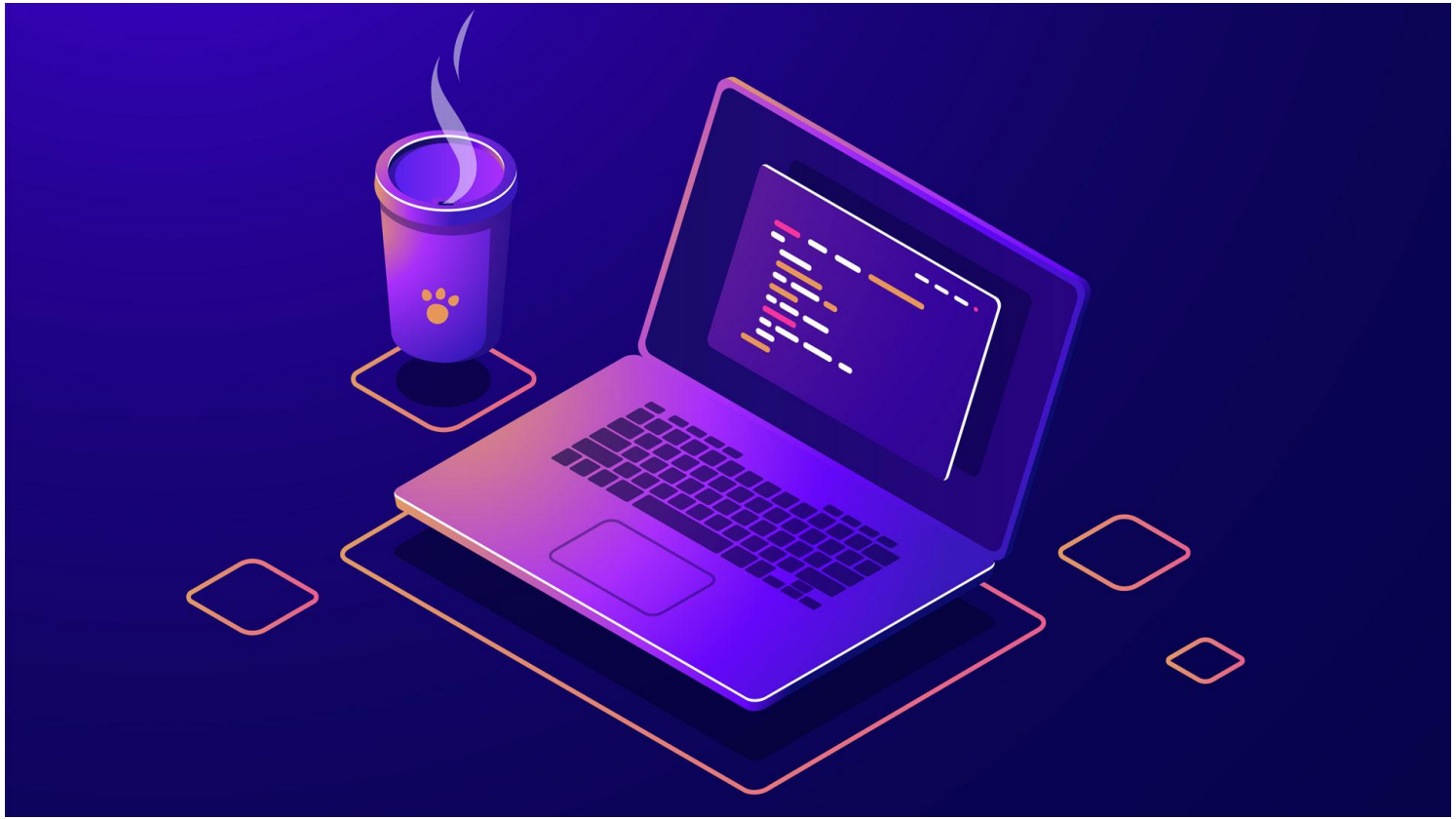
Granularity and Parallelism



Granularity and Parallelism

- The granularity of the task decomposition is determined by the computational size of the nodes (tasks) in the task graph

- Example: counting matches in our car dealer database

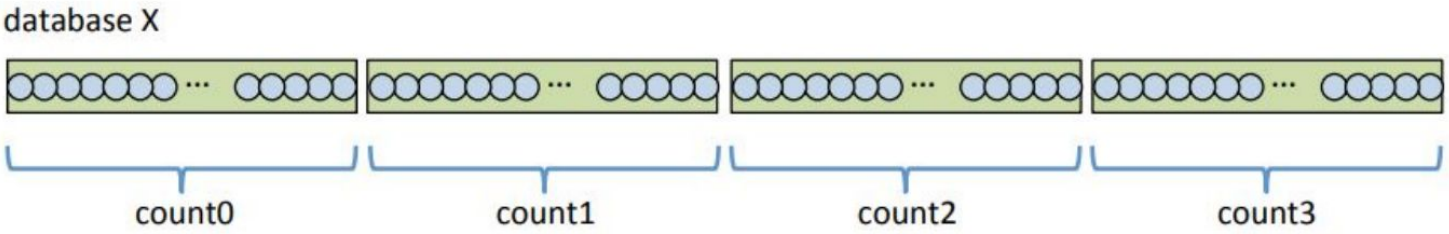
```
count = 0;
for ( i=0 ; i< n ; i++ )
    if (X[i].Color == "Green") count++;
```

Coarse-grain decomposition:
The whole loop is a task
Parallelism = 1

Fine-grain decomposition:
Each iteration of the loop is a task
Parallelism = n

Granularity and Parallelism

- Example: A task could be in charge of checking a number of consecutive elements m of the database:



with a potential *parallelism* $= n \div m$

$$1 < (n \div m) < n$$

Granularity and Parallelism

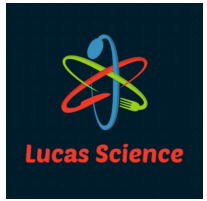
- It would appear that the parallelism is higher when going to fine-grain task decompositions

	Coarse grain	Fine grain	Medium grain
Number of tasks	1	n	n / m
Iterations per task	n	1	m
Parallelism	1	n	n / m

- However, tradeoff between potential parallelism and overheads related with its exploitation (e.g. creation of tasks, synchronization, exchange of data, ...)

Instructor Social Media

Youtube: Lucas Science



Instagram: lucaasbazilio

Twitter: lucasebazilio