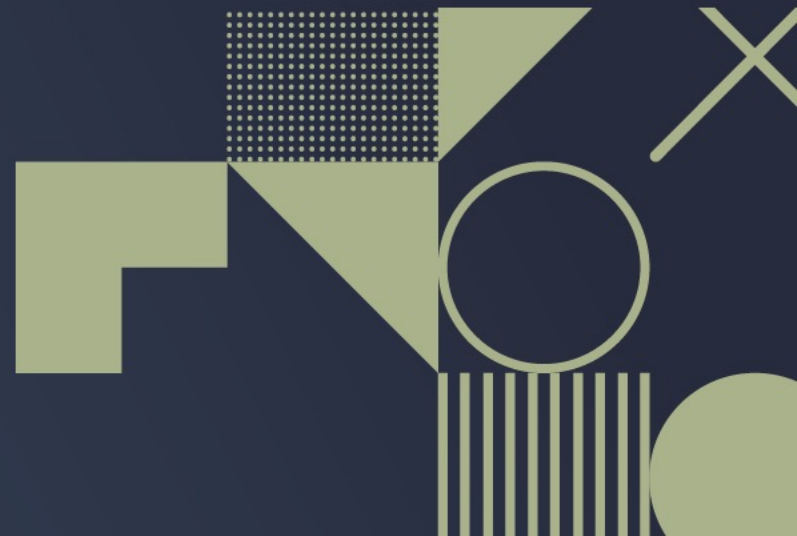
Enabling Large-Scale Correlated Electronic Structure Calculations

Scaling the RI-MP2 Method on Summit



GIUSEPPE M. J. BARCA

EXASCALE COMPUTING PROJECT

Australian National University

STATE

AMES LABORATORY
Creating Materials & Energy Solutions
U.S. DEPARTMENT OF ENERGY

Argonne NATIONAL LABORATORY

Flinders UNIVERSITY
ADELAIDE · SOUTH AUSTRALIA

D. Poole
ISU, Ames Lab

J. Galvez-Vallejo
ISU, Ames Lab

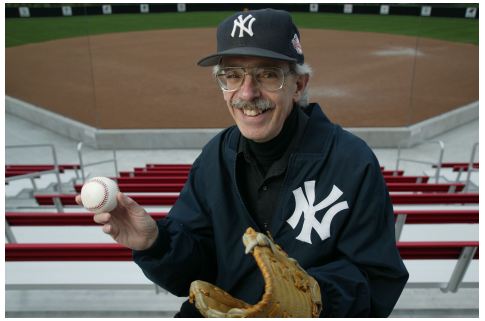
M. Alkan
ISU, Ames Lab

WE GAMESS

R. Stocks
ANU

Dr G. Barca
ANU

Prof A. Rendell
Flinders Univesity

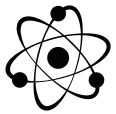


Prof M. Gordon
ISU, Ames Lab

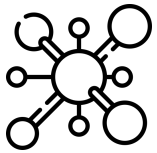
TABLE OF CONTENTS

THE SCALE OF MATTER

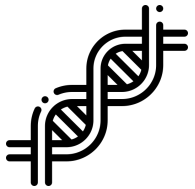
IN ATOMS



ATOM

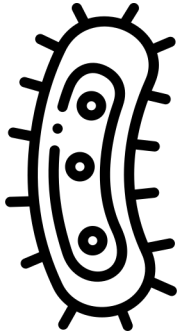


SMALL MOLECULE

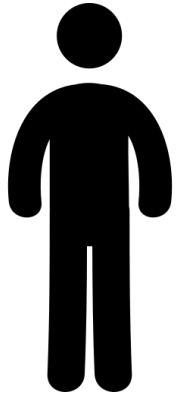


DNA

VIRUS



BACTERIA



HUMAN

CHEMISTRY BIOLOGY MEDICINE

MATTER SIMULATIONS

MATERIAL SCIENCE ENGINEERING

"The underlying physical laws necessary for the mathematical theory of a large part of physics and the whole of chemistry are thus completely known, and the difficulty is only that the exact application of these laws leads to equations much too complicated to be soluble."

P. A. M. Dirac

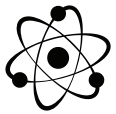
Proc. R. Soc. A, 123:714, 1929

$$\mathcal{H}\,\Psi = E\,\Psi$$

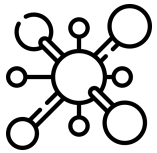
"To devise pragmatic approximations that are predictive without involving too much computation"

QUANTUM CHEMICAL CALCULATIONS

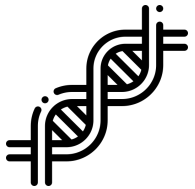
IN ATOMS



ATOM

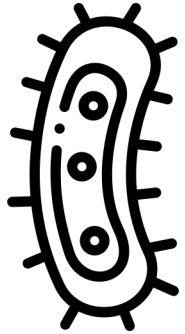


SMALL MOLECULE

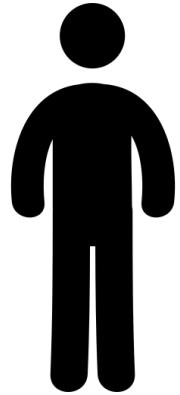


DNA

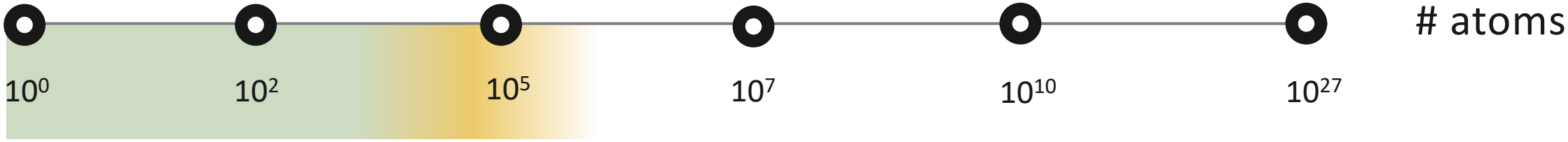
VIRUS



BACTERIA



HUMAN



CURRENT ($<10^4$)

EXASCALE (10^5)

Two Obstacles

Scalability

The amount of **computation required to solve** (accurately enough) **the Schrodinger equation scales as a high power of the number of atoms**, *N*, within a molecular system.

Hardware/Software

Inability of many quantum chemistry methods and algorithms (as traditionally designed) **to use efficiently novel (throughput-oriented) massively parallel processor architectures.**

