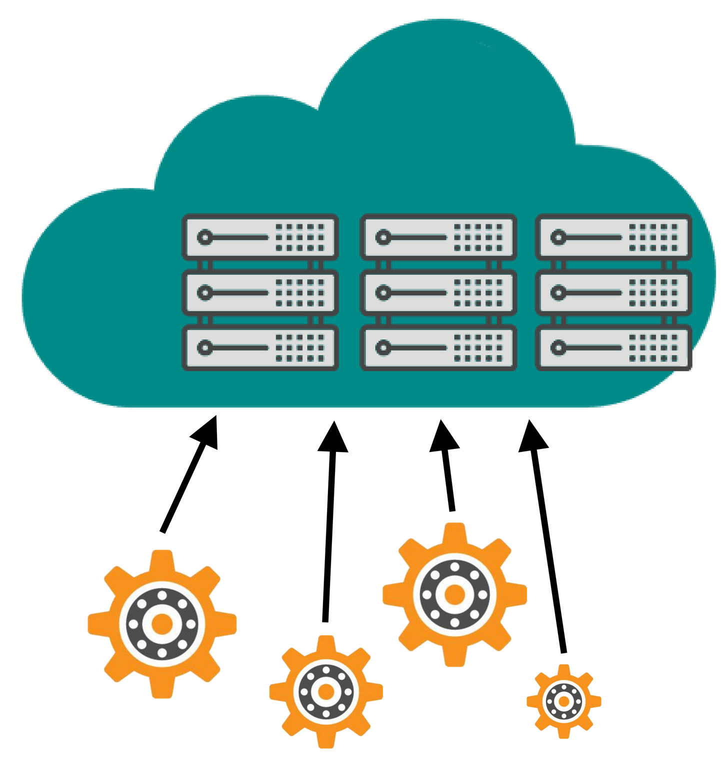
Leveraging LSTMs for interference-aware run-time system Predictability of ML cloud workloads

Dimosthenis Masouros, Sotirios Xydis, Dimitrios Soudris
Microprocessors and Digital Systems Laboratory, ECE, National Technical University of Athens, Greece
{demo.masouros, sxydis, dsoudris}@microlab.ntua.gr

1. The Problem



Rapid **increment** in the number of **workloads** uploaded and executed on the **Cloud**

These workloads are **co-located** on the same physical server machines causing **interference** to each other

How to **efficiently place** and **control** workloads on a DC environment**?**

2. State-of-the-art approaches

Predict workload performance **slowdown** or tail-latency due to interference **in a static one-off way** [1, 2].

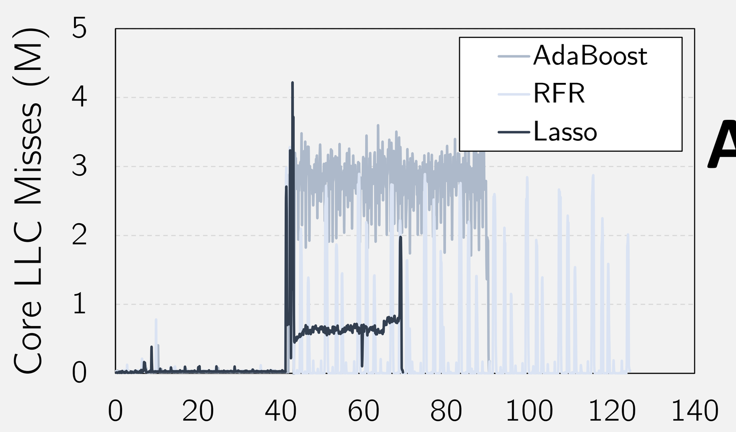
BUT

they **fail** to model the **impact** of each resource on the performance degradation

&

recent **advancements** in the system-level management of hardware **allow fine-grained resource tuning**
Power Capping [3], Cache Allocation [4], Resources usage (cgroups)

