These are times for a different implementation of the same algoritm. This imple
mentation uses a custom OpenMP reduction to keep track of the cell with the high
est average neighborhood value, and uses a HALO of 0s arount the data array to
cut down on the special cases that need to be handled (corners and edges of the
array). The total addition to the skeleton file for this implementation was abo
ut 50 lines of code (since I had to alter the array creation for the halo, and
define a class for the custom reducer).

For some help on custom reducers, see the following github. It does a custom S
UM, not a custom MAX, but it will get you started.

https://gist.github.com/eruffaldi/7180bdec4c8c9a11f019dd0ba9a2d68c

Around page 180 of the openMP manual, linked below, has all the details. be mi
ndful of your data types, and things fall into place.

http://www.openmp.org/mp-documents/OpenMP4.0.0.pdf

```
marmcke@vm-02$ ./a.out rand  1 10000 20000 0
largest average: 9461.78
found at cells: (618, 2726)
elapsed time: 1.06777
marmcke@vm-02$ ./a.out rand  2 10000 20000 0
largest average: 9461.78
found at cells: (618, 2726)
elapsed time: 0.536047
marmcke@vm-02$ ./a.out rand  4 10000 20000 0
largest average: 9461.78
found at cells: (618, 2726)
elapsed time: 0.476804
marmcke@vm-02$ ./a.out rand  1 20000 20000 0
largest average: 9531.33
found at cells: (17841, 4906)
elapsed time: 2.1195
marmcke@vm-02$ ./a.out rand  2 20000 20000 0
largest average: 9531.33
found at cells: (17841, 4906)
elapsed time: 1.09258
marmcke@vm-02$ ./a.out rand  4 20000 20000 0
largest average: 9531.33
found at cells: (17841, 4906)
elapsed time: 0.935394
```