No Ghost Cells, 3 CPUs

```
Partitioning_Type all(3);
floatArray v(3, 3, 3);
v.partition(all);
```

getBase(…) = (0,0,0)
getStride(…) = (1,1,1)
getBound(…) = (2,2,2)
getLength(…) = (3,3,3)
getGhost(…) = (0,0,0)

K=0 slice

(0,2) = 2 (1,2) = 2 (2,2) = 2

(0,1) = 1 (1,1) = 1 (2,1) = 1

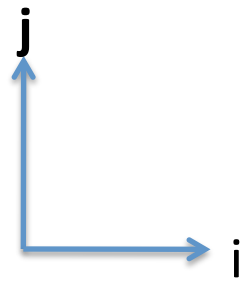
(0,0) = 0 (1,0) = 0 (2,0) = 0

(global, idx) = data ptr idx

CPU #0
getLocalBase(…) = (0,0,0)
getLocalStride(…) = (1,1,1)
getLocalBound(…) = (0,2,2)
getLocalLength(…) = (1,3,3)

CPU #1
getLocalBase(…) = (1,0,0)
getLocalStride(…) = (1,1,1)
getLocalBound(…) = (1,2,2)
getLocalLength(…) = (1,3,3)

CPU #2
getLocalBase(…) = (2,0,0)
getLocalStride(…) = (1,1,1)
getLocalBound(…) = (2,2,2)
getLocalLength(…) = (1,3,3)

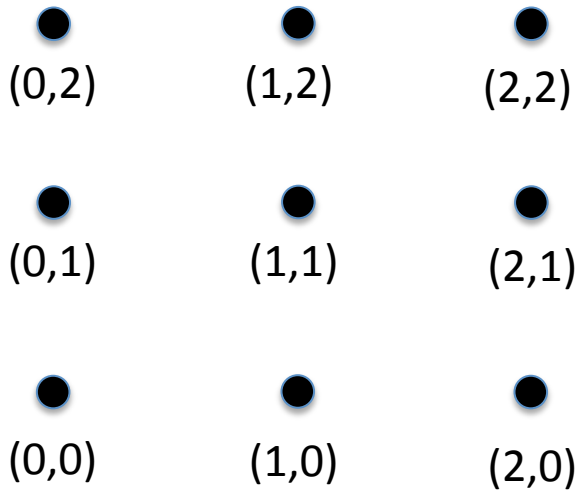


With 1^3 Ghost Cells, 3 CPUs

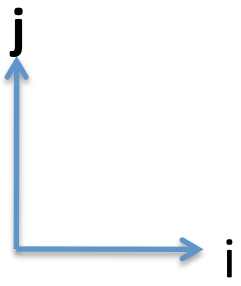
```
Partitioning_Type all(3);
all.partitionAlongAxis(0,TRUE,1);
all.partitionAlongAxis(1,TRUE,1);
all.partitionAlongAxis(2,TRUE,1);

floatArray v(3, 3, 3);
v.partition(all);
```

