Implications of Storage Class Memories (SCM) on Software Architectures

C. Mohan, IBM Almaden Research Center, San Jose, CA 95120
mohan@almaden.ibm.com http://www.almaden.ibm.com/u/mohan

13th International Workshop on High Performance Transaction Systems (HPTS)
Asilomar, USA, October 2009

© 2009 IBM
Corporation

Acknowledgements and References

Thanks to

- Colleagues in various IBM research labs in general
- Colleagues in IBM Almaden in particular

References:

- "Storage Class Memory, Technology, and Uses", Richard Freitas, Winfried Wilcke, Bülent Kurdi, and Geoffrey Burr, Tutorial at the 7th USENIX Conference on File and Storage Technologies (FAST '09), San Francisco, February 2009, http://www.usenix.org/events/fast/tutorials/T3.pdf
- "Storage Class Memory - The Future of Solid State Storage", Richard Freitas, Flash Memory Summit, Santa Clara, August 2009, http://www.flashmemorysummit.com/English/Collaterals/Proceedings/2009/20090812_T1B_Freitas.pdf
- "Storage Class Memory: Technology, Systems and Applications", Richard Freitas, 35th SIGMOD international Conference on Management of Data, Providence, USA, June 2009.

©2009 IBM Corporation

Storage Class Memory (**SCM**)

- A new class of data storage/memory devices
  - many technologies compete to be the ‘best’ SCM
- SCM *blurs the distinction* between
  - Memory (*fast, expensive, volatile*) and
  - Storage (*slow, cheap, non-volatile*)
- SCM features:
  - Non-volatile
  - Short access times (~ DRAM like)
  - Low cost per bit (disk like – by 2020)
  - Solid state, no moving parts

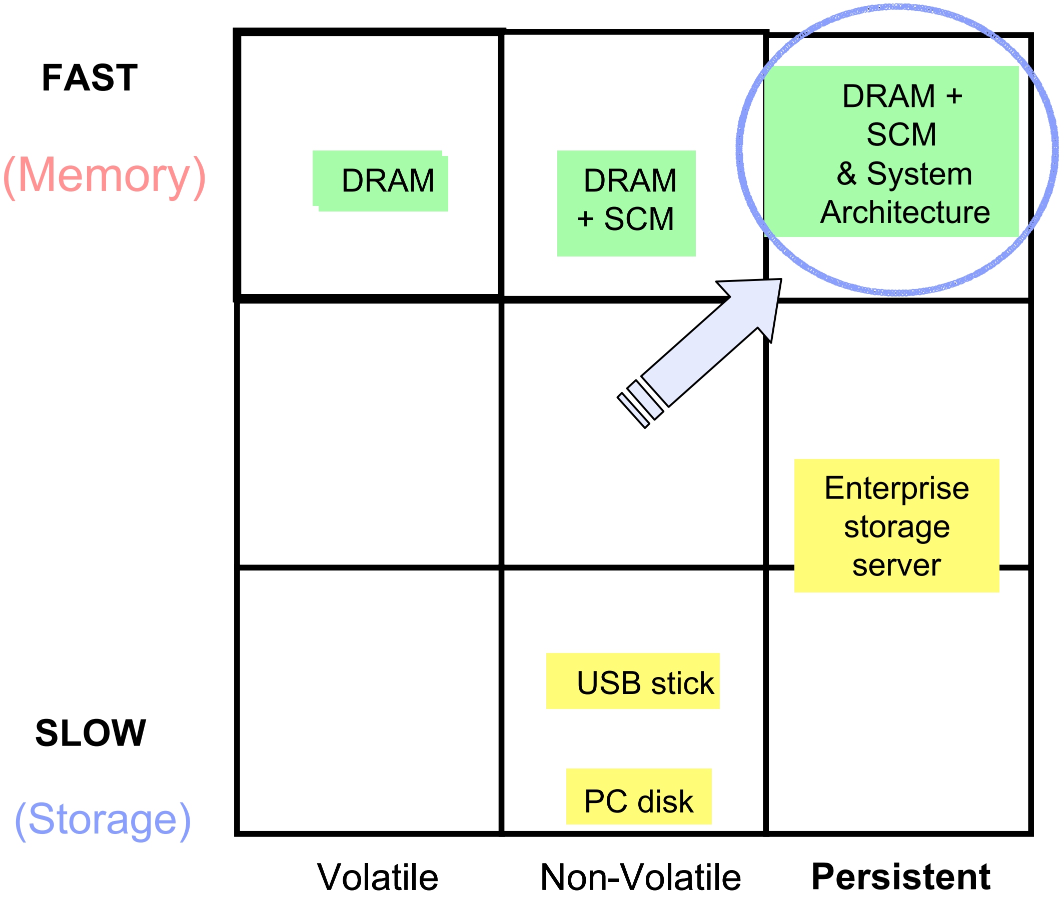
©2009 IBM Corporation

Industry SCM Activities

- SCM research in IBM
- Intel/ST-Microelectronics spun out Numonyx (FLASH & PCM)
- Samsung, Numonyx sample PCM chips
  - 128Mb Numonyx chip (90nm) shipped in 12/08 to select customers
    Samsung started production of 512Mb (60nm) PCM in 9/09
  - Working together on common PCM spec
- Over 30 companies work on SCM
  - including all major IT players

