

Homework 8

Chinese Name: _____

Pinyin Name: _____

Student ID: _____

E-Mail prefix: _____

10 1. Put T (True) or F (False) for each of following statement. [2 points each]

- (a) () Request-Level Parallelism (RLP) allows independent requests to execute simultaneously across multiple cores or servers.
- (b) () In a WSC, if requests frequently involve read-write conflicts, Request-Level Parallelism (RLP) can still significantly improve performance.
- (c) () ECC memory can automatically correct single-bit errors.
- (d) () ECC memory can eliminate all types of hardware memory errors.
- (e) () Using ECC memory in large-scale servers (e.g., WSCs) can improve system reliability.

26 2. **MapReduce [26 points]****Given:**

- Example access log:

```
user1 /home
user2 /products
user3 /home
user4 /products
user5 /contact
user6 /about
```

- Each line contains a **User ID** and a **URL**.
- Your goal: Count the total visits per URL and output the top 2 most visited URLs.

Tasks:

1. **Map Phase Output [8 points]:** List the key-value pairs $(URL, 1)$ generated by the Map phase.

Answer format example: $(URL1, 1), (URL2, 1), \dots$

Fill in your answer below:

2. **Shuffle Phase Aggregation [8 points]:** Group all values of Map outputs by URL (key).

Answer format example: $URL1 : [1, 1], URL2 : [1, 1, 1], \dots$

Fill in your answer below:

3. **Reduce Phase Output and Top-2 URLs [10 points]:** Sum counts for each URL and sort them.

Answer format example: $URL1 : 2, URL2 : 3$

Fill in your answer below:

Top-2 most visited URLs: _____, _____

28

 3. ECC [28 points]
Given:

- Data bits (8-bit): $d_1d_2d_3d_4d_5d_6d_7d_8 = 0\ 1\ 1\ 0\ 0\ 1\ 0\ 1$
- Parity bit insertion positions: all powers of 2 (bits 1, 2, 4, 8)
- Parity rule: **even parity** (each parity bit is set so that the total number of 1's in its coverage, including itself, is even)
- Total codeword length (don't consider the extended bit covering all bits.): 12 bits (8 data bits + 4 parity bits)

Tasks:

1. **Insert parity bits:** Write the 12-bit codeword. Fill in the value of each bit and mark each position as Parity (P) or Data (D) yourself. [8 points]

| Bit position | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|-----------------|---|---|---|---|---|---|---|---|---|----|----|----|
| Value | | | | | | | | | | | | |
| Type (mark P/D) | | | | | | | | | | | | |

2. **Complete the ECC coverage table:** For each parity bit, mark which bits it covers (use “√” to indicate coverage).

| Bit Position | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|--------------|---|---|---|---|---|---|---|---|---|----|----|----|
| p1 covers | | | | | | | | | | | | |
| p2 covers | | | | | | | | | | | | |
| p4 covers | | | | | | | | | | | | |
| p8 covers | | | | | | | | | | | | |

3. **Single-bit error detection and correction:** Suppose the received codeword is:

1 0 0 1 1 1 0 0 0 1 1 1

Fill in the Result column: use “√” if the parity check passes, or “X” if the parity check detects an error. After detecting the error, correct the codeword and write the corrected value.

| Bit check | Result |
|-------------|--------|
| Bit 1 check | |
| Bit 2 check | |
| Bit 4 check | |
| Bit 8 check | |

4. **Double-bit error:** Suppose a double-bit error occurs:

1 0 1 1 1 1 0 0 0 1 1 1

Fill in the Result column: use “√” if the parity check passes, or “X” if the parity check detects an error. Can we determine which bit(s) are wrong? Briefly explain your answer.

| Bit check | Result |
|-------------|--------|
| Bit 1 check | |
| Bit 2 check | |
| Bit 4 check | |
| Bit 8 check | |