

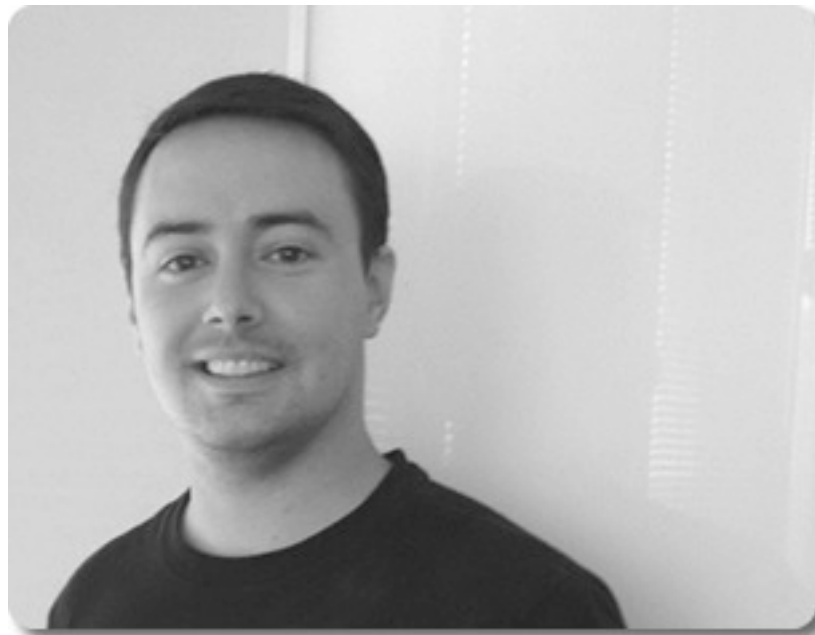
# DistArray:

## Distributed Arrays for Python

Robert Grant, Enthought  
[rgrant@enthought.com](mailto:rgrant@enthought.com)

11 June 2014

# Me



Robert Grant

# DistArray Team



Kurt Smith



Robert Grant



Blake Griffith



Mark Kness



Brian Granger

# Audience Poll

- Who has used NumPy?
- Who has used IPython Parallel?
- Who has written MPI code?
  - with mpi4py?

# DistArray

- Distributed NumPy-like arrays
- Built for data-parallel problems
- BSD Licensed
- Python 2.7 and 3.3 compatible
- Just released 0.3
  - not yet production ready

# Built on widely-used libraries

- NumPy
- IPython Parallel
- mpi4py
- h5py (optionally)

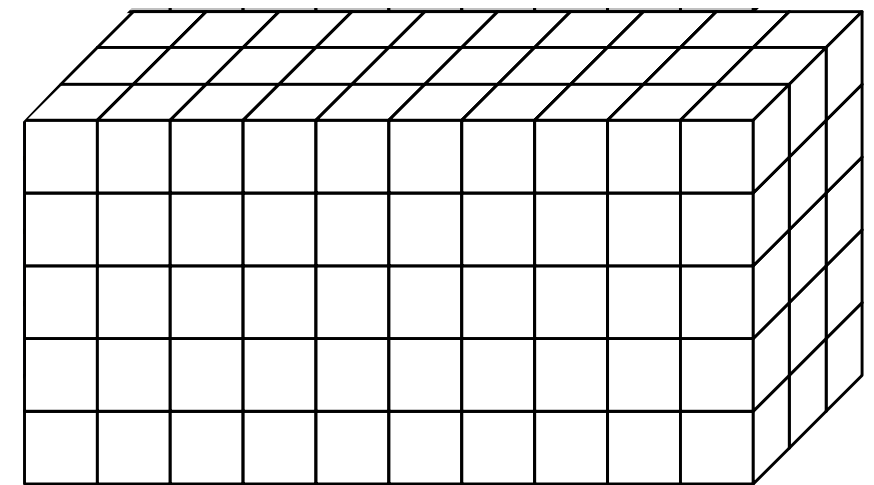
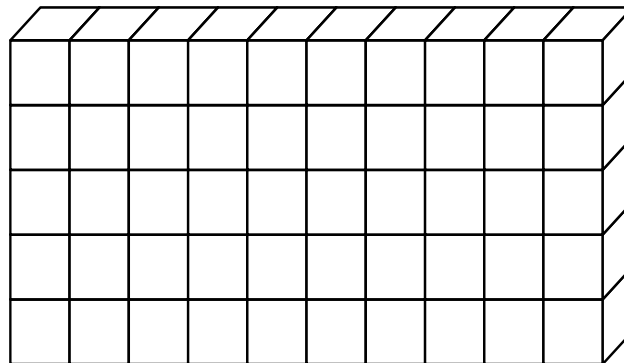
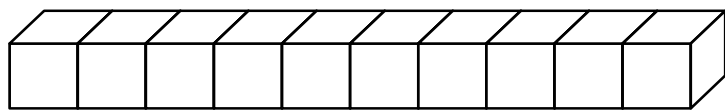


