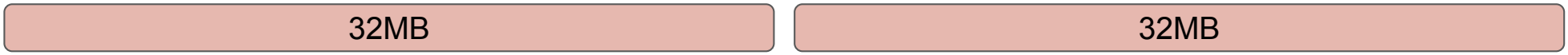
Encoding Data Under the Clay Code

Consider a file of size 64MB

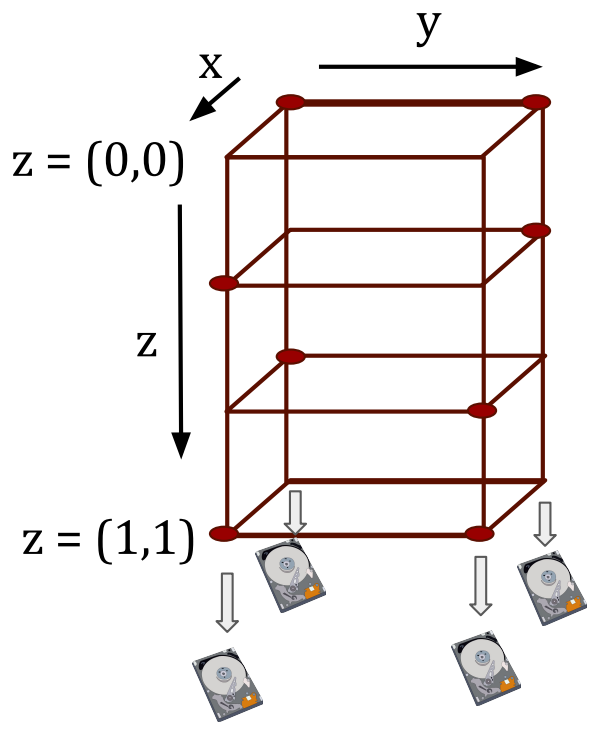
- We show encoding of the file using (k = 2, m= 2) Coupled Layer MSR code.

Break the file into k = 2 data chunks each of 32MB.



3D cube representation of Coupled Layer Code

32MB	32MB

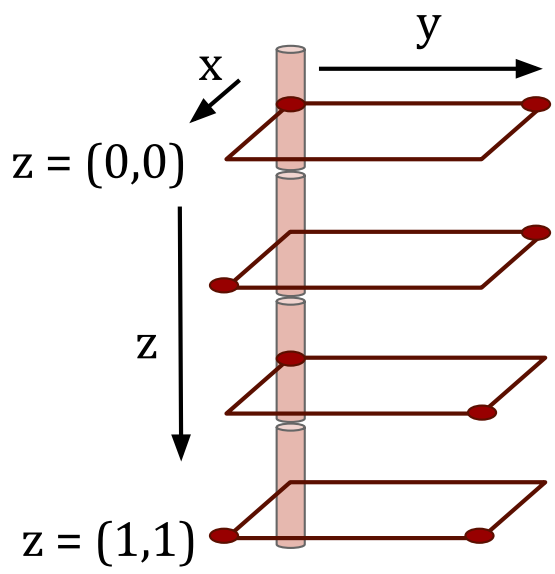


The cube has:

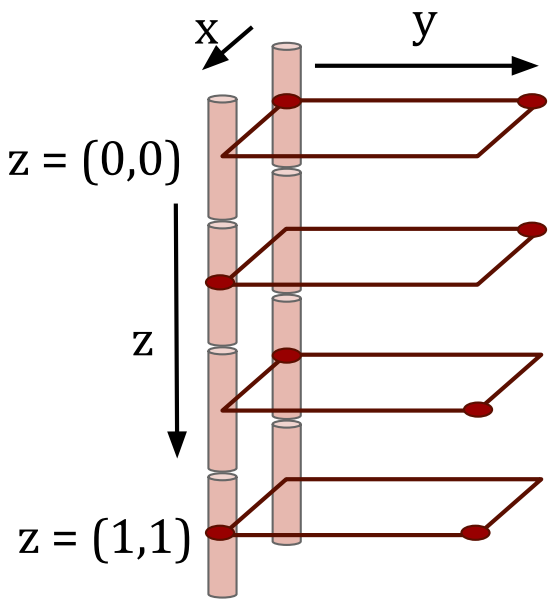
- 4 columns, which correspond to the 4 chunks (each of size 32MB, stored in a different disk/node).
- 4 horizontal planes.
- Each column has 4 points that correspond to sub-chunk of size 8MB

Place two 32MB chunks in two data nodes

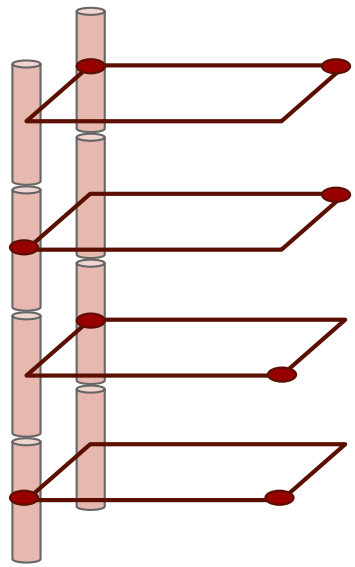
32MB



Place two 32MB chunks in two data nodes

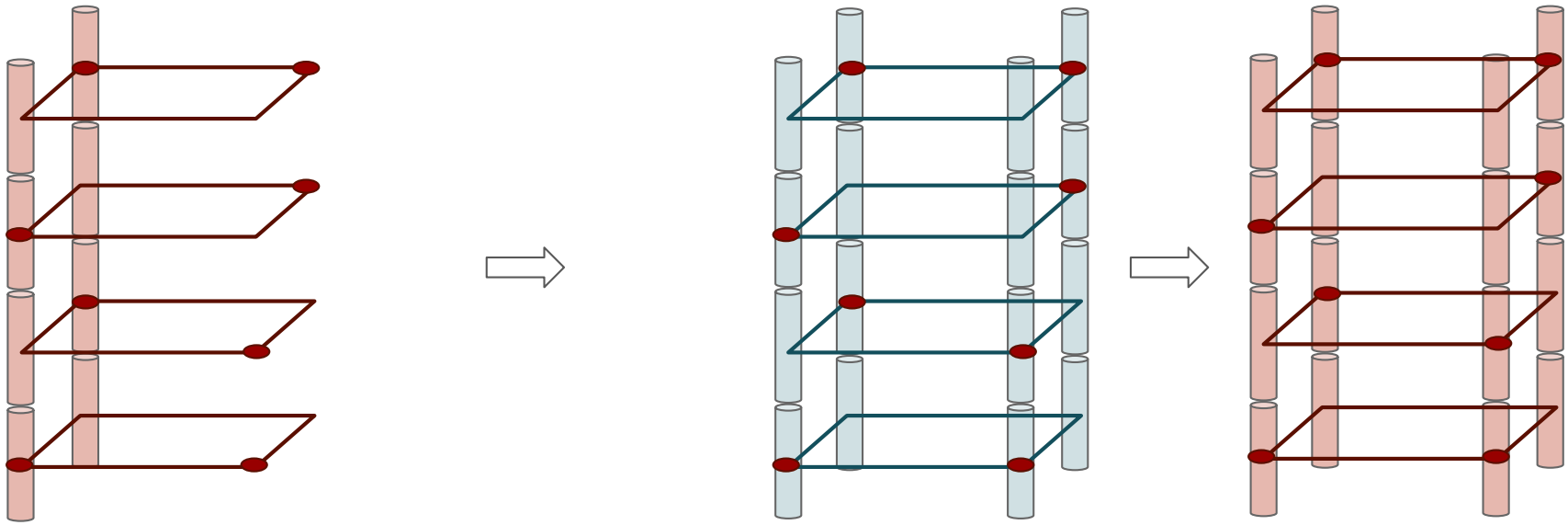


We now have the systematic nodes

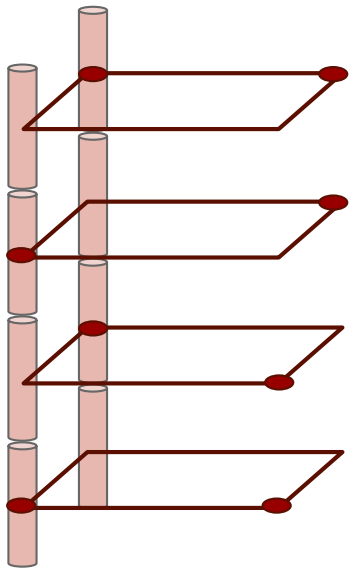


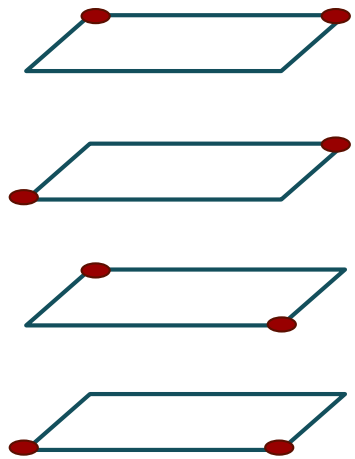
We will now compute the parity nodes

Will get there through an intermediate “Uncoupled data cube”

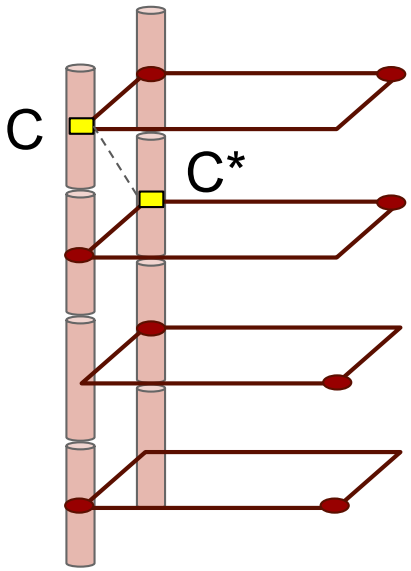


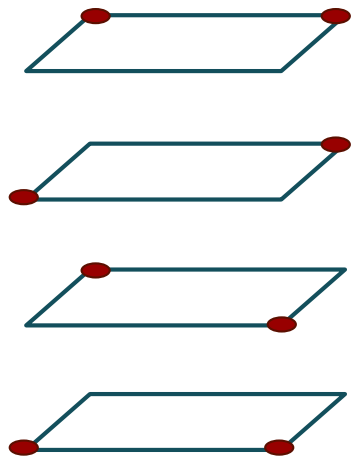
Start filling the virtual data cube on the right as follows



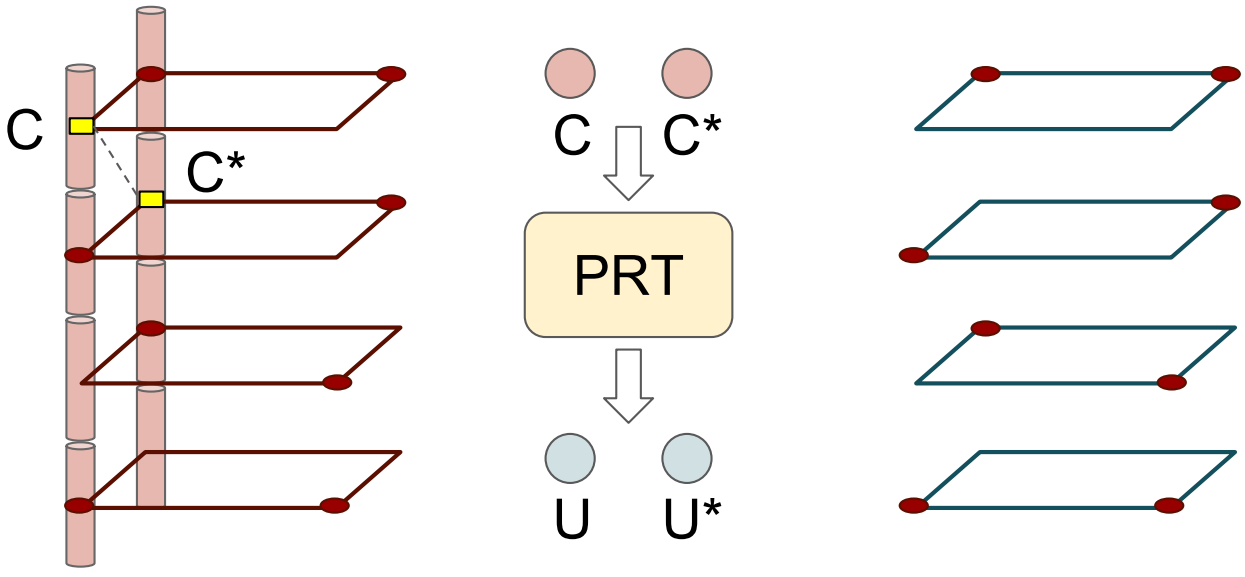


Certain pairs of points in the cube are “coupled”

