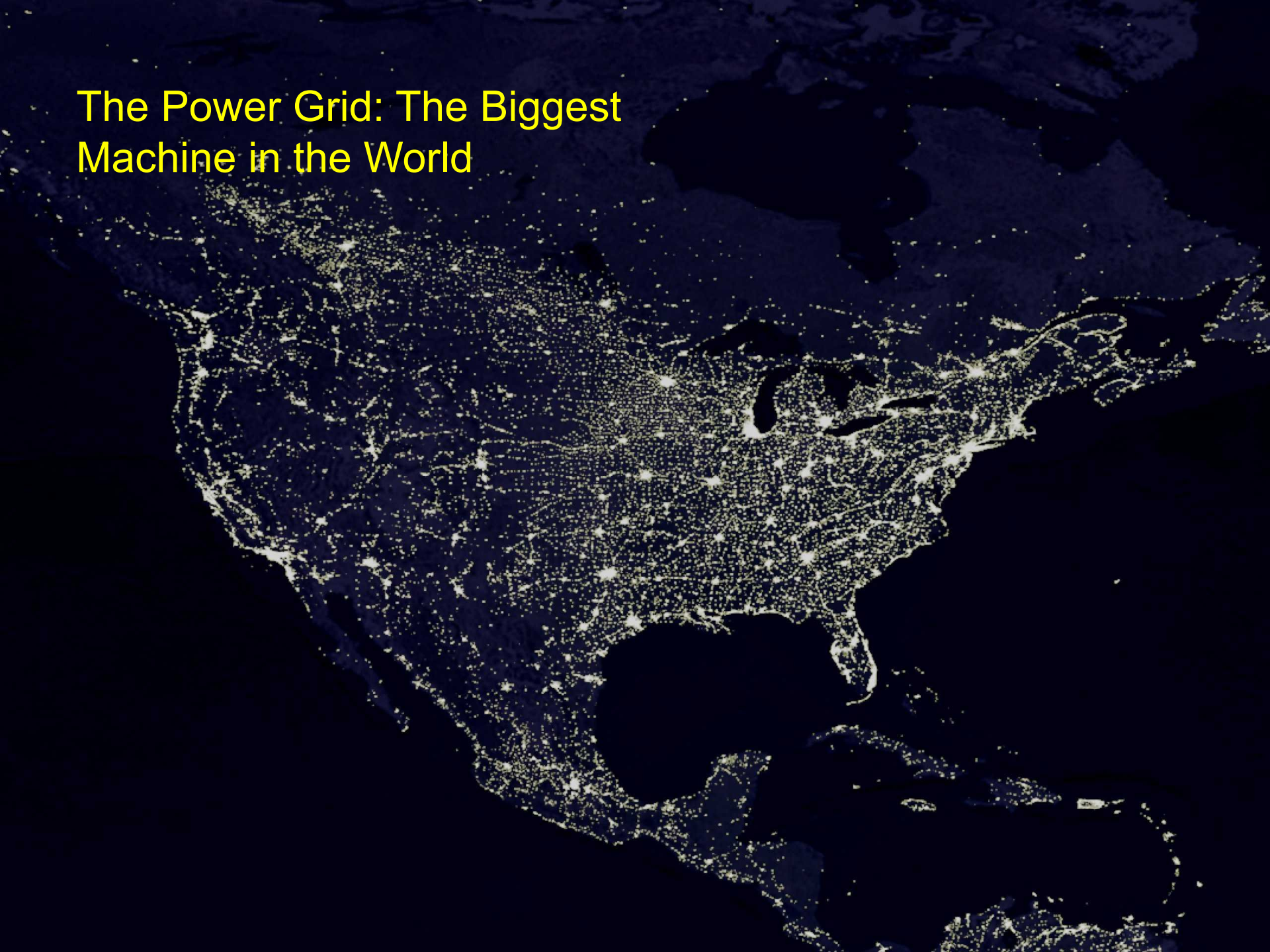
Application of PGAS Programming to Power Grid Simulation

