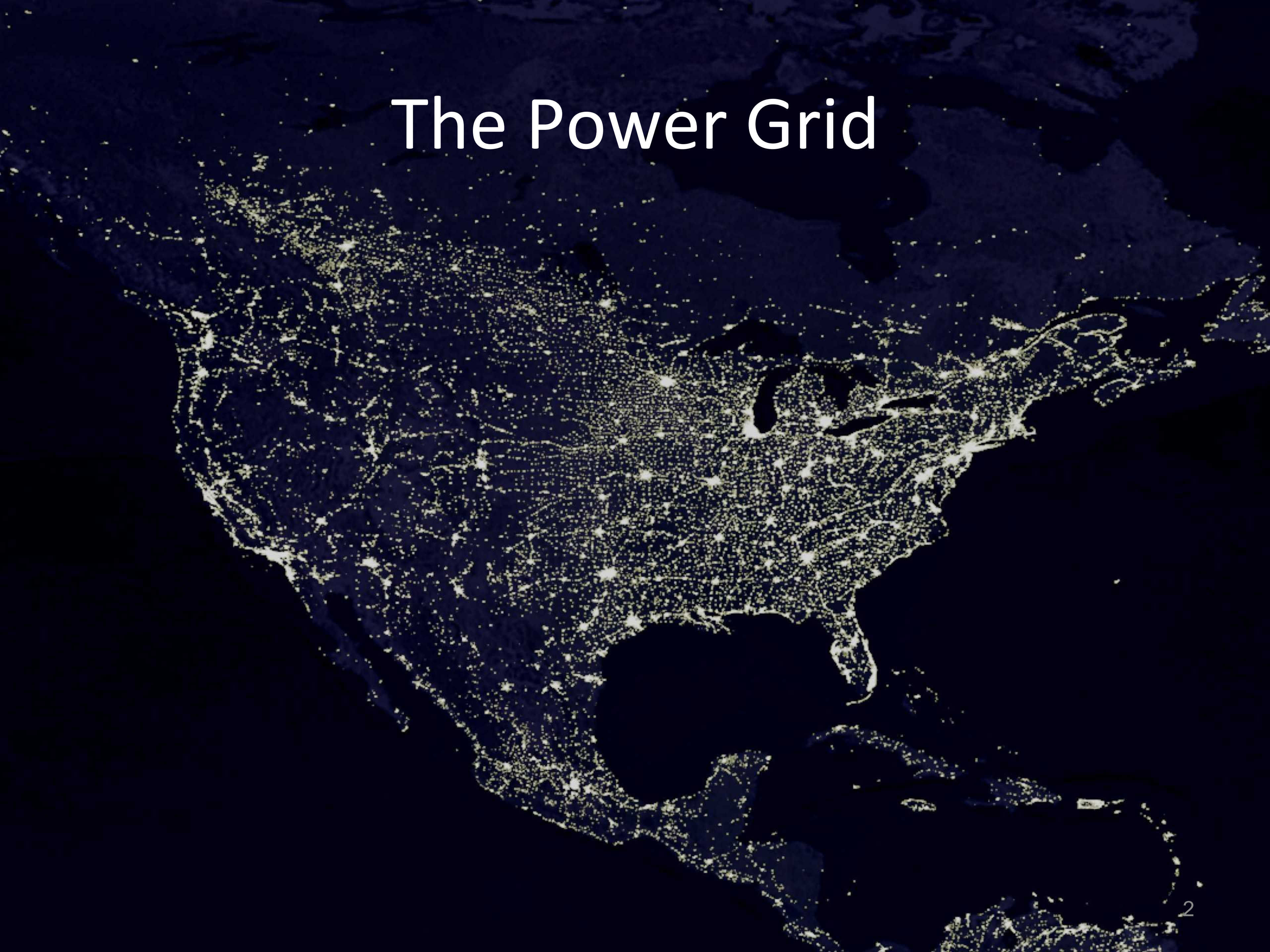
GridPACK™: A Framework for Developing Power Grid Simulations on High Performance Computing Platforms

Bruce Palmer, William Perkins, Yousu Chen, Shuangshuang Jin, David Callahan, Kevin Glass, Ruisheng Diao, Mark Rice, Stephen Elbert, Mallikarjuna Vallem, Zhenyu (Henry) Huang

Pacific Northwest
NATIONAL LABORATORY
Proudly Operated by Battelle Since 1965

The Power Grid



Motivation

- Power grid is enormously complex but is still mostly modeled using serial code
- To fit models onto a single core, numerous approximations and aggregations have been used
- Using parallel code would allow engineers to relax some of the current restrictions in their models but the barrier to creating parallel code is high

Considerations

- The power grid is represented as a graph
- Graph nodes (buses) and edges (branches) are highly heterogeneous. Different buses and branches can have very different qualities
- Most models describing the power grid are expressed in terms of non-linear algebraic equations involving complex variables
- Need to capture a diverse range of models and numerical approaches to cover power grid applications

GridPACK™ Framework

- Object-oriented design implemented in C++
- Use software templates and inheritance to create application-specific versions of most framework modules
- Hide all communication and minimize the number of parallel concepts that application developers must deal with
- Focus on “local” calculations in application
- Wrap math libraries to separate libraries from the rest of the framework and allow seamless substitution of other libraries in the future