©2009 IBM Corporation

Speed/Volatility/Persistency Matrix



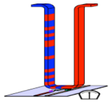
	Volatile	Non-Volatile	**Persistent**
FAST (Memory)	DRAM	DRAM + SCM	DRAM + SCM & System Architecture
			Enterprise storage server
SLOW (Storage)		USB stick PC disk	

Persistent storage will not lose data

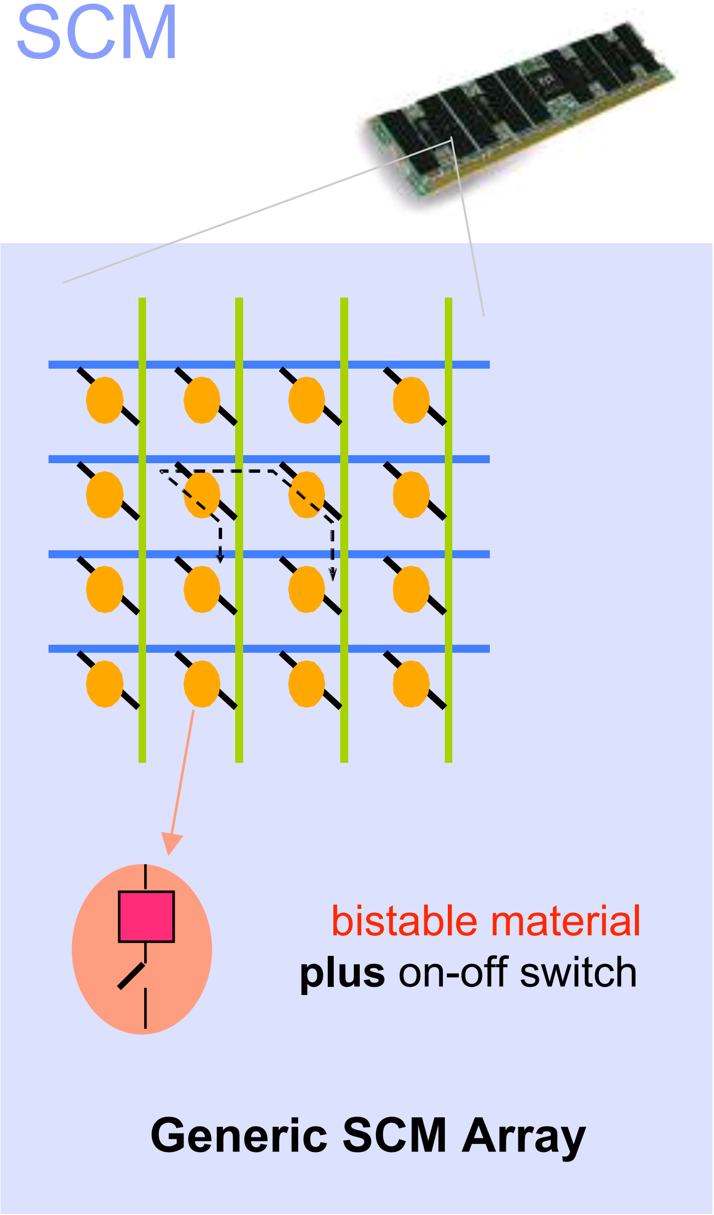
© 2009 IBM Corporation

Many Competing Technologies for SCM

- Phase Change RAM
  - most promising now (scaling)
- Magnetic RAM
  - used today, but poor scaling and a space hog
- Magnetic Racetrack
  - basic research, but very promising long term
- Ferroelectric RAM
  - used today, but poor scalability



