ENSTA BRETAGNE

Lab-STICC UMR 6285

Emerged and emerging NVM: technologies & challenges
SUPSEC workshop - Lannion

Jalil Boukhobza,
ENSTA-Bretagne, Lab-STICC UMR 6285

Who am I ?

https://www.ensta-bretagne.fr/boukhobza/
jalil.boukhobza@ensta-bretagne.fr

- **Education**
  - 1999 – Engineer in Electronics, INELEC, Algeria
  - 2000 – Master (DEA) in computer science, Univ. Versailles
  - 2004 – PhD Univ. Versailles, PRiSM Lab., Storage Systems
- **Prof. Exp.**
  - 2004-2006 – Research and teaching assistant, Univ. Versailles
  - 2006-2020 – Associate Professor, Univ. Bretagne Occidentale
  - 2013-* – Part time researcher at IRT b<>com, Rennes
  - 2016 Invited researcher, Hong Kong Polytechnic University
  - 2020-* Professor, ENSTA-Bretagne / team leader of SHAKER (Software/HArdware and unKnown Environment inteRactions)

- **Research topics:**
  - Storage and memory systems
    - Modeling / benchmarking / data placement / I/O optimization / I/O tracing
  - Domains: Cloud and Fog resource management, Embedded systems, HPC, DB
  - Current projects:
    - CEA: DataMeSS, data placement in multi-tiered storage systems
    - Atos: Energy I/O optimization for HPC with frugal and federated learning
    - NIST: cache optimization for NDN networks
    - AID project: DISPEED Intrusion Detection and Security/Performance/Energy tradeoff: a Study for Drone Swarms, with UBO, NAval Group, ICS FORTH, Scientific coordinato
    - IRT b<>com: service scheduling in heterogeneous systems (FPGA, GPU, GPP)

Disclaimer

- Several inventions, designs, implementations have been omited for time reasons
- Several new technologies/advances (2021-2022) have been skipped
- The focus of this presentation is not about computing / processing but about storage and memory
- Simplification = loss of information

- My vision of things is necessariliy biased
  - Computer scientist
  - Working on system research
  - Working in academia
  - not a « security » guy ☺

- Assumption about the audience:
  - Mainly computer scientists
  - Unaware of storage systems intricacies

- 2 types of slides
  - Normal slides
  - Quick slides →

Presentation outline

- Context
- Emerged NVM: flash memory
  - Basics & characteristics
  - Support
  - Integration and performance figures
  - Some contributions
- Emerging NVM
  - Definition & motivation
  - PCM
  - ReRAM
  - STT-RAM
- Conclusion

jalil.boukhobza@ensta-bretagne.fr

Presentation outline

- Context
- Emerged NVM: flash memory
  - Basics & characteristics
  - Support
  - Integration and performance figures
  - Some contributions
- Emerging NVM
  - Definition & motivation
  - PCM
  - ReRAM
  - STT-RAM
- Conclusion

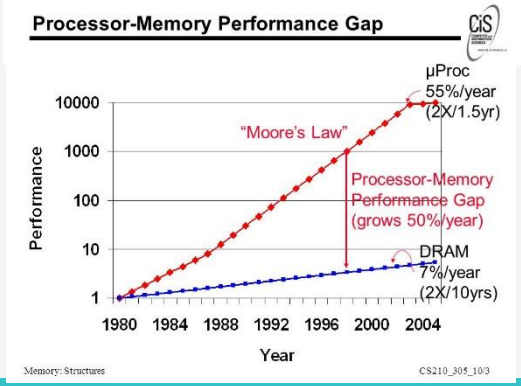
« It's the Memory, Stupid! »

A paper written by Richard Sites (1996), *lead designer* at DEC

"Across the industry, today's chips are largely able to execute code faster than we can feed them with instructions and data. There are no longer performance bottlenecks in the floating-point multiplier or in having only a single integer unit. The real design action is in memory subsystems— caches, buses, bandwidth, and latency."

"Over the coming decade, memory subsystem design will be the only important design issue for microprocessors."

– Richard Sites, after his article "It's The Memory, Stupid!", Microprocessor Report, 10(10),1996



Source: Spidey ed. Lug

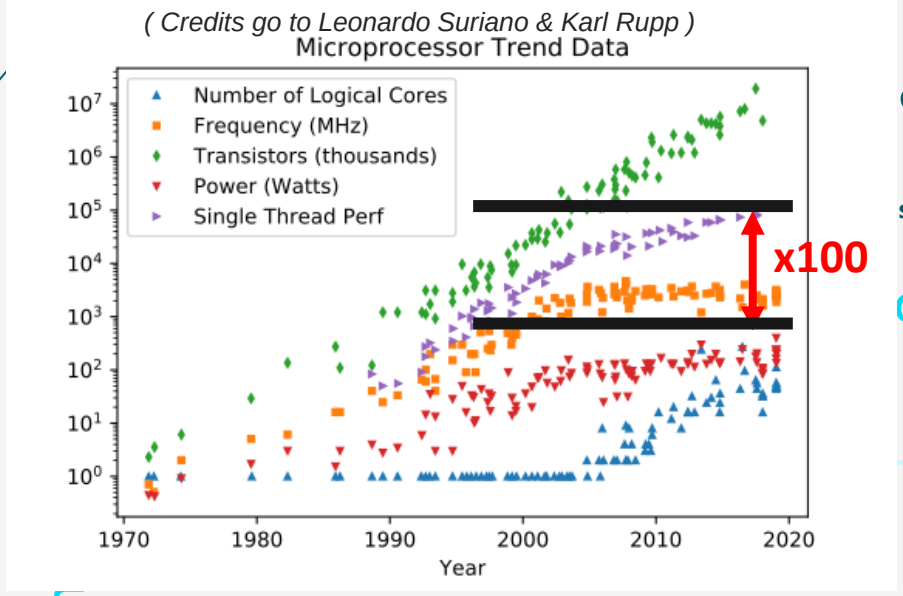
$$t_{avg} = p \times t_c + (1-p) \times t_m$$ *Wulf_94*

Now, although $(1-p)$ is small, it isn't zero. Therefore as t_c and t_m diverge, t_{avg} will grow and system performance will degrade. In fact, it will hit a wall.

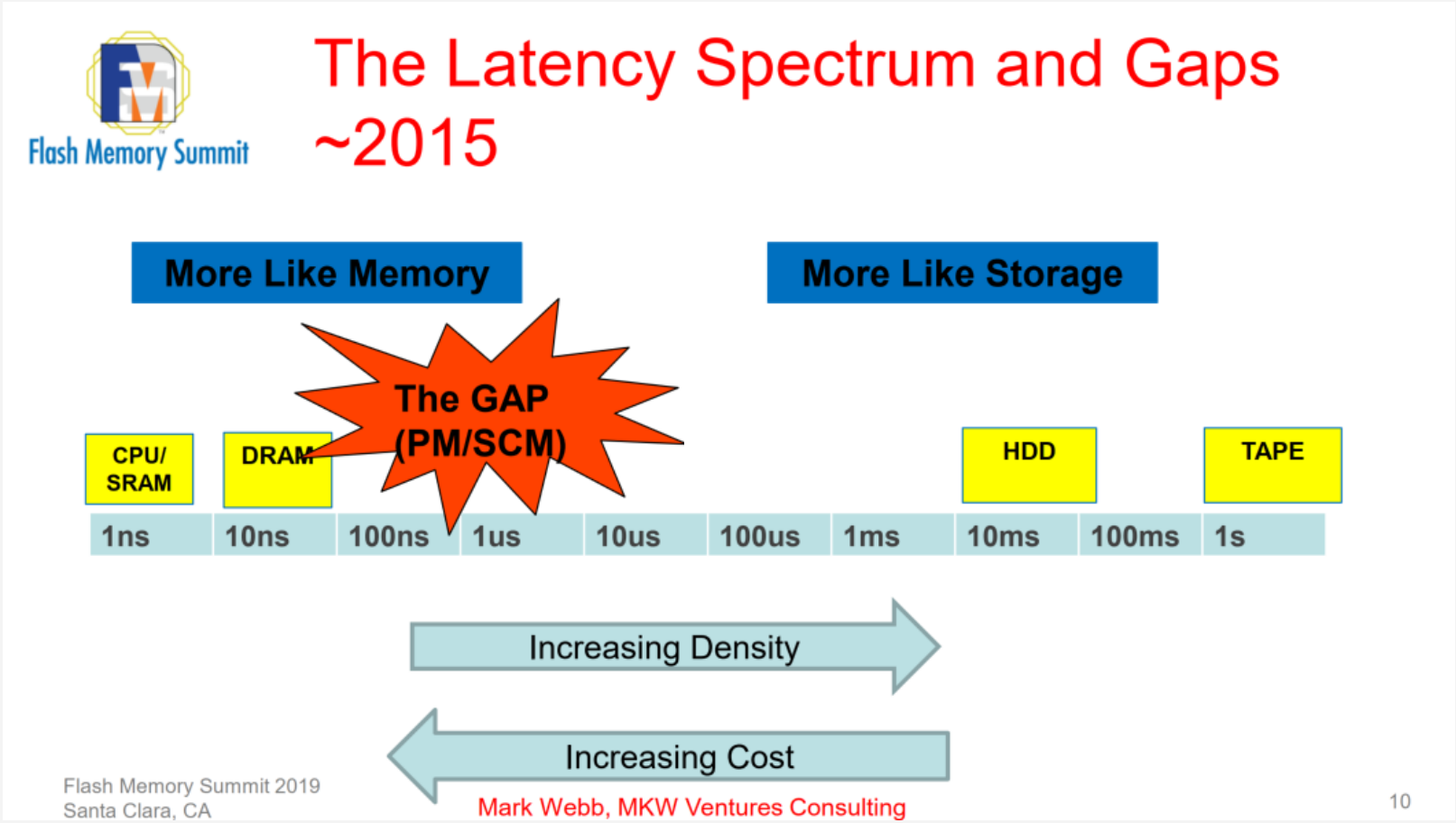
jalil.boukhobza@ensta-bretagne.fr

& Latency

schedule

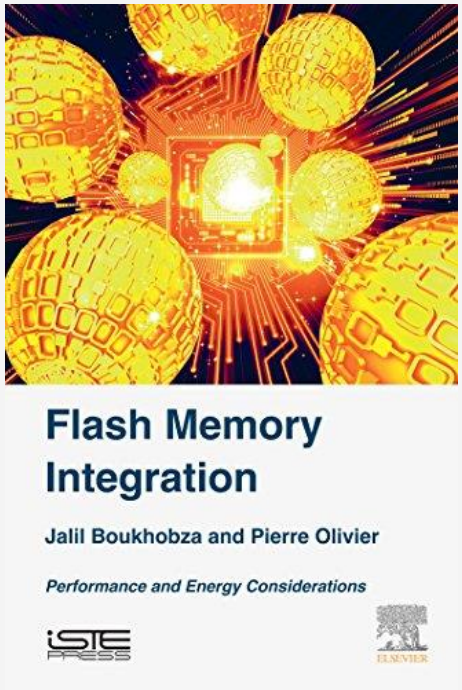


The gap with storage is even worse…

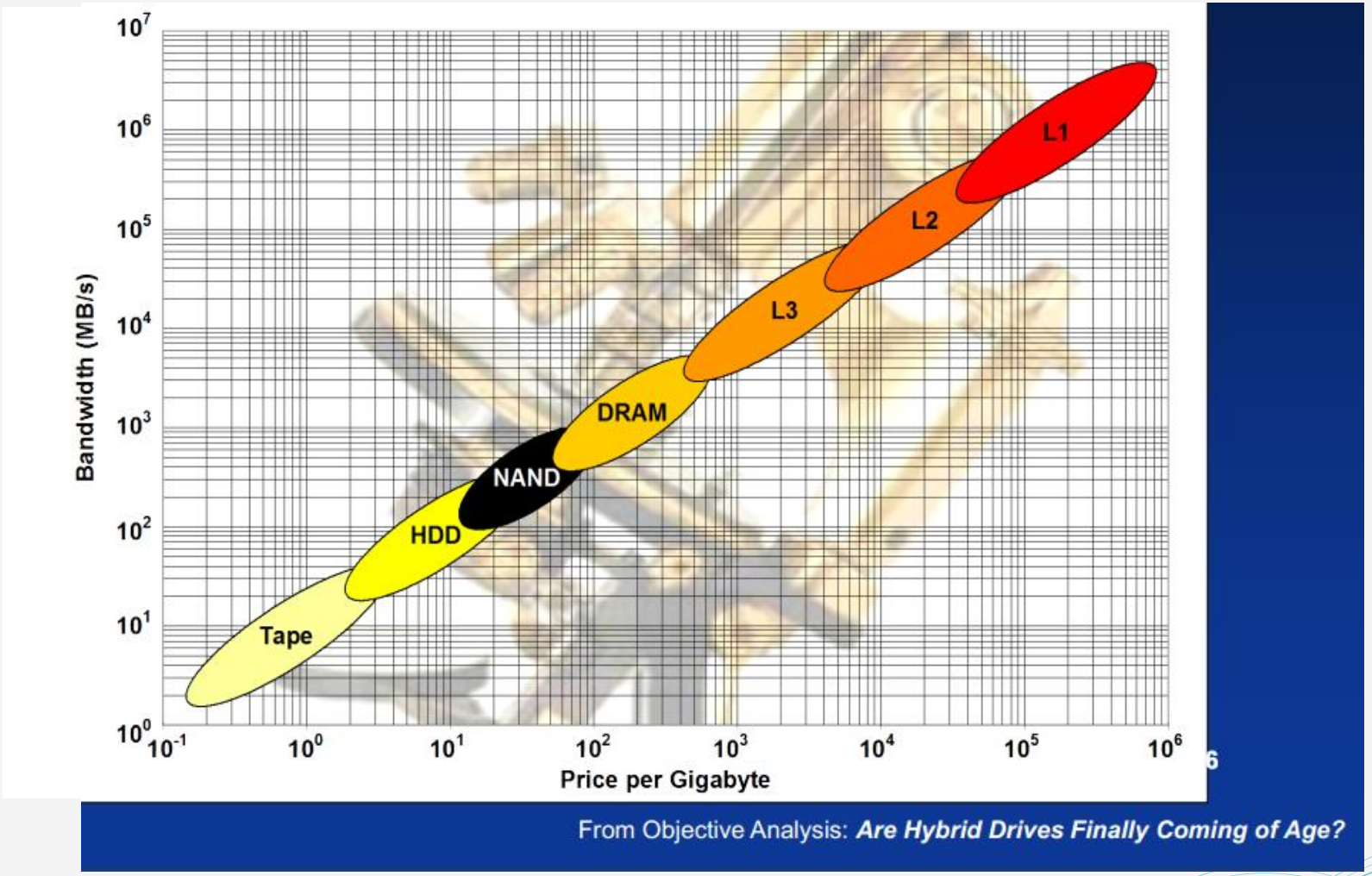


Presentation outline

- Context
- Emerged NVM: flash memory
  - Basics & characteristics
  - Support
  - Integration and performance figures
  - Some contributions
- Emerging NVM
  - Definition & motivation
  - PCM
  - ReRAM
  - STT-RAM
- Conclusion

