

# Efficiency: locality; OS intro

---

COSC 208, Introduction to Computer Systems, 2022-04-14

## Other caches

- Domain Name System (DNS) caches
  - DNS is used to translate domain names (e.g., `portal.colgate.edu`) into Internet Protocol (IP) addresses (e.g., `149.43.134.29`)
  - DNS entries—i.e., mappings from domain names to IP addresses—can be cached by a web browser, an operating system, and a recursive resolver
    - A recursive resolver is a DNS server within a network that receives DNS queries from clients and queries other DNS servers on the client's behalf in order to locate the desired DNS entry
  - Why are DNS entries cached?
    - Contacting a recursive resolver (and other DNS servers) is much slower than access a solid state drive (SSD)
    - Spatial locality — users often visit subdomains of a domain, e.g., `portal.colgate.edu`, `moodle.colgate.edu`, and `cs.colgate.edu` are subdomains of `colgate.edu`
    - Temporal locality — users often visit the same domain repeatedly
    - Mappings from domain names to IP addresses change infrequently —> DNS entries can be cached for hours or days
- Content distribution networks (CDNs)
  - Collection of geographically distributed servers that delivery content (e.g., streaming videos) to users
  - User's computers contact a server that is "nearby"
    - Ideally measured in terms of latency, which is a function of geographic distance, network routes, and network load
    - Analogy: time it takes to drive somewhere is a function of geographic distance, the route you take, and the amount of traffic on the road
  - CDN servers fetch and cache content from origin servers
  - Popular content (e.g., image from the front page of the NY Times) is more likely to already be cached

## Warm-up

Last worksheet Q3

## Optimizing loops for locality

Q1: Modify the following function to improve spatial locality

```
int *hundreds() {
    int *nums = malloc(sizeof(int) * 1000);
    for (int i = 0; i < 100; i++) {
        for (int j = 0; j < 1000; j+= 100) {
            nums[i+j] = i;
        }
    }
}
```

Q2: Modify the following function to improve temporal locality

```
int odds(int *nums, int length) {
    for (int i = 0; i < length; i++) {
        nums[i] = nums[i] % 2;
    }
    int count = 0;
    for (int j = 0; j < length; j++) {
        count += nums[j];
    }
    return count;
}
```

Q3: Modify the following function to improve spatial locality

```
void multiplication(int grid[][], int rows, int cols) {
    for (int c = 0; c < cols; c++) {
        for (int r = 0; r < rows; r++) {
            grid[r][c] = c * r;
        }
    }
}
```

Q4: Modify the following function to improve temporal locality

```
long stdev(int *nums, int length) {
    long sum = 0;
    for (int i = 0; i < length; i++) {
        sum += nums[i];
    }
    int mean = sum / length;
    sum = 0;
    for (int j = 0; j < length; j++) {
        int diff = nums[j] - mean;
        sum += diff * diff;
    }
    mean = sum / length;
    return sqrt(mean);
}
```