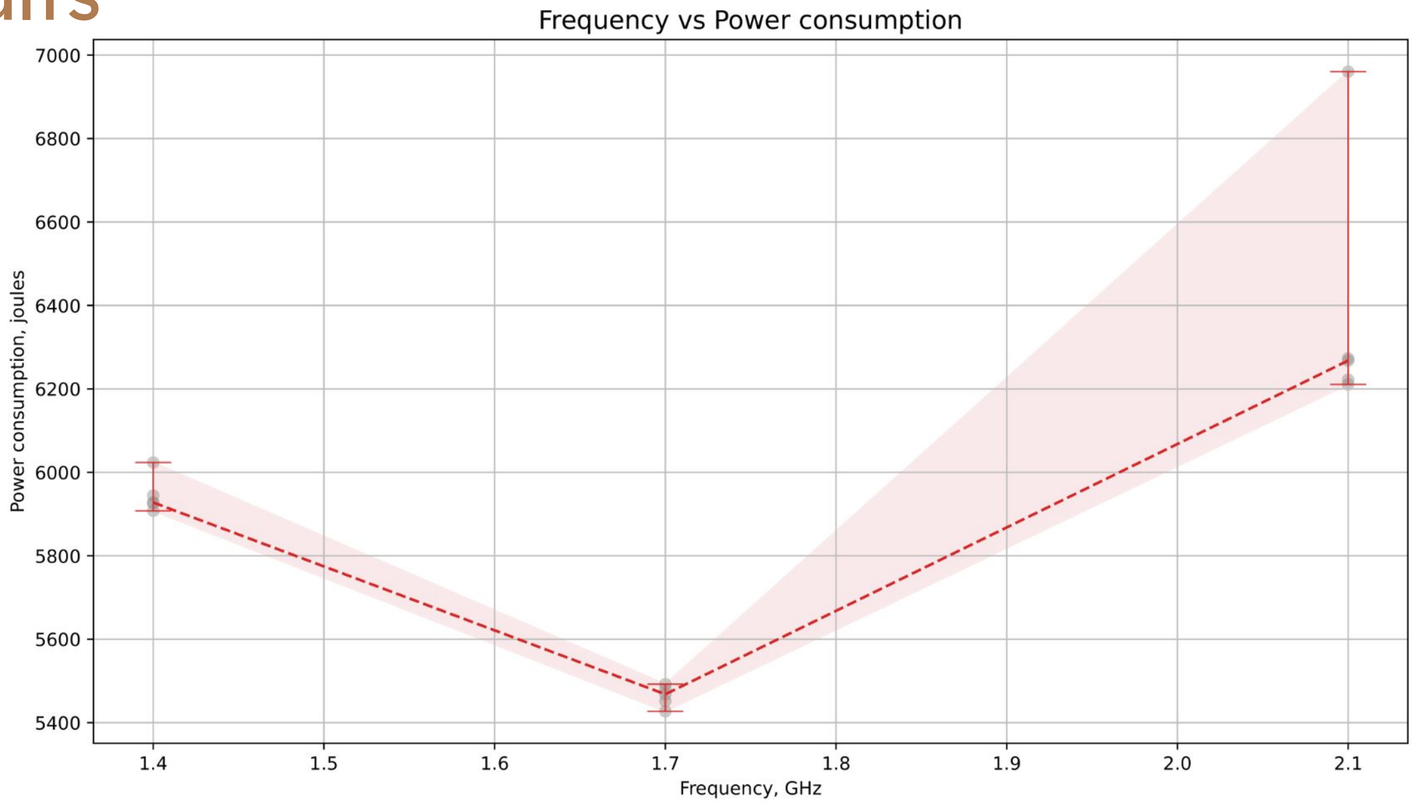
Frequency vs Score
Score
Frequency, GHz
140
130
120
110
100
90
1.4
1.5
1.6
1.7
1.8
1.9
2.0
2.1

