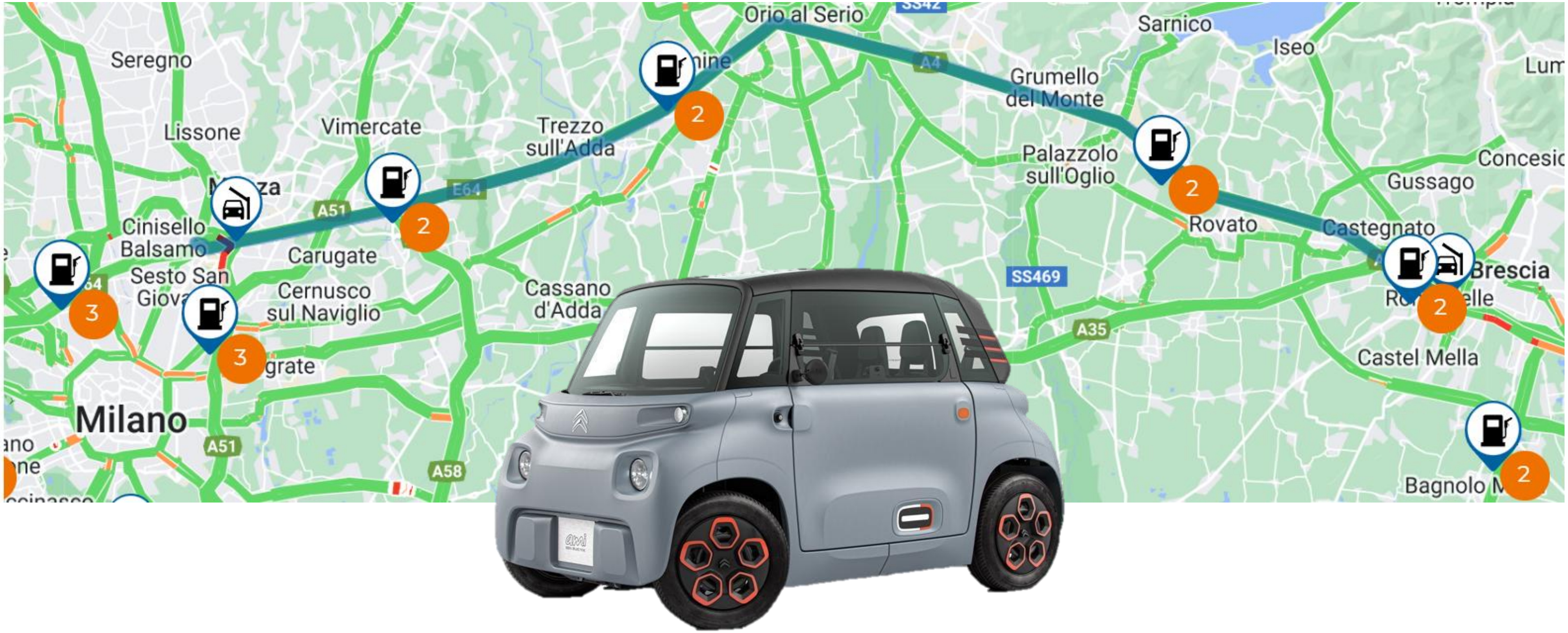
Performance Evaluation and Applications projects

