GPU accelerated *ab initio* nuclear structure calculations on ORNL Summit

TRIUMF-Helmholtz Workshop on Scientific Computing

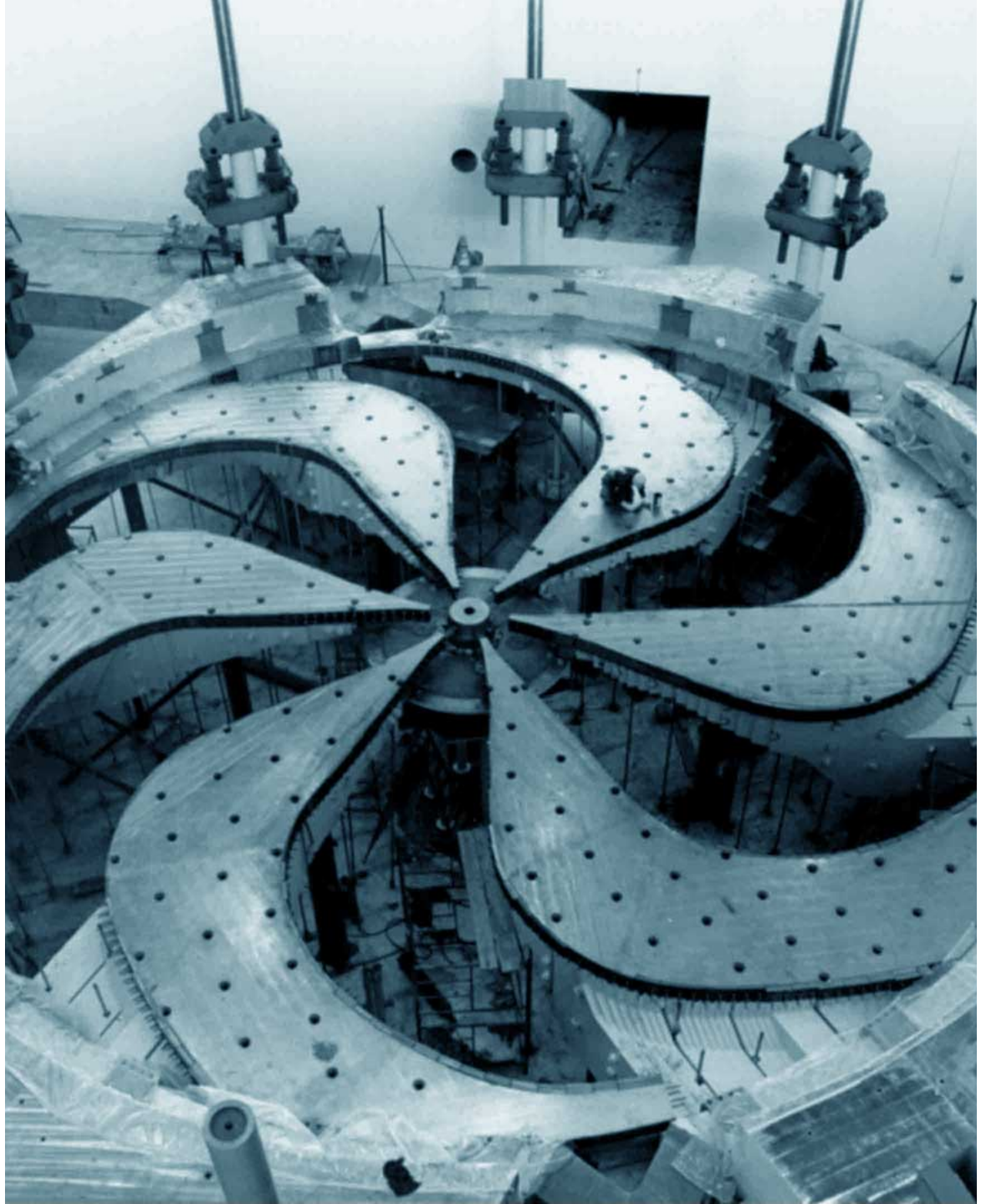
DESY, Hamburg, Germany, 16-17 September, 2019

Petr Navratil

TRIUMF

Collaborator: Kostas Kravvaris (LLNL)

2019-09-16

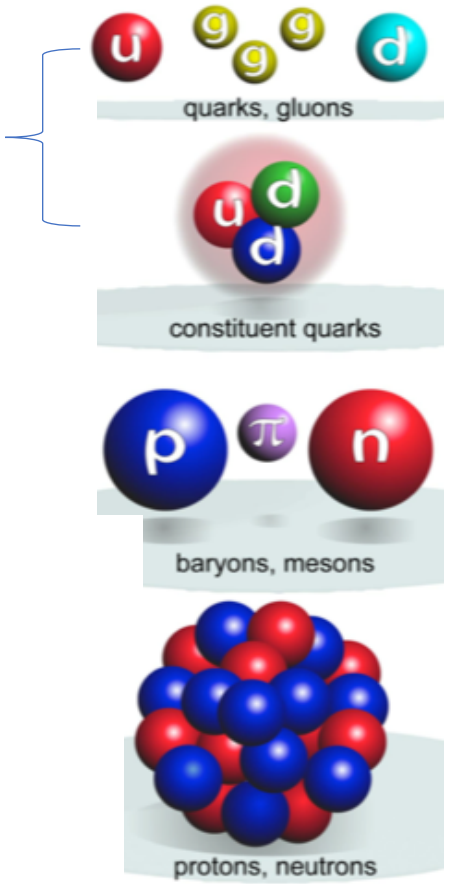


Outline

- *Ab initio* calculations in nuclear physics
- Implementation of large-basis sparse matrix diagonalization algorithm
- GPU acceleration of matrix vector multiplication on ORNL Summit and LLNL Lassen
- Example of an application to a nuclear reaction relevant for astrophysics

First principles or *ab initio* nuclear theory

Quantum Chromodynamics (QCD)



Genuine *Ab Initio*

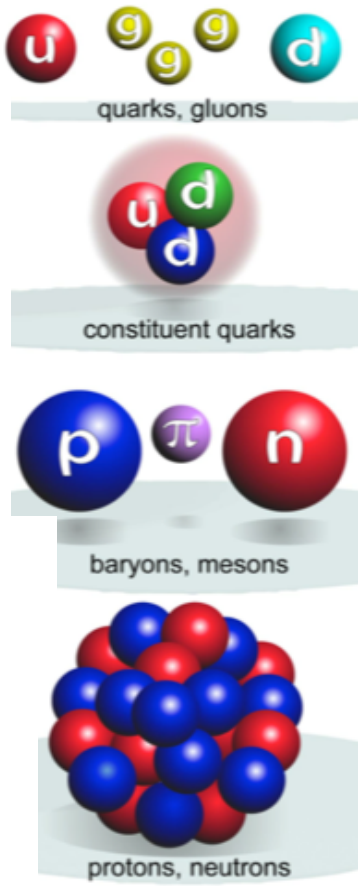
First principles or *ab initio* nuclear theory – what we do at present