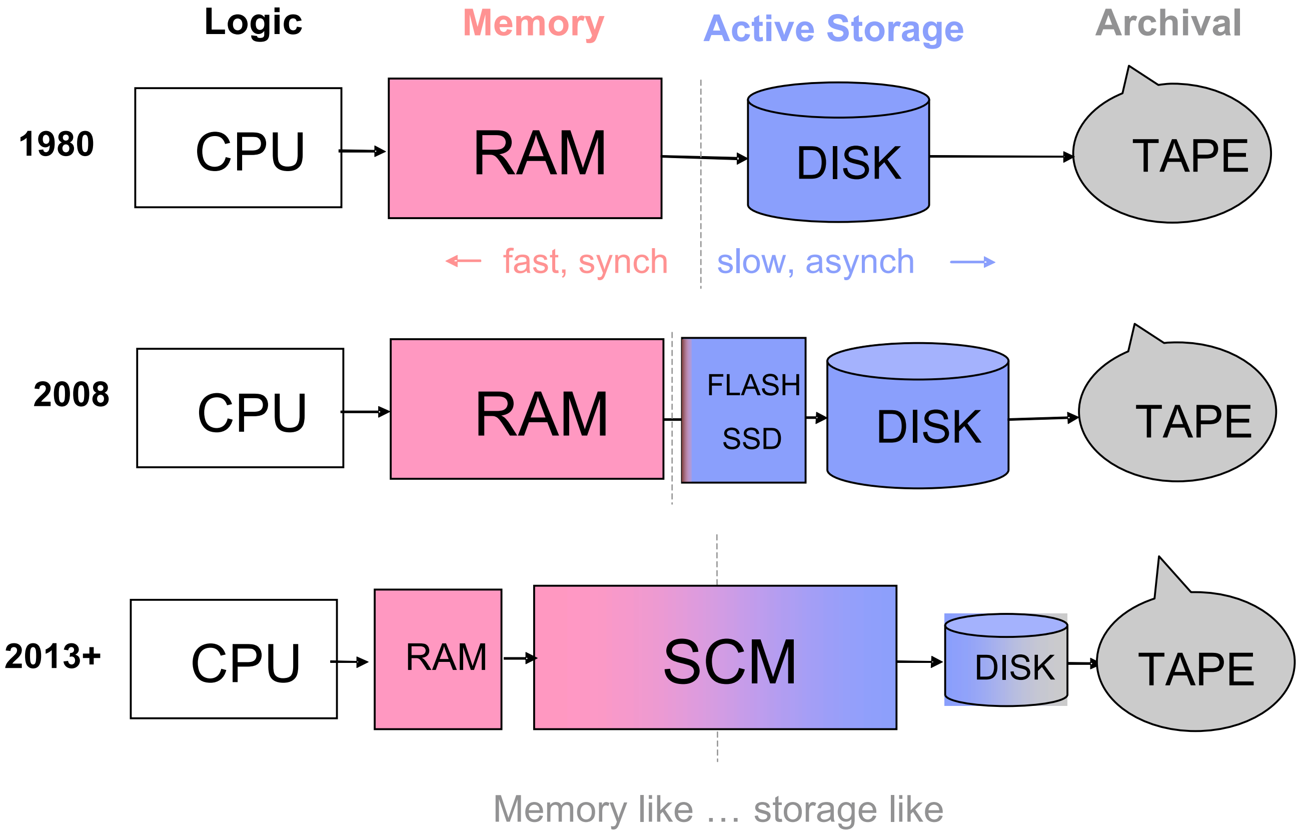
- Solid Electrolyte and resistive RAM (Memristor)
  - early development, maybe promising
- Organic, nano particle and polymeric RAM
  - many different devices in this class, unlikely
- Improved FLASH
  - still slow and poor write endurance



