BERKELEY LAB
Bringing Science Solutions to the World

U.S. DEPARTMENT OF ENERGY

ArrayUDF Explores Structural Locality for Faster Scientific Analyses

John Wu[1]

Bin Dong[1], Surendra Byna[1], Jialin Liu[1], Weijie Zhao[2], Florin Rusu[1,2]

[1]LBNL, Berkeley, CA

[2]UC Merced, Merced, CA

Two common approaches for supporting diverse data analysis operations

Customized Solutions

For each operation *P* **Do**

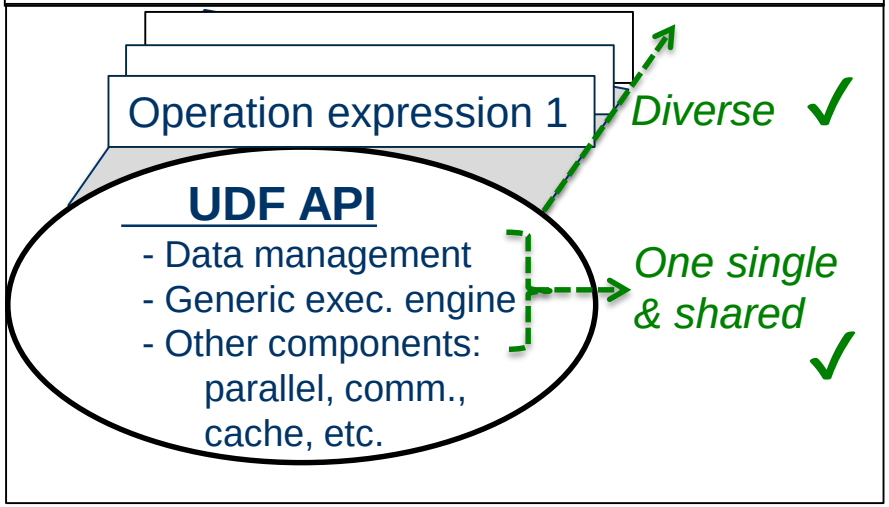
Develop *P*'s :

- Data management --> *Redundant* ✗
- Expression execution --> *Diverse* ✓
- Other components: --> *Redundant* ✗
  parallel,
  communication
  cache,
  etc.

End For

May lack expertise of the underlying systems to tune its performance ✗

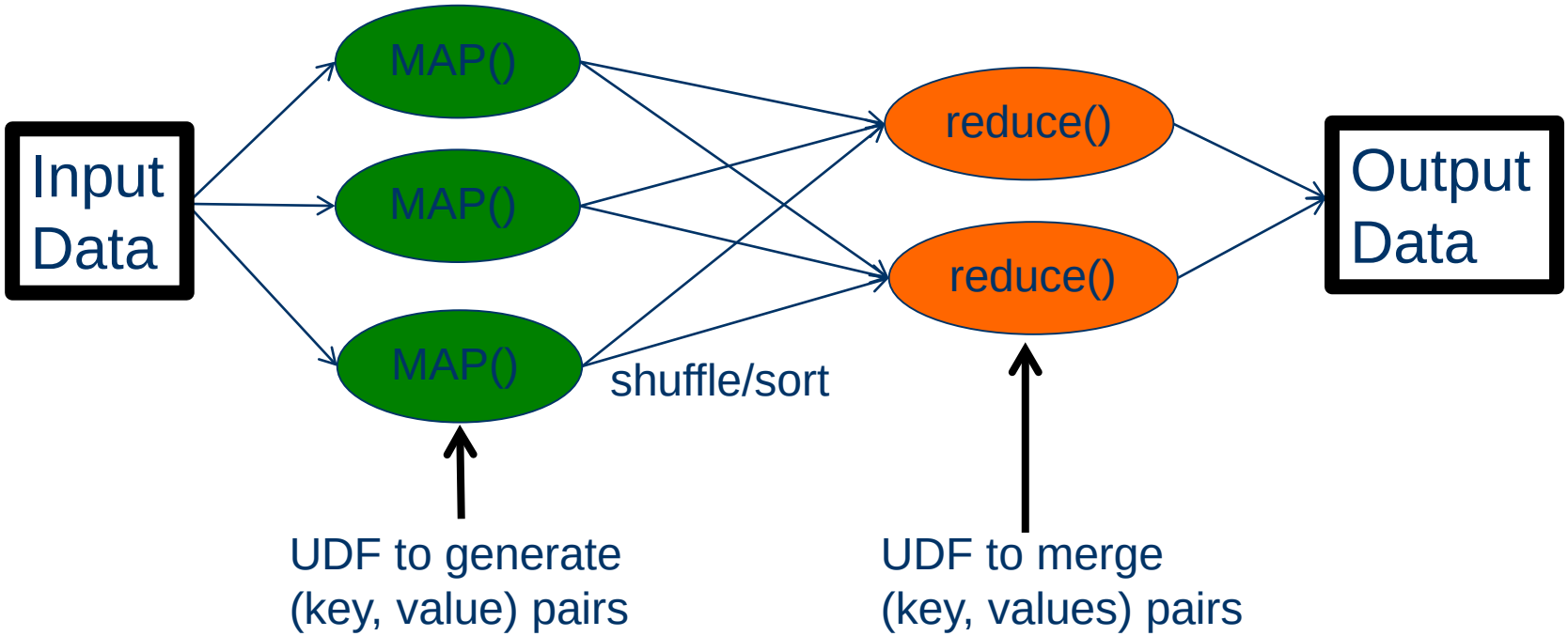
User-defined Functions (UDF)



