Distributed Arrays for Python

Robert Grant, Enthought

rgrant@enthought.com

8 July 2015

DistArray

- Distributed NumPy-like arrays
- Built for data-parallel problems
- BSD Licensed
- Tested on Python 2.7, 3.3, 3.4
- Latest version: 0.5
  - ready for some early testers!

DistArray Contributors

Kurt Smith

Robert Grant

Blake Griffith

Mark Kness

Brian Granger

Built on widely-used libraries

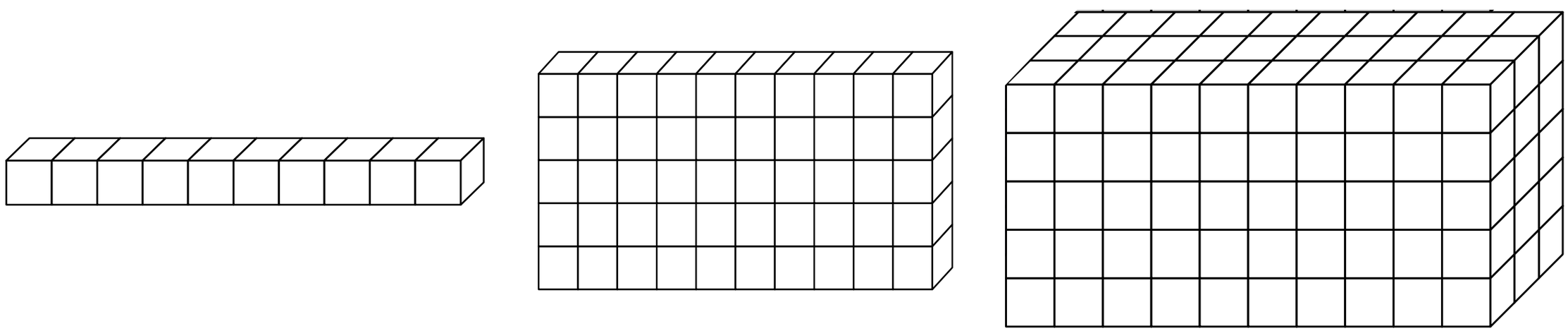
- NumPy
- IPython Parallel
- mpi4py

- h5py (optionally)

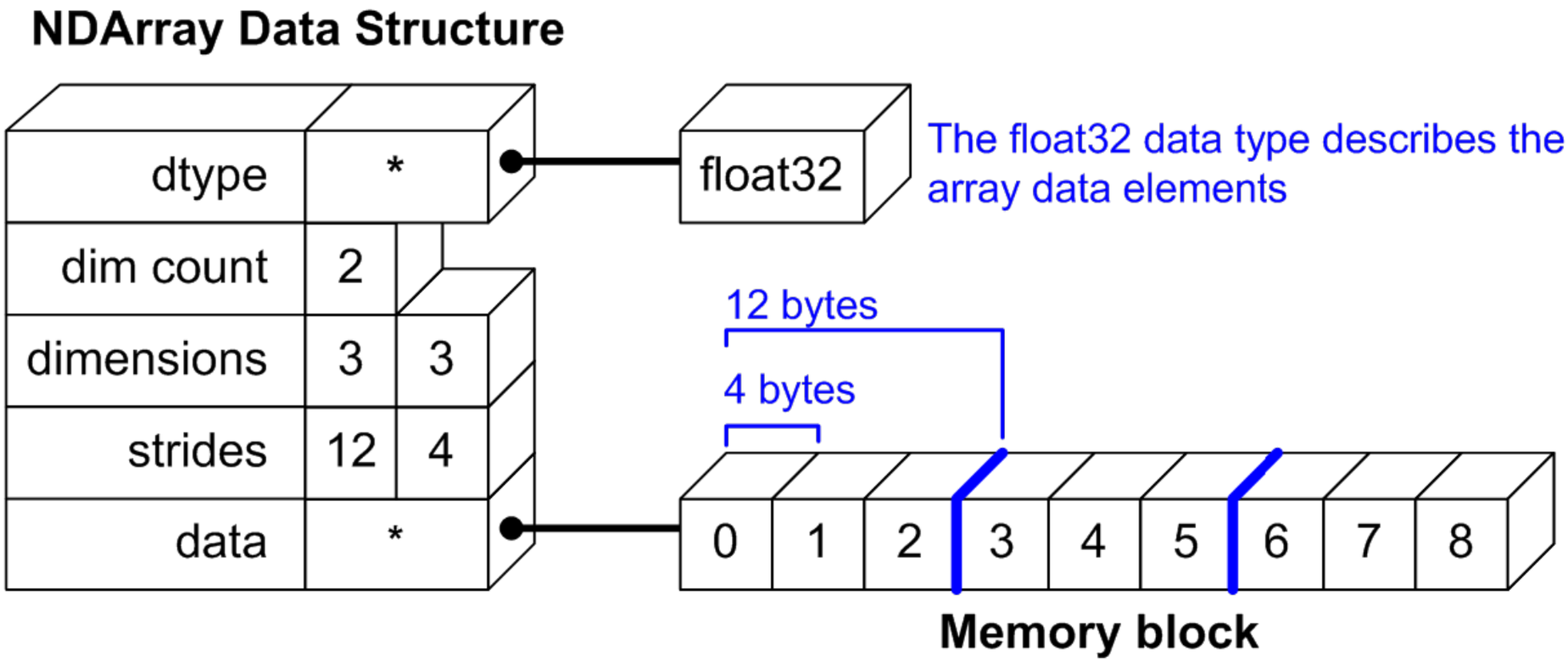
Built to interface with existing distributed scientific libraries