GridPACK™ Applications

- Y-matrix
- Powerflow
- Dynamic Simulation

Base Factory
- Network-wide Operations

Base Network Components
- Neighbor Lists
- Matrix Elements

Application Driver

Application Factory

Application Components

GridPACK™ Framework

Import Module
- PTI Formats
- Dictionary

Network Module
- Exchanges
- Partitioning

Configure Module
- XML

Export Module
- Serial IO
- PTI Formats

Task Manager

Math and Solver Module
- PETSc

Mapper

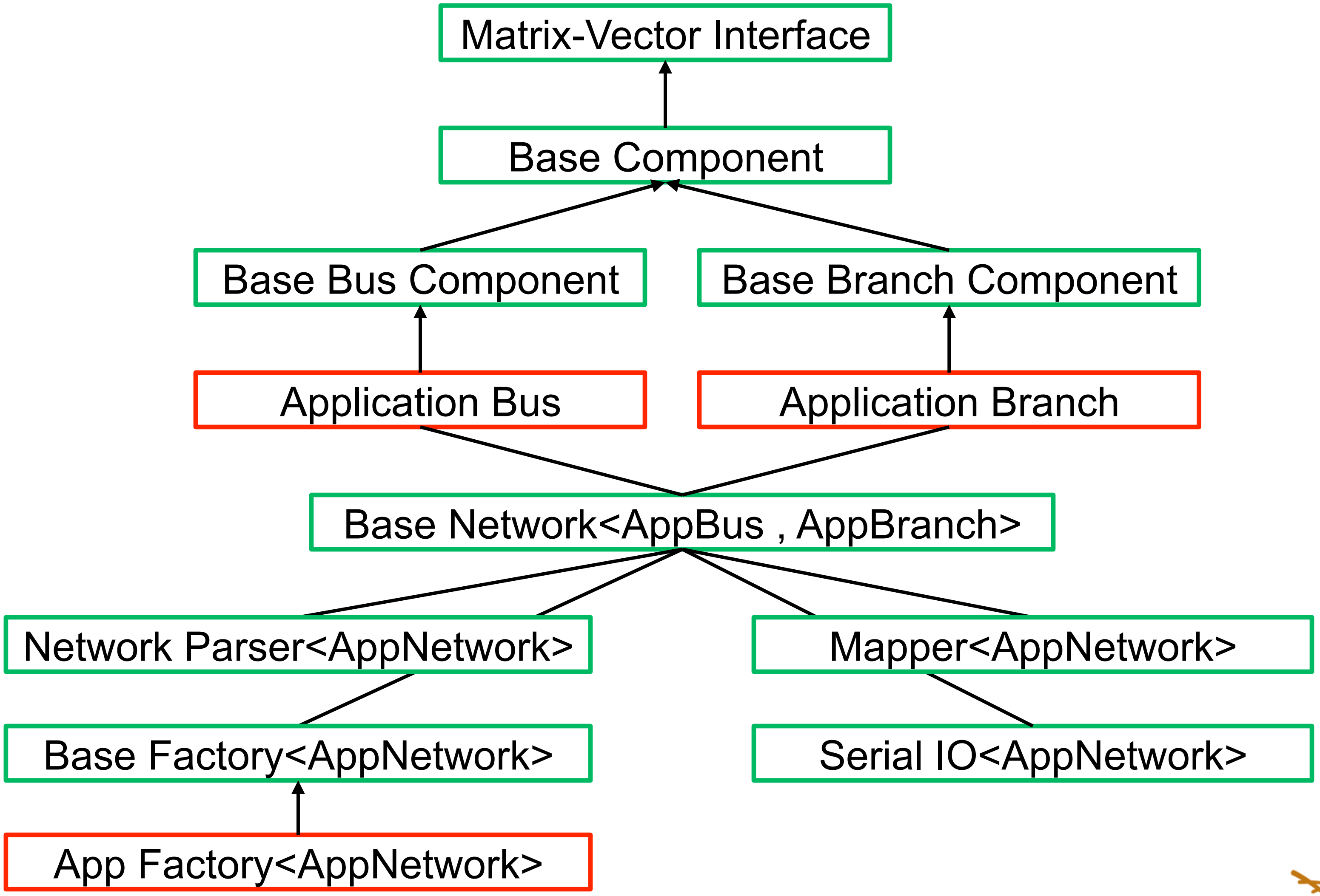
Utilities
- Errors
- Profiling

Core Data Objects

Matrices and Vectors

Power Grid Network

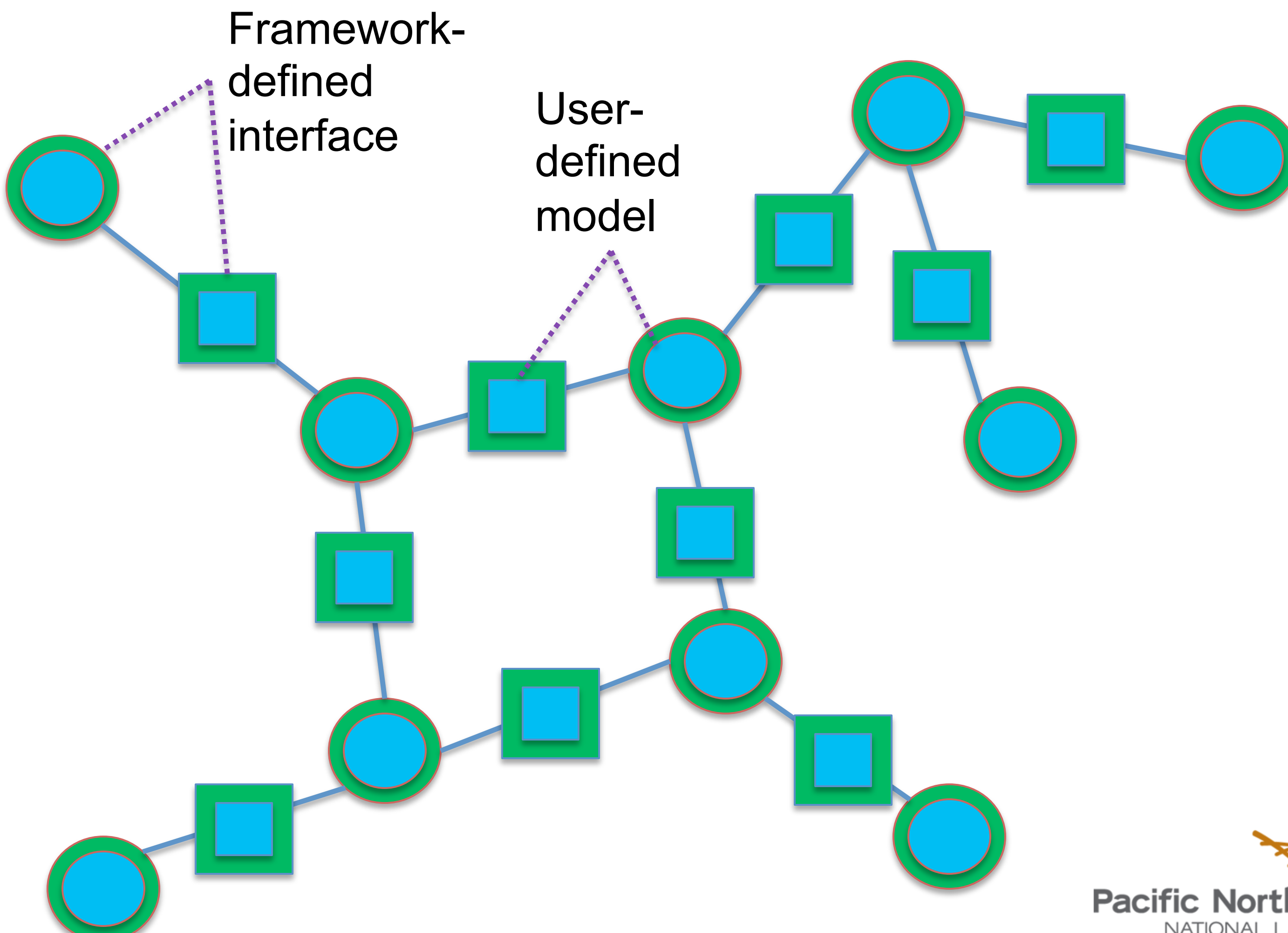
Software Hierarchy



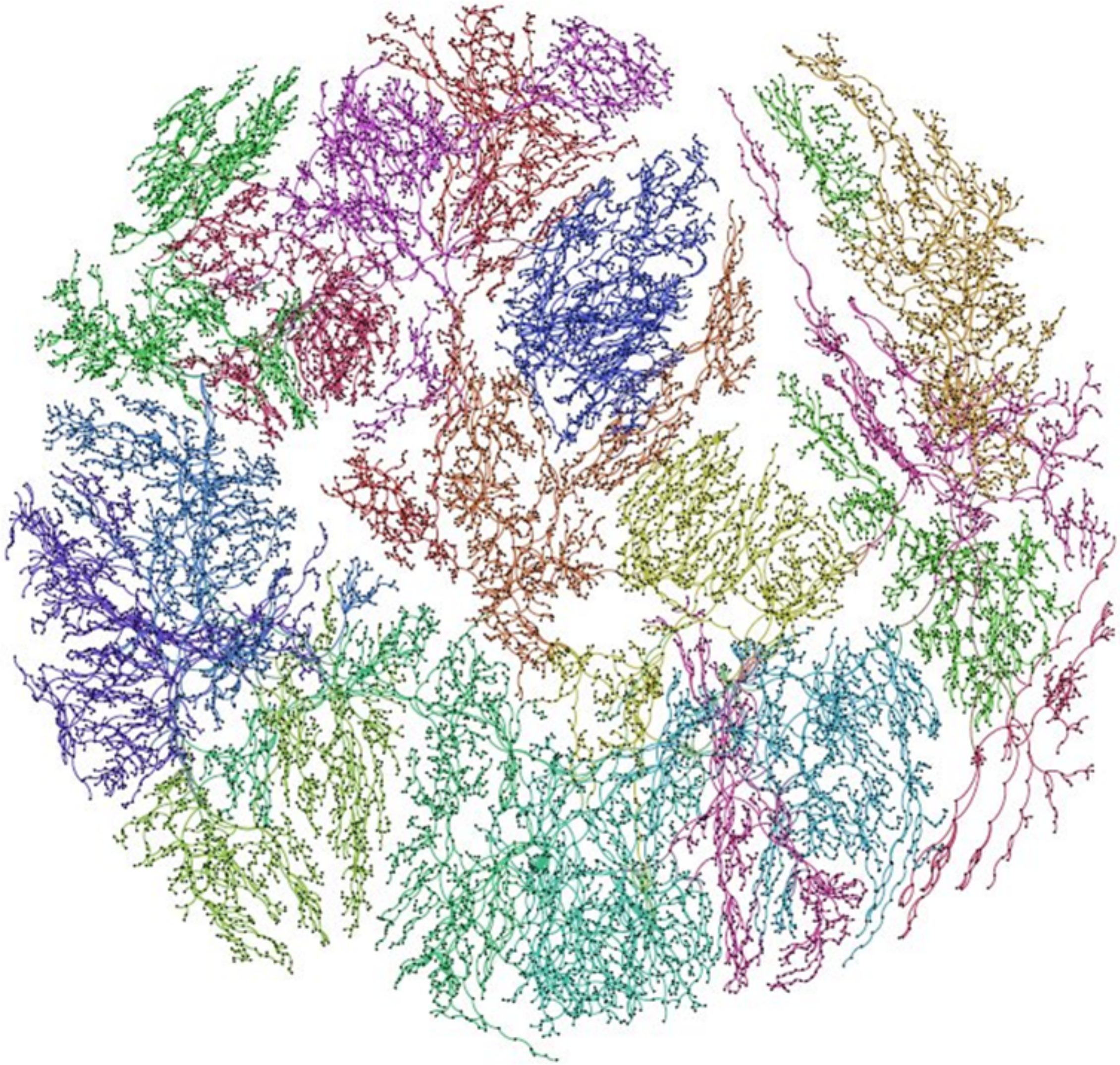
Network Module

- Templated class that takes application-specific bus and branch objects as template arguments
- Manages partitioning of network and exchange of data between processors
