Spices as a Caching Metaphor

Suppose that you cook often and like to use many different spices. Because you have so many spices you store all of them in a pantry. You also keep a spice rack in your kitchen where you store recently used spices so you don't have to walk as far to get them from the pantry. And finally, when you need the spices you put them on the counter to use in your recipe.

Although this is just a mundane way of storing spices, it is closely connected to the way computers store information and perform operations. Just as you have a pantry full of spices, a computer has its main memory where it stores any data or operations that it needs. Similarly, the computer has a cache that works similarly to your spice rack. This is where you store recently used spices and where the computer stores recently used data and instructions. Finally there is the counter, where you mix the spices together to create some food. The computer also does this by storing its data in registers instead of a counter and by performing logic operations on the data in its ALU instead of mixing ingredients in your kitchen.

Of course this process of storing spices is not going to be perfect all the time. Because the spice rack is filled with spices you have used most recently, you will find them there often, but sometimes you will have to look in the pantry to find the correct spice. This happens in computers as well and is called a cache hit and cache miss. The computer first checks its cache for data and if it is there then that's a hit, but if it isn't found it checks the main memory and that's a miss.

The way the system is set up increases efficiency most of the time. Instead of having to grab spices from the panty all the time, which is slow, you can instead quickly grab them from the rack. If you have used a spice from the pantry recently, you can instead leave it in the spice rack so if you need it again soon after, you can just grab it again. This would happen quite often if you are cooking similar recipes. This is true for computers too and is called temporal locality. When you use a spice once you are more likely to use it again soon so the computer saves recent data into the cache to be used again quickly. Another method to make the spice rack more useful is to keep spices often used together next to each other in the rack. Like if you find you often use salt and pepper together you can just keep them together. That way if you grab one from the pantry you might as well grab both since you'll probably use it soon as well. This is spatial locality and computers make use of it too. If a computer sees that it used one piece of data, it will also grab data stored next to it since those will probably be used soon too.