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

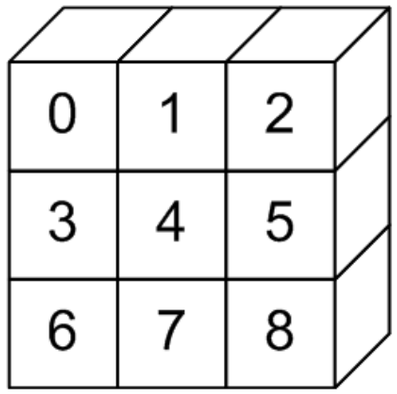
NumPy Arrays



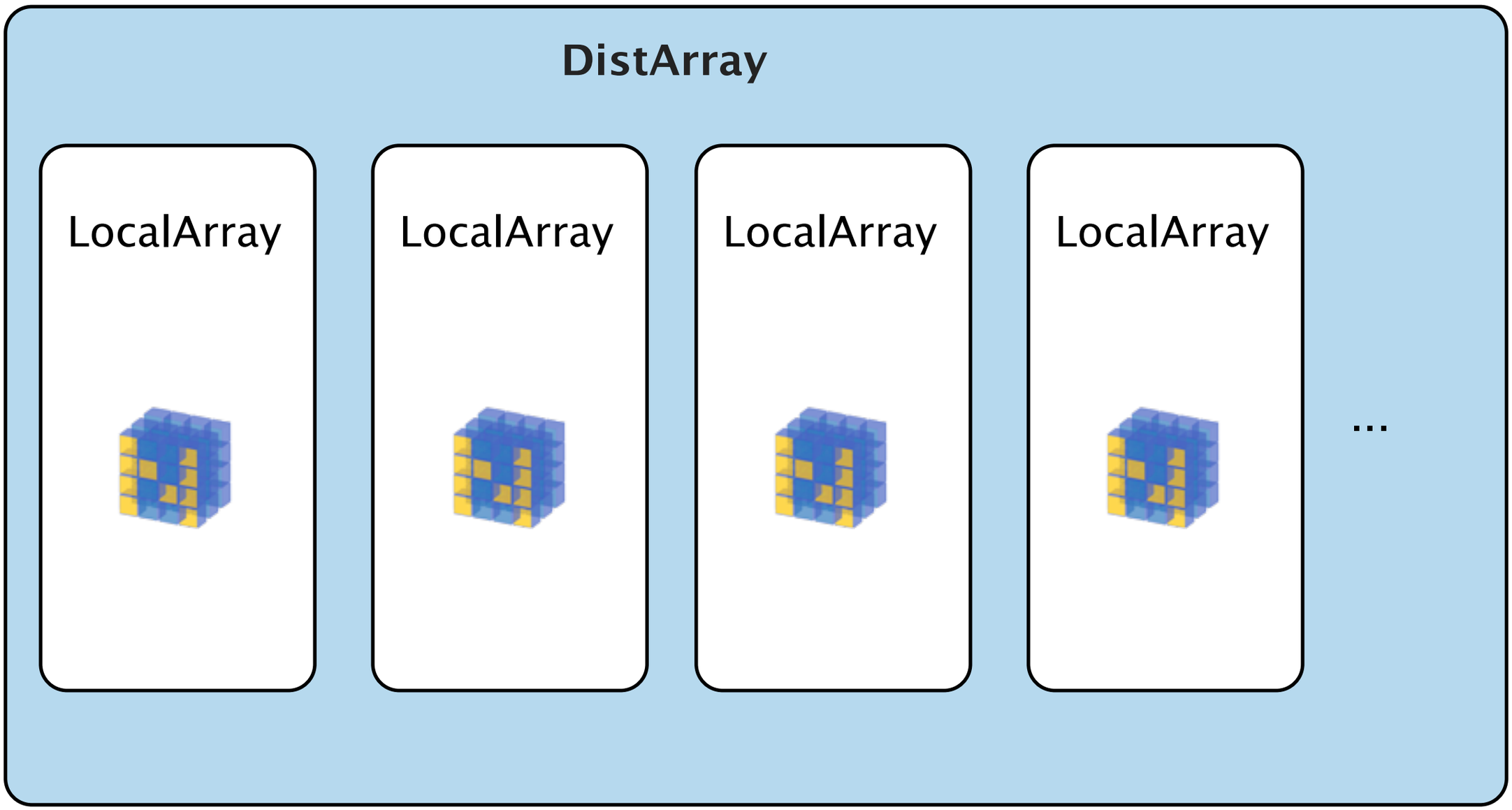
NumPy Arrays



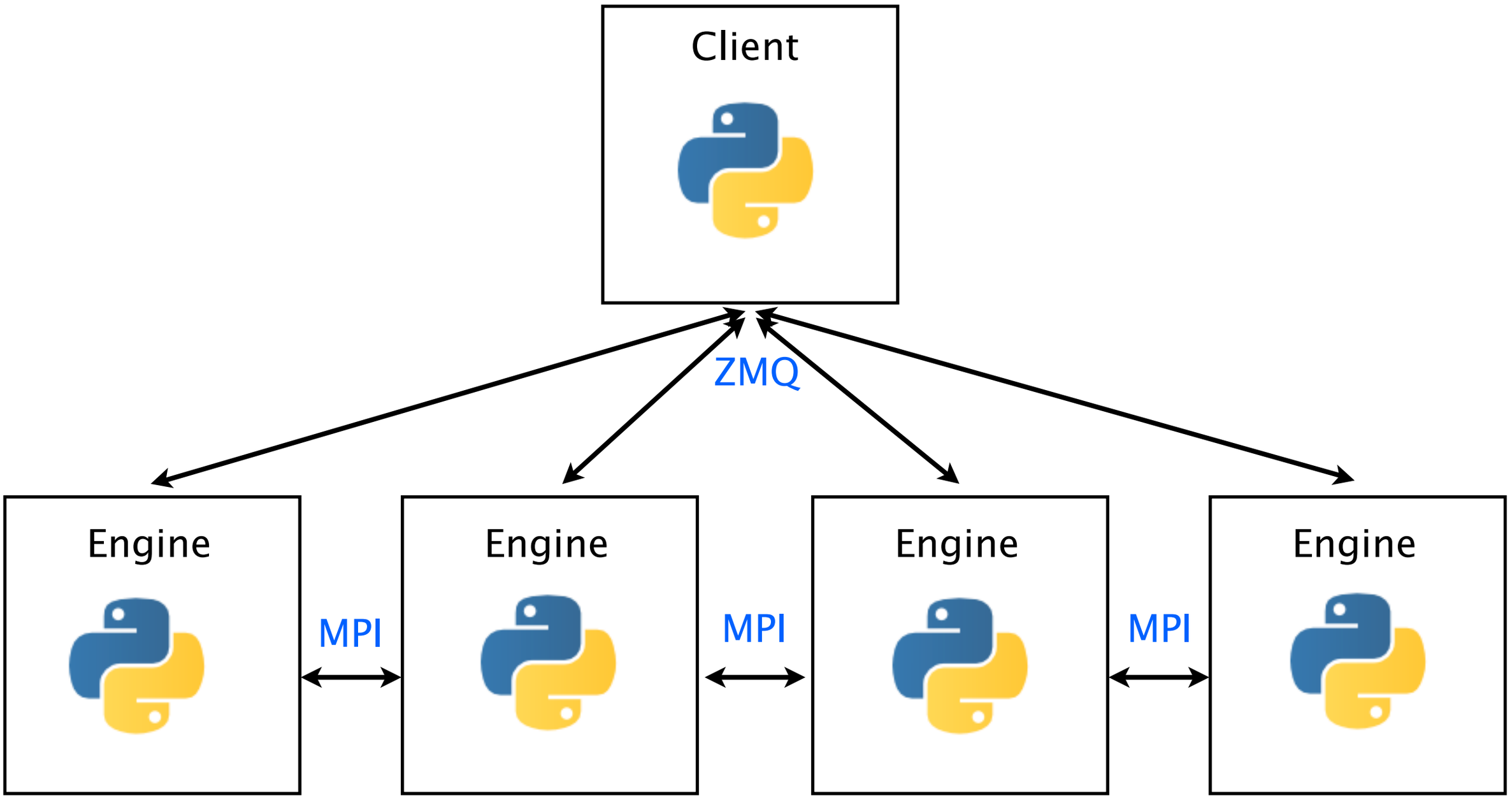
Python View:



