

# OpenMP Offload in the GAMESS Quantum Chemistry Package

September 29, 2023



Australian  
National  
University



EPAnalytics

Georgia  
Tech



OLD DOMINION  
UNIVERSITY

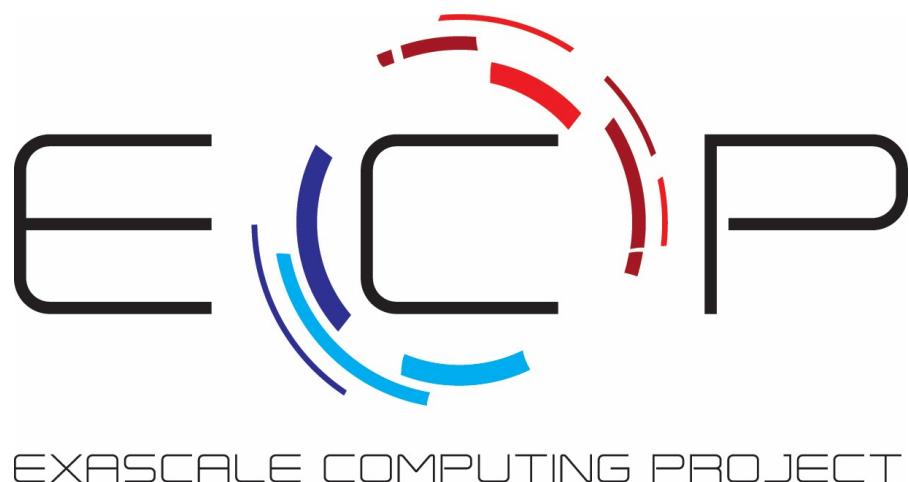
The Old Dominion University logo features a gold crown icon above the words 'OLD DOMINION UNIVERSITY' in a serif font. The 'O' in 'OLD' and the 'D' in 'DOMINION' are capitalized and have decorative flourishes.

UTEP

The UTEP logo is rendered in orange, with the letters 'U', 'T', 'E', and 'P' stacked vertically. The 'U' and 'T' are slightly slanted, and there is a small registered trademark symbol (®) to the right of the 'P'.

# Funding

This research was supported by the Exascale Computing Project (17-SC-20-SC), a joint project of the U.S. Department of Energy's Office of Science and National Nuclear Security Administration, responsible for delivering a capable exascale ecosystem, including software, applications, and hardware technology, to support the nation's exascale computing imperative.



# GAMESS ECP Team



Mark Gordon (PI)

Melisa Alkan (Stanford)

**Daniel Del Angel**

Dipayan Datta

Taylor Harville

**Buu Pham**

**Tosaporn Sattasathuchana**

Bryce Westheimer

**Peng Xu**

**Federico Zahariev**



Australian  
National  
University



Giuseppe Barca (co-PI)  
*Jorge Galvez*

Laura Carrington (co-PI)  
Ananta Tiwari  
*Sarom Leang*

David Sherril (co-PI)  
*David Poole*

Masha Sosonkina (co-PI)  
Viabhav Sundriyal (Intel)  
Eric Jensen

Shirley Moore (co-PI)  
Henry Moncada

\*Former member of the Mark S. Gordon Research Group

In Attendance

# Acknowledgements

- Colleen Bertoni\* (Argonne National Laboratory)
- **Dossay Oryspayev** (Brookhaven National Laboratory)
- Jack Deslippe (Lawrence Berkeley National Laboratory)
- Dmytro Bykov (Oak Ridge National Laboratory)
- AMD (Brian Cornille, Bob Robey)
- HPE Cray (Luke Roskop\*, Marcus Wagner)
- Intel (Brian Whitney, Xinmin Tian, Ravi Narayanaswamy)
- HPC Toolkit (John Mellor-Crummey, **Wileam Phan**, Laksono Adhianto)