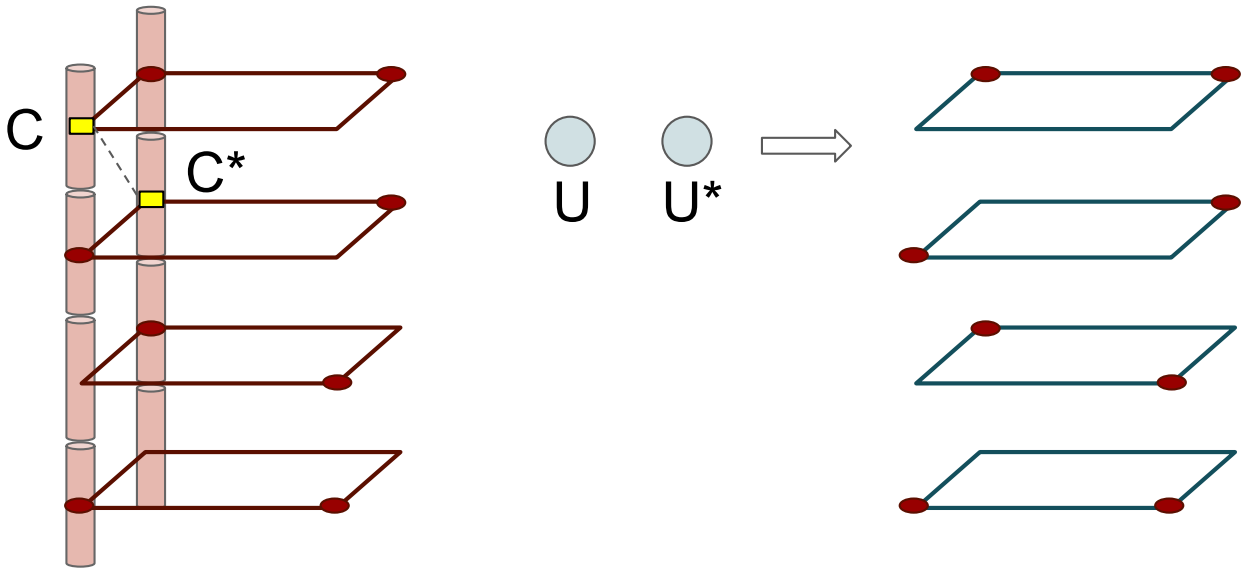




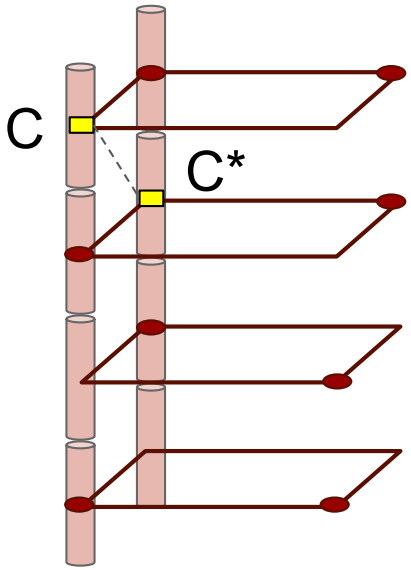
PRT is a 2x2 matrix transform

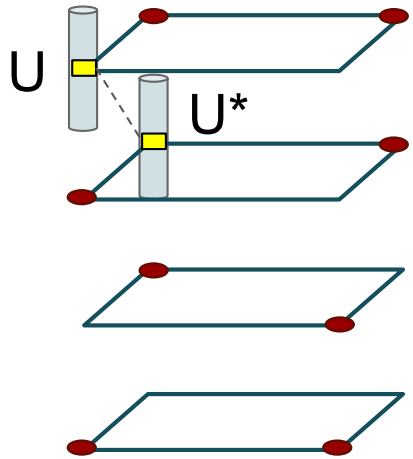


Place the sub-chunks obtained in the uncoupled data cube

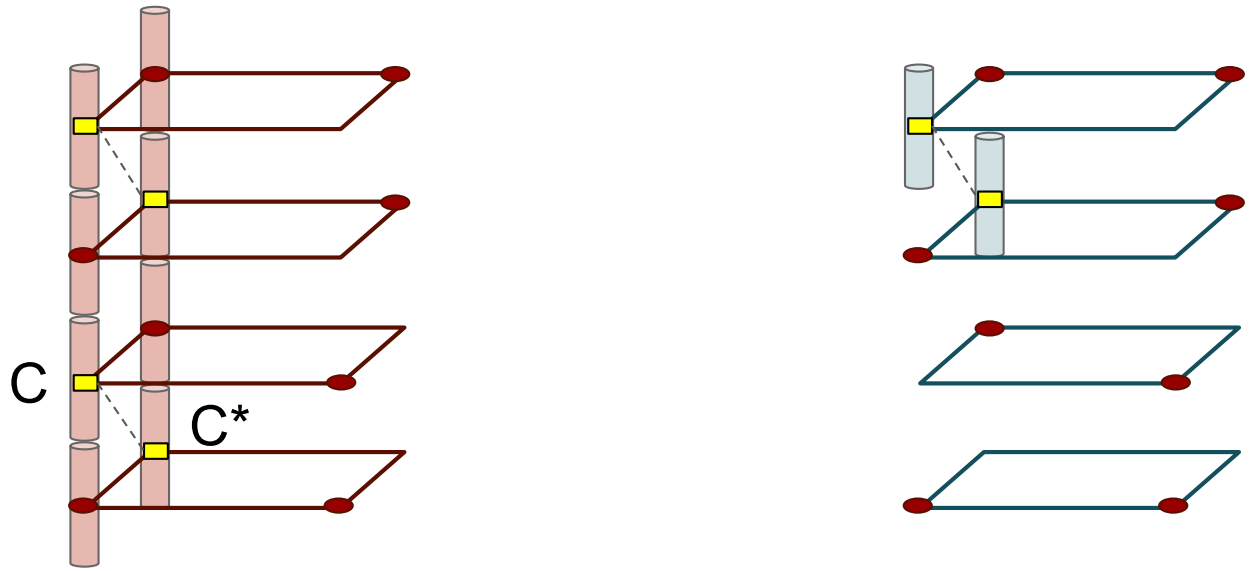


Place the sub-chunks obtained in the uncoupled data cube

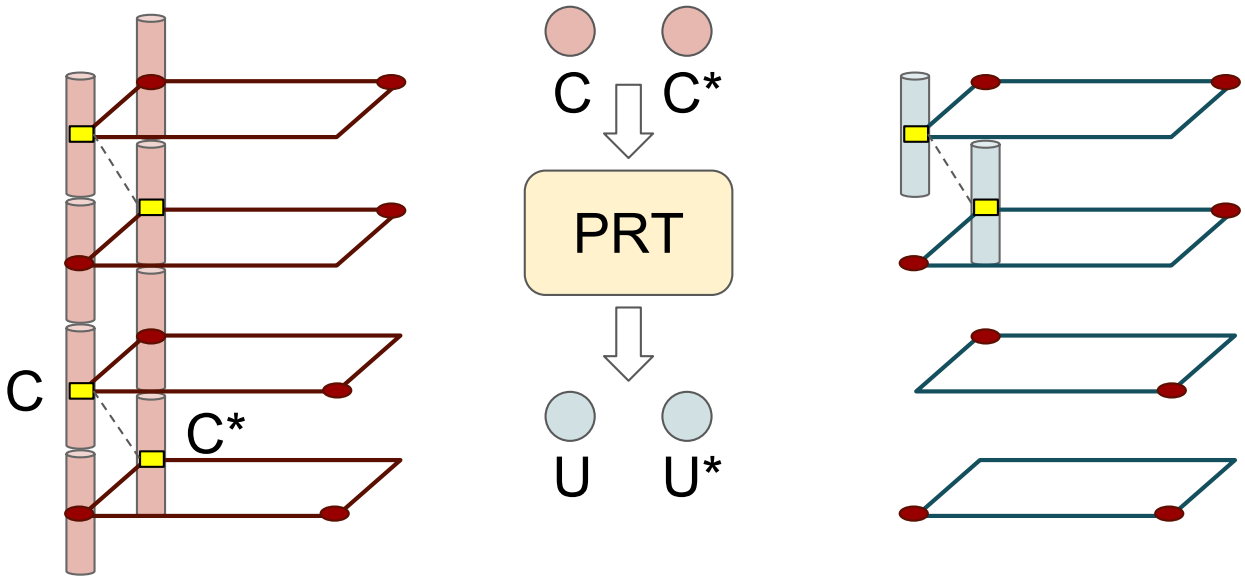




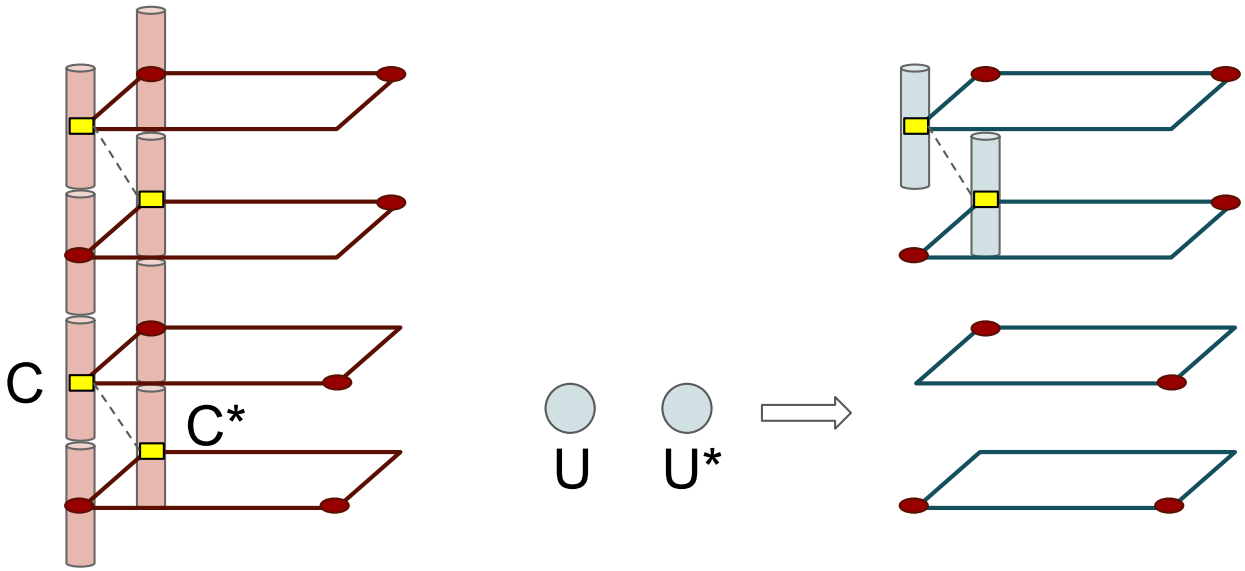
Place the sub-chunks obtained in the uncoupled data cube



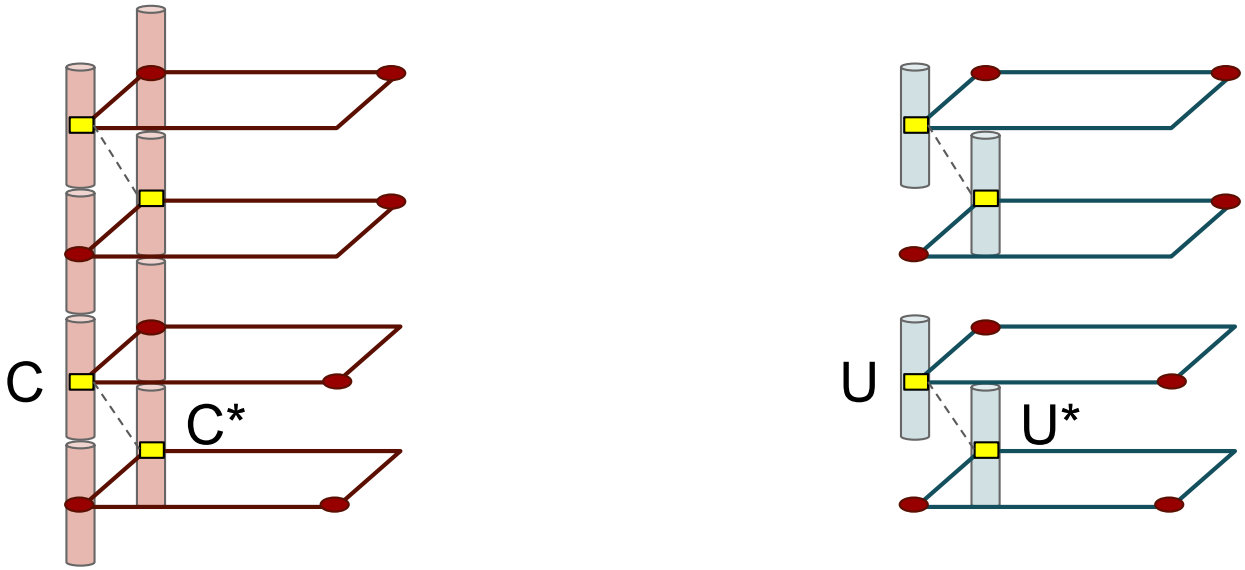
Place the sub-chunks obtained in the uncoupled data cube



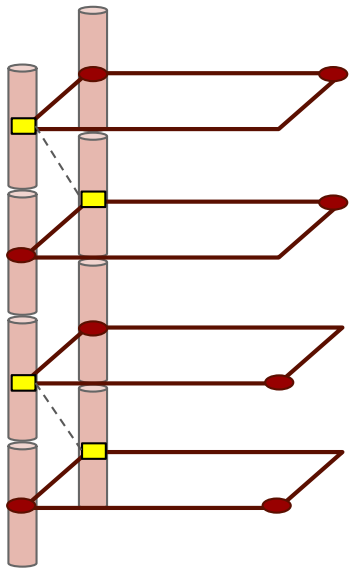
Place the sub-chunks obtained in the uncoupled data cube

