

The Opportunity Of A Simplified Memory Hierarchy By Non-Volatile Memory

Terry Yin

September 27, 2014

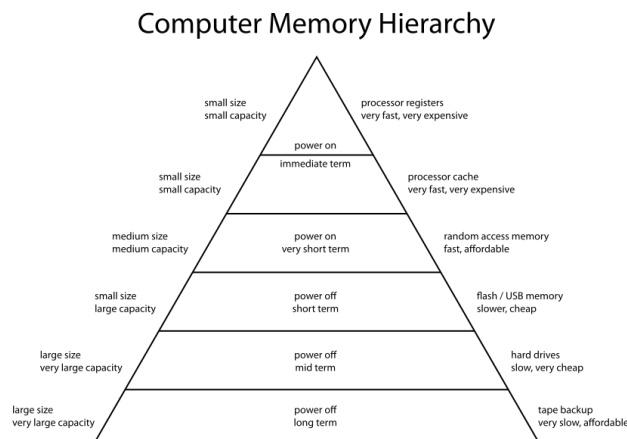
1 Challenge from the traditional computer memory hierarchy

About 30 years ago in one cold morning, I saw my father making fire under his truck to unfrozen the engine and then use the starting handle to start the vehicle. The whole process took like an hour, and it looked scary to me.

The basic memory hierarchy of the modern computer system has two levels, the main memory and the secondary memory.

	Main memory	Secondary memory
speed	fast	slow
persistent	volatile	non-volatile
price	expensive	cheap
address	byte	bulk
energy	inefficient	efficient

But the memory hierarchy in a real computer architecture is rarely just two levels. Because we couldn't get all the good things in one level, we had to resort to the good old tool of computer science, "another level of indirection", [Spinellis \(2007\)](#). So a typical memory hierarchy is like (picture from wikipedia.org):



And these are only the indirections of the hardware. On the software side, there are more levels of indirections, optimizing compilers, segmentation, memory paging, virtual memory, file system, hot spare,

just some examples. These are all the indirections between the processor and where the data is finally located.

The beautiful part of this story is, from a programmer's view there are only two levels, main memory and secondary memory.

The ugly part of this story is, too many levels in the memory hierarchy made the computer architecture over complicated. Some bugs become very tricky, and the compatibility becomes lower. Sub-optimization is very common, which mean the optimization in a part of the system lower the overall performance of the whole system. As [Drepper \(2007\)](#) pointed out:

“Hardware designers have come up with ever more sophisticated memory handling and acceleration techniques—such as CPU caches—but these cannot work optimally without some help from the programmer. Unfortunately, neither the structure nor the cost of using the memory subsystem of a computer or the caches on CPUs is well understood by most programmers.”

Another problem of having only the volatile memory as main memory is just like the engine starter story I shared in the beginning, starting a computer is very slow. Designing a high availability system has been very challenging and introduced too much extra complexity. The volatile memory also needs energy all the time to maintain the data. Given the demand for high availability services and fast developing data centers, the current memory hierarchy is becoming a big problem.

2 The impacts of non-volatile memory

These days in the winter, I just press a button, and my car will start immediately (Well, the fact that I live in Singapore and I don't actually own a car shouldn't matter). It's not because my car has a strong heater, or I have somebody helping me to spin the handle. It's because the liquid doesn't freeze, and the battery doesn't run out that fast now.

2.1 The ideal future

As the improvement in the cars, The emerging fast and cheap non-volatile memory may stop us from adding more levels in the memory hierarchy and eventually simplify the hierarchy.

“Simplicity is the ultimate sophistication.”

Leonardo da Vinci

Having the fast and massive non-volatile memory as the main memory, or the only level of memory, as in [Perez & De Rose \(2010\)](#), we can expect:

- Lower energy consumption
- improve current memory hierarchies
- allow memory systems to continue scaling up
- Devices that has higher availability and longer duration, e.g. normally-off computer, [Ando et al. \(2014\)](#)

Among all these opportunities brought by non-volatile memory, I think the most impacts will come from the simplification of the memory hierarchies. With a simple structure, computer systems will become more reliable and it allows us to build more complicated system based on it.

2.2 Some potential challenge when non-volatile memory goes upward in the memory hierarchy

The ideal one level memory hierarchy with cheap, massive, energy efficient, fast and non-volatile memory will not happen very easy nor very soon. Some concerns also showed like in [Perez & De Rose \(2010\)](#), “Actually, the introduction of these new technologies will probably result in **more complex hierarchies** instead of simpler ones.”

In some situation, volatile memory is an important security feature. With only non-volatile memory, to replicate that feature will add more complexity to the system.

3 Conclusion and what user can expect

The non-volatile memory will go up in the memory hierarchy and eventually simplify the memory hierarchy. Because of the usage of non-volatile memory as the main memory (maybe the only memory), user could expect more available computers. Just like the availability of cars today compares to 30 years in the cold winter. The computers will use a lot less energy than today. The mobile devices will have longer duration. Because of the simplification of the memory hierarchy, user will experience higher quality and more reliable computer systems.

References

- Ando, K., Fujita, S., Ito, J., Yuasa, S., Suzuki, Y., Nakatani, Y., Miyazaki, T. & Yoda, H. (2014), ‘Spin-transfer torque magnetoresistive random-access memory technologies for normally off computing (invited).’, *Journal of Applied Physics* **115**(17), 172607–1 – 172607–6.
URL: <http://search.ebscohost.com.ezproxy.liv.ac.uk/login.aspx?direct=true&db=a9h&AN=95982836&site=eds-live&scope=site>
- Drepper, U. (2007), ‘What every programmer should know about memory’, *Red Hat, Inc* **11**.
- Perez, T. & De Rose, C. A. (2010), Non-volatile memory: Emerging technologies and their impacts on memory systems, Technical report.
- Spinellis, D. (2007), Another level of indirection, in A. Oram & G. Wilson, eds, ‘Beautiful Code: Leading Programmers Explain How They Think’, O’Reilly and Associates, Sebastopol, CA, chapter 17, pp. 279–291.
URL: http://www.dmst.aueb.gr/dds/pubs/inbook/beautiful_code/html/Spi07g.html