2023 / 2024

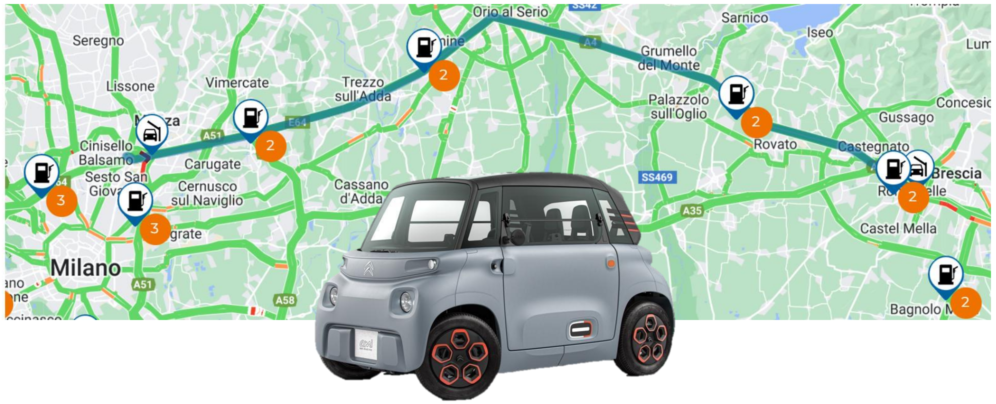
Project Type A

For students with ID (eight digits, “Codice Persona”) ending with 0, 2 or 5

Recharging of an electric car on a highway

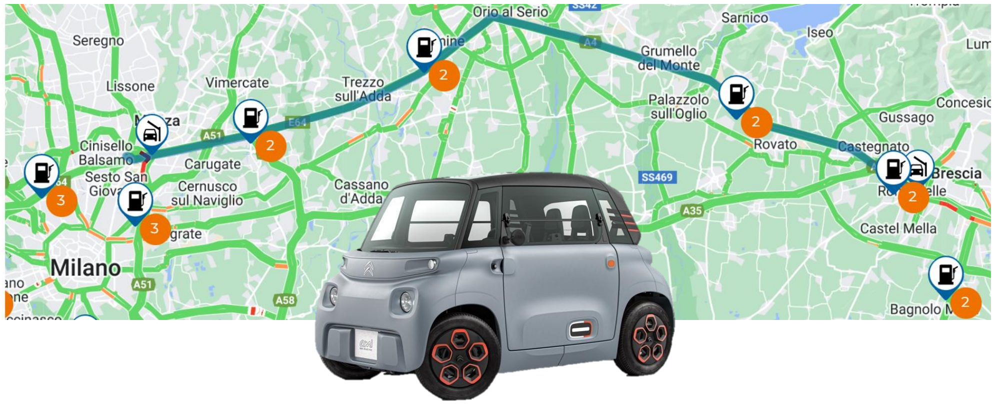


Recharging of an electric car on a highway



- An electric car has to travel a distance which is slightly longer than the one allowed by its battery capacity: it must stop **exactly once** for recharging on the way!
- There are four road segments on the route, with three charging stations in between: each station has a number of chargers available, and a different loads in number of requests.
- Considering a given probability of choosing exactly one of the charging station, compute the average total travelling time.

Recharging of an electric car on a highway

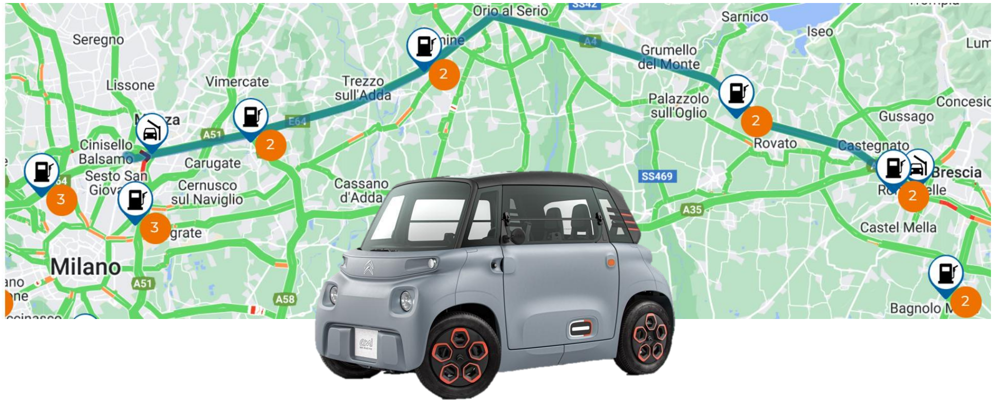


- The travelling times of the four segments, is distributed according to the following traces [all times are expressed in *minutes*]:

Segment	Trace
I	`TraceA-I.txt`
II	`TraceA-II.txt`
III	`TraceA-III.txt`
IV	`TraceA-VI.txt`

Recharging of an electric car on a highway

- Charging time are exponentially distributed, according to an exponential distribution, with an average of 30 minutes.
- The request rate by other cars at the station, and the number of chargers, is given in the following table.
- A station is identified by the number of the segments it is between.

