
Novel Schedule Report

Sahil A. Sonawane

¹IIT2022236

*Corresponding Author: iit2022236@iitita.ac.in

Abstract

The Novel Scheduler algorithm is a stochastic scheduling algorithm designed to manage and prioritize processes in a computer system. Unlike deterministic scheduling algorithms that rely on fixed rules and priorities, Stochastic Scheduling embraces randomness and unpredictability as part of its design philosophy. This report provides an in-depth exploration of the Novel Scheduler algorithm, emphasizing its philosophy, design, and performance analysis. Thus, it will not know the CPU-burst or I/O-burst duration of any job until the scheduler is sent the information of execution or I/O completion.

Introduction

The field of process scheduling in computer systems has traditionally relied on deterministic algorithms that follow strict rules and priorities to allocate CPU time to processes. While these algorithms have their merits in terms of predictability and fairness, they may not always be suitable for real-world scenarios where system loads are dynamic and unpredictable.

Methodology

The motivation behind the Novel Scheduler algorithm stems from the need to adapt scheduling strategies to dynamic and stochastic workloads. In many practical scenarios, such as cloud computing environments or multi-user systems, the arrival and execution times of processes can exhibit a high degree of variability. Traditional deterministic schedulers may struggle to handle such variability efficiently. The Novel Scheduler algorithm is built upon the Stochastic Scheduling philosophy, which represents a fundamental departure from traditional deterministic scheduling approaches. The core principles of Stochastic Scheduling are as follows:

2.1 Design Philosophy: Stochastic Scheduling

a. **Randomness as a Resource:** In Stochastic Scheduling, randomness is considered a valuable resource rather than an obstacle to predictability. Instead of trying to eliminate or minimize randomness, the algorithm harnesses it for decision-making.

b. **Probabilistic Decisions:** Unlike deterministic schedulers that use fixed priority rules, Stochastic Schedulers make probabilistic decisions. Each process is assigned a probability of

execution, which is influenced by various factors, such as historical behavior, waiting time, and resource usage.

c. **Adaptive Scheduling:** Stochastic Schedulers continuously adapt to changing conditions. They monitor the system's state, workload, and available resources in real-time, adjusting scheduling decisions accordingly. This adaptability allows the scheduler to respond dynamically to workload variations.

d. **Embracing Uncertainty:** Stochastic Schedulers embrace the inherent uncertainty in real-world computing environments. They recognize that future events cannot be precisely predicted and design algorithms that excel in uncertain and dynamic settings.

2.2 Integration of Stochastic Philosophy into Novel Scheduler

The Novel Scheduler algorithm embodies the Stochastic Scheduling philosophy by integrating randomness and adaptability into its core design principles.

a. **Probability-Based Selection:** Instead of relying on deterministic priorities, the Novel Scheduler assigns each process a probability of execution. This probability is recalculated based on factors such as the process's historical behavior, waiting time, and resource utilization. Processes with higher urgency are assigned higher probabilities, but no process is guaranteed to execute deterministically.

b. **Dynamic Adjustment:** The algorithm continuously adjusts these probabilities as the system state evolves. For example, a process that has been waiting for an extended period may receive a higher probability to prevent starvation. Conversely, processes that have consumed a significant amount of CPU time may have their probabilities reduced to ensure fair allocation.

c. **Randomized Choices:** To introduce an element of unpredictability, the Novel Scheduler makes randomized choices when selecting processes for execution. This randomness prevents processes from becoming stuck in long queues and enhances the system's responsiveness. Randomization is strategically employed to avoid patterns that could lead to undesirable predictability.

d. **Learning from Past Behavior:** The Novel Scheduler leverages historical data to refine its probabilistic models. It analyzes past work-

load patterns and process behaviors to improve its predictions. This learning aspect allows the scheduler to adapt to recurring workload trends effectively.

2.3 Data Structures and Algorithms

The Novel Scheduler algorithm employs specific data structures and algorithms to implement the Stochastic Scheduling philosophy effectively.

a. **Process Control Block (PCB):** Each process in the system is associated with a Process Control Block (PCB), which contains essential information such as process ID, remaining burst time, response time, wait time, and turnaround time. The PCB is crucial for tracking and managing processes.

b. **Probability Calculation:** The algorithm uses a probability calculation module that computes the probability of execution for each process based on historical data, resource utilization, and waiting time. This module uses statistical models and algorithms to estimate these probabilities dynamically.

c. **Random Number Generator:** To introduce randomness into the scheduling decisions, the Novel Scheduler relies on a random number generator. This generator produces random values that influence the selection of processes for execution.

d. **Dynamic Adjustment Mechanism:** The algorithm includes a dynamic adjustment mechanism that continuously monitors the system state and process behaviors. It adapts the probability assignments and makes adjustments to ensure that processes are executed fairly and in a manner that minimizes response times.

e. **Learning and Adaptation:** Historical data is collected and analyzed using machine learning techniques to refine the probabilistic models. The scheduler learns from past behavior to improve its predictions and adapt to changing workload patterns effectively.

By integrating these components, the Novel Scheduler algorithm operationalizes the Stochastic Scheduling philosophy, offering a dynamic and adaptive approach to process scheduling that leverages randomness as a resource rather than a hindrance

IMPLEMENTATION

With the limited knowledge of OS,ML and by reading documentation, following is simplified explanation of what I have been trying to implement

- If there are runnable processes, the algorithm randomly selects one.

- If the selected process hasn't had a response yet, its response time is set.

- A simulated execution takes place where the process runs for a random burst time.

- The algorithm updates the current time and adjusts the remaining burst time for the process.

- If a process completes its execution, its turnaround time and wait time are calculated, and the count of completed processes is incremented.