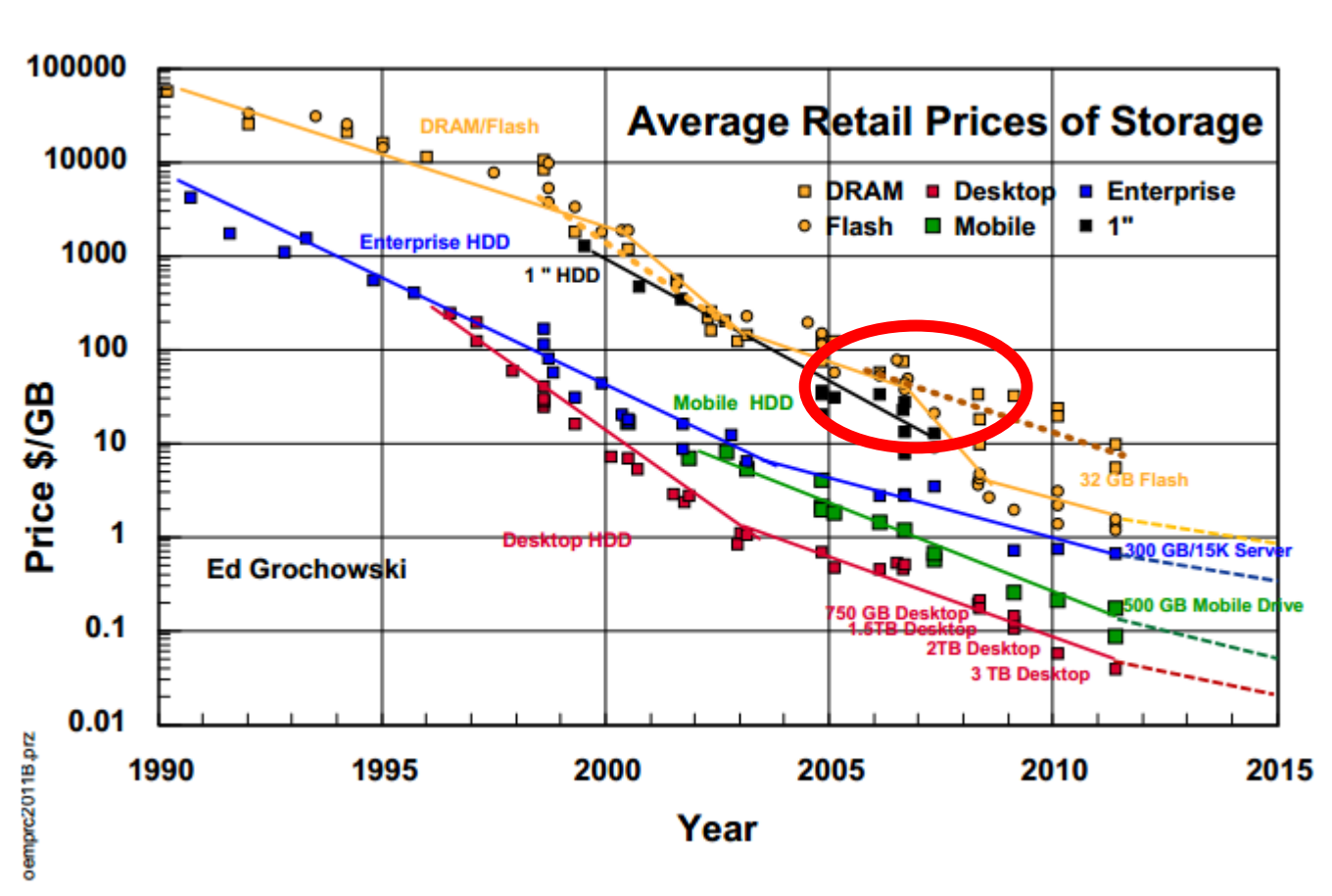


Context – ideal (memory) world



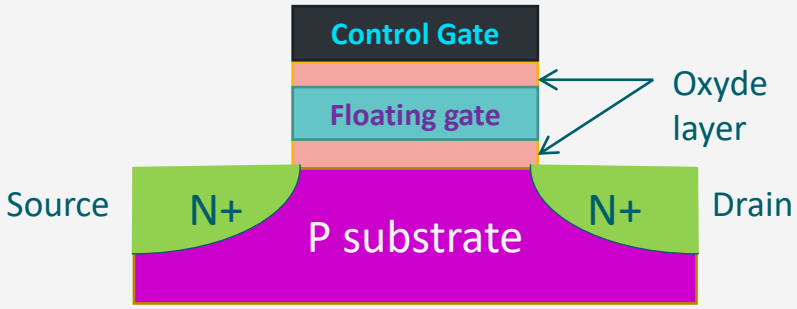
jalil.boukhobza@ensta-bretagne.fr

Price of memory



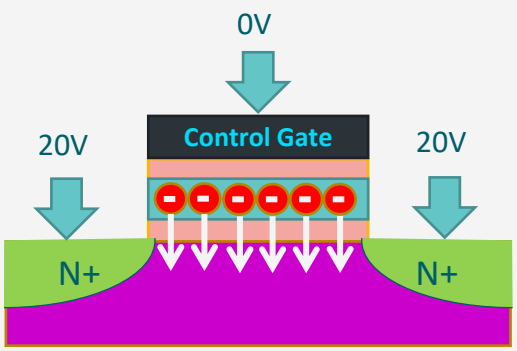
Flash memory cells

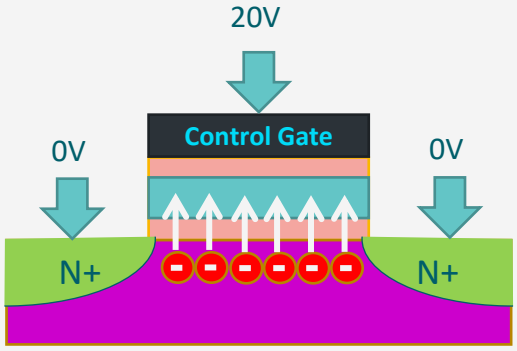
- Invented by F. Masuoka Toshiba 1980
- Introduced by Intel in 1988
- Type of EEPROM (Electrically Erasable & Programmable Read Only Memory)
- Use of Floating gate transistors
- Electrons pushed in the floating gate are trapped

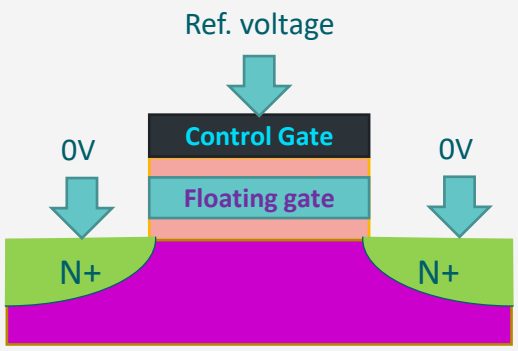


- 3 operations: program (write), erase, and read

Flash memory operations







Erase operation

- FN (Fowler-Nordheim) tunneling: Apply high voltage to substrate (compared to the operating voltage of the chip - usually between 7– 20V)
- → electrons off the floating gate
- Logic « 1 » in SLC

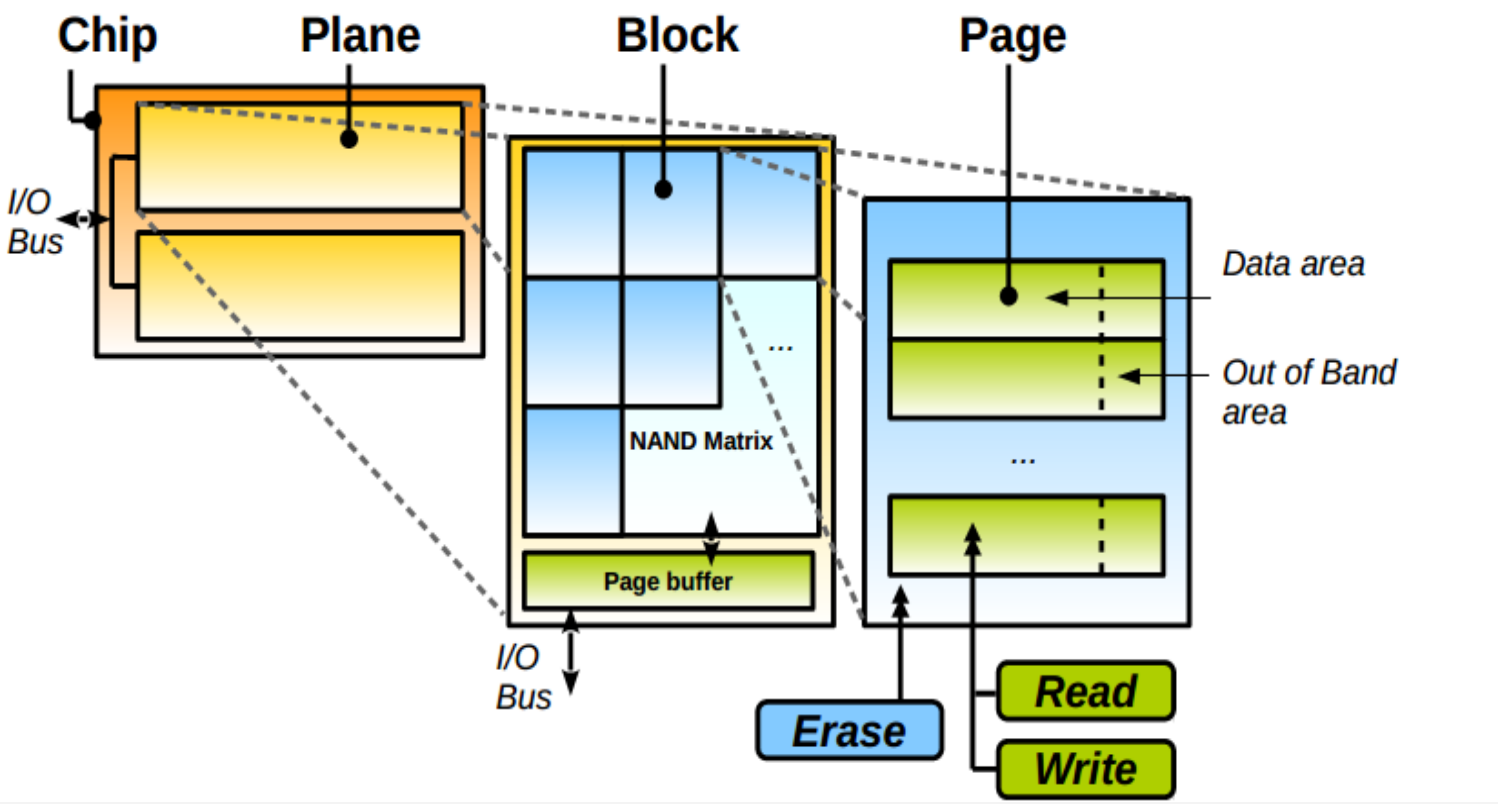
Program / write operation

- Apply high voltage to the control gate
- → electrons get trapped into the floating gate
- Logic « 0 »

Read operation

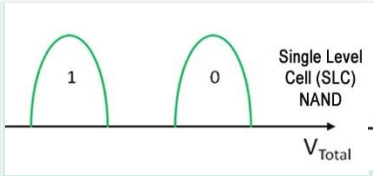
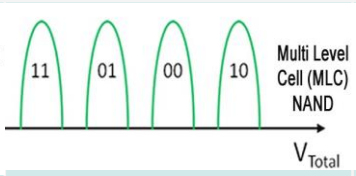
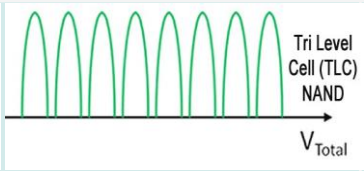
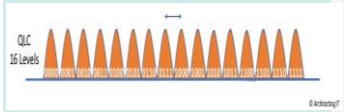
- Apply reference voltage to the control gate:
  - If floating gate charged: no current flow
  - If not charged; current flow

NAND flash memory architecture



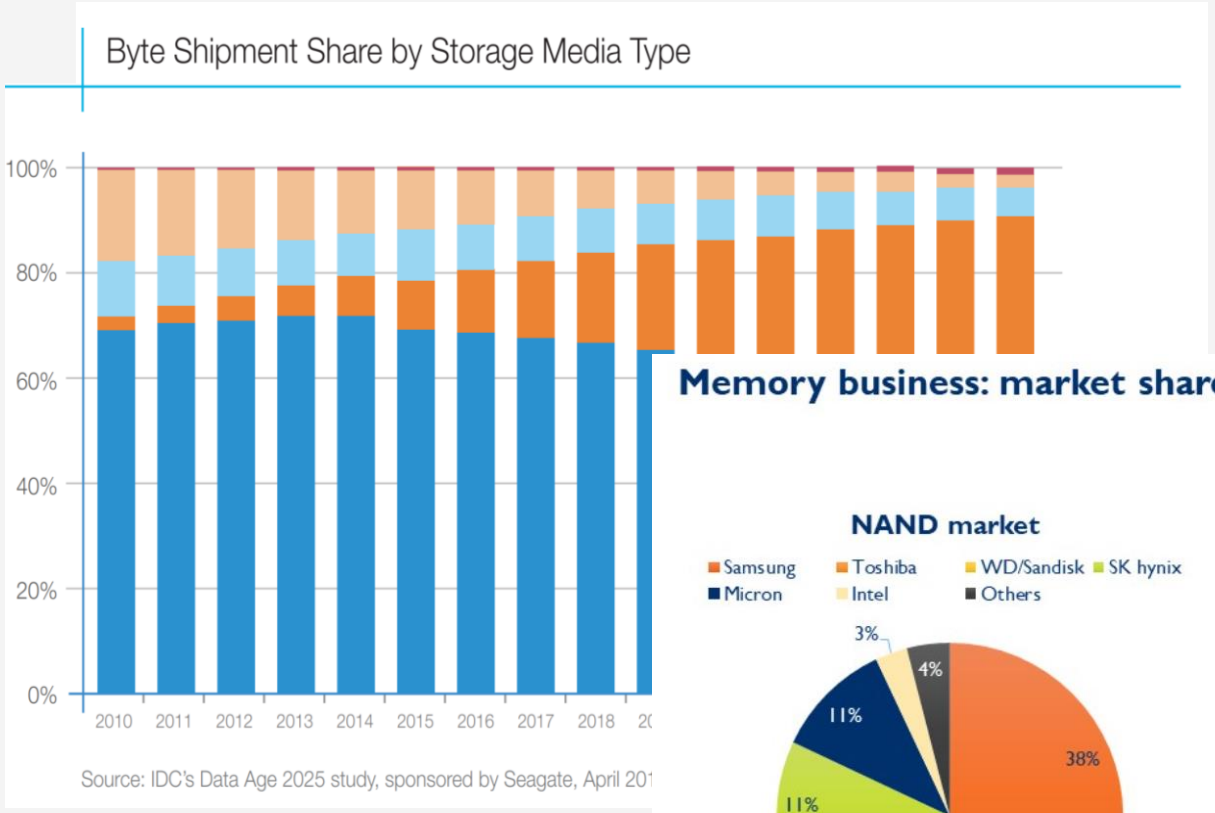
- Read/Write → page
- Erasures → blocks
- Page: 2-8KB
- Block: 128-4096 KB

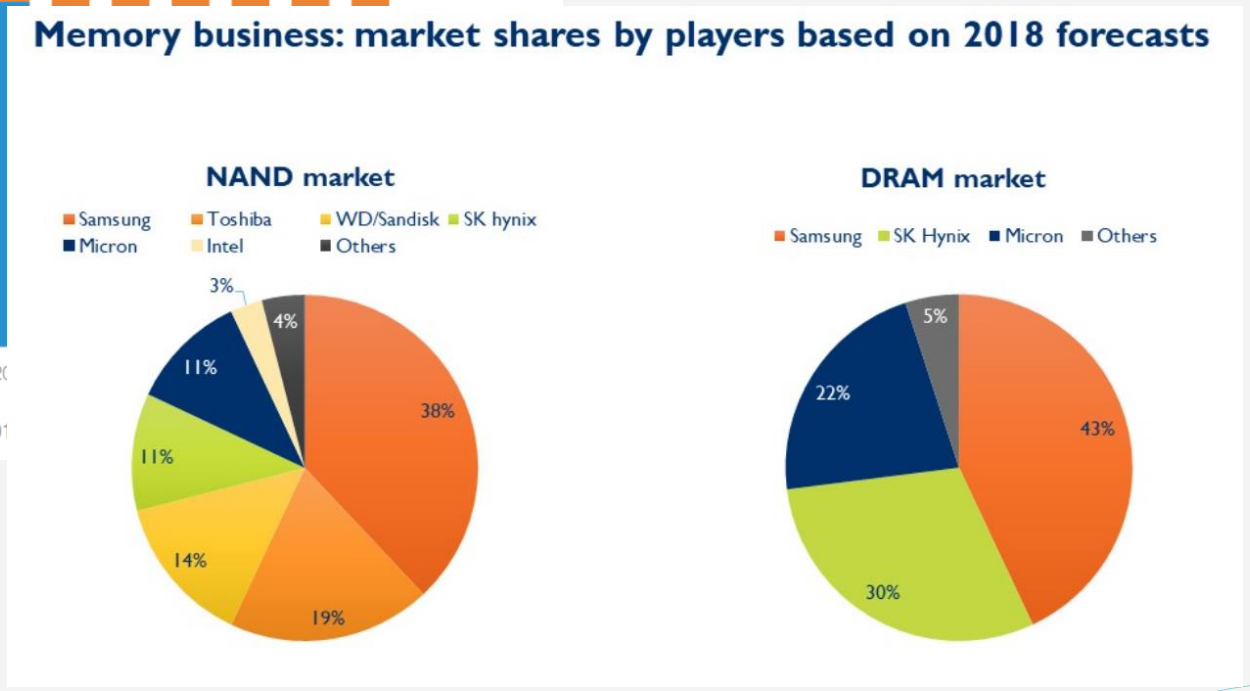
Different densities: SLC, MLC, TLC, QLC

	SLC (Single Level Cell)	MLC (Multi Level Cell)	TLC (Tri Level Cell)	QLC (Quad Level Cell)
				
Storage	1 bit / cell	2 bits / cell	3 bits /cell	4 bits/cell
Performance	++++	+++	++	+
Density	+	++	+++	++++
Lifetime (P/E cycles)	~ 100 000	~ 10 000	~5 000	~1000
ECC complexity	+	++	+++	++++
Applications	Embedded and industrial applications (high end SSDs…)	Most consumer applications (e.g. memory cards)	Low-end consumer applications not needing data updates (e.g. mobile GPS)	Write once, read many

jalil.boukhobza@ensta-bretagne.fr

Byte shipment share





jalil.boukhobza@ensta-bretagne.fr

Presentation outline

- Context
- Emerged NVM: flash memory
  - Basics & characteristics
  - Support
  - Integration and performance figures
  - Some contributions
- Emerging NVM
  - Definition & motivation
  - PCM
  - ReRAM
  - STT-RAM
- Conclusion

Flash memory constraints

Write/Erase granumarity asymmetry (**Cons1**)

Erase-before-write rule (**Cons2**)

Limited cell lifetime (**Cons3**)

+

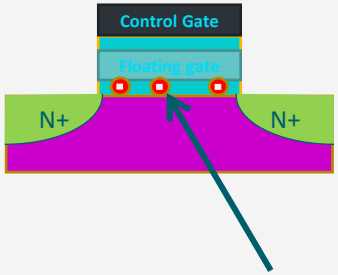
Data update:
- Invalidate
- Out-of-place update

+

Logical to physical mapping

Garbage Collection

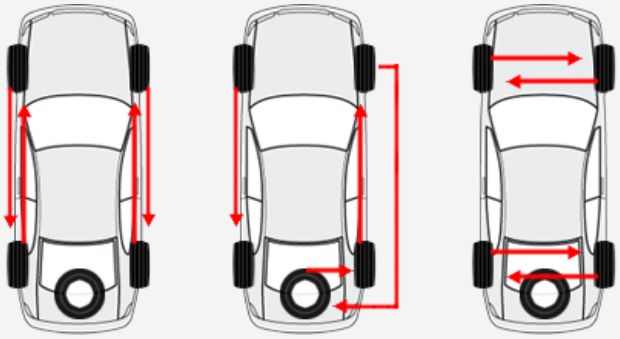
Wear leveling



- Electrons get trapped in the oxide layer deteriorating its characteristics
- Electrons cannot move from oxyde layer to floating gate

Wear leveling

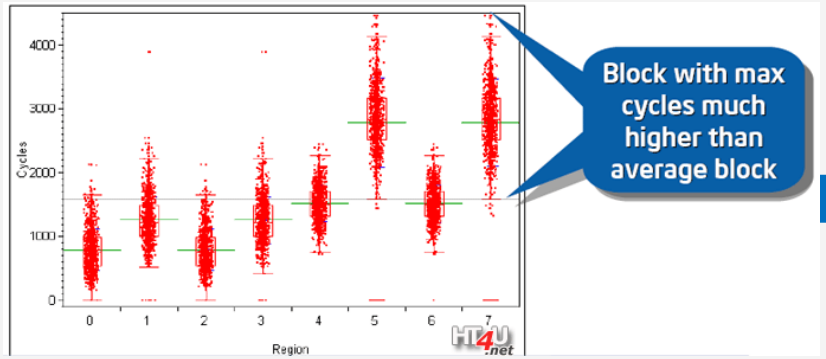
/ You already do that with your tyres …

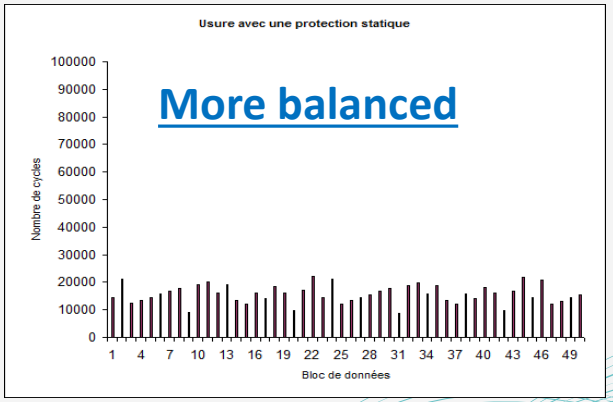


Pierelli
Courtesy

/ Keeping a balanced erasures' distribution over flash memory blocks.

http://www.presence-pc.com/tests/ssd-flash-disques-22675/5/





jalil.boukhobza@ensta-bretagne.fr

Garbage Collection

Inv. Op.