Quantum Chromodynamics (QCD)

Chiral Effective Field Theory (parameters fitted to NN data)

Current *ab anitio* nuclear theory

$$H\,\Psi^{(A)} = E\,\Psi^{(A)}$$

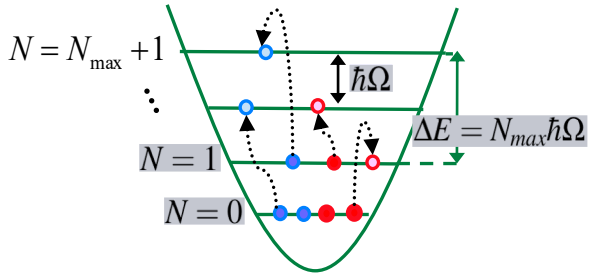


- *Ab initio*
  - ✧ Degrees of freedom: Nucleons
  - ✧ All nucleons are active
  - ✧ Exact Pauli principle
  - ✧ Realistic inter-nucleon interactions
    - ✧ Accurate description of NN (and 3N) data
  - ✧ Controllable approximations

Conceptually simplest *ab initio* method: No-Core Shell Model (NCSM)

NCSM

- Basis expansion method
  - Harmonic oscillator (HO) basis truncated in a particular way ($N_{\max}$)
  - Why HO basis?
    - Lowest filled HO shells match magic numbers of light nuclei (2, 8, 20 – ^{4}He, ^{16}O, ^{40}Ca)
    - Equivalent description in relative-coordinate and Slater determinant basis
- Short- and medium range correlations
- Bound-states, narrow resonances



$$(A)\ \Psi^{A} = \sum_{N=0}^{N_{\max}} \sum_{i} c_{Ni}\, \Phi^{HO}_{Ni}(\vec{\eta}_1, \vec{\eta}_2, \ldots, \vec{\eta}_{A-1})$$

$$(A)\ \Psi^{A}_{\mathrm{SD}} = \sum_{N=0}^{N_{\max}} \sum_{j} c^{\mathrm{SD}}_{Nj}\, \Phi^{HO}_{\mathrm{SD}\,Nj}(\vec{r}_1, \vec{r}_2, \ldots, \vec{r}_A) = \Psi^{A}\, \varphi_{000}(\vec{R}_{CM})$$

Progress in Particle and Nuclear Physics 69 (2013) 131–181

Contents lists available at SciVerse ScienceDirect

Progress in Particle and Nuclear Physics

journal homepage: www.elsevier.com/locate/ppnp

ELSEVIER

Review

Ab initio no core shell model

Bruce R. Barrett[a], Petr Navrátil[b], James P. Vary[c,*]

Multi-nucleon states in the Slater Determinant basis

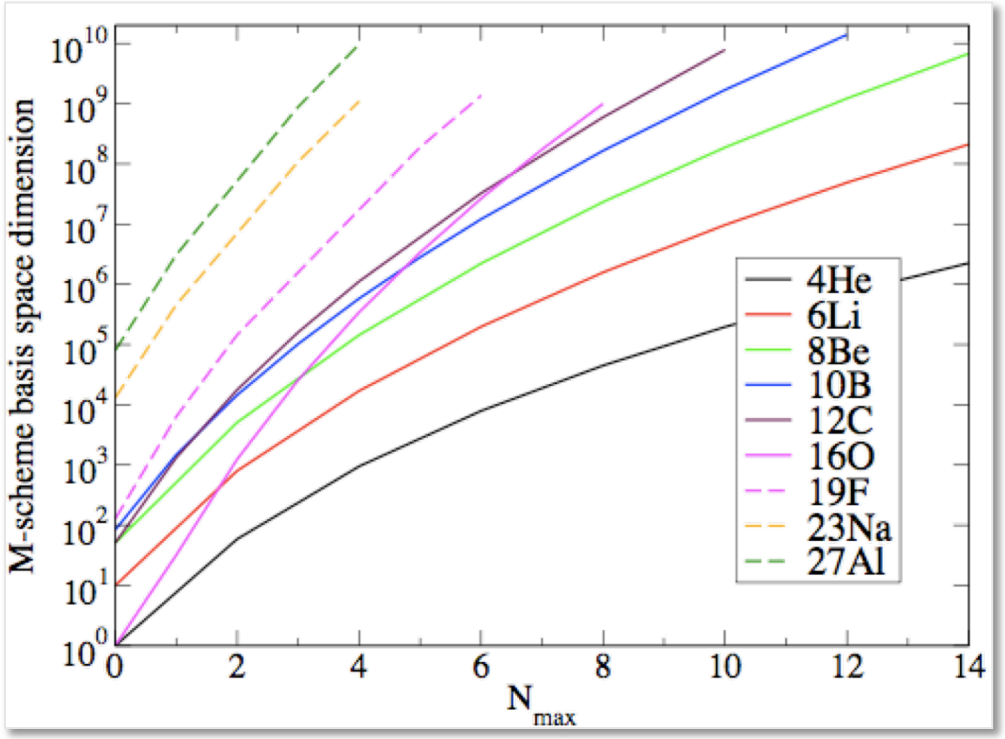
- Many-body HO Slater determinants

$$\left\langle \vec{r}_1\vec{\sigma}_1\vec{\tau}_1,\vec{r}_2\vec{\sigma}_2\vec{\tau}_2,\cdots,\vec{r}_A\vec{\sigma}_A\vec{\tau}_A \middle| a_l^+\cdots a_j^+ a_i^+ \middle| 0 \right\rangle$$

$$= \frac{1}{\sqrt{A!}} \begin{vmatrix} \varphi_i(\vec{r}_1) & \varphi_i(\vec{r}_2) & \cdots & \varphi_i(\vec{r}_A) \\ \varphi_j(\vec{r}_1) & \varphi_j(\vec{r}_2) & & \varphi_j(\vec{r}_A) \\ & \vdots & \ddots & \vdots \\ \varphi_l(\vec{r}_1) & \varphi_l(\vec{r}_2) & \cdots & \varphi_l(\vec{r}_A) \end{vmatrix}$$

$$\varphi_{nljm_j\frac{1}{2}m_t}(\vec{r},\vec{\sigma},\vec{\tau})$$

$$= R_{nl}(r)\left[Y_l(\hat{r})\otimes\chi^S_{\frac{1}{2}}(\vec{\sigma})\right]^j_{m_j}\chi^T_{\frac{1}{2}m_t}(\vec{\tau})$$



- Antisymmetrization is trivial
- Good M, M_T and parity quantum numbers, but not J and T
  - Huge number of basis states