Bruce Palmer

Pacific Northwest National Laboratory

Richland, WA

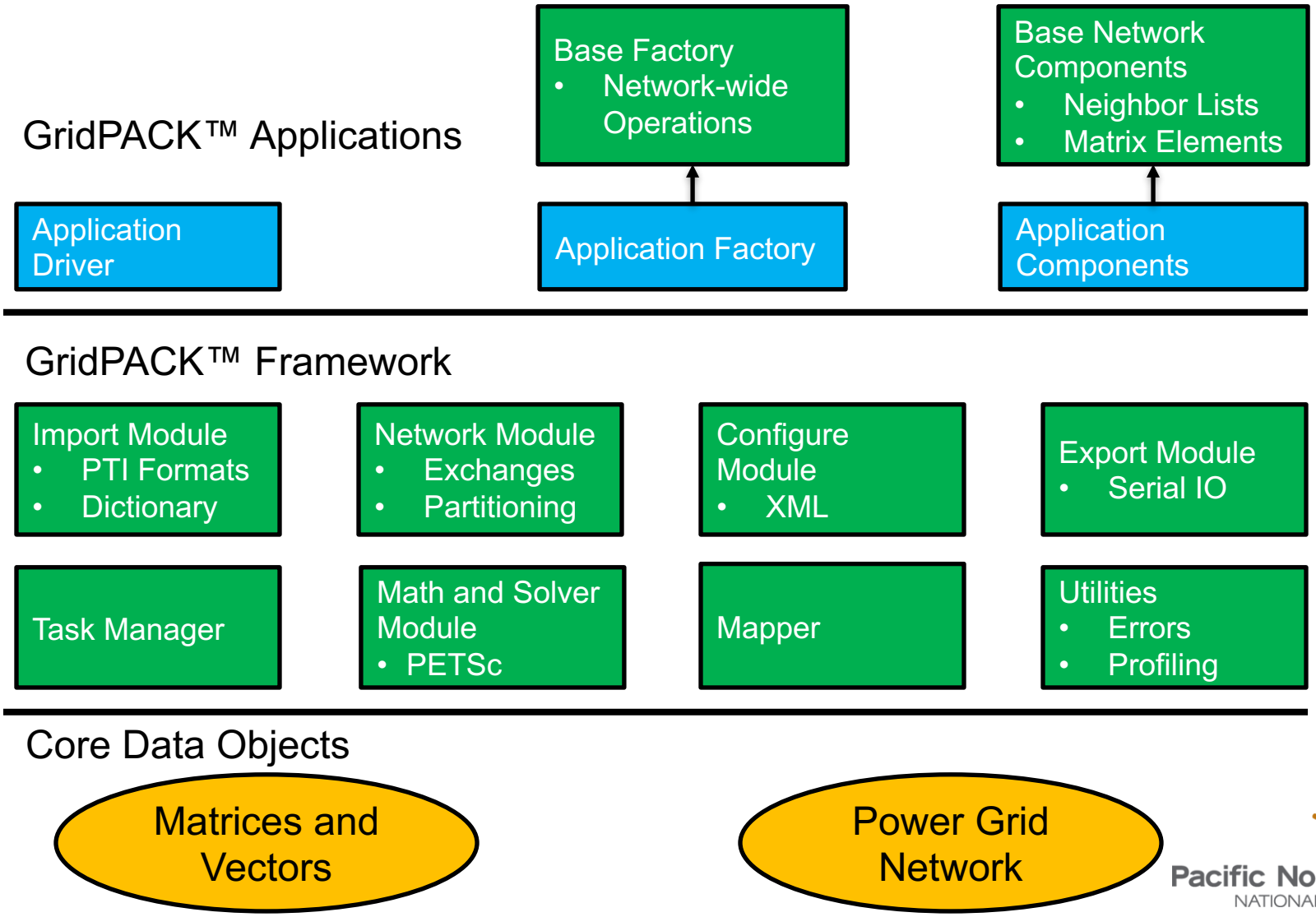
The Power Grid: The Biggest Machine in the World



Parallel Computing and the Power Grid

- Most problems are fairly small and still fit on workstations
- But…
  - There is interest in simulating problems very fast. Going from 2 minutes to 15 seconds can be important for real-time control
  - Contingency calculations can spawn thousands or even millions of individual simulations. These can all be run concurrently
  - Bigger calculations are on the horizon (optimization, transmission plus distribution, etc.)