# Built to interface with existing distributed scientific libraries

- Trilinos / PyTrilinos
- Global Arrays / GAI
- PETSc / petsc4py
- Elemental



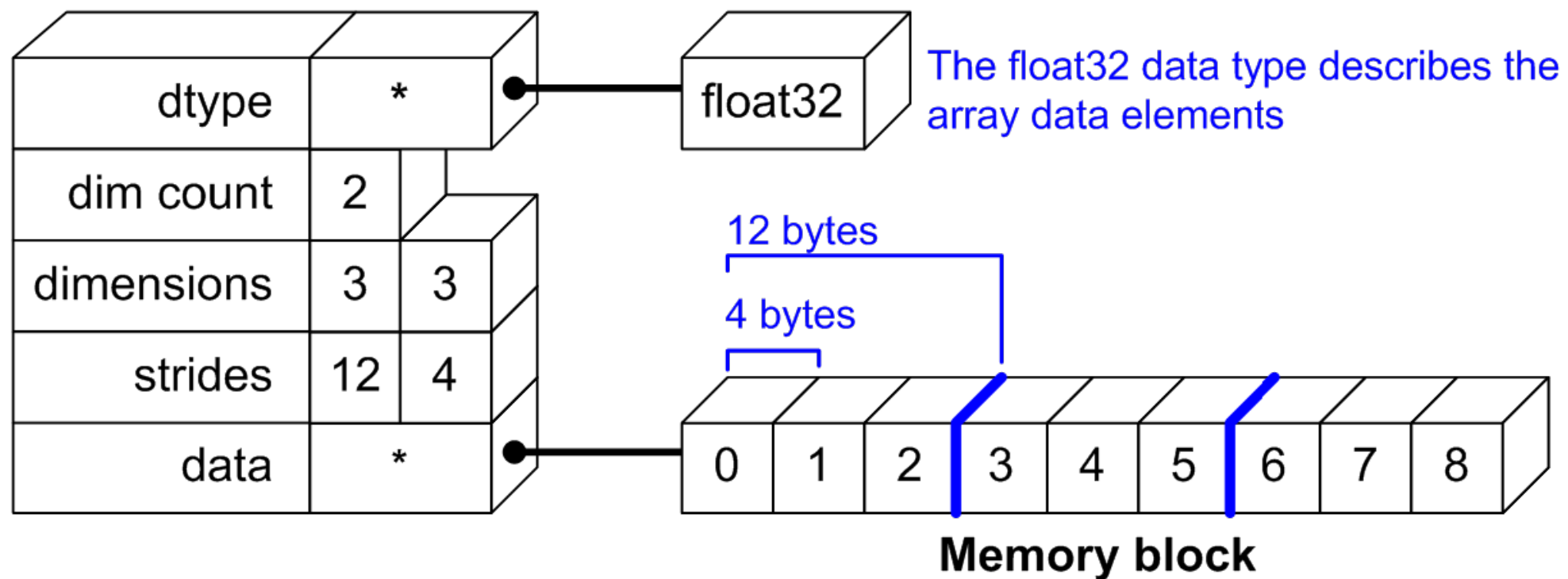
# NumPy Arrays



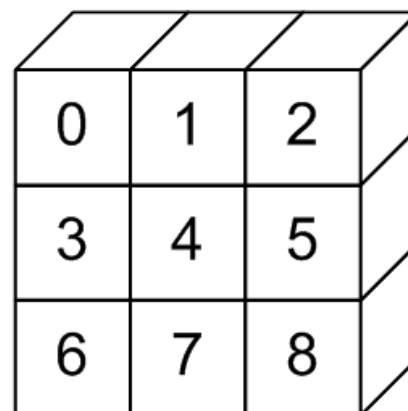