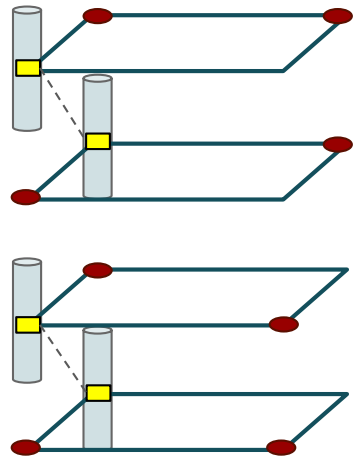


Place the sub-chunks obtained in the uncoupled data cube

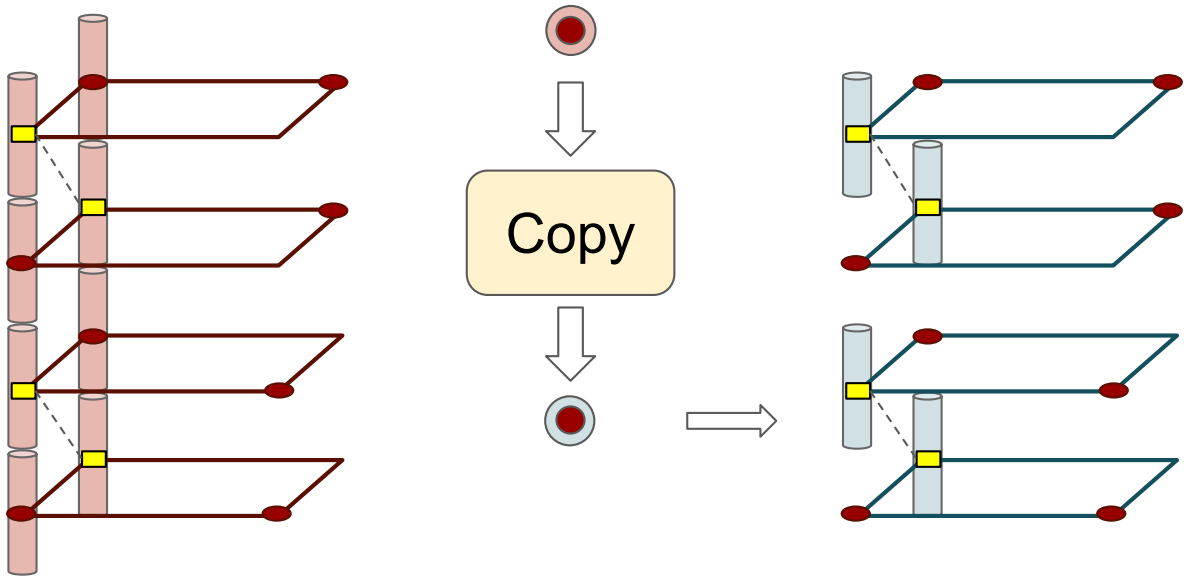


Place the sub-chunks obtained in the uncoupled data cube

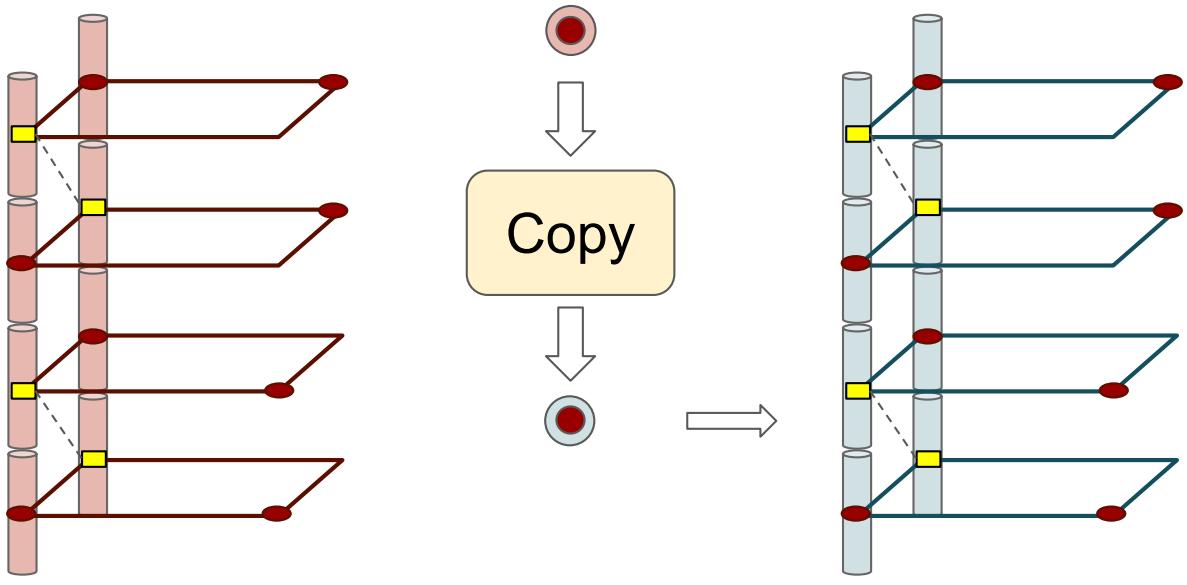




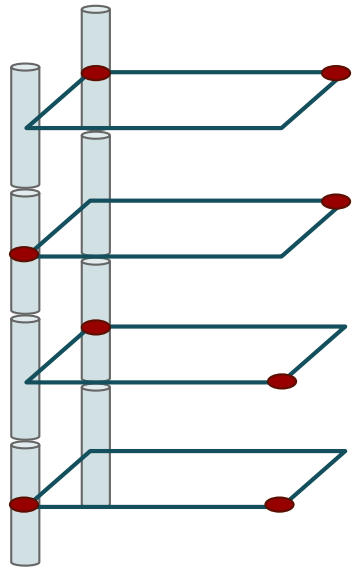
Red dotted sub-chunks are not paired, they are simply carried over



Red dotted sub-chunks are not paired, they are simply carried over

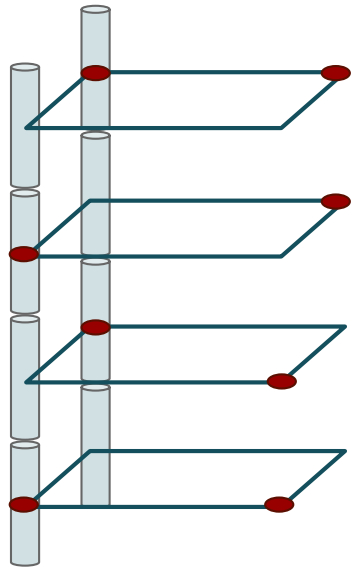


We now have data-part of the uncoupled data cube



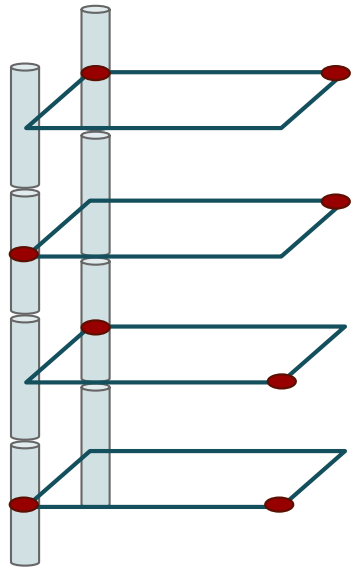
Each plane is Reed-Solomon encoded to obtain parity points (sub-chunks)

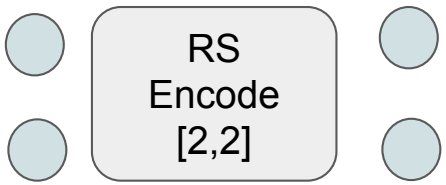
$z = (0,0)$



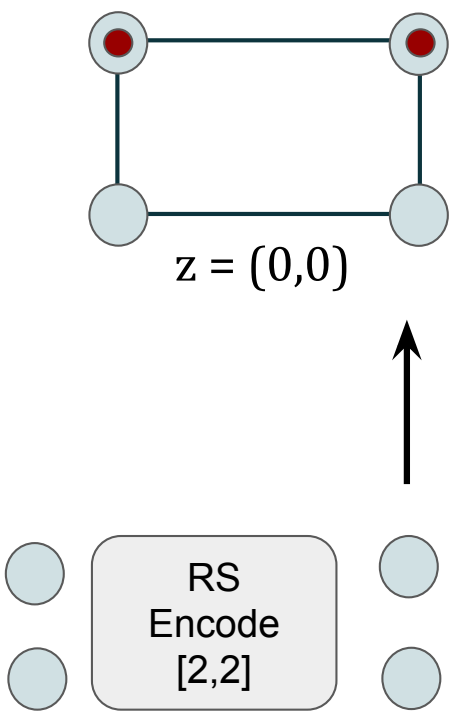
Each plane is Reed-Solomon encoded to obtain parity points (sub-chunks)

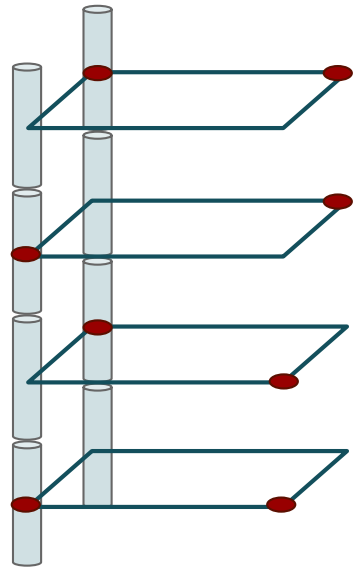
$z = (0,0)$



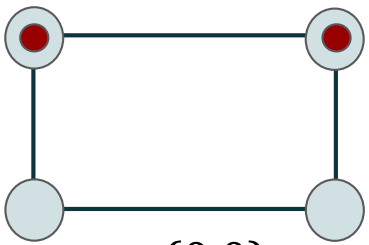


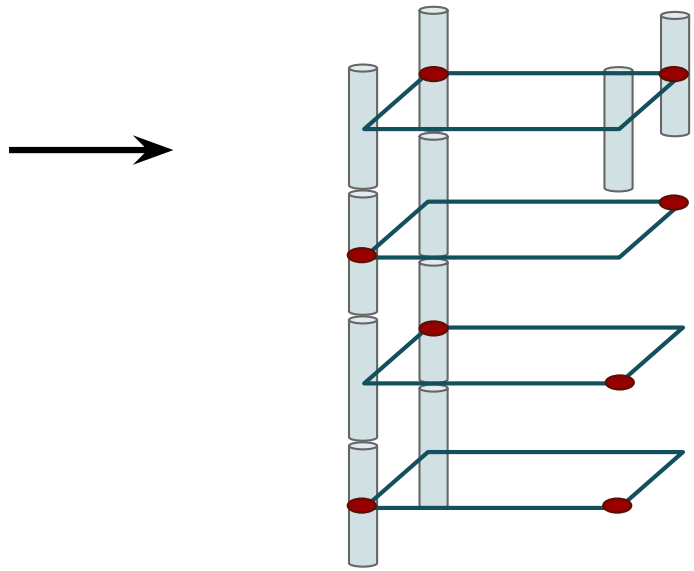
Each plane is Reed-Solomon encoded to obtain parity points (sub-chunks)



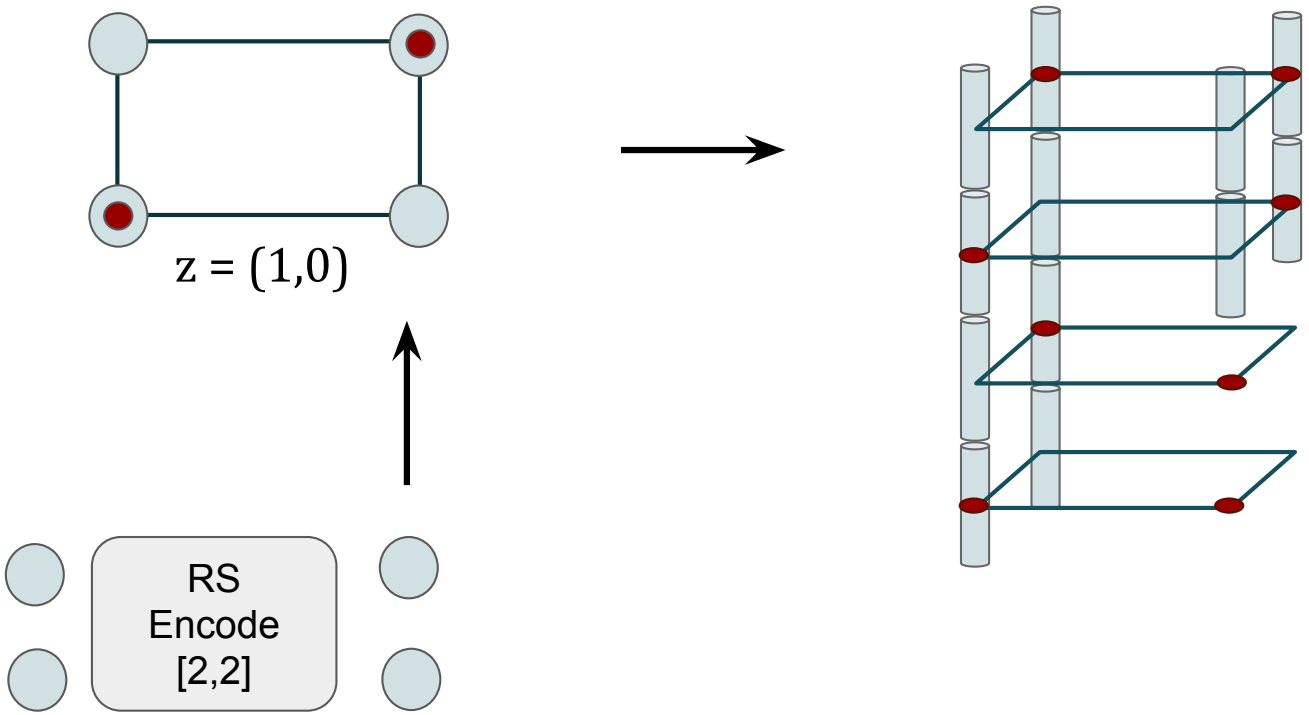


Each plane is Reed-Solomon encoded to obtain parity points (sub-chunks)