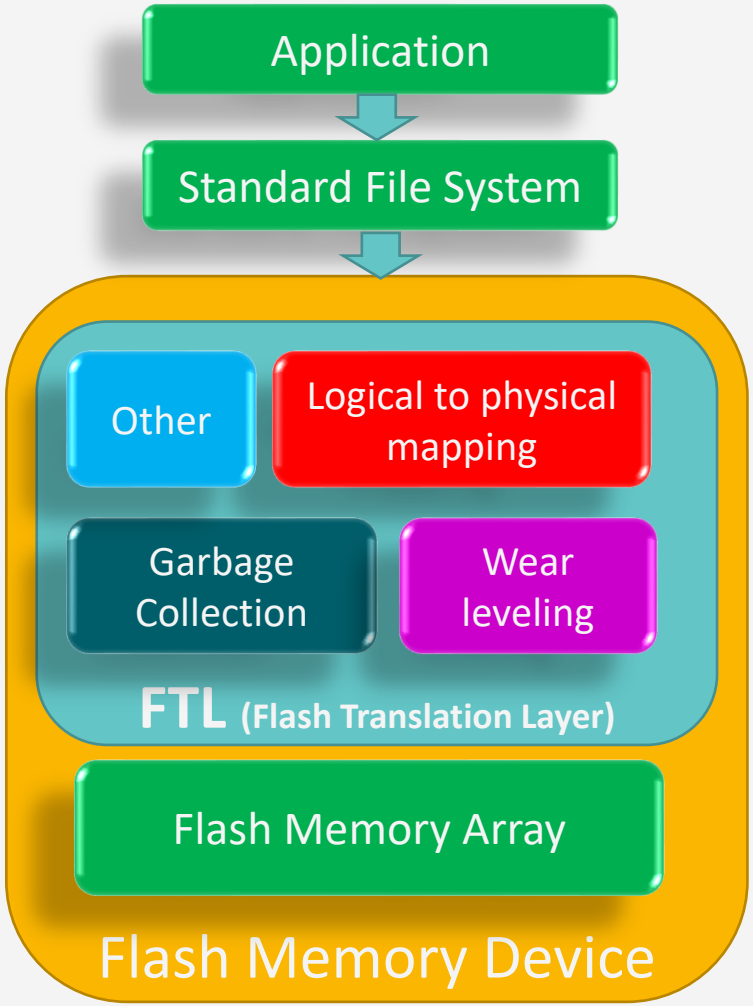
Moving people to a new city and « erasing » the old one to reuse the space !!!

- Moving valid pages from blocks containing invalid data and then erase/recycle the blocks

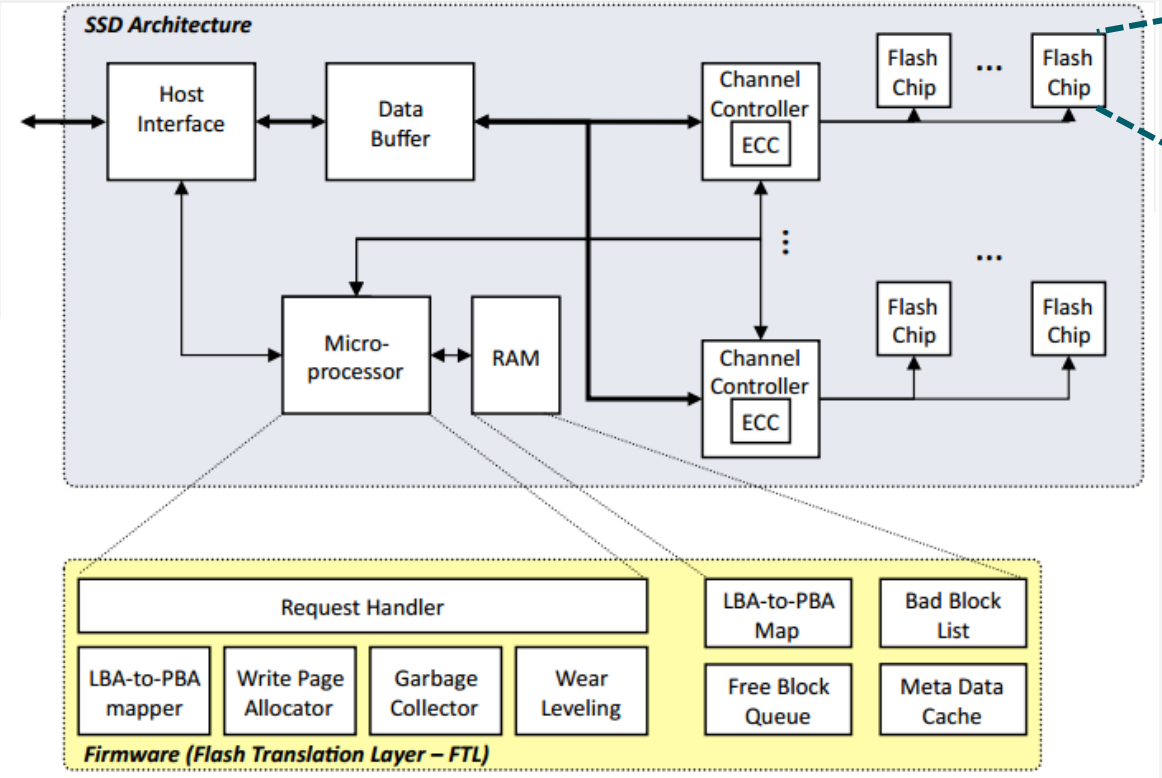
Presentation outline

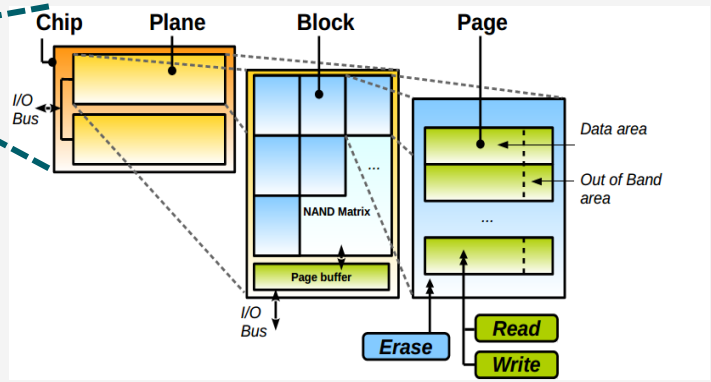
- Context
- Emerged NVM: flash memory
  - Basics & characteristics
  - Support
  - Integration and performance figures
  - Some contributions
- Emerging NVM
  - Definition & motivation
  - PCM
  - ReRAM
  - STT-RAM
- Conclusion

Flash memory structure



SSD architecture

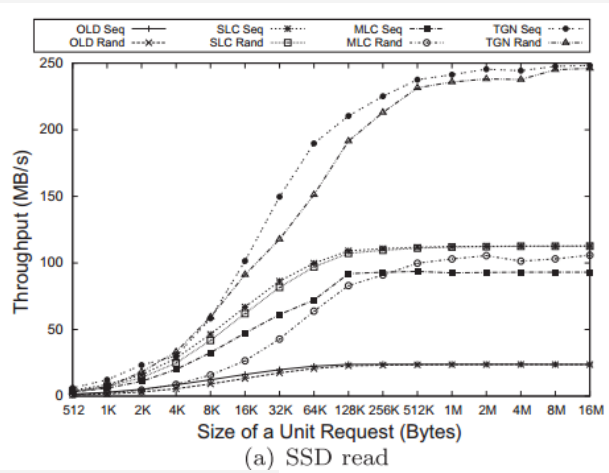




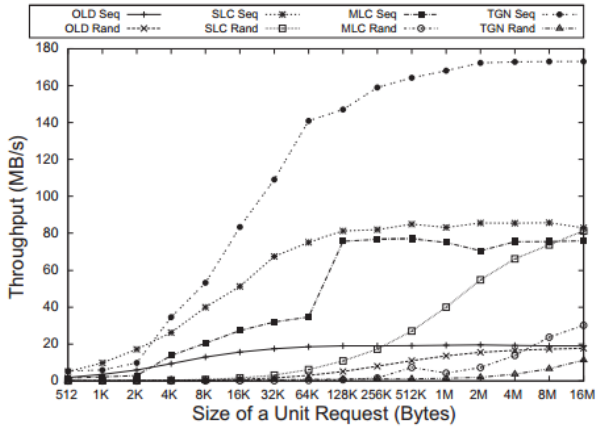
Source: Bonnet, Bouganim, Koltsidas, Viglas, VLDB 2011

Read/write asymmetry and disparity

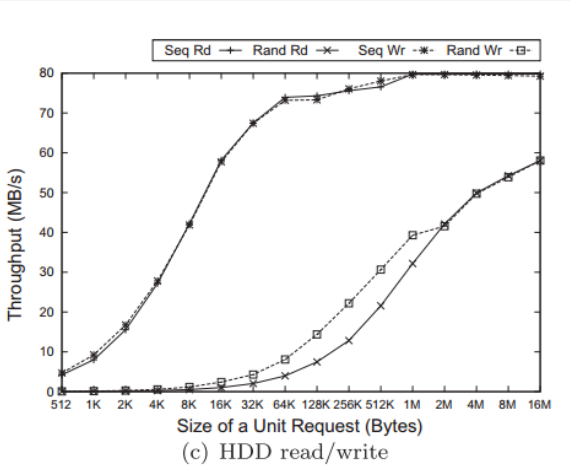
- **Flash disk performance is heterogeneous**
  - Depend on internal structure and workload
  - Performance disparities between SSDs from the same constructor and between different technologies are significantly high



(a) SSD read



(b) SSD write



(c) HDD read/write

Table 3
Specifications of the storage devices used in our work.

	OLD	MLC	SLC	TGN	HDD
Model	FSD32GB25M	1C32G	MSP7000	MMCRE28G5	WD1600BEKT
Vendor	Super talent	OCZ	MTron	Samsung	Western digital
Form factor	2.5 in.	2.5 in.	2.5 in.	2.5 in.	2.5 in.
Flash type/RPM	SLC	MLC	SLC	MLC	7200
Capacity	32 GB	32 GB	16 GB	128 GB	160 GB
Rd./Wr. Perf. (MB/s)	60/45	143/93	120/90	220/200	NA/NA

Performance of write operations

Flash performance needs time to reach steady state… and may oscillate between states …

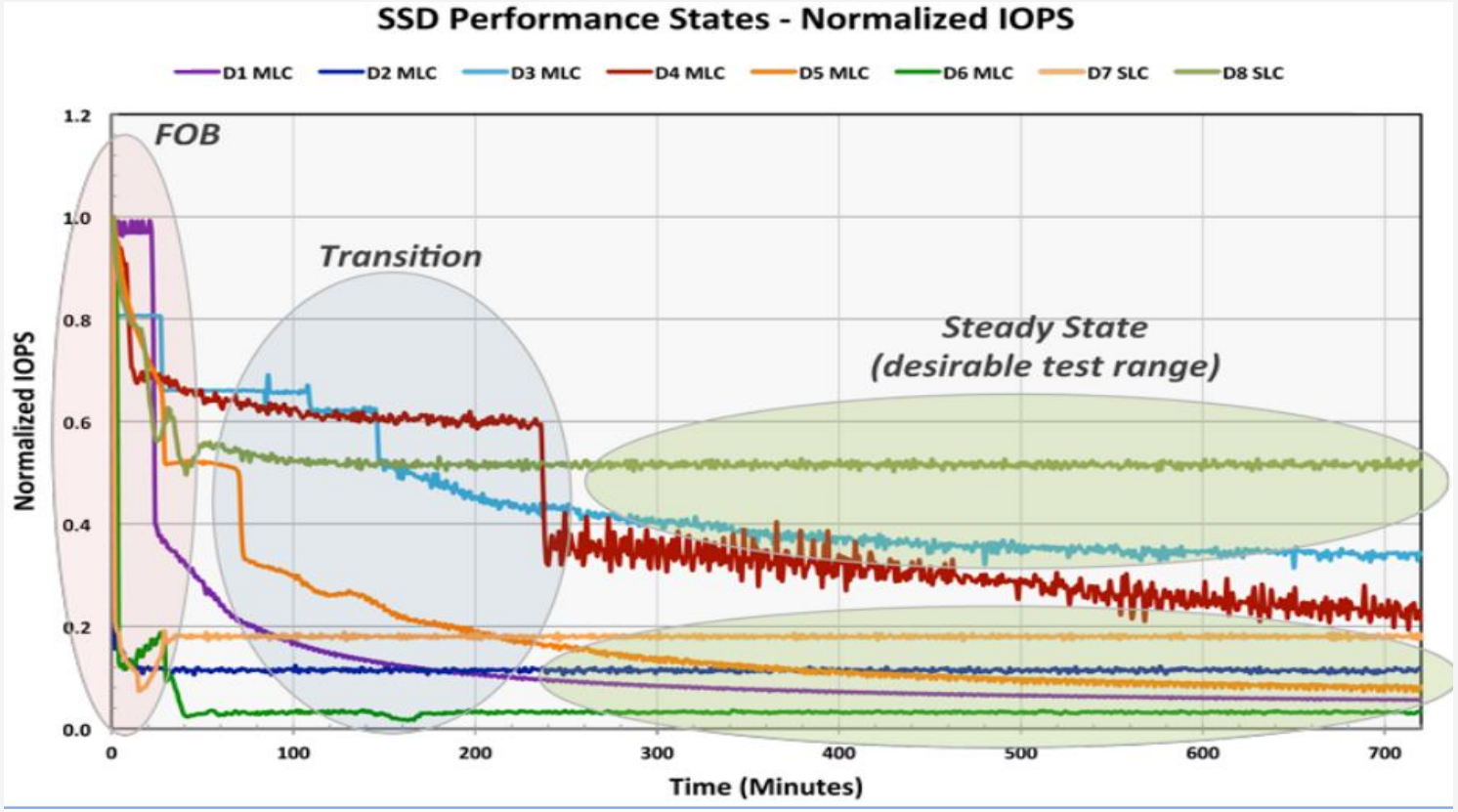


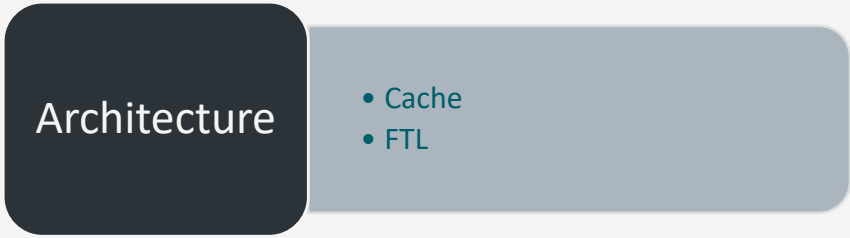
Figure 1-1 – NAND-based SSS Performance States (RND 4KiB Writes)

Source: http://snia.org/sites/default/files/SSS%20PTS%20Client%20-%20v1.1.pdf

Summary

As compared to traditional drives

- New constraints
  - Write/Erase granularity
  - Erase before write
  - Endurance
- New performance model
  - ~ Symmetric sequential / random read
  - More sensitive to writes (lifetime)
    - Asymmetric sequential / random writes
  - Sensitive to the fill rate

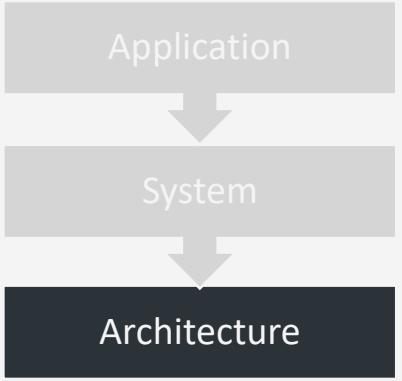


Presentation outline

- Context
- Emerged NVM: flash memory
  - Basics & characteristics
  - Support
  - Integration and performance figures
  - Some contributions
- Emerging NVM
  - Definition & motivation
  - PCM
  - ReRAM
  - STT-RAM
- Conclusion

Flash memory integration: architecture level

- Contributors: P. Olivier (UBO), S. Rubini (UBO), Y. Hadjadj Aoul (Univ. Rennes 1), L. Lemarchand (UBO)
- Motivation
  - FTL services (mapping) → reduce the write traffic [Chung09]
  - Cache → also reduces the write traffic [Wang13][Huang12][Kim08]
  - Most FTL and cache were built independently [Liao11]
- Problem statement
  - How to contribute in bridging the gap between cache and FTL for SSD ?
- Contribution (2011-2014)
  - C-lash*: a cache for flash to replace FTL services
  - CACH-FTL**: a Cache Aware Configurable Hybrid FTL
  - MaCACH***: an adaptive version of CACH-FTL



Lab-STICC UBO Université de Bretagne Occidentale

UMR IRISA UNIVERSITÉ DE RENNES 1

*J. Boukhobza, P. Olivier, S. Rubini, "**A cache management strategy to replace wear leveling techniques for embedded flash memory**", in proceedings of the International Symposium on Performance Evaluation of Computer and Telecommunication Systems (**SPECTS**), pp. 1-8, IEEE, SCS, The Hague, Jun. 2011

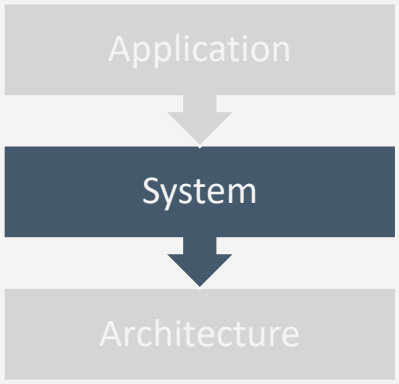
J. Boukhobza, P. Olivier, S. Rubini, "CACH-FTL: a cache aware configurable hybrid flash translation layer**", in proceedings of the EUROMICRO International conference on Parallel, Distributed, and Network based processing (**EUROMICRO PDP**), pp. 94-101, Belfast, Feb. 2013.