3. Motivation & Proposed Solution

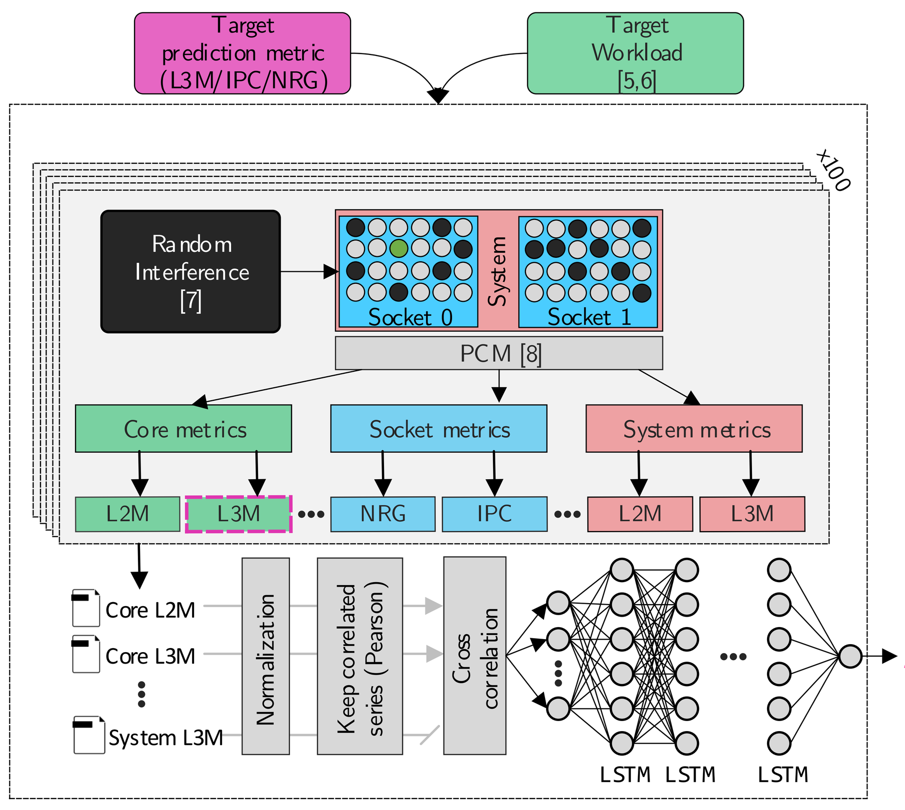


Applications experience **different phases** throughout their lifetime

Runtime schedulers should **dynamically predict** per application **resource needs under interference** to proactively control resources on the system

Leverage **Long Short-Term Memory (LSTM)** networks to

predict runtime system metrics under interference



4. Proposed Framework

Offline Part

> Workload execution with different interference
> Collect system metrics
> Design Space Exploration & Training

Online Part

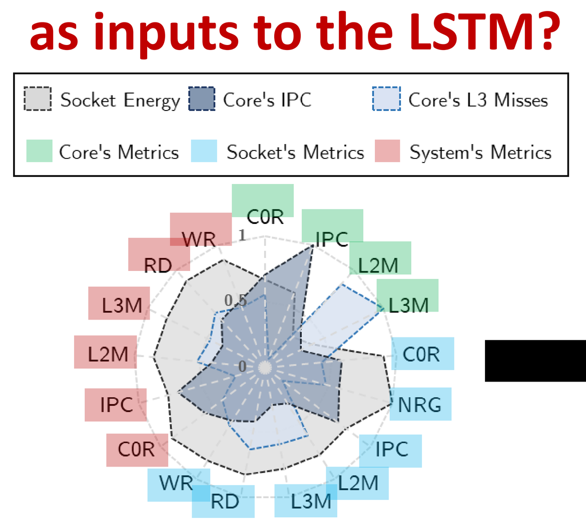
- Monitor workload during execution and predict future values of system metrics

5. Experimental Setup

- Applications from **scikit-learn**[5] and **cloudsuite**[6] as our target **workloads**
- Emulate **interference** using the **ibench** suite [7]
- **Monitor system** using Performance Counter Monitoring **(PCM) tool** [8] and collect system metrics.
- Train LSTM model to predict future values of desired metrics (IPC, LLC misses, Energy Consumption)