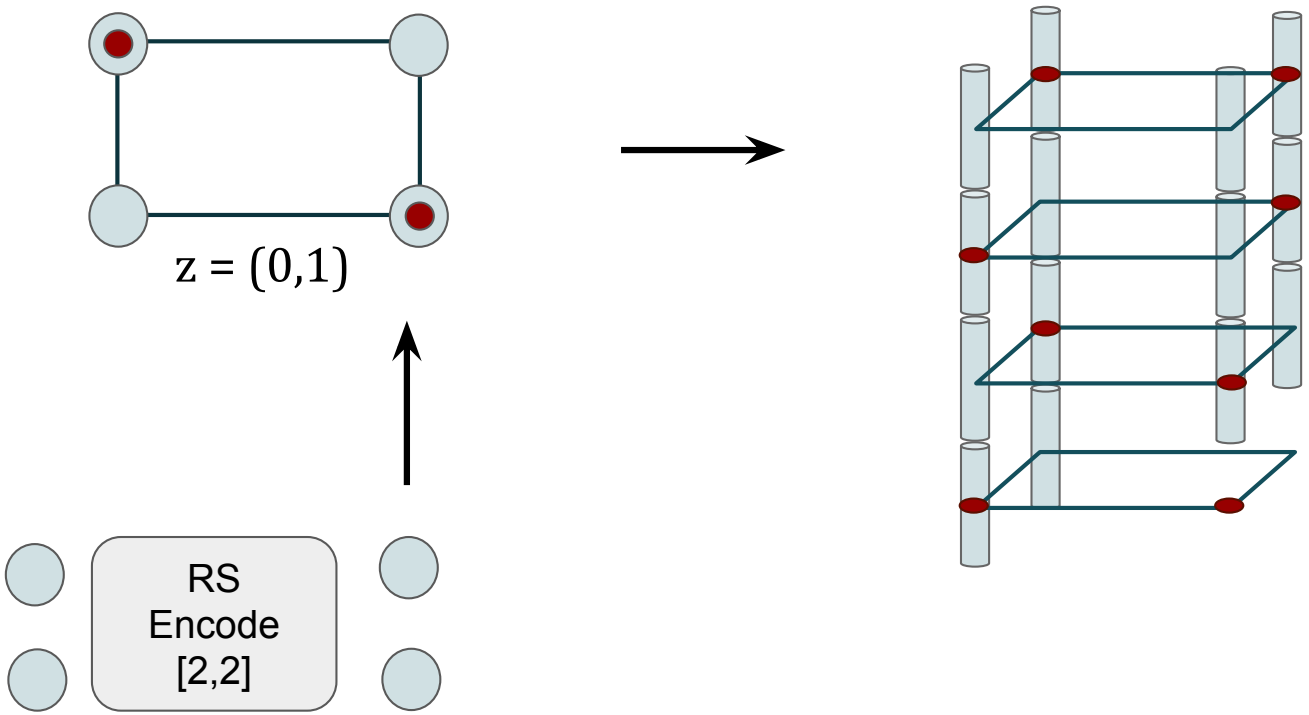




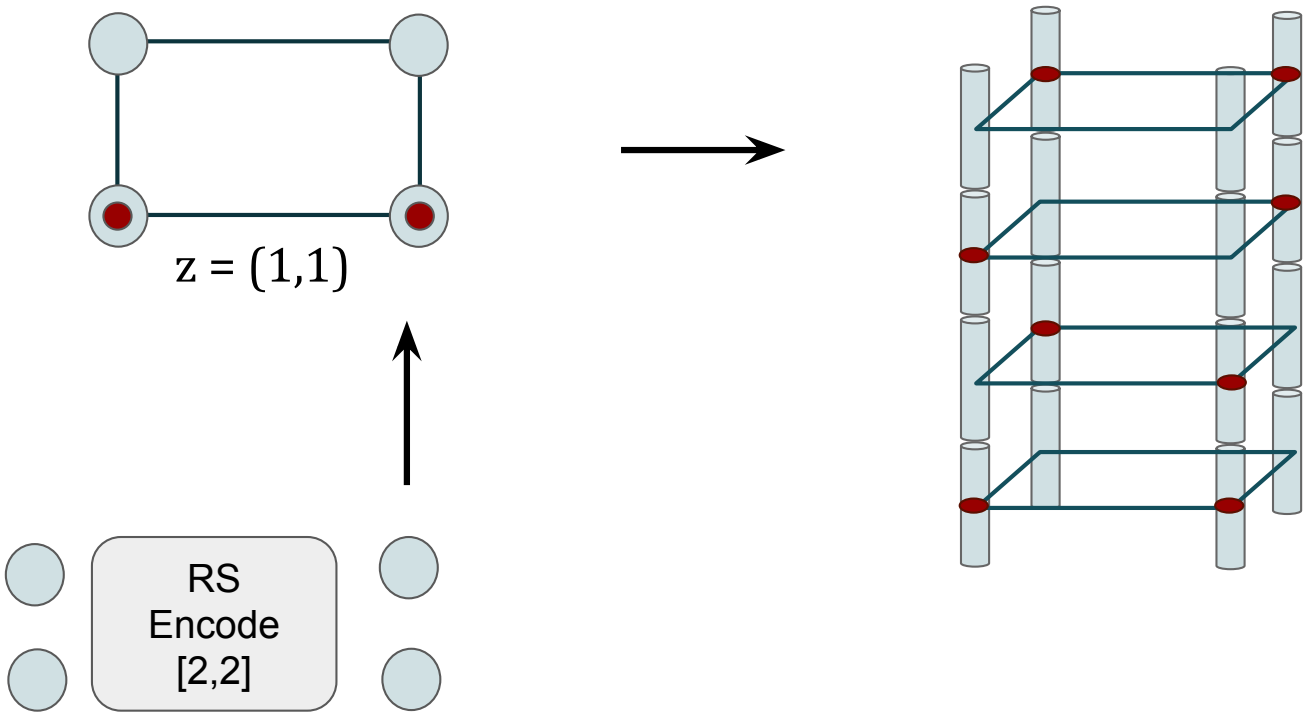
Each plane is Reed-Solomon encoded to obtain parity points (sub-chunks)



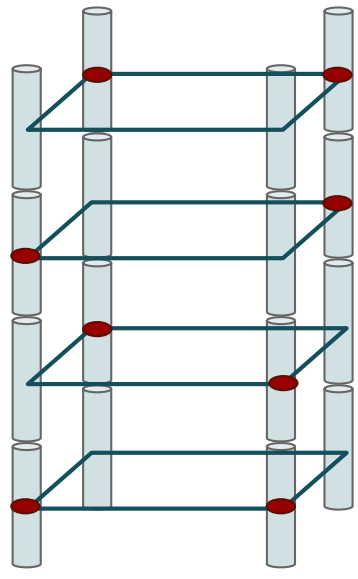
Each plane is Reed-Solomon encoded to obtain parity points (sub-chunks)



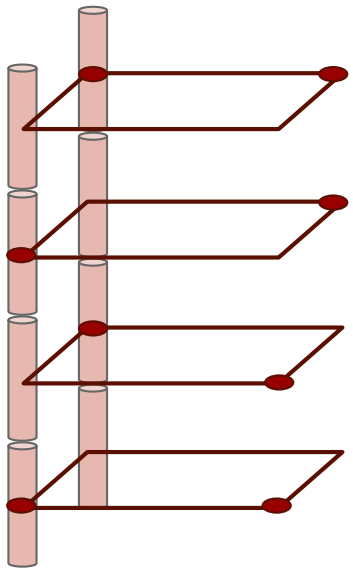
Each plane is Reed-Solomon encoded to obtain parity points (sub-chunks)

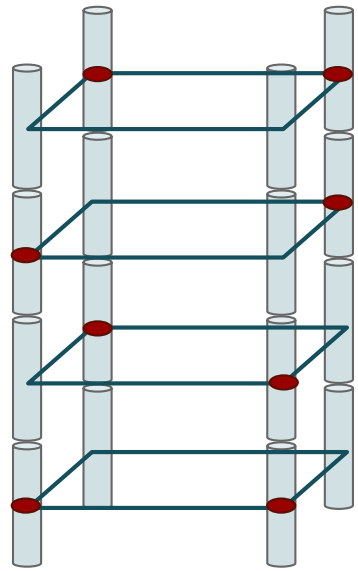


Now we have the complete Uncoupled data cube

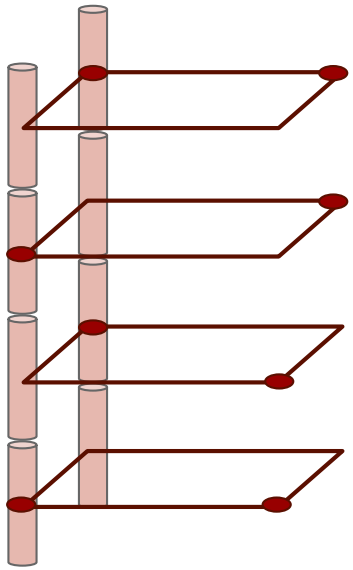


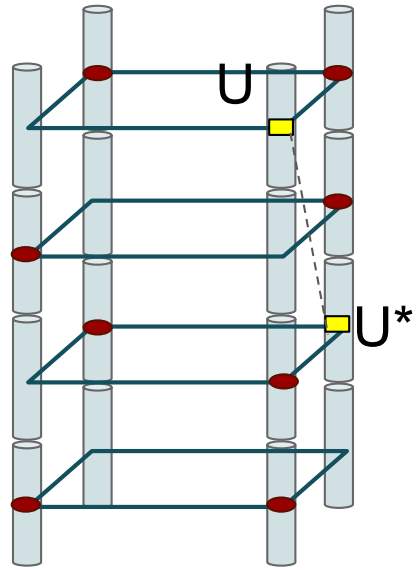
Parity sub-chunks of Coupled data cube can now be computed