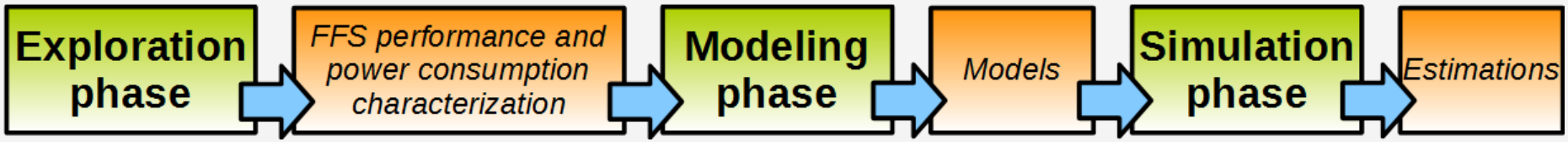
***J. Boukhobza, P. Olivier, S. Rubini, L. Lemarchand, Y.Hadjadj-Aoul, A. Laga, "**MaCACH: An adaptive cache-aware hybrid FTL mapping scheme using feedback control for efficient page-mapped space management**", **Journal of Systems Architecture**, Elsevier, Volume 61, Issues 3-4, pp.157-171, Mar. 2015.

jalil.boukhobza@ensta-bretagne.fr

Flash memory integration: system level

- Contributors: E. Senn (UBS)
- Context
  - PhD thesis of P. Olivier (supervised with E. Senn)
  - Support of the ANR project Open-PEOPLE (2011-2013)
- Problem statement
  - How are performance and energy consumption metrics impacted in embedded storage systems ?
    - For FTL [Bjørling10]
    - For FFS → none
- Contribution:
  - A methodology to estimate FFS storage performance/power consumption

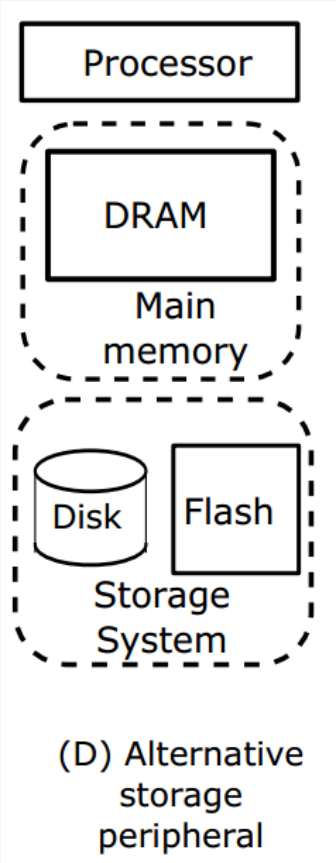




P. Olivier, J. Boukhobza, E. Senn, H. Ouarnoughi, "**A Methodology for Estimating Performance and Power Consumption of Embedded Flash File Systems**", in **ACM Transactions on Embedded Computing Systems**, 15(4): 79:1-79:25, 2016.

Hybrid storage for the Cloud

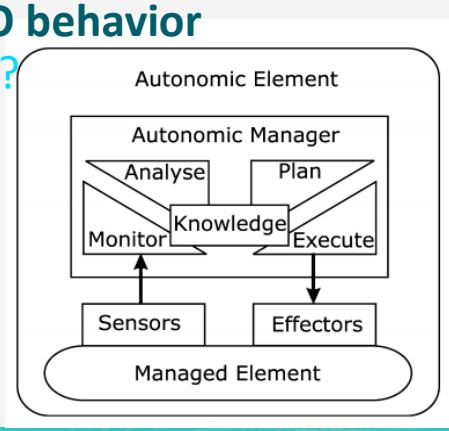
/ Many flash memory integration options:



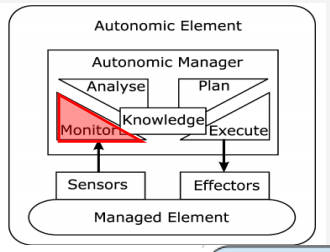
J. Boukhobza, "**Flashing in the Cloud: shedding some light on NAND flash memory storage systems**", in book Data Intensive Storage Services for Cloud Environments, , pp. 241-266, IGI Global Editor, ISBN13: 9781466639348, Apr. 2013

Hybrid storage for the Cloud

- Contributors: K. Boukhalfa (USTHB), S. Rubini, (UBO) F. Singhoff (UBO), L. Lemarchand (UBO)
- Context
  - PhD thesis of
    - H. Ouarnoughi, (supervised with F. Singhoff and S. Rubini),
    - D. Boukhelef (supervised with K. Boukhalfa, USTHB)
    - A. Chikhaoui (supervised with L. Lemarchand, and K. Boukhalfa)
  - Projects: IRT b<>com (2013-2016), PHC GHEEMaS (2016-2019)
- Problem statement
  - How to place data in a hybrid storage system taking into account **I/O behavior** [Zhang11], **Cloud context** [Le11] and **energy** [Kansal10][Colmant15]?
    - IaaS service
    - DBaaS service
- Contribution
  - MAPE-K [IBM01] loop for data placement



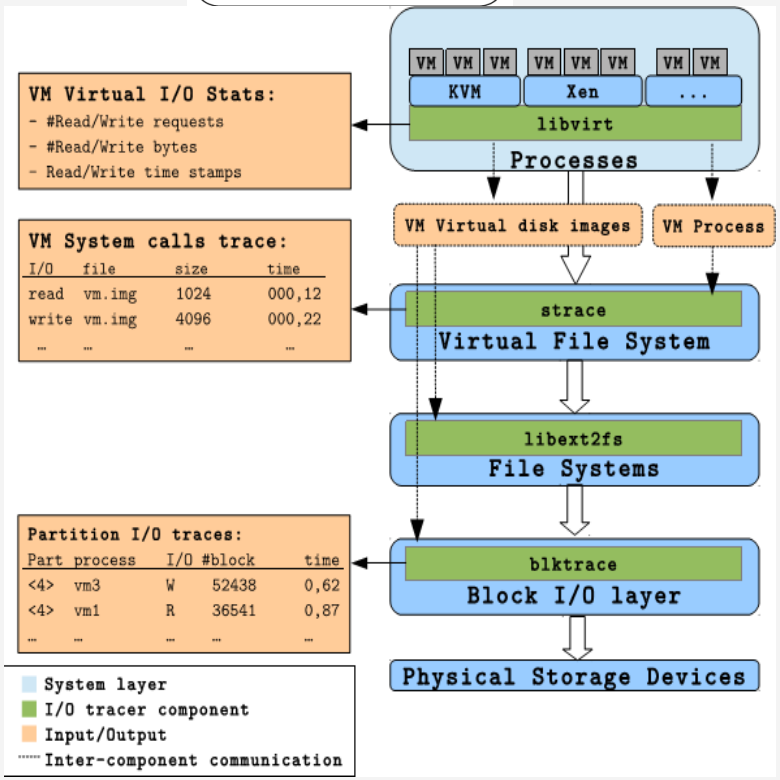
The « Monitor » step



- Supervise at all levels to understand VMs' I/O behavior:
  - Hypervisor level
  - Host level
  - File system level
  - I/O block level

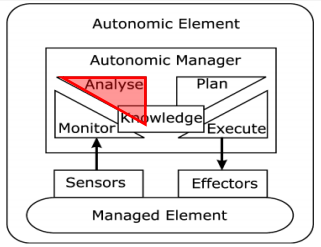
→Continued with FUI IDIOM (Integrated Device I/O Monitor): DDN, UBO, TSP, INRIA, CEA, Qarnot, Criteo, QuarsarDB

https://github.com/medislam/IOTracer

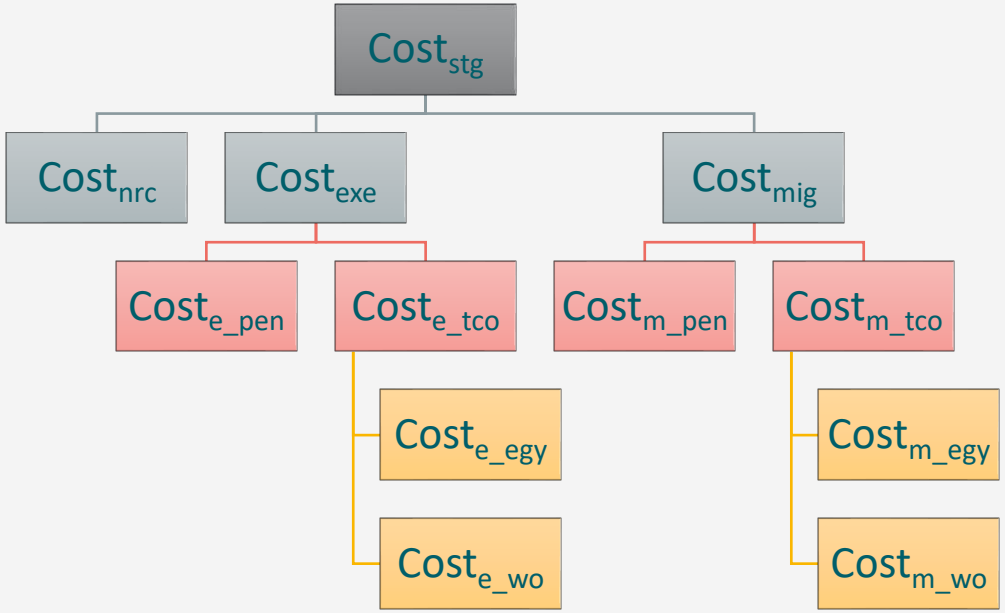


- H. Ouarnoughi, J. Boukhobza, F. Singhoff, and S. Rubini, "**A multi-level I/O tracer for timing and performance storage systems in IaaS Cloud, Real-time and distributed computing in emerging applications**", **IEEE REACTION**, pp.1-8, Rome, Dec. 2014.
- M. I. Naas, F. Trahay, A. Colin, P. Olivier, S. Rubini, F. Singhoff, J. Boukhobza, , **EZIOTracer: Unifying Kernel and User Space I/O Tracing for Data-Intensive Applications. ACM SIGOPS Oper. Syst. Rev**. 55(1): 88-98 (2021)

The « Analyze » step

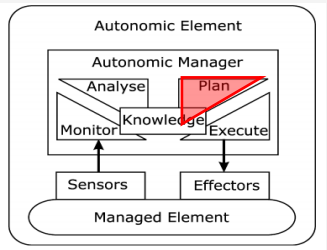


- Evaluate the storage cost for VMs
  - Execution/migration, energy/performance, wear out, I/O patterns



- H. Ouarnoughi, J. Boukhobza, F. Singhoff, S. Rubini, "**A Cost Model for Virtual Machine Storage in Cloud IaaS**", in proceedings of the EUROMICRO International conference on Parallel, Distributed, and Network based processing (**EUROMICRO PDP**), pp. 664-671, Heraklion, Feb. 2016.
- D. Boukhelef, J. Boukhobza, K. Boukhalfa, "**A Cost Model for DBaaS Storage**", 27th International Conference on Database and Expert Systems Applications (**DEXA**), pp. 223-240, Porto, Sep. 2016.
- Amina Chikhaoui, Kamel Boukhalfa, Jalil Boukhobza, **A Cost Model for Hybrid Storage Systems in a Cloud Federations**, Federated Conference on Computer Science and Information Systems (**FedCSIS**), Poznan, 2018

The « Plan » step

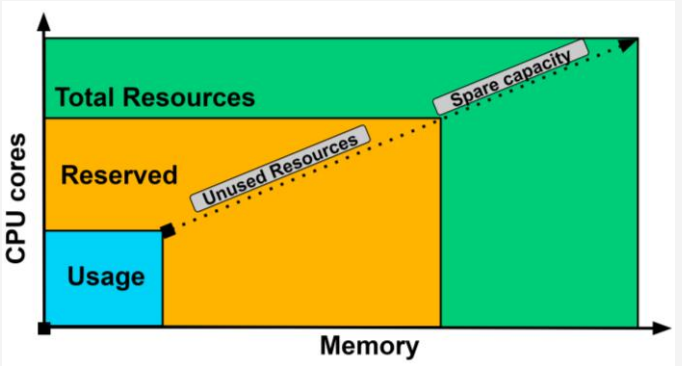


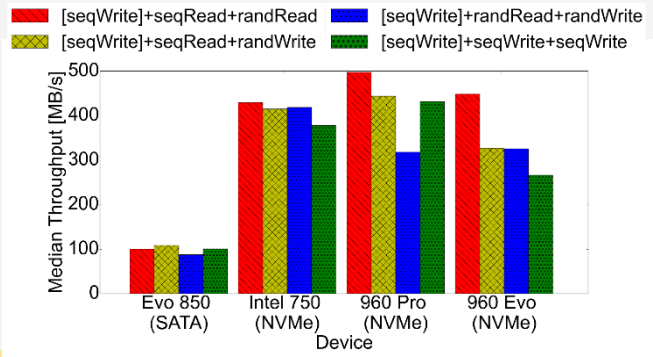
- Example: DBaaS context
  - How to place database objects in a hybrid storage system to minimize the overall cost and satisfying SLA.
- H-COPS: a heuristic cost based object placement strategy
  - Initialization
    - Place objects in the cheapest storage system
  - Feasibility
    - Storage constraint satisfaction
      - Move objects from overfilled devices: $\delta(o_{i,u_k}) = \frac{s_{o_{i,u_k}}}{d_{costly}(o_{i,u_k}) - d_{cheap}(o_{i,u_k})}$
    - SLA constraint
      - Move objects from HDD to SSD to guarantee hard SLA: $\lambda(o_{i,u_k}) = \frac{t_{exec}(o_{i,u_k}, d_{costly}) - t_{exec}(o_{i,u_k}, d_{cheap})}{d_{costly}(o_{i,u_k}) - d_{cheap}(o_{i,u_k})}$
  - Optimization
    - Tradeoffs between storage cost and penalties to guarantee soft SLA: $\mu(u_k) = Cost_{pnl}(u_k) - Cost_{SLA}(u_k)$

- D. Boukhelef, K. Boukhalfa, J. Boukhobza, H. Ouarnoughi, L. Lemarchand, "**COPS: Cost Based Object Placement Strategies on Hybrid Storage System for DBaaS Cloud**", The 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (**IEEE/ACM CCGRID**), May 2017.
- D. Boukhelef, J. Boukhobza, K. Boukhalfa, H. Ouarnoughi, L. Lemarchand, "**Optimizing the cost of DBaaS object placement in hybrid storage systems**", **Future Generation Computer Systems**, Elsevier, Volume 93, 2019, Pages 176-187 ,
- Amina Chikhaoui, Laurent Lemarchand, Kamel Boukhalfa, Jalil Boukhobza, **StorNIR, a Multi-Objective Replica Placement Strategy for Cloud Federations**, in Proceedings of the ACM SIGAP Symposium of Applied Computing (**ACM SAC**), 2021
- Amina Chikhaoui, Laurent Lemarchand, Kamel Boukhalfa, Jalil Boukhobza, **Multi-objective Optimization of Data Placement in a Storage-as-a-Service Federated Cloud**. **ACM Trans. Storage** 17(3): 22:1-22:32 (2021)

Exploiting Cloud spare ephemeral resources

- Contributors: O. Barais and L. D'Orazio (Univ. Rennes 1), A. Knefati (IRT), H. Ribeiro (IRT), L. Lemarchand (UBO)
- Context
  - PhD thesis of:
    - J.E. Dartois (supervised with O. Barais),
    - M. Handaoui (supervised with O. Barais and L. D'Orazio)
  - IRT b<>com project (2017-2020)
- Problem statement
  - How to efficiently use spare ephemeral Cloud resources while guaranteeing SLA (QoS) ?
- Contribution
  - Evaluate the resource capacity (the case of SSDs)* → ML algorithms
  - Modelling resource usage** → quantile regression + ML
  - Placing Big Data (MapReduce) jobs***