bistable material
plus on-off switch

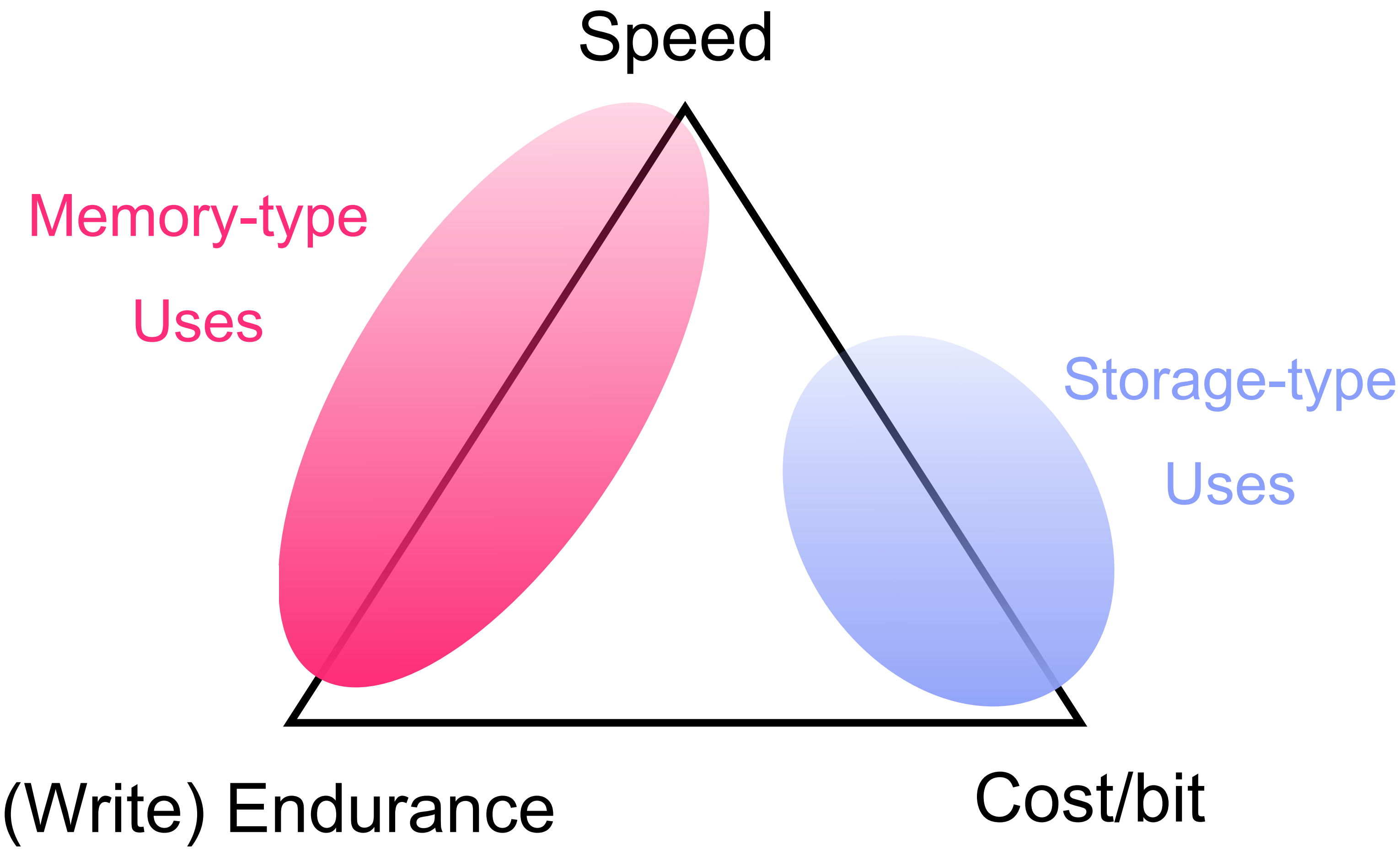
Generic SCM Array

SCM as Part of Memory/Storage Solution Stack



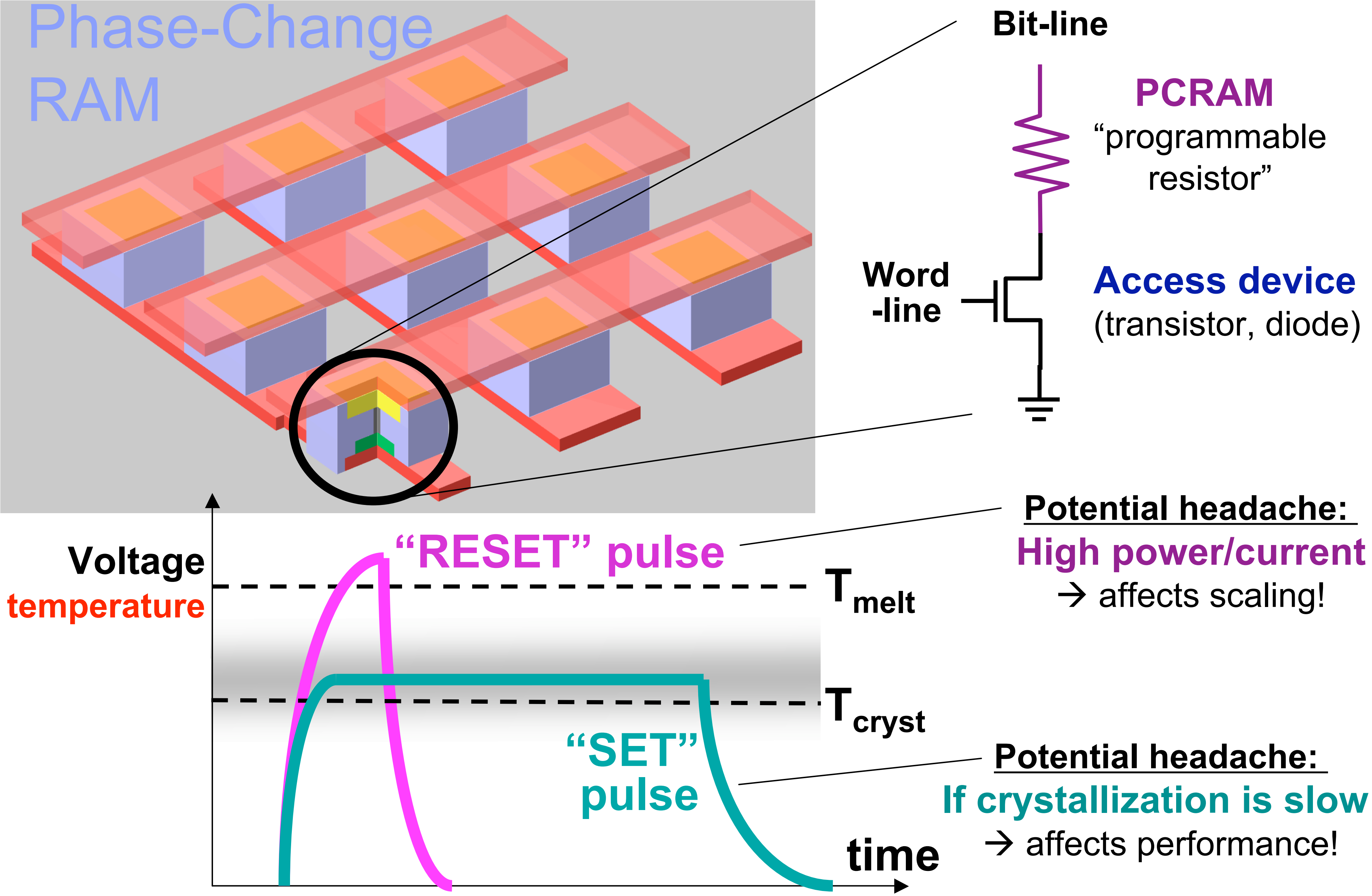
© 2009 IBM Corporation

SCM Design Triangle



© 2009 IBM Corporation

Phase-Change RAM



PCRAM
"programmable resistor"

Access device
(transistor, diode)

"RESET" pulse

Potential headache:
High power/current
→ affects scaling!

"SET" pulse

Potential headache:
If crystallization is slow
→ affects performance!

© 2009 IBM Corporation

If you could have SCM, why would you need anything else?