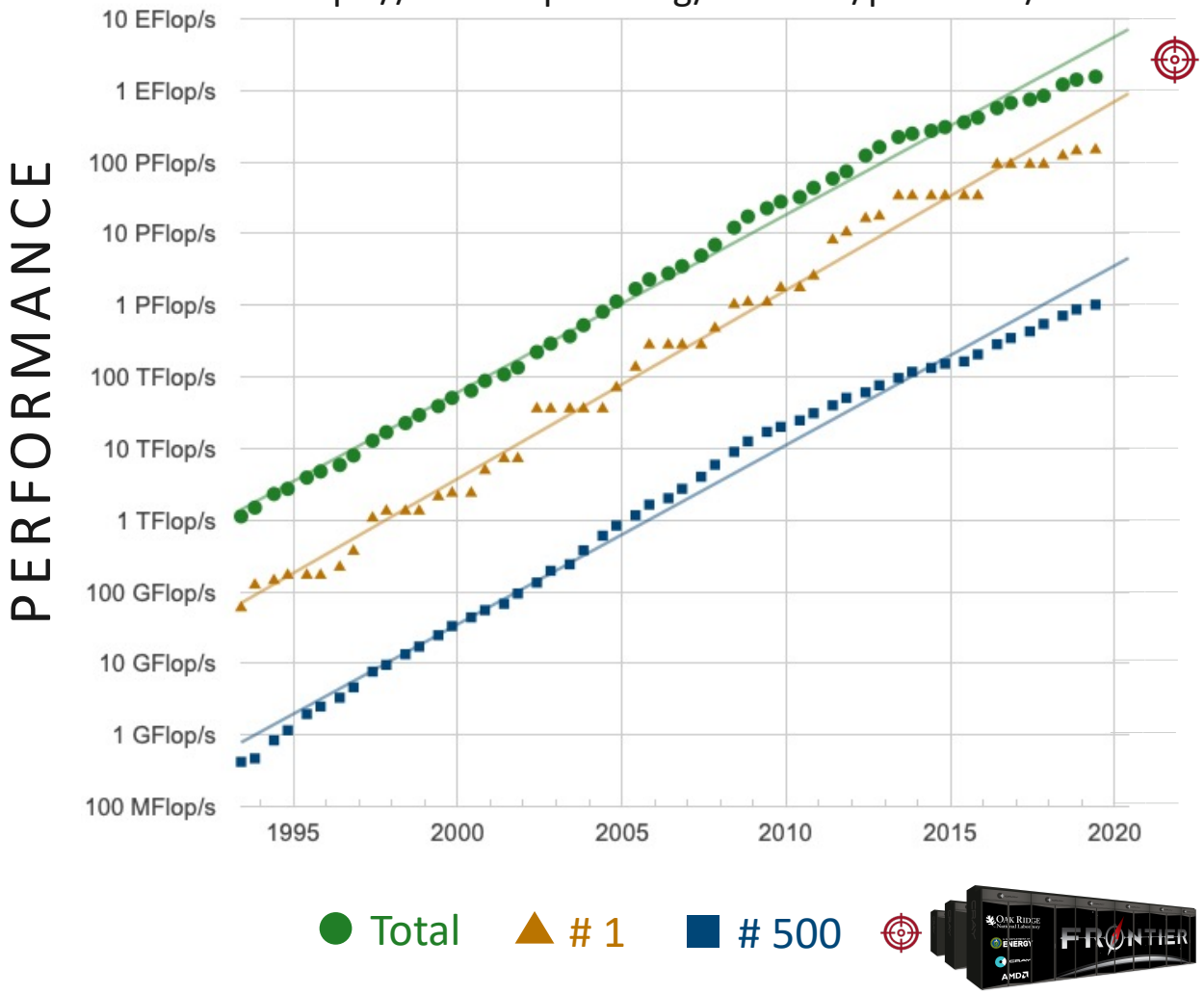
Scalability

COST VERSUS ACCURACY

Method	Scaling (time complexity)	Accuracy
Hartree-Fock	$\mathcal{O}(N^4)$	Qualitative
MP2	$\mathcal{O}(N^5)$	Accurate – predictive
CCSD	$\mathcal{O}(N^6)$	Accurate – predictive
CCSD(T)	$\mathcal{O}(N^7)$	Very accurate – predictive
FCI	$\mathcal{O}(N!)$	Exact (but NP!)

Current and emerging hardware

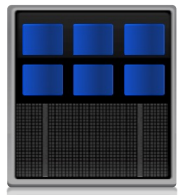
https://www.top500.org/statistics/perfdevel/



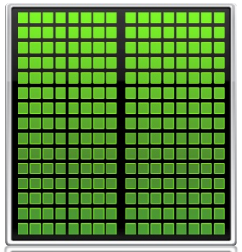
FROM PFLOP TO EFLOP

- Exascale is enabled by the paradigm shift from latency-oriented (CPU) to throughput-oriented architectures (GPU)
- Summit draws over **95% of throughput from GPUs**, CPUs are for flow admistration
- Efficient use of these heterogeneous architectures involves a **delicate interplay of "hosts" (CPUs) and "devices" (GPUs)**

CPU



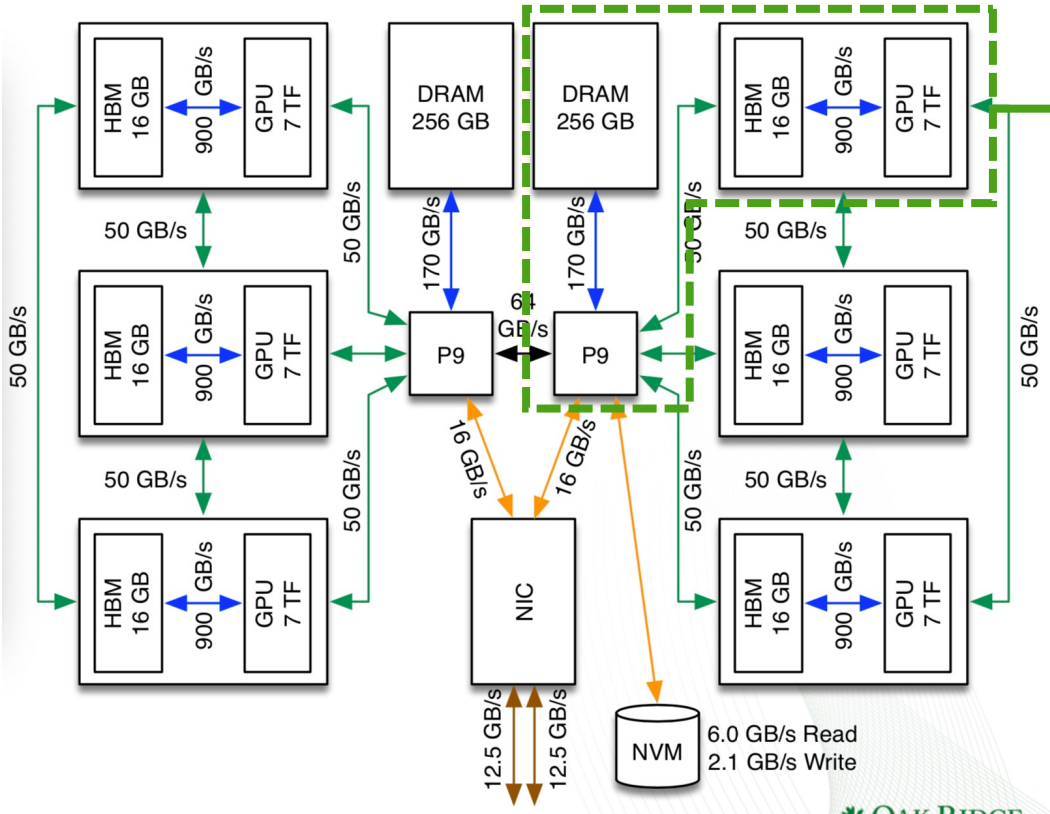
GPU



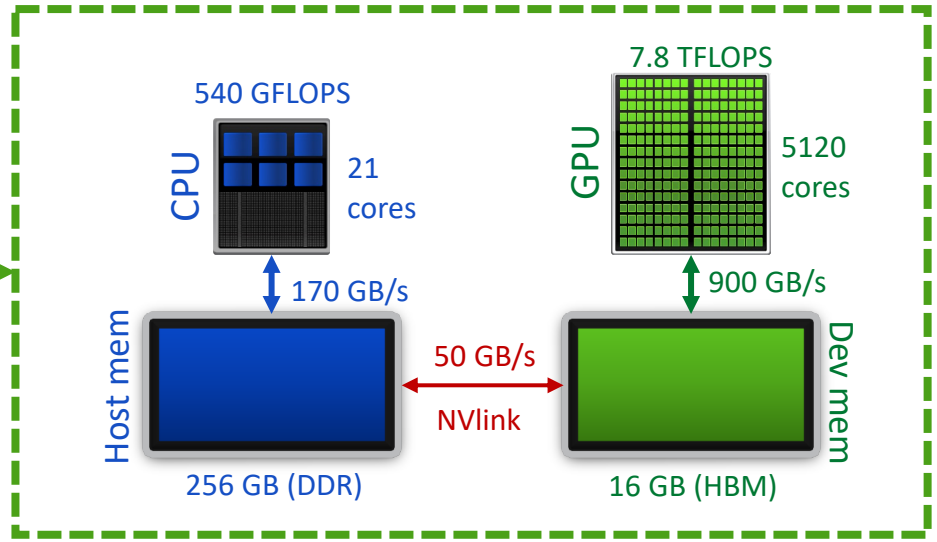
Challenges on a Summit node

HOST & DEVICE

ROLES & INTERFACING



Architecture of Summit node



- Transfer between host and device is a bottleneck
- Requires much more computation than data transfer
- Requires exposing massive parallelism for each GPU
- There are 4,608 such nodes on Summit, each with 6 GPU …

