Team Project Presentation

ccgroupjk

Jiannan Jiang, Yijia Zhang, Huiyi Zhang

April 28, 2022

15-619 (Spring 2022)

Web Framework

- Language: Java
- Web Framework: Vert.x
- Monolithic architectures

Live test	Cluster	Web tier	Effective RPS / Latency(ms) / CPU utilization		
			M1	M2	M3
Phase 2	K8s	5 m6i.large	**216224.50/2.17/ ~100%**	**56709.80/7.34/ ~100%**	22631.08/17.70/ ~100%
Phase 3	EKS managed nodegroups	5 c6g.xlarge	**285291.98/1.34 (~ 60%-80%)**	**120357.38/3.38/ ~100%**	20734.08/19.26/ ~40%
Phase 3 live test: mixed queries			**91117.49/2.38**	**40359.31/2.67**	**13823.66/3.74**

General Rules of Thumb

- Only one initialization of some global constants / class / connections and **reuse**.
  - Team name, big int to mod in RSA
  - Zip deflator/inflator
  - JDBC connections
- Avoid string ops
  - Have predefined strings, as "memcpy" is cheaper.
  - Use bytes when possible.
- Use bit operation
- Precompute things that has a determined effect
- *Better algorithm design (which we do not have time to discuss today).

M1 Specific Tricks

- Precompute things that has a determined effect

  - logistic mapping is a constant string to be XORed with

  - For both QR of size 21 and 25, the default output is always the same. (avoid filling `11101100 00010001` !)

  - When filling the response, edit a copy of the default output based on an index map based on the detected matrix size and rotation.

- Code example:

```java
public static final String encode_default_mat_21 = "01100110110110010010101110000000010110
public static final String encode_default_mat_25 = "01100110111000010110100001010011101100
public static final String lm_flipped_bin_list = "10011000110110101101011110010001010001011

public static final List<Integer> idx_to_pos_21_0 = Arrays.asList(440,439,419,418,398,397,
public static final List<Integer> idx_to_pos_21_1 = Arrays.asList(420,399,421,400,422,401,
public static final List<Integer> idx_to_pos_21_2 = Arrays.asList(0,1,21,22,42,43,63,64,84
public static final List<Integer> idx_to_pos_21_3 = Arrays.asList(20,41,19,40,18,39,17,38,

public static final List<Integer> idx_to_pos_25_0 = Arrays.asList(624,623,599,598,574,573,
public static final List<Integer> idx_to_pos_25_1 = Arrays.asList(600,575,601,576,602,577,
public static final List<Integer> idx_to_pos_25_2 = Arrays.asList(0,1,25,26,50,51,75,76,10
public static final List<Integer> idx_to_pos_25_3 = Arrays.asList(24,49,23,48,22,47,21,46,
```
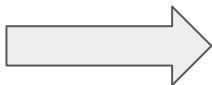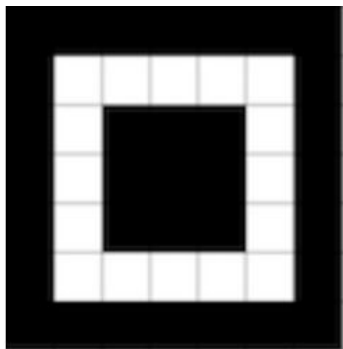
M1 Specific Tricks

- **Use bit operation!**
- Code example:

  During decoding, brute force checking if the matrix matches the 8x8 block below
  starting at position (i,j):

127
65
93


```java
for (int i = 0; i < 26; i++) {
    for (int j = 0; j < 26; j++) {
        // bit op
        if (((bin_int_list.get(i  )>>j)&127) != 127)continue;
        if (((bin_int_list.get(i+1)>>j)&127) != 65)continue;
        if (((bin_int_list.get(i+2)>>j)&127) != 93)continue;
        if (((bin_int_list.get(i+3)>>j)&127) != 93)continue;
        if (((bin_int_list.get(i+4)>>j)&127) != 93)continue;
        if (((bin_int_list.get(i+5)>>j)&127) != 65)continue;
        if (((bin_int_list.get(i+6)>>j)&127) != 127)continue;
```

(Note that the matrix to be decoded is stored as a list of ints, as strings are expensive)

M2 Specific Tricks (useful but not completely original)

- **Setup prespecified strings and use bit operation!**

- Use bytes when possible.

- Code example: cchash

```
73
74      // string from bytes method quoted from :
75      // https://stackoverflow.com/questions/5823290/how-best-to-convert-a-byte-array-to-a-string-buffer
76      public static final byte[] HEX_ARRAY = "0123456789abcdef".getBytes(StandardCharsets.US_ASCII);
77      public static String ccHash(String str, MessageDigest md)
78          throws DataFormatException, UnsupportedEncodingException, NoSuchAlgorithmException{
79          byte[] results_SHA256 = md.digest(str.getBytes("UTF-8"));
80          byte[] hexChars = new byte[8];
81          for (int j = 0; j < 4; j++) {
82              int v = results_SHA256[j] & 0xFF;
83              hexChars[j * 2] = HEX_ARRAY[v >>> 4];
84              hexChars[j * 2 + 1] = HEX_ARRAY[v & 0x0F];
85          }
86          return new String(hexChars, StandardCharsets.UTF_8);
87      }
```

M2 Specific Tricks

- **Check validity whenever has the chance, and terminate early if we already know the request is invalid.**

The End

Jiannan Jiang, Yijia Zhang, Huiyi Zhang