Exploring energy consumption of AI frameworks on a 64-core RV64 Server CPU

Giulio Malenza, Francesco Targa, Adriano Marques Garcia, Marco Aldinucci and Robert Birke

ITADATA-SciHPCExa-2024

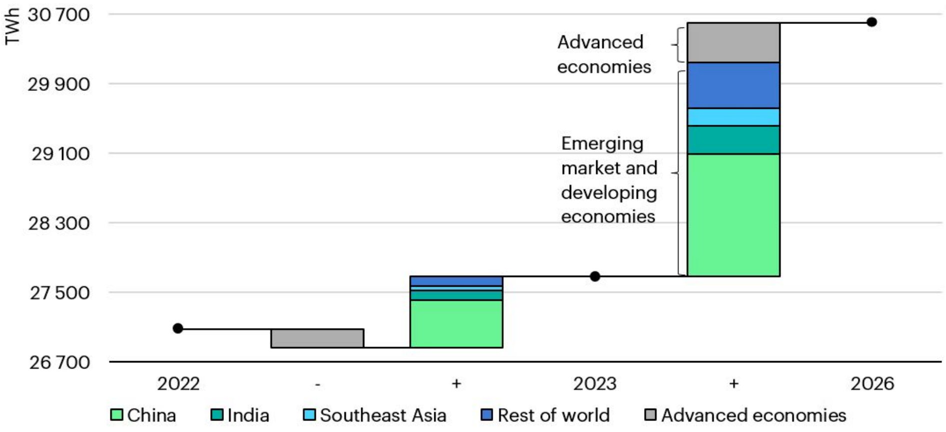
Outline

- Motivations & Goal
- RISC-V SOPHON SG2042
- AI Frameworks
- AI Models
- Scaling
- Power and energy consumption
- Conclusion

Motivation & Goal

AI energy demanding is increasing (2.2% in 2023);

It is estimated that the total energy consumed by "data centers" could exceed 1'000 TWh in 2026 (now ~460 TWh).



[1]

Use generically to indicate: **data centres, artificial intelligence (AI) and the cryptocurrency sector**

Goal: Estimanting energy consumption of AI <u>inference</u> using different frameworks

1- IEA (2024), Electricity 2024, IEA, Paris https://www.iea.org/reports/electricity-2024, Licence: CC BY 4.0

RISC-V SOPHON SG2042

Based on a smaller and open-source Instruction Set Architecture, potential more energy-saving.

Main characteristics:

- 64 core RISC-V C910 CPU (16 cores x 4 clusters)
- Standard Vector Instruction 0.7.1
- 64KB L1 cache per core
- 1MB L2 cache shared by each cluster
- 64MB LLC
- 32 Gen4 PCIe lanes 32GB/s
- 128GB of RAM DDR4 (3200MHz)
- Linux fedora-riscv 6.1.31

PyTorch

Developed by Meta AI → Design principles:

- TORCH VERSION=2.3.0

Compiled with:

- GCC 13.2.1
- C++ 17
- OpenBLAS-0.3.26
- OpenMP 4.5

Design principles:

- Pythonic: First class member of python ecosystem
- Researchers oriented: make easy to create models, data loaders, …
- Pragmatic performance: deliver compelling performance

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T.,Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Köpf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., Chintala,S.: Pytorch: An imperative style, high-performance deep learning library (2019), https://arxiv.org/abs/1912.01703

TensorFlow Lite

TensorFlow Lite

Developed by Google

- TFLITE VERSION=2.18.0

Compiled with:
- GCC 13.2.1
- C++ 17
- XNNPACK - highly optimized solution for neural network inference

- Lightweight version of TensorFlow
- Optimize inference problem on mobile and embedded device
- Develop using specific ML ISA instructions
- Target ISA was RISC-V vector

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., Kudlur, M., Levenberg, J., Monga, R., Moore, S., Murray, D.G., Steiner, B., Tucker, P., Vasudevan, V., Warden, P., Wicke, M., Yu, Y., Zheng, X.: Tensorflow: A system for large-scale machine learning (2016), https://arxiv.org/abs/1605.08695

ONNX Runtime

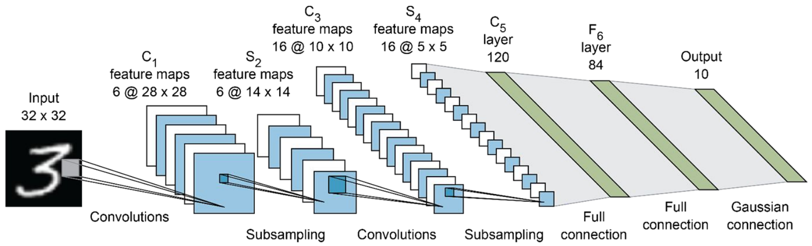
Developed by Microsoft

- Specialized and optimize for inference problems.
- Designed to be flexible and capable to executing inference on different hardware stacks
- Works with different executors providers (XnnpackExecutionProvider)

