CS 402: Introduction to Advanced Studies II

Homework-1

Kuldeep Patel | A20540269
kpatel181@hawk.iit.edu

Question 1
The Code written for question-1 is in Python. In with I have use matplotlib library.
As per the requirement of the question two graphs have been plotted
a)

Axes:
- X-Axis: Represents the Address.
- Y-Axis: Represents the Frequency of Address.

- **Range Analyzed**: (4394199, 4397400)
- **Activity Observed**:
  - Reads: 0
  - Writes: 0
  - Fetch: 164188
- **Observations**:
  - The majority of the frequencies lie near 7463.0.
  - It is observed that all the frequencies of addresses are above 7000, indicating a high frequency of access in this range.

Comment on cc1.din The absence of reads and writes and a high number of fetches suggest that the **cc1.din** dataset is primarily used for fetching data. This could indicate a scenario where data is being loaded into memory, possibly for read-only operations or processing where the data doesn't need to be modified.

2. spice.din

Axes:
- X-Axis: Represents the Address.
- Y-Axis: Represents the Frequency of Address.

- **Range Analyzed**: (268519512, 268519880)
- **Activity Observed**:
  - Reads: 15490
  - Writes: 2904
  - Fetch: 0
- **Observations**:
  - The plot shows a regular pattern where a high frequency (above 700) is observed after every 7 addresses with a lower frequency (around 100). The frequency of

operations is moderate (ranging from 0 to 800), yet the graph is not densely populated.

- There is a noticeable sudden increase in frequency after every 800 addresses.

Comment on spice. din: shows a substantial number of reads and writes, but no fetch operations. This indicates active reading from and writing to memory, typical of applications that are processing and modifying data. The pattern of access might be indicative of the algorithmic structure or data access patterns inherent in the application using this dataset. with as many as 15,490 read operations in this range, it becomes evident that this section of the memory is utilized for reading data. The occurrence of 2,904 write operations, although lower than the reads, indicates that the region is not exclusively for reading; it also accommodates occasional writes. The lack of fetch operations within this range implies that the CPU does not retrieve instructions from this particular section of the memory. This indicates that the CPU is not executing code from this area but primarily using it for data storage and access.

Question 2

First Matrix Multiplication

System-1 Performance		System-2 Performance	
Integer Multiplication	Double Multiplication	Integer Multiplication	Double Multiplication
1.076500542	1.125281959	2.8943196	7.1409719
1.081239542	1.087965792	2.9137231	8.0282349
1.082701375	1.2020625	3.1421749	8.1077560
1.081700208	1.0813665	2.9520978	9.1694287
1.078326542	1.139618166	3.1708451	8.9260686
Ave.=1.1000936418	Ave.=1.1320964834	Ave.=3.0148321	Ave.=8.27449202

Performance Comparison

Performance Ratio :

1. Integer Matrix Multiplication:

$$\frac{Time_{System2}}{Time_{System1}} = \frac{\text{Ave.=3.0148321}}{\text{Ave.=1.1000936418}} = 2.740523156798778$$

2. Double Matrix Multiplication

$\frac{Time_{System2}}{Time_{System1}} = \frac{Ave.=8.27449202}{Ave.=1.1320964834} = 7.308998960185269$

Clock Speed VS Performance

$$\text{Clock Speed} = \frac{\text{Clock Speed}_{System1}}{\text{Clock Speed}_{System2}} = 3.49/2.3 = 1.5173913043$$

Second Matrix Multiplication

System-1 Performance		System-2 Performance	
Integer Multiplication	Double Multiplication	Integer Multiplication	Double Multiplication
2.804447833	3.130360458	8.5754843	10.6168788
2.808079208	3.186523875	8.4156613	10.8374986
2.816238667	3.106714167	8.7598768	10.5191149
3.05089775	3.067507167	10.2719648	10.8991715
2.813399209	3.111457833	8.5051859	10.6233162
Ave.=2.8586125334	Ave.=3.1205127	Ave.=8.90563462	Ave.=10.699196

Performance Comparison

Performance Ratio :
1. Integer Matrix Multiplication:

$\frac{Time_{System2}}{Time_{System1}} = \frac{8.90563462}{3.1205127} = 3.115369612336983$

2. Double Matrix Multiplication

$$\frac{Time_{System2}}{Time_{System1}} = \frac{10.699196}{3.1205127} = 3.428666065034762$$

Clock Speed VS Performance

$$\text{Clock Speed Ratio} = \frac{\text{Clock Speed}_{System1}}{\text{Clock Speed}_{System2}} = 3.49/2.3 = 1.5173913043$$

This means the clock speed of System-1 is 51.73% faster than the clock speed of System-2. Generally, a faster clock speed leads to improved performance because the central processing unit (CPU) can carry out instructions more rapidly. Nonetheless, it is not the sole determinant of overall performance.

Based on the retail price of the two systems, which one is more cost effective (5)?

System Configuration table

System-1 Configuration:	**System-2 Configuration:**
Manufacturer: Apple (MacBook Air M2) CPU: 8 Core Memory: 8 GB OS: MacOS Sonoma 14.2.1 Compiler Info: JDK Version - 18 Clock Speed: 3.49 GHz Price: $999	Manufacturer: Acer (Nitro 5) CPU: 8 Core Memory: 8 GB OS: Windows 10(X64) Compiler Info: JDK Version - 18 Clock Speed: 2.30 GHz Price: $1100

Based on the details outlined in the specification table, it is evident that System-1, priced at $999, offers a better value in terms of performance compared to System-2, which is priced at $1100. System-1 demonstrates superior power and speed over System-2. Consequently, it can be concluded that the Apple MacBook Air M2 (System-1) is a more cost-effective choice.