- Assigns internal indices that are used by other framework modules

Schematic Diagram of Network Object



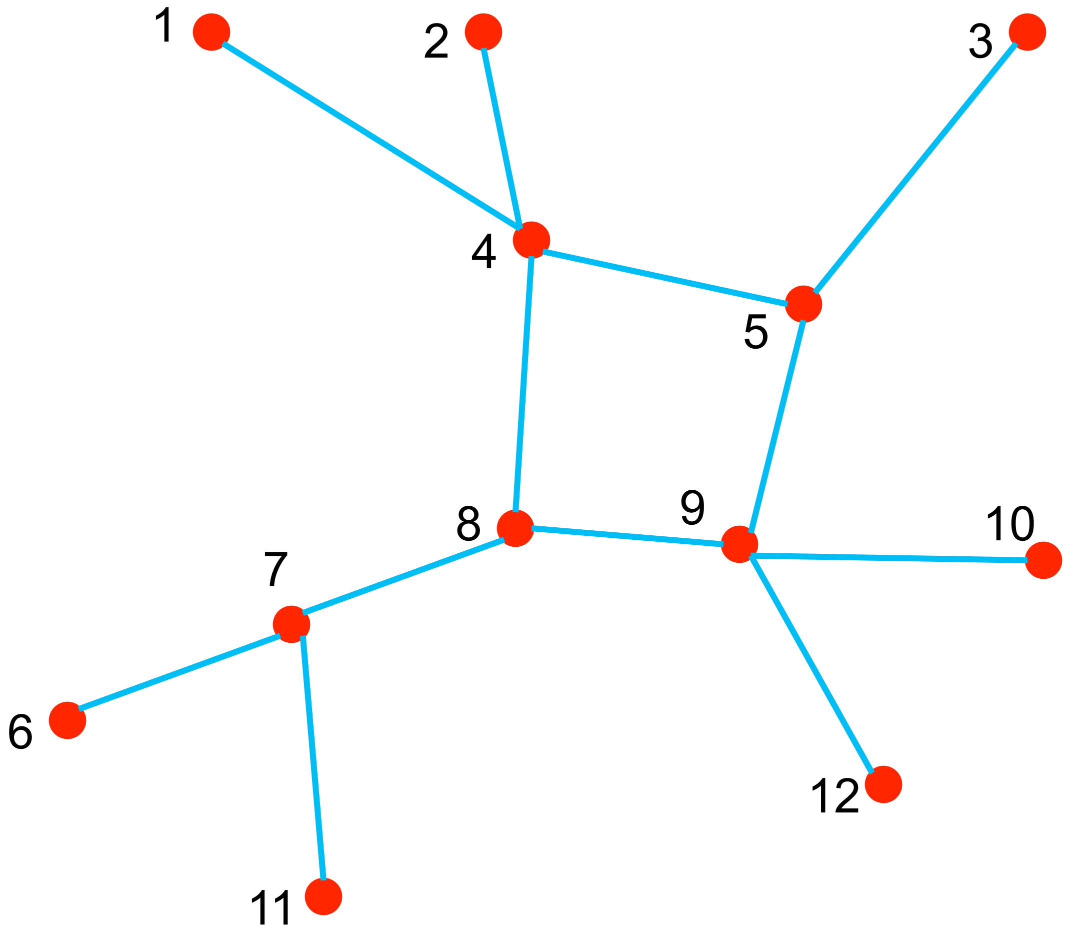
Network Partition



GridPACK™ Mappers

- Construct matrices from contributions from buses and branches
- Manage index transformations between grid location and matrix
- Construct matrices with sensible row partitions based on network partition