# Acknowledgements

- **Hackathons**

- GPU Hackathon, OLCF, 2019
- ECP OpenMP, SOLLVE, 2020
- GPU Hackathon, OLCF+NERSC, 2021
- ECP OpenMP, SOLLVE, 2021
- AMD, 2022
- Open, NERSC, 2022 (Apr., Nov.)
- Crusher, OLCF, 2022
- Frontier, OLCF, 2023
- Aurora, ALCF, 2023

- **Workshops**

- ATPESC, ALCF, 2019
- Aurora COE, ALCF, 2020
- Frontier COE, OLCF, 2020

- **Training**

- Spock, OLCF, 2021
- Crusher, OLCF, 2022
- Frontier, OLCF, 2023
- HIP, OLCF + NERSC, 2023

# Agenda

- GAMESS
- Electronic Structure Theory
- Initial Molecular Orbital Guess
- Resolution-of-Identity Møller-Plesset Second-Order Perturbation Theory (RI-MP2)
- Effective Fragment Molecular Orbital Method (EFMO)
- ECP Science Challenge
- Density Functional Theory (DFT)
- Issues, Workarounds, and Wishlist
- Questions

# **General Atomic and Molecular Electronic Structure System (GAMESS)**

Principal Investigator: Mark S. Gordon (Ames Laboratory)

Maintainer/Release Manager: Sarom S. Leang (EP Analytics, Inc.)

# GAMESS

- General purpose electronic structure code
- Focus on *ab initio* quantum chemistry
- Mark S. Gordon Quantum Theory Group/Iowa State University
- Lots of capabilities – too many to list  
[www.msg.chem.iastate.edu/gamess/capabilities.html](http://www.msg.chem.iastate.edu/gamess/capabilities.html)
- Two code bases:
  - GAMESS (Fortran 77/90/95/2003)
  - LIBCCHEM (C++)
- 2022 statistics : +9,000 downloads across 128 countries
- DOI: 10.1021/acs.jctc.3c00379 (accepted)



