RÉPUBLIQUE FRANÇAISE
Liberté
Égalité
Fraternité

ENSTA | INSTITUT POLYTECHNIQUE DE PARIS | Lab-STICC

LES TRANS NUMÉRIQUES

Du 2 au 5 février 2026
Rennes, France

FRANCE 2030 — PROGRAMME DE RECHERCHE CLOUD

Is AI Hitting the Storage Wall?

Why We Should Care ?

Jalil Boukhobza, ENSTA Institut Polytechnique de Paris, Lab-STICC, Brest Campus

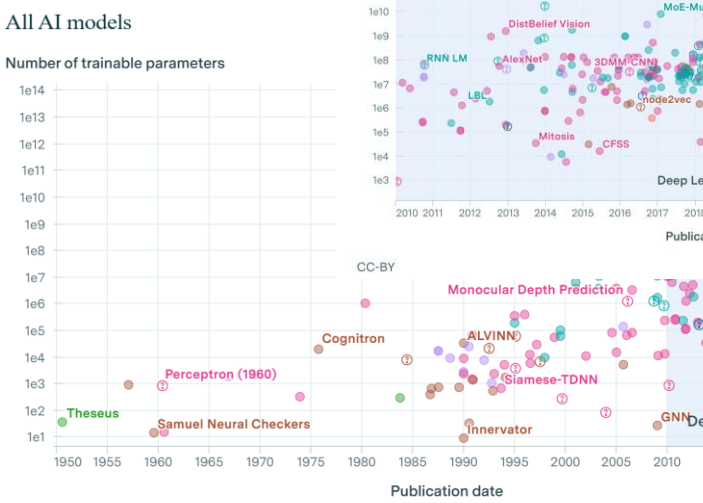
Outline

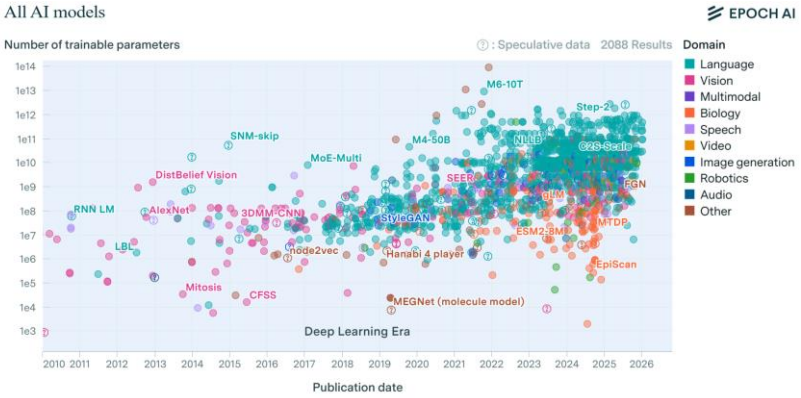
1. Some figures
2. Back to the 60s
3. LLM, what else ?

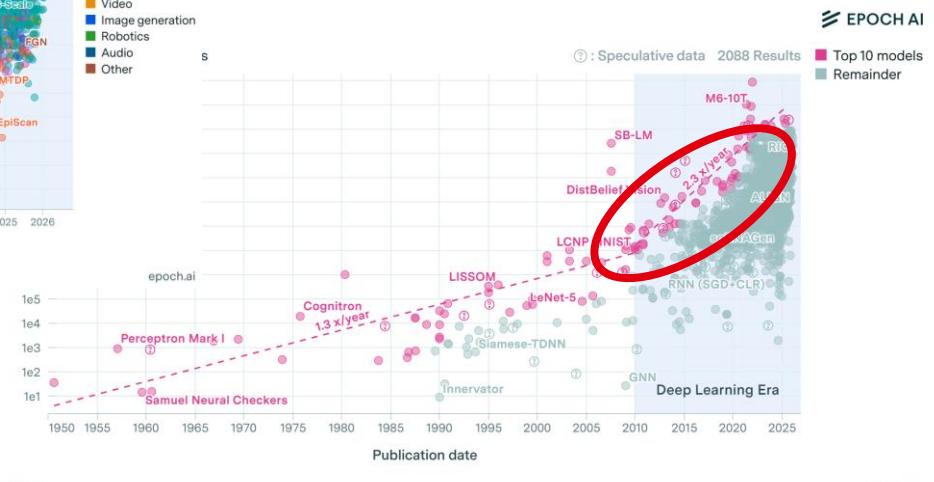
1. Some figures

Model size

A size perspective







- AI models are growing fast 2.3x per year (for the top 10 models)
- They are driven by language models
- Promising variability