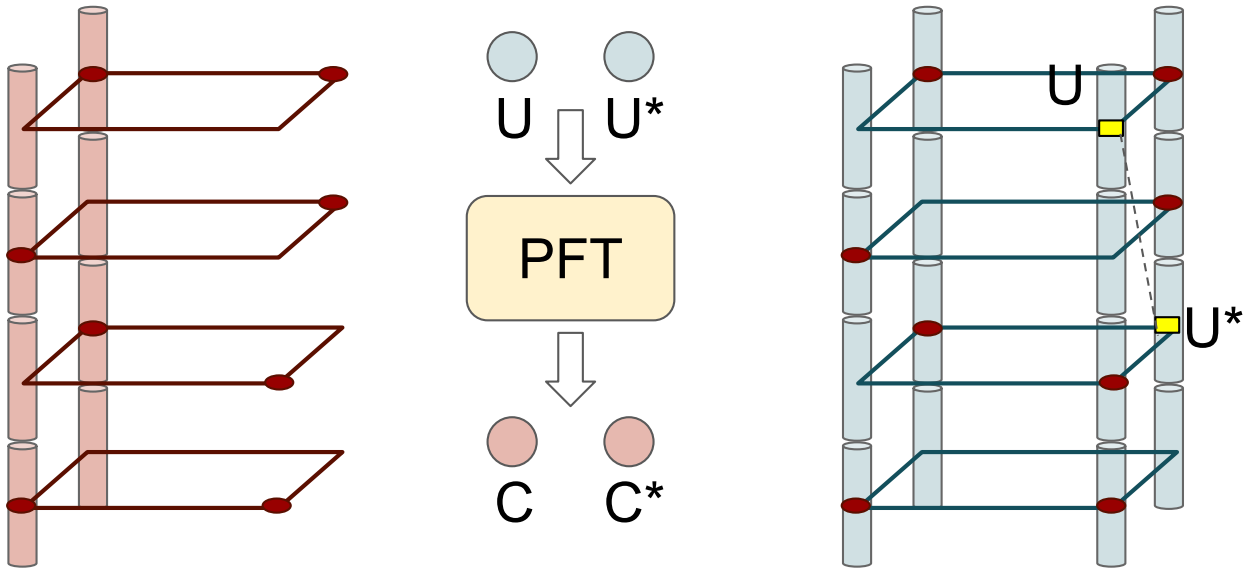


Perform PFT

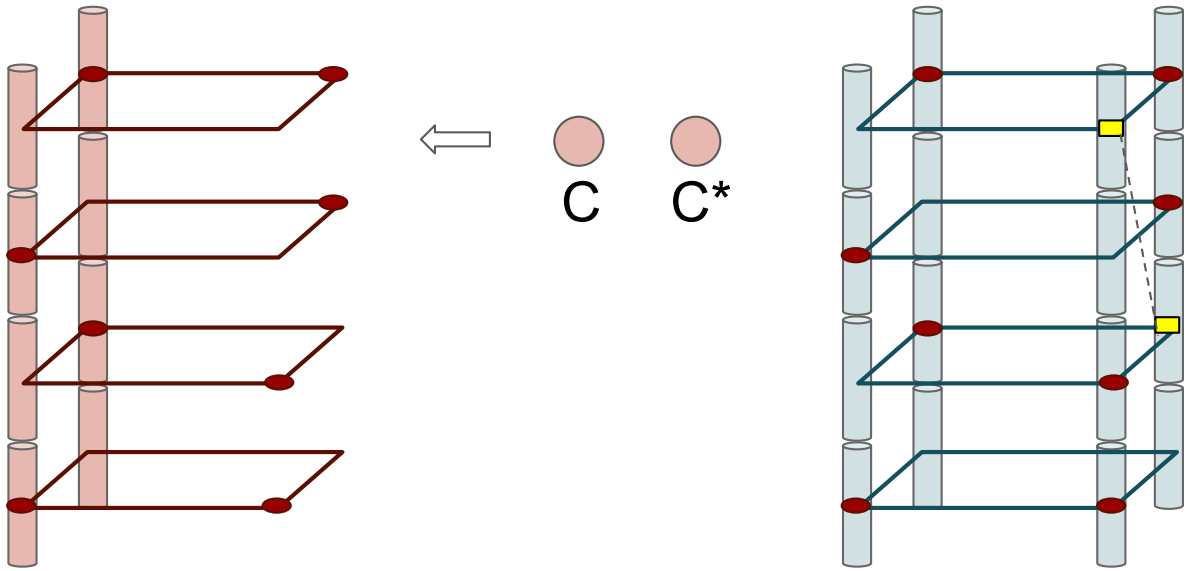




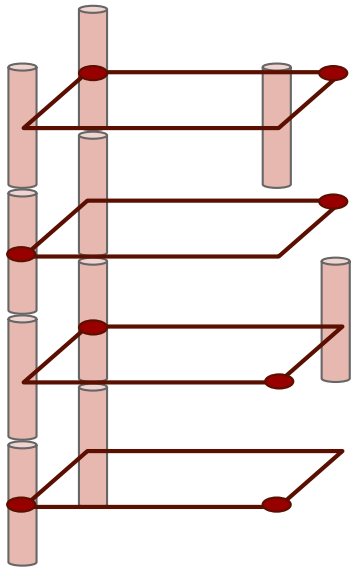
Perform PFT

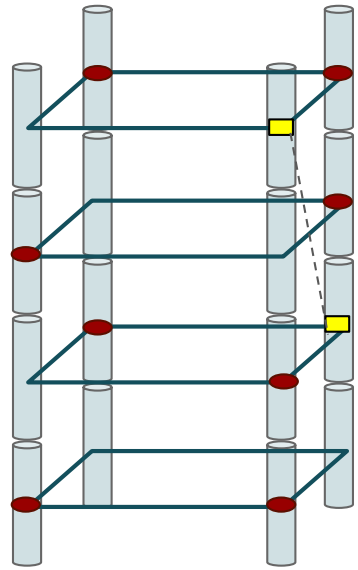


Perform PFT

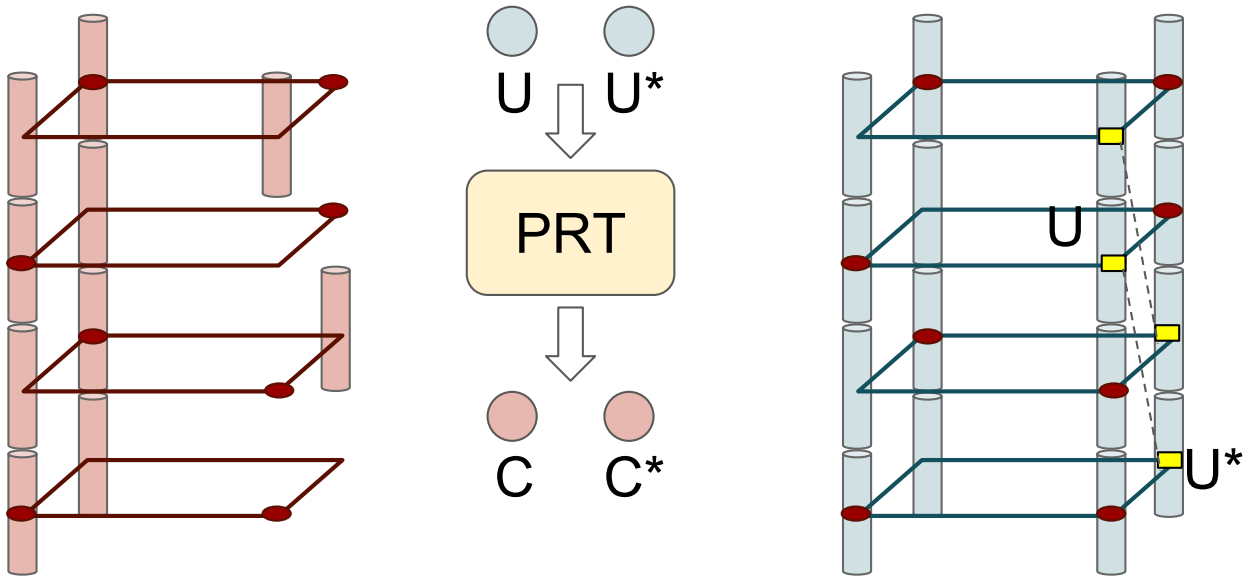


Perform PFT

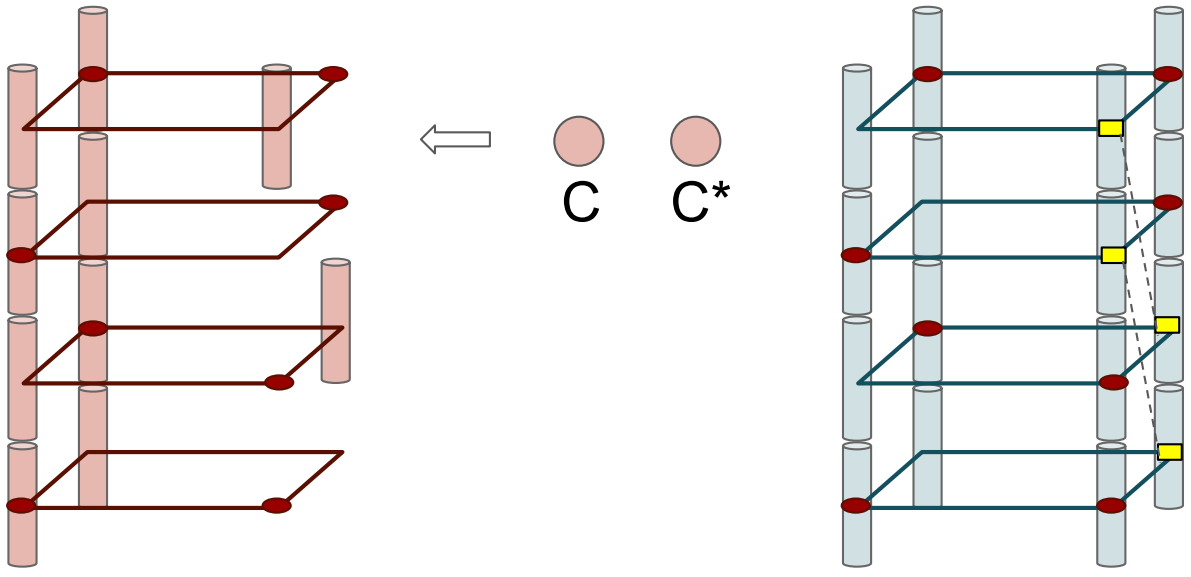




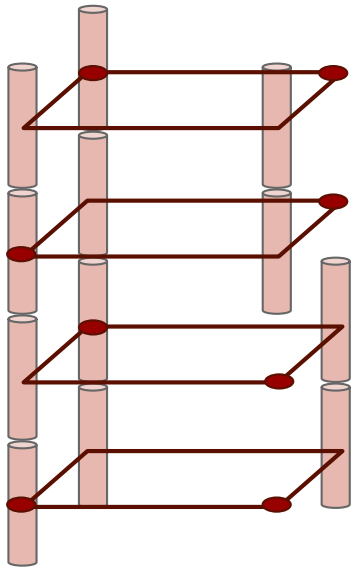
Perform PFT

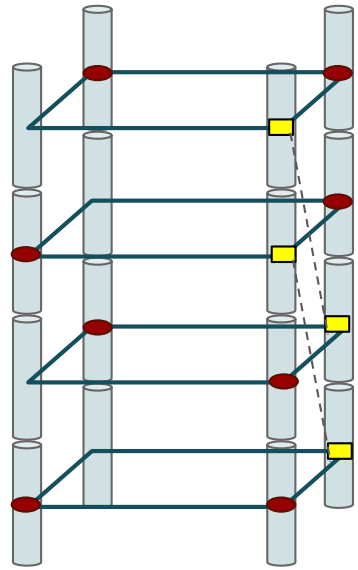


Perform PFT

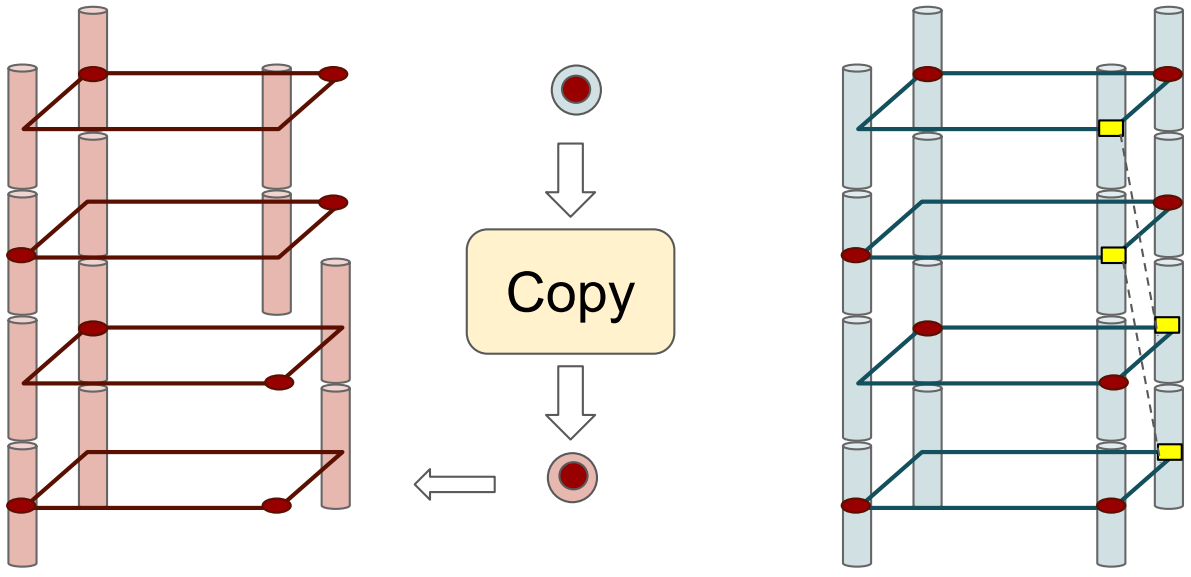


Perform PFT

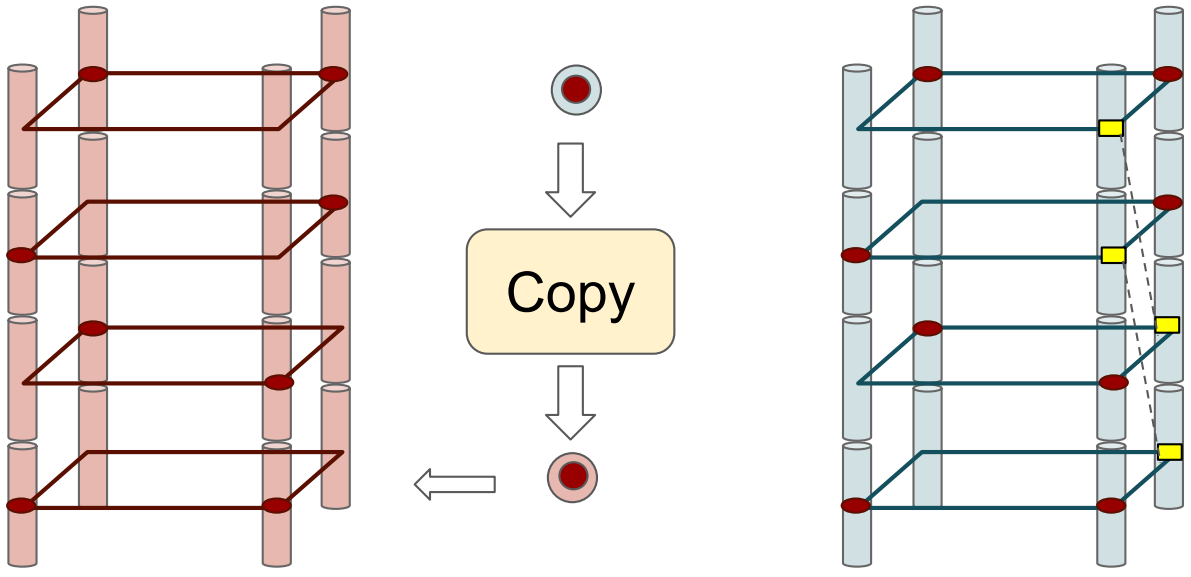




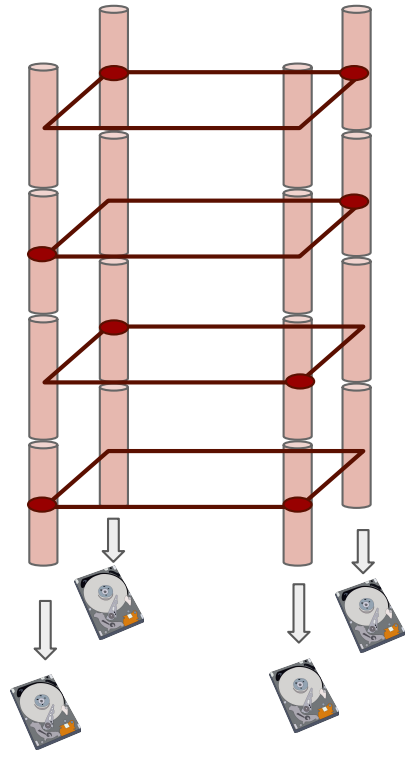
Red dotted sub-chunks are simply carried over



Red dotted sub-chunks are simply carried over

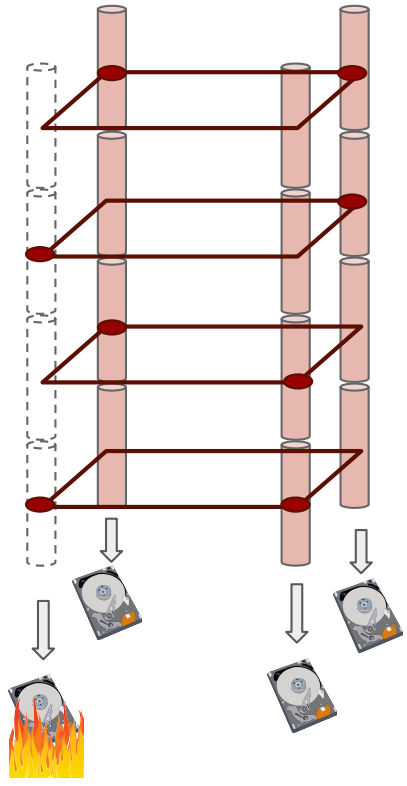


The encoding is now complete!