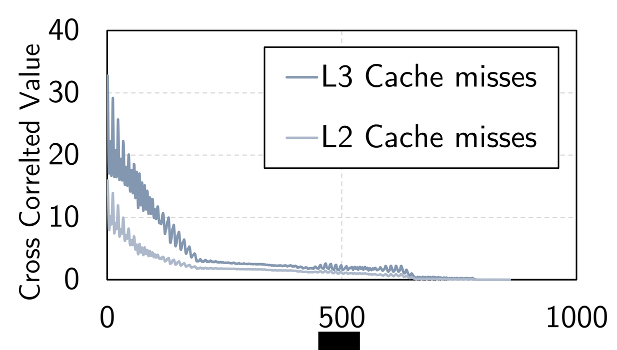
6. Design Space Exploration

Q1: What metrics to choose as inputs to the LSTM?



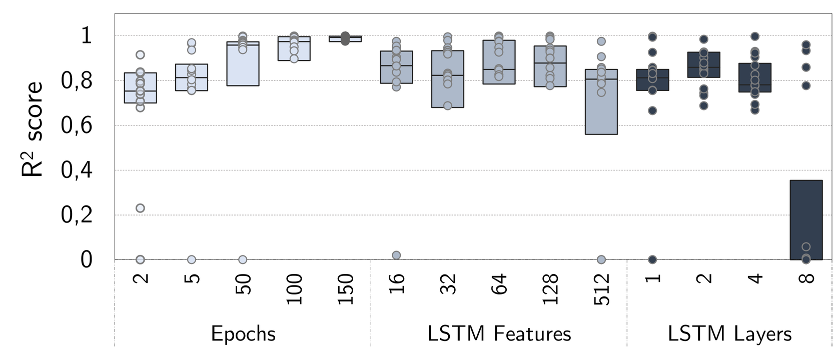
A1: Calculate the pearson correlation between all signals and select the two most correlated

Q2: How far back to seek for valuable information?



A2: Calculate the cross correlation of pearson correlated signals and select a proper value

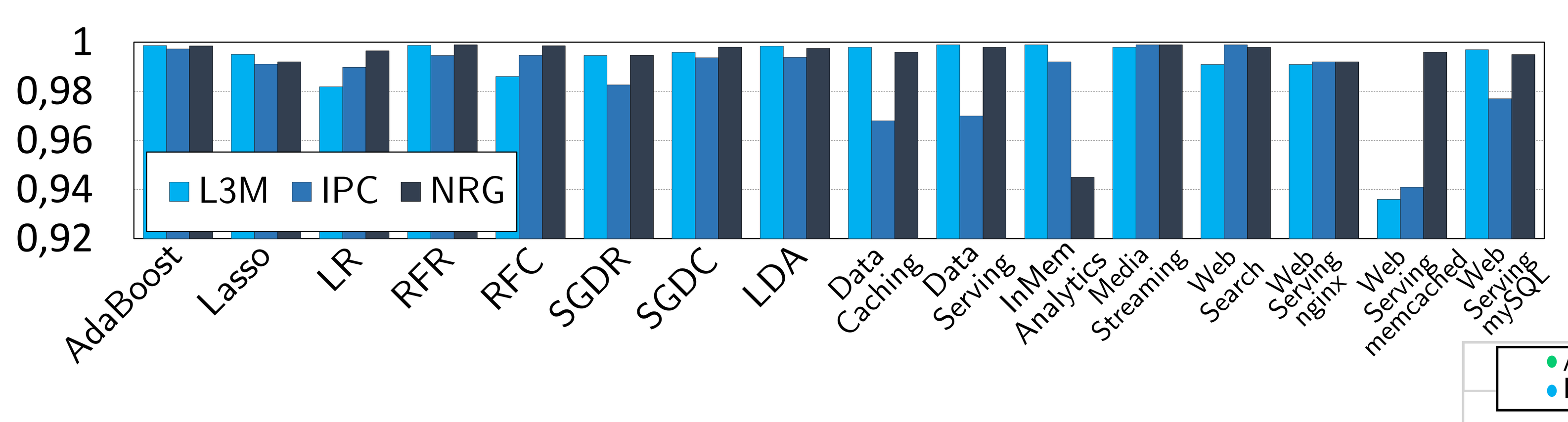
Q3: How many layers and features to use in the LSTM network?