Basis states: occupation representation

- How are Slater determinants (SD) actually represented in a computer program?
  - We are dealing with fermions, so a single-particle state is either occupied or empty, which in computer language translates to either 1's or 0's
  - A very useful approach is a bit representation known as M-scheme
    - HO single-particle states will have good j, m_j

$$a^+_{1,\frac{3}{2},-\frac{1}{2}} a^+_{1,\frac{3}{2},\frac{3}{2}} a^+_{1,\frac{1}{2},\frac{1}{2}} a^+_{0,\frac{1}{2},-\frac{1}{2}} |0\rangle =$$

ℓ, j, j_z

0	1	0	1	0	1	1	0
-3	-1	1	3	-1	1	-1	1
			$0p_{3/2}$				
				$0p_{1/2}$			
						$0s_{1/2}$	

$= 2^1 + 2^3 + 2^5 + 2^6 = 106$

$2m_j$

    - A single integer represents a complicated Slater determinant
  - While the many-body SD states will have good M, they do not have good J. States of good J must be projected and will be a combination of Slater determinants. Same for T and M_T
  - Eigenstates obtained by the diagonalization of the Hamiltonian in SD basis will have a good J and (approximately) good T

Getting the eigenvalues and eigenfunctions: Code NCSD (No-Core shell model in Slater Determinants)

— M-scheme basis:
  — Total M, $M_T=(Z-N)/2$ and parity conserved
- Setup Hamiltonian matrix $\langle \Phi_i | H | \Phi_j \rangle$
- Diagonalize using Lanczos algorithm
  — Bring matrix to tri-diagonal form ($\mathbf{v}_1$, $\mathbf{v}_2$ … orthonormal, H Hermitian)

$$
\begin{aligned}
H\mathbf{v}_1 &= \alpha_1\mathbf{v}_1 + \beta_1\mathbf{v}_2 \\
H\mathbf{v}_2 &= \beta_1\mathbf{v}_1 + \alpha_2\mathbf{v}_2 + \beta_2\mathbf{v}_3 \\
H\mathbf{v}_3 &= \beta_2\mathbf{v}_2 + \alpha_3\mathbf{v}_3 + \beta_3\mathbf{v}_4 \\
H\mathbf{v}_4 &= \beta_3\mathbf{v}_3 + \alpha_4\mathbf{v}_4 + \beta_4\mathbf{v}_5
\end{aligned}
$$

  — n^{th} iteration computes $2n^{th}$ moment
  — Eigenvalues converge to extreme (largest in magnitude) values
  — ~ 150-200 iterations needed for 10 eigenvalues (even for 10^9 states)

  — Eigenvectors obtained with simplified form of Gaussian elimination

Lanczos Diagonalization

Pick a random vector: $|v_1\rangle$

Compute a new vector: $|w'\rangle = H|v_1\rangle$

Store the overlap: $\alpha_1 = \langle v_1|w'\rangle$

Orthogonalize the two: $|w\rangle = |w'\rangle - \langle v_1|w'\rangle|v_1\rangle$

Loop for as many as we can: For $j = 1, \ldots, m$

- Store the norm of the new vector: $\beta_j = \sqrt{\langle w|w\rangle}$
- Normalize the new vector: $|v_j\rangle = |w_{j-1}\rangle/\beta_j$
- Compute new vector: $|w'_j\rangle = H|v_j\rangle$
- Store Overlap: $\alpha_j = \langle w'_j|v_j\rangle$
- Obtain new vector: $|w_j\rangle = |w'_j\rangle - \alpha_j|v_j\rangle - \beta_j|v_{j-1}\rangle$

Diagonalize only small tridiagonal matrix

$$\begin{pmatrix} \alpha_1 & \beta_2 & & & \\ \beta_2 & \alpha_2 & \beta_3 & & \\ & \beta_3 & \alpha_3 & \ddots & \\ & & \ddots & \ddots & \\ & & & \beta_j & \alpha_j \end{pmatrix}$$

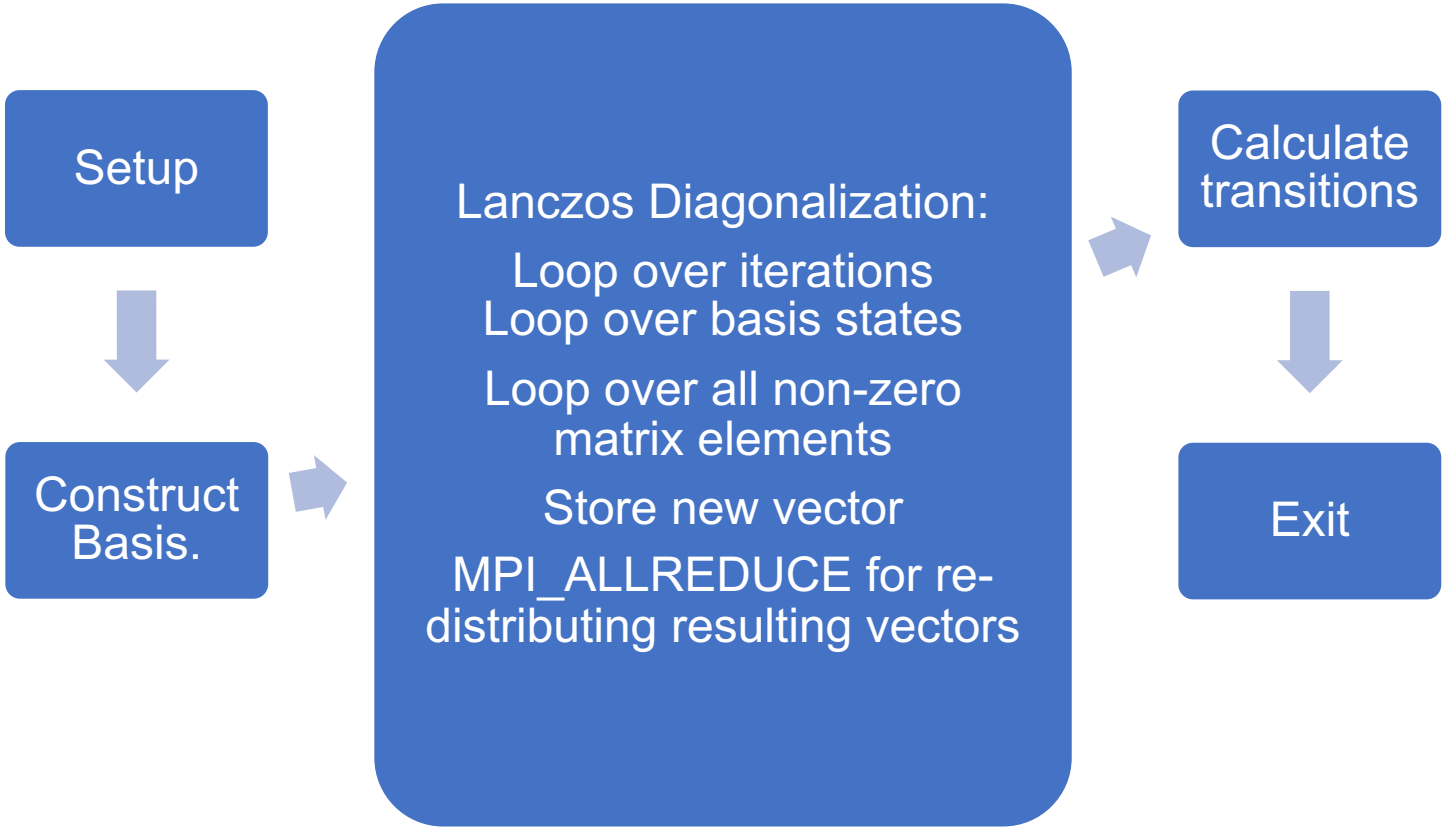
Note: Lanczos vectors are expansions over the many-nucleon basis states (dimension N)