The path to
large scale matter simulations

To devise **novel** computational chemistry **methods** and **implementations** that

1. Have a **reduced computational complexity** – while retaining the required accuracy.
2. **Operate synergistically with throughput-oriented massively parallel hardware.**

Reduce scaling & enhance distributed parallelism: The fragmentation method

MANY-BODY EXPANSION

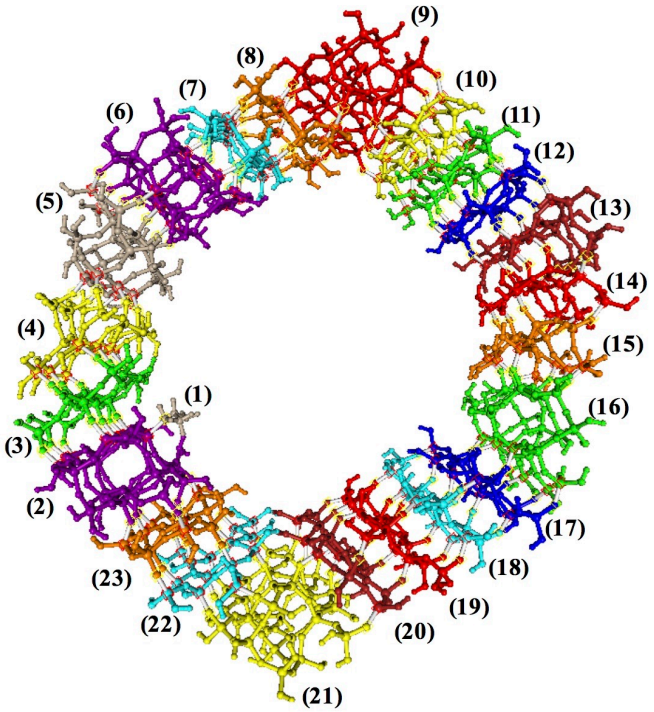
$$E_{HF} \cong \sum_{frag\ I} E_I + \sum_{\substack{frag\ I \\ frag\ J}} E_{IJ} + \Delta_{corr}$$

$$E_I = E_I^{HF} + E_I^{MP2}$$

$energy\ for\ frag\ I$

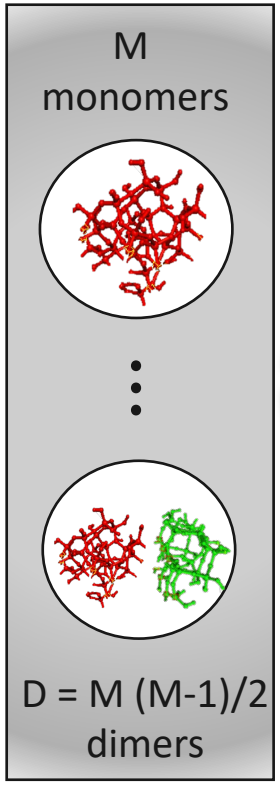
$O(M^2)\ cost$

Big molecule

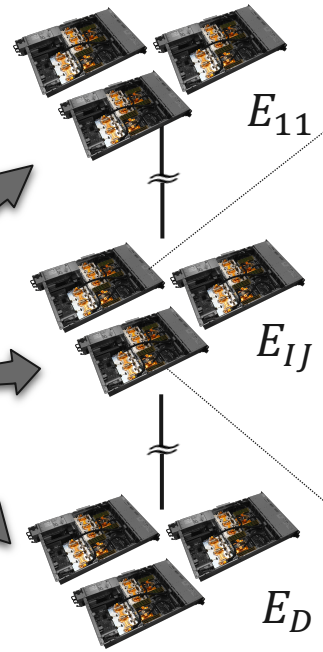


M fragment monomers

Fragment queue

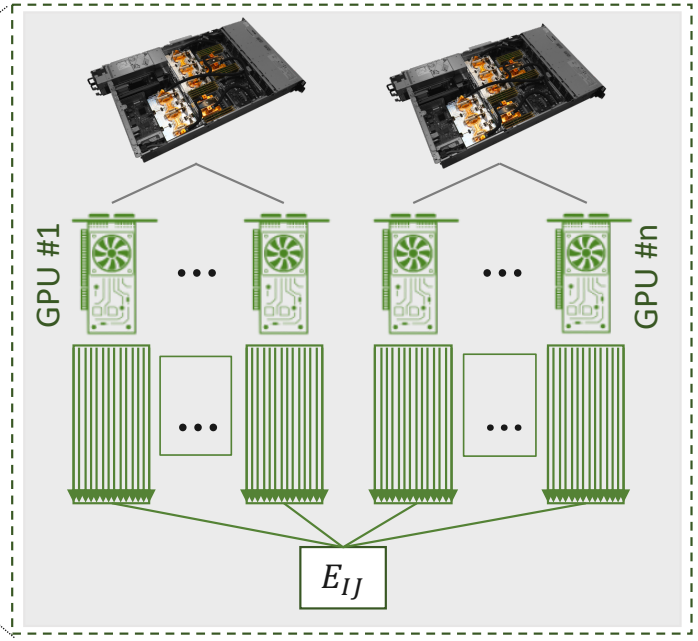


MPI Group #1



MPI Group #D

MPI Group #IJ



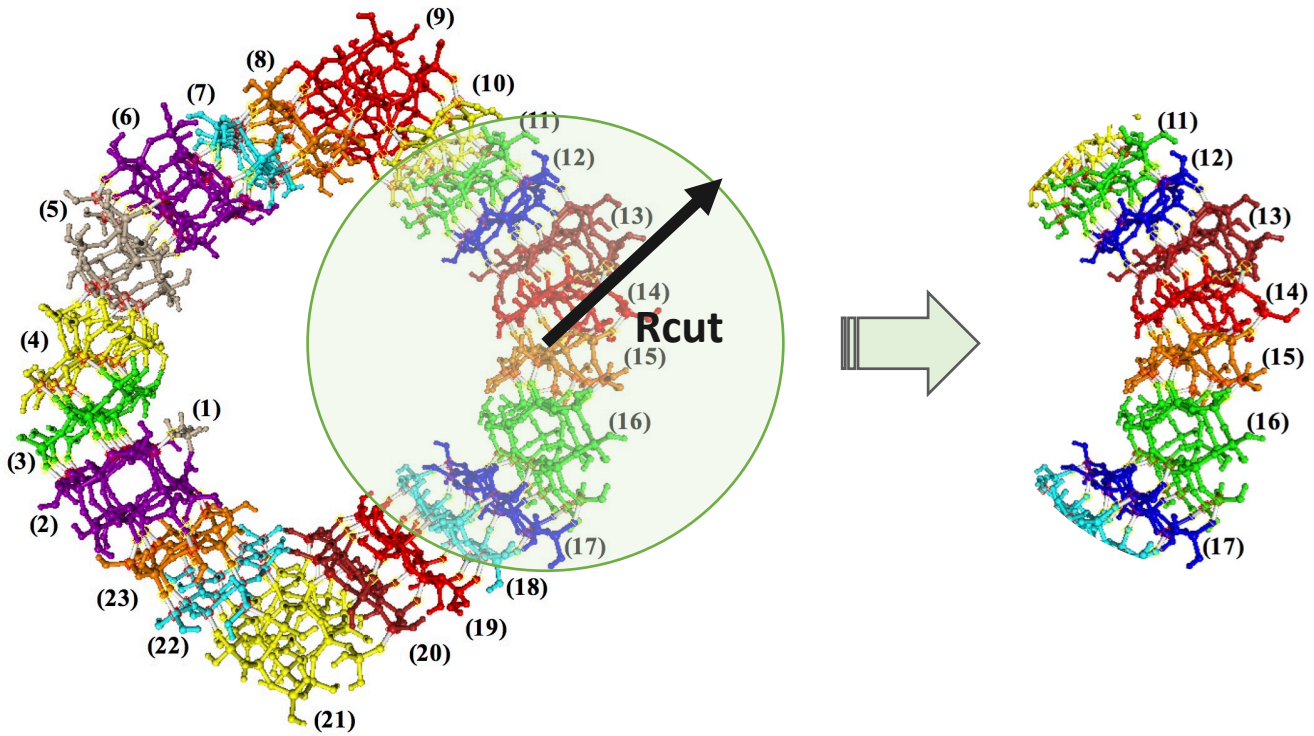
Many-GPU MPI-Group
calculation on fragment dimer “IJ”