Professionally tuned ✓

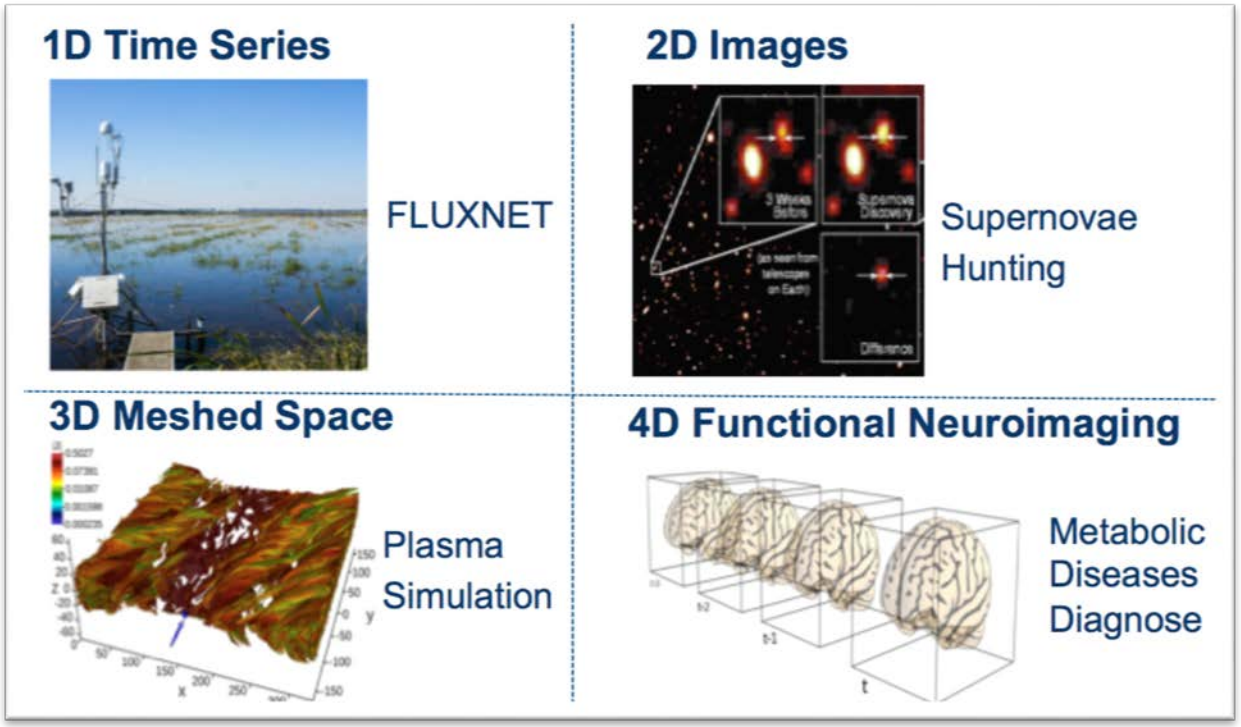
UDF is at heart of modern big data system

Examples: MapReduce in Apache Hadoop and Spark



MapReduce is not an optimal fit for scientific data analysis

Reason 1: most scientific data are multi-dimensional arrays



Pictures Credit: Kyle Hemes, Peter Nugent, Suren Byna, etc.