GridPACK™ Framework



Pacific Northwest NATIONAL LABORATORY

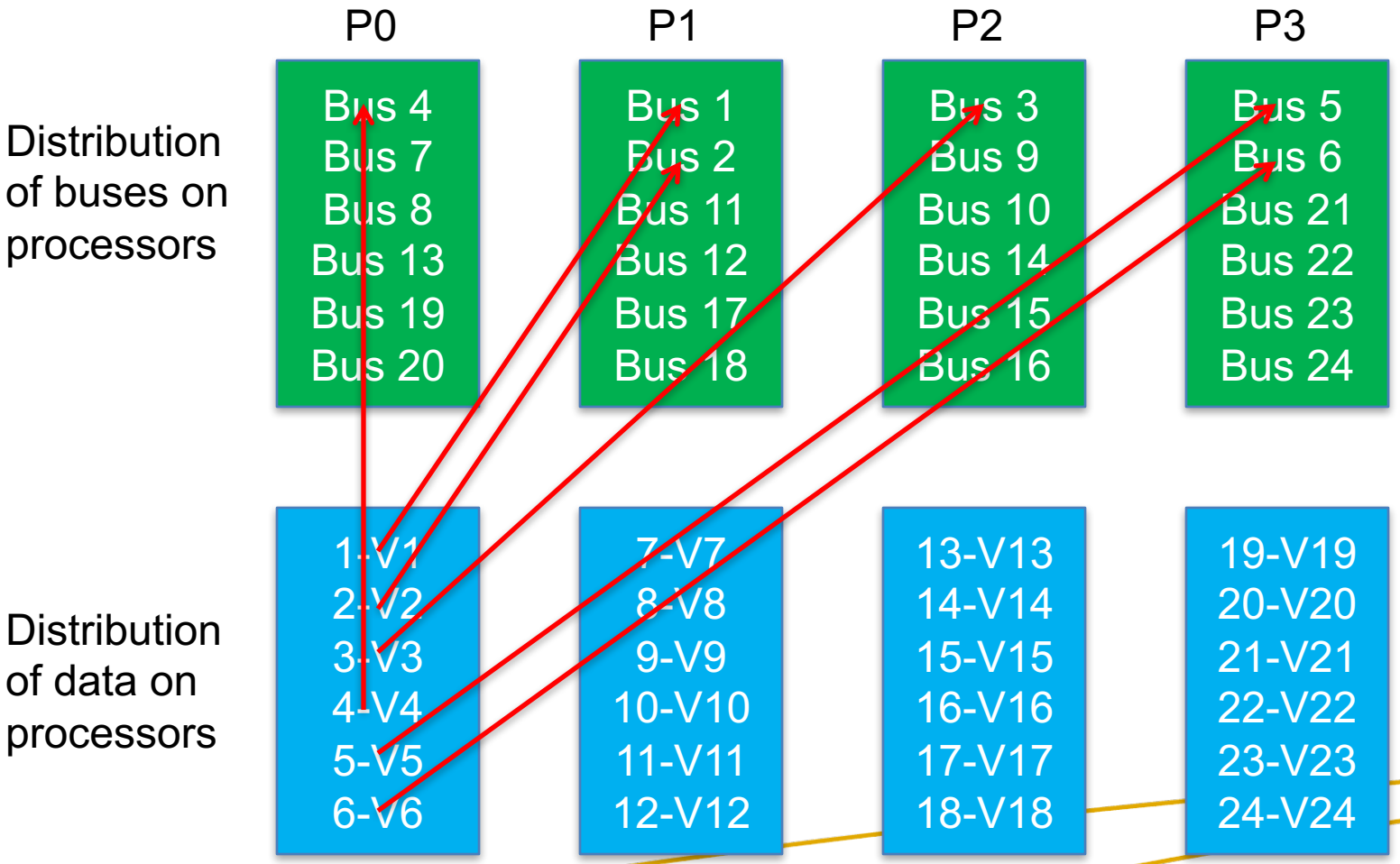
Global Arrays Communication Library

- Distributed arrays with a global address space
- One-sided access using put/get/accumulate (structured access) and scatter/gather/scatter-accumulate (unstructured access)
- Read-increment functionality
- Native runtimes: Infiniband, Portals4, and Cray
- MPI-based runtimes: Two-sided, progress ranks, multi-threaded, progress threads, and MPI-3

Parallel Tasks Implemented using PGAS

- Ghost bus (node) and branch (edge) exchanges (scatter/gather)
- I/O (scatter/get)
- Task manager (read-increment)
- Collective hash tables (put/get and read-increment)
- etc.

