Progress on NuShellX

Aiming for Petascale

NSCL

SciDAC
Scientific Discovery through Advanced Computing

NuShellX Overview

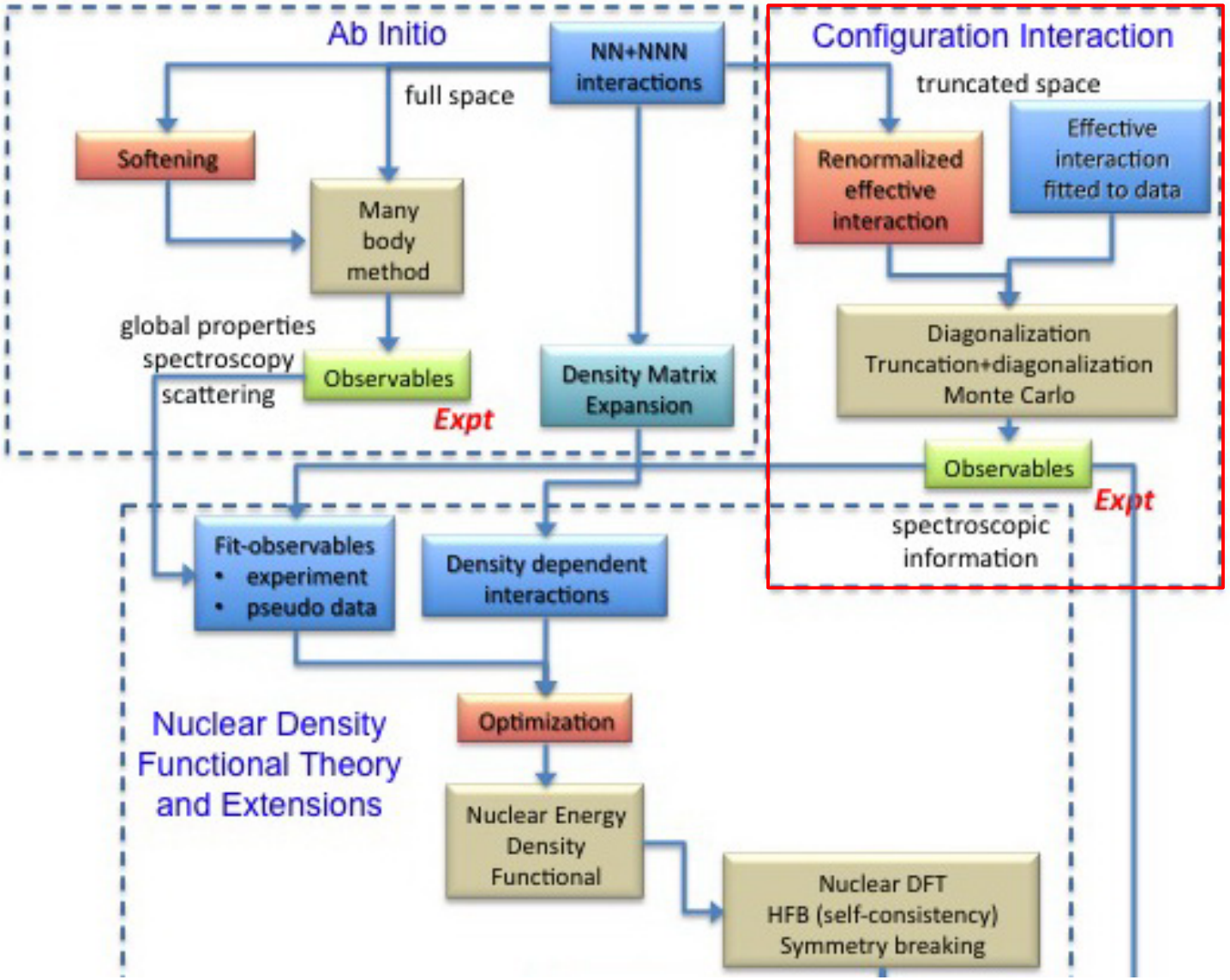
- Is a CI code.
- Calculates matrix elements on the fly.
- Descendant of the OXBASH and NuShell codes with influences from the Antoine and Nathan codes.
- Uses a J-scheme. Starts with good-J proton and neutron bases. Good-J pn basis generated from vector coupling:

$$| \, [(J_p, \alpha_p) \otimes (J_n, \alpha_n)] J >$$

NSCL

SciDAC Scientific Discovery through Advanced Computing

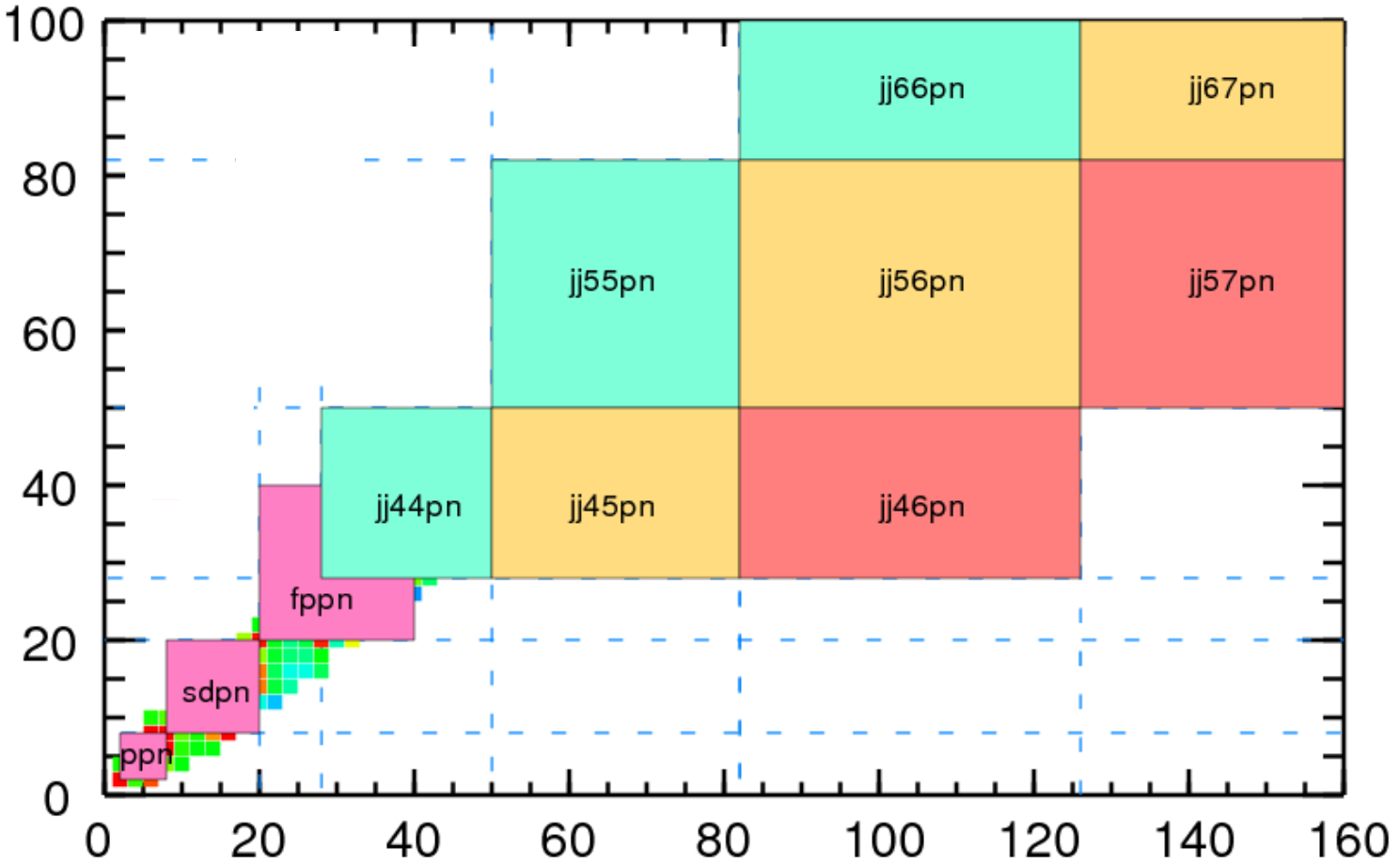
CI Code



NSCL

SciDAC
Scientific Discovery through Advanced Computing

Model Spaces



NSCL

SciDAC
Scientific Discovery through Advanced Computing

Year 5 Goals

- Understand the scalability barriers in NuShellX to enable the most effective use of Graphic Processing Units (GPUs) and leadership-class machines.
- We decided to pursue leadership-class machines for now.
- Thus, we implemented an MPI/OpenMP hybrid code.

NSCL

SciDAC
Scientific Discovery through Advanced Computing

Evolution of the Code

- The original NuShellX code is implemented using OpenMP only.
- The OpenMP-only code doesn't scale well beyond 16 to 24 cores.
- An MPI augmentation of the existing code was decided upon.

Development Activity

- Improved installation process for people building from sources on Unix systems.
- Ported to NERSC environment.
- Worked with NuShellX creator, Bill Rae, to restructure the code to be better suited to MPI.
- Created a skeleton code outlining MPI communications.
- Merged Bill Rae's work into skeleton code to create a MPI/OpenMP hybrid code.

NSCL

SciDAC
Scientific Discovery through Advanced Computing

OpenMP Details

- Lanczos iteration used to converge upon eigenenergies.
- At each iteration, matrix operations performed for pp, nn, and pn spaces.
- Partitioned into blocks of submatrices.
- One row of submatrices is processed at a time.
- Each OpenMP thread works on a submatrix.

NSCL

SciDAC
Scientific Discovery through Advanced Computing

MPI Details

- Master-worker configuration.
  - Master sends a row of submatrices to a worker when that worker is available.
  - Master processes a row of submatrices when all other workers are occupied.
  - Master receives results from workers as they become available.