Model compute

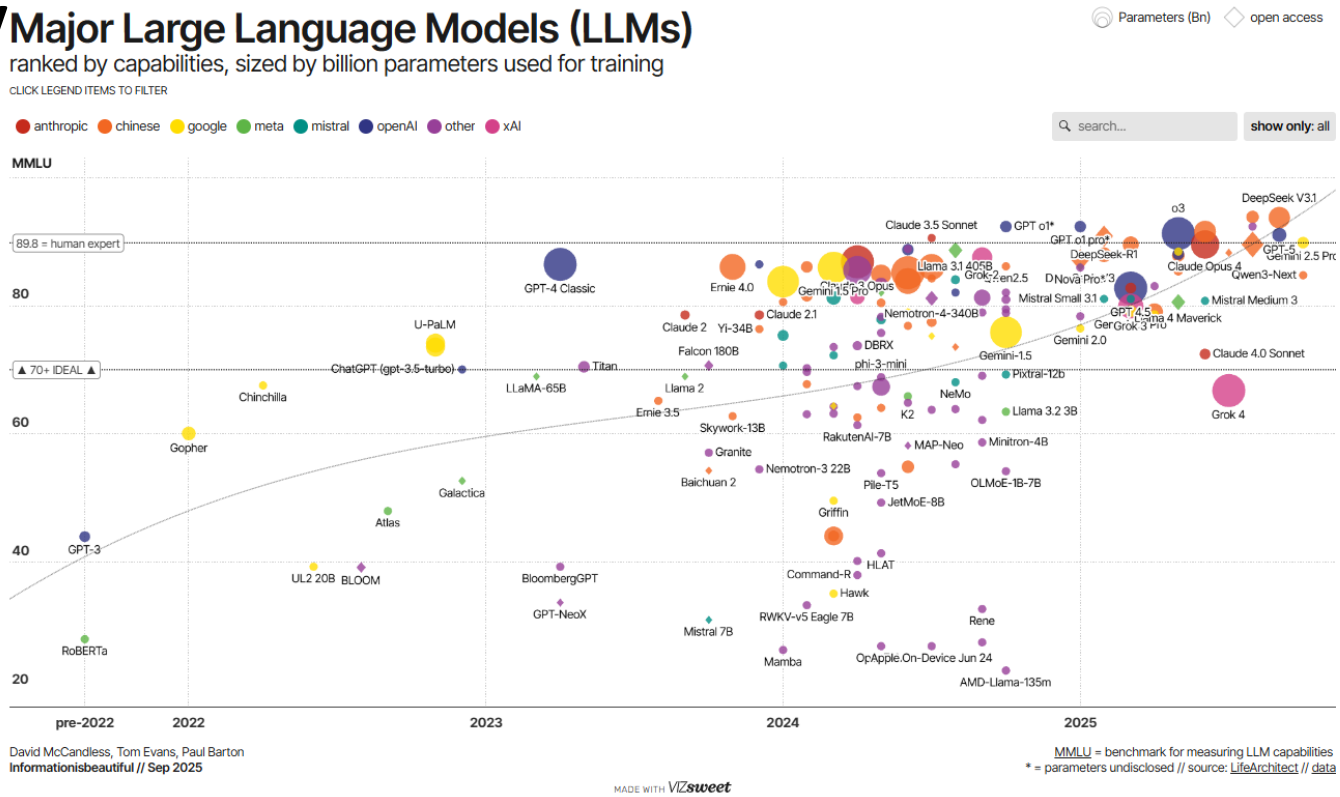
A compute perspe

- Training compute is growing faster 4.7x per year (for the top 10 models)
- They are driven by language models too
- Promising variability too

Model efficiency

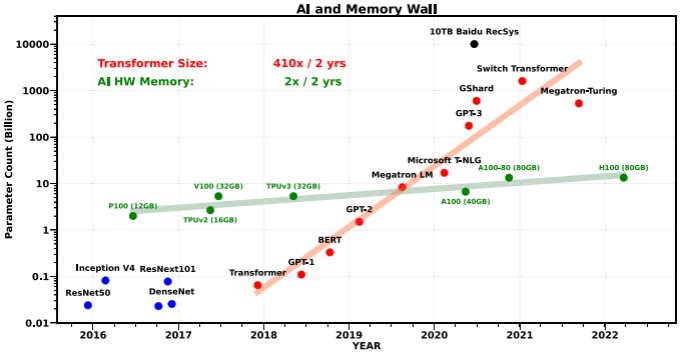
A quality perspective

- Efficiency is correlated to size
- There is room for tradeoffs
- Specialized SLM ? (what is « small » ? → see next slides)