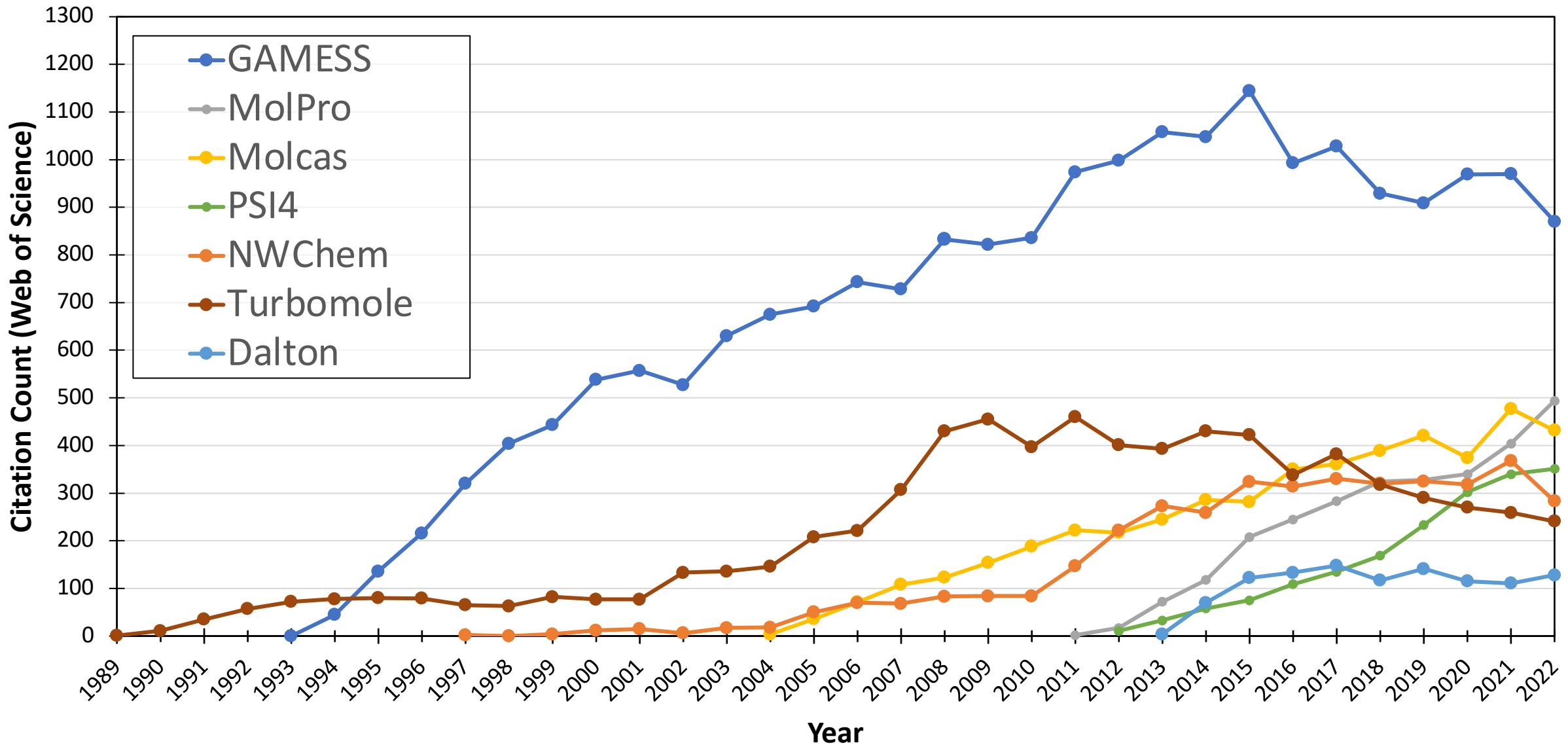
# History

- Early 1980 - Fork of HONDO 5 from NRCC (Michel Dupuis)
- 1982 - Mike Schmidt joined Mark Gordon's group at NDSU (**80,000 LOC**)
- 1991 - Parallelization (T. Windus)
- 1992 - Mark Gordon's group moves to ISU 
- 1996 - Distributed Data Interface (DDI) (G. Fletcher), EFP method (J. Jensen)
- 2004 - DDI for SMP using System V shared memory
- 2004 - DDI for subgroups, release of the FMO method (D. Fedorov) (**650,000 LOC**)
- 2007 - Release of the EFMO method (S. Pruitt)
- 2010 - Release of GAMESS-LIBCCHEM (A. Asadchev)
- 2017 - Mike Schmidt enters partial-retirement (**759,000 LOC**)

# History (OpenMP Parallelization in GAMESS)

- 2016 – Two-electron integrals (V. Mironov)
- 2017 – Two-electron gradient code and HF algorithm w/ shared Fock (V. Mironov)
- 2017 – RI-MP2 energy code (B. Pham)
- 2018 – RI-MP2 gradient code (B. Pham)
- 2019 – One-electron integrals, DFT, FMO, PCM (V. Mironov)
- 2019 – QM-EFP2 energy code (P. Xu, T. Sattasathuchana)
- 2020 – RI-CC (D. Datta)