The fragmentation method:
Achieving linear scaling

MANY-BODY EXPANSION

$$E_{HF} \cong \sum_{frag\ I} E_I + \sum_{\substack{I<J \\ R_{IJ}<Rcut}} E_{IJ} + \Delta_{corr}$$

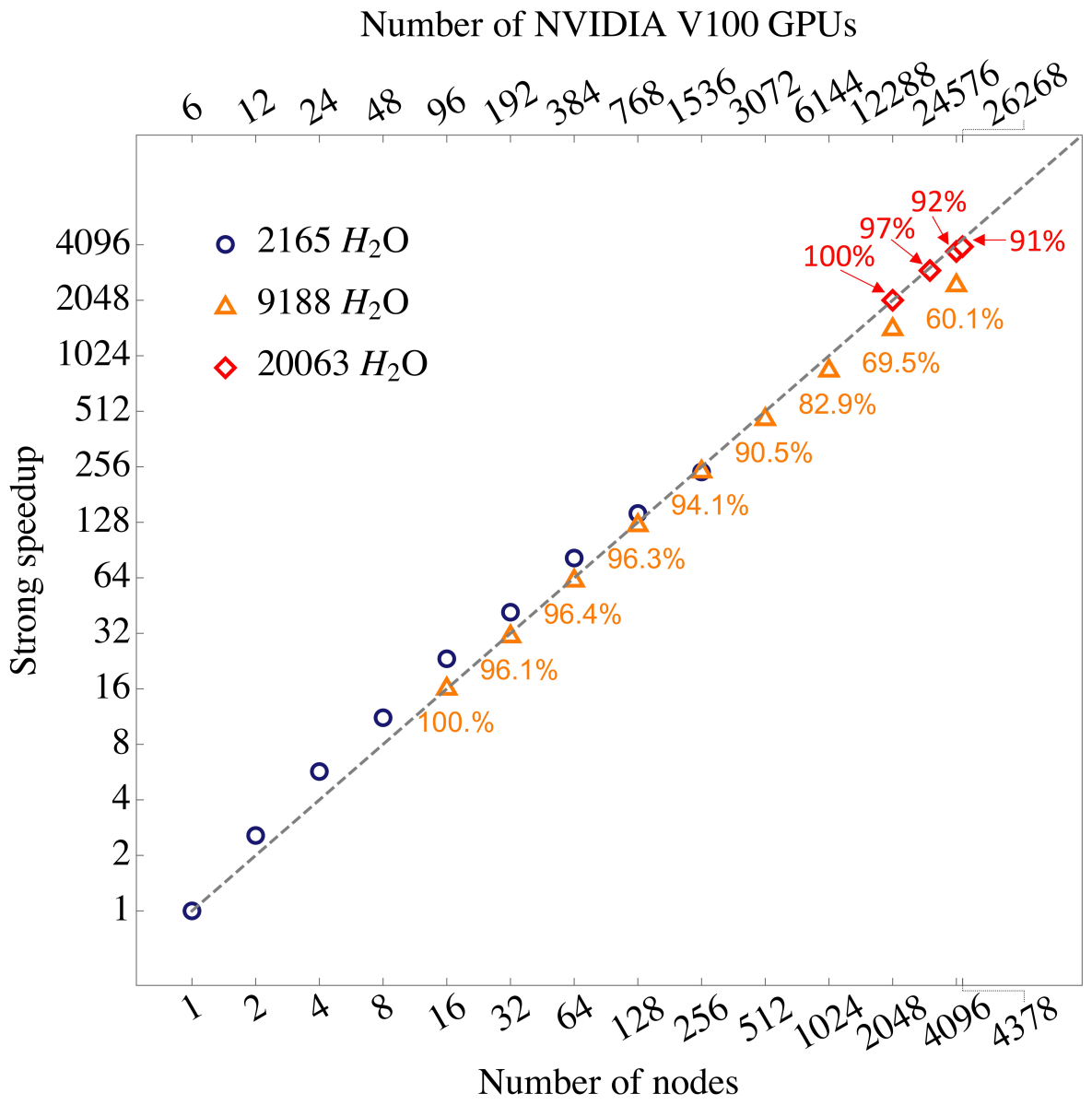
$$E_I = HF + MP2 \quad energy\ for\ frag\ I$$



M fragment monomers

- Each monomer "I" is couple only with $O(1)$ monomers "J"
- In total only $O(M)$ dimers are computed ➛ linear computational complexity
- For sufficiently large $\boldsymbol{Rcut}$, no accuracy is lost!

HF performance: strong scaling on Summit



MP2 via tensor decomposition : RI-MP2

$$E^{(2)} = \sum_{ij}^{N_o} \sum_{ab}^{N_v} \frac{G_{ia}^{jb}(2G_{ia}^{jb} - G_{ib}^{ja})}{\epsilon_i + \epsilon_j - \epsilon_a - \epsilon_b} \qquad G_{ia}^{jb} \approx \sum_{PQ}^{N_x} D_{ia}^{P} J_{PQ}^{-1} D_{jb}^{Q} \qquad N_x \gg N > N_v > N_o$$

		FLOPS	*MEM*
AO to MO transform	$D_{ia}^{P} = \sum_{\mu\nu}^{N} C_{i\mu} C_{a\nu} d_{\mu\nu}^{P}$	$\mathcal{O}(N^4)$	$\mathcal{O}(N^3)$
Form B	$B_{ia}^{P} = \sum_{Q}^{N_x} D_{ia}^{P} J_{PQ}^{-1}$	$\mathcal{O}(N^4)$	$\mathcal{O}(N^3)$
Form G	$G_{ia}^{jb} = \sum_{P}^{N_x} B_{ia}^{P} D_{jb}^{P}$	$\mathcal{O}(N^5)$	$\mathcal{O}(N^2)$
Increment $E^{(2)}$	$E^{(2)} \mathrel{+}= \frac{G_{ia}^{jb}(2G_{ia}^{jb} - G_{ib}^{ja})}{\epsilon_i + \epsilon_j - \epsilon_a - \epsilon_b}$	$\mathcal{O}(N^4)$	$\mathcal{O}(1)$