Converting array to (key, value) is expensive because of explicitly handling coordinate

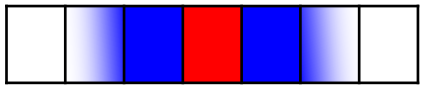
MapReduce is not an optimal fit for scientific data analysis (continued)

Reason 1: most scientific data are multi-dimensional arrays

→ Converting array to (key, value) is expensive

Reason 2: most scientific data analysis operations have structures such as stencils

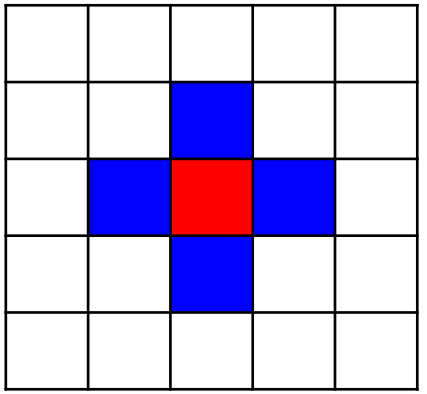
Moving Average



Structure locality:

Pattern of neighbors accessed during an analysis operation

2D Poisson Equation Solver (Discrete)



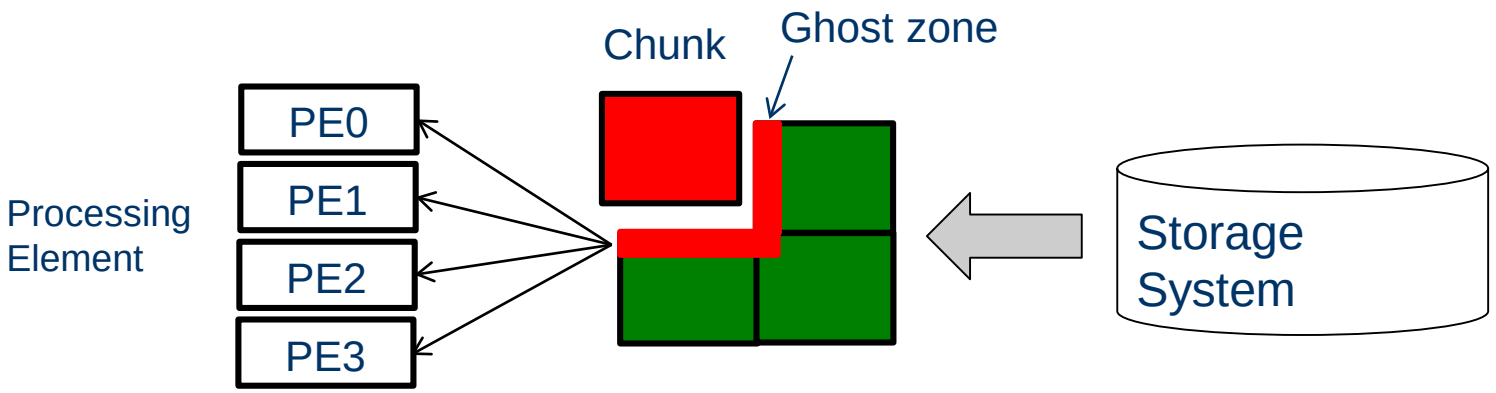
→ Map deals with a single element at a time

→ Reduce requires to duplicate each cell for all neighborhood cells

→ Reduce only happens after expensive shuffle

ArrayUDF: user-defined scientific data analysis on arrays

- Stencil-based user-defined function
  - ➡ Structural locality aware array operations
- Native multidimensional array data model
  - ➡ In-situ data processing in scientific data formats, e.g., HDF5
- Optimal and automatic chunking and ghost zone handling
  - ➡ Fast large array processing in parallel & out-of-core manner