# NumPy Arrays

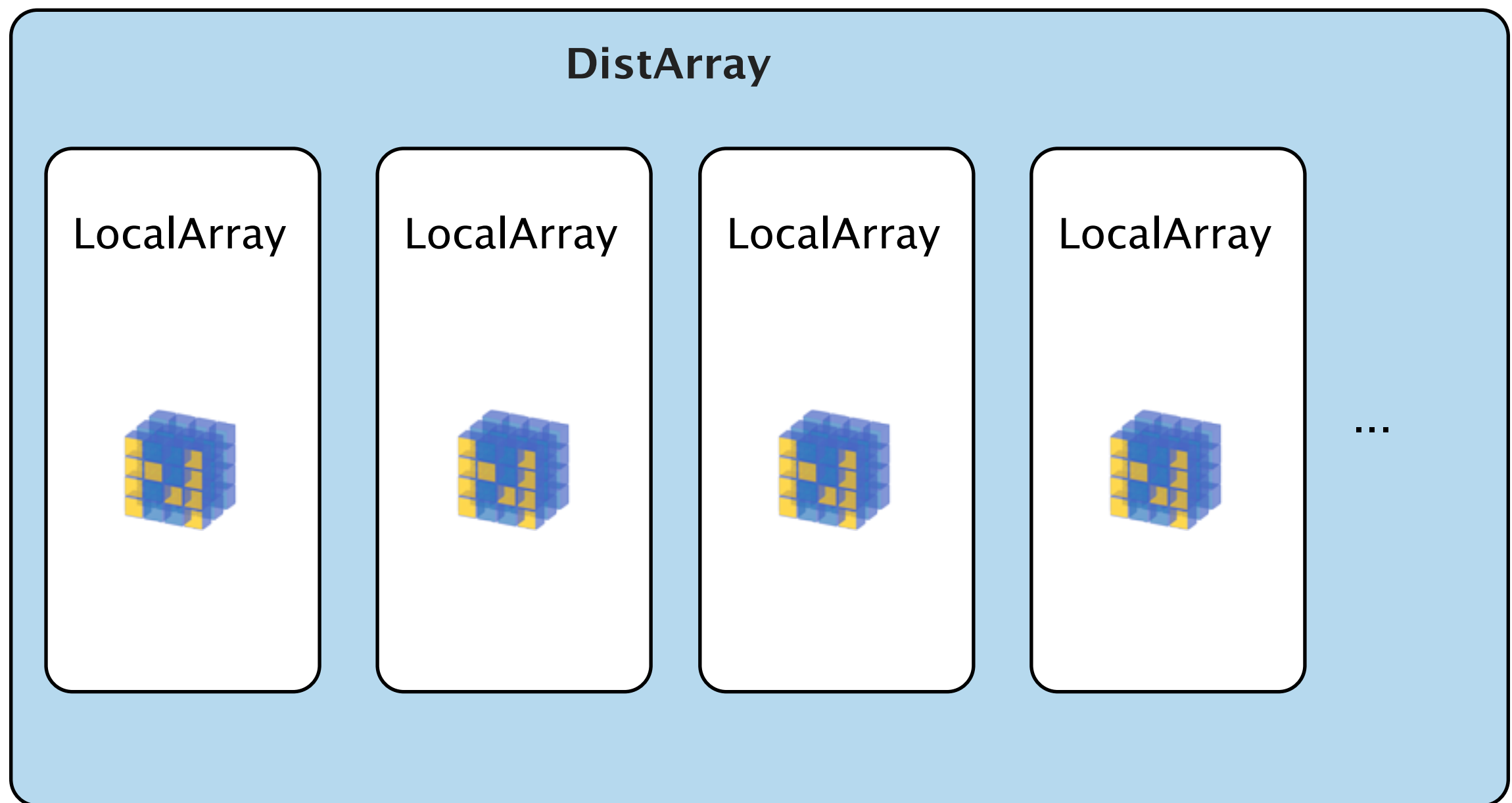
## NDArray Data Structure



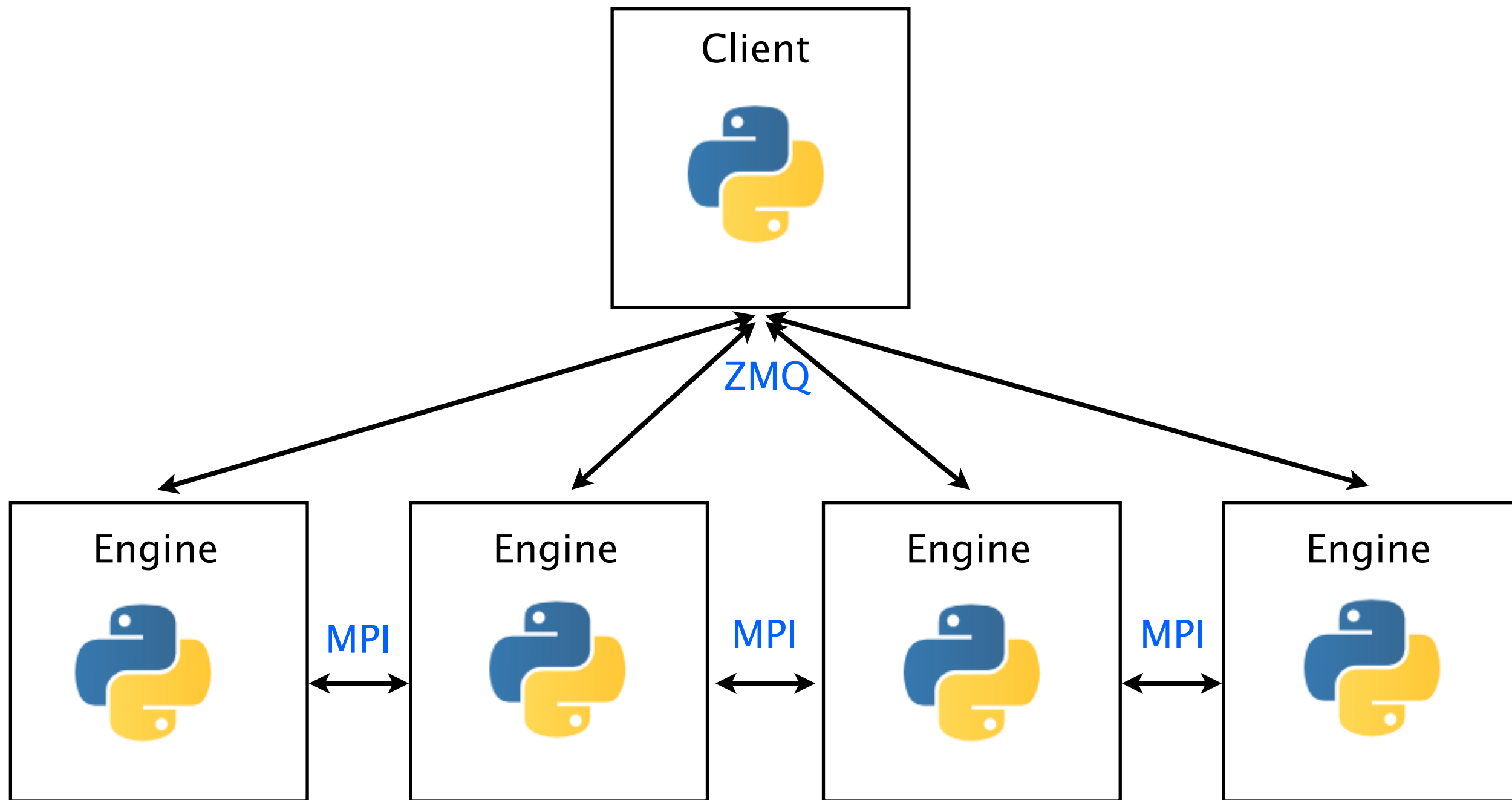
## Python View:



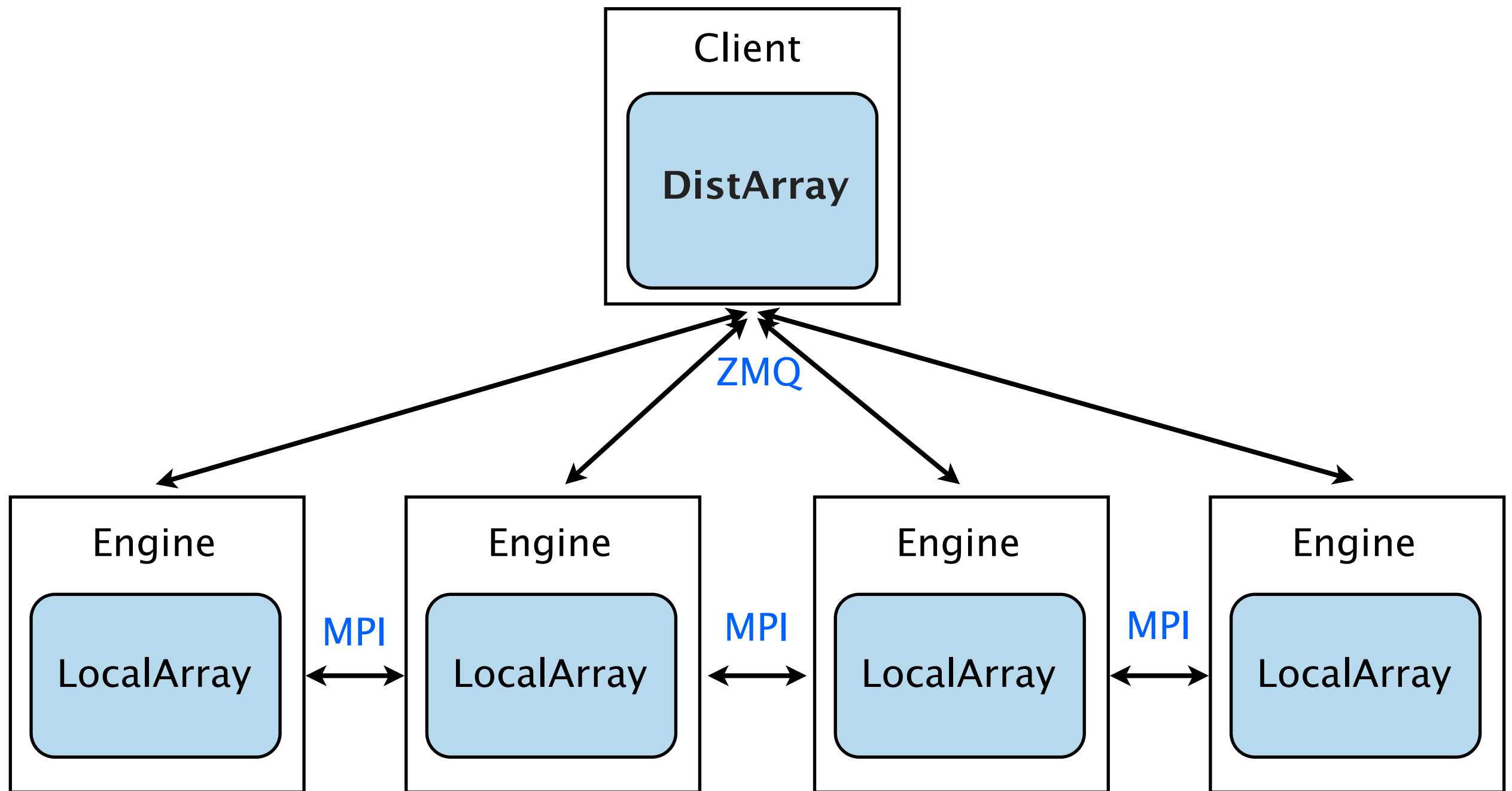
# DistArrays



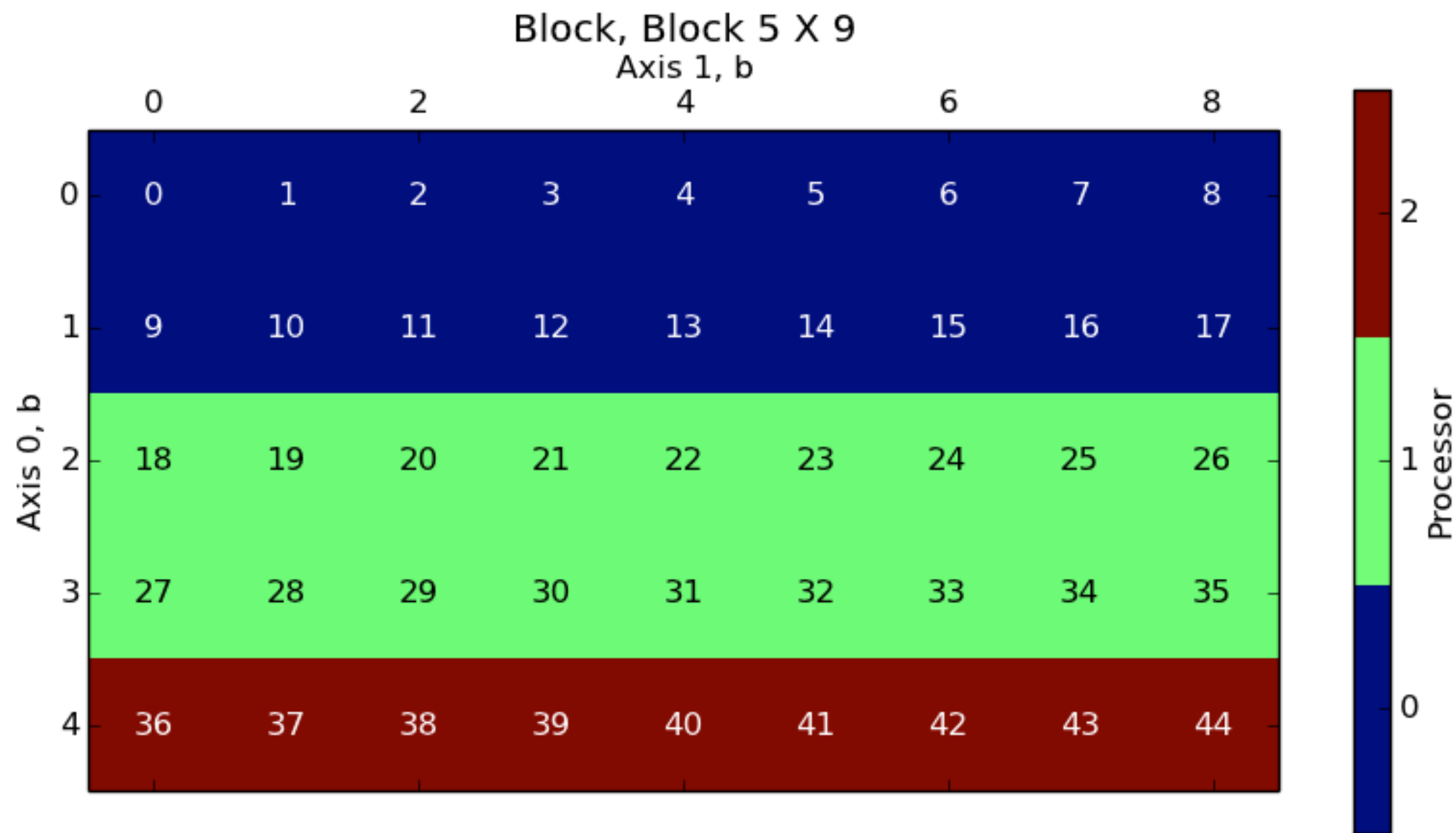
# IPython Parallel



# DistArrays



# Distributions – Block



# Distributions – Block

Local Arrays

Process (0, 0)