Results

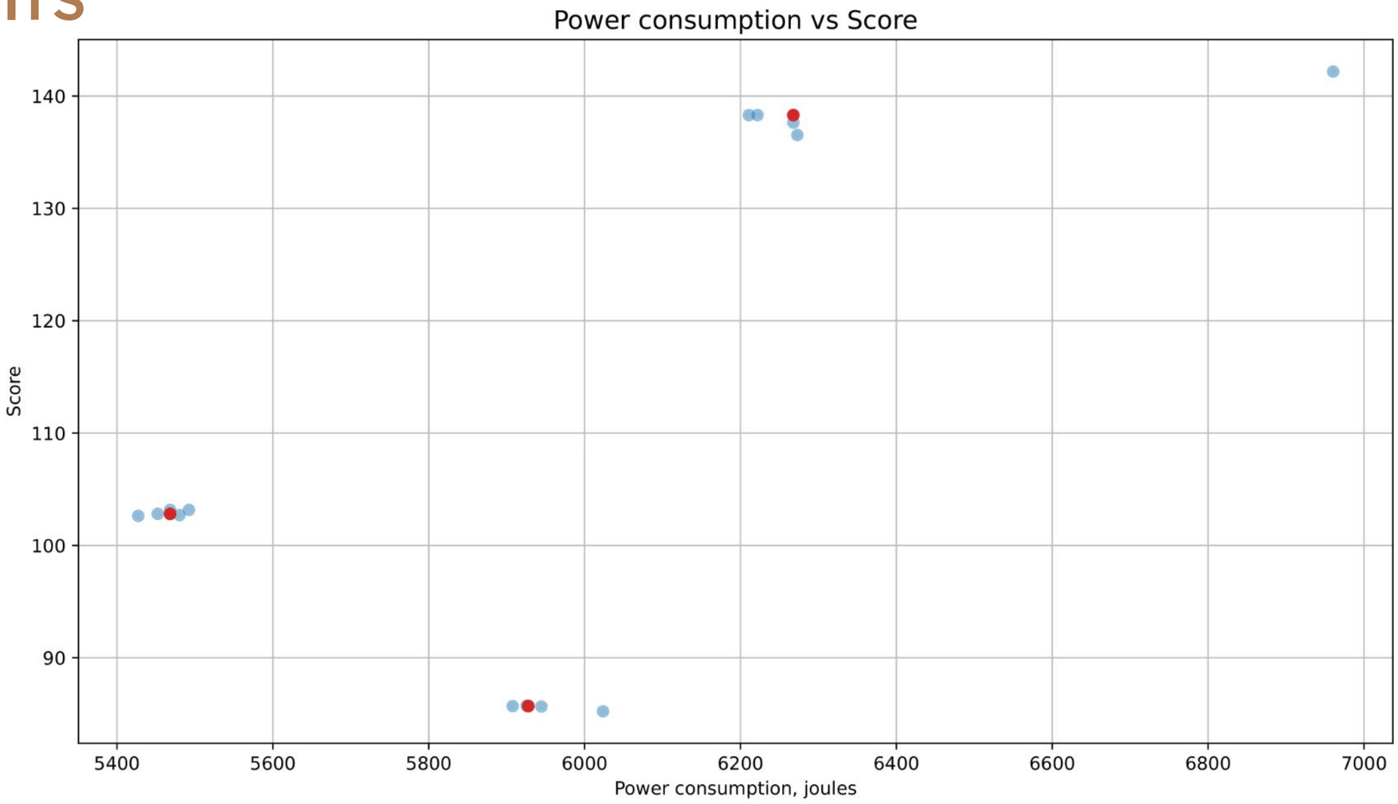


Frequency vs time
Time, s
Frequency, GHz
700
650
600
550
500
450
400
1.4
1.5
1.6
1.7
1.8
1.9
2.0
2.1