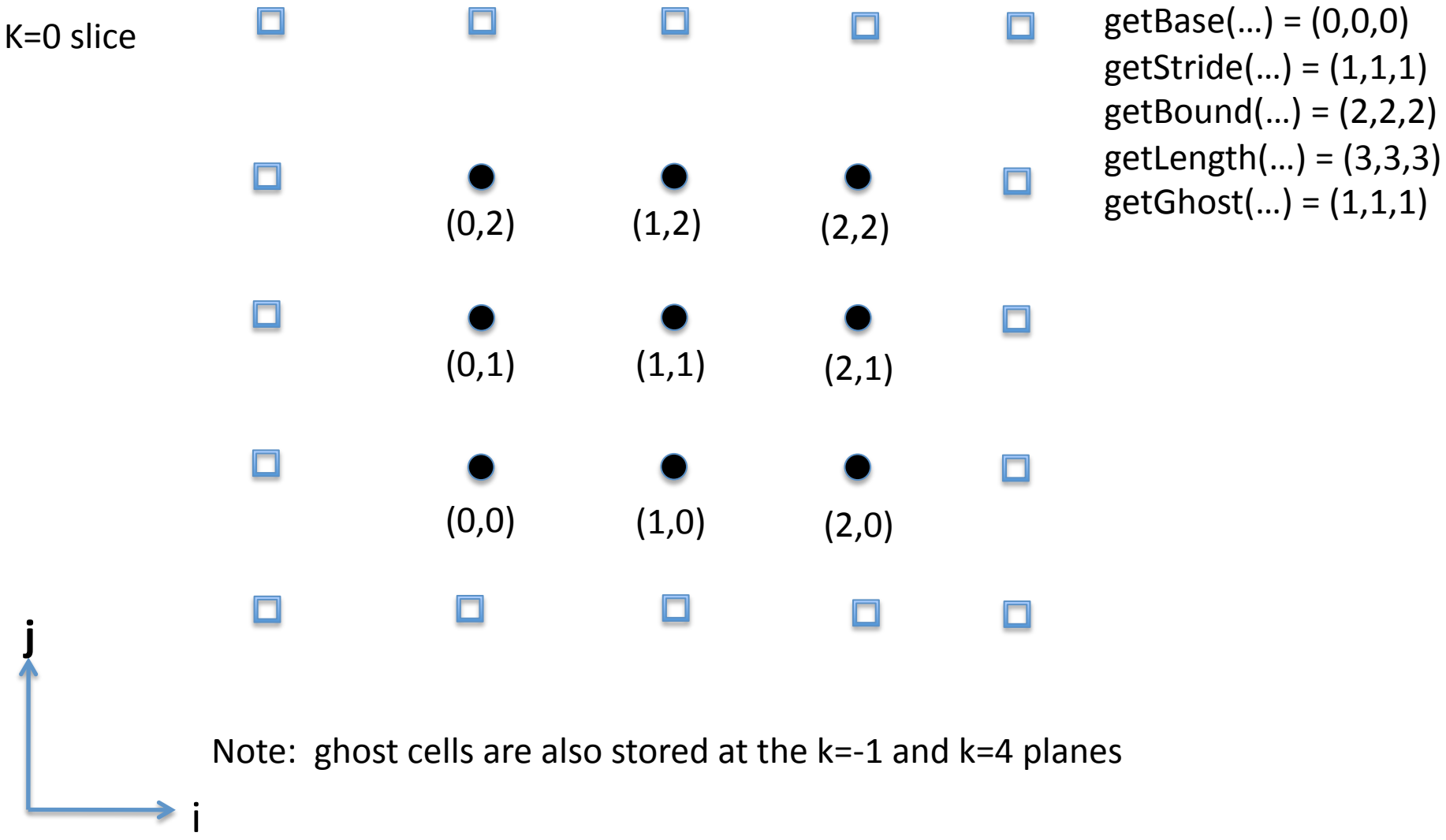
K=0 slice



getBase(…) = (0,0,0)
getStride(…) = (1,1,1)
getBound(…) = (2,2,2)
getLength(…) = (3,3,3)
getGhost(…) = (1,1,1)

