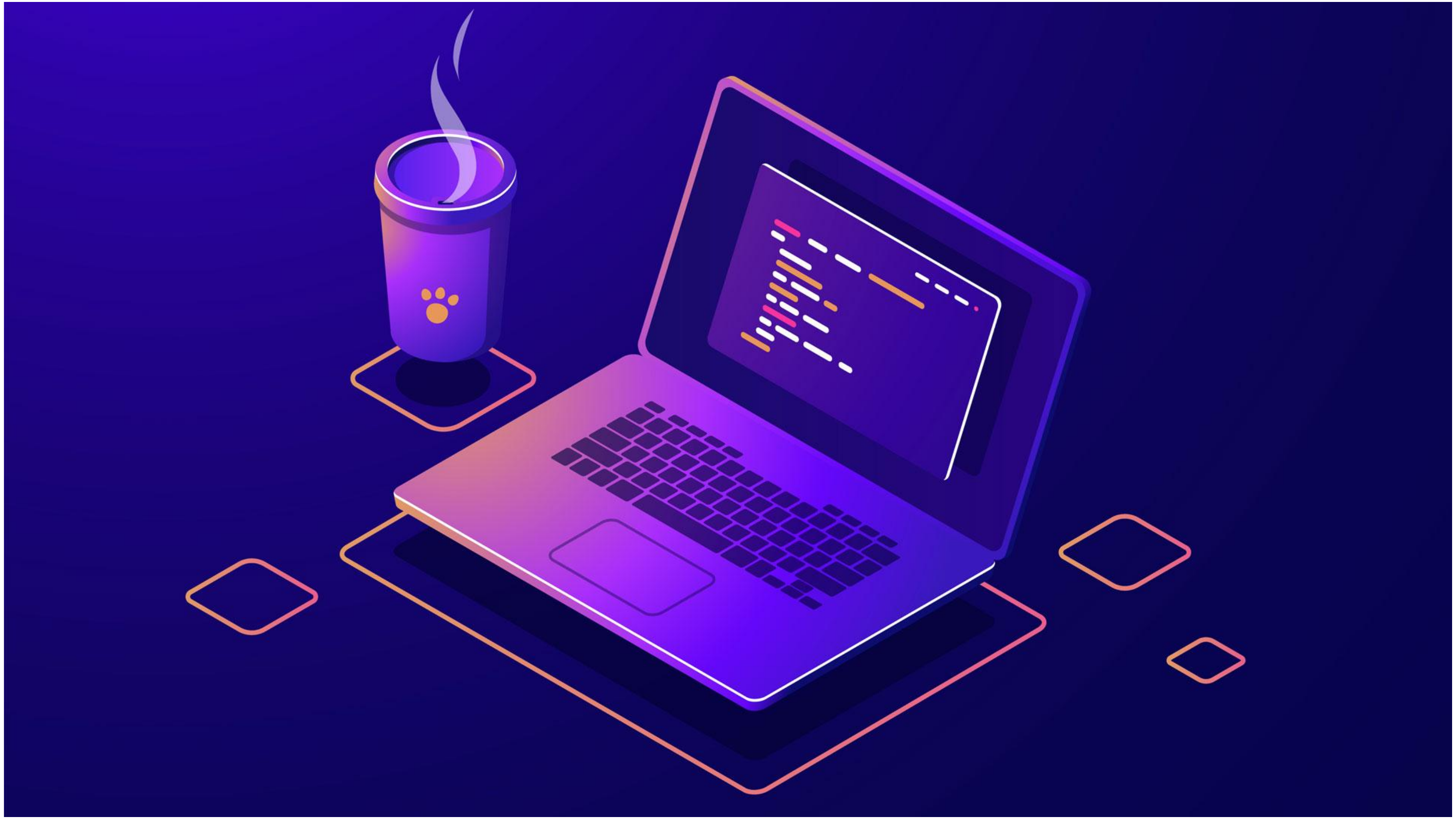
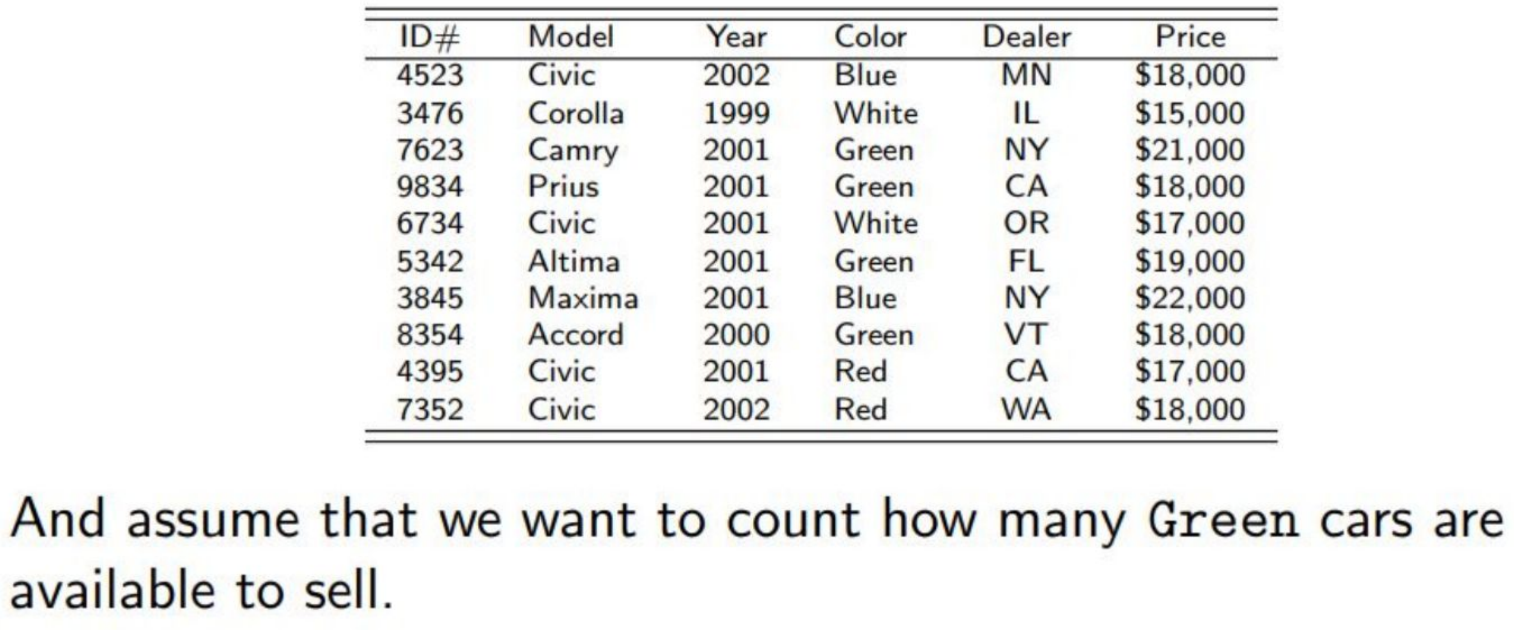
Expressing Tasks