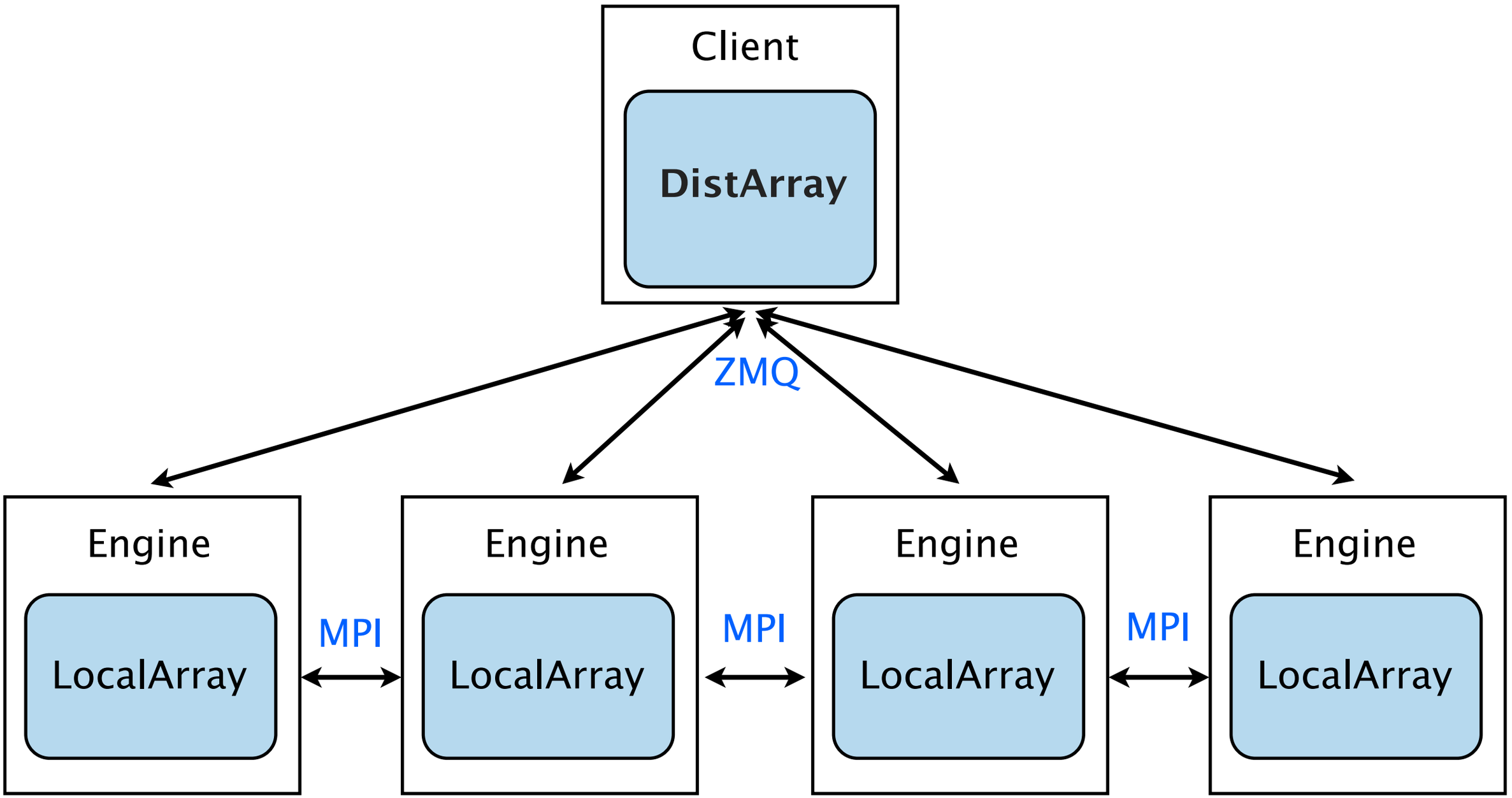
DistArrays



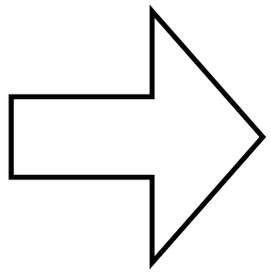
IPython Parallel



DistArrays


Client
DistArray
ZMQ
Engine
LocalArray
MPI
Engine
LocalArray
MPI
Engine
LocalArray
MPI
Engine
LocalArray

Demo Notebook



Roadmap (beyond 0.5)

Near-term features and improvements include:

- array re-distribution capabilities;
- lazy evaluation and deferred computation for latency hiding;
- interoperation with Trilinos; and
- distributed broadcasting support.