Overall Best Architecture
4 Layers
128 Features
150 training epochs

A3: Explore the impact of different design parameters on the accuracy of the model.

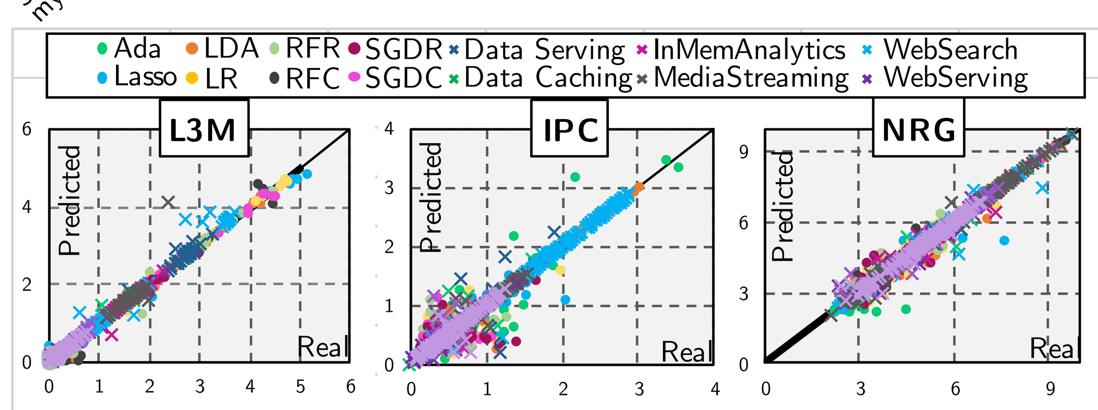
7. Evaluation



High predictability of system-level metrics under interference, achieving on average

$$R^2 = 0.987$$

High Level of accuracy for all the three target prediction variables



References

[1] Christina Delimitrou and Christos Kozyrakis. 2013. Paragon: QoS-aware scheduling for heterogeneous datacenters. InACM SIGPLAN Notices, Vol. 48. ACM,77–88
[2] David Lo, Liqun Cheng, Rama Govindaraju, Parthasarathy Ranganathan, andChristos Kozyrakis. 2015. Heracles: Improving resource efficiency at scale. InACM SIGARCH Computer Architecture News, Vol. 43. ACM, 450–462.
[3] Howard David, Eugene Gorbatov, Ulf R Hanebutte, Rahul Khanna, and ChristianLe. 2010. RAPL: memory power estimation and capping. InProceedings of the 16thACM/IEEE international symposium on Low power electronics and design. ACM,189–194.
[4] Andrew Herdrich, Edwin Verplanke, Priya Autee, Ramesh Illikkal, Chris Gianos,Ronak Singhal, and Ravi Iyer. 2016. Cache QoS: From concept to reality in the Intel®Xeon®processor E5-2600 v3 product family. InHigh Performance ComputerArchitecture (HPCA), 2016 IEEE International Symposium on. IEEE, 657–668.
[5] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel,Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss,Vincent Dubourg, and others. 2011. Scikit-learn: Machine learning in Python.Journal of machine learning research12, Oct (2011), 2825–2830.
[6] M. Ferdman, A. Adileh, O. Kocberber, S. Volos, M. Alisafaee,D. Jevdjic, C. Kaynak, A. D. Popescu, A. Ailamaki, and B. Falsafi,"Clearing the clouds: A study of emerging scale-out workloadson modern hardware,"Proceedings of the Seventeenth InternationalConference on Architectural Support for Programming Languages andOperating Systems, 2012.
[7] Processor Counter Monitor (PCM). [Online]. Available: https://github.com/opcm/pcm

This work has been partially funded by EU Horizon 2020 programme under grant agreement No 801015 EXA2PRO (https://exa2pro.eu/).