Motivation

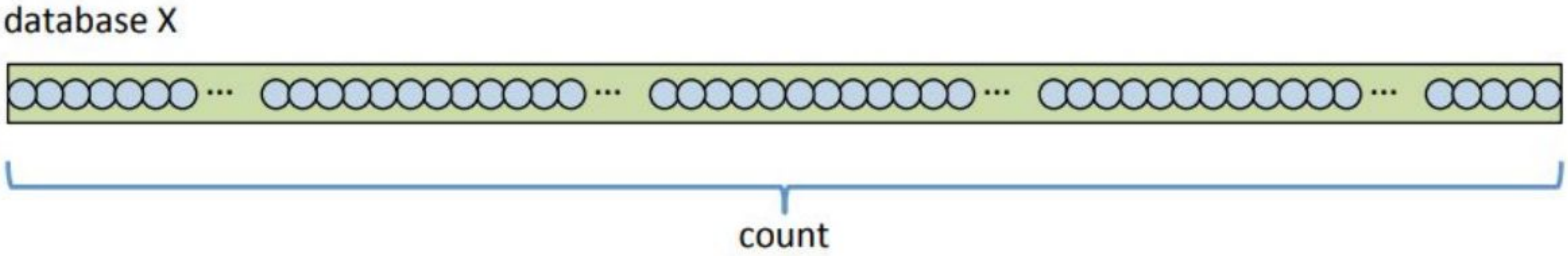


ID#	Model	Year	Color	Dealer	Price
4523	Civic	2002	Blue	MN	$18,000
3476	Corolla	1999	White	IL	$15,000
7623	Camry	2001	Green	NY	$21,000
9834	Prius	2001	Green	CA	$18,000
6734	Civic	2001	White	OR	$17,000
5342	Altima	2001	Green	FL	$19,000
3845	Maxima	2001	Blue	NY	$22,000
8354	Accord	2000	Green	VT	$18,000
4395	Civic	2001	Red	CA	$17,000
7352	Civic	2002	Red	WA	$18,000