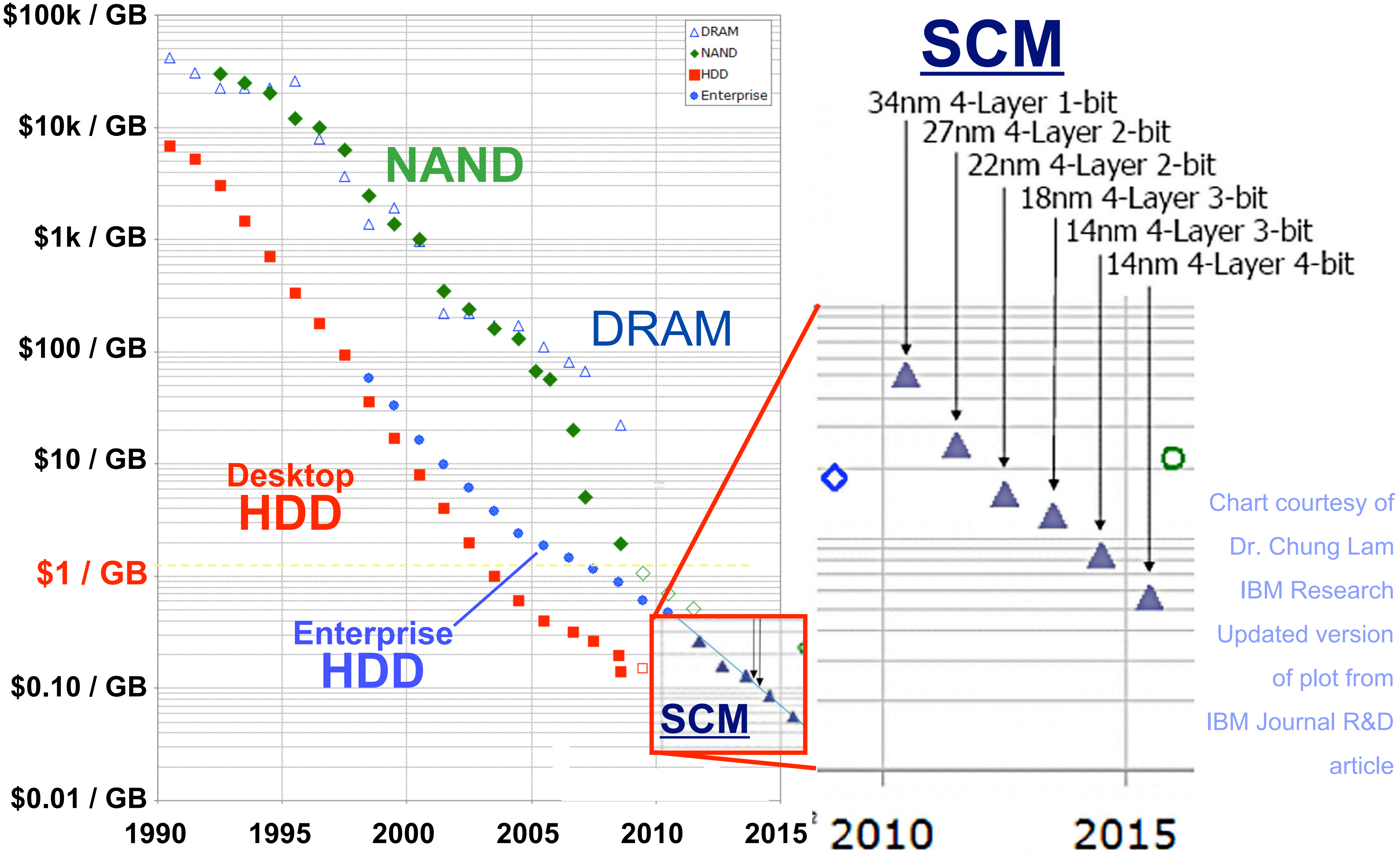
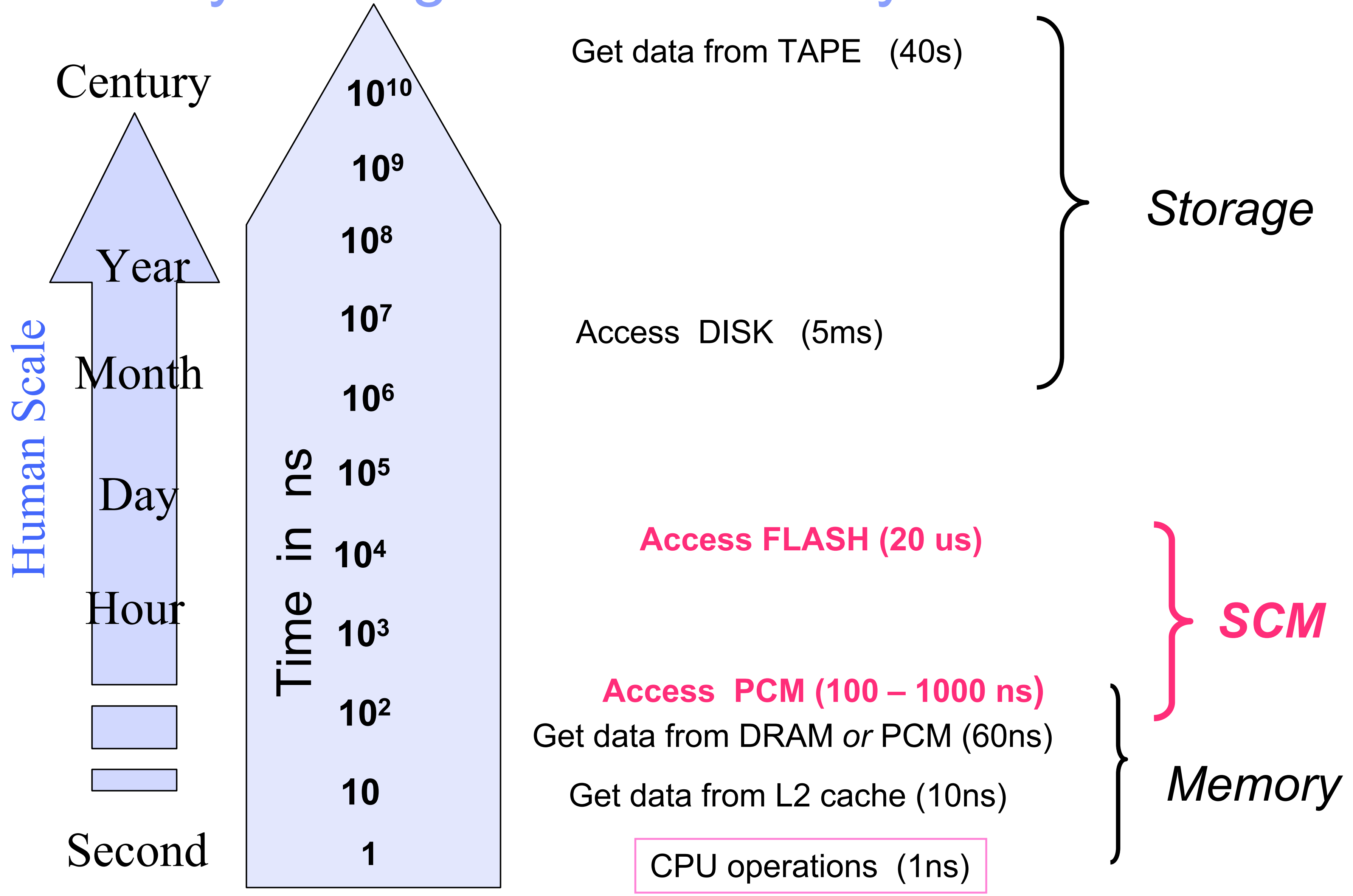


Chart courtesy of Dr. Chung Lam IBM Research

Updated version of plot from IBM Journal R&D article

© 2009 IBM Corporation

Memory/Storage Stack Latency Problem



Get data from TAPE (40s)

Access DISK (5ms)

Storage

Access FLASH (20 us)

Access PCM (100 – 1000 ns)

SCM

Get data from DRAM *or* PCM (60ns)

Get data from L2 cache (10ns)

CPU operations (1ns)