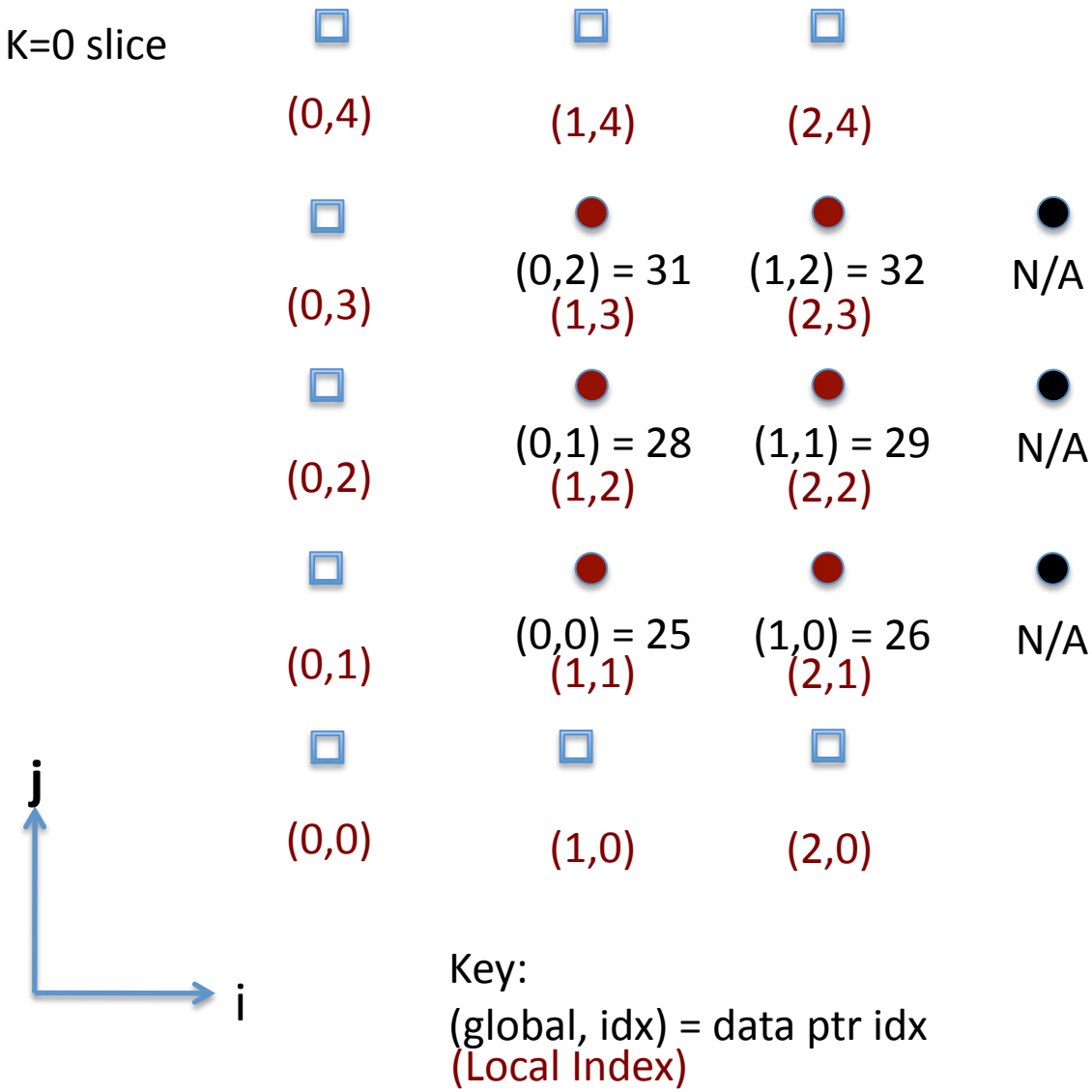


Ghost Cells, 3 CPUs



Note: ghost cells are also stored at the k=-1 and k=4 planes

Ghost Cells, 3 CPUs… as seen by CPU #0

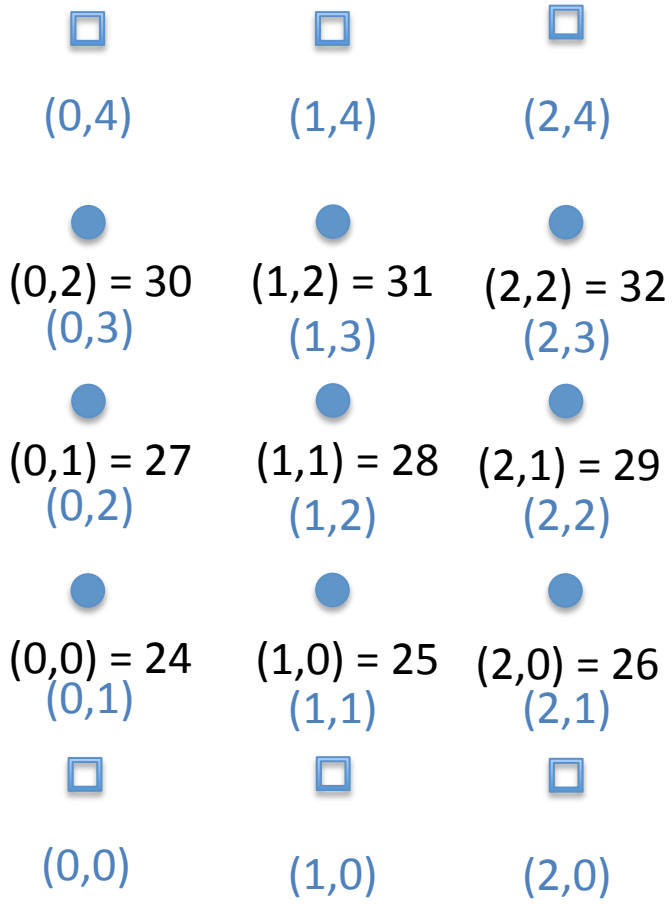


getBase(…) = (0,0,0)
getStride(…) = (1,1,1)
getBound(…) = (2,2,2)
getLength(…) = (3,3,3)

CPU #0
getLocalBase(…) = (0,0,0)
getLocalStride(…) = (1,1,1)
getLocalBound(…) = (1,2,2)
getLocalLength(…) = (2,3,3)

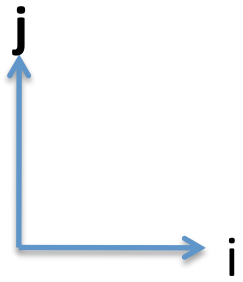
Ghost Cells, 3 CPUs … as seen by CPU #1

K=0 slice



getBase(…) = (0,0,0)
getStride(…) = (1,1,1)
getBound(…) = (2,2,2)
getLength(…) = (3,3,3)

CPU #1
getLocalBase(…) = (0,0,0)
getLocalStride(…) = (1,1,1)
getLocalBound(…) = (2,2,2)
getLocalLength(…) = (3,3,3)



Key:
(global, idx) = local data ptr idx
(Local Index)

Ghost Cells, 3 CPUs … as seen by CPU #2