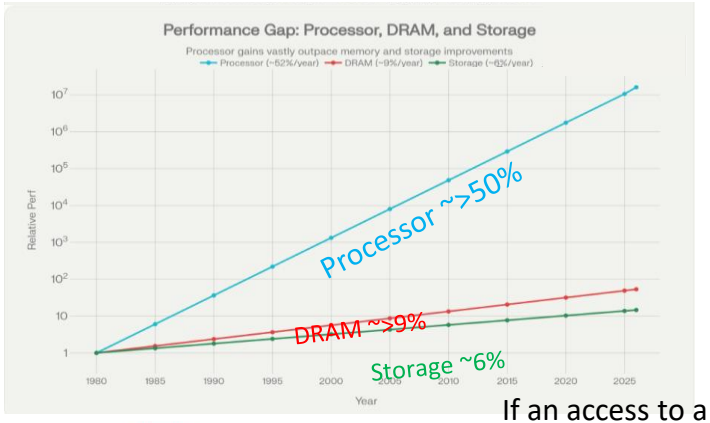


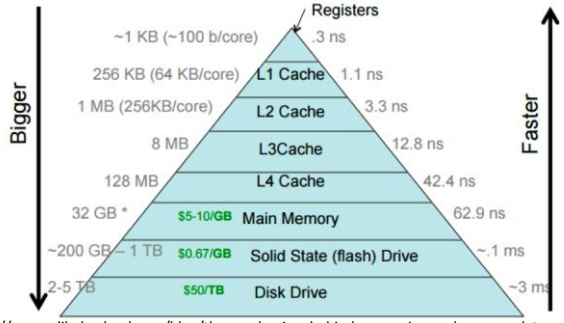
Memory wall or storage wall … a double wall effect

Tier	Total VRAM	Max simultaneous 7B models
Edge	1–8 GB	0–1
PC	12–24 GB	2–4
Server	80–640 GB	10–50
Data center	TBs	100s



A. Gholami, Z. Yao, S. Kim, C. Hooper, M. W. Mahoney and K. Keutzer, "AI and Memory Wall," in *IEEE Micro*, vol. 44, no. 3, pp. 33-39, May-June 2024,



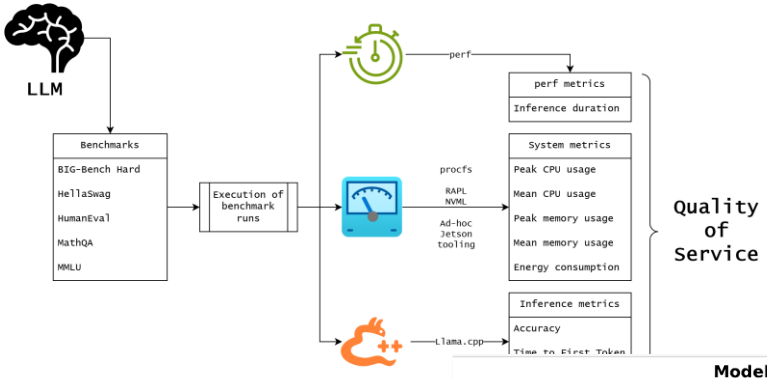


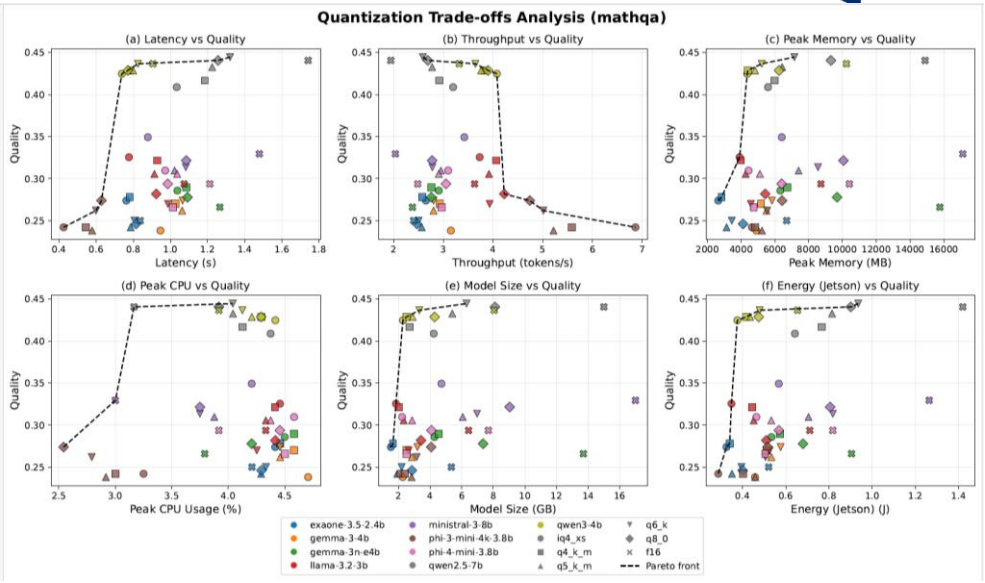
https://www.alibabacloud.com/blog/the-mechanism-behind-measuring-cache-access-latency_599384

