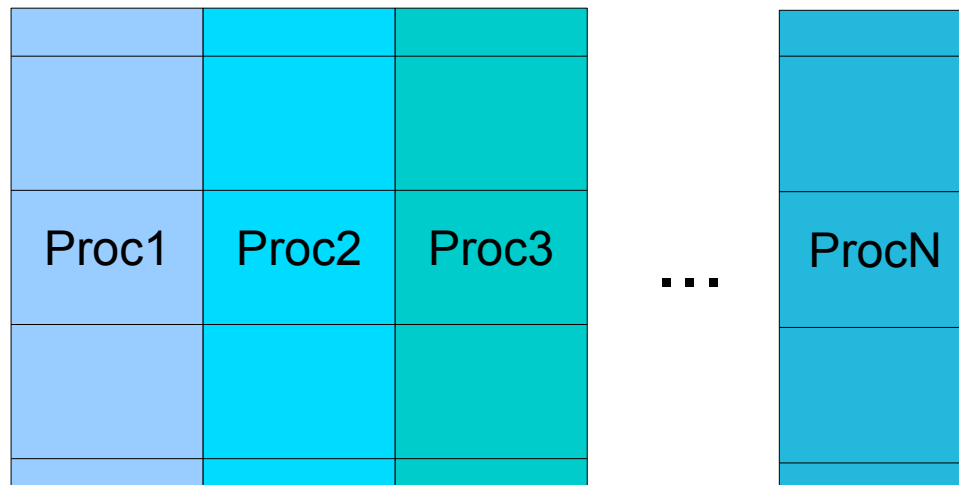


# DataHolder Component

Rob Armstrong

# Two Dimensional Strip-Decomposed Grid

- Pretty lame right now but it works.
  - Takes a dependency radius (stencil) to compute overlap for updating
  - TwoD strips by processor



# DataHolder Exports a “BlockPort”

```
/* ===== */

/* extra stuff that would have been handled by the ParameterPorts
previously */

/** The number of participating processors */
int getNumProcs();

/** The rank of this processor */
int getMyProc();

/** Set the number of rows for the whole problem */
void setRows(in int rows);

/** Set the number of columns for the whole problem */
void setCols(in int cols);

/* ===== */
```

```
/* The following is more or less the BlockPort implementation. */

void compute();
void update();
void setOverlapUniform(in int radius,
                       out opaque ccaBlock /* CCA_Block **ccaBlock */
                       );
void getBlock(out opaque ccaBlock /* CCA_Block **ccaBlock */);
void releaseBlock(out opaque ccaBlock /* CCA_Block **ccaBlock */);
```

Opaque becomes a void\* in C/C++: void\* ccaBlock  
realCCABlock = (CCA\_Block\*) ccaBlock

# What is important to the testbed?

- What else is needed?
  - More general partitioning scheme (i.e. A component)
  - More general datatypes (?) than just doubles.
  - Global operations (all reduce, etc.)