# Automatic Parallelization of Numerical Python Applications using the Global Arrays Toolkit

Jeff Daily, Robert R Lewis

Pacific Northwest NATIONAL LABORATORY
Proudly Operated by Battelle Since 1965

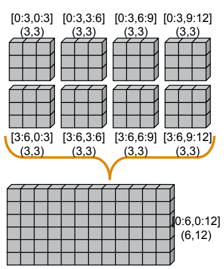
## Motivation

- Lots of NumPy applications
  - NumPy (and Python) are for the most part single-threaded
  - Resources underutilized
    - Computers have multiple cores
    - Academic/business clusters are common
- Lots of parallel libraries or programming languages
  - Message Passing Interface (MPI), Global Arrays (GA), X10, Co-Array Fortran, OpenMP, Unified Parallel C, Chapel, Titianium, Cilk
  - Can we transparently parallelize NumPy?

## Design for Global Arrays in NumPy (GAiN)

- All documented NumPy functions are collective.
- GAiN programs run in SPMD fashion.
- Not all arrays should be distributed.
- GAiN operations should allow mixed NumPy/GAiN inputs.
- Reuse as much of NumPy as possible (obviously).
- Distributed nature of arrays should be transparent to user.
- Use owner-computes rule to attempt data locality optimizations.

## The `gain.ndarray` in a Nutshell




- Global shape and *P* local shapes
- Memory allocated from Global Arrays library, wrapped in local `numpy.ndarray`
- The memory distribution is static
- Global operations translate to *P* local `numpy` operations

## GAiN is a Functioning Prototype

- Released with Global Arrays 5.1
- What's finished:
  - Ufuncs (all, but not `reduceat` or `outer`)
  - `ndarray` (mostly)
  - `flatiter`
  - *`numpy dtype`s are reused!*
  - Various array creation and other functions:
    - `zeros, zeros_like, ones, ones_like, empty, empty_like`
    - `eye, identity, fromfunction, arange, linspace, logspace`
    - `dot, diag, clip, asarray`
- Everything else doesn't yet exist, including column-major (i.e. Fortran) array ordering

## How to Use GAiN

Ideally, change one line in your script:

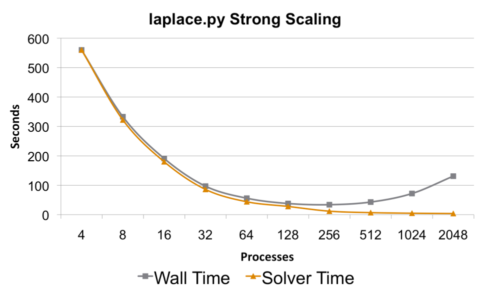
```
#import numpy
import ga.gain as numpy
```

Run using the MPI process manager:

```
$ mpiexec -np 4 python script.py
```

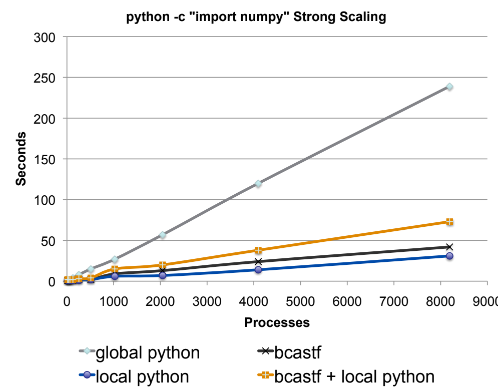
## Scaling of Python Interpreter

- laplace.py: Laplace equation using an iterative finite difference scheme.
- Wall clock and iterative solver times are compared.
- Discrepancy caused by many Python instances reading and writing same files on shared file system during module loading.




## Scaling the Python Interpreter

- Diskless compute nodes e.g. BlueGene/P
  - Walla library by William Scullin of Argonne
  - Use Python's "import hooks" or modified interpreter
  - Process 0 reads from disk, broadcasts libraries and modules
- Compute nodes with local disks e.g. chinook at EMSL
  - Process 0 broadcasts installed Python and required modules to local disks on compute nodes ('bcastf' below)
  - Run Python from local compute nodes' disks
  - Reduces contention utilizing local disk copies




## Analysis

- The above test only loads standard Python modules and all numpy modules
- Contention for the global python case would only get worse as additional modules are loaded; not so for local disk

## Future Work

- Performance comparison between import hooks, modifying the Python interpreter, and bcastf to local disks
- DOE SBIR proposal to further develop GAiN is under review

## About Pacific Northwest National Laboratory

The Pacific Northwest National Laboratory, located in southeastern Washington State, is a U.S. Department of Energy Office of Science laboratory that solves complex problems in energy, national security and the environment, and advances scientific frontiers in the chemical, biological, materials, environmental and computational sciences. The Laboratory employs 4,000 staff members, has a $760 million annual budget, and has been managed by Ohio-based Battelle since 1965.

For more information about the science you see here, please contact:

**Jeff Daily**
Pacific Northwest National Laboratory
P.O. Box 999, MS-K7-90
Richland, WA 99352
(509) 372-6548
jeff.daily@pnl.gov

### Software Website

http://www.emsl.pnl.gov/docs/global/

### Collaborators

Washington State University Tri-Cities

(A portion of) The research was performed using EMSL, a national scientific user facility sponsored by the Department of Energy's Office of Biological and Environmental Research and located at Pacific Northwest National Laboratory.