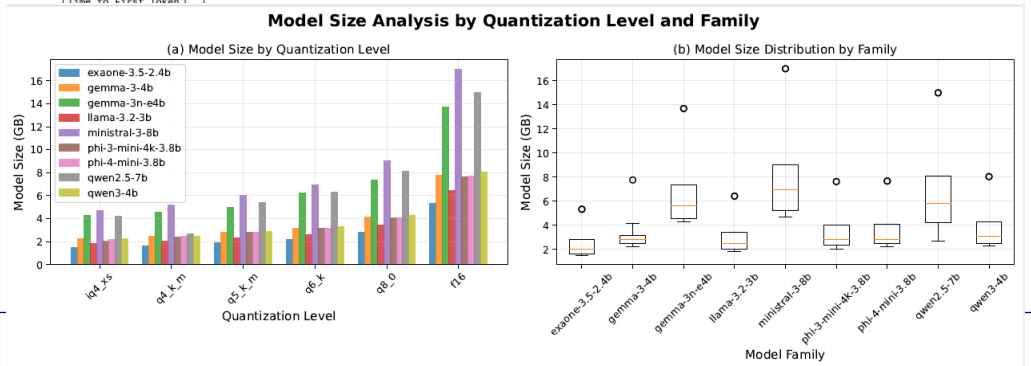
If an access to a register was a 1m distance:

- x4 - L1 Cache → 4m
- x3 - L2 Cache → 12m
- x4 - L3 Cache → 48m
- x3 - L4 Cache → ~150m
- x4 - DRAM → ~600m
- X1500 - SSD → 900 km
- x3 - HDD → 2700km

Work in progress: How small are SLMs ?







2. Back to the 60s

1957-1962: remember the Virtual memory

Tom Kilburn at the Ferranti Mark I computer
https://www.computerhistory.org/timeline/1962/

- ❑ Memory was too expensive
- ❑ Process address spaces were too large
- ❑ Manual management was too cumbersome
- ❑ From mainframes → personal computers → embedded devices

"The concept of virtual memory emerges from a team under the direction of Tom Kilburn at the University of Manchester on its Atlas computer. Virtual memory permitted a computer to ***use its storage capacity to switch rapidly among multiple programs or users and was a key requirement for timesharing."*** https://www.computerhistory.org/timeline/1962/

1960s, swapping on the road to multi-programming

- Memory was too small because expensive
- Increase multi-programming threshold
- Switching between full programs (before adding paging)
- From mainframes → personal computers → embedded devices

"*The term originated in the time-sharing systems of the early 1960s. These systems had insufficient memory to allow* ***more than one program*** *to be loaded (swapped in) for execution at a time. At the end of a time slice or stop for input/output, their operating systems swapped out the executing program and then swapped in another program. A single program could be swapped many times during its execution. Swapping was a perfect description of the main work of time-sharing -- switching the CPU from one program to another.*"

https://denninginstitute.com/pjd/PUBS/ENC/swapping08.pdf

PDP-1, one of the first systems integrating full swapping
https://www.computerhistory.org/timeline/1962/

How about ~~processes~~ large models?

- Memory ~~was~~ is too small because expensive
- Increase multi-~~programming~~ model threshold
- Switching between full ~~programs~~ models (before adding ~~paging~~ ???)
- From ~~mainframes~~ datacenters → personal computers → embedded devices

"The term originated in the time-sharing systems of the early 1960s. These systems had insufficient memory to allow more than one ~~program~~ model to be loaded (swapped in) for execution at a time. At the end of a time slice or stop for input/output, the operating system swapped out the executing ~~program~~ model and swapped in another ~~program~~ model. A single ~~program~~ model could be swapped many times during its execution. Swapping was the perfect description of the main work of time-sharing -- switching the CPU or GPU? from one ~~program~~ model to another."

https://denninginstitute.com/pjd/PUBS/ENC/swapping08.pdf

- Memory ~~was~~ is still too expensive
- ~~Process address spaces~~ Models are too large
- Manual management ~~was~~ is still too cumbersome
- From ~~mainframes~~ datacenters → personal computers → embedded devices

*"~~The concept of virtual memory emerges from a team under the direction of Tom Kilburn at the University of Manchester on its Atlas computer.~~ Virtual memory ??? Would permit a computer to **use its storage capacity to switch rapidly among multiple ~~programs~~ models ~~or users and~~ is a key requirement for timesharing.** models."*

https://www.computerhistory.org/timeline/1962/

Models are not simple process address spaces…

Context:

- ❑ Memory pressure / cost
- ❑ Size
- ❑ Manual management is not feasible
- ❑ Storage performance disparity

Similarities:

- ❑ Need automatic management
- ❑ Whole lifecycle management: loading, management, cleaning
- ❑ Different QoS
- ❑ Multi-tenancy is important
  - ❑ Shared memory space/model
  - ❑ isolation
- ❑ datacenters → personal computers → embedded/edge devices

Differences:

- ❑ Model data structures could be scanned fully during inference (not just a subset of pages)
  - ❑ But other structures might be partially read (KV cache, RAG chunks, …)
- ❑ Latency is/might be important → storage performance is an issue
- ❑ Models are configurable
- ❑ …


Efficient Memory Management for Large Language Model Serving with *PagedAttention*

Woosuk Kwon[1,*] Zhuohan Li[1,*] Siyuan Zhuang[1] Ying Sheng[1,2] Lianmin Zheng[1] Cody Hao Yu[3]
Joseph E. Gonzalez[1] Hao Zhang[4] Ion Stoica[1]
[1]UC Berkeley [2]Stanford University [3]Independent Researcher [4]UC San Diego


BlockLLM: Multi-tenant Finer-grained Serving for Large Language Models

Bodun Hu, *The University of Texas at Austin*
Jiamin Li, *Microsoft Research*
Le Xu, *The University of Texas at Austin*
Myungjin Lee, *Cisco Research*
Akshay Jajoo, *Cisco Research*
Geon-Woo Kim, *The University of Texas at Austin*
Hong Xu, *The Chinese University of Hong Kong*
Aditya Akella, *The University of Texas at Austin*