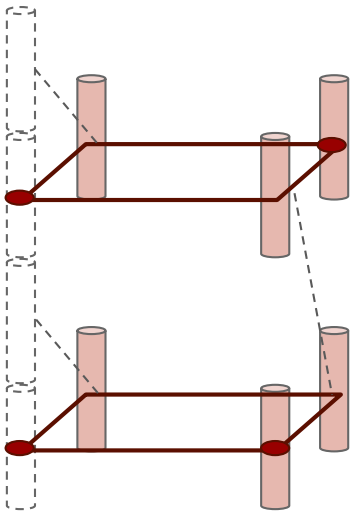


Recovery from single node failure

Node Repair: One node fails

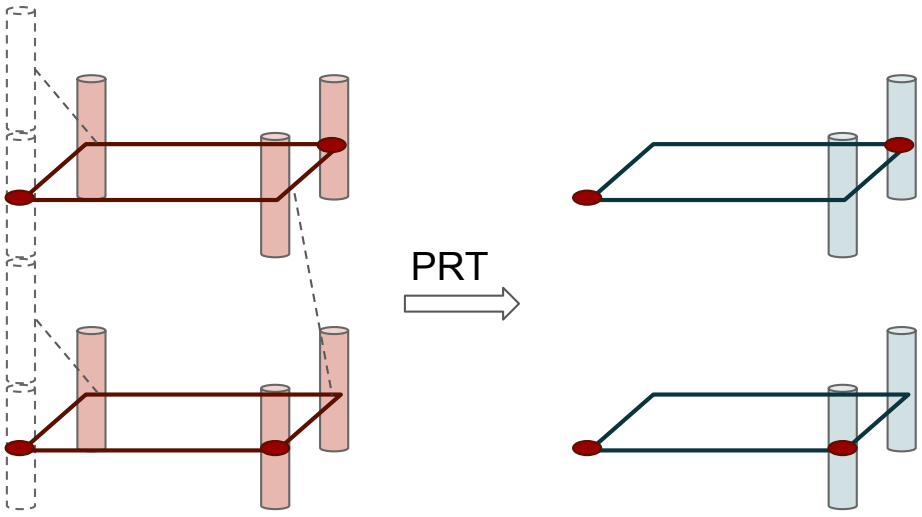


Only half of planes participate in repair

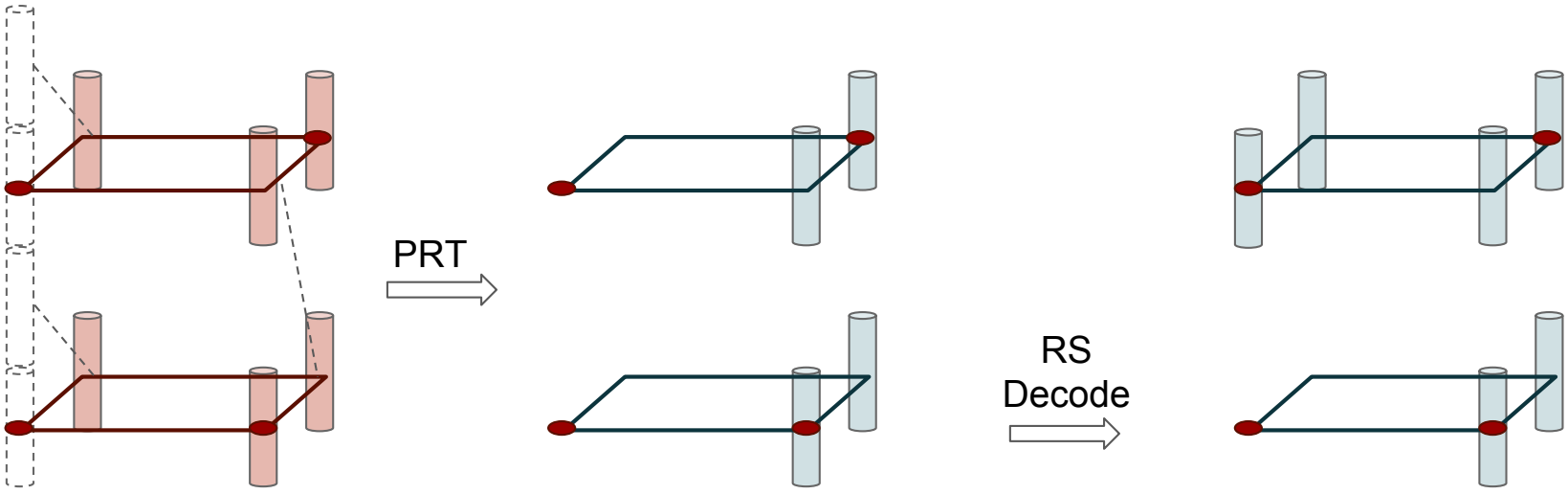


- Total Helper Data = 8MB X 3 X 2 = 48MB
- Opposed to RS code = 32MB X 2 = 64MB
- Much larger savings seen for m > 2

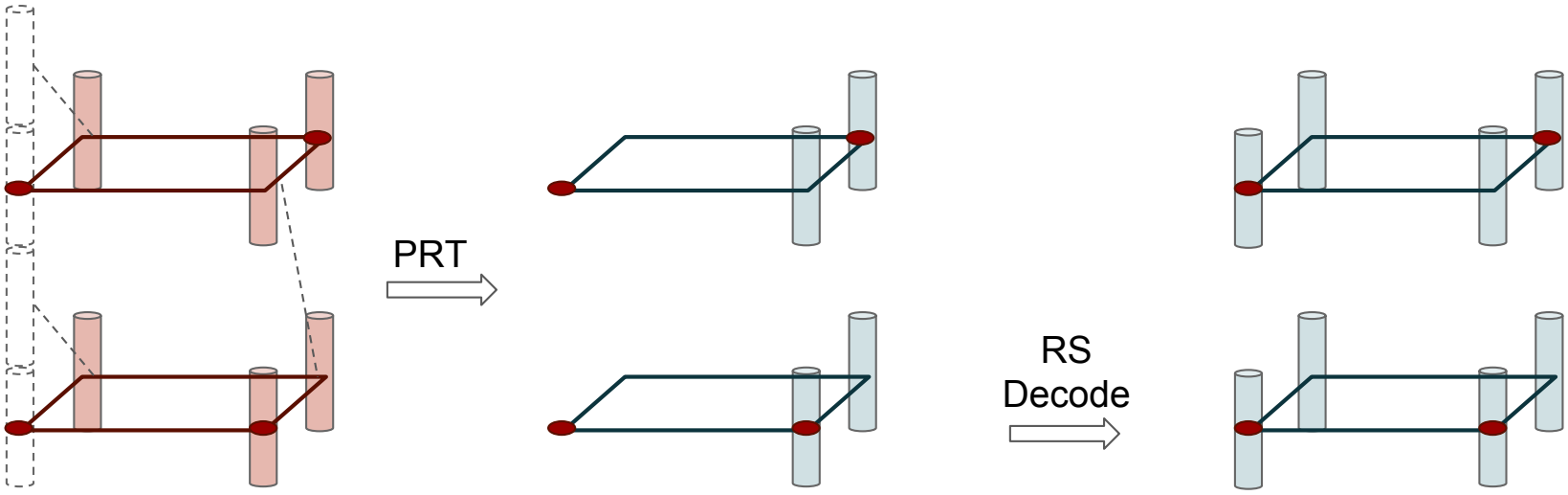
Perform PRT to get possible uncoupled sub-chunks



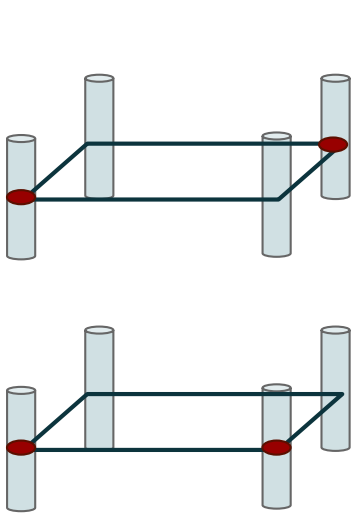
Run RS decoding on each of the selected planes

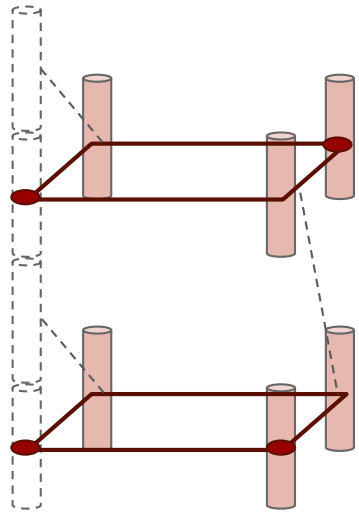


Run RS decoding on each of the selected planes

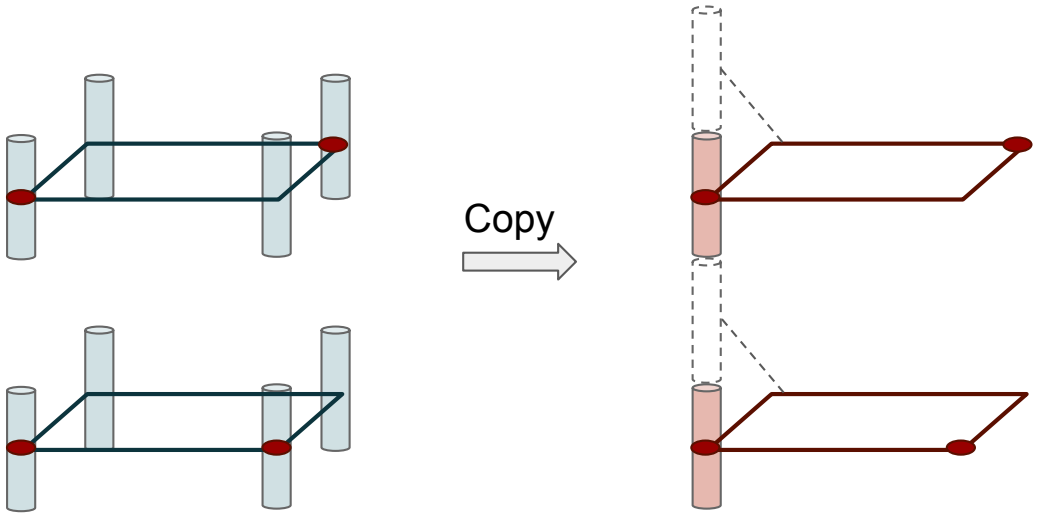


We now have the following sub-chunks available

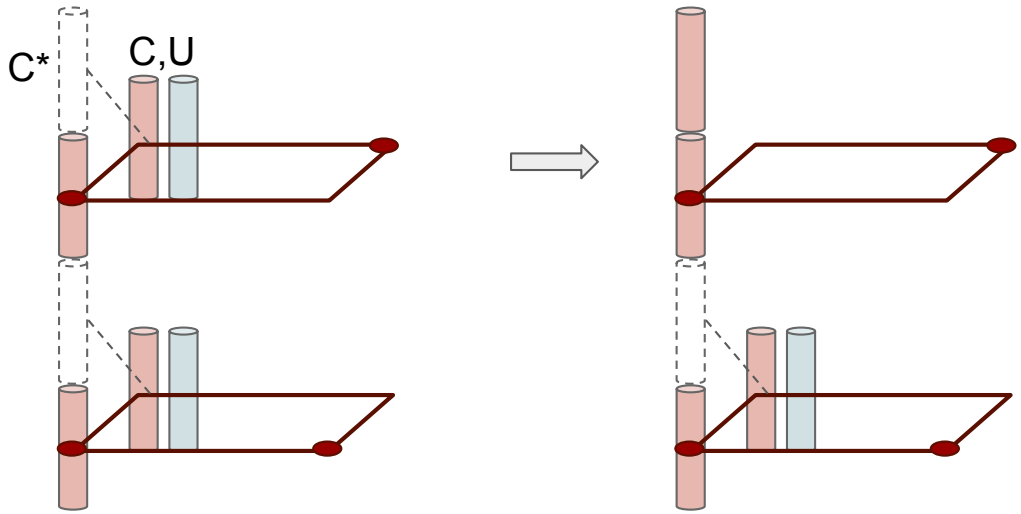




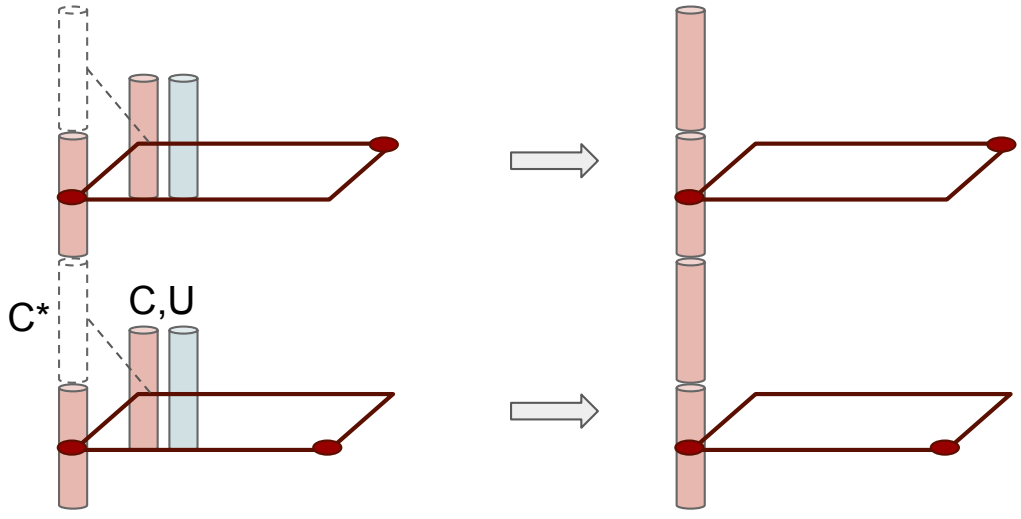
Half the number of required sub-chunks are now already computed