Memory

© 2009 IBM Corporation

Speed and Price Comparisons

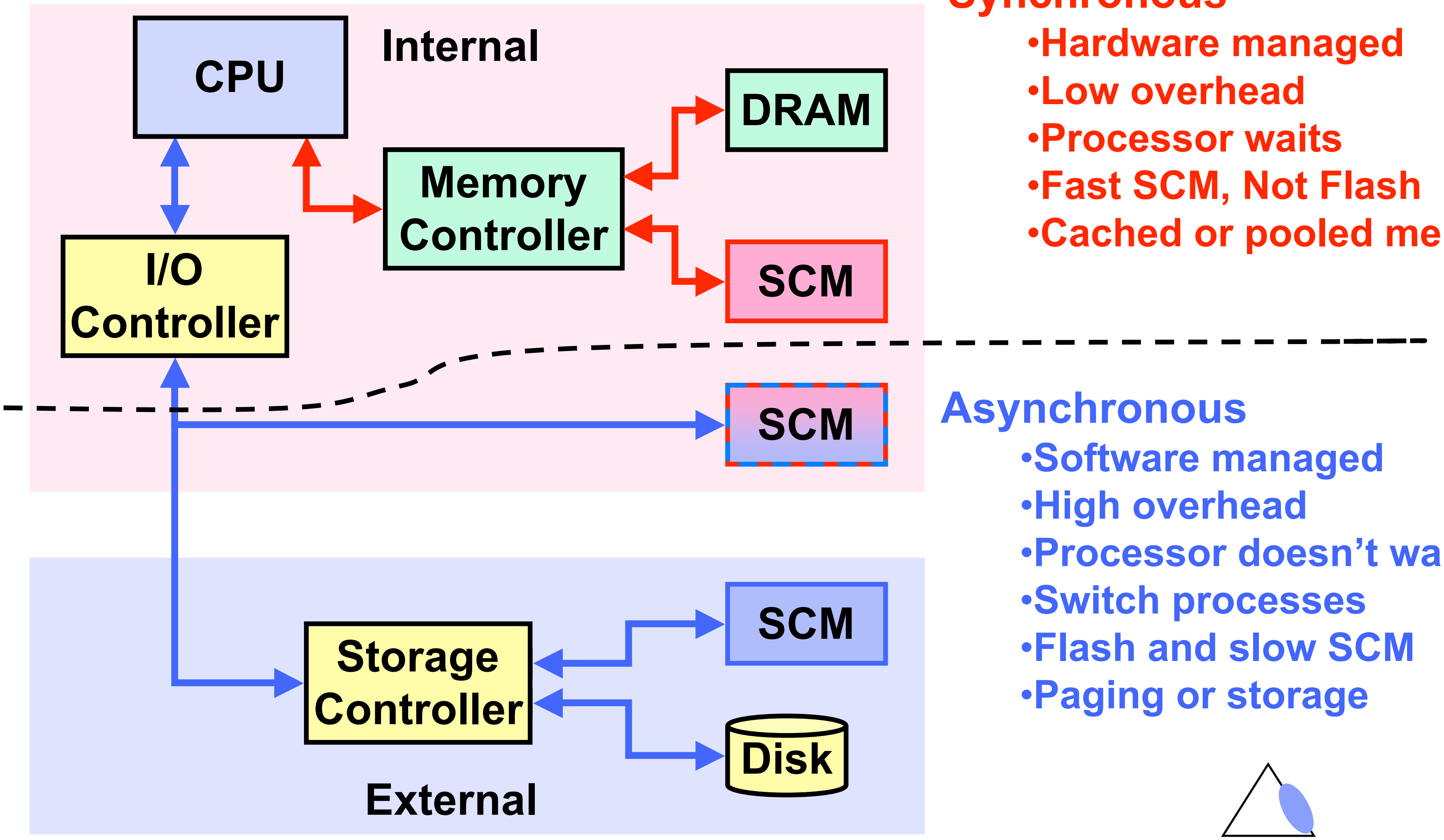
	Read Access Time	Max BW R/W	Power/GB (Max BW)	Power/GB (Idle)	Price 2009-2010
DIMM DDR3	80 – 200 ns	10GB/s	2W/GB	1.1W/GB .125W/GB (STR)	$75 - 80 / GB
SLC Flash Sata DIMM	15 - 125 us	~250MB/s	0.05W/GB	0.003W/GB	$ 3.5 – 4 / GB
MLC Flash Sata DIMM	15 - 125 us	~250MB/s	0.05W/GB	0.003W/GB	$ 1.5 – 2 / GB
SSD SLC Flash	> 20 us	300/145MB/s	0.05W/GB	0.003W/GB	$24 – 35 / GB
SSD MLC Flash	> 25 us	100/100MB/s	0.05W/GB	0.003W/GB	$ 8 -- 12 / GB
Enterprise Disk	5 ms <1 ms cache hit	~112MB/s	0.15W/GB	0.07W/GB	$ 0.80 -- 1.50 / GB
Disk SATA	13 ms <1 ms cache hit	~105MB/s	0.075W/GB	0.07W/GB	$ 0.30 – 0.50 / GB

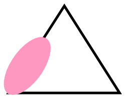
© 2009 IBM Corporation

2013 Possible Device Specs

Parameter	DRAM	PCM-S	PCM-M
Capacity		128 Gbits	16 Gbits
Feature Size F	32nm	32nm	32nm
Effective cell size	6 F^2	0.5 F^2	2 F^2
Read latency	60ns	800ns	300ns
Write latency	60ns	1400ns	1400ns
Retention time	ms	2-10 years	Strongly temp. dependent

