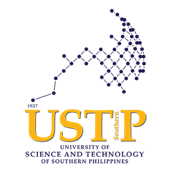
## UNIVERSITY OF SCIENCE AND TECHNOLOGY OF SOUTHERN PHILIPPINES

Claro M. Recto Avenue, Lapasan, Cagayan de Oro City

## COLLEGE OF ENGINEERING AND ARCHITECTURE

*Department of Computer Engineering*

Applications of Functions

Presented to the Faculty of Computer Engineering as Partial Fulfillment for Differential Calculus

## Kyle Justin Arshad Butanas

## Lenard O. Saren

## Students

**Engr. Darwin Jone H. Jupiter**

Instructor

# INTRODUCTION

# As a first-year computer engineering student, I am excited to explore the fundamental relationships that govern software development and data transmission in computer systems. This project focuses on two important areas: how CPU clock speed affects software compilation time and how network bandwidth influences data transmission efficiency in cloud storage systems.

# In our first investigation, we examine how the speed of a CPU, measured in gigahertz (GHz), impacts the time it takes to compile large software projects. Through our experiments, we learned that as we increase the clock speed, the compilation time decreases. However, this relationship isn’t straightforward; it follows a pattern of diminishing returns at higher speeds. By modeling this relationship, we can better understand how to optimize compilation processes and make our programming tasks more efficient.

# In the second part of our project, we look into the data transmission efficiency of cloud systems, where the amount of bandwidth (in Mbps) significantly affects how quickly data can be uploaded or downloaded. We discovered that while increasing bandwidth can enhance transmission speeds, the efficiency improvements become smaller as bandwidth continues to rise. Using a logarithmic function, we will calculate how different bandwidth levels affect transmission efficiency, helping us find the additional bandwidth needed to achieve optimal performance.

# Through this project, I aim to gain valuable insights into these relationships and their practical applications in real-world scenarios. By using mathematical models to analyze our findings, we hope to identify strategies for improving software compilation and data transmission, laying a strong foundation for my future studies in computer engineering.

# PROBLEM

**Realistic CPU Clock Speed vs. Compilation Time Model**

A group of computer engineering students is working on a project that involves compiling a large piece of software. The time it takes to compile the software depends on the clock speed of their CPU. The relationship between the CPU clock speed in GHz and the compilation time in minutes is given by the function. The goal is to avoid zero and look for a balance and accurate with the inverse relationship.

**Processing Power and Execution Time**

After some trials, the we discover that as we increase the processing power of our CPU, the execution time decreases, but not linearly. So we determine that the relationship can be modeled using a quadratic function due to diminishing returns at higher clock speeds. We should find the execution time if the CPU runs at 3 GHz and so on.

**Network Bandwidth vs. Data Transmission Efficiency in a Cloud System**

A computer engineering student working on optimizing data transmission in a cloud storage system. As users upload or download files from the cloud, the network’s bandwidth usage grows. However, as the bandwidth increases, the efficiency of data transmission or how effectively data is sent with minimal delay or loss which improves at a diminishing rate and increases in bandwidth bring large improvements in transmission speed, but further increases yield smaller gains. This relationship can be modeled with a logarithmic function. They should calculate the data transmission efficiency when the bandwidth is 10 Mbps, 50 Mbps, and 100 Mbps and determine how much additional bandwidth is needed to achieve 90% efficiency if the current bandwidth is 50 Mbps and the constant.

# SOLUTION

1. *f*(x) = k/x + c

*Where:*

*f(x) is the compilation time in minutes.*

*x* i*s* the clock speed in GHz.

*k* is a constant that adjusts the sensitivity of the time to the clock speed.

c is the minimum compilation time, representing a limit that the time approaches but never

reaches zero.