The longer-term roadmap includes:

- integration with other packages that subscribe to the Distributed Array protocol;
- distributed fancy indexing;
- out-of-core computations;
- support for distributed sorting and other non-trivial distributed algorithms; and
- end-user control over communication and temporary array creation, and other performance aspects of distributed computations.

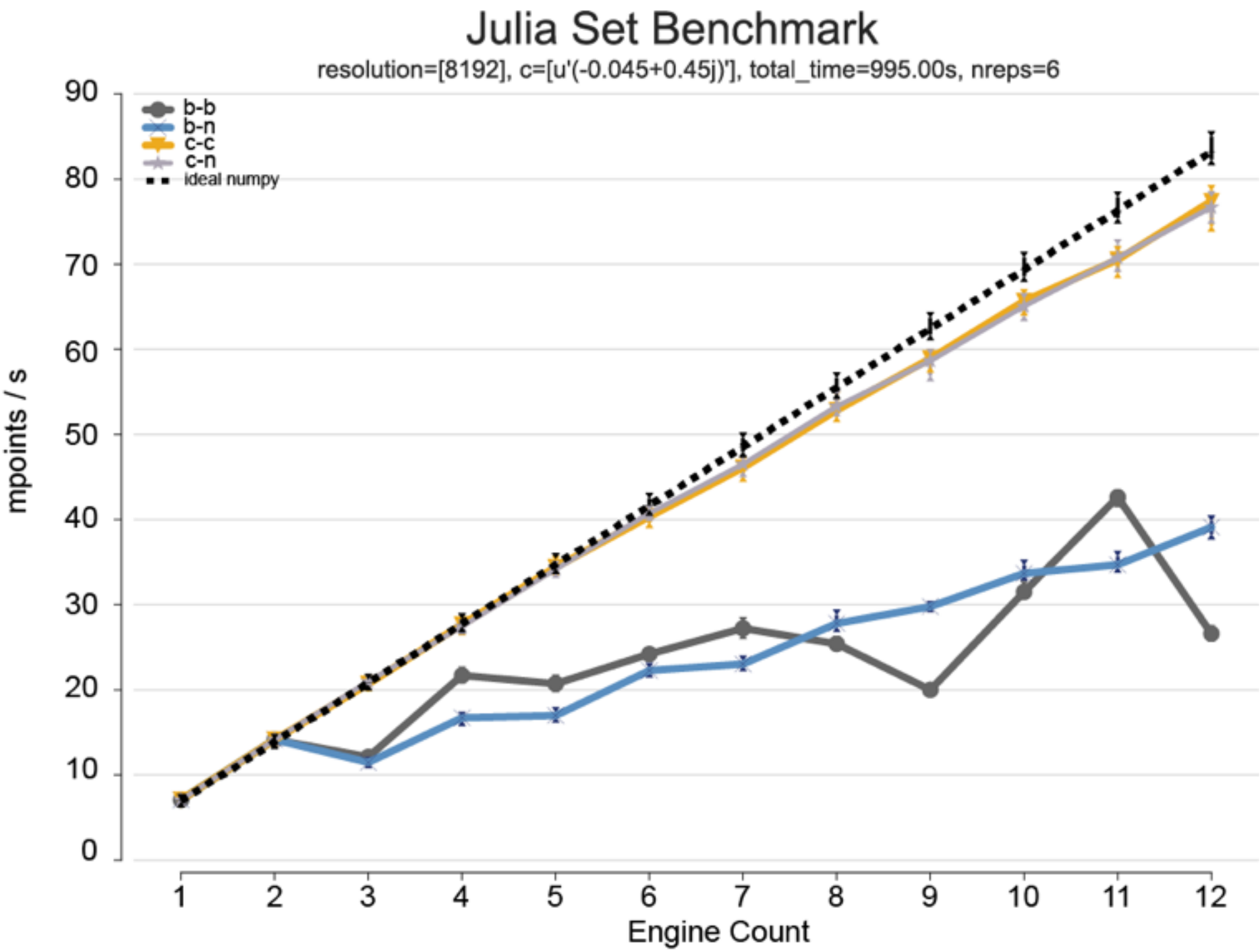
Questions?