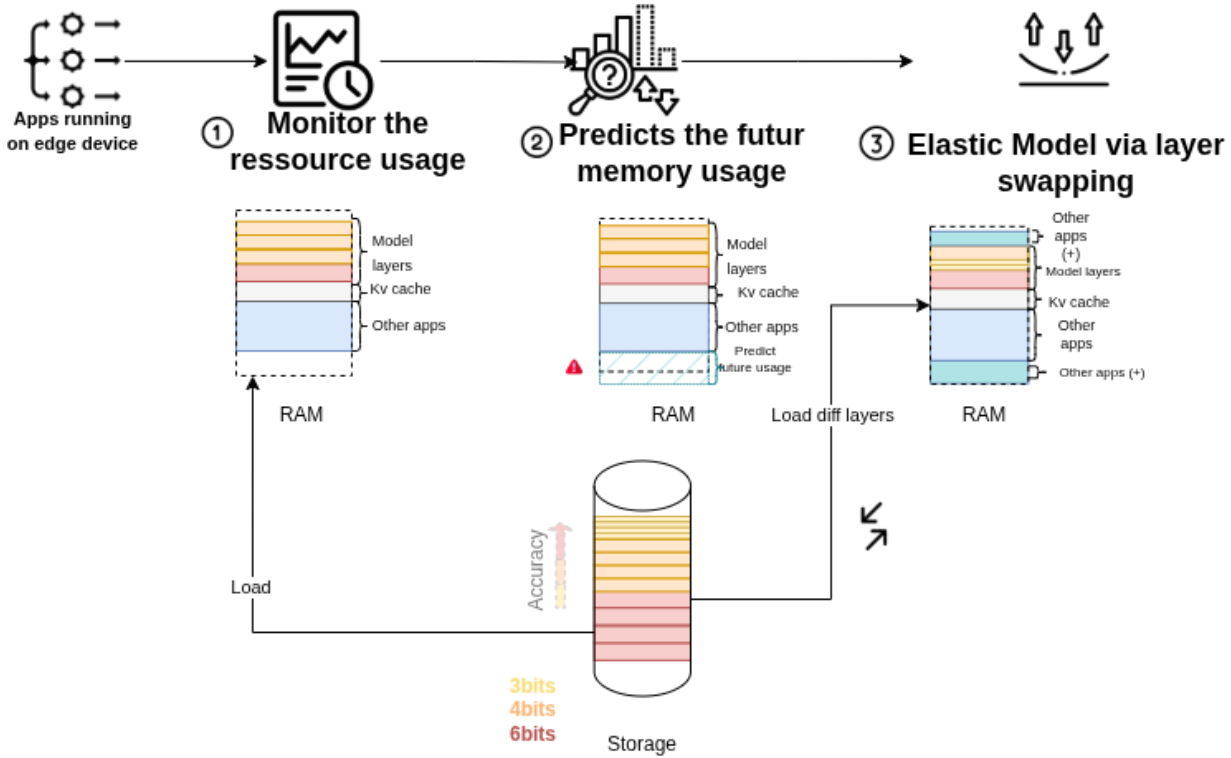

Engine-Agnostic Model Hot-Swapping for Cost-Effective LLM Inference

Radostin Stoyanov, University of Oxford, Oxford, United Kingdom, radostin.stoyanov@eng.ox.ac.uk
Viktória Spišaková, Masaryk University, Brno, Czech Republic, spisakova@ics.muni.cz
Adrian Reber, Red Hat, Stuttgart, Germany, areber@redhat.com
Wesley Armour, University of Oxford, Oxford, United Kingdom, wes.armour@eng.ox.ac.uk
Marcin Copik, ETH Zurich, Zurich, Switzerland, marcin.copik@inf.ethz.ch
Rodrigo Bruno, INESC-ID, Instituto Superior Técnico, University of Lisbon, Lisbon, Portugal, rodrigo.bruno@tecnico.ulisboa.pt

Work in progress:

An ELASTIC LLM for Edge Use cases

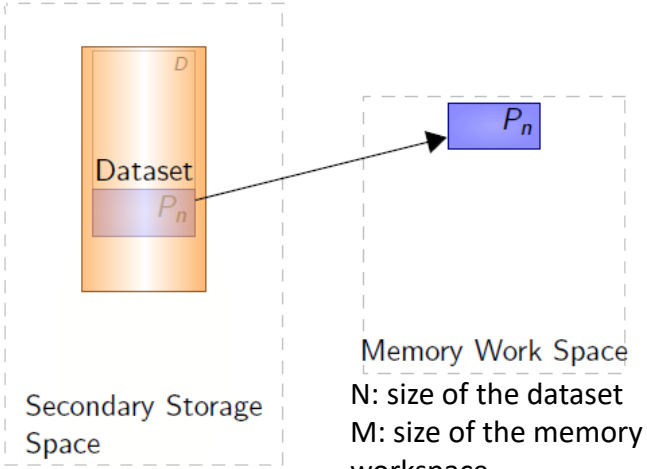
- **Motivation**: Edge LLMs run at different model sizes on devices where multiple apps compete for limited memory
- **Idea**: Design an **elastic LLM** that adjusts its model size under memory pressure from co-running applications, balancing accuracy and resource usage with minimal overhead, using **mixed quantization.**



3. LLM, what else ?

Learning on the edge

- Several ML algorithms → go through all (or most) of the dataset during the learning process
- When **the dataset size > memory workspace** → I/O swapping issues ↗↗↗



N: size of the dataset
M: size of the memory workspace

Iteration : 2

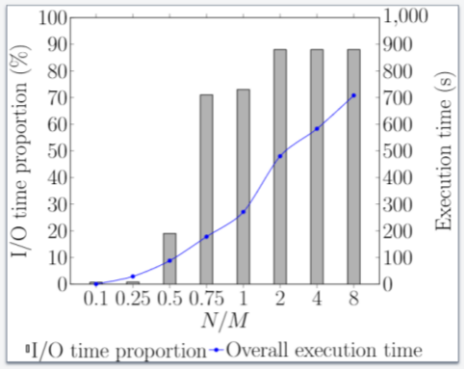
For → some initial state … until something converges
- **read all (or at least a large part of) the dataset**
- **perform some (more or less complex) processing**

update the model

Tightly coupled instructions

Each dataset loading « read all… » = a lot of swapping operations (depending on the dataset to memory proportion).
Very poor temporal locality