Architecture



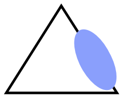


Synchronous

- Hardware managed
- Low overhead
- Processor waits
- Fast SCM, Not Flash
- Cached or pooled memory

Asynchronous

- Software managed
- High overhead
- Processor doesn’t wait
- Switch processes
- Flash and slow SCM
- Paging or storage



© 2009 IBM Corporation

Challenges with SCM

- **Asymmetric performance**
  - Flash: writes much slower than reads
  - Not as pronounced in other technologies
- **Bad blocks**
  - Devices are shipped with bad blocks
  - Blocks wear out, etc.

- **The “fly in the ointment” is write endurance**
  - In many SCM technologies writes are cumulatively destructive
  - For Flash it is the program/erase cycle
  - Current commercial flash varieties
    - Single level cell (SLC) → 10^5 writes/cell
    - Multi level cell (MLC) → 10^4 writes/cell
  - Coping strategy → Wear leveling, etc.

© 2009 IBM Corporation

Shift in Systems and Applications

	▪ ***DRAM – Disk – Tape***	▪ ***DRAM – SCM – Disk – Tape***
Main Memory:	– Cost & power constrained – Paging not used – Only one type of memory: volatile	– Much larger memory space for same power and cost – Paging viable – Memory pools: different speeds, some persistent – Fast boot and hibernate
Storage:	– Active data on disk – Inactive data on tape – SANs in heavy use	– Active data on SCM – Inactive data on disk/tape – Direct Attached Storage?
Applications:	– Compute centric – Focus on hiding disk latency	– Data centric comes to fore – Focus on efficient memory use and exploiting persistence – Fast, persistent metadata

© 2009 IBM Corporation

PCM Use Cases

1. PCM as disk
2. PCM as paging device
3. PCM as memory
4. PCM as extended memory

© 2009 IBM Corporation

Let Us Explore DBMS as Middleware Exploiter of PCM

© 2009 IBM Corporation

PCM as Logging Store – Permits > Log Forces/sec?

- Obvious one but options exist even for this one!
- Should log records be written directly to PCM or

  first to DRAM log buffers and then be forced to PCM (rather than disk)
- In the latter case, is it really that beneficial if ultimately you still want to have log on disk since PCM capacity won't be as much as disk – also since disk is more reliable and is a better long term storage medium
- In former case, all writes will be way slowed down!

© 2009 IBM Corporation

PCM replaces DRAM? - Buffer pool in PCM?

- **This PCM BP access will be slower than DRAM BP access!**
- **Writes will suffer even more than reads!!**
- **Should we instead have DRAM BPs backed by PCM BPs?**

  This is similar to DB2 z in parallel sysplex environment with BPs in coupling facility (CF)

  But the DB2 situation has well defined rules on when pages move from DRAM BP to CF BP
- **Variation was used in SafeRAM work at MCC in 1989**

© 2009 IBM Corporation

Assume whole DB fits in PCM?

- **Apply old main memory DB design concepts directly?**
- **Shouldn't we leverage persistence specially?**
- **Every bit change persisting isn't always a good thing!**
- **Today's failure semantics lets fair amount of flexibility on tracking changes to DB pages – only some changes logged and inconsistent page states not made persistent!**
- **Memory overwrites will cause more damage!**
- **If every write assumed to be persistent as soon as write completes, then L1 & L2 caching can't be leveraged – need to do write through, further degrading perf**

© 2009 IBM Corporation

Assume whole DB fits in PCM? …

- **Even if whole DB fits in PCM and even though PCM is persistent, still need to externalize DB regularly since PCM won't have good endurance!**
- **If DB spans both DRAM and PCM, then**
  - need to have logic to decide what goes where – hot and cold data distinction?
  - persistency isn't uniform and so need to bookkeep carefully

© 2009 IBM Corporation

What about Logging?

- **If PCM is persistent and whole DB in PCM, do we need logging?**
- **Of course it is needed to provide at least partial rollback even if data is being versioned (at least need to track what versions to invalidate or eliminate); also for auditing, disaster recovery, …**

© 2009 IBM Corporation

High Availability and PCM

- **If PCM is used as memory and its persistence is taken advantage of, then such a memory should be dual ported (like for disks) so that its contents are accessible even if the host fails for backup to access**
- **Should locks also be maintained in PCM to speed up new transaction processing when host recovers**

© 2009 IBM Corporation

Start from Scratch?

- **Maybe it is time for a fundamental rethink**
- **Design a DBMS from scratch keeping in mind the characteristics of PCM**
- **Reexamine data model, access methods, query optimizer, locking, logging, recovery, …**
- **What are the killer apps for PCM? For flash, they are consumer oriented - digital cameras, personal music devices, …**

© 2009 IBM Corporation