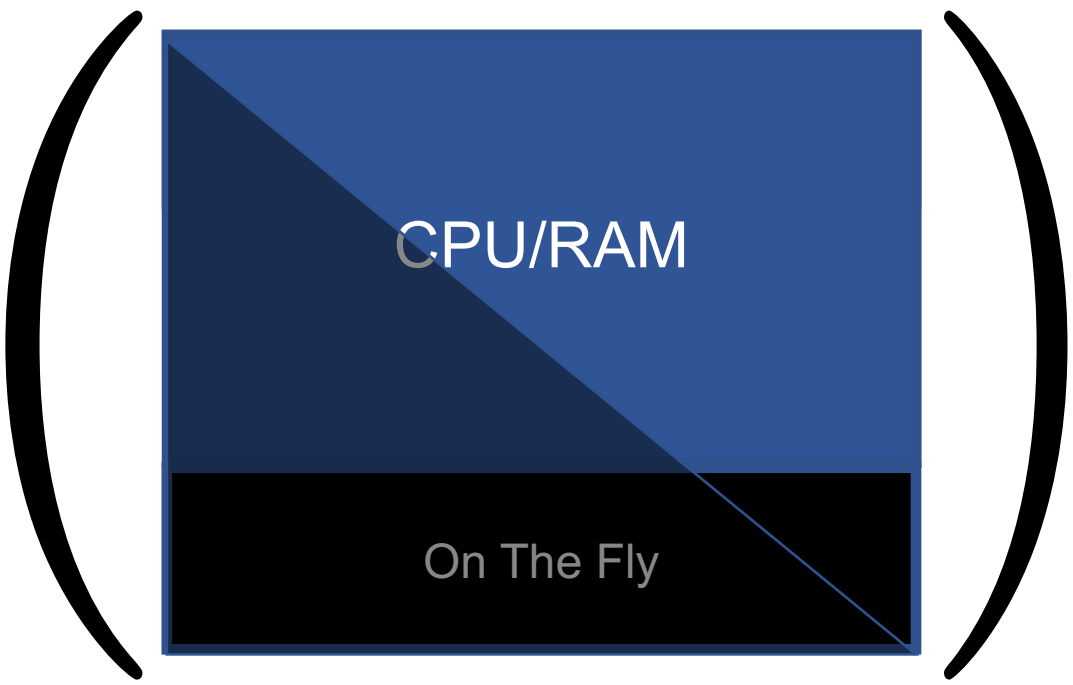
NCSD Main Loop Structure

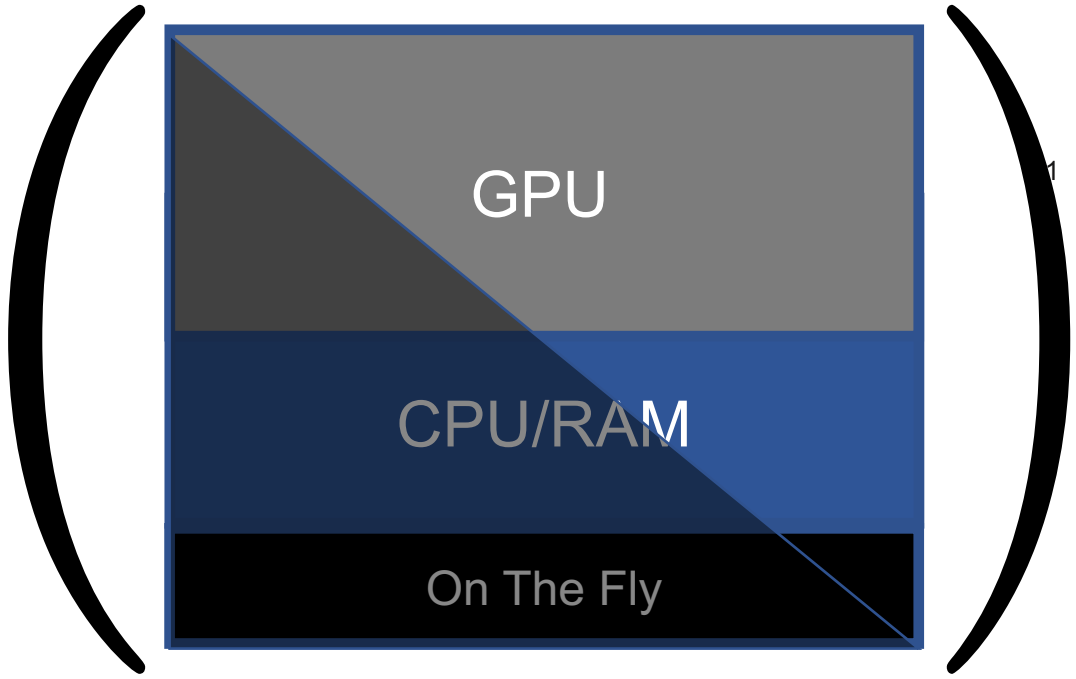
NCSD – Fortran 90 with MPI, OpenMP and CUDA GPU accelerations, capable of running ~12,000 MPI tasks

Lanczosh diagonalization, bit operations, hashing, partial or full storing of non-zero Hamiltonian matrix elements in memory

NN+3N interaction input, can calculate N_{max} basis sequence (0(1),2(3),…N_{max}) in a single run







- We construct the matrix elements in RAM as it is generally more readily available than GPU memory.
- Matrix elements are transferred to GPU and perform the matrix-vector multiplications on the GPU.
- Due to mixed float/double precision for matrix elements/vector elements, no CUBLAS intrinsic function is used.

- Matrix is currently saved in Compressed Sparse Row format. Will change to Compressed Sparse Block in the future for better GPU efficiency.
- Current code does SpMV multiplication; by using SpMM for a block of vectors, GPU usage will further increase.