- ONNX Runtime VERSION=1.17.0

Compiled with:
- GCC 13.2.1
- C++ 17
- XNNPACK

AI Models



https://www.superannotate.com/blog/guide-to-convolutional-neural-networks

CNN Networks

[1] VGG-16 - 2014

- Extend AlexNet network increasing depth using 3x3 conv. filters;

[2] ResNet-50 - 2015

- Extend VGG-16 introducing “short-cut” to avoid the vanishing gradient problem

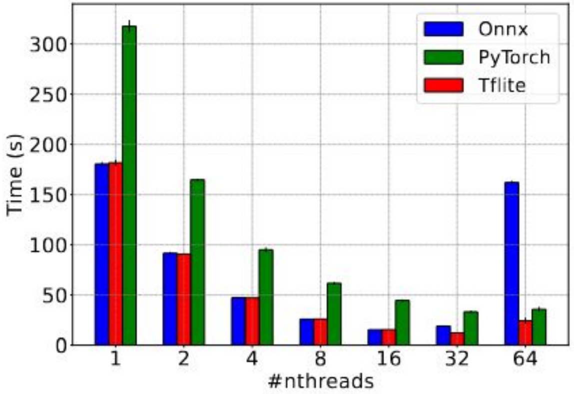
[3] MobileNet-V2 - 2017

- Mobile vision applications
- Use depthwise separable convolutions to reduce computational cost

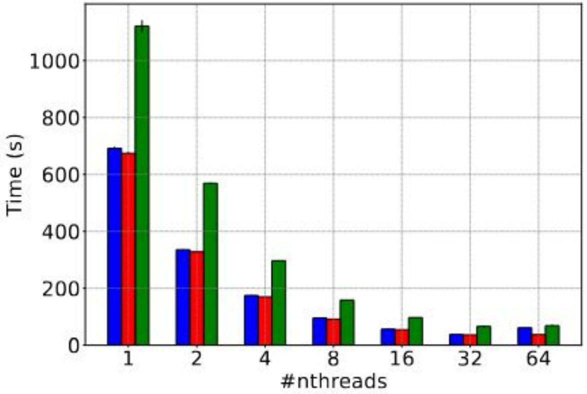
1. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition (2015), https://arxiv.org/abs/1409.1556
2. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition(2015), https://arxiv.org/abs/1512.03385
3. Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H.: Mobilenets: Efficient convolutional neural networks for mobile vision applications (2017), https://arxiv.org/abs/1704.04861