0	1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16	17

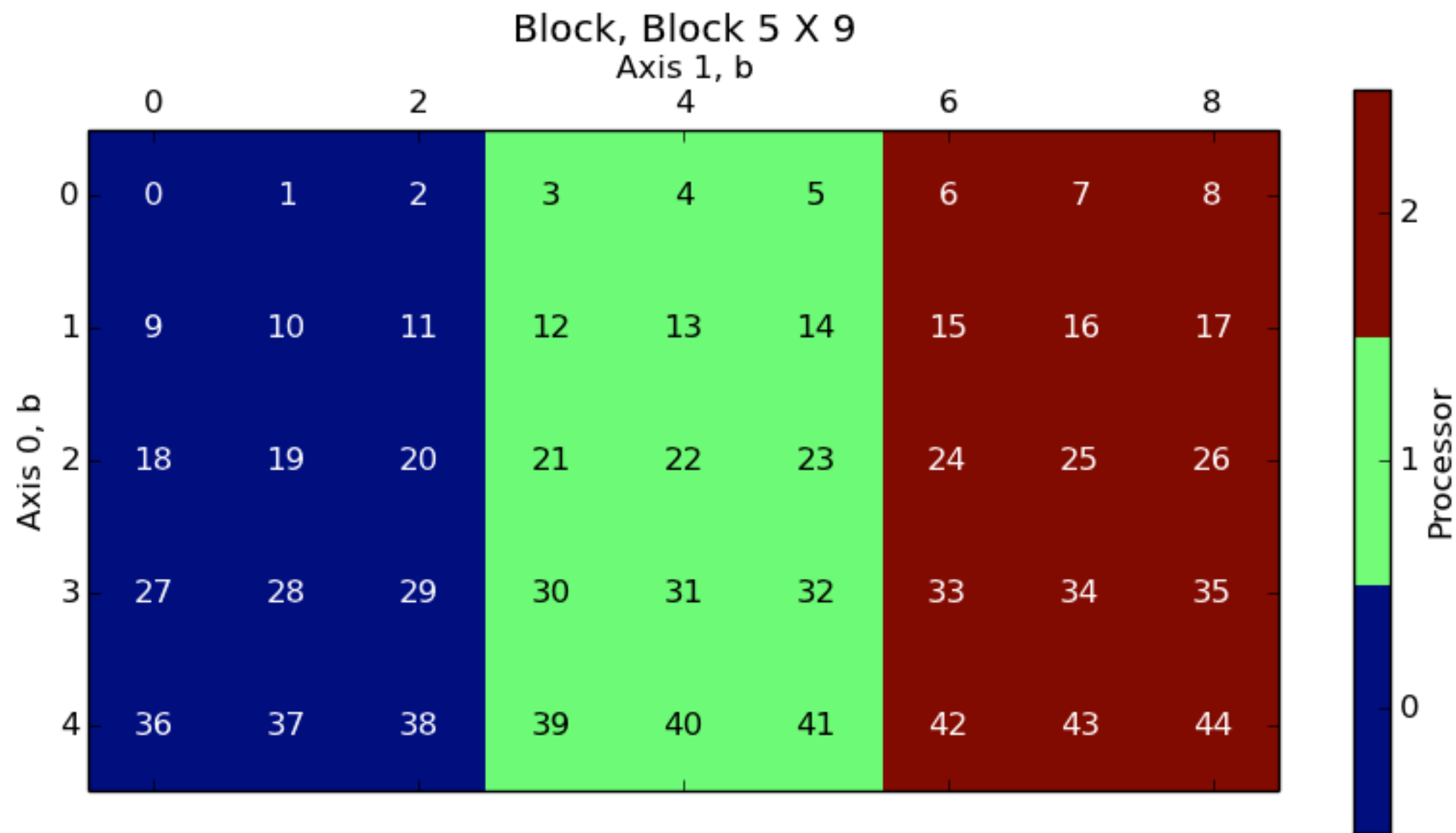
Process (1, 0)

18	19	20	21	22	23	24	25	26
27	28	29	30	31	32	33	34	35

Process (2, 0)

36	37	38	39	40	41	42	43	44
----	----	----	----	----	----	----	----	----

# Distributions – Block



# Distributions – Block

Local Arrays

Process (0, 0)

0	1	2
9	10	11
18	19	20
27	28	29
36	37	38

Process (0, 1)