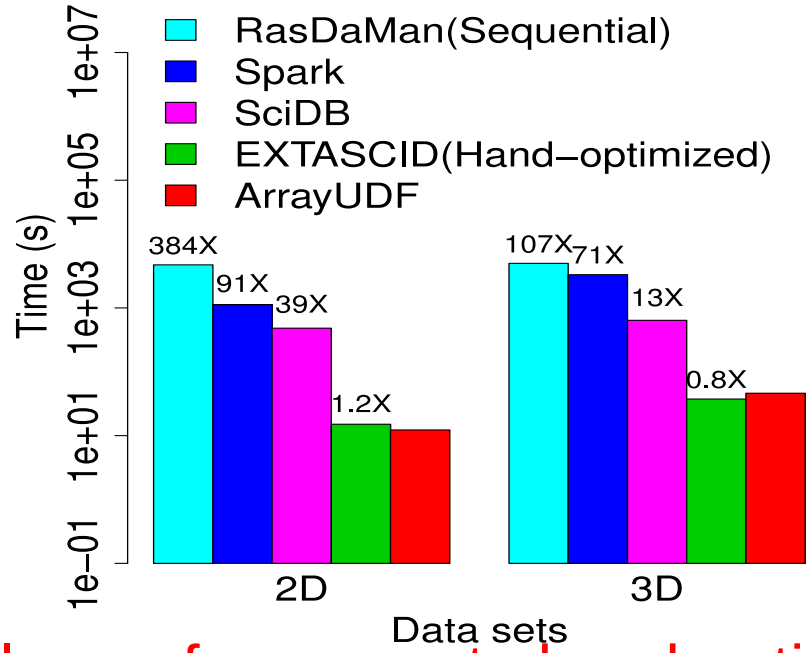
Evaluations

- Hardware:
  - Edison, a Cray XC30 supercomputer at NERSC
  - 5576 computing nodes, 24 cores/node, 64GB DDR3 Memory
- Software
  - ArrayUDF
  - Spark 1.5.0
  - SciDB 16.9
  - RasDaMan 9.5 (sequential version)
  - EXTASCID(hand-optimized version)
  - Hand-optimized C/C++ code
- Workloads
  - Two synthetic data sets (i.e., 2D and 3D) for micro benchmarks
    - Window operators, chunking strategy, trial-run, etc.
  - Four real scientific data sets (i.e., S3D, MSI , VPIC , CoRTAD)
    - Overall performance tests /w generic UDF interface