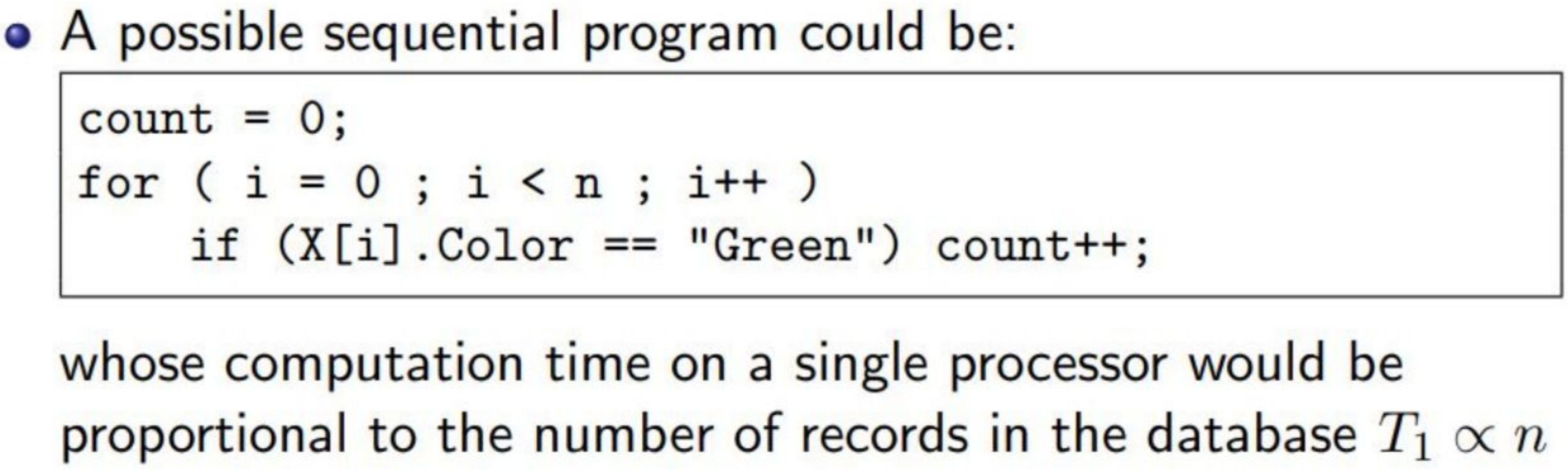
And assume that we want to count how many Green cars are available to sell.

Motivation

- One could traverse all the records `X[0]` ... `X[n-1]` in the database `X` and check if the `Color` field matches the required value `Green`, storing the number of matches in variable `count`



Motivation



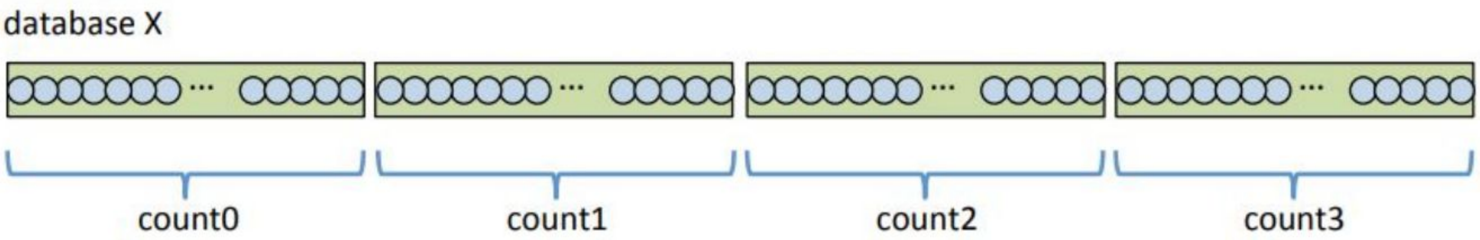
- A possible sequential program could be:

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

whose computation time on a single processor would be proportional to the number of records in the database $T_1 \propto n$

Tasks

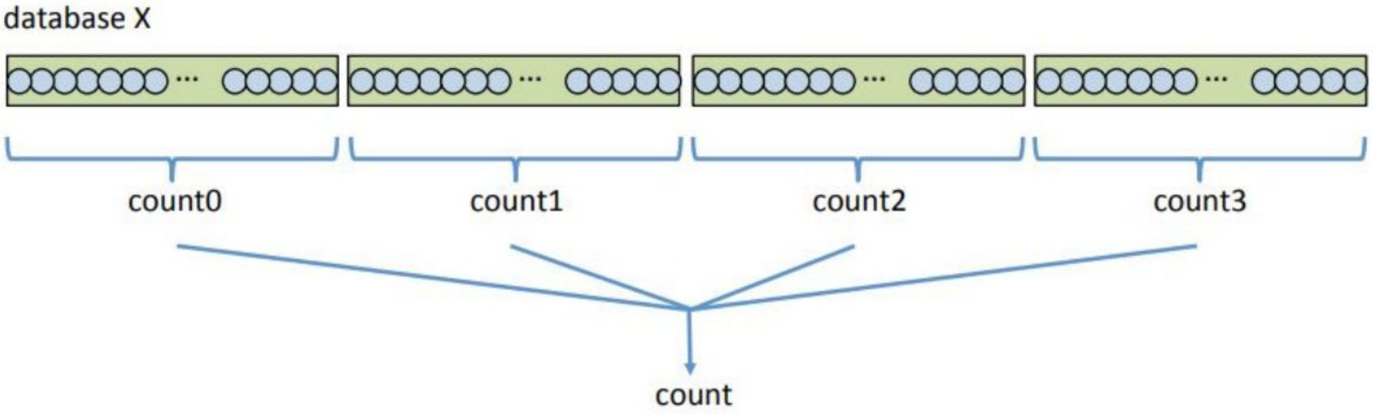
- One could divide the traversal in P groups (tasks), for example for $P = 4$:



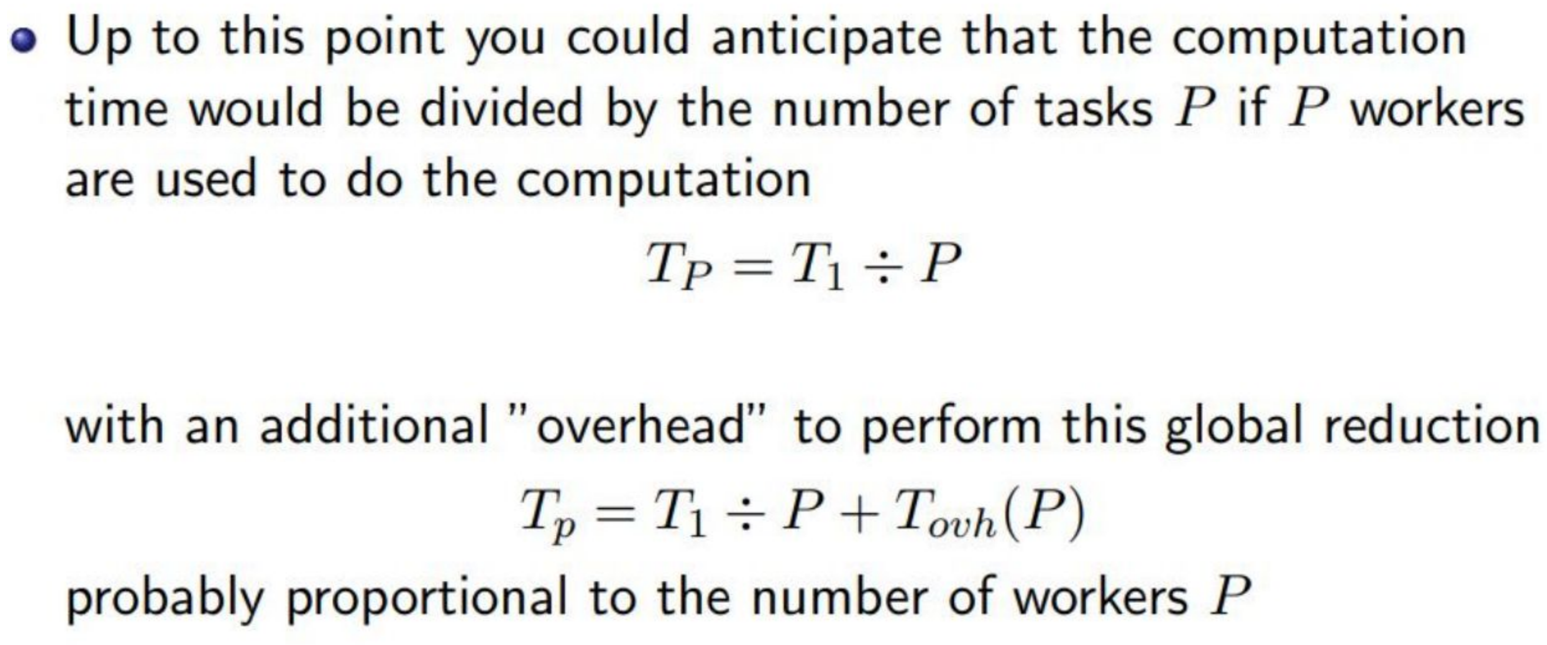
checking the `Color` field for a subset of $n \div P$ consecutive records, and counting on a per–task ”private” copy of variable `count`

Tasks

- However, we still need to "globally" count the number of records found that match the condition by combining the individual "private" counts into the original count variable



Tasks



- Up to this point you could anticipate that the computation time would be divided by the number of tasks P if P workers are used to do the computation

$$T_P = T_1 \div P$$

with an additional "overhead" to perform this global reduction

$$T_p = T_1 \div P + T_{ovh}(P)$$

probably proportional to the number of workers P

Instructor Social Media

Youtube: Lucas Science

Instagram: lucaasbazilio

Twitter: lucasebazilio