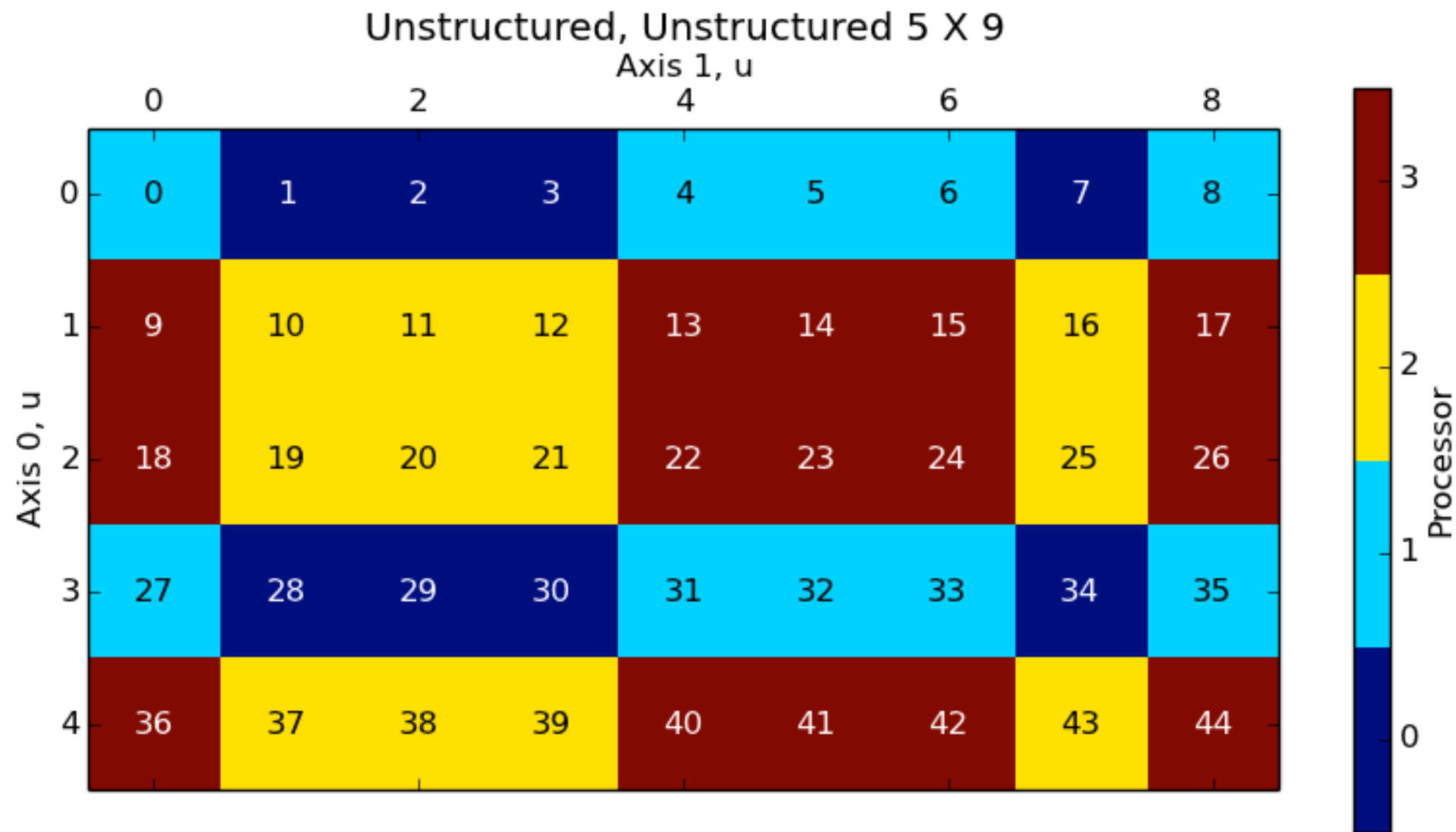
3	4	5
12	13	14
21	22	23
30	31	32
39	40	41

Process (0, 2)

6	7	8
15	16	17
24	25	26
33	34	35
42	43	44



# Distributions – Unstructured



# Distributions – Unstructured

Local Arrays

Process (0, 0)

29	30	34	28
2	3	7	1

Process (0, 1)

33	32	35	27	31
6	5	8	0	4

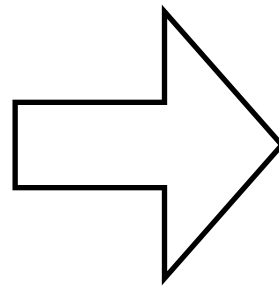
Process (1, 0)

38	39	43	37
20	21	25	19
11	12	16	10

Process (1, 1)

42	41	44	36	40
24	23	26	18	22
15	14	17	9	13

# Demo Notebook



# Roadmap (beyond 0.4)

- Redistribution methods
- Integration with Trilinos and other packages that subscribe to the Distributed Array Protocol
- Distributed broadcasting
- Distributed fancy indexing
- MPI-only communication for non-interactive deployment on clusters and supercomputers
- Lazy evaluation and deferred computation for latency hiding
- Out-of-core computations
- Extensive examples, tutorials, documentation
- Support for distributed sorting and other non-trivial distributed algorithms
- End-user control over communication and temporary array creation, and other performance aspects of distributed computation

# Questions?

- [github.com/enthought/distarray](https://github.com/enthought/distarray)
- [distarray.readthedocs.org](https://distarray.readthedocs.org)
- [distributed-array-protocol.readthedocs.org](https://distributed-array-protocol.readthedocs.org)
- Robert Grant – [rgrant@enthought.com](mailto:rgrant@enthought.com)

# Acknowledgement and Disclaimer

This material is based upon work supported by the Department of Energy under Award Number DE-SC0007699.

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.