- github.com/enthought/distarray
- distarray.readthedocs.org
- distributed-array-protocol.readthedocs.org
- Robert Grant – rgrant@enthought.com

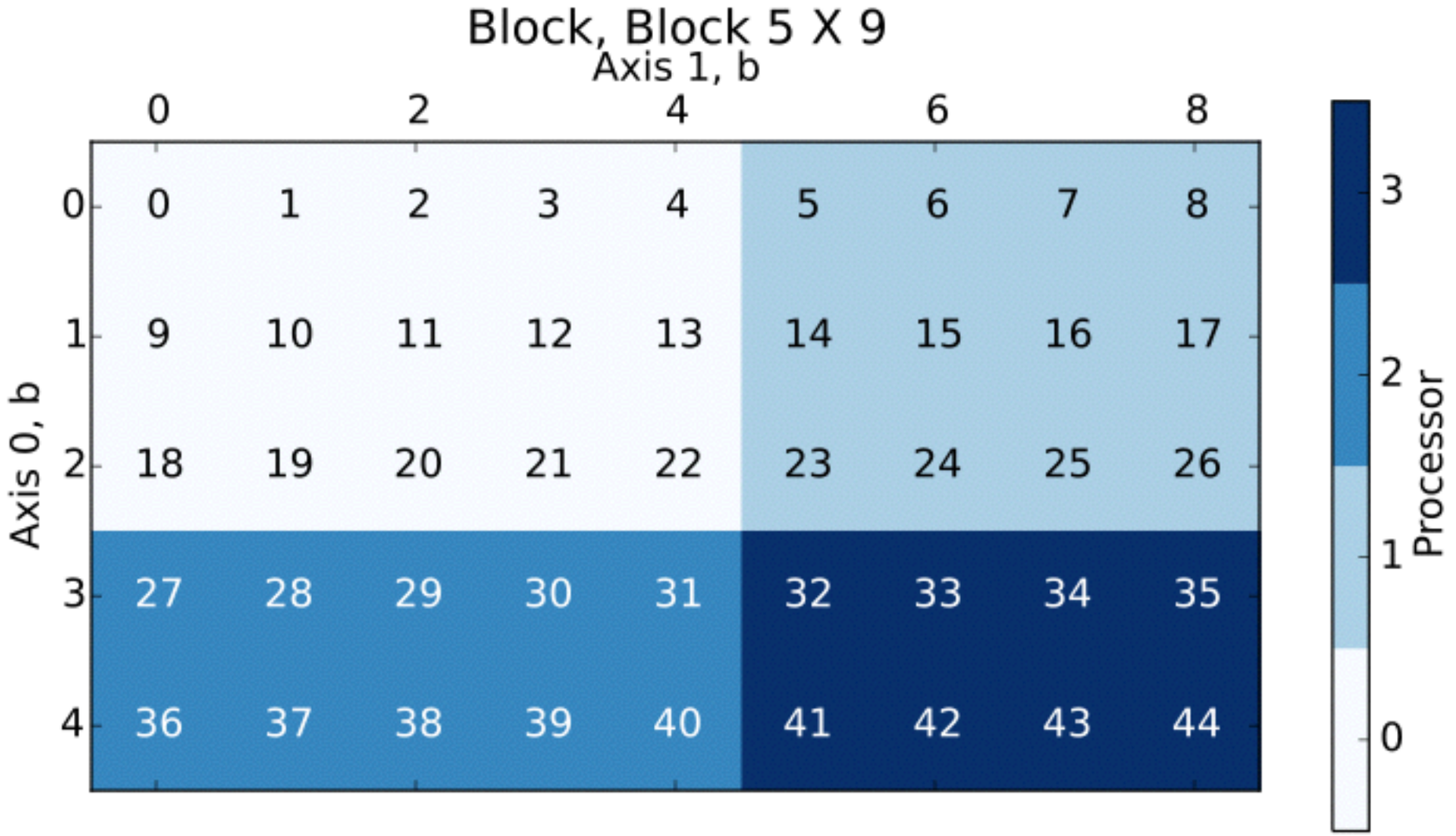
Acknowledgement and Disclaimer

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

Benchmark


Julia Set Benchmark
resolution=[8192], c=[u'(-0.045+0.45j)'], total_time=995.00s, nreps=6
b-b
b-n
c-c
c-n
ideal numpy
mpoints / s
90
80
70
60
50
40
30
20
10
0
1
2
3
4
5
6
7
8
9
10
11
12
Engine Count