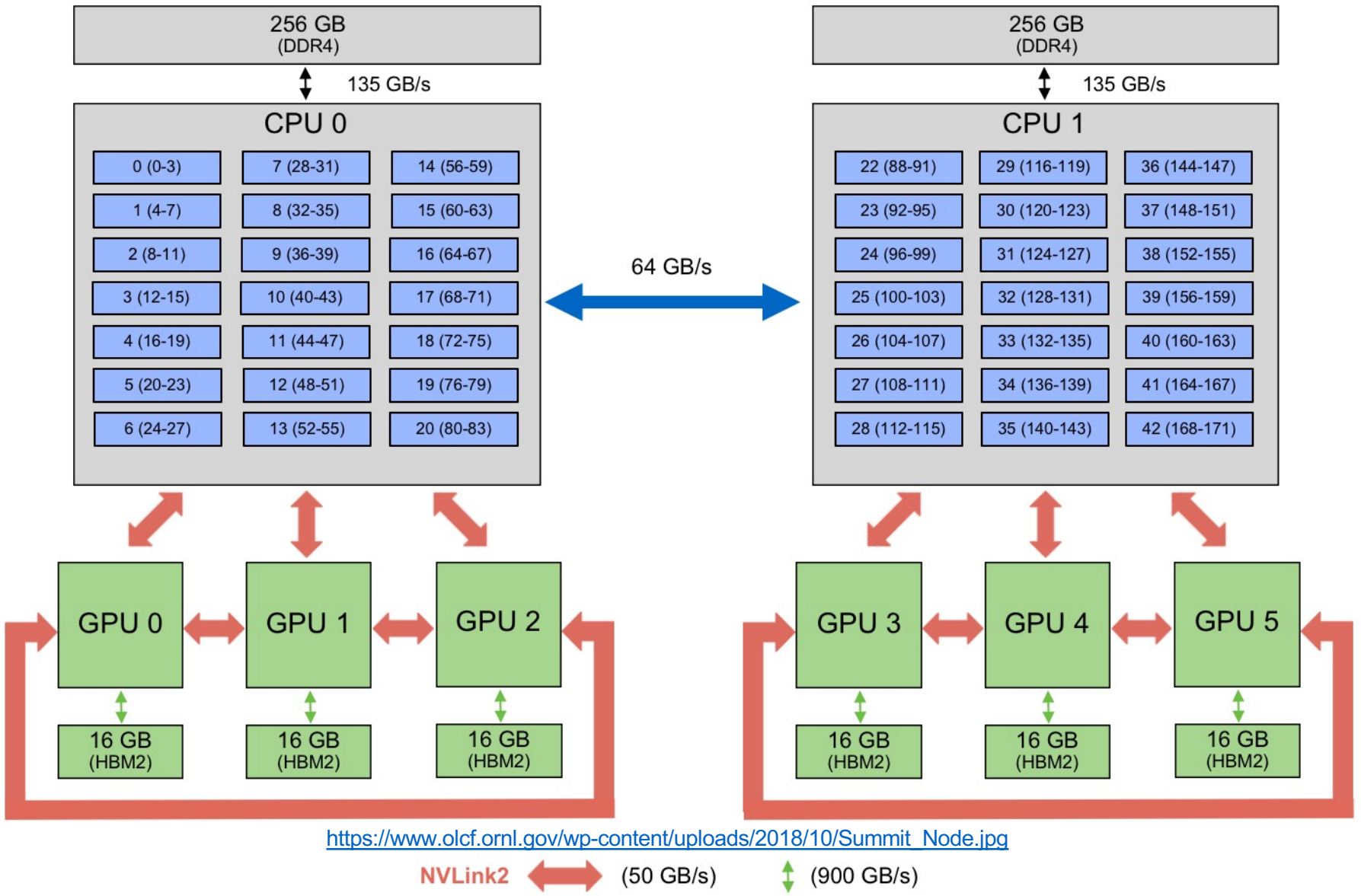
SUMMIT - Oak Ridge National Laboratory's 200 petaflop supercomputer

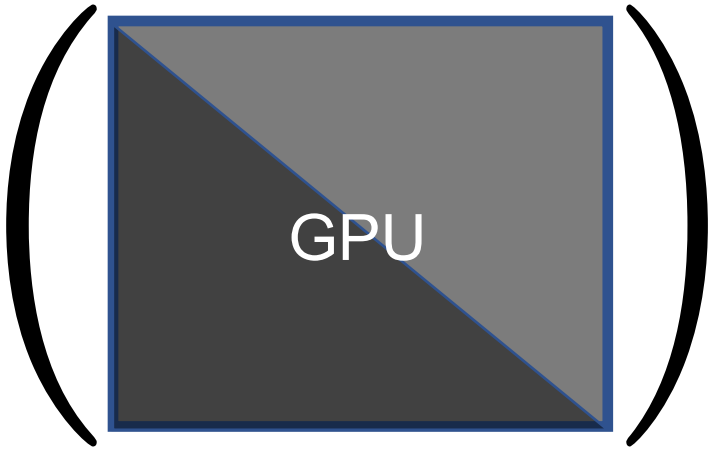
- Specifications
  - **Processor:** IBM POWER9™ (2/node)
  - **GPUs:** 27,648 NVIDIA Volta V100s (6/node)
  - **Nodes:** 4,608
  - **Node Performance:** 42TF
  - **Memory/node:** 512GB DDR4 + 96GB HBM2
  - **NV Memory/node:** 1600GB
  - **Total System Memory:** >10PB DDR4 + HBM + Non-volatile
  - **Interconnect Topology:** Mellanox EDR 100G InfiniBand, Non-blocking Fat Tree
  - **Peak Power Consumption:** 13MW

- INCITE award “Nuclear Structure and Reactions”