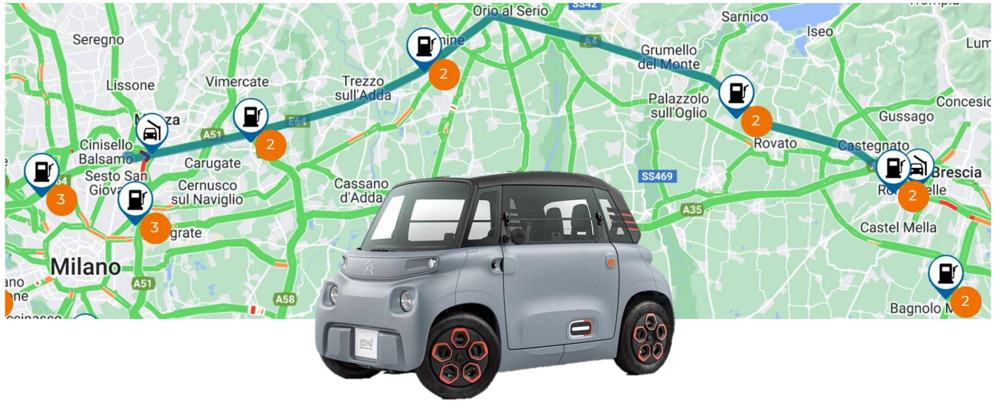


Station	Other traffic [car / hour]	Number of chargers
I-II	6	4
II-III	4	3
III-IV	5	3

The arrival rate of other cars at the stations can be considered a Poisson process.

Recharging of an electric car on a highway

- Determine the *best stopping probability distribution*: test a few alternatives of probabilities of stopping at each station, and for each scenario determine the average travelling time.

