Scheduling disciplines in the G/G/1 queue with hard deadlines - Homework Assignment (DM), Évaluation de Performances, INFO4 Polytech Grenoble

ANSELMI Jonatha

2025-02-14

Votre devoir sera rédigé en français ou en anglais sous forme d'un document HTML généré à l'aide de R/Markdown et publié sur rpubs en prenant soin de bien laisser le code apparent et de fixer la graine de votre générateur à l'aide de la fonction set.seed au tout début du document afin qu'il soit possible de reproduire vos données avec exactitude.

Vous enverrez l'url rpubs de votre devoir par mail à jonatha.anselmi AT inria.fr avant le 2025-03-31 à 12h00 en indiquant dans le sujet [INFO4-EP] DM.

Objective

The goal of this homework assignment is to simulate and analyze the dynamics of a G/G/1 queue under various scheduling disciplines, where jobs have *hard* deadlines. Specifically, each arriving job must be completed before its individual deadline. The deadline is considered "hard" because if it is not met, the job becomes useless and is consequently removed from the system.

The student is encouraged to build upon the code developed in class for simulating the dynamics of the G/G/1 queue with soft deadlines, as it will provide a convenient foundation for this assignment. This code is available at https://rpubs.com/janselmi/GG1_soft_deadline

Roadmap

Step 0: Warm up

Discuss the advantages and disadvantages of the scheduling disciplines mentioned above in terms of their applicability.
Consider factors such as the amount of information required and potential challenges in implementing them in a real system.

Step 1: Scheduling disciplines implementation

Extend the R code of the G/G/1 queue linked above to simulate the G/G/1 queue of interest under the following scheduling disciplines:

1. **FCFS**: jobs are processed following the order of their arrival. If a job's deadline expires while it is being processed, the job is removed from the system. The server does not know the service times in advance.
2. **EDF** (Earliest Deadline First): jobs are processed according to the earliest deadline with preemption, i.e., the current job is interrupted if a new job with an earlier deadline arrives. The server does not know the service times in advance.
3. **EDF-NP** (Earliest Deadline First Non-preemptive): As EDF but jobs are not interrupted once their processing has started. The server does not know the service times in advance.

4. **SRPT** (Shortest Remaining Processing Time): the job with the shortest remaining processing time is served first. Here, the jobs whose remaining processing times overflow their corresponding deadlines are not considered, i.e., removed from the system.
5. **SRT2D** (Shortest Remaining Time to Deadline): the job with the shortest remaining time to deadline is served first. Note that "the remaining time to deadline" is always non-negative because jobs that cannot be completed by the deadline are removed from the system.
6. **SERT2D** (Shortest Expected Remaining Time to Deadline): as SRT2D but instead of the remaining time to deadline, we consider the *expected remaining time to deadline*. The server does not know the service times in advance (but it knows their distribution function).

Step 2: Performance measures implementation

We are interested in comparing the performance of the previous scheduling disciplines with respect to:

1. **Response Time** (RT): the mean time that completed job spends in the system (in the queue and processing phases). Note that the average is only taken over the jobs that have been actually completed.
2. **Missed Deadline Ratio** (MDR): the proportion of dropped jobs, i.e., jobs that did not complete.

Implement these performance measures in your code.

Step 3: Code correcteness

Take a scheduling discipline of your choice. Simulate the dynamics induced by 10 arrivals and provide numerical evidence that your code is correct (using the print or similar commands).

Step 4: Performance evaluation

Run your code to create a table of the form:

Discipline	RT	MDR

Rely on this table to comment the results you get. In particular, compare the performance induced by the above scheduling disciplines and provide a suggestion on the scheduling discipline that performs the best.

Assumptions

- The arrival process is Poisson with rate λ.
- Service times are all independent and have the distribution of the random variable S. The random variable S is uniform over [0,10] with probability 0.85 and and uniform over [90,110] with probability 0.15.
- The deadline of any job is $x = 5$ times larger than its service time.
- In the G/G/1 queue, it is assumed that the server processes jobs with speed one.
- The system load ρ is 0.9 - recall that $\rho = \lambda/\mathbb{E}[S]$.

Notes

Beyond the correctness of the points above, the final note will depend on how the student will make their homework clear. In particular,

- Justify any further assumption made.
- Add comments in the code to explain what you are doing.
- Make your simulations as robust as possible (e.g., the number of simulated jobs should be large enough).