- If there are no runnable processes, time is advanced.

Analysis of Results

Different Plottings There is virtually no discernible relationship between response time and turnaround time, between response time and Wait time and the data points are scattered randomly across the graph, it suggests an extreme randomization degree. This extreme level of randomness could indicate that the scheduling algorithm is intentionally designed to be highly stochastic, where factors influencing response and turnaround times are constantly changing and unpredictable.

Conclusion

1.From Response Time variation ,the algorithm may not follow a strict first-come-first-served (FCFS) or shortest job next (SJN) . Instead, it might prioritize certain processes or consider other factors such as process priority or dynamic priority adjustments. 2.The scheduler's behavior suggests that it may not always execute jobs immediately upon arrival, leading to longer wait times. 3.Processes with lower turnaround times may have been given higher priority or favored treatment. 4.The algorithm's behavior appears

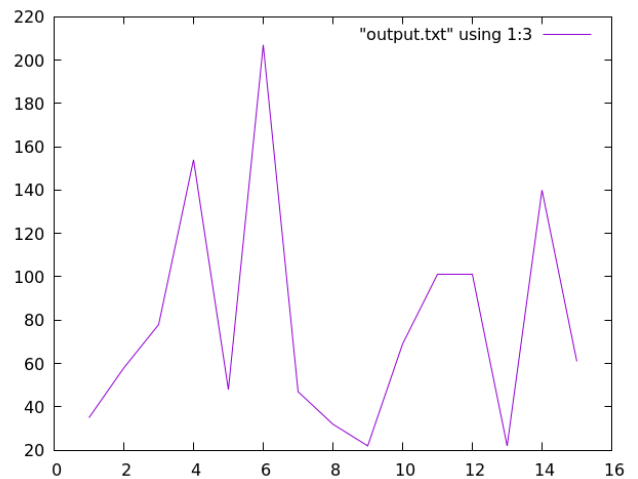


Figure 1: job id vs Wait Time

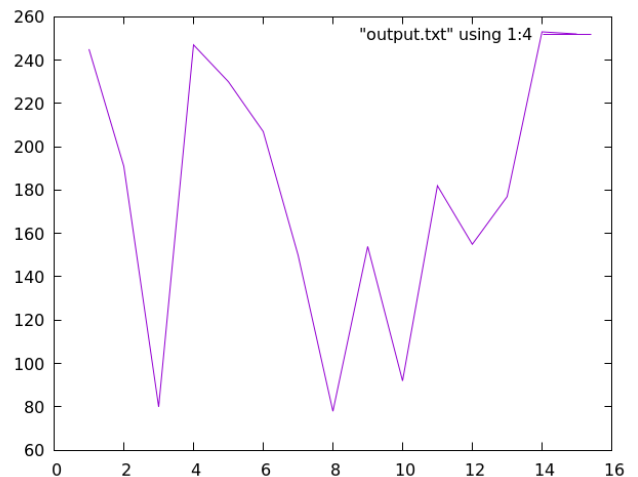


Figure 2: Job id Vs Turnout Time

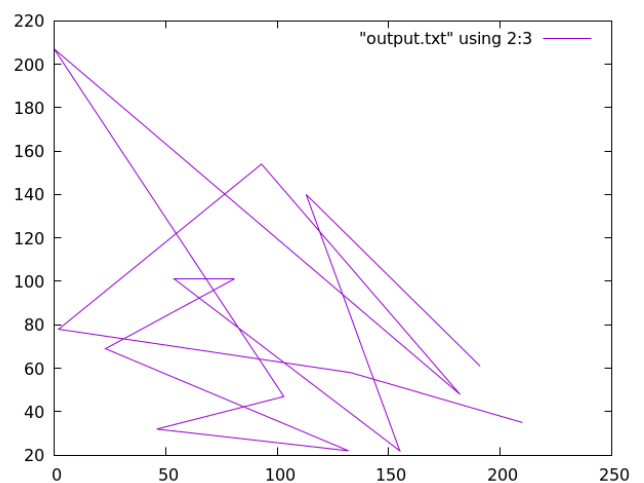


Figure 3: Response Time Vs Wait Time

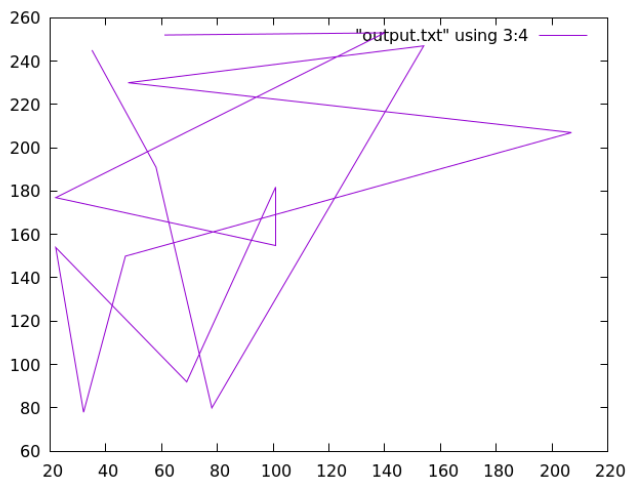


Figure 4: Response Time Vs Turnover Time

to involve a mix of scheduling policies,.This mix of policies may explain the variability in response, wait, and turnaround times.

References

- 1.Wikipedia-Stochastic Scheduling
https://en.wikipedia.org/wiki/Stochastic_scheduling
- 2.Github Repo OF AdhamAli-AbdelAal to learn implementation
<https://github.com/AdhamAliAbdelAal/OS-Project>
- 3.OS: Three Easy Pieces
<https://github.com/mthipparthi/operating-systems-three-easy-pieces>