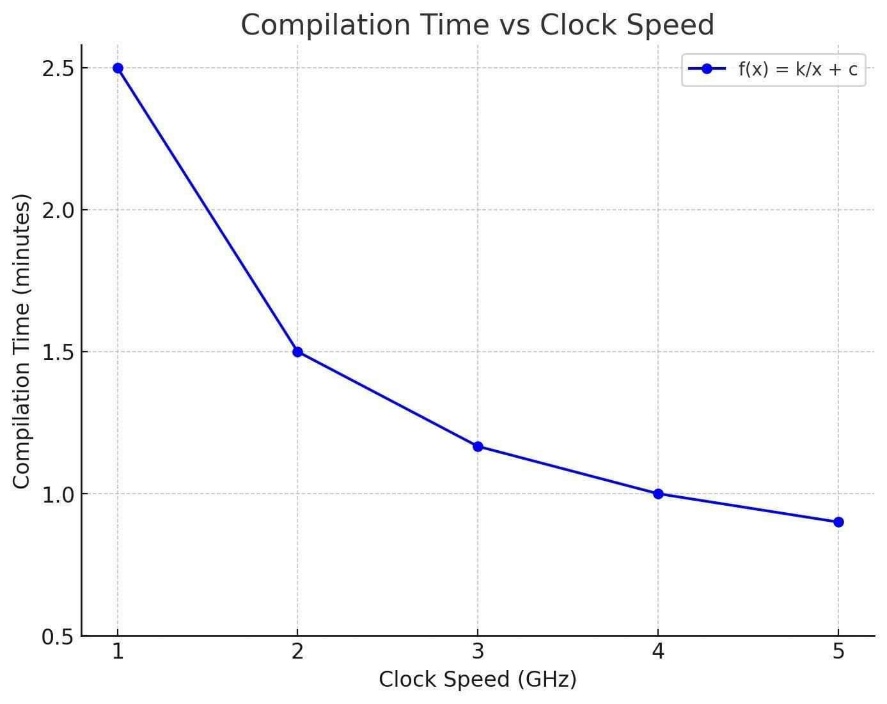
🡺*f*(1) = 2/1 + 0.5 = 2.5 minutes at 1 GHz

🡺*f*(2) = 2/2 + 0.5 = 1.5 minutes at 2 GHz

🡺*f*(3) = 2/3 + 0.5 = 1.17 minutes at 3 GHz

🡺*f*(4) = 2/4 + 0.5 = 1 minute at 4 GHz

🡺*f*(5) = 2/5 + 0.5 = 0.9 minute at 5 GHz



1. *g*(x) = -0.2x² + 4c + 10

Where:

*g(c)* is the execution time in seconds.

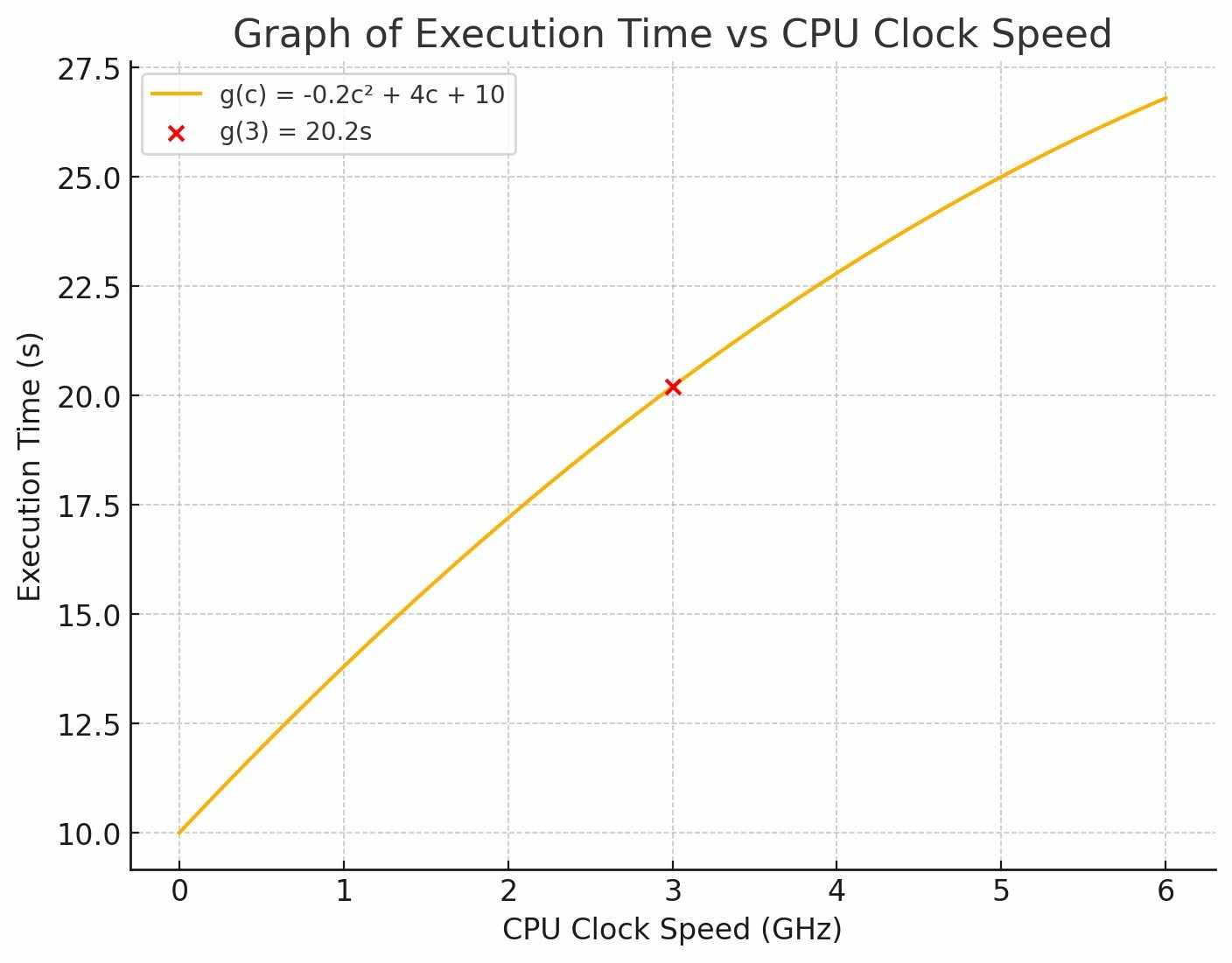
c is the clock speed of the CPU in GHz.

