

The implementation of caching in computer science is motivated by the desire to make computers run as quickly and effectively as possible, while optimizing its memory and resource allocation with the highest degree of efficiency. With this in mind, caching may be effectively explained through analogy. In a sense, Batman is akin to a user of the memory hierarchy. He starts in the Batcave, which contains everything he needs to fight crime, similar to how a computer operates based off of its knowledge of what is stored in the main memory. When Batman goes out to fight crime in Gotham City, he goes out on the streets with the heavy duty equipment that he is likely to use- loaded into the Batmobile, which is the cache in this sense. Batman knows which gadgets he is most likely to use based on two key ideas: temporal locality and spatial locality. According to temporal locality, Batman can reasonably presume that if he uses a certain gadget a lot when he fights crime, then it follows that he will use it again soon in the future, and so it makes sense to keep such gadgets with him, or cached. Spatial locality reflects the assumption that if you need one item that is stored in the Batcave (main memory), then the items stored near that one item will likely come in handy as well, so it is useful to take physically adjacent tools along in the Batmobile (cache). In other words, if Batman knows he will need a weapon from his weapon shelf, then it makes sense to bring along other weapons nearby on the same weapons shelf for good measure. When he needs to fight crime, Batman hops out of the Batmobile and takes the gear that he knows he needs with him, which he stores in his utility belt- similar to how a computer keeps data in its registers for even easier access. When he needs to immediately use weapons and gadgets, Batman's hands control and operate them in real time, just like how the ALU functions in a computer. In the event that Batman needs a tool that is not currently in his utility belt (registers), he may go back and check the Batmobile (cache) to see if he brought the right items along with him. If the right tools are in the Batmobile (cache), then spatial and temporal locality worked as intended and we have a cache hit. If the Batmobile (cache) does not have the items he needs, Batman must go all the way back to the Batcave (main memory) to retrieve them. If Batman didn't have the Batmobile (cache), he would have to travel all the way back to the Batcave (main memory) every time he needed a tool that he didn't have in his utility belt (registers), slowing him down drastically. With this in mind, we can see how caching plays its role by speeding up the efficiency and processing of modern computers.