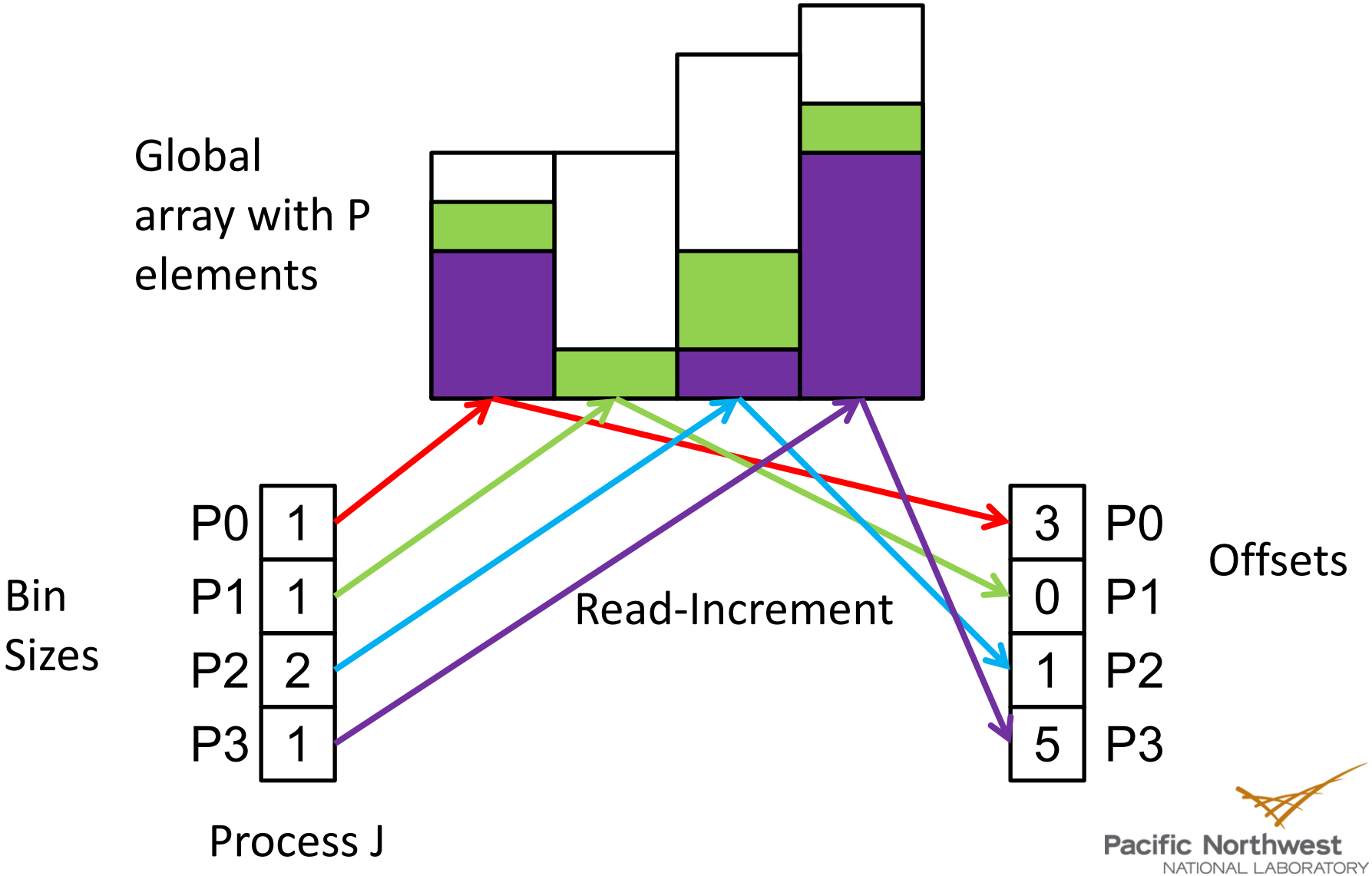
Collective Hash Tables



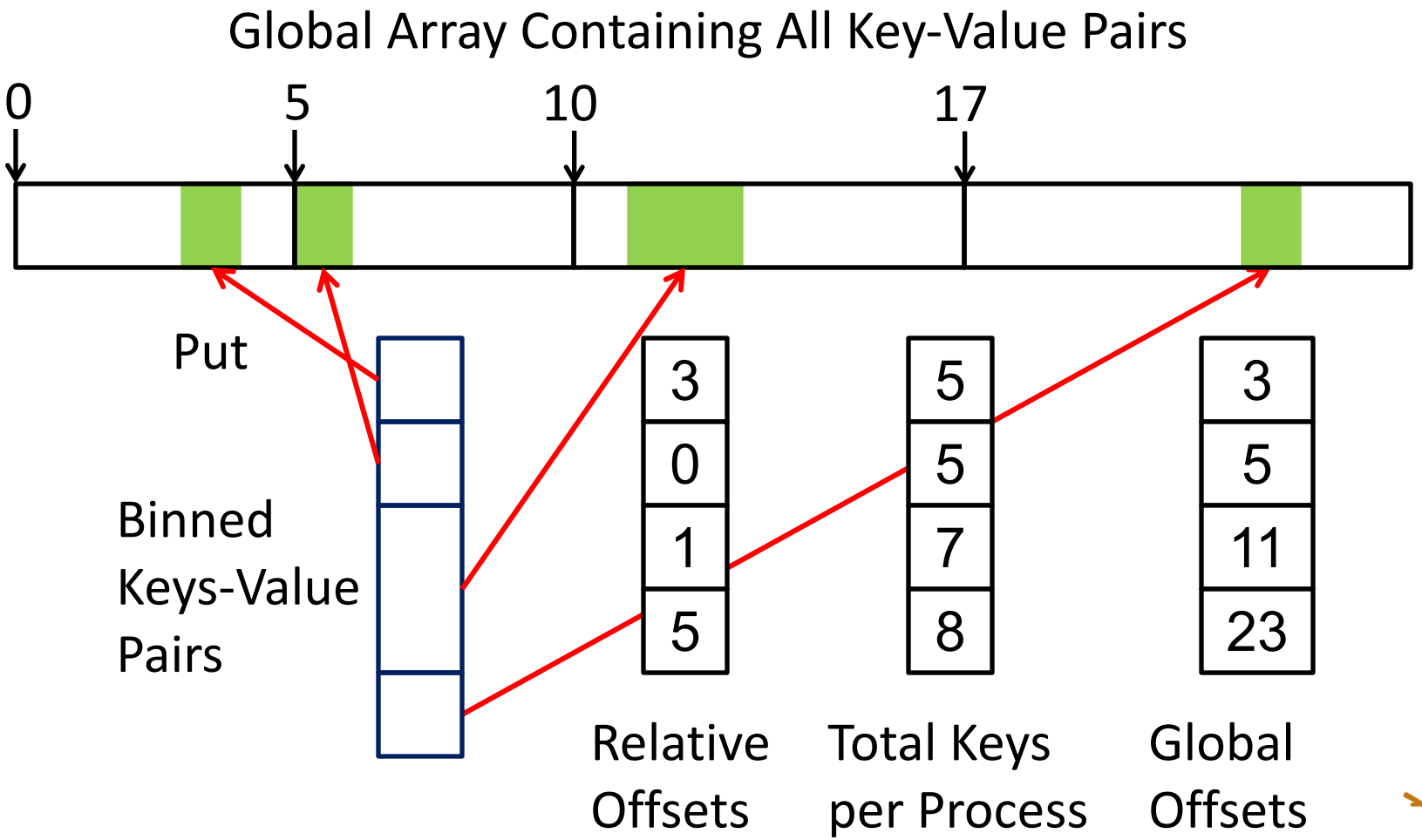
Hash Algorithm

- Setup and access are collective operations
  - Requests can be bundled and processed all at once