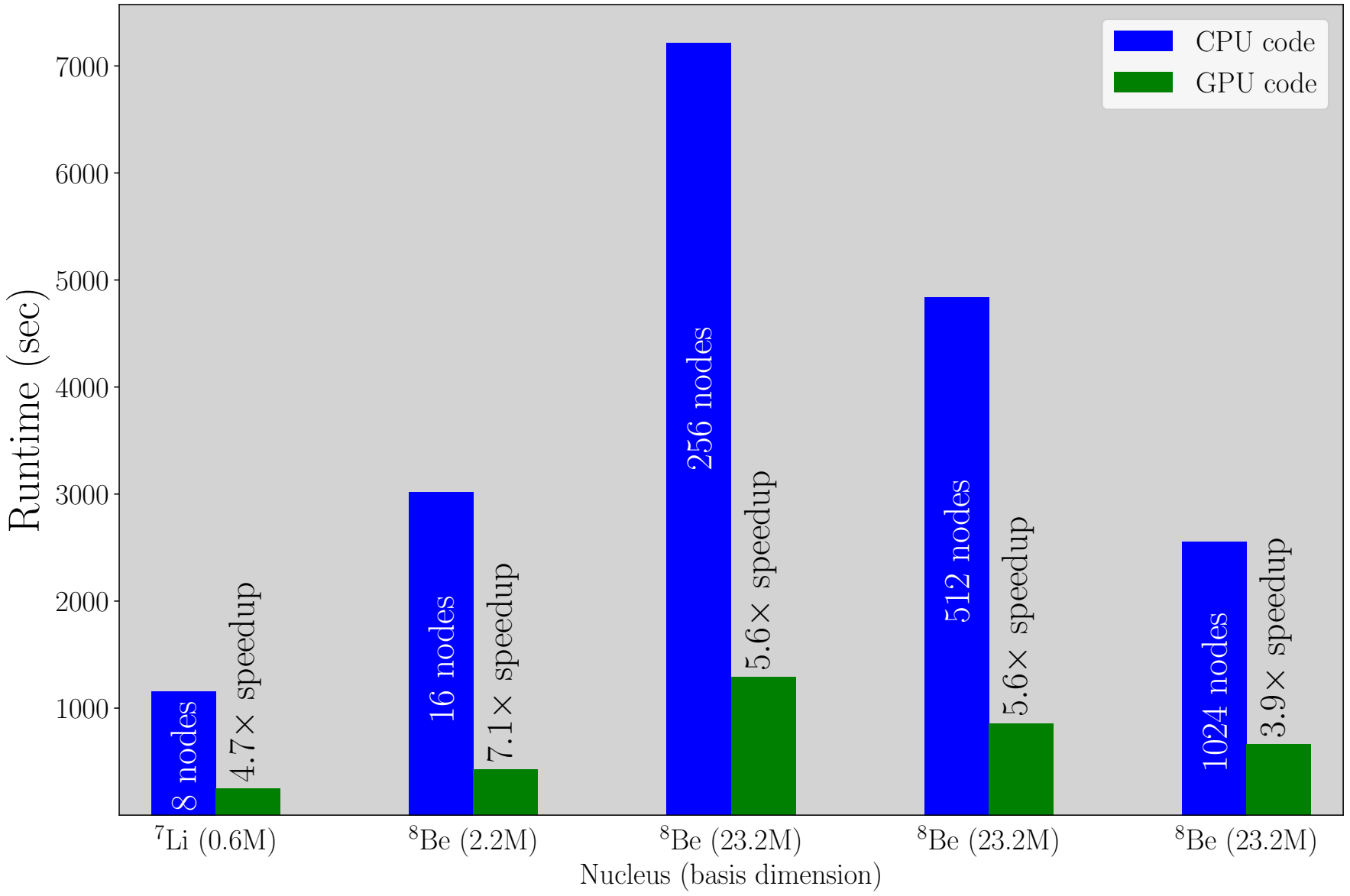
Anatomy of a Summit Node:

- Fat Node allows for more freedom in resource management.
- 16GB of memory/GPU means they can be used for storage of not only the vectors, but also the matrix elements.
- 512GB of RAM/node allow for the entire Hamiltonian Matrix to be stored.
- The 42 hyper-threaded cores accelerate the construction of the matrix.





- Speedup decreases at high node count as the non-GPU accelerated part of the code dominates.
- Memory footprint also decreased by a factor of 2 from more efficient storage of matrix elements.
- SUMMIT has enough nodes so that the entire matrix fits on GPU memory.

