Learn about caching, a popular technique to increase computer performance?

In this article, I would like to introduce you to Caching, a very popular technique to increase computer performance. This is a 2-part series of articles, part 1 will explain the basic principles of cache, and common ways to apply cache in practice. Part 2, I will explain the CPU

What is cache?

Caching, or caching is a technique to increase the performance of a computer system. If comparing Cache vs cookies, the Cache technique has many outstanding advantages. It works on one mechanism: storing a copy of frequently accessed data in a location that has a faster access speed than the original location, so the overall operating speed of the system. be increased. The original location is called the backing store, and the location with faster access is called the cache.

Why is this technique used? Everything has its cause. In real life, we often encounter general situations like this:

You have 10 data files, numbered from 1 to 10. The first time you access file number 3, there is a high chance that in the next few accesss you will continue to access file number 3 or one several neighboring files (2 or 4)

The closest example is programming. Suppose you are an Angular front-end developer, your project source code has hundreds of files, but in the actual work process, you rarely open scattered files in different folders, but you often access files in the same directory (eg azo-detail).

Picture 1 of Learn about caching, a popular technique to increase computer performance?

Situations like this are called **Locality of reference.**

Here are some more examples to make it easier for you to visualize:

- Suppose a user goes to the Internet to download a certain image (logo / banner), then after the download is complete, it is highly likely that the image will be retrieved to open it.

- Your bookshelf has 10 books, including a very thick novel and you plan that every day you read 1 hour, it will take 1 month to finish. So from the 2nd day of reading on, the possibility of you picking up that unfinished novel is much higher than other books.

- Suppose you go to GenK and click on a category and see an article in it, then there is a high chance that after reading it, you will continue reading an article in the same category.

So if a system uses cache, the first time there is a request for data (cache miss), the system will load that data from the backing store, and save a copy at the cache. On subsequent requests, the data will be loaded from the cache (cache hit).

Picture 2 of Learn about caching, a popular technique to increase computer performance?

Thus, the larger the cache, the higher the performance, but that means the greater the production cost. How big the cache is is a matter of performance/cost trade-off. But no matter how large the capacity of the cache is, at some point it will be full. In order to continue using it, we must reduce it by removing the data that is least likely to be updated. There are two common algorithms for deciding what data should be removed from the cache:

+ LRU (Least Recently Used) , which means that the most recently accessed data is prioritized to keep. Let's say the cache is full and currently has 3 data fields, data1, data2, data3. The last access was 5 minutes ago, 2 minutes ago and 3 minutes ago, respectively. Now, as we want to save more data4 into the cache, we keep data2 and data3, discard data1.

+ TLRU (Time-aware LRU) , only different from the above algorithm in that, each data field in the cache has an additional parameter that is an expiration time, so when it expires, it will automatically be removed from the cache. .

Picture 3 of Learn about caching, a popular technique to increase computer performance?

For example, the cache is containing data like this, the current time (according to UNIX time) is 1500000100, the data with the key equal to 431 will be removed from the cache.

This way has many benefits:

+ Ensure that a certain type of data will be reloaded on a regular basis, and this cycle is up to you to decide. For example, an international e-commerce site that lists selling prices in different currencies can use this to update hourly rates.

+ When there are many internal access situations for a certain data set, we can adjust this expiration time to decide which one to save longer, and which one to drop earlier.

Ways to update data when using cache

In fact, there are many situations where we want to update data, there are many different ways to handle it, depending on the system. In general, there are 2 groups of ways:

Update data in both cache and backing store.

This group is divided into two ways:

Write back : Update the data in the cache first, then proceed to update the data in the backing store. This way has the advantage of optimizing performance, but there is a risk of data loss if the system has a problem without updating the data in the backing store.

Write through : Update data in both cache and backing store in parallel. This way will ensure that the data in the cache and the backing store are always synchronized with each other, but will have to trade off in terms of system performance.

Only update data in the backing store

Invalidate cache : the system will send a request to invalidate any data in the cache, even all of it. On subsequent visits, the data will load from the updated backing store.

Versioning : creates a different version than the one in the cache. This way is very easy to see when coding web, programmers often append a parameter after the path that references the css file, or static images like the logo file. When editing the css file, or changing the logo, the programmer only needs to change the $version variable (manually or by command) and the user will see the latest update. The old data in the cache is still saved, but here, we are requesting data with a different key that does not exist in the cache.

Picture 4 of Learn about caching, a popular technique to increase computer performance?

Popular applications of cache

Page cache

When a computer program is opened, the computer can load all or part of the source code (compiled) into RAM for easy data access because of the speed of accessing data from RAM. is also greater than the speed of accessing data from the hard drive. At this time, RAM is the cache and the computer's hard drive is the backing store.

Web browser cache

When you visit a website for the first time, the static resources of that website such as css files, js files, static images (logos, icons) will be saved on your computer. The next time you visit, your machine will load these resources from the machine, not from the server anymore, saving bandwidth and speeding up page loading.

Data cache is the result of a heavy computation

Backing stores are usually places with large data storage capacity but low access speed. However, the backing store can also be understood as heavy tasks with long computation time.

Let's say you design a puzzle program for thousands of participants. Before starting to solve a puzzle, each player will be given one data: the average time to solve this puzzle. This number is calculated by taking the median of the array of previous players' completion times. Thus, to come up with this number, the system will have to sort the other time array, and then find the median from there. The array sort operation is a task with a complexity of O(n log n) (where n is the number of elements of the array).

Thus, if each game, the system has to calculate this number, the performance of the system will be greatly affected. The solution is to cache this result and send it to the user, because anyway this number is for reference only, not essential for the player. The calculation to update this result can be done once at the end of the day.

CDN – Content Delivery Network

GenK is a leading technology website in Vietnam with tens of millions of monthly reads from all over Vietnam. Assuming that GenK's server is located in Hanoi, the access speed of readers in Ho Chi Minh City will be lower than the access speed of readers in Da Nang. In general, the closer to the data source, the faster the data retrieval speed will be. So, if we store a copy of GenK at a server in Binh Duong, users in Ho Chi Minh City will read GenK at a much faster rate. That is the basic operating principle of CDN service providers. When a user uses a product that loads resources from a CDN, the service checks where the resource requests are coming from, and then returns the resources stored in the cache servers closest to them. the location where those requests are generated.

Picture 5 of Learn about caching, a popular technique to increase computer performance?