  - Progress is assumed on all processors
- Create a hash function that maps bus (node) indices to a processor
  - Processor doesn't need to own the bus, but it is the place to go for information
- Create a hash table that maps all bus (node) indices to the processors that contain the bus
- Use the hash table to route data to processors that own the bus

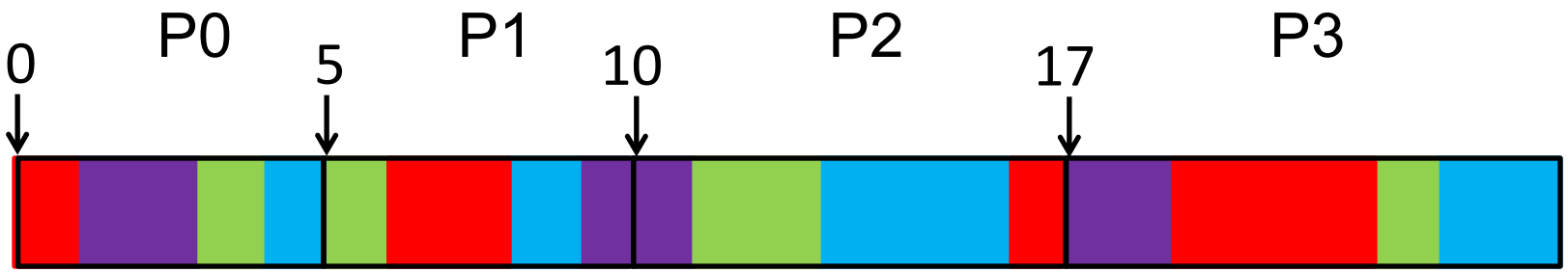
Initialization:Evaluate Bin Sizes and Offsets



