1 Pre-Check
This section is designed as a conceptual check for you to determine if you conceptually understand and have any misconceptions about this topic. Please answer true/false to the following questions, and include an explanation:

For more information on higher level vs. lower level, visit https://en.wikipedia.org/wiki/High-and_low-level

The higher the PUE the more efficient the datacenter is.

False. The ideal PUE is 1.0.

Hamming codes can detect any type of data corruption.

False. They cannot detect all three bit errors.

All RAID levels improve reliability.

False. Raid 0 actually decreases reliability.

2 Hamming ECC
Recall the basic structure of a Hamming code. We start out with some bitstring, and then add parity bits at the indices that are powers of two (1, 2, 8, etc.). We don’t assign values to these parity bits yet. Note that the indexing convention used for Hamming ECC is different from what you are familiar with. In particular, the 1 index represents the MSB, and we index from left-to-right. The \(i\)th parity bit \(P\{i\}\) covers the bits in the new bitstring where the index of the bit under the aforementioned convention, \(j\), has a 1 at the same position as \(i\) when represented as binary. For instance, 4 is \(0b100\) in binary. The integers \(j\) that have a 1 in the same position when represented in binary are 4, 5, 6, 7, 12, 13, etc. Therefore, \(P4\) covers the bits at indices 4, 5, 6, 7, 12, 13, etc. A visual representation of this is:

![Hamming ECC Diagram](https://en.wikipedia.org/wiki/hamming_code)

Source: https://en.wikipedia.org/wiki/Hamming_code
2.1 How many bits do we need to add to 0011₂ to allow single error correction?

\[ m \] parity bits can cover bits 1 through \( 2^m - 1 \), of which \( 2^m - m - 1 \) are data bits. Thus, to cover 4 data bits, we need 3 parity bits.

2.2 Which locations in 0011₂ would parity bits be included?

Using \( P \) to represent parity bits: PP0P011₂

2.3 Which bits does each parity bit cover in 0011₂?

Parity bit 1: 1, 3, 5, 7
Parity bit 2: 2, 3, 6, 7
Parity bit 3: 4, 5, 6, 7

2.4 Write the completed coded representation for 0011₂ to enable single error correction. Assume that we set the parity bits so that the bits they cover have even parity.

100011₂

2.5 How can we enable an additional double error detection on top of this?

Add an additional parity bit over the entire sequence.

2.6 Find the original bits given the following SEC Hamming Code: 0110111₂. Again, assume that the parity bits are set so that the bits they cover have even parity.

Parity group 1: error
Parity group 2: okay
Parity group 4: error
To find the incorrect bit’s index, we simply sum up the indices of all the erroneous bits.
Incorrect bit: 1 + 4 = 5, change bit 5 from 1 to 0: 0110011₂
0110011₂ → 1011₂

2.7 Find the original bits given the following SEC Hamming Code: 1001000₂

Parity group 1: error
Parity group 2: okay
Parity group 4: error
Incorrect bit: 1 + 4 = 5, change bit 5 from 1 to 0: 1001100₂
1001100₂ → 0100₂

3 RAID

3.1 Fill out the following table:

<table>
<thead>
<tr>
<th>RAID 0</th>
<th>Configuration</th>
<th>Pro/Good for</th>
<th>Con/Bad for</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Split data across multiple disks</td>
<td>No overhead, fast read / write</td>
<td>Reliability</td>
</tr>
<tr>
<td>RAID 1</td>
<td>Mirrored Disks: Extra copy of data</td>
<td>Fast read / write, Fast recovery</td>
<td>High overhead $\rightarrow$ expensive</td>
</tr>
<tr>
<td>RAID 2</td>
<td>Hamming ECC: Bit-level striping, one disk per parity group</td>
<td>Smaller overhead</td>
<td>Redundant check disks</td>
</tr>
<tr>
<td>RAID 3</td>
<td>Byte-level striping with single parity disk.</td>
<td>Smallest overhead to check parity</td>
<td>Need to read all disks, even for small reads, to detect errors</td>
</tr>
<tr>
<td>RAID 4</td>
<td>Block-level striping with single parity disk.</td>
<td>Higher throughput for small reads</td>
<td>Still slow small writes (A single check disk is a bottleneck)</td>
</tr>
<tr>
<td>RAID 5</td>
<td>Block-level striping, parity distributed across disks.</td>
<td>Higher throughput of small writes</td>
<td>The time to repair a disk is so long that another disk might fail in the meantime.</td>
</tr>
</tbody>
</table>

4 Warehouse-Scale Computing

Sources speculate Google has over 1 million servers. Assume each of the 1 million servers draw an average of 200W, the PUE is 1.5, and that Google pays an average of 6 cents per kilowatt-hour for datacenter electricity.

4.1 Estimate Google’s annual power bill for its datacenters.

$$1.5 \cdot 10^6 \text{ servers} \cdot 0.2\text{kW/server} \cdot 0.06\text{$/kW-hr} \cdot 8760\text{ hrs/yr} \approx $157.68\text{ M/year}$$

4.2 Google reduced the PUE of a 50,000-machine datacenter from 1.5 to 1.25 without decreasing the power supplied to the servers. What’s the cost savings per year?

$$\text{PUE} = \frac{\text{Total building power}}{\text{IT equipment power}} \implies \text{Savings} \propto (\text{PUE}_{old} - \text{PUE}_{new}) \cdot \text{IT equipment power}$$

$$(1.5 - 1.25) \cdot 50000\text{ servers} \cdot 0.2\text{kW/server} \cdot 0.06\text{$/kW-hr} \cdot 8760\text{hrs/yr} \approx $1.314\text{ M/year}$$