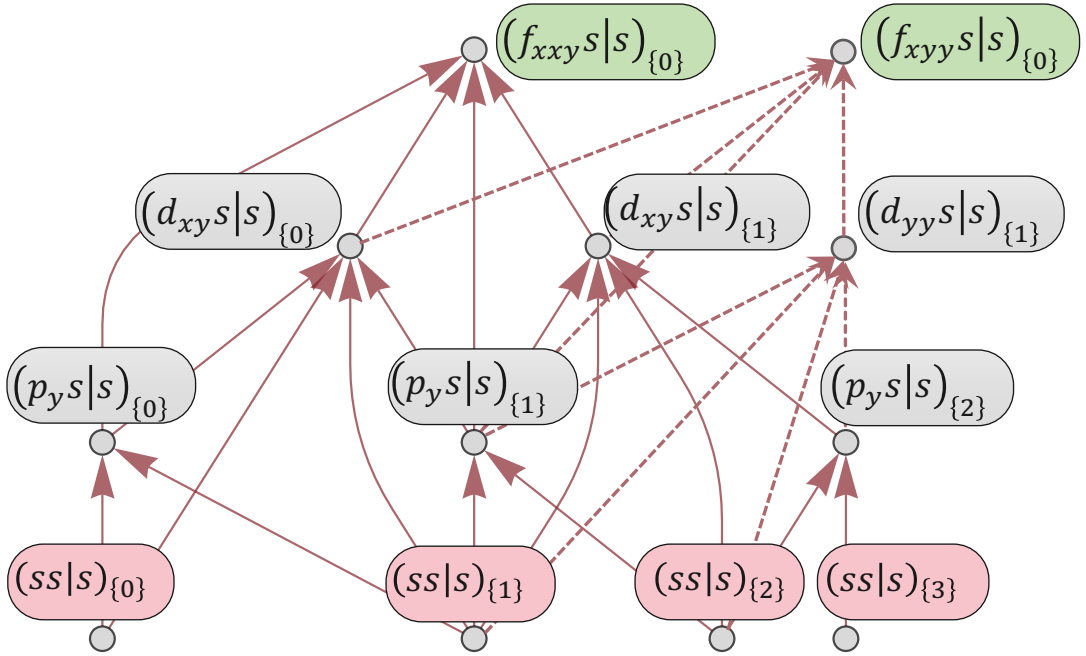
RI-MP2
Algorithmic Challenges
On Many-GPU Architecture

- Fast evaluation of two- and three-center integrals, J_{PQ} and $d_{\mu\nu}^{P}$, on GPU
- Tensors are too large for GPU memory, efficient overlap of GPU ⇔ Host data transfer with computation
- Dynamic workload balancing across multiple GPUs

Three-Center Integrals on GPU

$$d_{\mu\nu}^{P} = \sum_{k_1}^{K_M}\sum_{k_2}^{K_N}\sum_{k_3}^{K_P} g_{k_1k_2k_3}^{\mu\nu P} \qquad g_{k_1k_2k_3}^{\mu\nu P} = \iint \frac{\varphi_{\mu,k_1}^{\boldsymbol{m}}(\boldsymbol{r}_1)\varphi_{\nu,k_2}^{\boldsymbol{n}}(\boldsymbol{r}_1)\varphi_{P,k_3}^{\boldsymbol{p}}(\boldsymbol{r}_2)}{|\boldsymbol{r}_1 - \boldsymbol{r}_2|} d\boldsymbol{r}_1 d\boldsymbol{r}_2 \qquad \boldsymbol{m} = (m_x, m_y, m_z)$$
$$m = m_x + m_y + m_z$$

- Computation of $\mathcal{O}(N^3)$ $d_{\mu\nu}^{P}$ three center integrals
  - Large number ➡ good for parallelism
- However, $d_{\mu\nu}^{P}$ have heterogenous FLOP cost
  - FLOP cost changes based on K_M, K_N, K_P, $\boldsymbol{m}, \boldsymbol{n}, \boldsymbol{p}$ **(bad for workload balance)**
- $d_{\mu\nu}^{P}$ must be computed in classes with same $\boldsymbol{m}, \boldsymbol{n}, \boldsymbol{p}$
  - $d_{\mu\nu}^{P}$ are computed recursively, and integrals with same $\boldsymbol{m}, \boldsymbol{n}, \boldsymbol{p}$ share many recursive intermediates **(bad for workload balance and parallelism)**



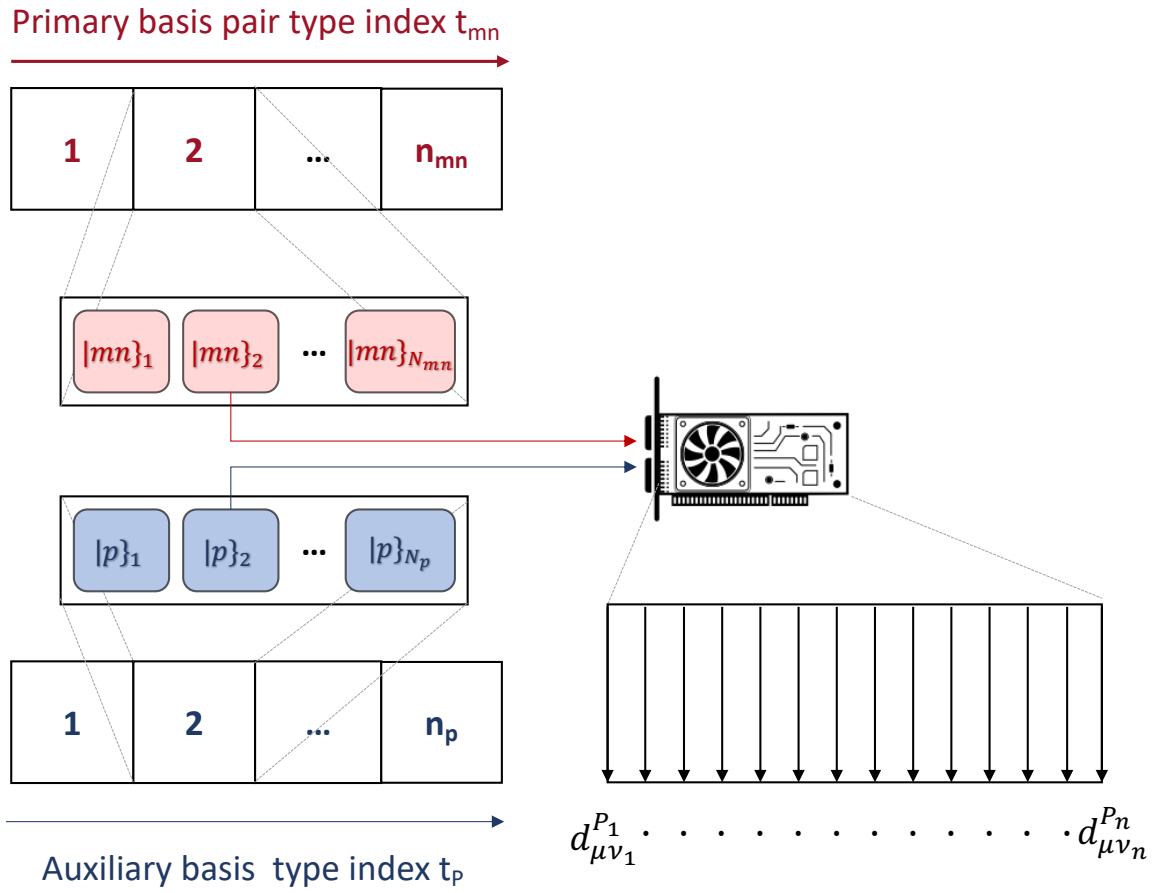
Three-Center Integrals on GPU

$$d_{\mu\nu}^{P} = \sum_{k_1}^{K_M} \sum_{k_2}^{K_N} \sum_{k_3}^{K_P} g_{k_1 k_2 k_3}^{\mu\nu P} \qquad g_{k_1 k_2 k_3}^{\mu\nu P} = \iint \frac{\varphi_{\mu,k_1}^{\boldsymbol{m}}(\boldsymbol{r}_1)\varphi_{\nu,k_2}^{\boldsymbol{n}}(\boldsymbol{r}_1)\varphi_{P,k_3}^{\boldsymbol{p}}(\boldsymbol{r}_2)}{|\boldsymbol{r}_1 - \boldsymbol{r}_2|} d\boldsymbol{r}_1 d\boldsymbol{r}_2$$