Initialization: Put All Data in Global Array



Initialization: Key-Value Pairs Are Distributed



Key-value pairs from P0

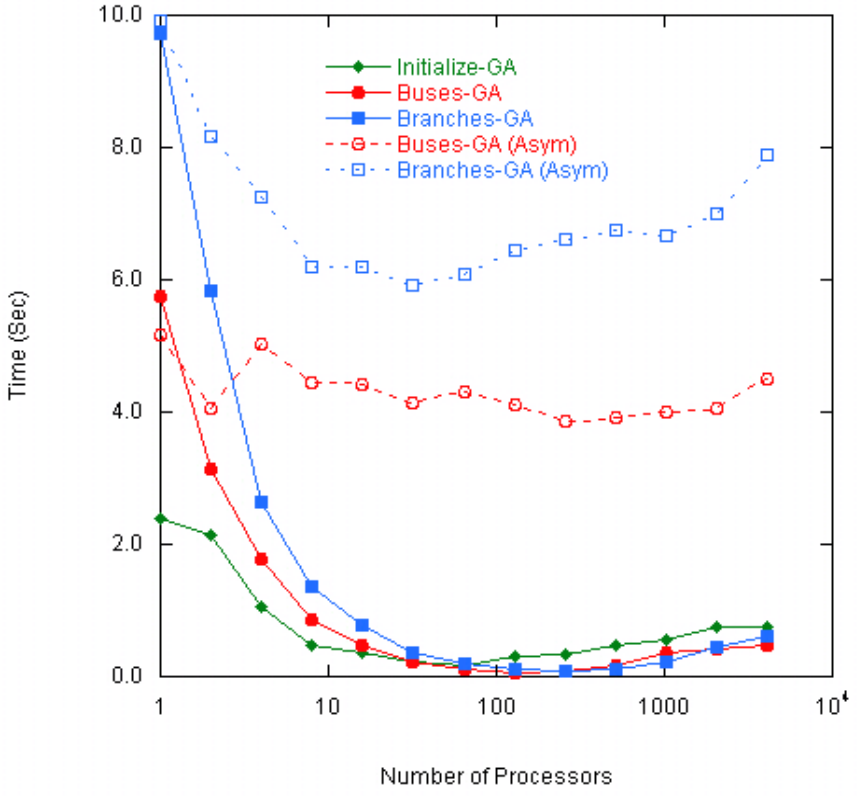
Key-value pairs from P1

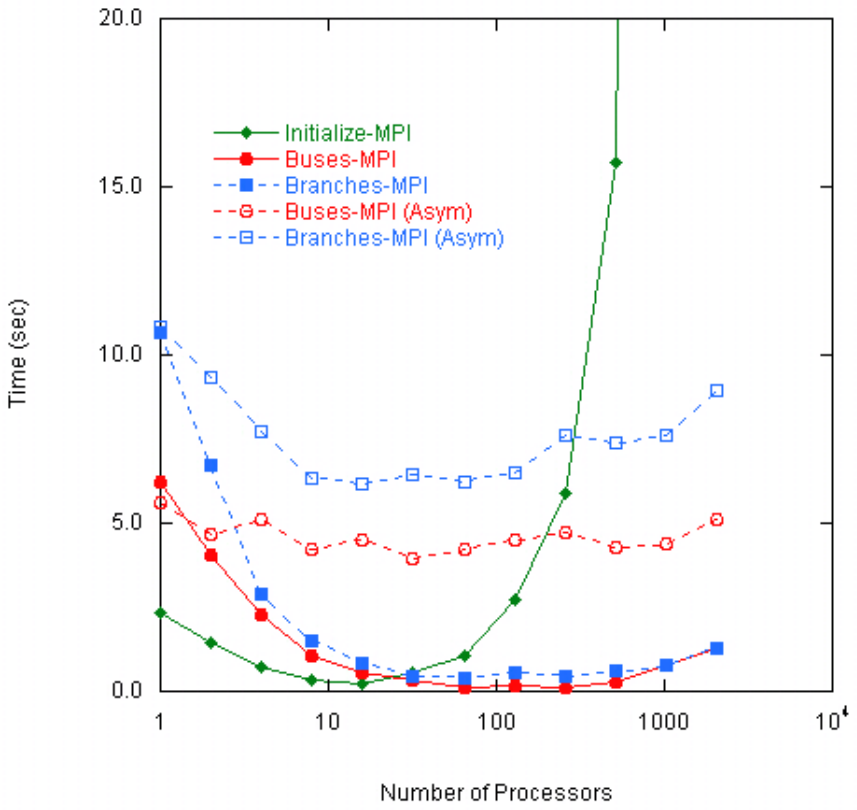
Key-value pairs from P2

Key-value pairs from P3

Pacific Northwest
NATIONAL LABORATORY

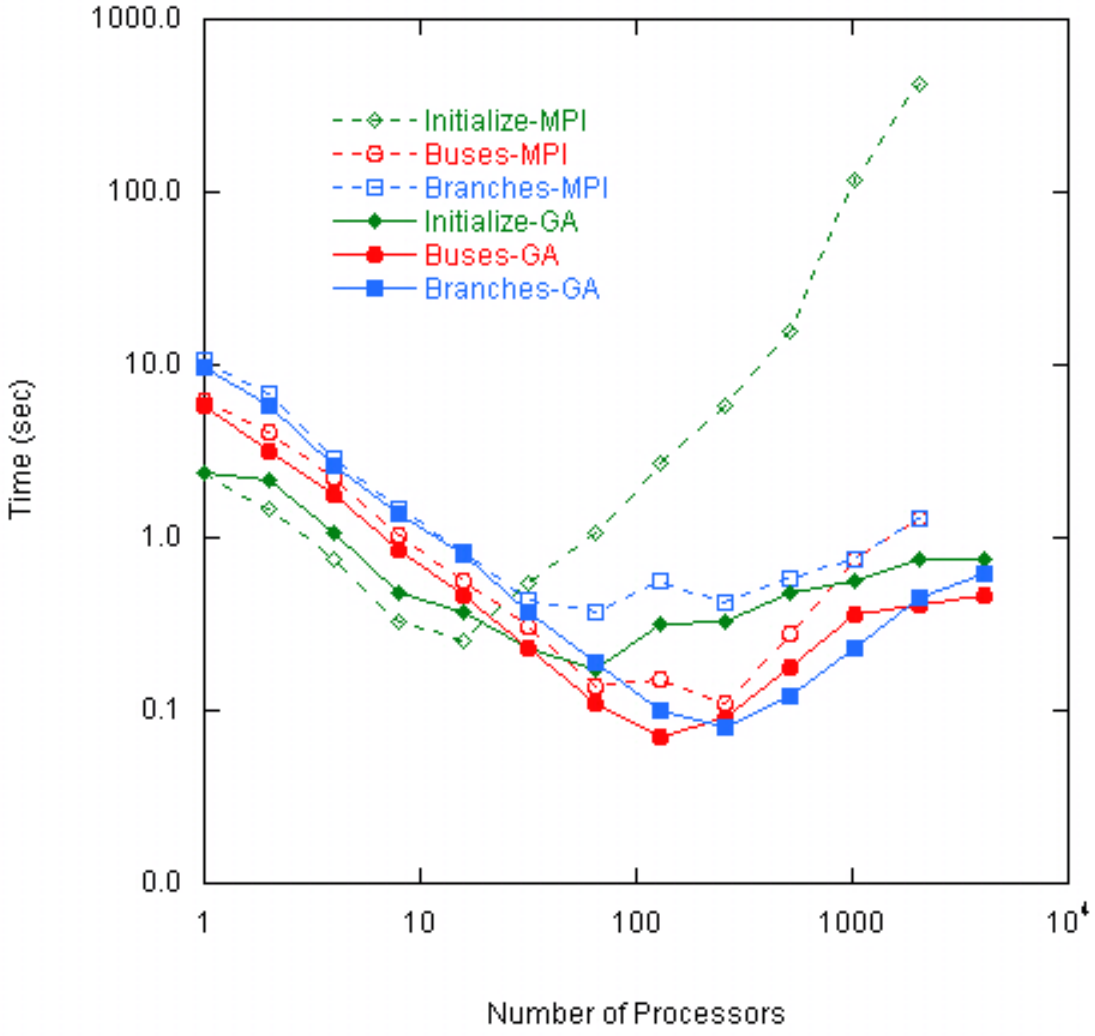
Results





Pacific Northwest
NATIONAL LABORATORY

Results



Pacific Northwest
NATIONAL LABORATORY

Conclusions

- PGAS programming models are easily applied to power grid applications
- Performance is comparable to MPI for many functions, in some cases it is better
- Collective hashing can be used to redistribute data efficiently based on matching keys
- GridPACK available at https://www.gridpack.org