Distributions - Block-Block



Distributions - Block-Block

Local Arrays

Process (0, 0)

0	1	2	3	4
9	10	11	12	13
18	19	20	21	22

Process (0, 1)

5	6	7	8
14	15	16	17
23	24	25	26

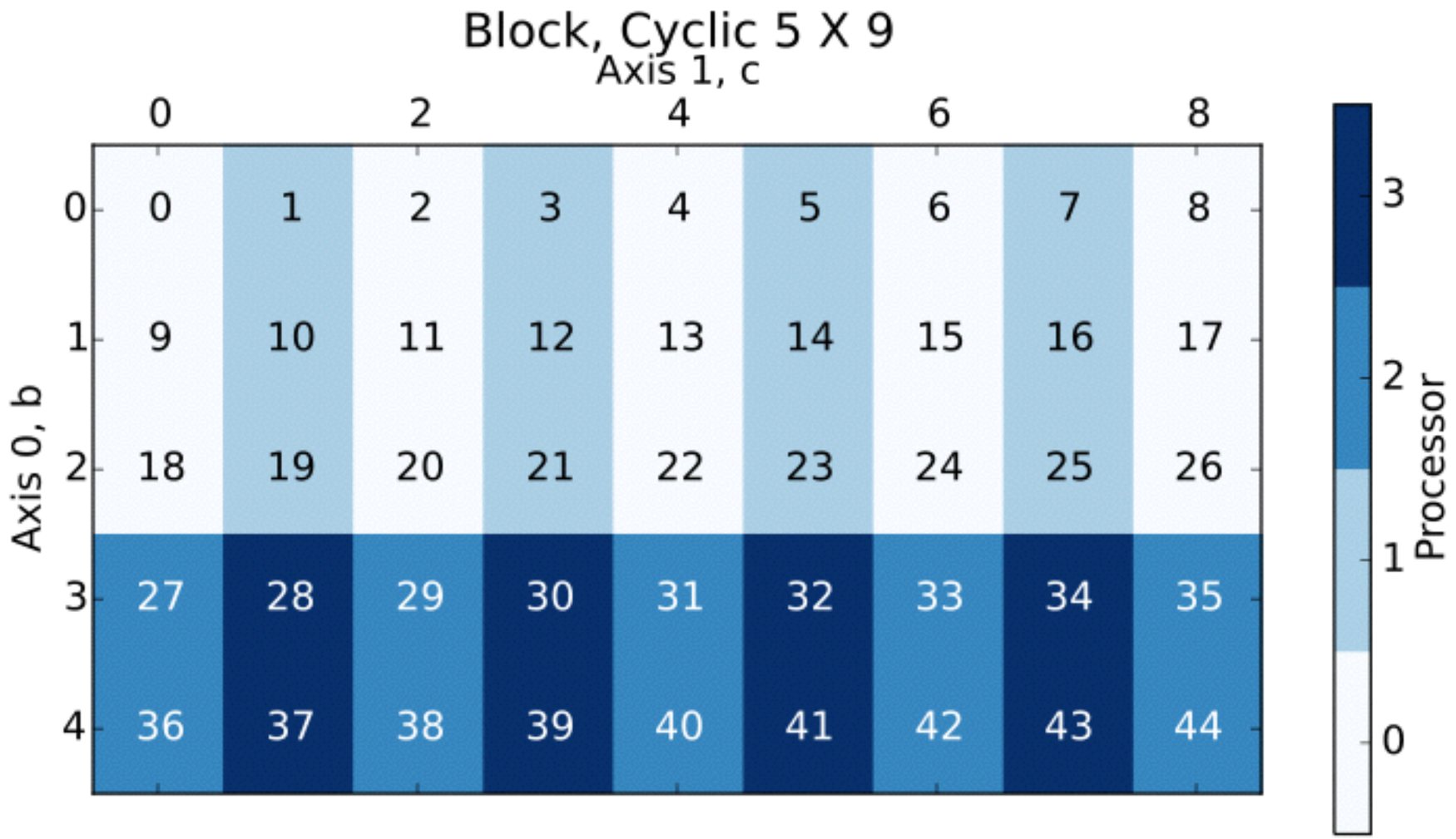
Process (1, 0)

27	28	29	30	31
36	37	38	39	40

Process (1, 1)

32	33	34	35
41	42	43	44

Distributions - Block-Cyclic



Distributions - Block-Cyclic

Local Arrays

Process (0, 0)

0	2	4	6	8
9	11	13	15	17
18	20	22	24	26

Process (0, 1)