$$\boldsymbol{m} = (m_x, m_y, m_z)$$

$$m = m_x + m_y + m_z$$

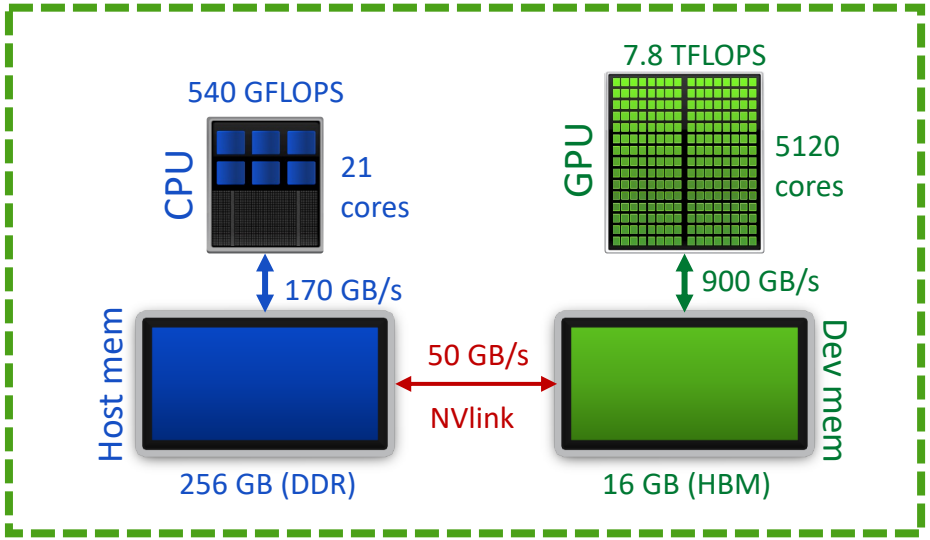
- The basis functions pairs $\mu\nu$ are classified by type t_{mn} based on m, n, K_M and K_N (same for P).
- Pairs $\mu\nu$ with same t_{mn} are gathered into batches $|mn\}$
- Batches with the same t_{mn} go in the same bin
- Dynamic workload distribution dispatches batches of $|mn\}$ and $|p\}$ by decreasing FLOP cost to GPUs
- The GPU computes $d_{\mu\nu}^{P}$ arising from $\{mn| \otimes |p\}$, parallelized over thousands of threads, each thread computing the same class of integrals (same FLOP cost)



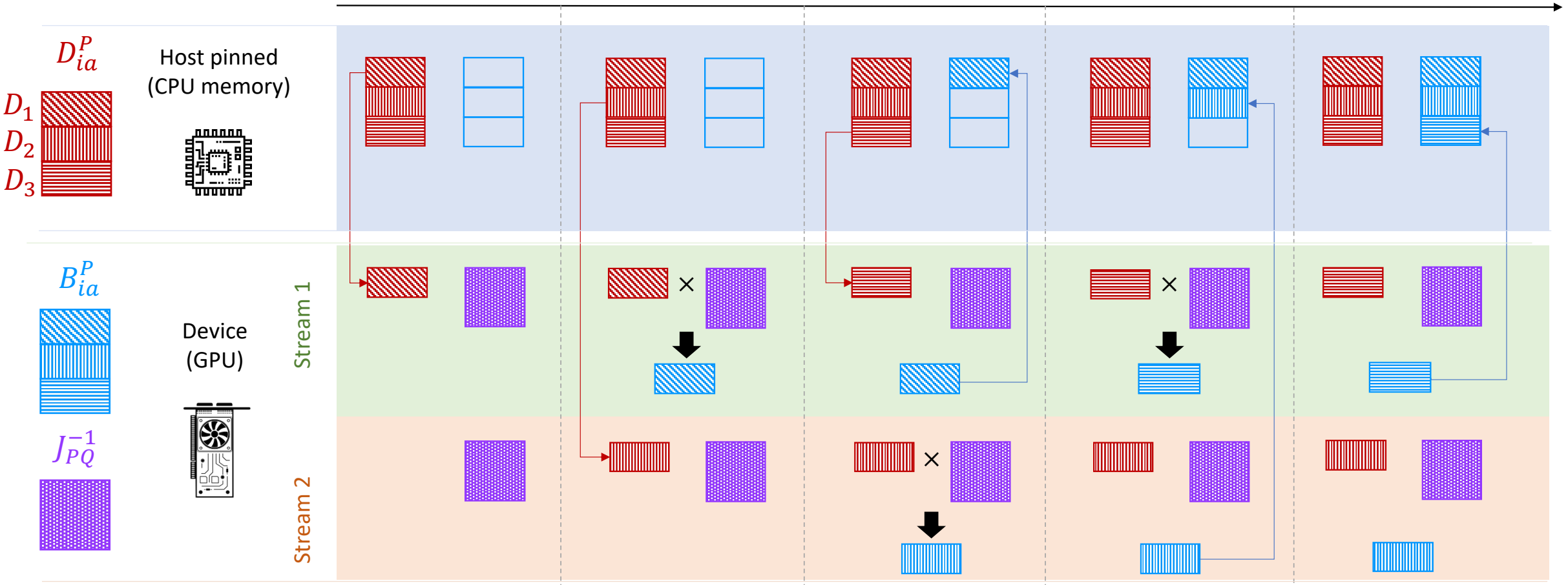
Dealing with GPUs' small memory

```
1  forall μ, ν do
2  |  foreach batch 𝒫 do
3  |  |  forall P ∈ 𝒫 do
4  |  |  |  Compute d^P_μν slice on GPU
5  |  |  end
6  |  |  Transfer d^P_μν slice GPU → host
7  |  end
8  end
9  forall P, Q do
10 |  Compute J_PQ on GPU
11 end
12 Form J^-1 by inverting J on GPU
13 Transfer C host → GPU
14 forall μ, a do
15 |  foreach batch 𝒫 do
16 |  |  Transfer d^P_μν slice host → GPU
17 |  |  forall P ∈ 𝒫 do
18 |  |  |  Compute slice A^P_μa = Σ^N_ν C_νa d^P_μν on GPU
19 |  |  end
20 |  |  Transfer A^P_μa slice GPU → host
21 |  end
22 end
```