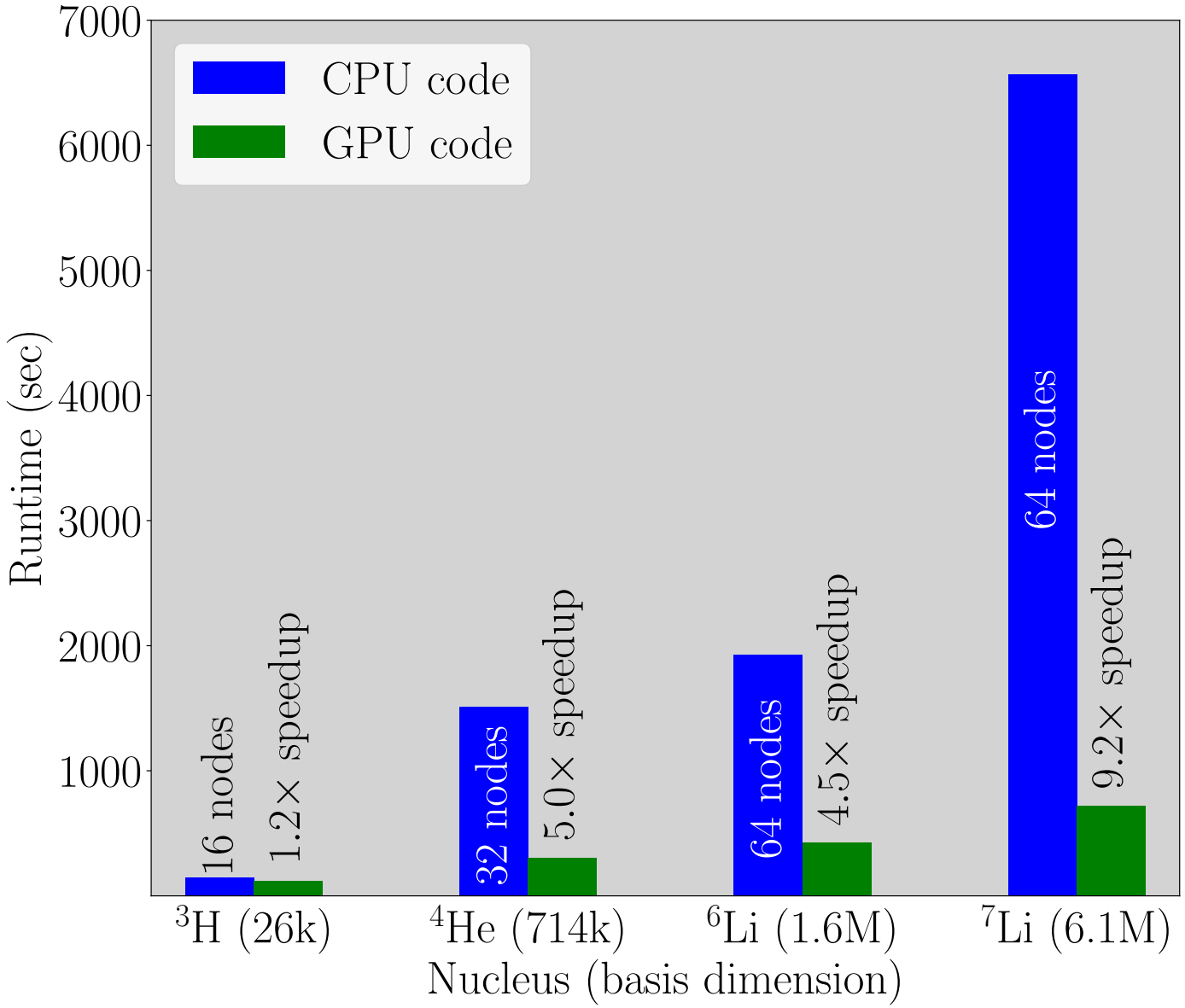


LASSEN machine at LLNL

- 4 GPUs/node
- 256GB RAM
- 756 Nodes total

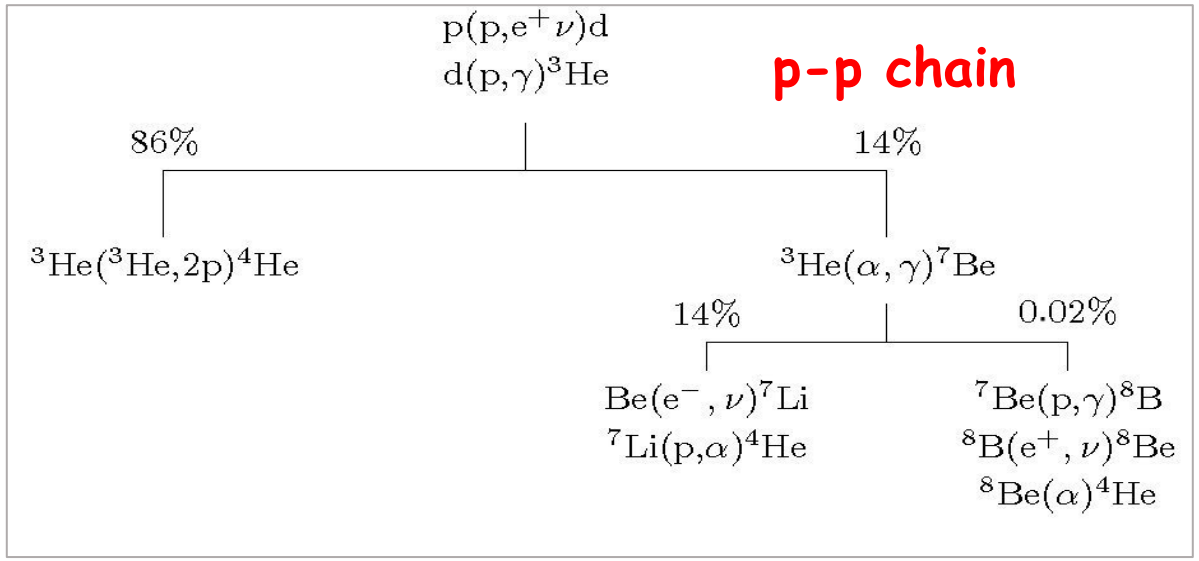
Need different strategy for large calculations!

- Use multiple MPI tasks/GPU.
- One task fully utilizes the GPU and the rest work in pure CPU mode.
- Allows for better distribution of the matrix between RAM/GPU memory.
- Implement MPI window object for sharing memory within a single node, further reducing memory footprint.



Example of an application of the GPU accelerated code: p+^{11}C scattering and ^{11}C(p,γ)^{12}N capture

- ^{11}C(p,γ)^{12}N capture relevant in hot *p-p* chain: Link between pp chain and the CNO cycle - bypass of slow triple alpha capture ^{4}He(αα,γ)^{12}C



$$^{3}He(\alpha,\gamma)^{7}Be(\alpha,\gamma)^{11}C(p,\gamma)^{12}N(p,\gamma)^{13}O(\beta^{+},\nu)^{13}N(p,\gamma)^{14}O$$

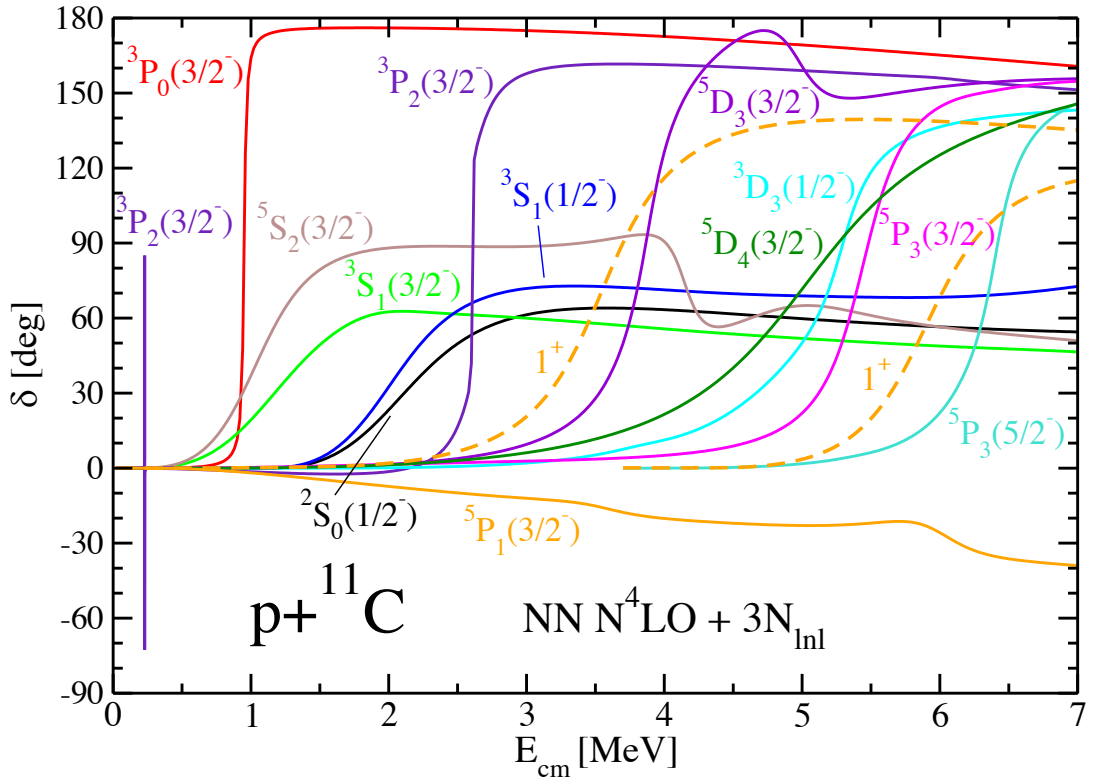
$$^{3}He(\alpha,\gamma)^{7}Be(\alpha,\gamma)^{11}C(p,\gamma)^{12}N(\beta^{+},\nu)^{12}C(p,\gamma)^{13}N(p,\gamma)^{14}O$$