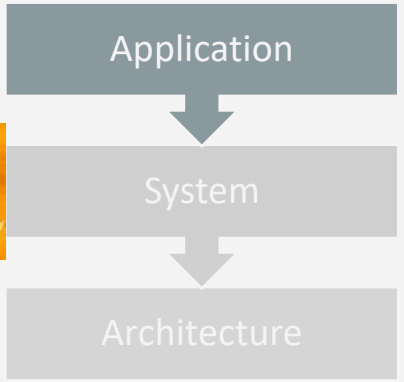




- *J-E. Dartois, J. Boukhobza, A. Knefati, O. Barais, Investigating, **Machine Learning Algorithms for Modeling SSD I/O Performance for Container-based Virtualization**, in **IEEE Transactions on Cloud Computing, 2019**
- **J-E. Dartois, A. Knefati, J. Boukhobza, O. Barais, **Using Quantile Regression for Reclaiming Unused Cloud Resources while achieving SLA**, in proceedings of the 10th IEEE International Conference on Cloud Computing Technology and Science (**IEEE CloudCom**), pp. 89-98, Nicosia, December 2018
- ***J-E. Dartois, H. Ribeiro, J. Boukhobza, O. Barais, **Cuckoo: a Mechanism for Exploiting Ephemeral and Heterogeneous Cloud Resources**, in proceedings of the IEEE International Conference on **Cloud** Computing (**IEEE CLOUD**), Milano, 2019.
- Mohamed Handaoui, Jean-Emile Dartois, Laurent Lemarchand, Jalil Boukhobza, **Salamander: a Holistic Scheduling of MapReduce Jobs on Ephemeral Cloud Resources**, In Proceedings of the 20th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (**IEEE/ACM CCGRID**), May 2020

Databases for flash memory

- Contributors: F. Singhoff (UBO), M. Koskas (KoDe)
- Context
  - PhD thesis of A. Laga (supervised with F. Singhoff)
  - Projects: CIFRE with KoDe Software (2015-2018)

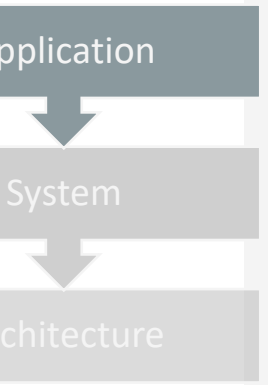


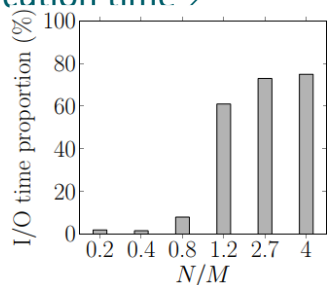
- Problem statement
  - How to take profit of SSD performance for database systems?
    - **Application**: some sorting algorithms for flash [Andreou09][Park09] → scalability issues
    - **System**: read-ahead based on sequential I/Os → more applications turn to random I/Os (SSD performance model)
- Contribution
  - **Application**: revisit sorting algorithms for SSD → MONTRES (Merge ON-The-Run External Sorting)*
  - **System**: revisit kernel (page cache) prefetching algorithms → Lynx**

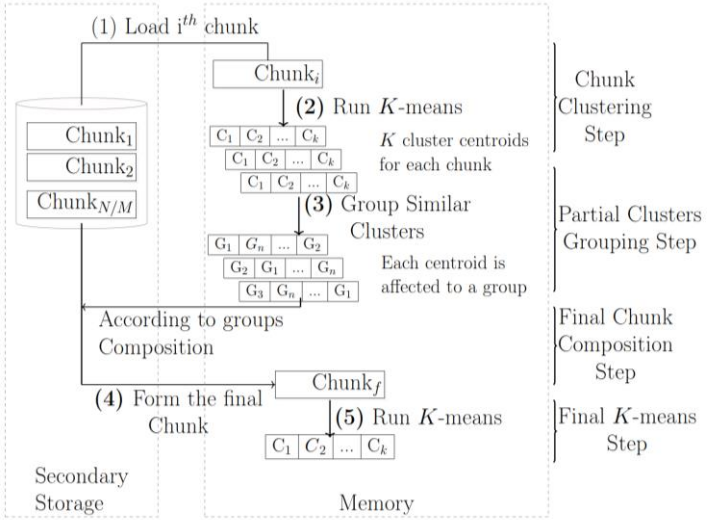
*A. Laga, J. Boukhobza, F. Singhoff, M. Koskas, "**MONTRES : Merge ON-The-Run External Sorting Algorithm For Large Data Volumes On SSD Based Storage Systems**", **IEEE Transactions on Computers**, 66(10), 1689-1702, 2017.
A. Laga, **J. Boukhobza, M. Koskas, F. Singhoff, "**Lynx: A Learning Linux Prefetching Mechanism For SSD Performance Model**", The 5th Non-Volatile Memory Systems and Applications Symposium (**IEEE NVMSA**), Daegu, Aug. 2016.

Optimizing IA applications (the case of K-means and Random forrests)

- **Contributors: S. Rubini (UBO), Y-H. Chang (A. Sinica)**
- **Context**
  - K-means, popular in embedded systems
  - Memory stress ↗ (dataset size N / memory workspace M) ⇒ execution time ↗
- **Problem statement**
  - How to avoid several spanning through all the data when the memory workspace is small
- **Contribution**
  - Divide and conquere
  - 1 K-means → several small (enough) K-means + merge results
- **Results:** 90% less I/Os 50% execution time reduction
  - … very simple solution







- C. Slimani, S. Rubini and J. Boukhobza. "**K-MLIO: Enabling K-Means for Large Data-sets and Memory Constrained Embedded Systems**", in Proc, of the IEEE International Symposium on the Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (**IEEE MASCOTS**), Rennes, October 2019
- C. Slimani, S. Rubini, C-F. Wu, Y-H. Chang, J. Boukhobza. "**RaFIO: A Random Forest I/O-Aware algorithm**", in Proc, of the ACM Symposium of Applied (**ACM SAC**), March 2021

Presentation outline

- Context
- Emerged NVM: flash memory
  - Basics & characteristics
  - Support
  - Integration and performance figures
  - Some contributions
- Emerging NVM
  - Definition & motivation
  - PCM
  - ReRAM
  - STT-RAM
- Conclusion

NVM definition

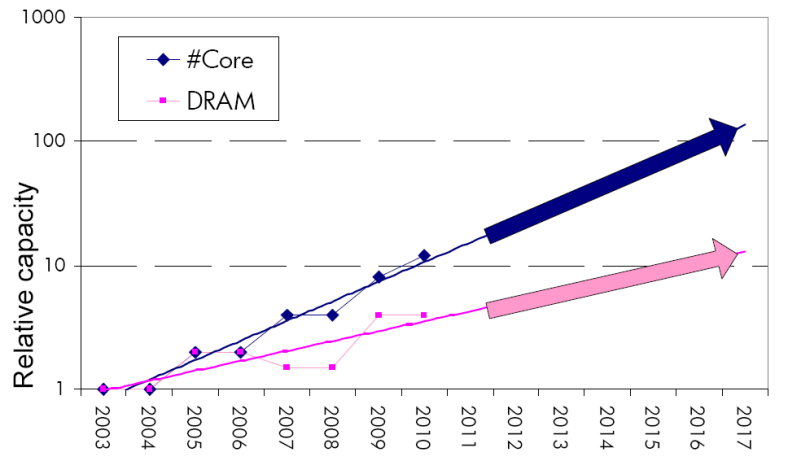
- Non-Volatile-Memory, emerged during the last 2 decades
- Definition: Solid state memory (no moving parts) that do not need to have their memory contents periodically refreshed (source : http://searchstorage.techtarget.com/definition/nonvolatile-memory)
- Flash memory, Phase Change memory (PRAM or PCM), Resistive Memory (ReRAM), Magneto-resistive Memory (STT-RAM), Ferroelectric memories (FeRAM) …

- **Why ?,** mainly because of …
  - DRAM scaling
  - Energy consumption

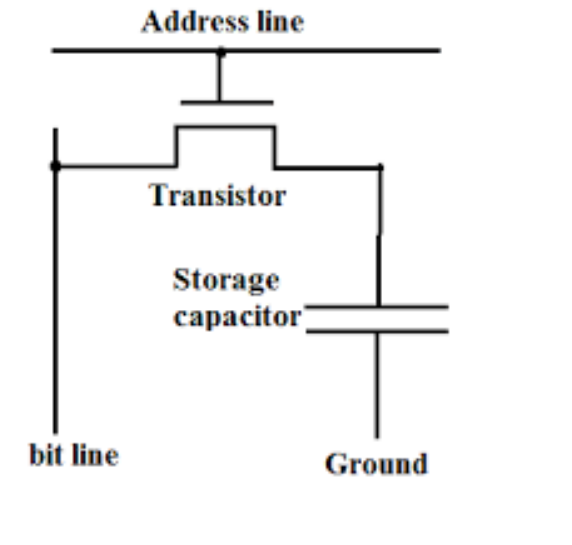
DRAM scaling issue

- Stores the charge in a capacitor
  - Size of capacitor matters → large for a reliable sensing
  - Size of transistor matters → large to reduce leakage and increase retention time
  - Scaling is difficult (ITRS)

Core count doubling ~ every 2 years
DRAM DIMM capacity doubling ~ every 3 years



Source: Onur Mutlu



Charge vs Resistive memory

- Charge memory
  - Write operation: capture charge Q
  - Read operation: detect the voltage V
  - Example: DRAM, flash memory

- Resistive memory
  - Write operation: pulse current dQ/dt
  - Read operation: detect the resistance R
  - Example: PCM, STT-RAM, ReRAM

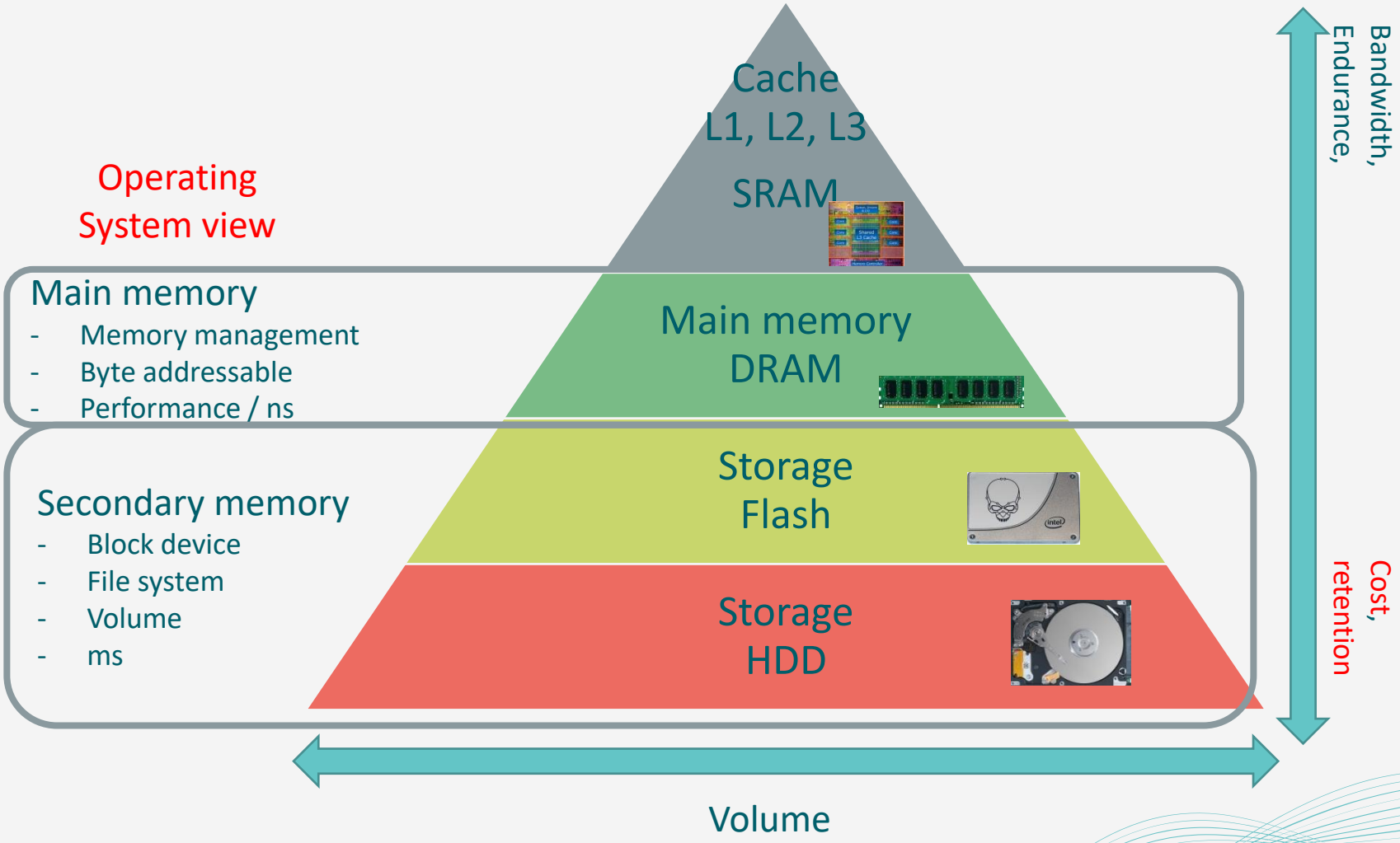
NVM: why is it so attractive ?

* Jalil Boukhobza, Stéphane Rubini, Renhai Chen, Zili Shao, **Emerging NVM: A Survey on Architectural Integration and Research Challenges**, **ACM Transactions on Design Automation of Electronic Systems (TODAES),** 23(2), 14:1-14:32 (2018).

	SRAM	DRAM	HDD	NAND flash	STT-RAM	ReRAM	PCM
Cell size (F^2)	120-200	6-10	N/A	4-6	6-50	4-10	4-12
Write endurance	10^{16}	$>10^{15}$	$>10^{15}$ (pb: mechanical parts)	10^4-10^5	10^{12}-10^{15}	10^8-10^{11}	10^8-10^9
Read Latency	~0.2-2ns	~10ns	3-5ms	15-35 µs	2-35ns	~10ns	20-60ns
Write Latency	~0.2-2ns	~10ns	3-5ms	200-500µs	3-50ns	~50ns	20-150ns
Leakage Power	High	Medium	(mechanical parts)	Low	Low	Low	Low
Dynamic Energy (R/W)	Low	Medium	(mechanical parts)	Low	Low/High	Low/High	Medium/High
Maturity	Mature	Mature	Mature	Mature	Manufactured	Test chips	Manufactured

Sources: [Vetter15] [Mittal15] [Xia15] [Wang'14] J.[Suresh14] [Baek13] [Maena15]

NVM Integration

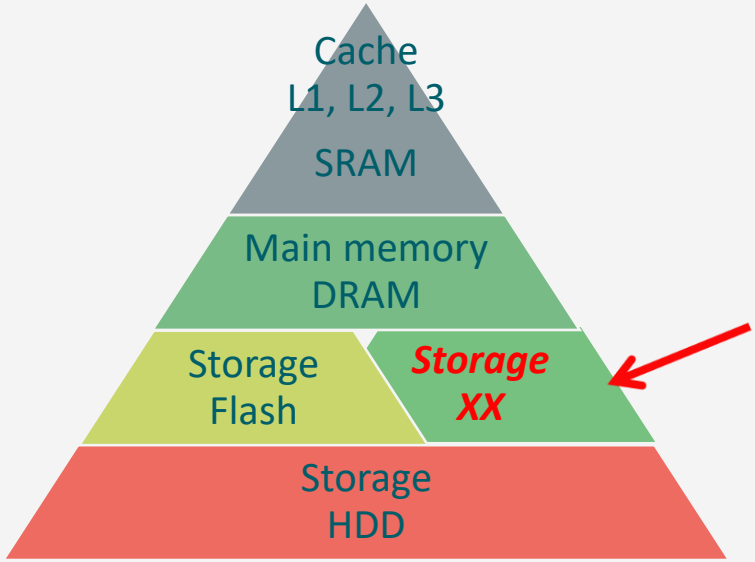


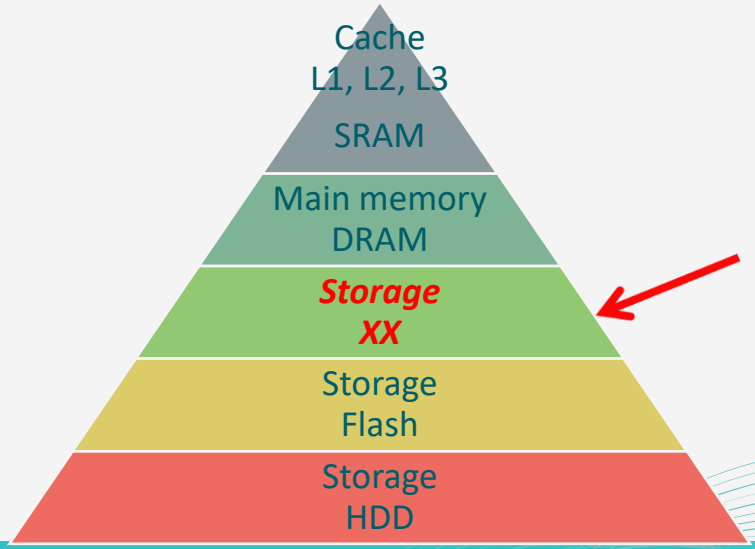
jalil.boukhobza@ensta-bretagne.fr

Horizontal and Vertical integration

- Horizontal integration
  - Same interface as an existing memory
  - Data placement (controller, OS, …)

- Vertical integration
  - Different interface
  - Cache subsystem



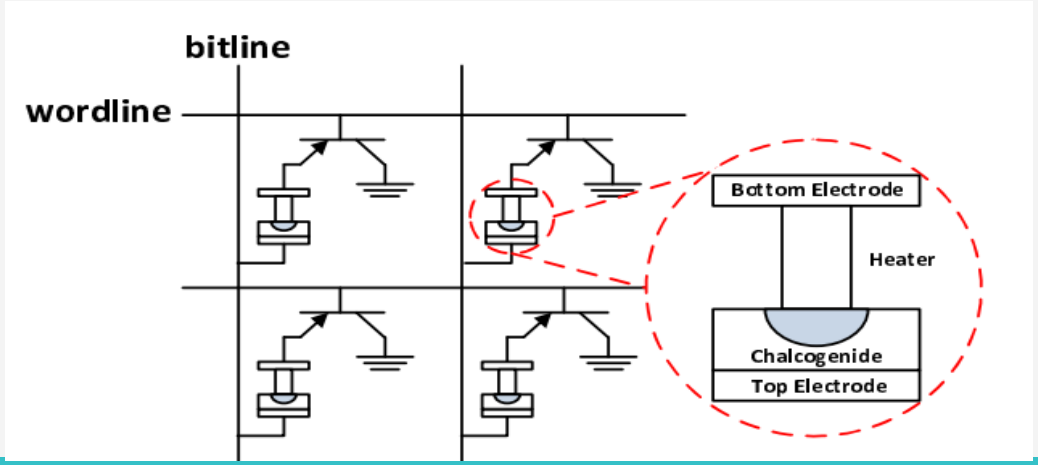


Presentation outline

- Context
- Emerged NVM: flash memory
  - Basics & characteristics
  - Support
  - Integration and performance figures
  - Some contributions
- Emerging NVM
  - Definition & motivation
  - PCM
  - ReRAM
  - STT-RAM
- Conclusion

