# VIDEO SCRIPT

## Slide 1

The doubly linked-list is a well node data structure where the data are stored in a list of nodes each one pointing, not only to the next one, but also to the previous one. On both ends, we have what we call anchors which are fixed to the left, and the right.

## Slide 3

When we want to add a node to the list, we make it point to its neighbors and then, change the pointers accordingly.

## Slide 8

But now, if multiple threads want to make modify the list, we need to make it thread safe.

## Slide 9

Most of the algorithms, lock-free or not, use the same ACQUIRE, APPLY and RELEASE phases. If during the acquire phase, a process needs a lock owned by another process, it waits for it to release it and it thus creates a hold-and-wait chain

## Slide 10

In the Built-in Coloring for Highly Concurrent Doubly-Linked Lists paper, the authors came up with an innovative algorithm allowing us to reduce the size of the hold-and-wait chain.

## Slide 11

This algorithm is called CAS-Chromo and is completely lock-free. As its name says, it uses Compare and Swap operations to make changes atomically.

## Slide 12

The idea is that each node of the list will have a color different to the one of its neighbors in addition to a reference to the operation that is using it.

## Slide 13

Talking about operations, those have a state, a subject node which is the one it has to add or remove and a source node which is adjacent to it.

The Memento contains a copy of the nodes metadata before starting the operation. That way, we will be able to detect inconsistencies.

## Slide 14

The available operations of CAS-chromo are listed here.

## Slide 15

We now have 2 processes willing to perform 2 different operations. A push left and a pop left.

## Slide 16

Both operations are in their INIT state and the green one has already prepared its new node.

## Slide 17

Let's say the green operation first goes to the acquire phase. It acquires the red node as it does only need to acquire the two leftmost node to perform a push operation.

## Slides 18

Now, the orange process wants to acquire the 3 leftmost nodes. It gets the green first but cannot go further as the red one is already owned by the green process.

## Slide 19

Instead of waiting for it, CAS introduces the concept of helping. That way the orange process will help performing the push left.

## Slide 20

And the node acquisition can continue.

## Slide 21

As all the needed nodes have been acquired, we can move the push operation to its apply phase where it will point to its neighbors

## Slide 22

Get a legal color which is different from the one of its neighbors

## Slide 23

And be pointed by its right

## Slide 24

And left neighbor.

## Slide 25

The operation continues to the release phase

## Slide 26

and ends in the final one.

## Slide 27

The pop left operation can now continue but thanks to its Memento, it detects that there have been changes in the list

## Slide 28

It successively moves to a release with contention

## Slide 29

And a final with contention state

## Slide 30

Before going back to the initial state

## Slide 31

It can then acquire the nodes without any trouble

## Slide 32

Start applying changes by first

## Slide 33

decoloring the right node to make sur the changes will maintain a legal coloring

## Slide 34

changing the next

## Slide 35

and previous references

## Slide 36

and finally getting the node

## Slide 37

before re-applying a legal color to the node 1 and moving to the release phase.

## Slide 38

after releasing the nodes, the operation is complete and moves to the final phase