1	3	5	7
10	12	14	16
19	21	23	25

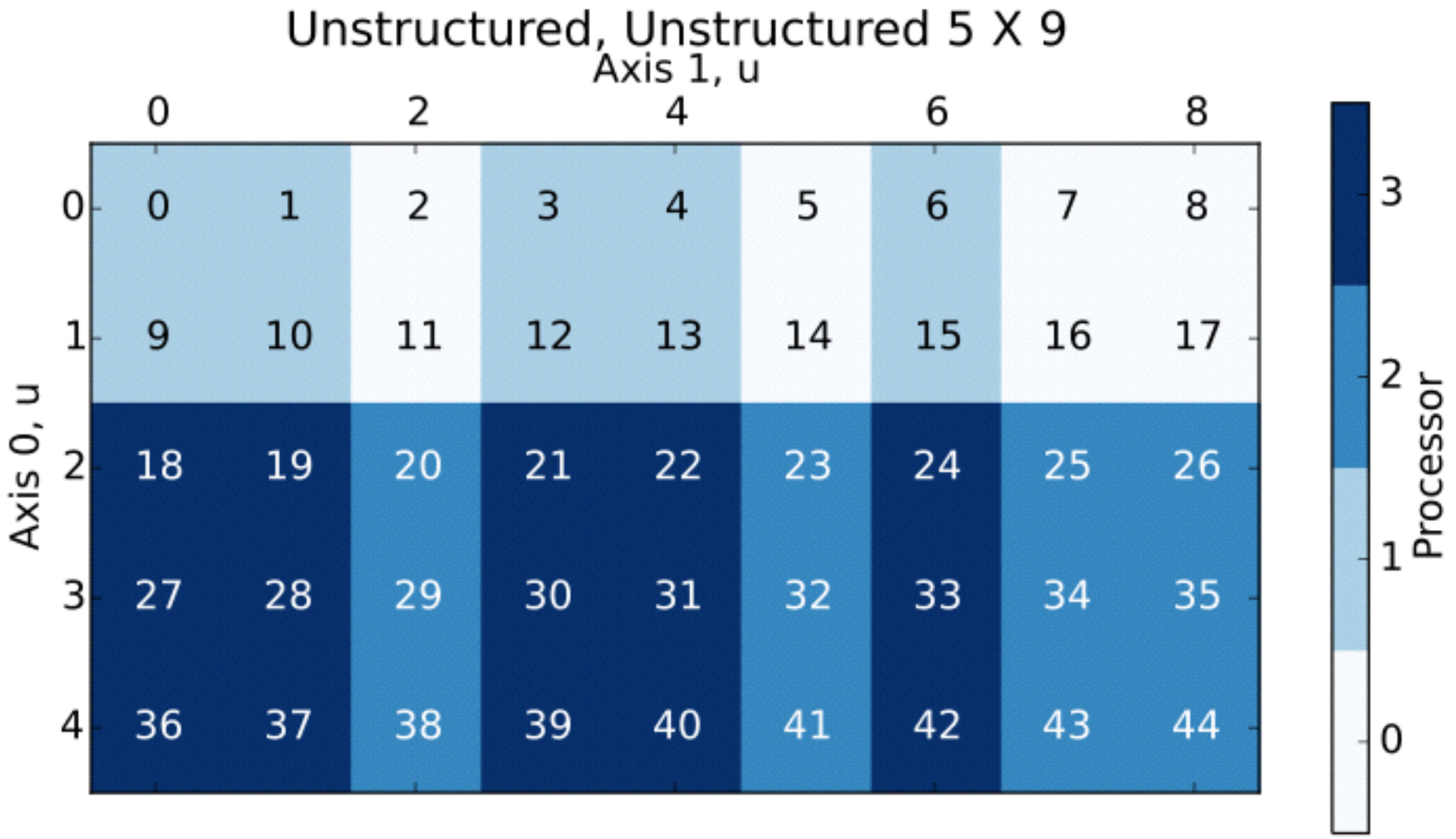
Process (1, 0)

27	29	31	33	35
36	38	40	42	44

Process (1, 1)

28	30	32	34
37	39	41	43

Distributions - Unstruct.-Unstruct.



Distributions – Unstruct.-Unstruct.

Local Arrays

Process (0, 0)

16	17	11	14
7	8	2	5

Process (0, 1)

13	15	10	12	9
4	6	1	3	0

Process (1, 0)

43	44	38	41
25	26	20	23
34	35	29	32

Process (1, 1)

40	42	37	39	36
22	24	19	21	18
31	33	28	30	27