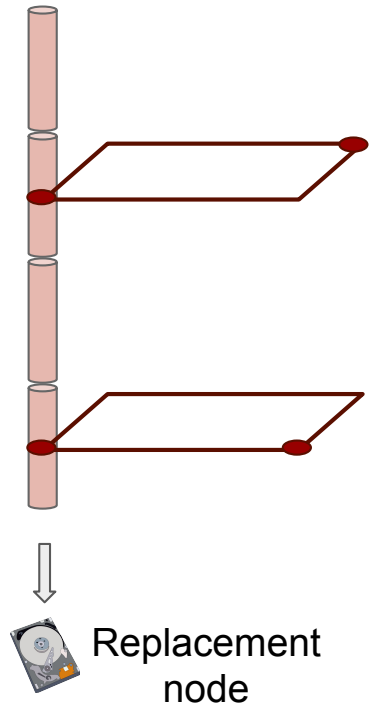
Compute C* from C and U



Compute C* from C and U

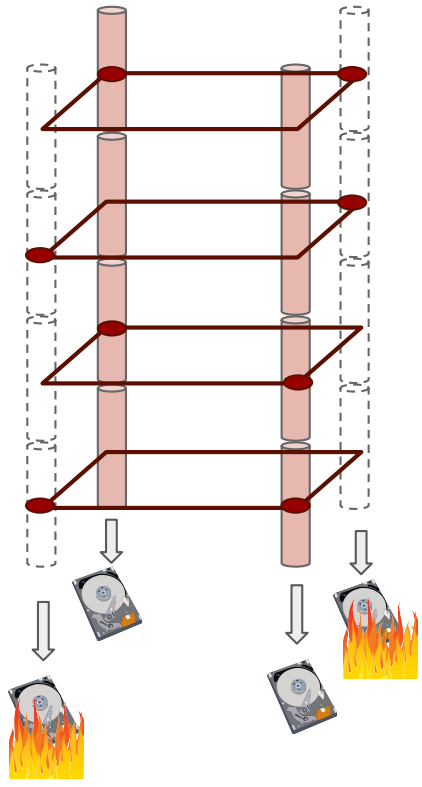


Content of failed node is now completely recovered

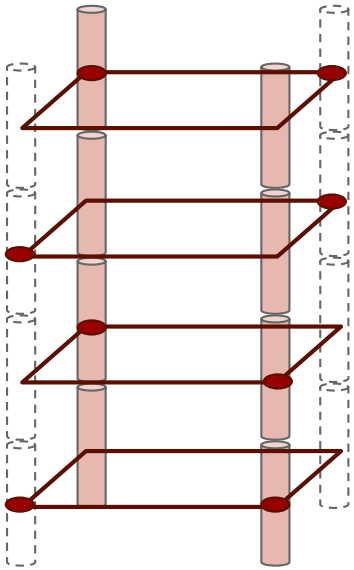


MDS Property of Clay Code

Decode: Two nodes fail



Assign Intersection Score to each plane



z = (0,0)

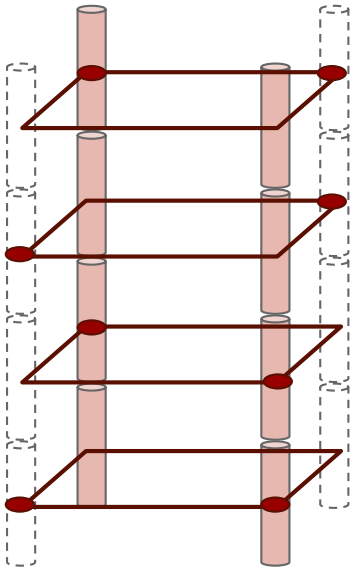
z = (1,0)

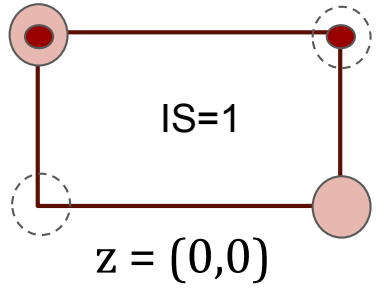
z = (0,1)

z = (1,1)

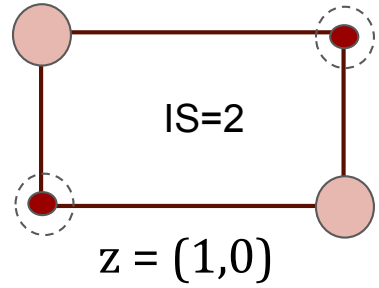
Intersection score is given by the number of hole-dot pairs

Assign Intersection Score to each plane

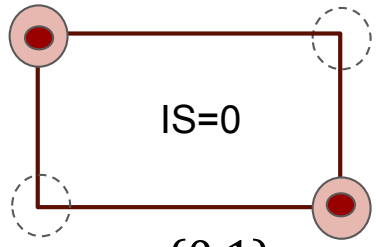




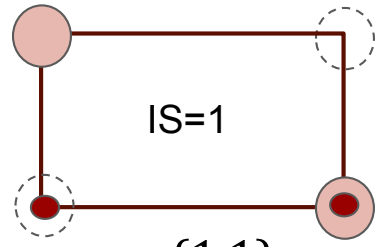
$z = (0,0)$



$z = (1,0)$



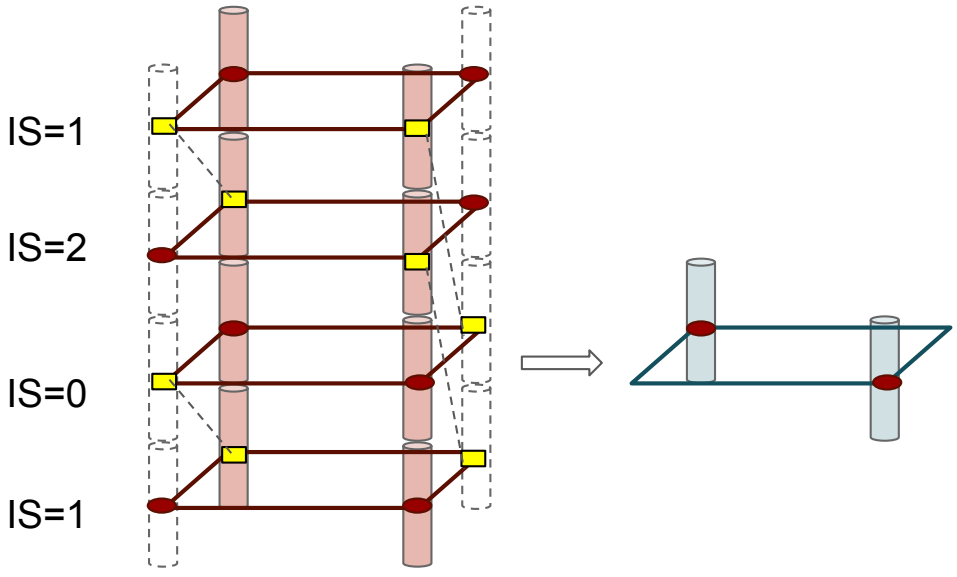
$z = (0,1)$



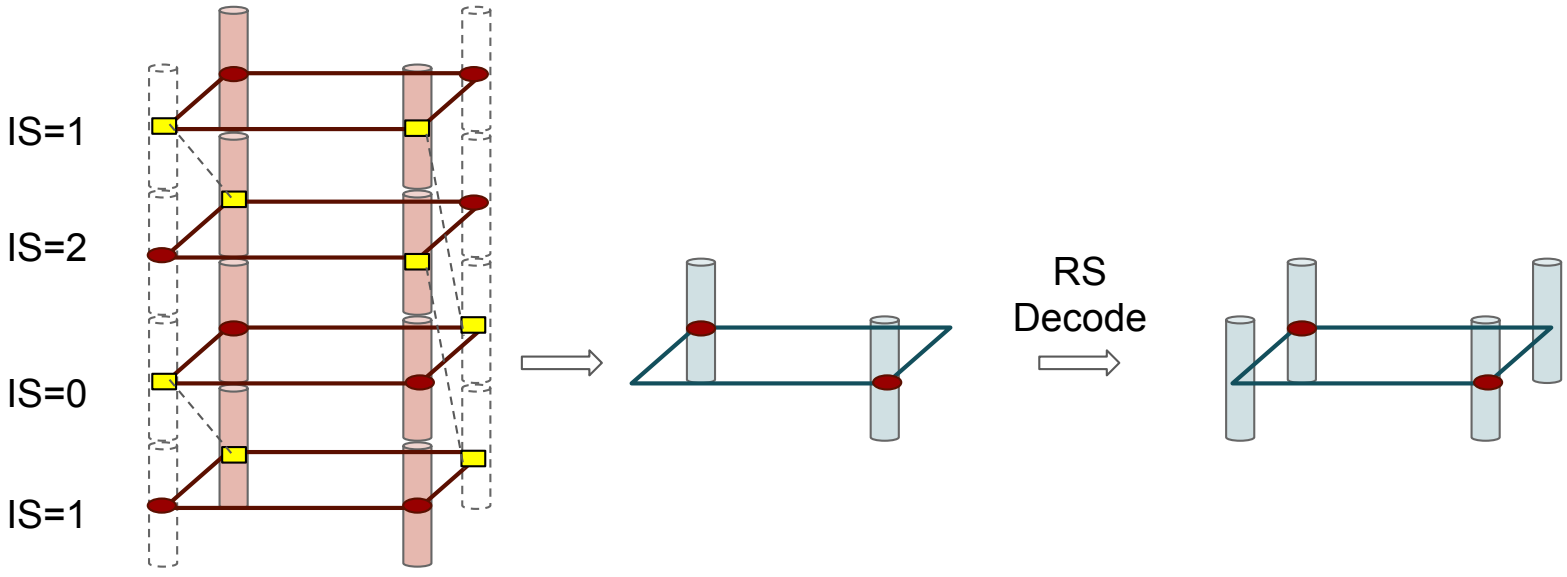
$z = (1,1)$

Intersection score is given by the number of hole-dot pairs

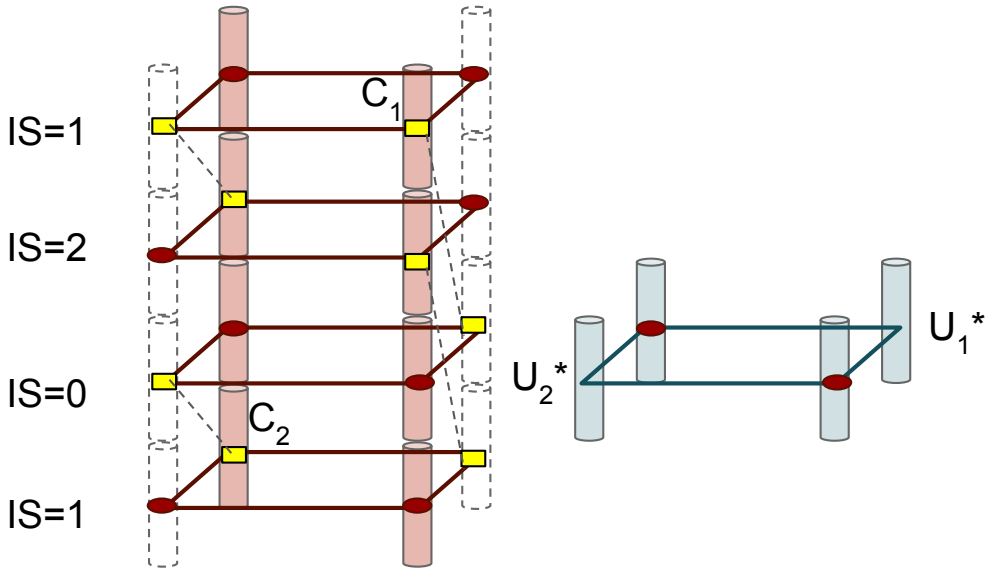
For non erased nodes, get the uncoupled sub-chunks for planes with IS=0