Comparison with peer systems on a single compute node

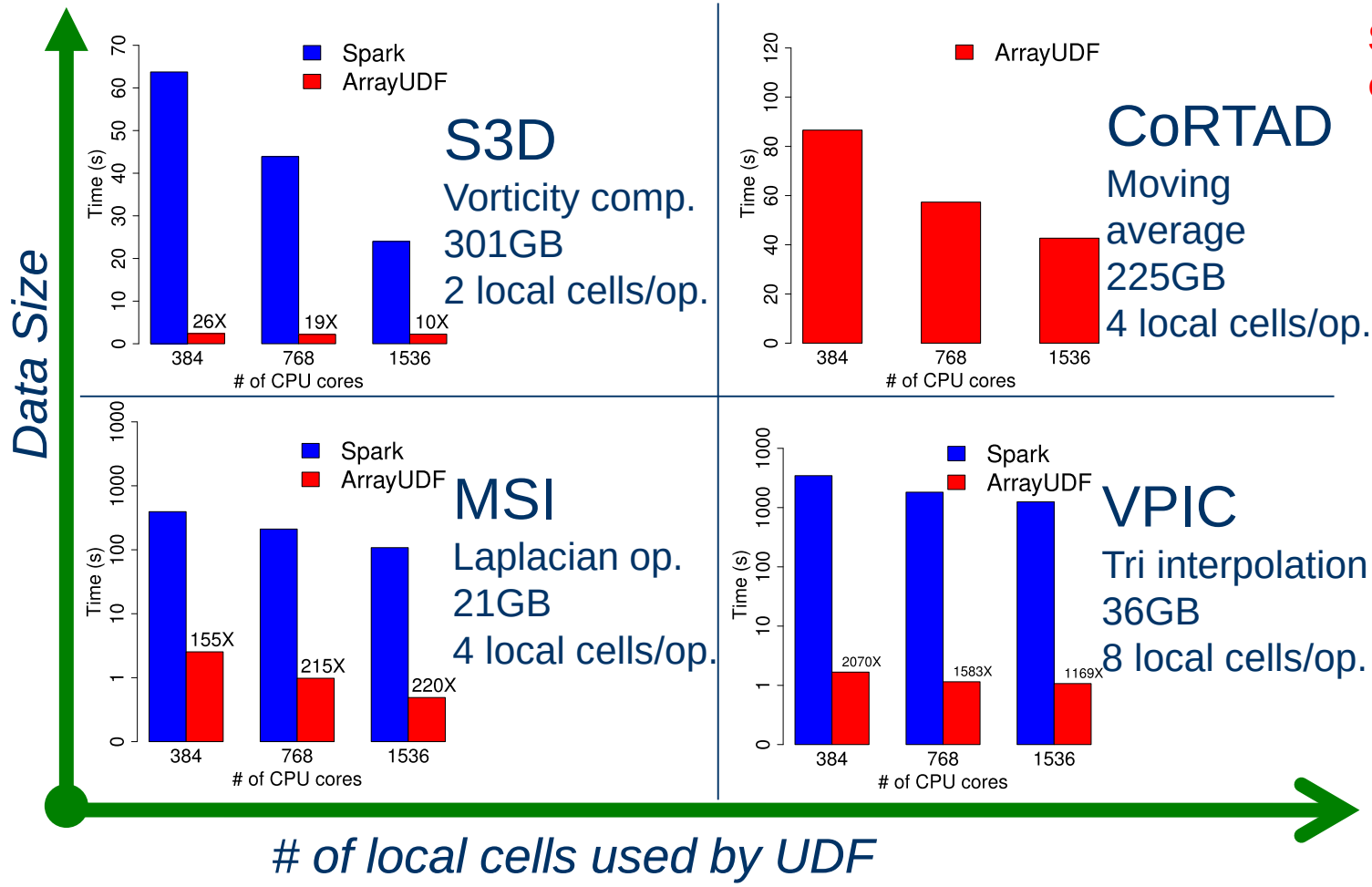
- "Window" concept comes from SciDB and RasDaMan, where an operator is applied to all members within the regular neighborhood

Average on
- window 2x2 for 2D
- window 2x2x2 for 3D



- ArrayUDF has similar performance to hand-optimized code
- ArrayUDF is as much as 384X faster than peer systems

Comparison with Spark on large datasets



Spark experiences out-of-memory:

- **large data size**
- **more local cells**

We observed ArrayUDF is **2070X** faster

More details available at
https://doi.org/10.1145/3078597.3078599

BERKELEY LAB

Summary

- ArrayUDF: User-defined scientific data analysis on arrays
  - Stencil based UDF for structural locality-aware operations
  - Native array model & *in situ* array processing in HDF5, etc.
  - Auto & Optimal chunking and ghost zone methods for parallel or out-of-core array processing
- ArrayUDF provides similar performance as a hand-optimized code
- ArrayUDF is as much as *2070X* faster than Spark
- ArrayUDF source code: https://bitbucket.org/arrayudf/
- Future work
  - Python and other language interface (Done)
  - more in-situ formats: NetCDF, PnetCDF, ADIOS, etc.

More details available at
https://doi.org/10.1145/3078597.3078599

Acknowledgments

- Nicholas Chaimov from University of Oregon for suggestions to set up Spark on Edison at NERSC
- Office of Advanced Scientific Computing Research, Office of Science, U.S. Department of Energy, support for the SDS project and a DOE Career award (Program managers: Laura Biven and Lucy Nowell) under contract number DE-AC02-05CH11231

- National Energy Research Scientific Computing Center