**Samonorom Chum**

**Thursday May 8, 2025**

**EECE 4811/5811 Operating System Spring 2025**

**Professor Tseng**

**HW7**

1. **RAID Simulation in Go (6 pt)**

* **Overview**: You will simulate various RAID levels using Go. Rather than working with actual hardware, you will use regular files to represent physical disks. This approach allows you to focus on the logic of RAID levels and understand the trade-offs between performance and redundancy.
* **Objectives/Concepts**: By the end of this assignment, you should be able to:
* Understand the key differences and trade-offs between RAID 0, 1, 4, and 5.
* Implement data striping, mirroring, and parity logic.
* Simulate physical disks using files.
* Measure and analyze read/write performance of different RAID levels.
* **Specification of your RAID implementation:**
* Use **five files** (e.g., disk0.dat to disk4.dat) to simulate physical disks.
* Each disk file should support **basic read and write** operations at specified block offsets.
* **Define a block size** (e.g., 512 bytes or 4 KB) and maintain a logical mapping from logical blocks to physical locations.
* You should implement **RAID 0, 1, 4, and 5 – use XOR for RAID 5**
* Interface – Each RAID level should implement this interface.

**type RAID interface {**

**Write(blockNum int, data []byte) error**

**Read(blockNum int) ([]byte, error)**

**}**

* Note: Ensure files are **flushed (fsync)** properly to simulate real disk write delays.
* Note1: you don’t need to worry about edge cases, such as corrupted files, disk failures, read/write beyond bounds.
* **Benchmarking and Evaluation**:
* Write a benchmark tool that:

1. Writes a large chunk of data (e.g., 100 MB) in blocks.
2. Reads back the data in blocks.
3. Measures and reports the time taken for each operation (total and per block).
4. Run the benchmark for all four RAID levels using identical data sizes and workload

* Plot or print a summary comparing, under different workload and configuration:

1. Write performance
2. Read performance
3. Effective storage capacity

* Compare the performance numbers you have and the RAID analysis in the textbook. Does the trend match? If not, briefly identify potential reasons.

**Report:**

1. **Running the File:**

* Open Ubuntu (I used WSL and Ubuntu 24.04) and go to the “hw7\_raid\_sim/” folder.
* Inside the folder “hw7\_raid\_sim”
* Enter the command “go run . –level=raid[Chosen #] --disks=[Chosen #] –size=[Choosen #]”
  + For raid, you can replace the “[chosen #] next to “raid” with a 0, 1, 4, or 5 for which level of raid you want.
  + For disks, you can replace the “[chosen #] next to “disks” for the number of disks you want.
  + For size, you can replace the “[chosen #] next to “size” for the size you want in MB.
  + **Example:**

**“go run . --level=raid5 --disks=5 --size=100”**

1. **Project Structure:**

* The directory structure is as follows:

hw7\_raid\_sim/

├── .gitignore

├── go.mod

├── main.go

├── benchmark/

└── benchmark.go

├── data/

├── disk0.dat

├── disk1.dat

├── disk4.dat

├── disk5.dat

├── disk/

└── disk.go

├── raid/

├── interface.go

├── raid0.go

├── raid1.go

├── raid4.go

├── raid5.go

└── raid\_test.go

└── utils/

└── timer.go

1. **RAID Implementations:**

The project supports the following RAID levels:

* + RAID 0: Striping without parity or redundancy.
  + RAID 1: Mirroring across all disks.
  + RAID 4: Block-level striping with a dedicated parity disk.
  + RAID 5: Block-level striping with distributed parity.

1. **Benchmarking Tool**

The benchmark tool writes and reads a user-defined amount of data to each RAID level. It measures the total time taken for writes and reads, helping assess the performance of each implementation.

* + **Need to add measurement**
  + **Need to add plot**
  + **Need to answer questions.**

1. **Conclusion**

This RAID simulation and benchmarking tool allows the user to run RAID0 , RAID1, RAID4, and RAID5 (using XOR). The measurement shown

1. **References:**
2. ChatGPT, “Assistance with HW7 Raid Simulation in Go,” OpenAI, personal communication, May. 7, 2025.
3. **Prompts Used with ChatGPT:**
   1. Uploaded in ChatGPT, and used the prompt “help me get started with HW”
   2. Prompt ChatGPT to help uninstall my old Ubuntu and WSL software, and install WSL and Ubuntu 24.04.
   3. I then prompted ChatGPT to “help me download gedit, Go, and other needed software inside Ubuntu”.
   4. I then prompted ChatGPT to “help me use VS Code within Ubuntu”.
   5. I prompted ChatGPT to “help me get started again”, which it helped me develop skeleton codes.