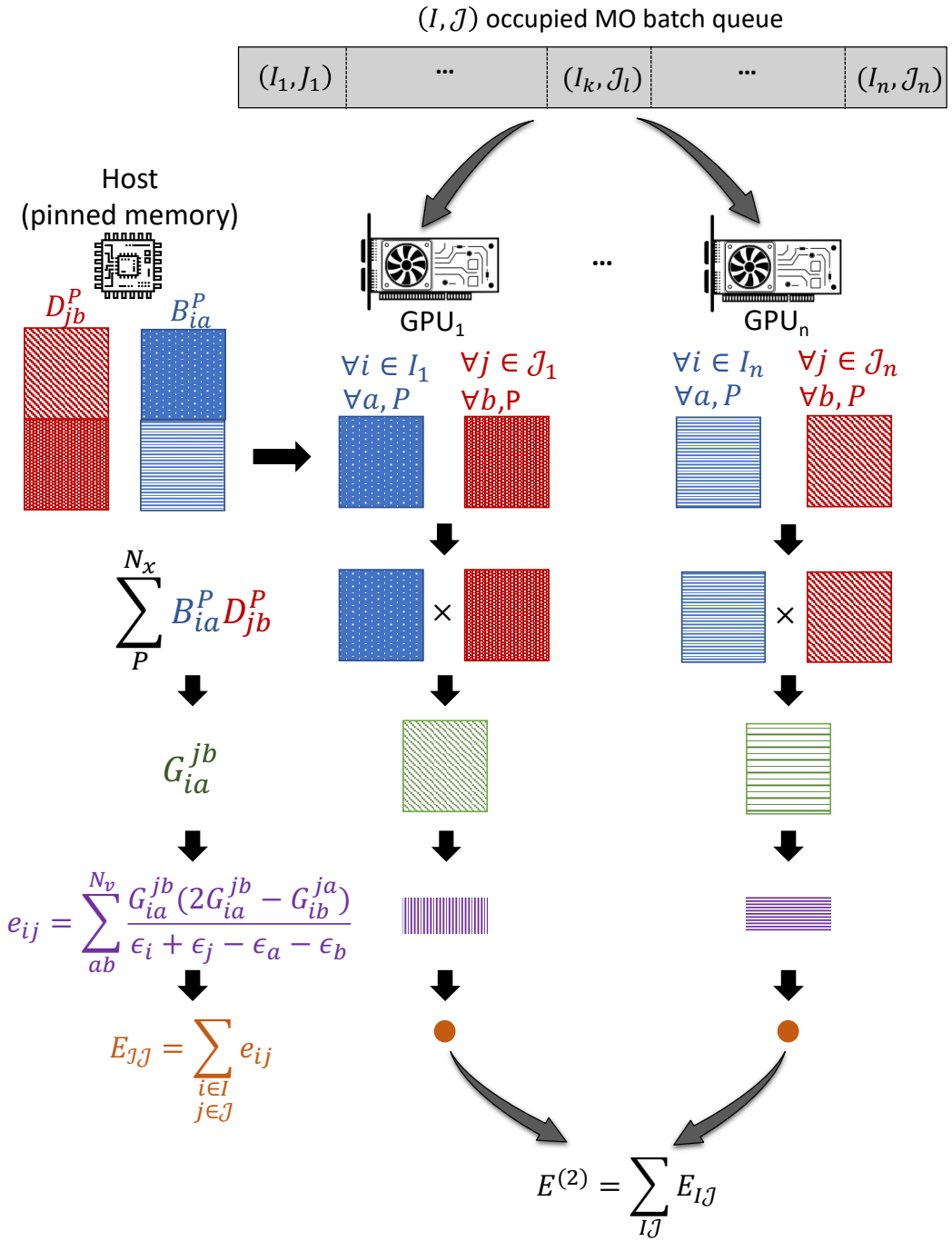
- All tensors (e.g. $\boldsymbol{d}, \boldsymbol{D}, \boldsymbol{B}$) involved in the RI-MP2 calculation, except $\boldsymbol{J}^{-1}$, are too large to be stored in the GPU memory
- The tensors are partitioned into slices
- Each batch is computed on the GPU, then stored to the host
- Tensor contractions require repeated Device ⇔ Host transfer of slices (**dangerous memory bottleneck**)



Double-stream tensor transfer/computation scheme



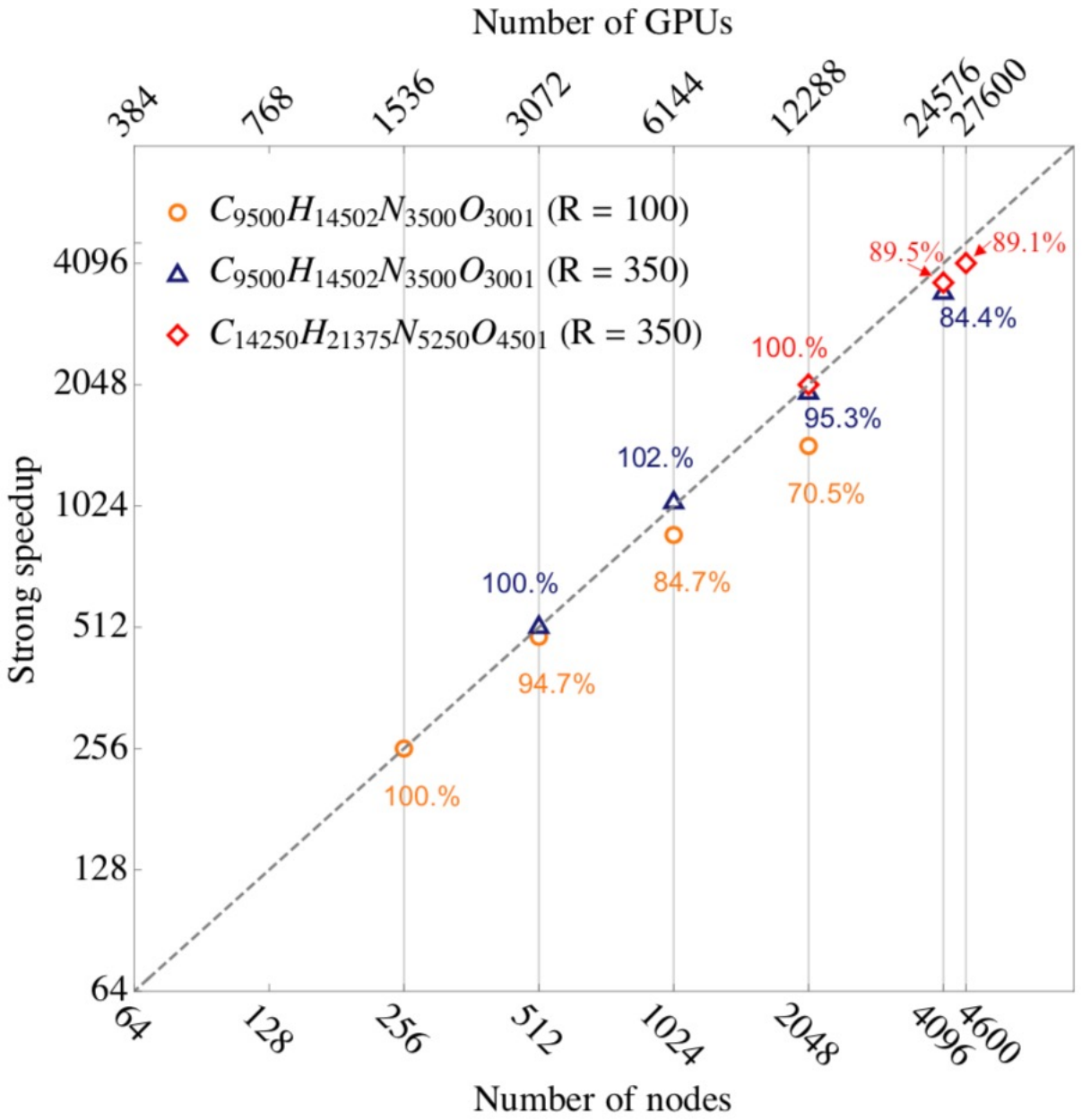
$$B_{ia}^{P} = \sum_{Q}^{N_x} D_{ia}^{P} J_{PQ}^{-1}$$



Multi-GPU energy accumulation

- MPI Coordinator/Worker scheme: Each workers is associated with one GPU
- Each rank has access to the full $\boldsymbol{B}$ and $\boldsymbol{D}$ tensors in host-pinned memory
- Each rank forms queue of occupied MO batch pairs (I, J)
- (I,J) pairs are distributed from the coordinator to the workers (GPUs) using dynamic load balancing
- Each GPU computes the G_{ia}^{jb} slice for all a and b indices, and for $i \in I$ and $j \in J$
- The MP2 energy is accumulated by each worker into e_{ij} arrays
- The e_{ij} are reduced across ranks only **once** at the end