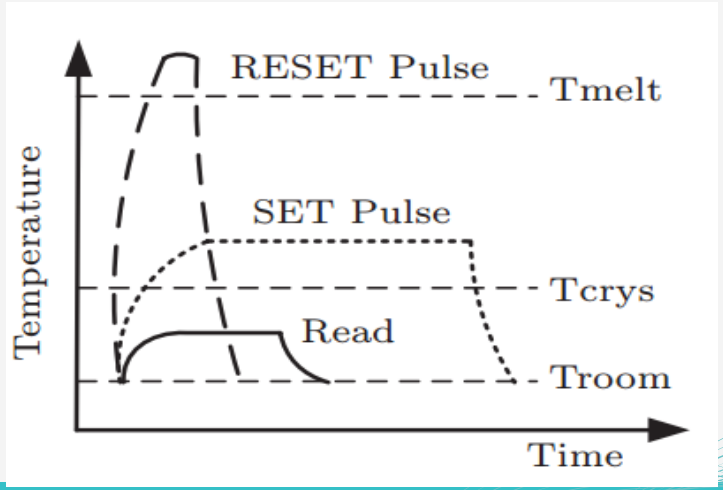
Phase Change random access memory (PCM or PRAM)

- Memory cell: thin layer of chalcogenide such as Ge2Sb2Te5 (GST) + two electrodes wrapping the chalcogenide + heater
- Resistive NVM → use of resistance to represent a bit
  - High resistance (0), Low resistance (1)
- chalcogenide material : rapid and reliable amorphous-to-crystalline phase-change process electrically initiated



PCM -2-

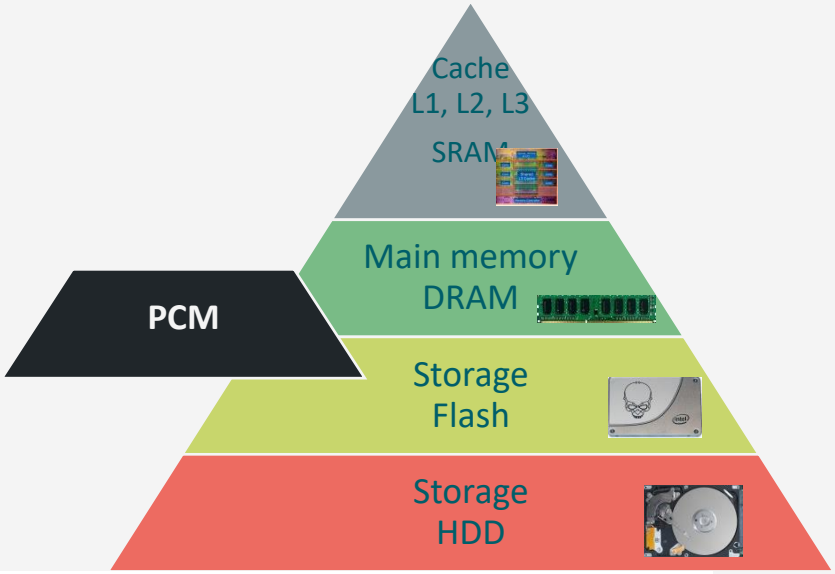
- Short but high voltage pulse → GST heated above the melting temperature T_{melt}
  - Amorphous state, high resistance, (Reset, bit=0)
- Long pulse but low voltage → GST heated above the crystallization temperature T_{crys}
  - Crystalline state, low resistance, (Set, bit=1)
- Ratio between the resistance of the material in a SET and a RESET phase [Kim08]
  - comprised between 10^2 and 10^4
  - → MLC (Multi level Cell)

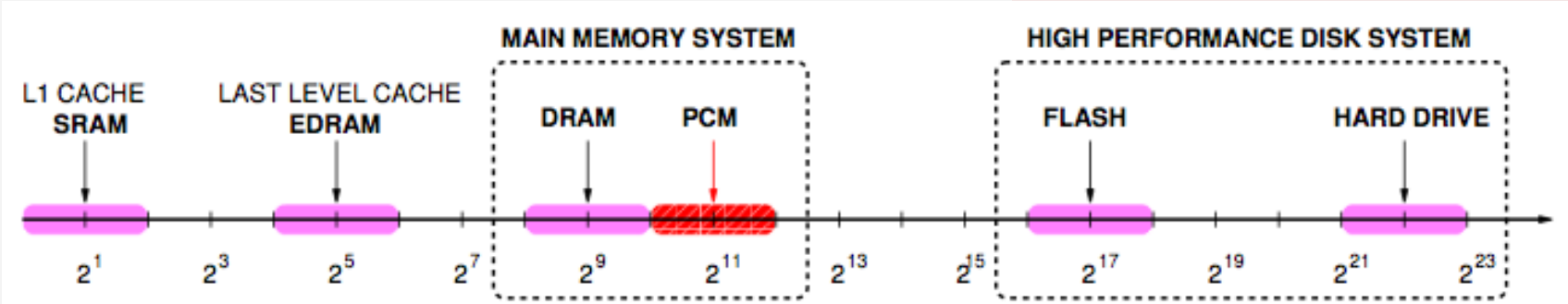


jalil.boukhobza@ensta-bretagne.fr

PCM integration

	DRAM	NAND flash	PCM
Write endurance	>10^{15}	10^4-10^5	**10^8-10^9**
Read Latency	~10ns	15-35 µs	**20-60ns**
Write Latency	~10ns	200-500µs	**20-150ns**





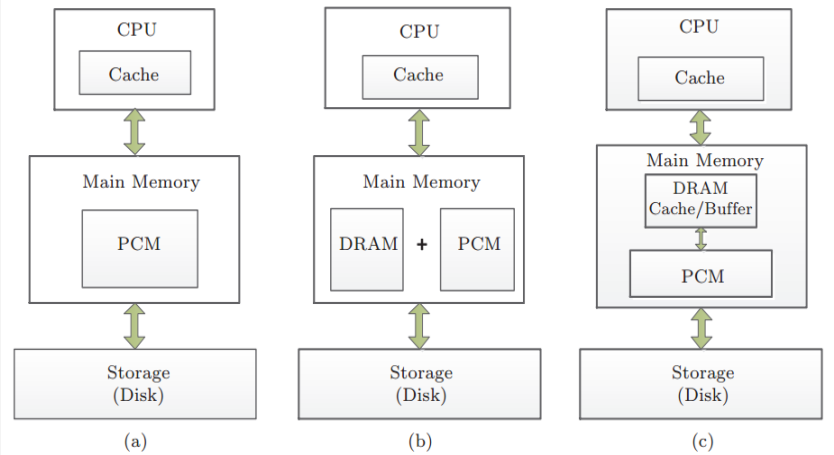
Typical Access Latency (in terms of processor cycles for a 4 GHz processor)

Source: Onur Mutlu

PCM integration -2-

- PCM in memory
  - a) Replacement for DRAM [Park et al. 2015]
  - b) Data placement: data placement at different levels
    - Controller [Dhiman et al. 2009; Qureshi and Srinivasan 2009; Lee et al. 2009]
    - System [Lee et al. 2014; Salkhordeh and Asadi 2016; Wei et al. 2015]
    - Application [Kannan et al. 2016]
  - c) DRAM as a cache to absorb data writes [Qureshi and Srinivasan 2009 ; Awad et al. 2016]

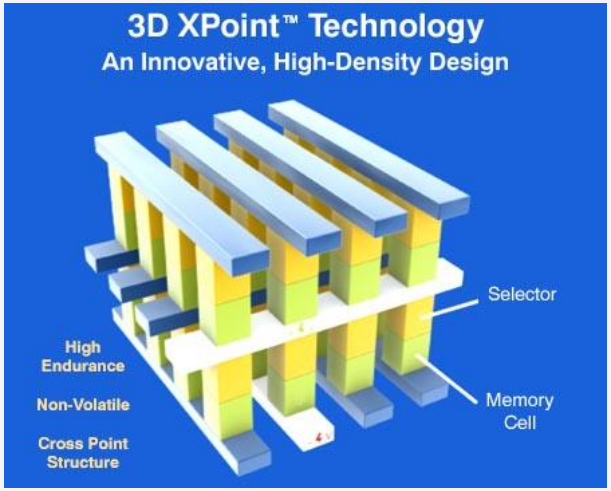
- PCM in storage system
  - Competitor for flash memory [Akel et al. 2011]
  - Buffer for flash device [Sun et al. 2010 ; Liu et al. 2011]



Research issues

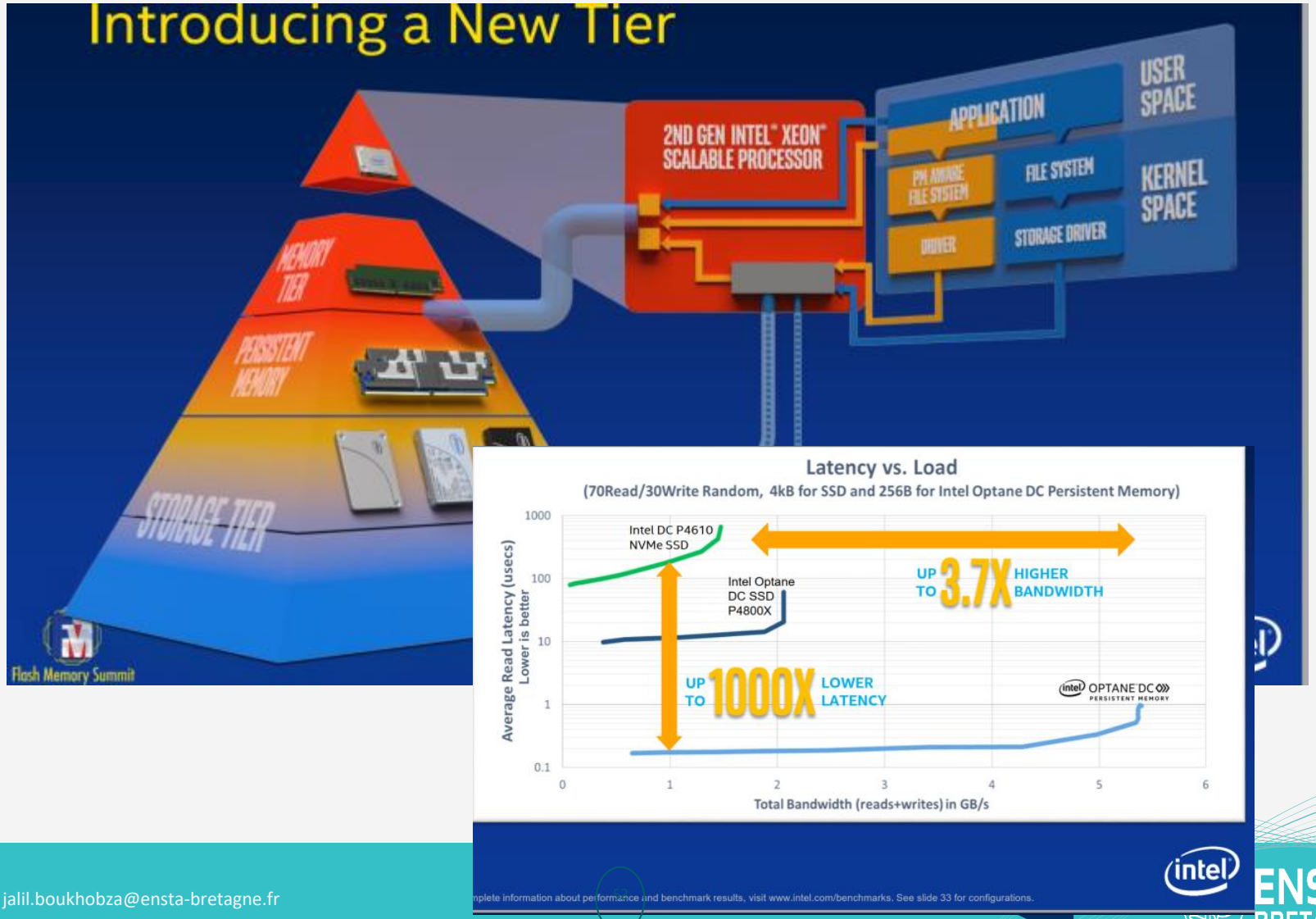
- Write latency
  - Reducing the write bits (only modified ones)
  - Increasing the parallelism of writing bits (set & reset)
  - Better scheduling of writes and reads (to hide write delays)
- Endurance
  - Optimizing electrical properties (optimal reset current)
  - Minimizing write operations (adding a buffer, unmodified bits, etc)
  - Wear leveling
- Write disturbance
  - Minimum inter-cell space, data coding and writing techniques
- Energy saving
  - Reducing the write latency → see above
  - Exploit write energy asymetry (set & reset)

PCM maturity, the case of 3D Xpoint



- Announced by Micron/Intel in 2015
- Intel Optane Memory Media
- PCM technology, 20nm
- Integrated in fast NVMe SSD and DIMM modules
- More Optane 3D Xpoint bits sold than all other emerging memories combined in 2019
- 2nd generation in 2020
- For memory integration: not supposed to replace DRAM, it supplements it (DRAM invisible to application, verticaland horizontal integration)
  - ~Xpoint: DRAM → 5:1 (Intel recommendation)

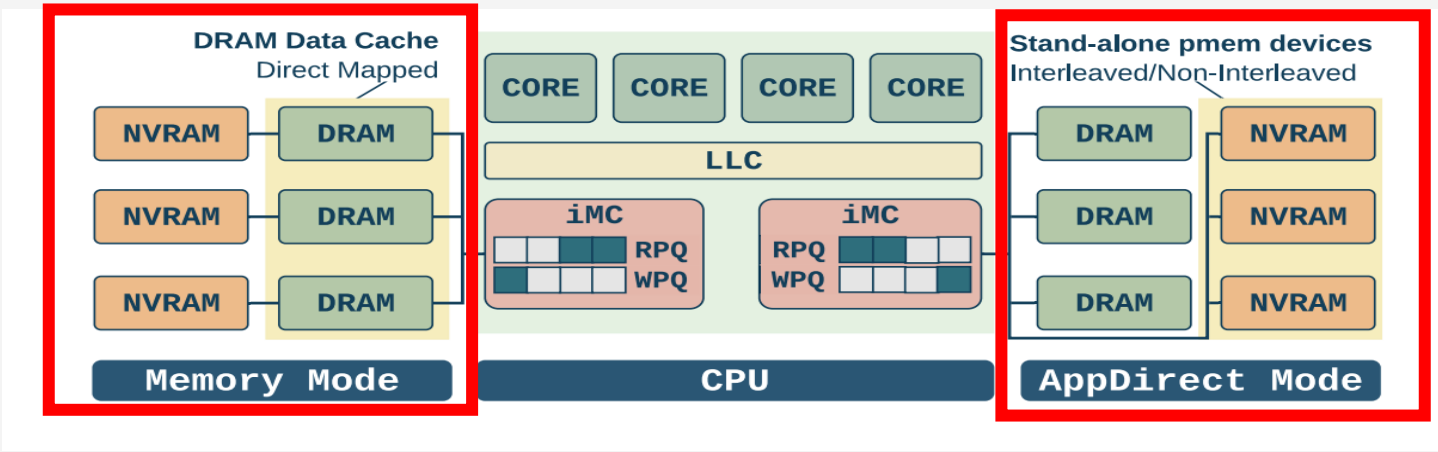
Performance and integration



jalil.boukhobza@ensta-bretagne.fr

Intel Optane DC architecture and performance figures

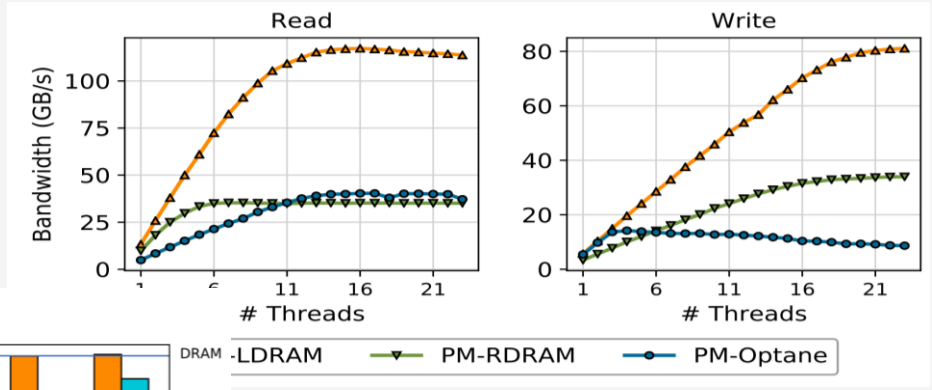
- 2 modes
  - Memory mode
    - Vertical integration with DRAM being a cache to NVRAM
    - → for scalability
  - AppDirect mode
    - Horizontal integration
    - → for persistance (performance compared to traditional storage) + bye addressability (execute in-place)

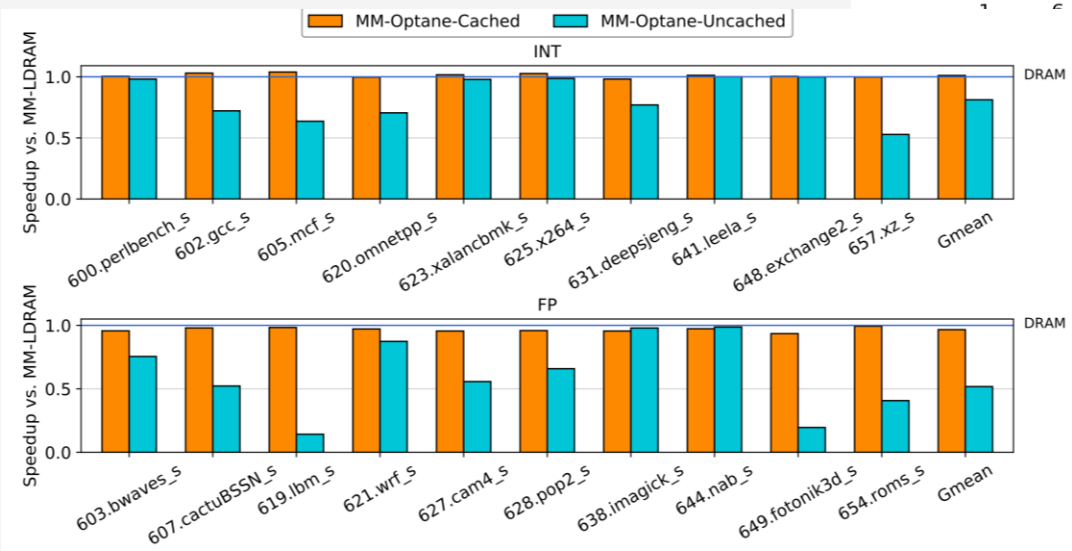


Zixuan Wang, Xiao Liu, Jian Yang, Theodore Michailidis, Steven Swanson, Jishen Zhao, Characterizing and Modeling Non-Volatile Memory Systems, IEEE Micro 2020.