Results

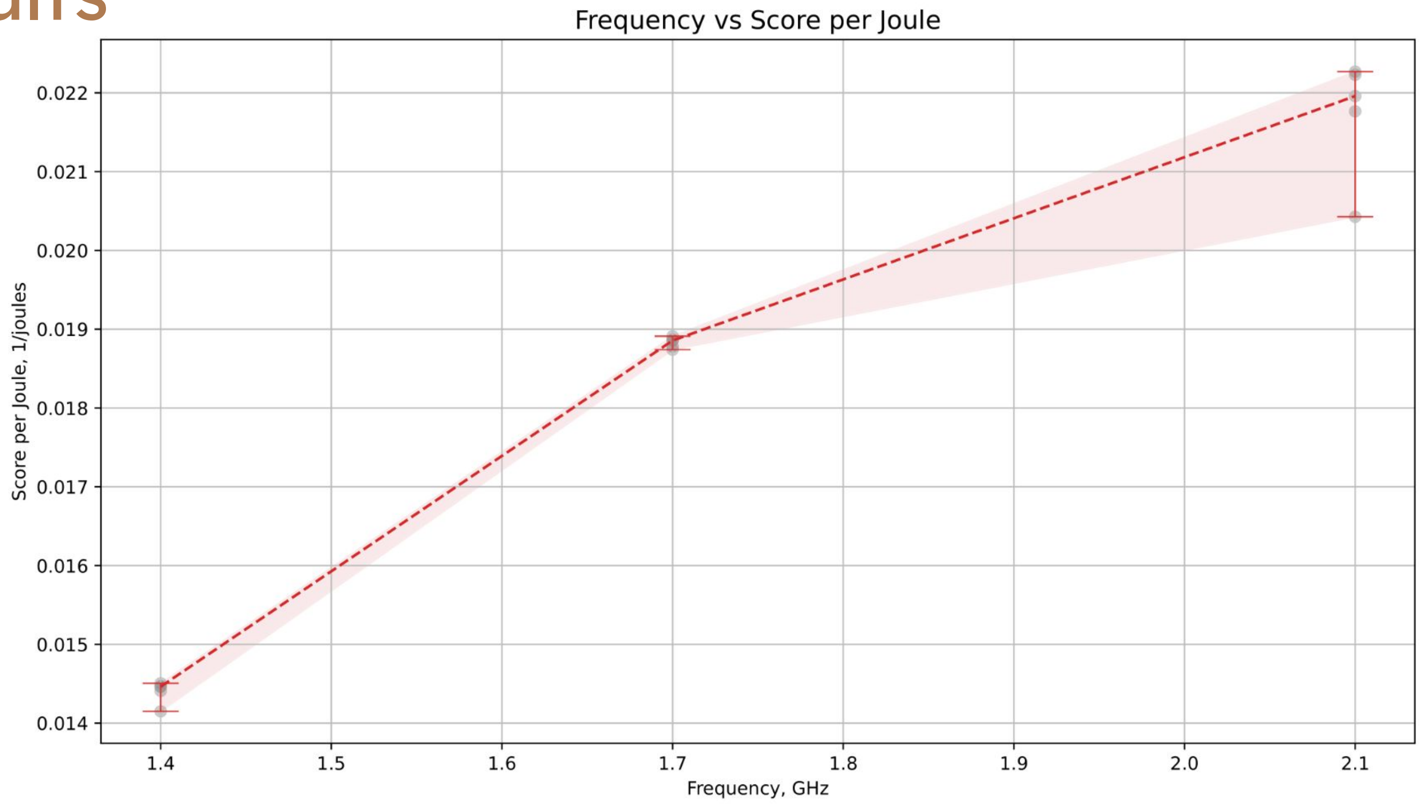

Frequency vs Power consumption
7000
6800
6600
6400
6200
6000
5800
5600
5400
Power consumption, joules
1.4
1.5
1.6
1.7
1.8
1.9
2.0
2.1
Frequency, GHz

Results



Power consumption vs Score
Score
140
130
120
110
100
90
5400
5600
5800
6000
6200
6400
6600
6800
7000
Power consumption, joules

Results


Frequency vs Score per Joule
Score per Joule, 1/joules
0.022
0.021
0.020
0.019
0.018
0.017
0.016
0.015
0.014
1.4
1.5
1.6
1.7
1.8
1.9
2.0
2.1
Frequency, GHz

Summary

- Limiting the CPU frequency reduces power consumption, until a certain point
- Performance and execution time changes monotonically depending on the frequency
- Score to Energy consumption ratio also decreases with limiting the frequency, which means there is more performance loss than energy efficiency gain
- There is no silver bullet: decision making should be careful and and focused on current problems to choose the best trade-off

Experiment 2

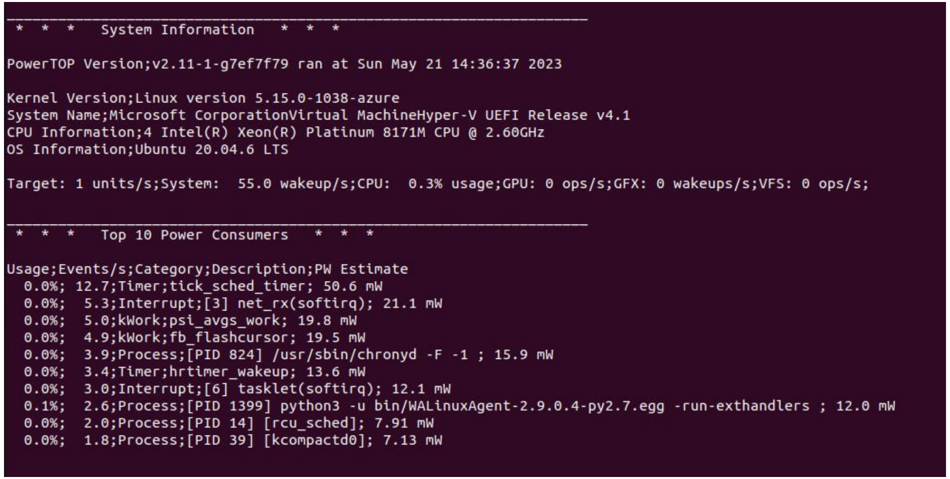
- **Goal**: validate measurement of power consumption
  - Same software setup as in main experiment
  - The laptop is disconnected from charging
  - Charge level of battery recorded before and after benchmark run (`upower`)
- **Important notice**: the laptop battery is not monitored as actively as the CPU (the info is updated only once every 30 seconds, with a certain step of accuracy), so the measurement is approximate.

Results - the ratio of perf's power consumption measurement to battery discharging (statistics of 3 runs):

	ratio		
frequency	**mean**	**max**	**min**
1.7 GHz	0.606138015	0.668914919	0.57420385
2.1 GHz	0.689998177	0.72618657	0.67078084

Future Steps

- More detailed tests at different frequencies
- More comprehensive benchmarking
- Detailed metrics for power consumption
- Experiments on other platforms (e.g. Cloud VM)



PowerTOP

Azure Virtual Machine

Source code:

github.com/andrii0yerko/ER-IPT-HACK_2023

Thank for your attention!