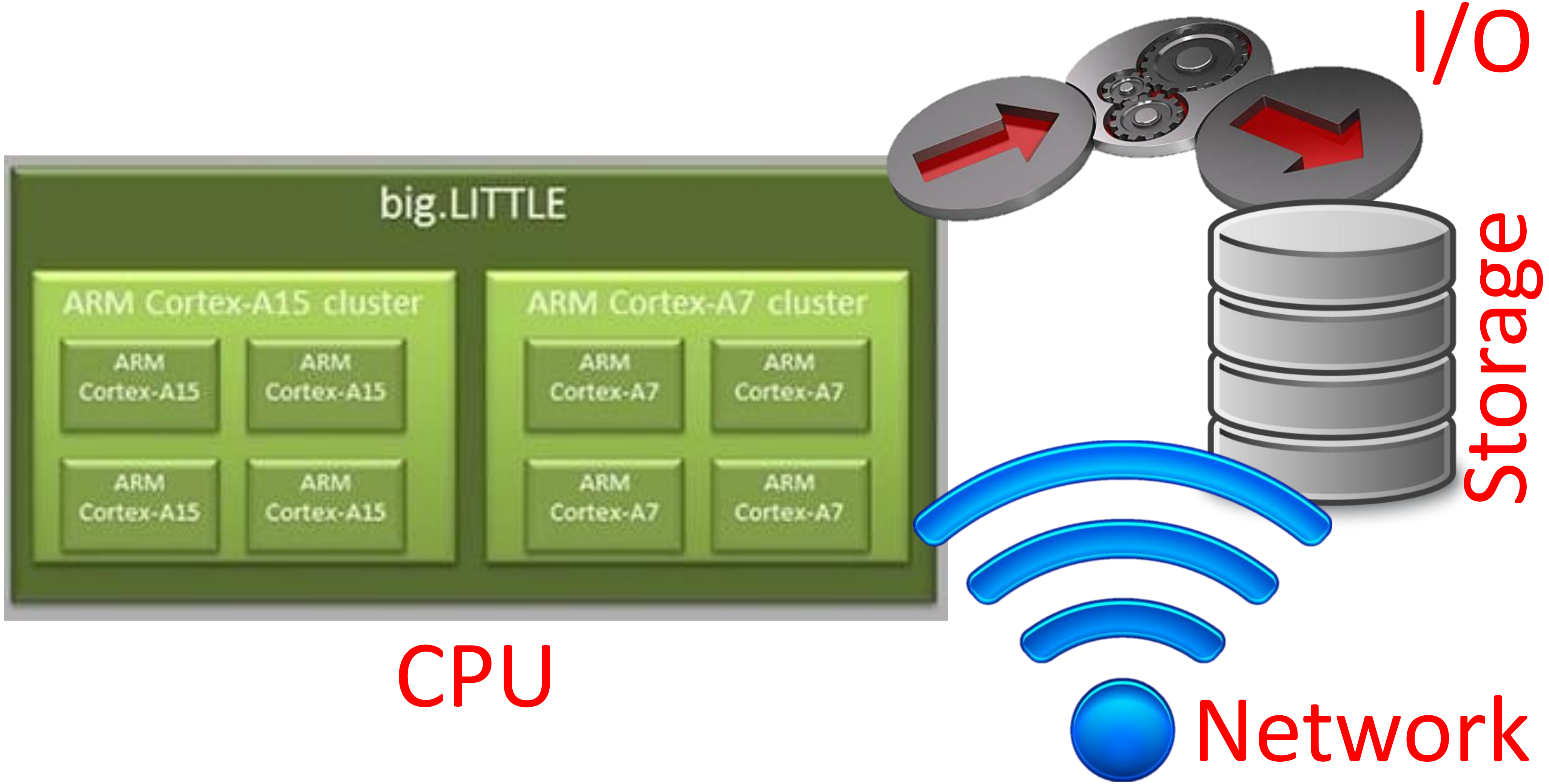


- Hint: the motion of the car can be considered as a closed system, with a single job, where the car once it has completed its course, it is teleported back to the initial position to immediately start another trip.
- Other cars competing for the charger, can be seen as an open process.

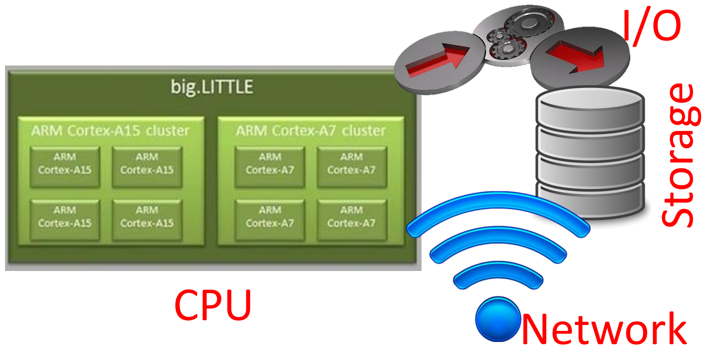
Project Type B

For students with ID (eight digits, “Codice Persona”) ending with 1 or 4

Performance of a Big-Little architecture



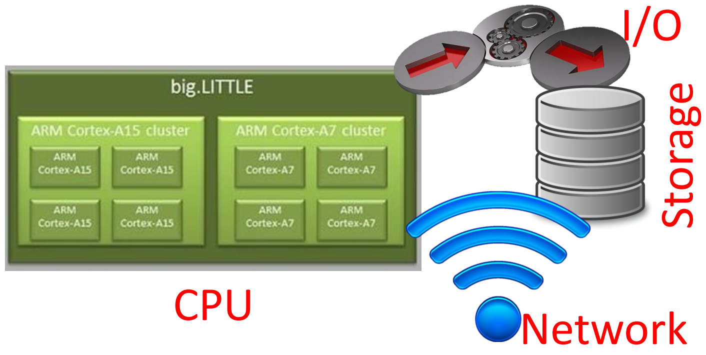
Performance of a Big-Little architecture



- A system characterized by a Big-Little architecture is characterized by *4 high performance* cores, and *8 energy efficient* cores.
- It is used by $N_B = 10$ heavy computation tasks, and $N_L = 32$ low computation tasks.
- The scheduler will mainly schedule heavy computation tasks on the high performance cores, while low computation tasks on the energy efficient cores. However, to better use the resources, there is also a small probability that tasks will be assigned the other way round.