Example with Random forests on a BB board

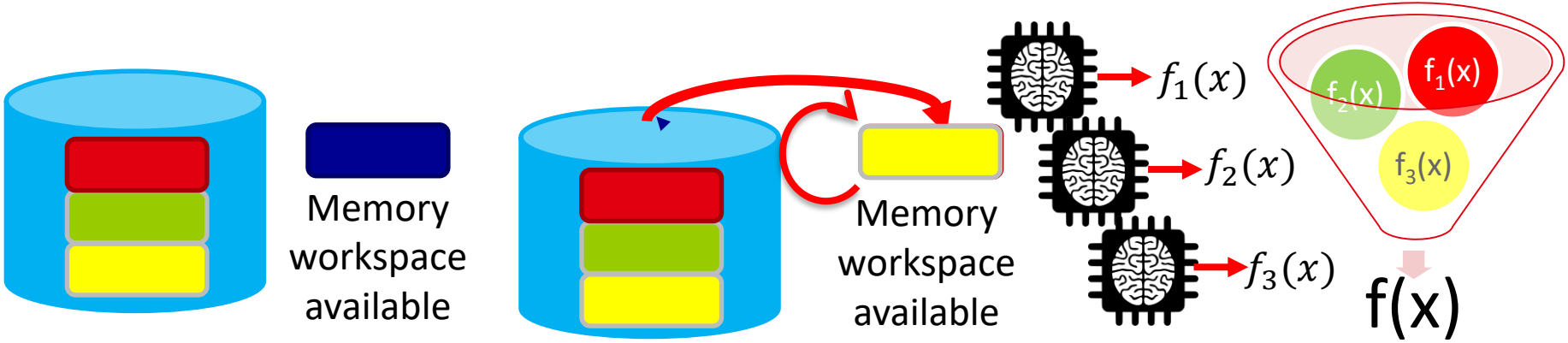


Training on the edge … you can't help yourself from swapping !

Subdivide the dataset

Process each chunk separatly

Aggregate the models



2019 IEEE 27th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS)

K-MLIO: Enabling K-Means for Large Data-sets and Memory Constrained Embedded Systems

Camélia Slimani
Univ Brest
Lab-STICC, CNRS, UMR 6285
F-29200 Brest, France
Camelia.Slimani@univ-brest.fr

Stéphane Rubini
Univ Brest
Lab-STICC, CNRS, UMR 6285
F-29200 Brest, France
rubini@univ-brest.fr

Jalil Boukhobza
Univ Brest
Lab-STICC, CNRS, UMR 6285
F-29200 Brest, France
boukhobza@univ-brest.fr

- ❑ Train the GMM on data chunks
- ❑ Train on a subset of data
- ❑ Merge the models incrementally
- ❑ Reduce I/Os by 80% and training time by >40%

IEEE TRANSACTIONS ON COMPUTERS, VOL. 72, NO. 6, JUNE 2023 1595

Accelerating Random Forest on Memory-Constrained Devices Through Data Storage Optimization

Camélia Slimani, Chun-Feng Wu, *Member, IEEE*, Stéphane Rubini, Yuan-Hao Chang, *Senior Member, IEEE*, and Jalil Boukhobza, *Senior Member, IEEE*

- ❑ Train the k-means on data chunks that fit in the memory
- ❑ Merge the whole models by sampling the dataset
- ❑ Reduce I/Os by 90% and training time by >50%

ACM SAC' 24

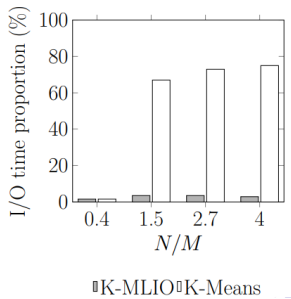
PIGMMaLIOn: a Partial Incremental Gaussian Mixture Model with a Low I/O Design

Meriem Bouzouad[1,2], Yasmine Benhamadi[1,2], Camélia Slimani[1], Jalil Boukhobza[1]
[1] ENSTA Bretagne, Lab-STICC, CNRS, UMR 6285, Brest, France
[2] École Nationale Supérieure D'Informatique (ESI), Algiers, Algeria
{im_bouzouad,iy_benhamadi}@esi.dz,{camelia.slimani,jalil.boukhobza}@ensta-bretagne.fr

- ❑ Load only necessary data for tree building
- ❑ Dynamic loading and mergin of models
- ❑ Reduce I/Os by 80% and training time by more than 60%

N/M	I/O time (%)		K-MLIO I/O time reduction
	K-means	K-MLIO	
1.5	67	3.5	94%
2.7	73	3.5	95%
4	75	3	96%

Table: Réduction du temps d'E/S de K-MLIO

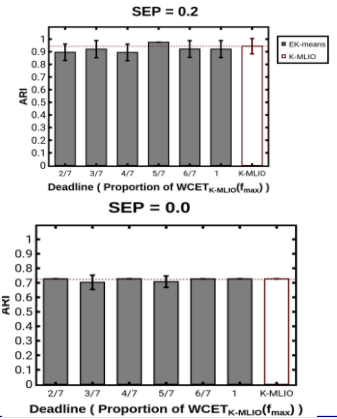


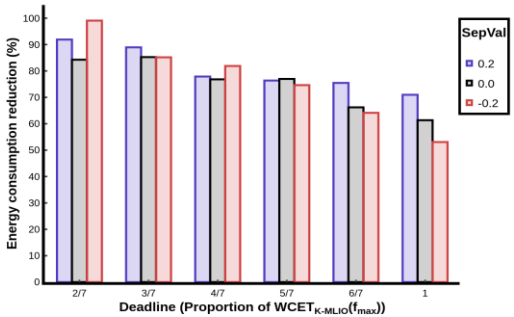
- Training under time budget
- Reducing energy consumption with DVFS

