Caching

Caching

- Caching is at very least OS adjacent
- Heavily used in OS
  - Where?
- And often in ‘real software’

Caching

- Caching is at very least OS adjacent
- Heavily used in OS
  - Where?
- <here the professor pauses and pretends you all can answer>
- <and you think deeply>

Caching

- Caching is at very least OS adjacent
- Heavily used in OS
  - Where?
- <here the professor pauses and pretends you all can answer>
- <and you think deeply>
- <about how much you look forward to a question like this on the next Quiz in April> :-)

Caching

- Caching is at very least OS adjacent
- Heavily used in OS
  - Where?
- Memory page locations
  - TLB is a type of Cache
- Memory pages themselves
  - In L1, L2 etc hardware caches
- Memory maps of files that are read in
- So used a lot in OS

Caching

- So what is caching?

Caching

- So what is caching?
  - I’ll pause here while you furiously google

Caching

- So what is caching?
  - According to AWS (amazon web services)
  - “In computing, a cache is a high-speed data storage layer which stores a subset of data, typically transient in nature, so that future requests for that data are served up faster than is possible by accessing the data’s primary storage location. Caching allows you to efficiently reuse previously retrieved or computed data.”

Caching

- We talk about caching all the time – but rarely show you how you might do it I think
- So where do we cache in software development?

Web cache

- Everyone is familiar with the web cache that {favorite browser} keeps
  - Firefox/chrome/edge/safari
- Recently visited web pages and their assets (images etc) saved locally on disk
- What does this save you

Web cache

- Everyone is familiar with the web cache that {favorite browser} keeps
  - Firefox/chrome/edge/safari
- Recently visited web pages and their assets (images etc) saved locally on disk
- What does this save you
- …. I’ll wait ……

Web cache

- Everyone is familiar with the web cache that {favorite browser} keeps
  - Firefox/chrome/edge/safari
- Recently visited web pages and their assets (images etc) saved locally on disk
- What does this save you
- …. I’ll wait ……
- …………..

Web cache

- Everyone is familiar with the web cache that {favorite browser} keeps
  - Firefox/chrome/edge/safari
- Recently visited web pages and their assets (images etc) saved locally on disk
- What does this save you
- Don’t have to download all of it again on internet.
  - If you have 5g or fiber – no big deal
  - If you are in southeastern Ohio and have 2G cellular or Rural Maine and you have dish internet – much bigger deal.

What other kinds of cache?

- What other kinds of cache are used in software dev a lot?

What other kinds of cache?

- What other kinds of cache are used in software dev a lot?
  - Cloudflare is a giant web cache
    - Basic idea: you have one server somewhere
    - Cloudflare has many servers – and your users seamlessly connect to a cloudflare mirror near you
      - Your site is just as fast in New York as it is New Zealand
      - St. Louis and Sao Paolo
      - And so on.
  -

What other kinds of cache?

- What other kinds of cache are used in software dev a lot?
  - Cloudflare is a giant web cache
    - Basic idea: you have one server somewhere
    - Cloudflare has many servers – and your users seamlessly connect to a cloudflare mirror near you
      - Your site is just as fast in New York as it is New Zealand
      - St. Louis and Sao Paolo
      - And so on.
  - Redis (https://redis.io/docs/about/) is another caching setup
    - Mirror your database or other large data store in memory
    - So we don’t have to pay the price for going to disk.

Implementing a simple cache.

- First really simple cache: the lookup table
  - Look up table basics:
    - You calculate a result – that perhaps takes some time to calculate
    - Store that result in an array
    - When we need the result, check if it is in the array already, if so use it, if not calculate it and then add it to the table

Revising an old ‘friend’

- When you were first learning recursion

Revising an old ‘friend’

- When you were first learning recursion
  - Yes I saw half of you shudder – even through time delayed video

Revising an old ‘friend’

- When you were first learning recursion
  - You likely looked at the first two over simplified examples
  - Fibonacci and factorial
  -

Revising an old ‘friend’

- When you were first learning recursion
  - You likely looked at the first two over simplified examples
  - Fibonacci and factorial
  - For those of you who blotted out middle/high school math
  - Factorial: n! = n*(n-1)*(n-2)*(n-3)...*1
    - Eg: 5! = 5*4*3*2*1
  - Fibonacci :
    - fib(1) → 1
    - fib(2) → 1
    - fib(n) → fib(n-1) + fib(n-2)
    - Eg 1,1,2,3,5,8,13…..

Let’s remind ourselves and try it out

- We will use this function to drive our recursive functions
- Let’s type it in and I’ll narrate in case something isn’t familiar.

```
func main() {
        reader := bufio.NewReader(os.Stdin)
        fmt.Print("calculate which fibonacci number:")
        text, err := reader.ReadString('\n')
        if err != nil {
        log.Fatal("error reading - giving up ", err)
        }
        text = strings.TrimSpace(text)
        num, err := strconv.Atoi(text)
        if err != nil {
        log.Fatal("you didn't put in an int - giving up ", err)
        }
        fmt.Println("Fib of ", num, " is ", fib(num))
}
```

Comparing the performance

- Let’s look at the performance differences
  - Some of you remember this from your early recursion
  - Some may have blocked it out.
- And then your instructor likely said some thing like,
  - “you could use a lookup table to cache the results”
  - So let’s look at how you could do that too.