- Worker OpenMP loop doles out individual submatrices from a received row to the OpenMP threads.

Scaling Overview

- MPI/OpenMP hybrid code was tested up to 128 cores (using 8 cores per node) on local HPC system.
- MPI/OpenMP hybrid code shows improved performance compared to OpenMP-only code.
- MPI/OpenMP hybrid code still has room for improvement; we are only now beginning to wring additional scaling from it. More on that later.

NSCL

SciDAC
Scientific Discovery through Advanced Computing

Benchmarks

Number of Cores	Times (s)
1	6052
8	772
16	609
32	354
64	203
128	162

^{48}Cr J=6

About a factor of 1.7 speedup per doubling of # of cores.

NSCL

SciDAC
Scientific Discovery through Advanced Computing

How to Improve Scalability

- Reduce MPI payloads. (Some unnecessary data may be getting transferred.)
- Move from synchronous communications to asynchronous communications.
- Experiment with MPI one-sided communications.
- Dedicate communication handler threads on master process.
- Use distributed master processes and have them talk to a dedicated master-master.

NSCL

SciDAC
Scientific Discovery through Advanced Computing

Conclusions

- MPI/OpenMP hybrid code has been produced and works.
- A scaling factor of about 1.7 per doubling of cores has been achieved.
- We are exploring ways to achieve a higher scaling factor through various techniques.

Questions?

NSCL

SciDAC
Scientific Discovery through Advanced Computing

NuShellX Summary

- We met our Year 5 goals of:
  - analyzing scalability constraints on the code,
  - evaluating options for reaching the petascale level,
  - and producing a working MPI/OMP hybrid code.
- The hybrid code has been shown to scale to a much larger number of cores than the original OpenMP-only code.
- We have a number of goals for further improvements to the code. These will be discussed during the presentation.

NSCL

SciDAC
Scientific Discovery through Advanced Computing