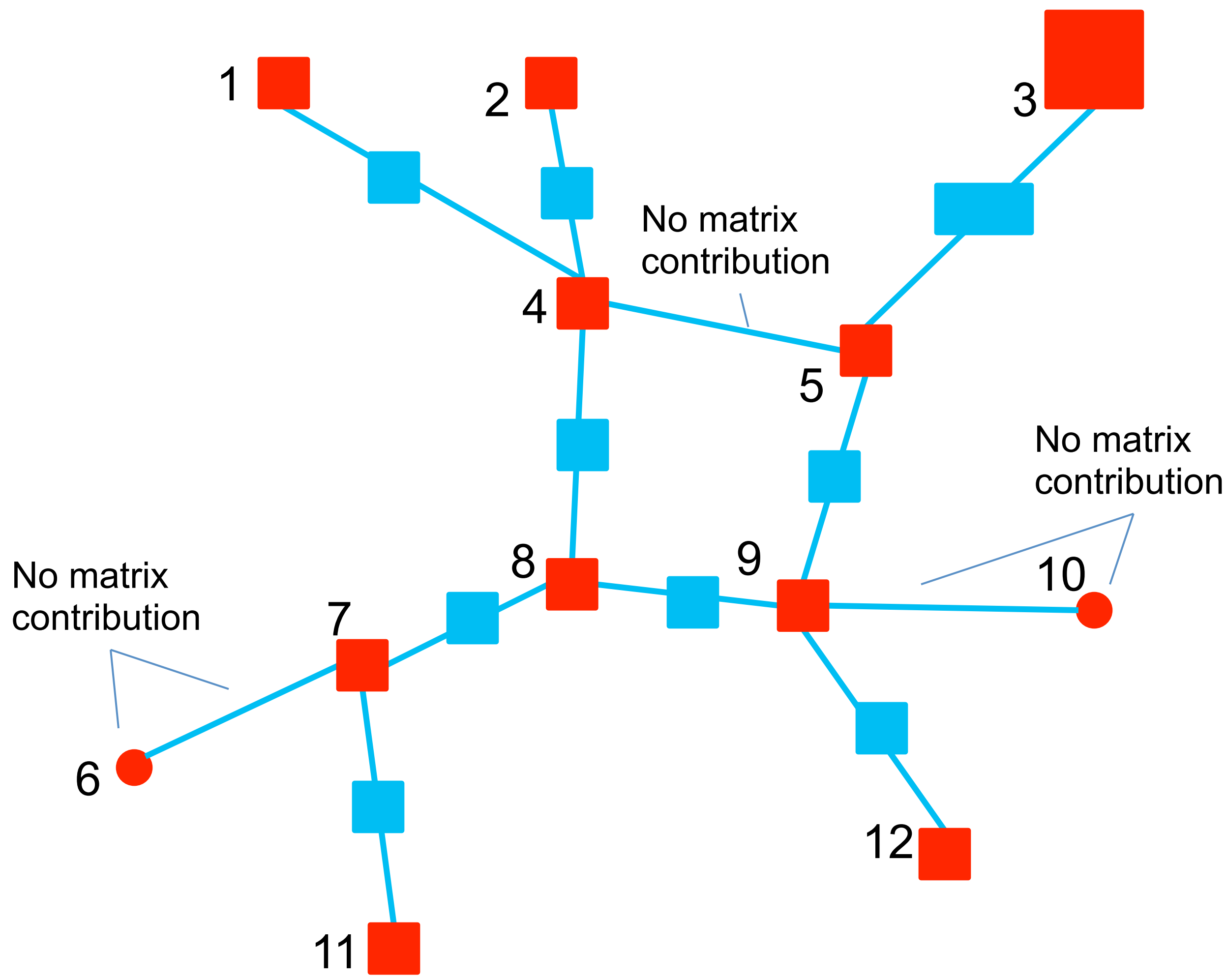
GridPACK™ Mappers: Initial Network



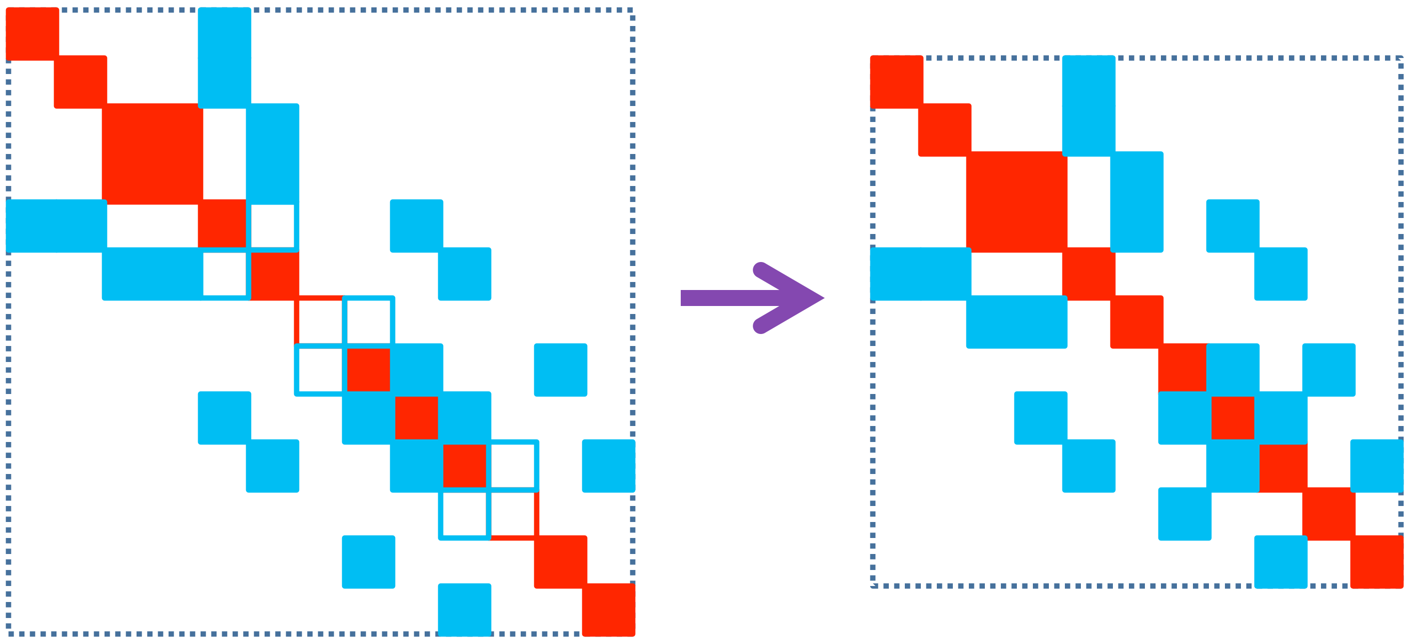
Pacific Northwest
NATIONAL LABORATORY

Proudly Operated by Battelle *Since 1965*

GridPACK™ Mappers: Matrix Contributions



GridPACK™ Mappers: Matrix Generation



Initial Placement

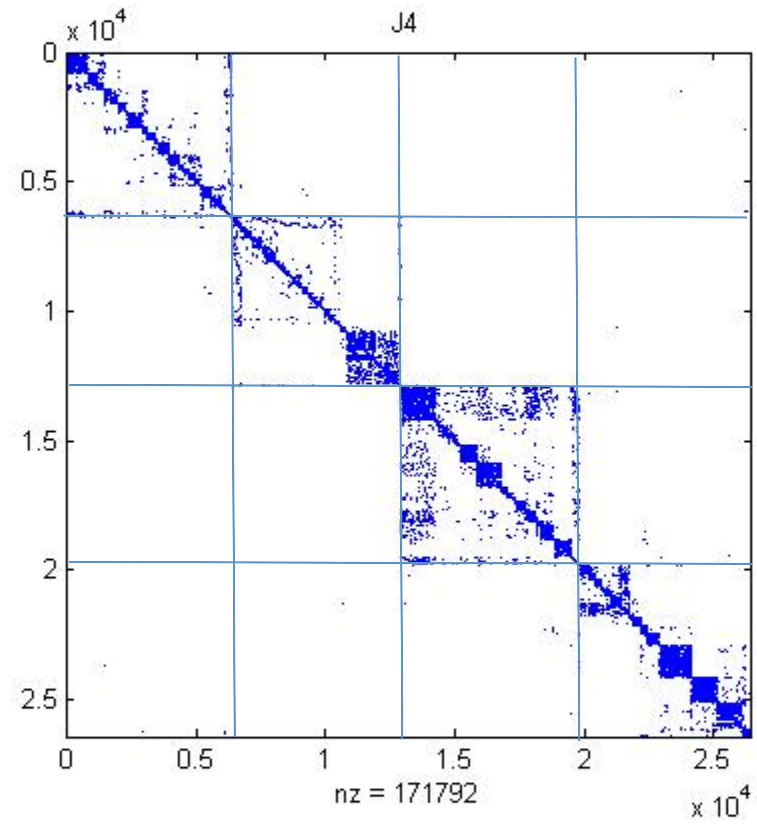
Final Matrix

Matrix Contributions

- Calculations are “local”
  - Only involve sums over neighboring branches or sums over the two buses at the ends of a branch

$$Y_{ii} = -\Sigma_j Y_{ij}$$

- Each bus/branch evaluates its local block, the mapper builds the matrix in a consistent manner



Math Module

- Wraps a parallel solver library (currently PETSc) and provides high level interface for manipulating matrices and vectors
- Provides interface for setting up distributed matrices and vectors
- Supports basic matrix/operations such as matrix-vector multiply, vector norms, matrix transpose, dot products, etc.
- Supports linear and non-linear solvers and preconditioners

Powerflow Application

```
typedef BaseNetwork<PFBus,PFBranch> PFNetwork;
typedef PFFactory<PFNetwork> PFfactory;
Communicator world;
shared_ptr<PFNetwork> network(new PFNetwork(world));

PTI23_parser<PFNetwork> parser(network);
parser.parse(“network.raw”);
network->partition();

PFfactory factory(network);
factory.load();
factory.setComponents();
factory.setExchange();

network->initBusUpdate();
factory.setYBus();
factory.setSBus();
factory.setMode(RHS);
BusVectorMap<PFNetwork> vMap(network);
shared_ptr<Vector> PQ = vMap.mapToVector();

factory.setMode(Jacobian);
FullMatrixMap<PFNetwork> jMap(network);
shared_ptr<Matrix> J = jMap.mapToMatrix();
shared_ptr<Vector> X(PQ->clone());
```