Performance of a Big-Little architecture

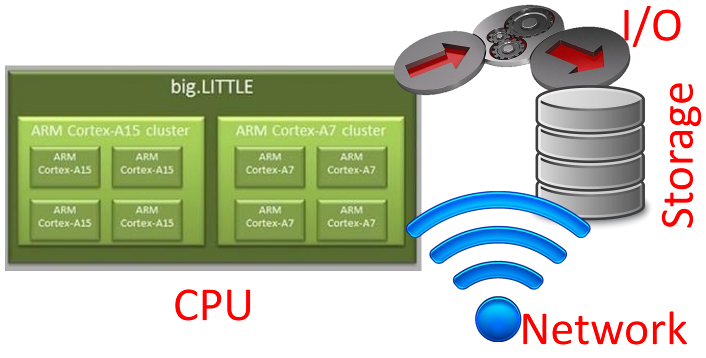
- The execution times of the tasks on the cores are collected in the following traces (all expressed in *sec*):



	High Performance Cores	Energy Efficient Cores
Heavy computation tasks	`TraceB-HH.txt`	`TraceB-HE.txt`
Low computation tasks	`TraceB-LH.txt`	`TraceB-LE.txt`

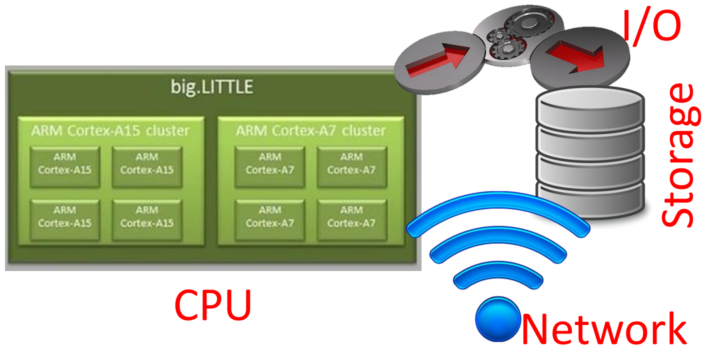
Performance of a Big-Little architecture

- The system is also composed by an *I/O subsystem*, a *Storage* component and *Network* access.
- All these components can be considered working in processor sharing, with an exponential service time (different per type of job), whose average is described in these tables:



	I/O	Storage	Network
Heavy computation tasks	*50 msec*	*200 msec*	*5 msec*
Low computation tasks	*150 msec*	*10 msec*	*120 msec*

Performance of a Big-Little architecture



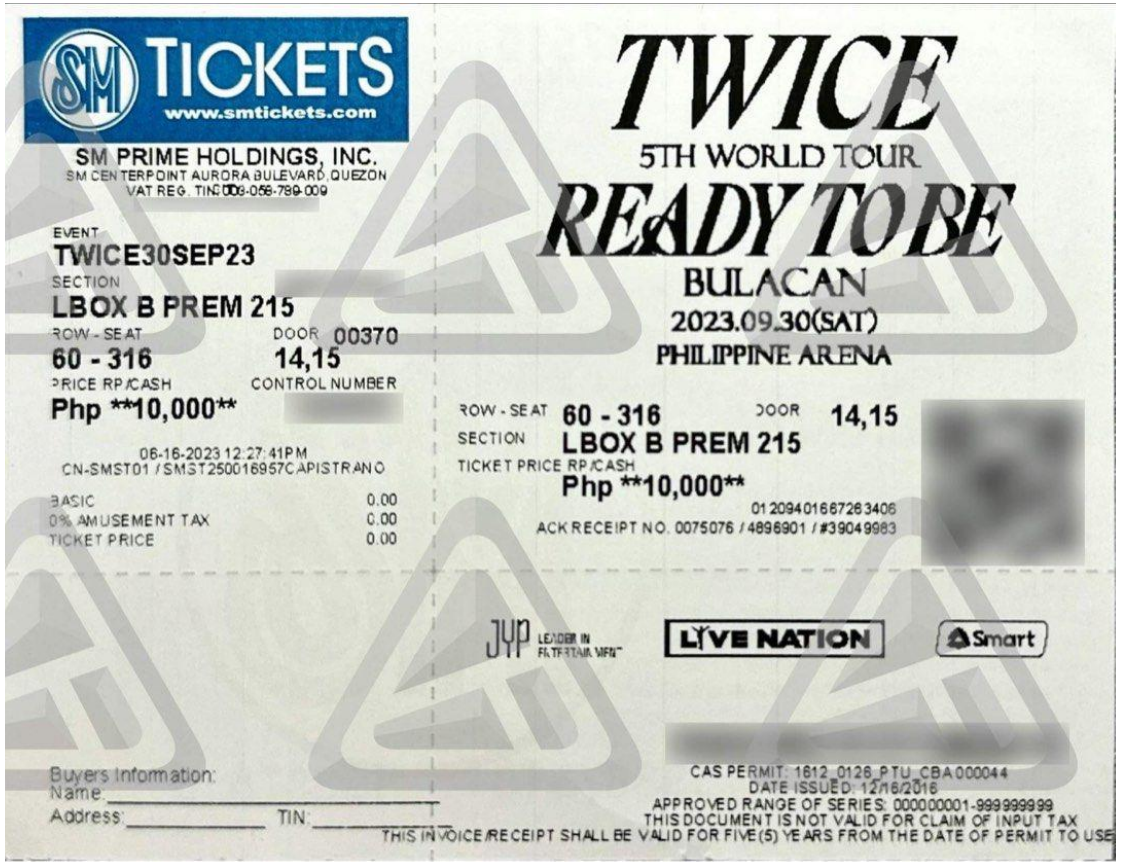
- Determine the *best assignment probability distribution*: test a few alternatives of probabilities of assigning a heavy computation task to an efficiency core and of assigning low computation task to a high performance core. Determine the system throughput in each scenario.