🡺G(3) = -0.2(3)² + 4(3) + 10

🡺 -0.2(9) + 12 + 10 ]

🡺 -1.8 + 12 + 10

🡺 20.2s

****

1. 𝐸(𝐵)=𝐴Log(𝐵+1)

Where:

𝐸(𝐵) is the data transmission efficiency (measured in %),

𝐵 is the bandwidth in Mbps

𝐴 is a constant that depends on the network’s initial conditions

**For B = 10 Mbps**:

🡺E(10) = 20*log*(10+1) 🡺 20*log*(11)

🡺 *log*(11) ≈ 1.0414

🡺E(10) = 20×1.0414 🡺 20.828%

**For B = 50 Mbps**:

🡺E(50) 🡺 20*log*(50+1)

🡺 *log*(51) ≈ 1.7076

🡺E(50) = 20×1.7076 = 34.152%

**For B = 100 Mbps:**

🡺E(100) = 20*log*(100+1) 🡺20*log*(101)

🡺*log*(101) ≈ 2.0043

🡺E(100)=20×2.0043b= 40.086%

*E*(B) = A*log*(B + 1),

where E(B) = 90 A = 20

🡺90 = 20*log*(B+1)

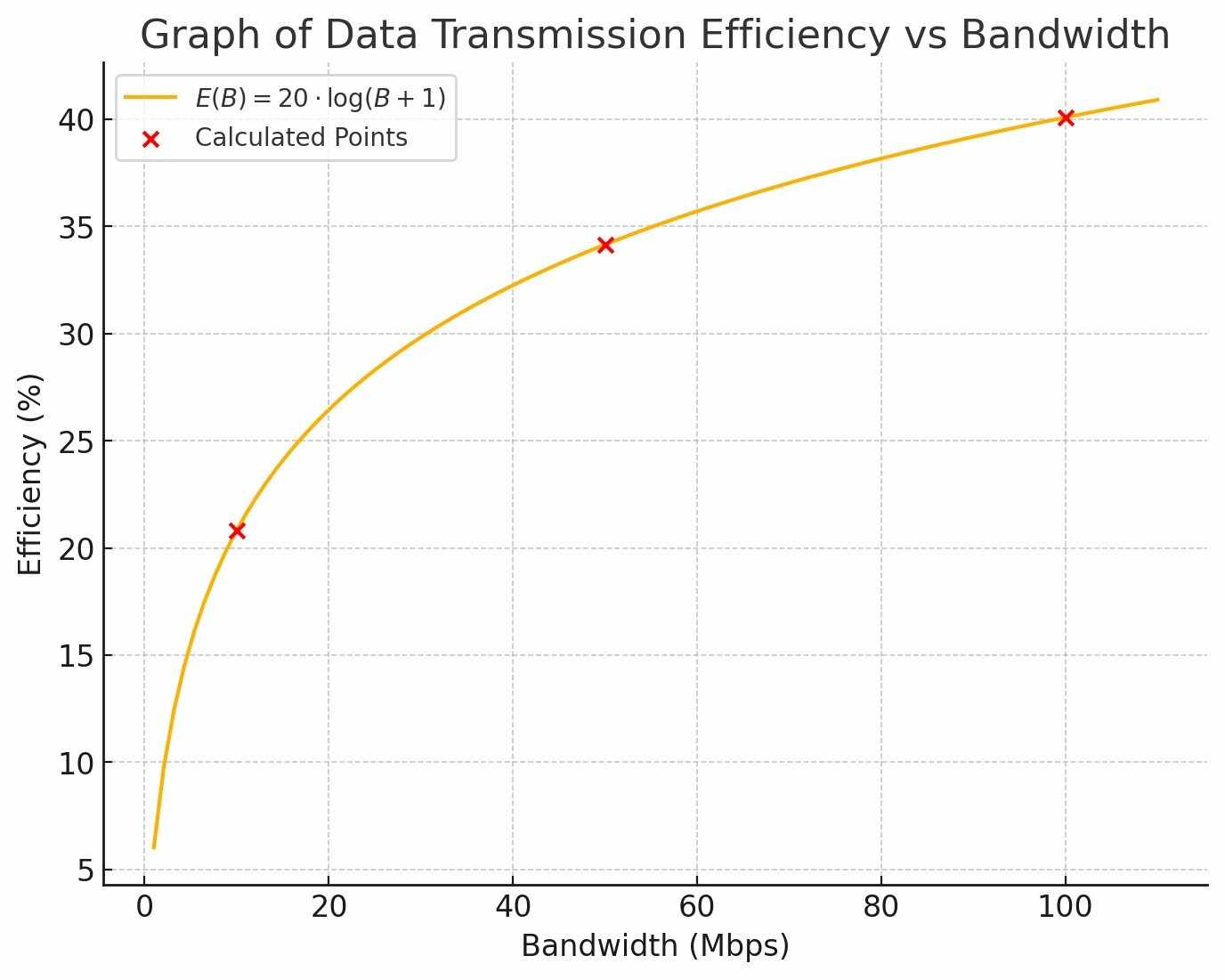
🡺90/20 = 20/20 *log* (B+1)