Intel Optane DC architecture and performance figures -2-

- Latency
  - Reads 2 to 3x slower than DRAM (~80ns → ~300ns)
  - Writes same in latency (unless device saturated) → cached writes (~86ns → 94ns)
- Bandwidth
  - AppDirect mode (NVRAM)
- What about real apps ?

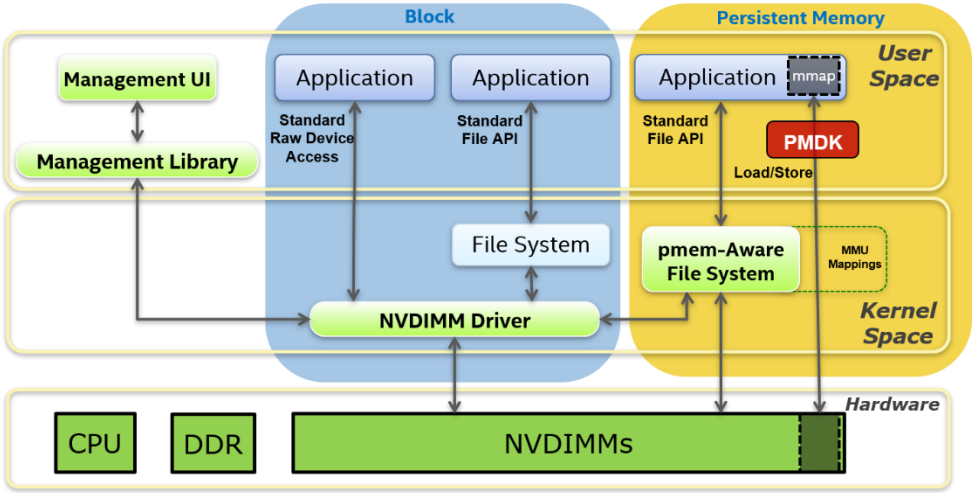




J. Izraelevitz, J. Yang, L. Zhang, J. Kim, X. Liu, A. Memaripour, Y. Joon Soh, Z. Wang, Y. Xu, S. R. Dulloor, J. Zhao, S. Swanson, Basic Performance Measurements of the Intel Optane DC Persistent Memory, 2019Module https://arxiv.org/pdf/1903.05714.pdf

Some Intel Optane challenges …

- Designing new persistant memory file systems: NOVA FS, Ext4-DAX, XFS-DAX → reducing I/O stack latencies
- Dealing with read/write performance asymetry
  - Indexing
  - App design
  - …



https://docs.pmem.io/persistent-memory/getting-started-guide/what-is-pmdk

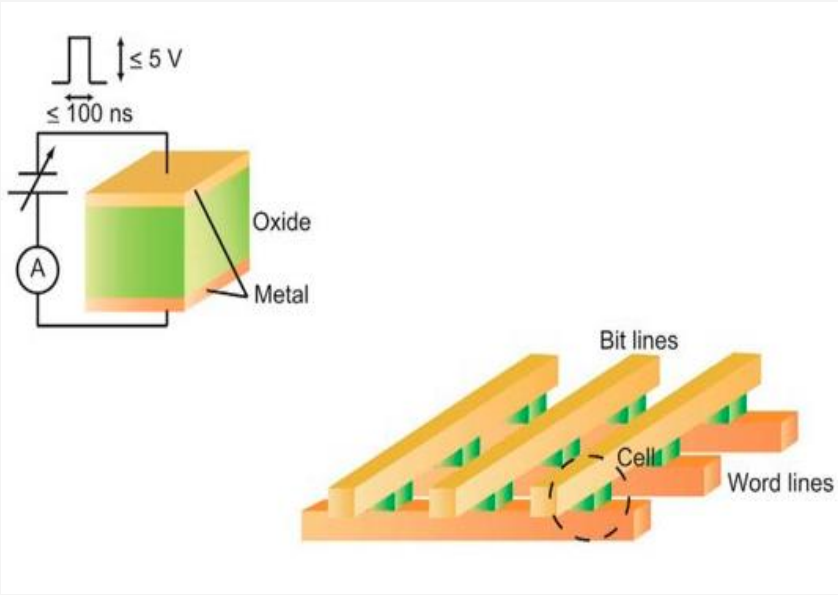
- Design of new API for NVM programming https://www.snia.org/tech_activities/standards/curr_standards/npm and https://docs.pmem.io/persistent-memory/getting-started-guide/what-is-pmdk
- New caching mechanisms
- Manage consistency issues between persistant storage devices
- Managing interference and bandwidth regulation for multi tenant applications (within NVM, between NVM and DRAM)
- …

Presentation outline

- Context
- Emerged NVM: flash memory
  - Basics & characteristics
  - Support
  - Integration and performance figures
  - Some contributions
- Emerging NVM
  - Definition & motivation
  - PCM
  - ReRAM
  - STT-RAM
- Conclusion

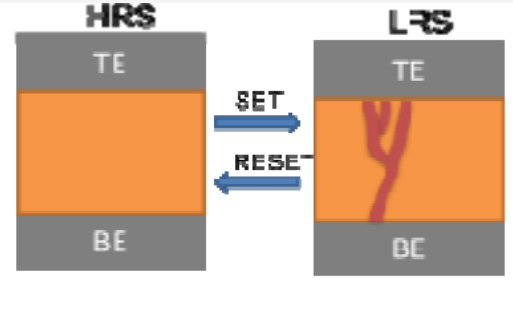
Resistive RAM & maturity

- Not the only « resistive » RAM.
- Compatible with conventional semi-conductor fabrication process.
- ReRAM cell → two terminal metal-insulator-metal (MIM)
  - An insulating or resistive material (I) sandwiched between
  - Two electrode conductors (M)
- A low resistance value → logical "1"
- A high resistance value → logical "0"



ReRAM and memristor

- Basic principle: formation (Low Resistive State) and disruption (High Resistive State) of a conductive filament → to shunt the top and bottom electrodes (M) through the resistive oxide layer

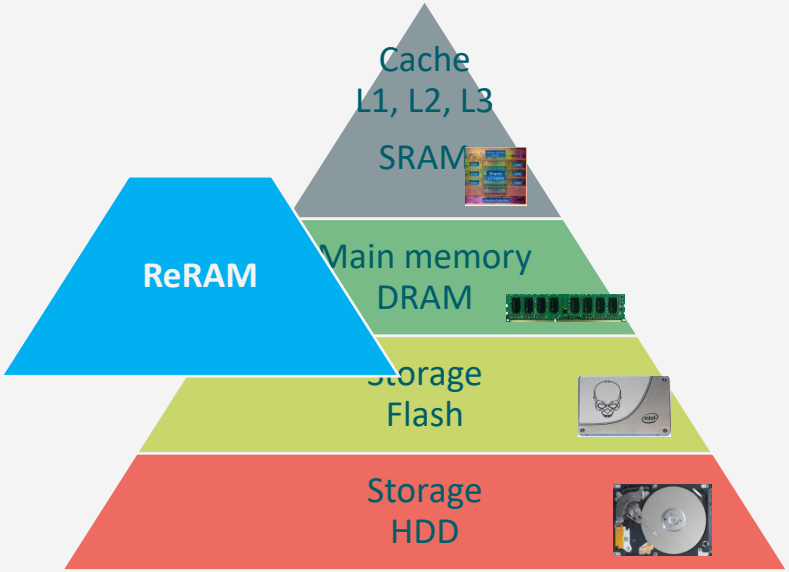


Source: Clermidy et al. 2014

- ReRAM:
  - Unipolar devices: switching depends on the amplitude of the applied voltage
  - Bipolar devices: switching depends on voltage polarity → **memristors**
- Memristor : oxygen vacancy creation → electric path between top and bottom electrodes (TiOx, ZrOx, NiO, …)
- Concept of memristor → 1971 invented by Leon Chua [Chua71]
  - Fourth fundamental building block (along with resistor, capacitors and inductors).
  - Memory resistor: "… the ability to indefinitely store resistance values"

ReRAM integration

	SRAM	DRAM	NAND flash	ReRAM
Cell size (F^2)	120-200	60-100	4-6	4-10
Write endurance	10^{16}	$>10^{15}$	10^4-10^5	**10^8-10^{11}**
Read Latency	~0.2-2ns	~10ns	15-35 µs	**~10ns**
Write Latency	~0.2-2ns	~10ns	200-500µs	**~50ns**



ReRAM integration

- ReRAM in cache memory
  - Design exploration in several cache levels [Dong et al. 2012], horizontal integration with SRAM in L2 cache [Mittal and Vetter 2015a], with FPGA [Clermidy et al. 2014]
- ReRAM in main memory
  - Hybrid: [Hassan et al. 2015]
  - Replacement: [Xu et al. 2015]
- ReRAM in storage systems
  - Competitor for flash memory [Jung et al. 2013]
  - Horizontal integration [Tanakamaru et al. 2014; Sun et al. 2014; Fujii et al. 2012]

ReRAM maturity
(source: M.Webb Flashmemory summit 2019)

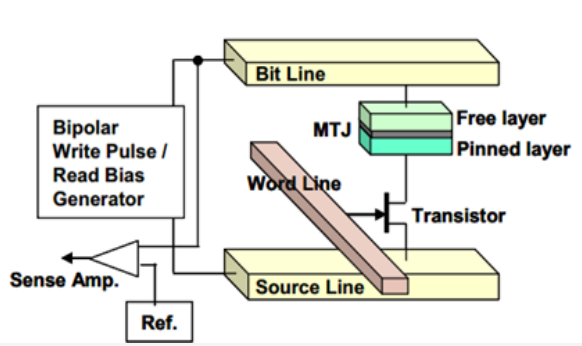
- Lots of technology papers
- Came from embedded applications (Kbits or Mbits)
  - Unity (RAMBUS), Adesto, Panasonic, Fujitsu, etc.
- 2018: Crossbar Inc, ReRAM available (1T1R) from foundry
- 2017: 4DS, 1000+ cells (comparable to DRAM speeds), partnering with IMEC
- Embedded ReRAM announced by Intel (22nm, 5ns access time)
- Considered for AI and neuromorphic applications
- Why? → **scalable, low cost and « simple », no significant new fabrication tools needed**

Presentation outline

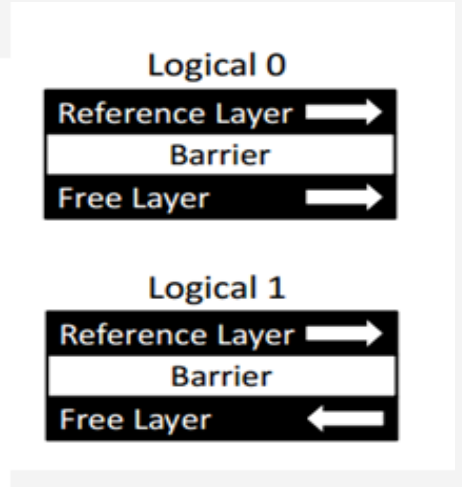
- Context
- Emerged NVM: flash memory
  - Basics & characteristics
  - Support
  - Integration and performance figures
  - Some contributions
- Emerging NVM
  - Definition & motivation
  - PCM
  - ReRAM
  - STT-RAM
- Conclusion

STT-RAM (Spin Torque Transfer RAM)

- Information carrier is the Magnetic Tunnel Junction (MTJ)
- Magnetic Tunnel Junction (MTJ) device
  - Reference layer: Fixed magnetic orientation
  - Free layer: Parallel or anti-parallel



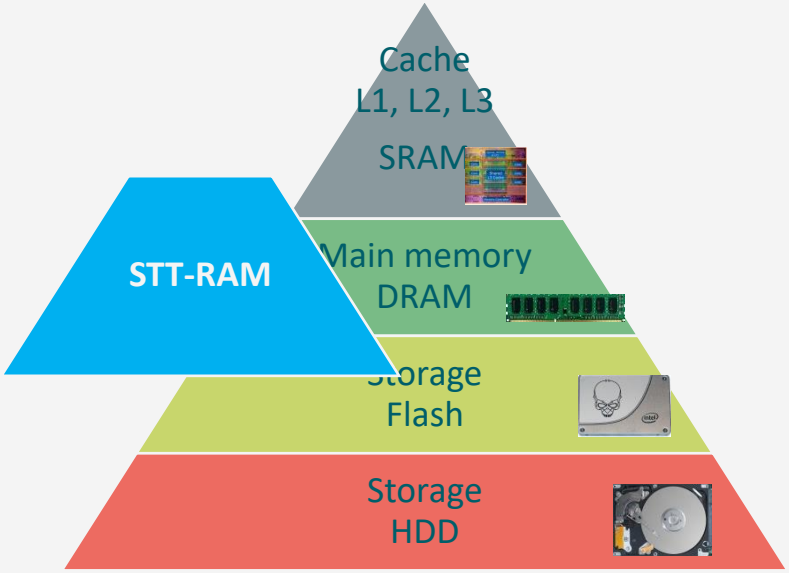
- Magnetic orientation of the free layer determines logical state of device
  - High resistance → anti-parallel stat: 1
  - Low resistance → parallel state: 0

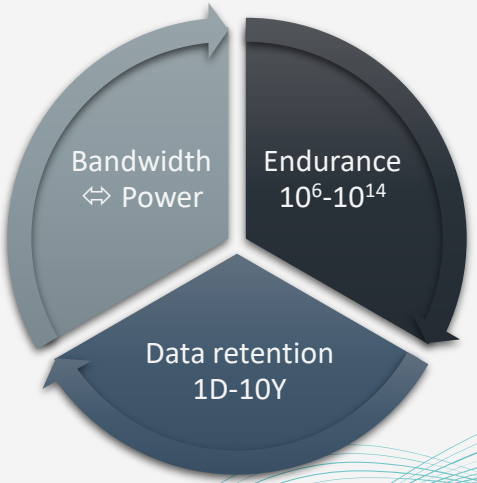


- Operations:
  - Write: Push large current through MTJ to change orientation of free layer
  - Read: Sense current flow

STT-RAM Integration

	SRAM	DRAM	NAND flash	STT-RAM
Cell size (F^2)	120-200	60-100	4-6	6-50
Write endurance	10^{16}	$>10^{15}$	10^4-10^5	**10^{12}-10^{15}**
Read Latency	~0.2-2ns	~10ns	15-35 µs	**2-35ns**
Write Latency	~0.2-2ns	~10ns	200-500µs	**3-50ns**





jalil.boukhobza@ensta-bretagne.fr

STT-RAM (source: Onur Mutlu)

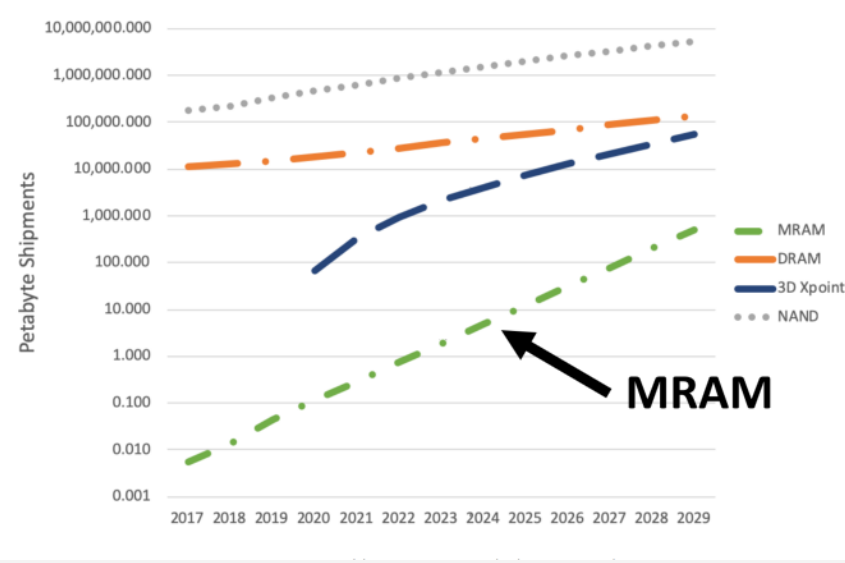
- Pros over DRAM
  - Better technology scaling (capacity and cost)
  - Non volatile → Persistent
  - Low idle power (no refresh)
- Cons
  - Higher write latency
  - Higher write energy
  - Poor density (currently, but high potential)
- Another level of freedom
  - Can trade off non-volatility for lower write latency/energy (by reducing the size of the MTJ)