- Hint: the two sets of cores can be considered as two independent processor sharing stations.

Project Type C

For students with ID (eight digits, “Codice Persona”) ending with 3, 6 or 7

Concert ticket service



Concert ticket service

- A concert ticketing service has to handle a large number of requests in a short amount of time.
- The system can allow at most *N = 1000* pending requests at a time.
- The service will be composed by four stages:
    - Welcome message (very quick)
    - Seat selection
    - Payment processor
    - Ticket issuing

Concert ticket service

- The durations of the four stages, is distributed according to the following traces [all times are expressed in *ms*]:

Segment	Trace
Welcome	`TraceC-W.txt`
Seat selection	`TraceC-S.txt`
Payment	`TraceC-P.txt`
Ticket issuing	`TraceC-I.txt`

Concert ticket service

- Requests arrive at a rate of *6000 req / min* in the *first 8 hours*.
- Then they will arrive at a rate of *300 req / min* for the following *6 days and 16 hours*.
- The service will propose a new concert every week.
- Requests not admitted, will be dropped.

Concert ticket service

- Try different configurations in number of cores executing each stage, to obtain an average response time below *5 minutes*. Try also to minimize the average drop probability below 25%

- Hint: the arrival can be considered a Markov Modulated Poisson Process, with the given rates and durations. Utilization can help in determining which stage could require more instances to provide a better service.

Project Type D

For students with ID (eight digits, “Codice Persona”) ending with 8 or 9

Performance of a Video production company

Performance of a Video production company

- The production of a an episode of a series can considered as the execution of six stages:

    1. Story writing
    2. Shooting
    3. Audio editing
    4. Video editing
    5. Visual Effects production
    6. Compositing

- All stages must be done in sequence, except stages 3, 4 and 5 that can be done in parallel: however, all three needs to be completed to proceed to stage 6.

Performance of a Video production company

- The durations of the last four stages, is distributed according to the following traces [all times are expressed in *days*]:

Segment	Trace
Audio editing	`TraceD-A.txt`
Video editing	`TraceD-V.txt`
VFX	`TraceD-X.txt`
Compositing	`TraceD-C.txt`

- *Writing* and *shooting* can be considered *Erlang distributed*, with the following parameters:

	K	λ [$days^{-1}$]
Shooting	9	3
Writing	16	4

Performance of a Video production company

- The target throughput is producing a new video every three weeks, but keep the production of each one lasting less than two months.
- The production can afford two other units to work either on audio, video or VFX. Several episodes can be produced at the same time.
- By testing different configuration in terms of number of episodes being produced, and number of units being used, find the optimal configuration (i.e. best throughput).

- Hint: increasing the number of episodes done in parallel increases the throughput, but increases also the production time. Utilization can help find the bottleneck