Fragmentation SCF+RI-MP2: strong scaling



COMPUTATIONAL DETAILS

Results were **obtained on** the OLCF **Summit** system for

- **$(Asp\text{-}Ala\text{-}Hys\text{-}Lys)_{500}$ (>30k atom, >120k electrons)** using 6-31G(d)/cc-pVDZ-RIFIT ⊳312,525 primary and 1,273,596 auxiliary basis functions
- **$(Asp\text{-}Ala\text{-}Hys\text{-}Lys)_{750}$ (>45k atom, >180k electrons)** using 6-31G(d)/cc-pVDZ-RIFIT ⊳ 464,293 primary and 1,896,846 auxiliary basis functions
- **Different dimer cutoff radii (R)** were used to test convergence of the energy
- **All electron calculation (no frozen core)**

$(Asp\text{-}Ala\text{-}Hys\text{-}Lys)_{750}$ Parallel efficiencies

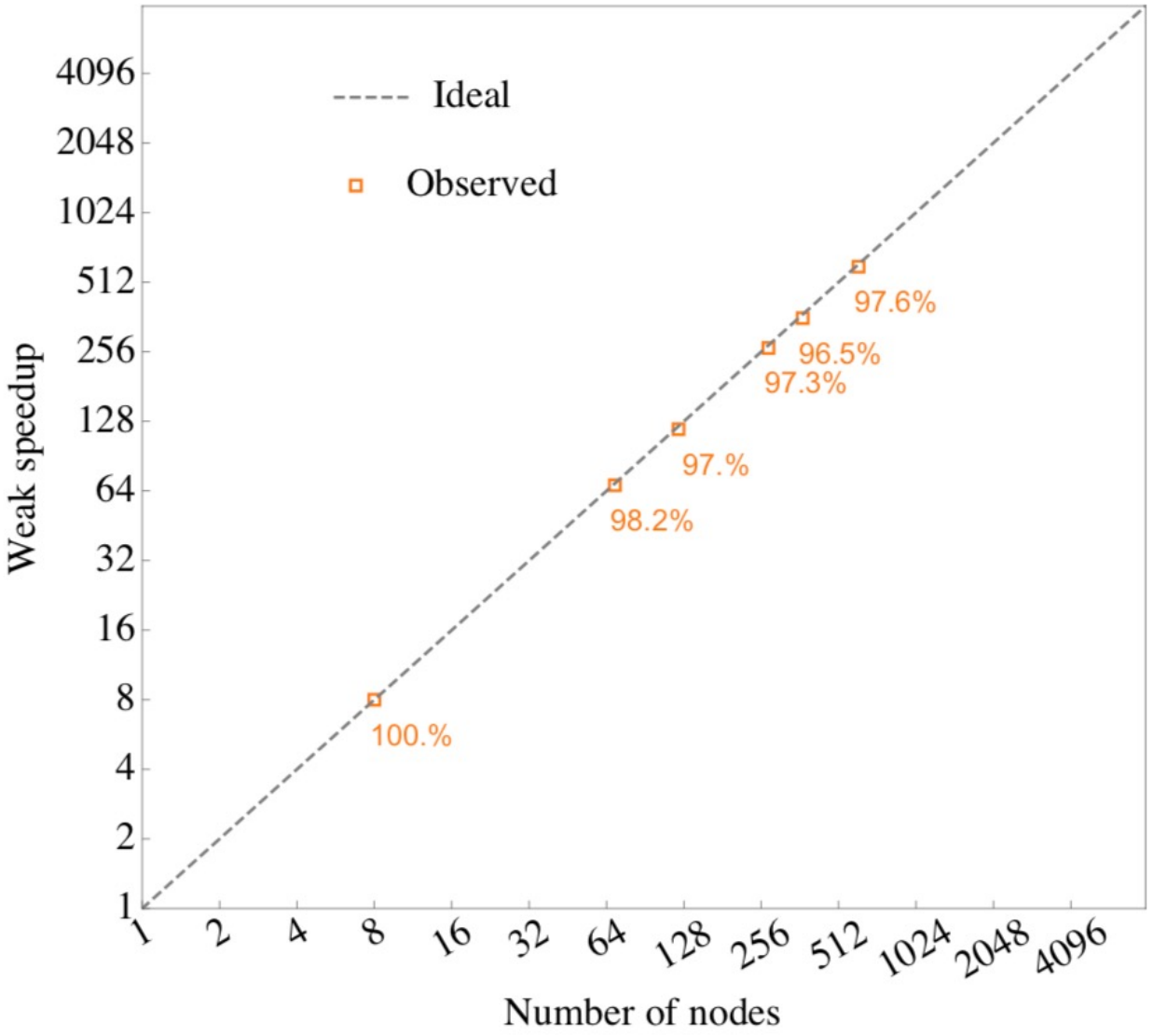
- ◎ 89.5% on 4096 and 24576 V100 GPUs
- ◎ 89.1% on 4600 nodes and 27,600 GPUs (99.8% of Summit)

SCF+RI-MP2: timings

Molecule	N	N_x	R	Number of nodes	Wall time (s)
$C_{9500}H_{14502}N_{3500}O_{3001}$	312,525	1,273,596	100	256	2015
				512	1064
				1024	594.5
				2048	357.5
$C_{9500}H_{14502}N_{3500}O_{3001}$	312,525	1,273,596	350	512	3537.7
				1024	1734.5
				2048	928.0
				4096	524.2
$C_{14250}H_{21375}N_{5250}O_{4501}$	464,293	1,896,846	350	2048	1354.3
				4096	756.3
				4600	676.5

- **$(Asp\text{-}Ala\text{-}Hys\text{-}Lys)_{750}$ ran in ~11 minutes on 4600 nodes**
- With 464,293 primary basis function, 1,896,846 auxiliary basis functions, >45,000 atoms and correlating >180,000, was **the largest ab initio correlated quantum chemistry calculation ever performed** (previous largest 2,440 atoms and 6,800 correlated electrons)

SCF+RI-MP2 : weak scaling



COMPUTATIONAL DETAILS

- The total **computational workload is proportional to the square of the number of fragment dimers** (no dimer screening)
- To ensure an approximately constant workload across nodes, **weak scaling data was obtained by increasing the number of nodes as the number of dimers**
- Reference systems **238 H_2O on 8 nodes**

- **The 612-node calculation ran on a 2086-H_2O system, yielding a speedup of 74.7x with 8 nodes as a reference**

SCF+RI-MP2:
Performance Comparison and Validation

Molecule	N	N_x	Wall time (s)	Speedup over LibCChem CPU	Speedup over LibCChem GPU
Adenine-Thymine	505	2103	3.68	25.8	5.7
$(Asp\text{-}Ala\text{-}Hys\text{-}Lys)_2$	750	3027	11.75	46	5.4
C_{60}	900	3960	10.8	42.1	4.2

Performance against existing GAMESS/LibCChem CPU and GPU:

- **All calculations used cc-pVDZ/cc-pVDZ-RIFIT**
- **LibCChem CPU used all 42 cores of single IBM P9 on Summit**
- **LibCChem GPU and new code used a single V100 GPU**

Validation:

- **Result were validated again results from the third-party Q-Chem software, using the S22(3) test set, and variable-size water clusters, using a variety of basis sets**

Summary of Contributions

- Designed and implemented a number of algorithmic innovations compared to the state of the art
- These innovations enable to perform linear scaling SCF+MP2 calculations at unprecedented molecular scales
- Excellent weak and strong scaling is demonstrated up to the 99.8% of the Summit system, using up to 27,600 V100 GPUs.
- Performed the largest ab initio electronically correlated calculation to date on a polypeptide including >45,000 atoms and correlating >180,000, using almost 2 million basis functions
- This work renders significantly larger scale ab initio correlated quantum chemistry calculations feasible, with a major impact on chemical, physical, biological and engineering sciences.

QUESTIONS