Read in network from external file and partition network

Initialize network components

Evaluate matrix components and create right hand side vector

Create Jacobian matrix

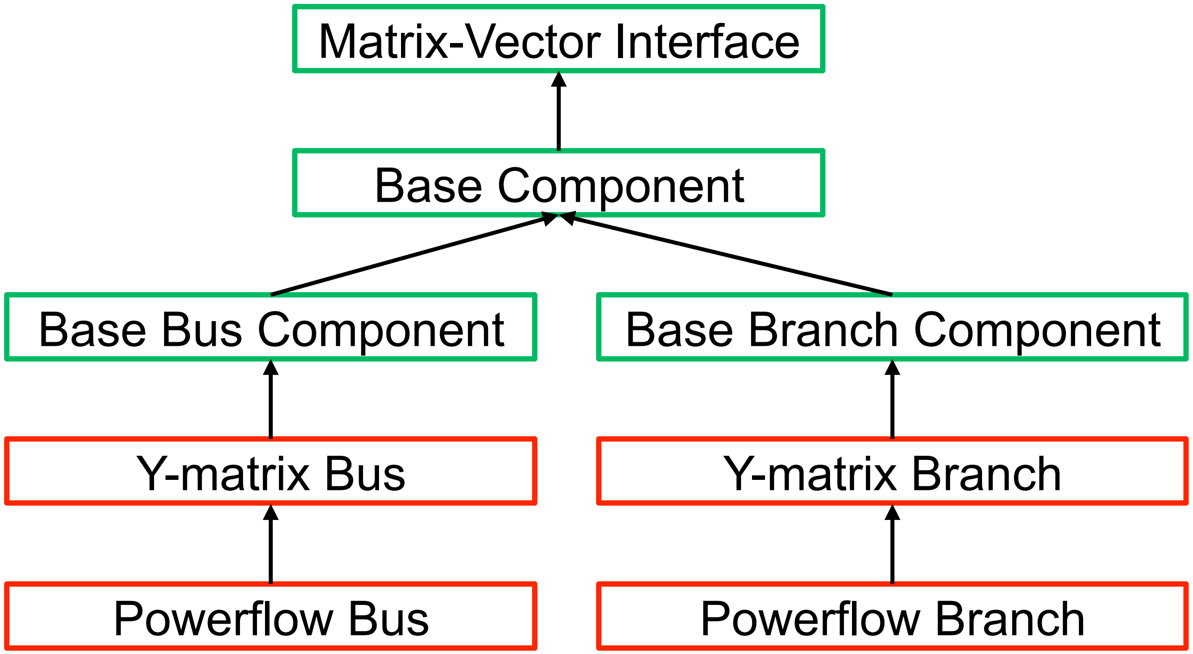
Powerflow Application

```
double tolerance = 1.0e-6;
int max_iteration = 20;
ComplexType tol;
LinearSolver solver(*J);

int iter = 0;

solver.solve(*PQ, *X);
tol = PQ->normInfinity();

while (real(tol) > tolerance && iter < max_iteration) {
  factory.setMode(RHS);
  vMap.mapToBus(X);
  network->updateBuses();
  vMap.mapToVector(PQ);
  factory.setMode(Jacobian);
  jMap.mapToMatrix(J);
  solver.solve(*PQ,*X);
  tol = PQ->normInfinity();
  iter++;
}
```