IEEE MASCOTS'23

Training K-means on Embedded Devices: a Deadline-aware and Energy Efficient Design

Hafsa Kara Achira*†, Camélia Slimani *, Jalil Boukhobza*
*ENSTA Bretagne, Lab-STICC, CNRS, UMR 6285, F-29200 Brest. France
† École Nationale Supérieure D'Informatique (ESI), Algiers, Algeria
Email: ih_karaachira@esi.dz, camelia.slimani@ensta-bretagne.fr, jalil.boukhobza@ensta-bretagne.fr





Applied Computing Review

Adapting Gaussian Mixture Model Training to Embedded/Edge Devices: A Low I/O, Deadline-aware and Energy Efficient Design

Meriem Bouzouad
ENSTA Bretagne, Lab-STICC, CNRS, UMR 6285
France
im_bouzouad@esi.dz

Yasmine Benhamadi
ENSTA Bretagne, Lab-STICC, CNRS, UMR 6285
France
iy_benhamadi@esi.dz

Camélia Slimani
ENSTA Bretagne, Lab-STICC, CNRS, UMR 6285
France
camelia.slimani@ensta-bretagne.fr

Jalil Boukhobza
ENSTA Bretagne, Lab-STICC, CNRS, UMR 6285, France
Insititute of Information Science, Academia Sinica, Taiwan
jalil.boukhobza@ensta-bretagne.fr

Conclusion

RICHARD SITES

and Storage…

It's the Memory, Stupid!

When we started the Alpha architecture design in 1988, we estimated a 25-year lifetime and a relatively modest 32% per year compounded performance improvement of implementations over that lifetime (1,000× total). We guestimated about 10× would come from CPU clock improvement, 10× from multiple instruction issue, and 10× from multiple processors.

5, 1996 MICROPROCESSOR REPORT

I expect that over the coming decade memory subsystem design will be the *only* important design issue for microprocessors.

Richard L. Sites is a senior architect at Digital Equipment Corp., where he co-led the design of the Alpha architecture. He can be reached at sites@pa.dec.com.

Some references

[1] Jalil Boukhobza, Pierre Olivier, Wen Sheng Lim, Liang-Chi Chen, Yun-Shan Hsieh, Shin-Ting Wu, Chien-Chung Ho, Po-Chun Huang, and Yuan-Hao Chang. 2025. A Survey on Flash-Memory Storage Systems: A Host-Side Perspective. ACM Trans. Storage 21, 3, Article 24 (August 2025),

[2] Jalil Boukhobza, Pierre Olivier, Flash Memory Integration, Performance and Energy Considerations, ISBN: 9781785481246, ISTE Press - Elsevier, Published Date: 13th March 2017, Page count: 266.

[3] Meriem Bouzouad, Yasmine Benhamadi, Camélia Slimani, and Jalil Boukhobza. 2024. Adapting Gaussian Mixture Model Training to Embedded/Edge Devices: A Low I/O, Deadline-Aware and Energy Efficient Design. SIGAPP Appl. Comput. Rev. 24, 2 (June 2024), 5–18.

[4] Meriem Bouzouad, Yasmine Benhamadi, Camelia Slimani, and Jalil Boukhobza. 2024. PIGMMaLIOn: a Partial Incremental Gaussian Mixture Model with a Low I/O Design. In Proceedings of the 39th ACM/SIGAPP Symposium on Applied Computing (SAC '24). Association for Computing Machinery, New York, NY, USA, 428–435.

[5] Hafsa Kara Achira, Camélia Slimani, Jalil Boukhobza, Training K-Means on Embedded Devices: A Deadline-Aware and Energy Efficient Design, 31st International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS), pp. 1-8, Stony Brook, NY, USA, 2023

[6] Camélia Slimani, Stéphane Rubini and Jalil Boukhobza. "K-MLIO: Enabling K-Means for Large Data-sets and Memory Constrained Embedded Systems", in Proceedings of the IEEE International Symposium on the Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (IEEE MASCOTS), pp. 262-268, Rennes, October 2019

[7] Camélia Slimani, Chun-Feng Wu, Stéphane Rubini, Yuan-Hao Chang, Jalil Boukhobza, Accelerating Random Forest on Memory-Constrained Devices Through Data Storage Optimization. IEEE Trans. Computers 72(6): 1595-1609, 2023

RÉPUBLIQUE FRANÇAISE
Liberté
Égalité
Fraternité

Jalil.boukhobza@ensta.fr

Retrouvez les #TransNumériques sur LinkedIn

cea
cnrs
ifp Energies nouvelles
Institut Mines-Télécom
INRAE
Inria
Université Gustave Eiffel
UGA Université Grenoble Alpes
université PARIS-SACLAY
anr agence nationale de la recherche