Scaling

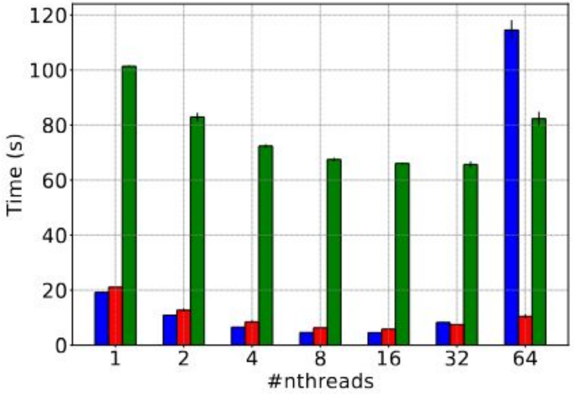
ResNet-50



VGG-16



MobileNet-V2



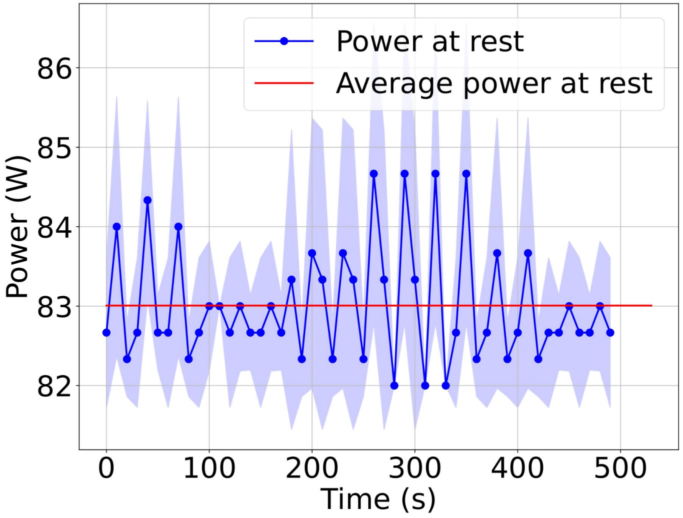
Best Configurations
Number of threads

Model	Torch	TFLite	ONNX
ResNet50	32	32	16
VGG16	32	32	32
MobilenetV2	32	16	16

TAPO P125M

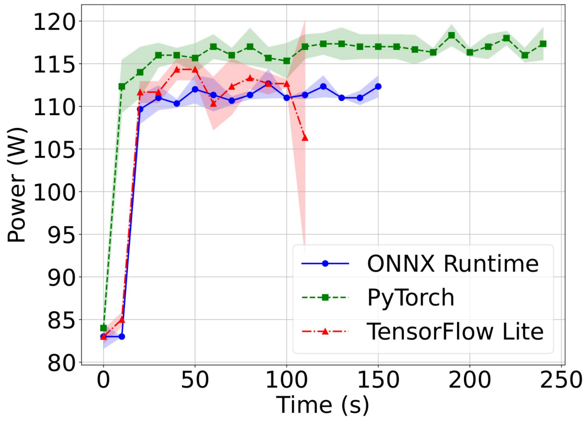
- SG2042 has NO hardware counters dedicated to energy/power consumption;
- Tapo P125M Power Meter, power and energy measured at each second;
- Simulations were performed taking power and energy measures each 10 s;
- Average power consumed at rest: 83.01±0.99 W
  Average energy consumed at rest: 11.33±0.47 Wh.

Power consumption at rest

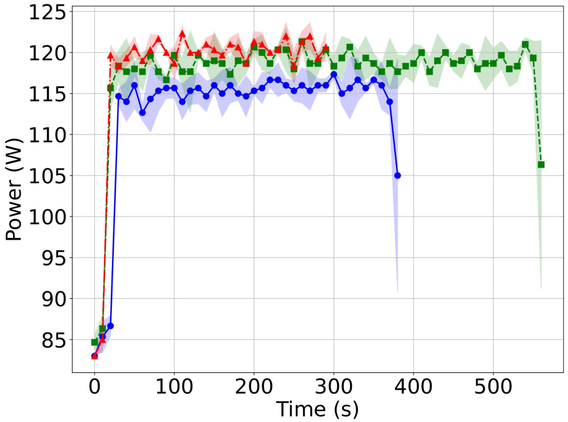


Power consumption

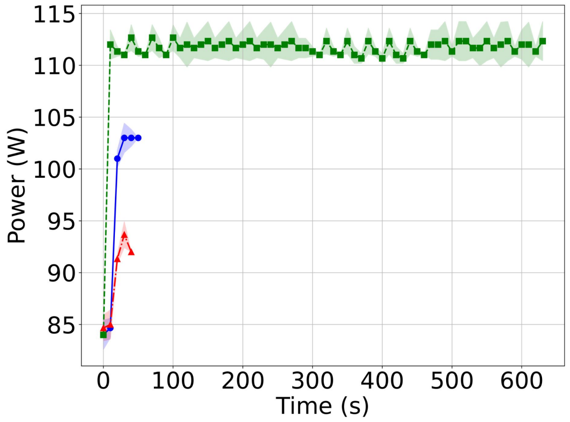
ResNet-50



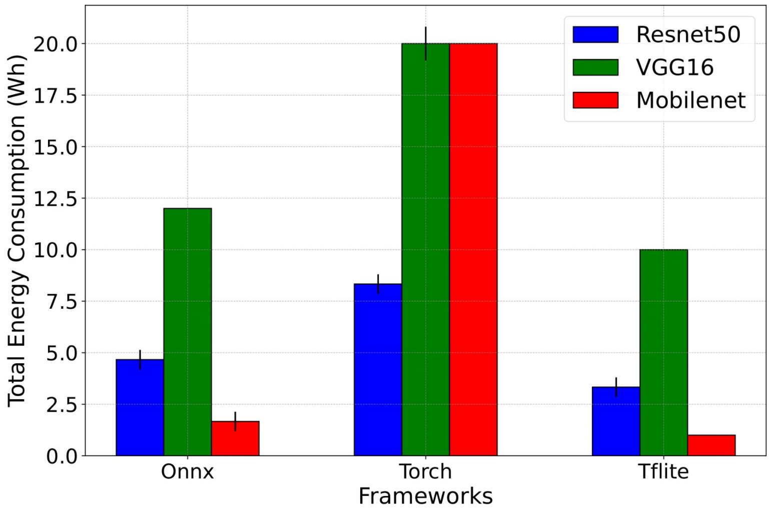
VGG-16



MobileNet-V2



Energy consumption



Energy calculated as the difference between the final and initial energy values read by Tapo.

Model	ONNX vs TFLite	PyTorch vs TFLite
ResNet-50	1.39X	2.42X
VGG-16	1.2X	2.0X
MobileNet	1.6X	20X

Conclusion

- This work explores the energy consumption of the three most popular AI frameworks;
- TensorFlow Lite is the most energy-saving framework;
- ONNX shows comparable performance with a maximum loss of 1.6X;

In the future:

- Analyze performance more deeply, using different acceleration libraries;
- Consider other hardware architectures;
- Increase the number of AI frameworks to be tested;

Acknowledgements
This work has been partially supported by the Spoke 1 ``FutureHPC \& BigData'' of the ICSC--Centro Nazionale di Ricerca in High Performance Computing, Big Data and Quantum Computing and hosting entity, funded by European Union—Next GenerationEU; and by the European Union under the project DYMAN (grant n. 101161930).