🡺4.5 = *log* (B+1)

B+1 = 104.5 = B + 1 = 31622.8

🡺B + 1 – 1 = 31622.8 – 1

🡺31621.8 Mbps

****

# FIGURES

🡺 = equals or the value after the solving

≈ = approximately

*log* = logarithm

a – z = variables

# CONCLUSION

As first-year computer engineering students, our exploration into the relationship between CPU clock speed and compilation time has revealed several important insights:

1. CPU Speed and Compilation Time: Initially, we expected that faster clock speeds would directly reduce compilation time, but we discovered that the relationship isn’t linear. Instead, the decrease in time slows down as clock speed increases, which led us to model this relationship using a quadratic and inverse function. These models better reflect real-world performance, where increasing speed yields diminishing returns.
2. Execution Time Model: Using the quadratic model , we calculated the execution time for a 3 GHz CPU to be 20.2 seconds. This further demonstrated that while higher CPU speeds reduce execution time, there’s always a limit to how much improvement we can achieve.
3. Network Bandwidth Efficiency: Similarly, in our exploration of network bandwidth vs. data transmission efficiency, we found that increasing bandwidth improves data transmission but at a diminishing rate. For example, going from 10 Mbps to 100 Mbps only improves efficiency from 20.8% to 40.1%. To achieve 90% efficiency, we calculated that a massive increase in bandwidth, to around 31,622.8 Mbps, would be needed, which is impractical.

Whether improving CPU clock speed or network bandwidth, we consistently observed returns. This means that after a certain point, adding more resources yields smaller improvements, which is a key concept in system optimization. The goal in both cases is to find a balance where increasing resources like CPU power or bandwidth provides meaningful benefits without excessive waste. Both the inverse and quadratic models help create more accurate representations of system performance. This understanding will be valuable as we continue to optimize hardware and network systems in future projects. Through these calculations, we learned the importance of realistic modeling when it comes to system performance, a crucial skill in our journey as future computer engineers..

# REFERENCES

R. Jain, “The Art of Computer Systems Performance Analysis,” Wiley, 1991.

John G. Proakis and Dimitris G. Manolakis, “Digital Signal Processing: Principles, Algorithms, and Applications,” 4th ed. Pearson, 2006.

William Stallings, “Computer Networking with Internet Protocols and Technology,” 7th ed. Pearson, 2016.

Additional Reminders:

* Enumerate your names alphabetically.
* Do not change the formatting of the cover page.
* Follow the following format in editing this paper:
  + Paper Format:
    - A4 Paper Size
    - 1” TB, 0.75” LR Margin
    - Font Style: **DOMITIAN** (Do not use any other font style.)
    - Font Size: 12pt
* You may use subsections if you wish. Use the following font sizes for formatting:

# Section (15pt, Bold)

* + - ***Subsection (13, Bold-Italic)***
      * *Sub-subsection (12, Italic)*
* Submit this paper in a clear sliding folder (white).
* This paper will be graded according to the rubrics present in the syllabus.