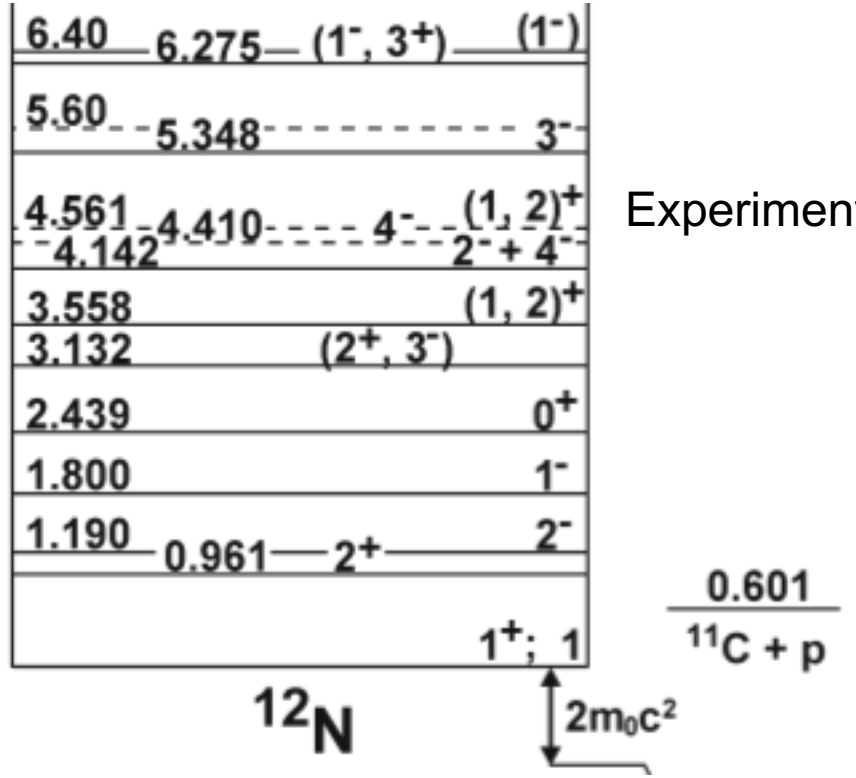
$$^{11}\mathrm{C}(\beta^{+}\ \nu)^{11}\mathrm{B}(p,\alpha)^{8}\mathrm{Be}(^{4}\mathrm{He},^{4}\mathrm{He})$$

Example of an application of the GPU accelerated code: p+^{11}C scattering and ^{11}C(p,γ)^{12}N capture

Measurement of ^{11}C(p,p) resonance scattering planned at TRIUMF TUDA facility

- NCSM with continuum calculations of ^{11}C(p,p) with chiral NN+3N under way
  - ^{11}C and ^{12}N NCSM eigenstates calculated with NCSD on Summit using GPU acceleration
  - 1024 nodes, 6144 MPI tasks with 1 GPU/task and 7 OpenMP threads/task
  - Largest matrix dimensions: 131 million for ^{11}C and 167 million for ^{12}N (~7 hours to get 9 eigenstates)

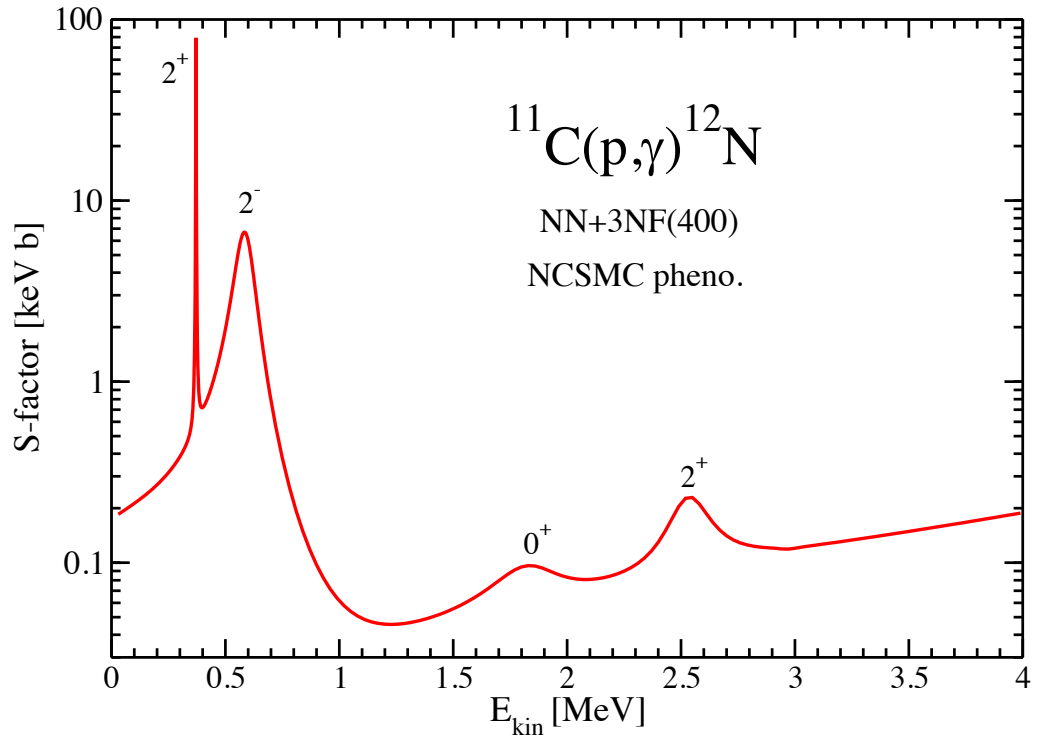




Example of an application of the GPU accelerated code: p+^{11}C scattering and ^{11}C(p,γ)^{12}N capture

Measurement of ^{11}C(p,p) resonance scattering planned at TRIUMF TUDA facility

- NCSM with continuum calculations of ^{11}C(p,p) with chiral NN+3N under way
  - ^{11}C and ^{12}N NCSM eigenstates calculated with NCSD on Summit using GPU acceleration
  - 1024 nodes, 6144 MPI tasks with 1 GPU/task and 7 OpenMP threads/task
  - Largest matrix dimensions: 131 million for ^{11}C and 167 million for ^{12}N (~7 hours to get 9 eigenstates)



NCSMC calculations to be validated by measured cross sections and applied to calculate the ^{11}C(p,γ)^{12}N capture

Conclusions

- *Ab initio* calculations of nuclear structure and reactions becoming feasible beyond the lightest nuclei
  - Make connections between the low-energy QCD, many-body systems, and nuclear astrophysics
- Large-scale computational problems
  - Codes running on massively parallel supercomputers
- Taking advantage of GPU acceleration with a significant speedup demonstrated on ORNL Summit and LLNL Lassen
- Enables nuclear physics applications such as calculations of nuclear reactions relevant for astrophysics

TRIUMF

Thank you!
Merci!
Danke!

Discovery, accelerated