STT-RAM integration

- STT-RAM in cache memory
  - In all cache levels [Li et al. 2012; Komalan et al. 2014; Senni et al. 2015]
  - In last cache levels [Syu et al. 2013; Cheng et al. 2016]
  - Vertical [Wu et al. 2009] and horizontal [Sun et al. 2009; Wu et al. 2009; Jadidi et al. 2011; Li et al. 2011] integration
- STT-RAM in main memory
  - Replacement of DRAM [Kultursay et al. 2013]
  - Horizontal integration [Yang et al. 2013]
  - Vertical integration [Suresh et al. 2014]
- STT-RAM in storage system
  - Horizontal and replacement [Lee et al. 2014]
  - Vertical [Kang et al. 2015]

STT-RAM maturity

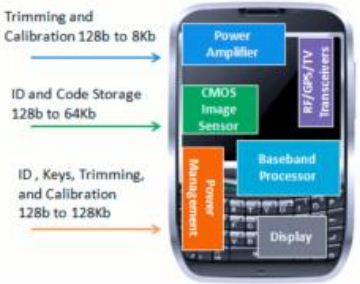
- Embedded MRAM
  - Potential replacement for eFlash and SRAM (because of scaling)
  - Multiple announcements: TSMC and Intel (22nm), Samsung (28nm) …
- Standalone MRAM
  - No replacement for DRAM or NAND expected
  - Battery backed or low density DRAM, some NOR applications
  - Everspin shipping 256Mb
  - Announced 1Gb availability

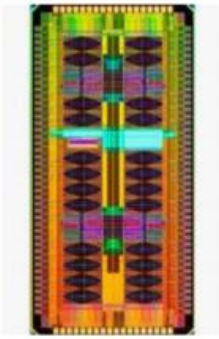


Source: D. Worledge, IBM, 2019

MRAM Applications

	Standalone	Embedded Non-volatile	Mobile Cache	Last Level Cache
Application examples	• Replace battery-backed SRAM or DRAM • Buffer for hard disk drive • Replace DRAM	Replace NOR eFlash to store: ➓ Microcontroller code ➓ Encryption key storage ➓ Trimming and calibration	Replace SRAM for low performance & power apps • Wearable electronics • Co-processors • Internet of Things	• Fast dense memory for L3 or L4 cache • Alternative to eDRAM
Key requirements	• Lower temp process is OK • 0 C – 70 C operation • 256 Mb – 1 Gb and up • 30 - 70 ns read/write • High endurance (10^{10} - 10^{15})	• 400C process required • -40 C – 105/125/150 C • 1 – 64 Mb • 30 ns read, 200 ns write • Low endurance: 10^{6}	• 400C process required • 0 C – 85 C operation • 1 – 64 Mb • 10 ns read/write • High endurance (10^{12} - 10^{17})	• 400C process required • 0 C – 85 C operation • 1 Gb and up • 1 - 2 ns read/write • Unlimited endurance (10^{18})





Increasing difficulty

Daniel Worledge | MRAM Developer Day | 8/5/19 | p 11 | © 2019 IBM

Source: S. Bertolazzi, Yole Développement, Flash memory summit 2019

MRAM DEVELOPER DAY

The MRAM Ecosystem is Growing

An increasing number of players are involved in the MRAM arena

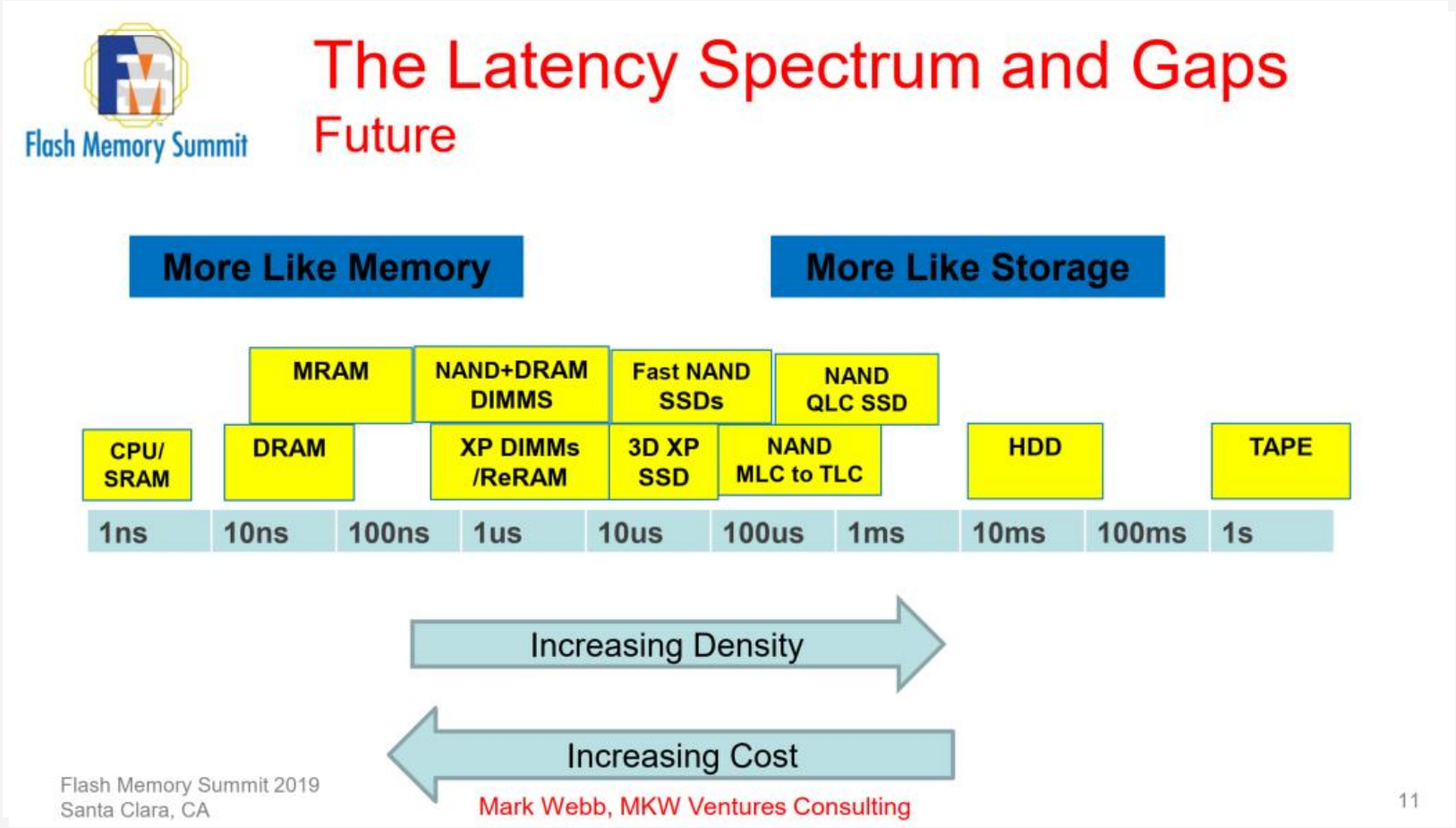
MRAM IP and Design	EVERSPIN, avalanchetechnology, SPIN MEMORY, TDK, IBM, HFC Semiconductor, QUALCOMM, arm, SYNOPSYS, NVE CORPORATION, Inston, numem, SHANGHAI CIYU, HIKSTOR TECHNOLOGY CO., LTD
Embedded MRAM manufacturers	Players in mass production or close to mass production: tsmc, GLOBALFOUNDRIES, SAMSUNG, intel; UMC, TOWERJAZZ, CROCUS NANO ELECTRONICS
Stand-alone MRAM manufacturers	40nm, up to 128Mb: SONY, avalanchetechnology; 40nm, 28nm (256M, 1Gb): GLOBALFOUNDRIES, EVERSPIN; Toggle Manufacturing: SILTERRA, EVERSPIN; Expected: 28nm, 22nm: UMC, avalanchetechnology

Presentation outline

- Context
- Emerged NVM: flash memory
  - Basics & characteristics
  - Support
  - Integration and performance figures
  - Some contributions
- Emerging NVM
  - Definition & motivation
  - PCM
  - ReRAM
  - STT-RAM
- Conclusion

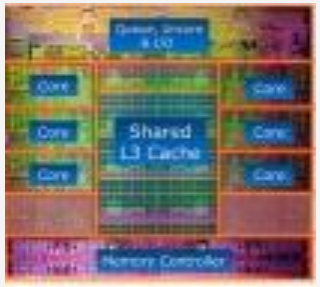
NVM characteristics

- Common characteristics
  - Byte addressable
  - Performance properties → very good for read, good for write
  - Energy properties → static power ↘, dynamic power ?
  - Scalability
- Constraints to deal with
  - Performance asymmetry
    - Between write and read
    - Writing 1 ≠ writing 0
  - Energy consumption asymmetry
  - Wear out


Flash Memory Summit
The Latency Spectrum and Gaps
Future
More Like Memory
More Like Storage
MRAM
NAND+DRAM DIMMS
Fast NAND SSDs
NAND QLC SSD
CPU/ SRAM
DRAM
XP DIMMs /ReRAM
3D XP SSD
NAND MLC to TLC
HDD
TAPE
1ns
10ns
100ns
1us
10us
100us
1ms
10ms
100ms
1s
Increasing Density
Increasing Cost
Flash Memory Summit 2019
Santa Clara, CA
Mark Webb, MKW Ventures Consulting
11

NVM takeaway

- **You cannot avoid it !**
- **As a processor cache (mainly STT-RAM)**
  - High frequency access → low latency and high endurance
    - NVM ☹
  - Last level cache → acceptable
    - Many write operations → wear leveling
  - Technology compatibility



- **As a main memory (STT-RAM, PCM & ReRAM)**
  - Asymmetric performance / power consumption
  - Vertical integration → DRAM as cache
  - Horizontal integration → data placement

- **As a storage system (PCM & ReRAM)**
  - Vertical integration → example: hybrid disks
  - Horizontal integration → data placement controller ?, OS?, application ?

Integration in state-of-the-art work

		MRAM	eRAM	PCM	FeRAM
Cache (different levels)	Horizontal	[Oboril et al. 2015; Li et al. 2012; Syu et al. 2013; Li et al. 2014; Wu et al. 2009; Jadidi et al. 2011; Li et al. 2011; Wang et al. 2014 ; Komalan et al. 2014; Cheng et al. 2016]	[Wang et al. 2014a; Komalan et al. 2013; Mittal and Vetter 2015]	[Wu et al. 2009; Joo et al. 2010]	
	Vertical	[Oboril et al. 2015; Senni et al. 2014; Sun et al. 2009; Wu et al. 2009; Samavatian et al. 2014; Smullen et al. 2011; Jog et al. 2012; Zhou et al. 2009b; Goswami et al. 2013; Yazdanshenas et al. 2014; Ahn et al. 2012; Rasquinha et al. 2010; Park et al. 2012; Chen et al. 2013; Kwon et al. 2014; Jokar et al. 2016; Senni et al. 2015; Cheng et al. 2016]	[Dong et al. 2013; Wang et al. 2013; Jokar et al. 2016]	[Wu et al. 2009]	
	Replacement	[Oboril et al. 2015; Smullen et al. 2011; Sun et al. 2011; Guo et al. 2010; Goswami et al. 2013; Wang et al. 2015]	[Dong et al. 2013]		
Main Memory	Horizontal	[Yang et al. 2013; Suresh et al. 2014; Wei et al. 2015]	[Hassan et al. 2015; Wei et al. 2015]	[Dhiman et al. 2009; Park et al. 2010; Bock et al. 2011; Suresh et al. 2014; Zhou et al. 2009a; Sun et al. 2015; Wei et al. 2015; Lee et al. 2014; Salkhordeh and Asadi 2016; Wei et al. 2015; Kannan et al. 2016; Dulloor et al. 2016; Wu et al. 2016 ; Li et al. 2012 ; Oikawa 2014; Gao et al. 2015]	[Lee et al. 2007; Suresh et al. 2014]
	Vertical	[Suresh et al. 2014]		[Qureshi and Srinivasan 2009; Suresh et al. 2014; Awad et al. 2016; Wu et al. 2016]	[Suresh et al. 2014; Jung et al. 2010]
	Replacement	[Kultursay et al. 2013; Wang et al. 2014; Jin et al. 2014]	[Xu et al. 2013 ; Xu et al. 2015]	[Lee et al. 2009; Chen et al. 2012; Park et al. 2015]	[Baek et al. 2013]
Storage	Horizontal	[Lee et al. 2014]	[Tanakamaru et al. 2014; Sun et al. 2014; Fujii et al. 2012]	[Sun et al. 2010; Caulfield et al. 2010; Park et al. 2010]	[Moon et al. 2008]
	Vertical	[Kang et al. 2015]		[Liu et al. 2011 ; Kang et al. 2015]	
	Replacement	[Lee et al. 2014]	[Jung et al. 2013]	[Akel et al. 2011; Kim et al. 2014]	[Baek et al. 2013]

Jalil Boukhobza, Stéphane Rubini, Renhai Chen, Zili Shao, **Emerging NVM: A Survey on Architectural Integration and Research Challenges, ACM Transactions on Design Automation of Electronic Systems (TODAES),** 23(2), 14:1-14:32 (2018).

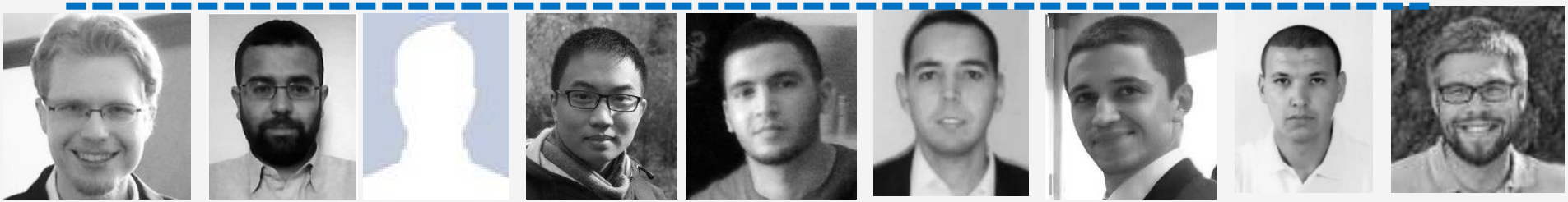
Not discussed in this presentation

- FeRAM: Ferroelectric RAM / Ferroelectric capacitor
- NRAM: Nano RAM / carbon nanotubes
- Racetrack memories:
  - Skyrmion: nanoscale structures that form in magnetic materials, taking the shape of a vortex, hence the name "**magnetic vortex**"
- In-memory computing → one day GDR SoC2 (15/2/2021)
  - https://www.gdr-soc.cnrs.fr/2021/01/22/programme-journee-thematique-in-memory-computing-from-device-to-programming-model/
- Flash memory
  - Open Channel SSDs
  - Interfaces the flash memory for better performance

Special thanks to …

Colleagues

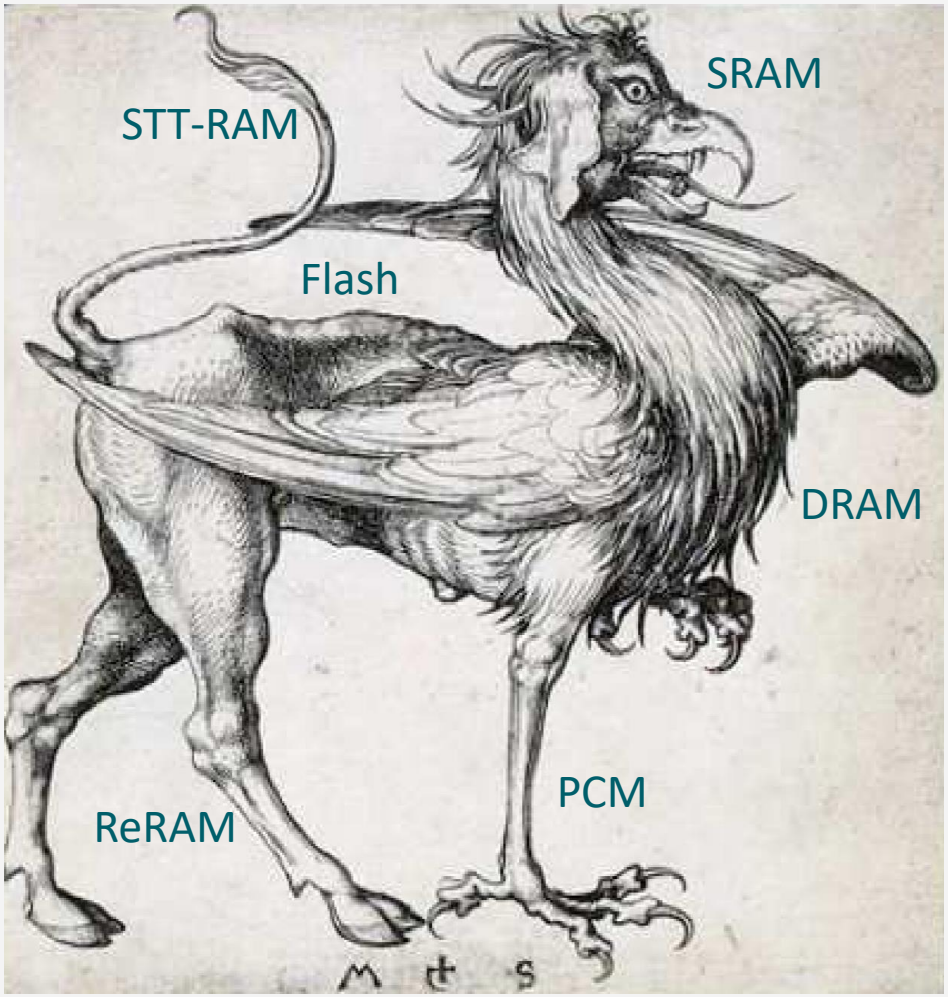
Collaborators



PhD students

jalil.boukhobza@ensta-bretagne.fr

Memory chimera



https://www.ensta-bretagne.fr/boukhobza/
jalil.boukhobza@ensta-bretagne.fr