RS decode to get the remaining uncoupled-subchunks

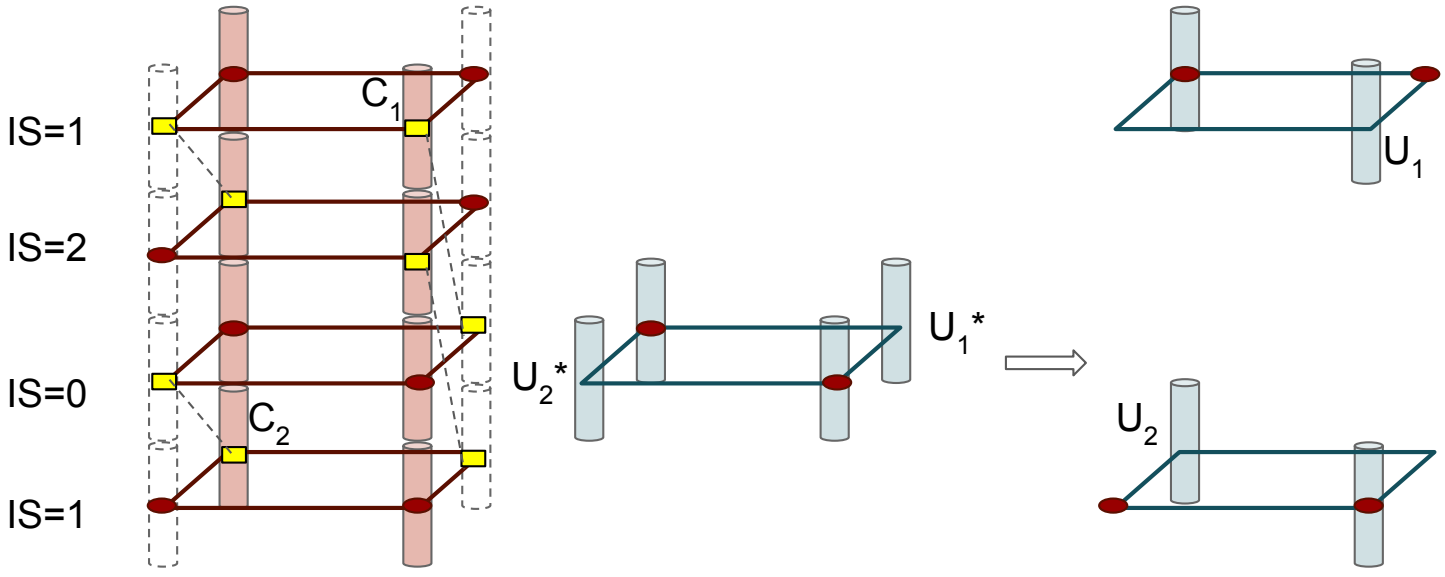


We now have following sub-chunks



Known sub-chunks

For non erased nodes, get the uncoupled sub-chunks for planes with IS=1

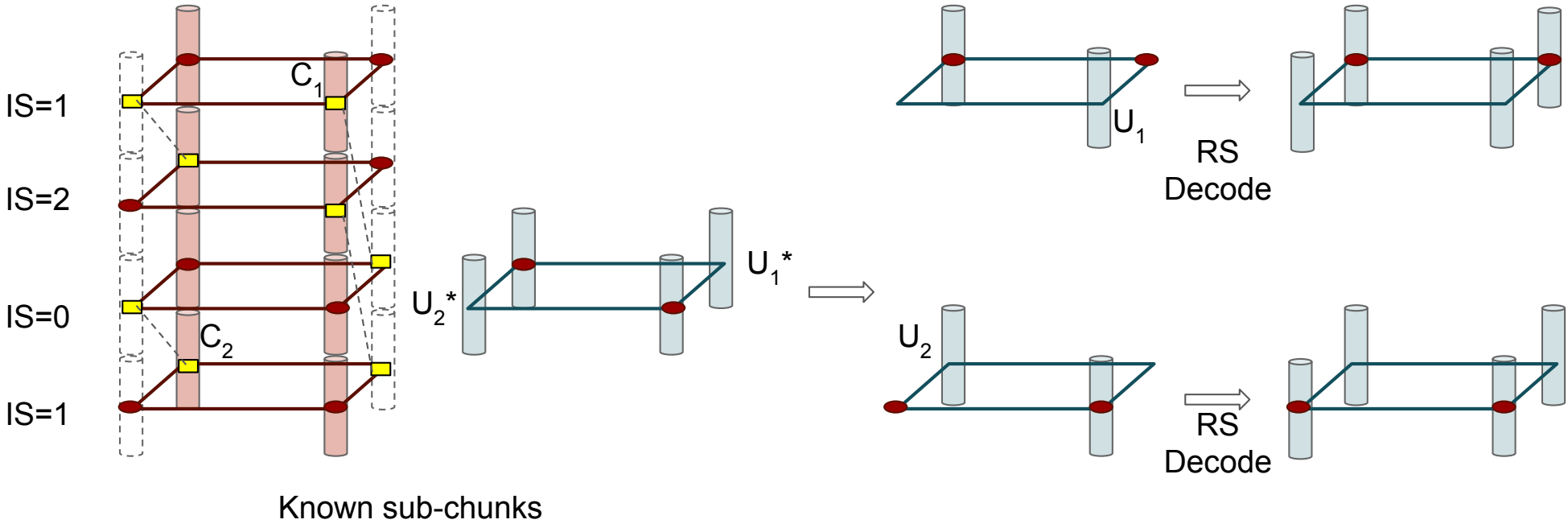


Known sub-chunks

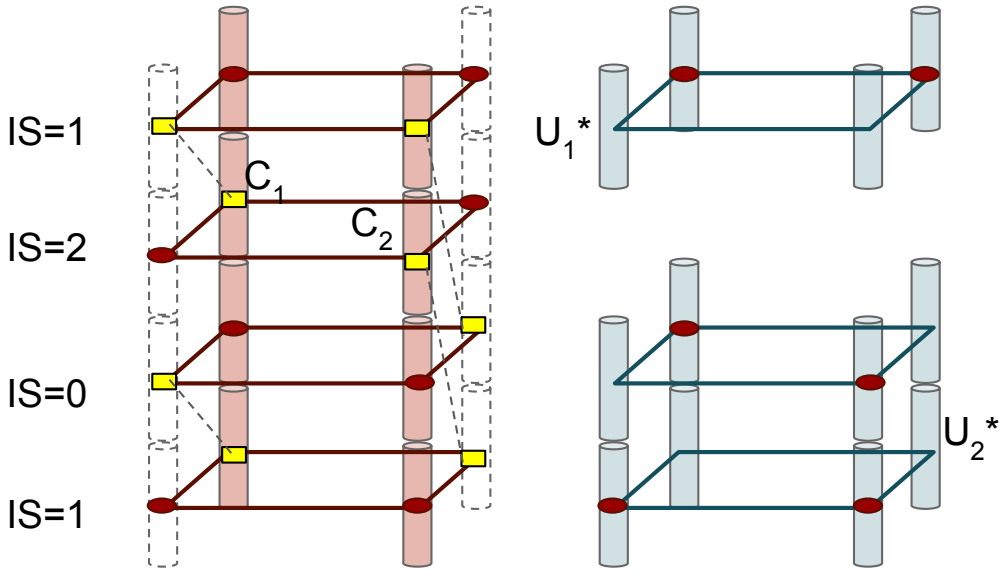
Get U_2 from U_2* and C_2

Get U_1 from U_1* and C_1

RS decode to get the remaining uncoupled-subchunks

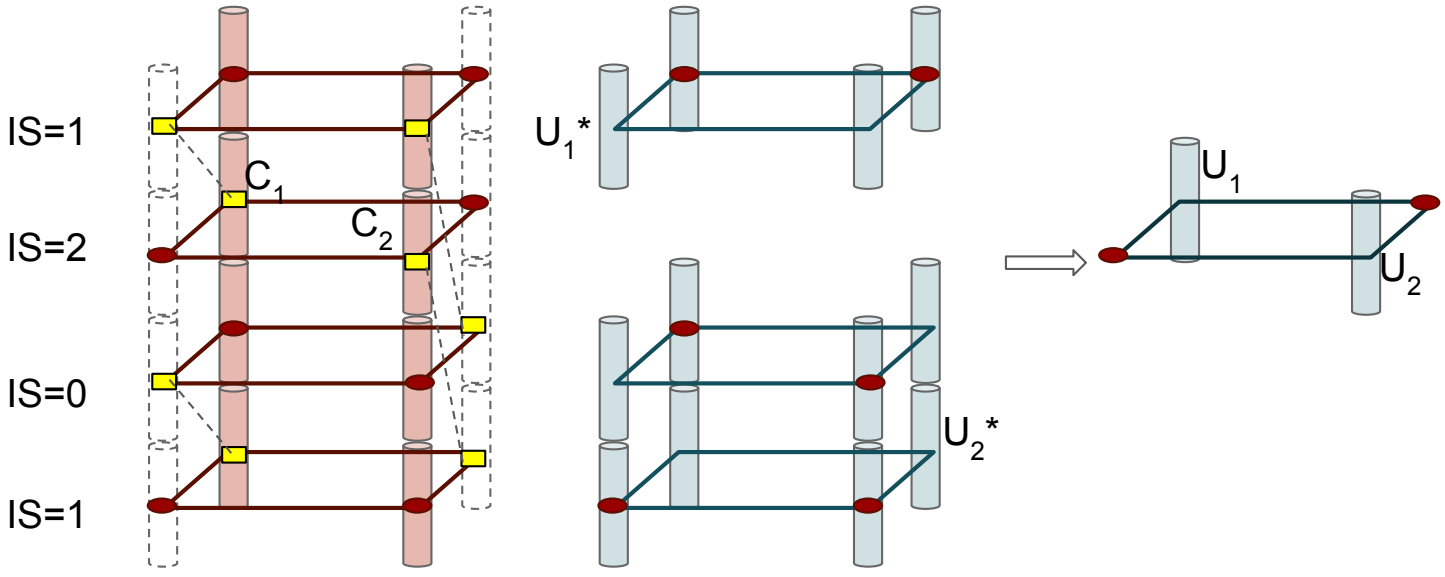


We now have the following sub-chunks



Known sub-chunks

For non erased nodes, get the uncoupled sub-chunks for planes with IS=2

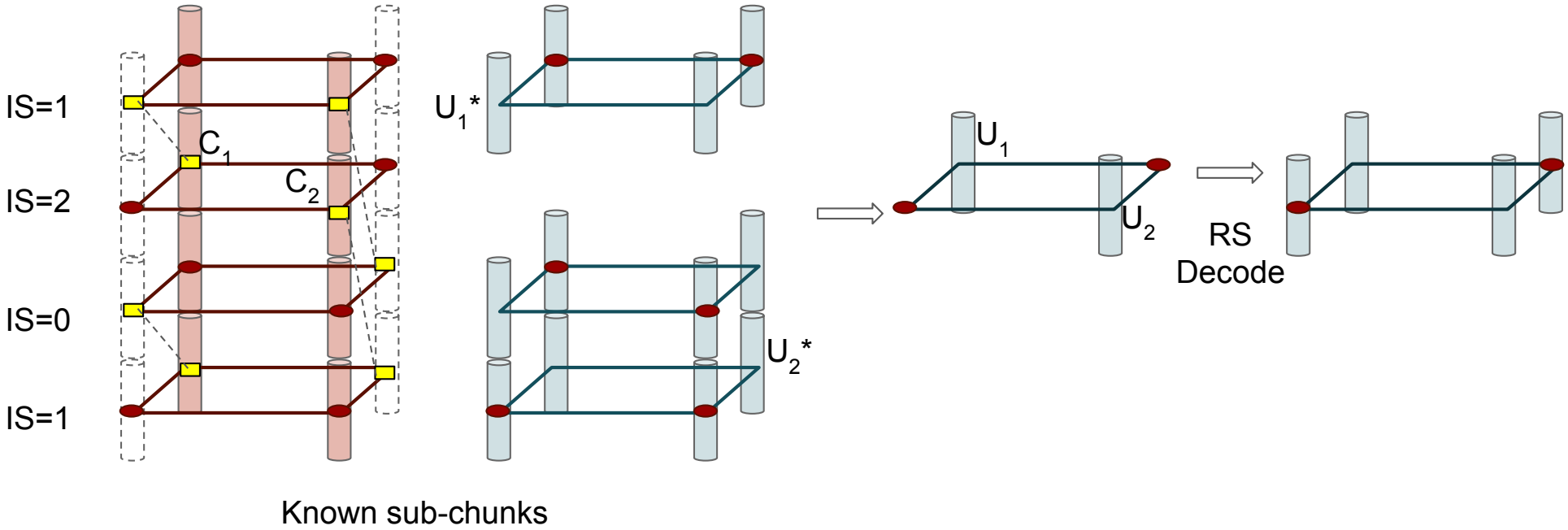


Known sub-chunks

Get U_2 from U_2* and C_2

Get U_1 from U_1* and C_1

Get the uncoupled sub-chunks for planes with IS=2

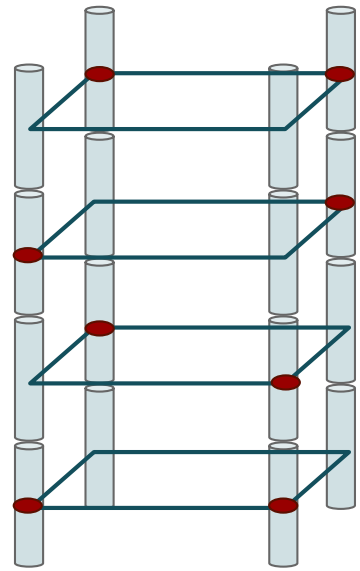


Known sub-chunks

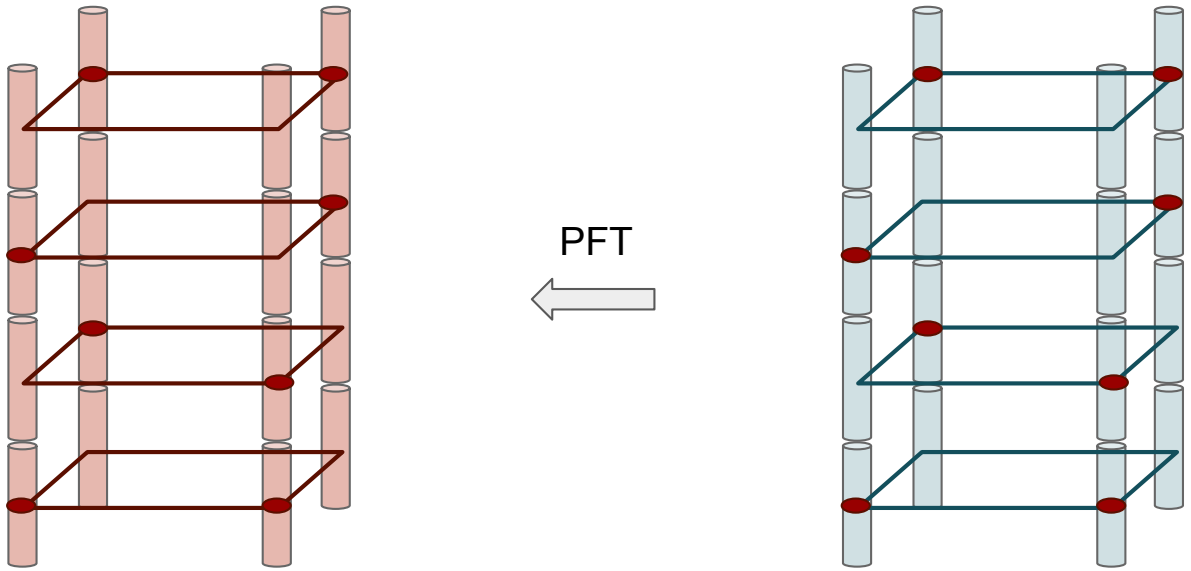
Get U_2 from U_2* and C_2

Get U_1 from U_1* and C_1

We now have all the uncoupled sub chunks



The coupled sub chunks can now be computed using PFT



The decoding is now complete