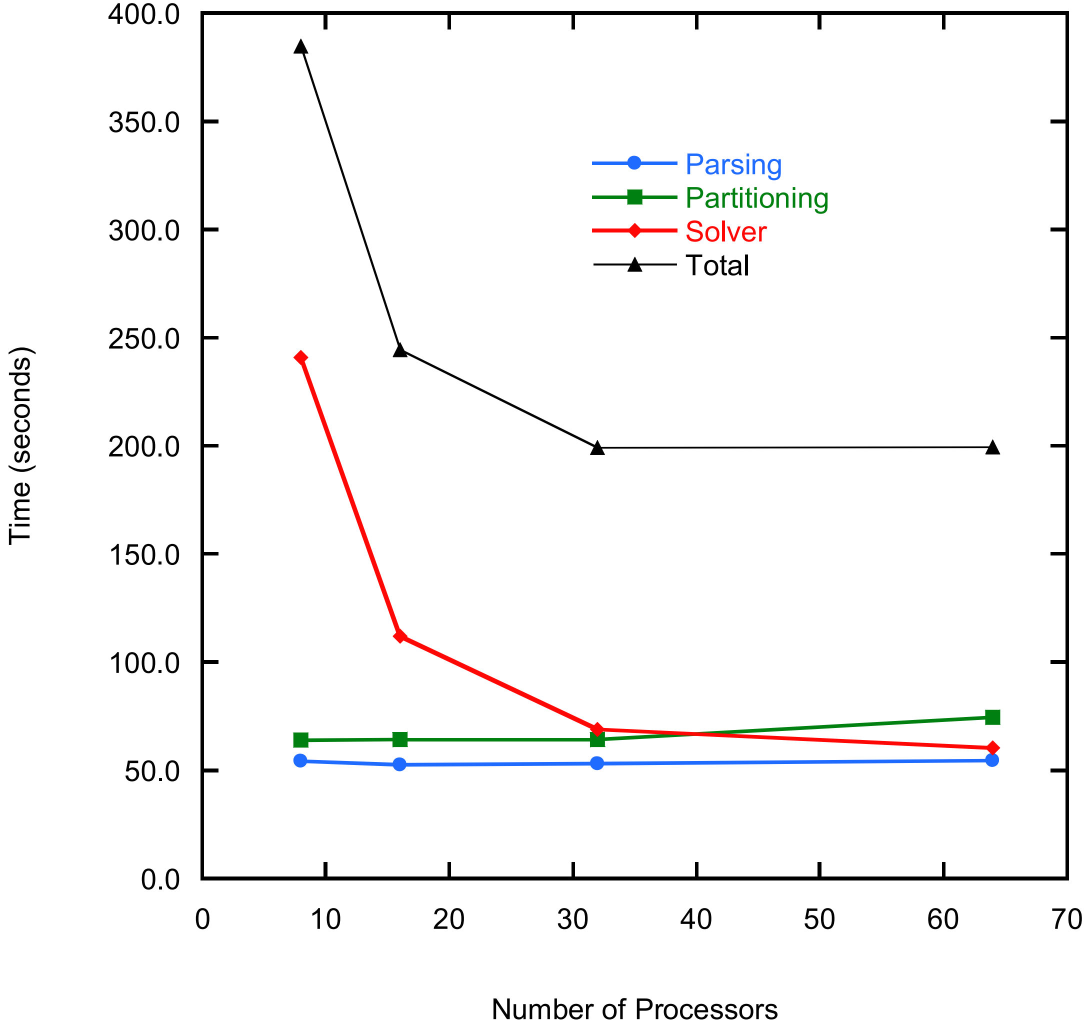
Create solver and perform initial solve

Execute Newton-Raphson iterative loop

Component Hierarchy



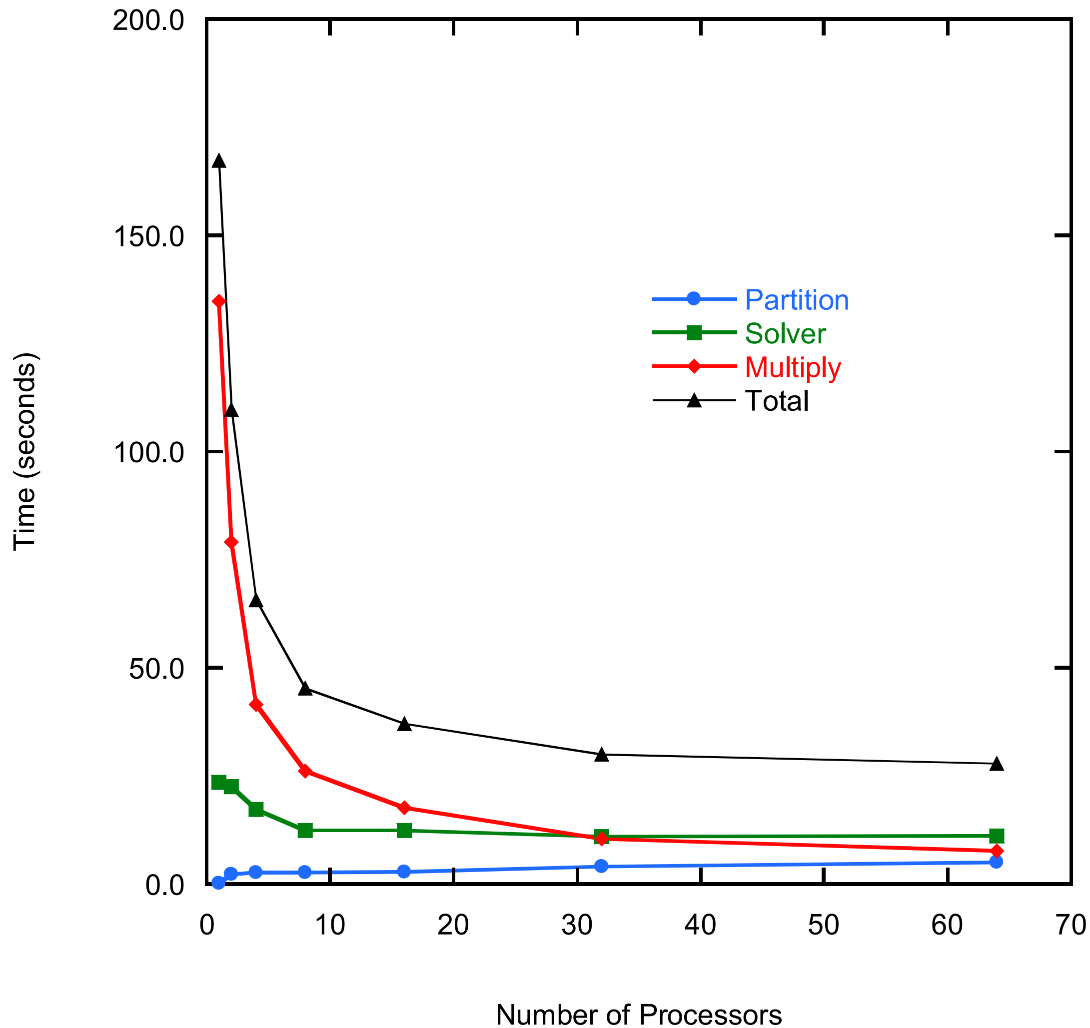
Powerflow Scaling for Artificial 777646 Bus Network



Pacific Northwest
NATIONAL LABORATORY

Proudly Operated by Battelle *Since 1965*

Dynamic Simulation



Simulation of 16351 bus WECC network

Current Activities

- Development of object-oriented Fortran 2003 interface
- Development of more generalized matrix-vector interface to support applications where dependent and independent variables are associated with both buses and branches (not just buses)
- Investigating new methods for distributing data to the network (distributed hashing algorithms)

Conclusions

- A software framework for developing parallel power grid applications has been developed
- Several different types of power grid applications have been developed using the framework. These applications demonstrate parallel speedup
- GridPACK™ is available for download at https://gridpack.org
- Contact bruce.palmer@pnnl.gov

Acknowledgments

- This work is supported by the U.S. Department of Energy (DOE) through its Advanced Grid Modeling Program.
- Computing resources were provided by Pacific Northwest National Laboratory through its PNNL Institutional Computing program

- GridPACK™ is available for download at https://gridpack.org