# Citations by Year



# Citations by Year (References Tracked)

- **GAMESS**
  - 1993: <https://doi.org/10.1002/jcc.540141112>
  - 2005: <https://doi.org/10.1016/B978-044451719-7/50084-6>
  - 2020: <https://doi.org/10.1063/5.0005188>
- **NWChem**
  - 1995: <https://doi.org/10.1002/qua.560560851>
  - 2000: [https://doi.org/10.1016/S0010-4655\(00\)00065-5](https://doi.org/10.1016/S0010-4655(00)00065-5)
  - 2010: <https://doi.org/10.1016/j.cpc.2010.04.018>
  - 2020: <https://doi.org/10.1063/5.0004997>
  - 2021: <https://doi.org/10.1021/acs.chemrev.0c00998>
- **PSI4**
  - 2011: <https://doi.org/10.1002/wcms.93>
  - 2017: <https://doi.org/10.1021/acs.jctc.7b00174>
  - 2020: <https://doi.org/10.1063/5.0006002>
- **Dalton/LSDalton**
  - 2011: <https://doi.org/10.1002/wcms.93>
- **Molpro (Commercial)**
  - 2012: <https://doi.org/10.1002/wcms.82>
  - 2020: <https://doi.org/10.1063/5.0005081>
- **Turbomol (Commercial)**
  - 1989: [https://doi.org/10.1016/0009-2614\(89\)85118-8](https://doi.org/10.1016/0009-2614(89)85118-8)
  - 2014: <https://doi.org/10.1002/wcms.1162>
  - 2020: <https://doi.org/10.1063/5.0004635>
- **Molcas**
  - 2003: [https://doi.org/10.1016/S0927-0256\(03\)00109-5](https://doi.org/10.1016/S0927-0256(03)00109-5)
  - 2010: <https://doi.org/10.1002/jcc.21318>
  - 2016: <https://doi.org/10.1002/jcc.24221>
  - 2019: <https://doi.org/10.1021/acs.jctc.9b00532>
  - 2020: <https://doi.org/10.1063/5.0004835>

# Electronic Structure Theory

- Describe the motions of electrons
- Many-body problem with no closed solution
- Development and scaling of computational methods to describe electronic motion in atoms and molecules

# Electronic Schrödinger Equation

$$\hat{H}_{elec} \Psi_{elec} = E_{elec} \Psi_{elec}$$

$$\hat{H}_{elec} = -\frac{1}{2} \sum_{i=1}^N \nabla_i^2 - \sum_{i=1}^N \sum_{A=1}^M \frac{Z_A}{r_{iA}} + \sum_{i < j}^N \frac{1}{r_{ij}}$$

- $\hat{H}_{elec}$  is electronic Hamiltonian operator
- $\Psi_{elec}$  is electronic wavefunction of the system
- $E_{elec}$  is the electronic energy of the system
- $N$  is the total number of electrons
- $M$  is the total number of nuclei
- Born-Oppenheimer approximation

A. Szabo and N. S. Ostlund, Modern Quantum Chemistry: Introduction to Advanced Electronic Structure Theory. McGraw-Hill, New York, 1989.

# Electronic Schrödinger Equation (cont.)

$$\hat{H}_{elec} \Psi_{elec} = E_{elec} \Psi_{elec}$$

$$\hat{H}_{elec} = -\frac{1}{2} \sum_{i=1}^N \nabla_i^2 - \sum_{i=1}^N \sum_{A=1}^M \frac{Z_A}{r_{iA}} + \sum_{i < j}^N \frac{1}{r_{ij}}$$

- $\Psi_{elec}$  contains all information about the system
  - Geometries (equilibrium, transition states)
  - Vibrational frequencies (IR spectra)
  - Excited states (UV/VIS spectra)
  - Dipole moment, polarizability
  - Barrier heights and reaction paths
  - Reaction rates with transition state theory
  - Thermodynamic properties with statistical mechanics

# Hartree-Fock (HF or SCF or RHF) Method

$$\hat{H}_{elec}^{HF} \Psi_{elec}^{HF} = E_{elec}^{HF} \Psi_{elec}^{HF}$$

$$\hat{H}_{elec}^{HF} = -\frac{1}{2} \sum_{i=1}^N \nabla_i^2 - \sum_{i=1}^N \sum_{A=1}^M \frac{Z_A}{r_{iA}} + \sum_{i=1}^N V_i^{HF}$$

- Based on the independent particle model
- Electron-electron repulsion terms is replaced with an effective field produced from the averaged position of all other electrons
  - Results in an iterative process starting from a guessed set of functions (initial guess)
- Electron correlation energy,

$$E_{correlation} = E - E_{HF}$$

# Beyond Hartree-Fock

- Increased sophistication of methods to approach exact solution to the electronic Schrödinger equation
- Formal scaling of methods based on system size,  $N$



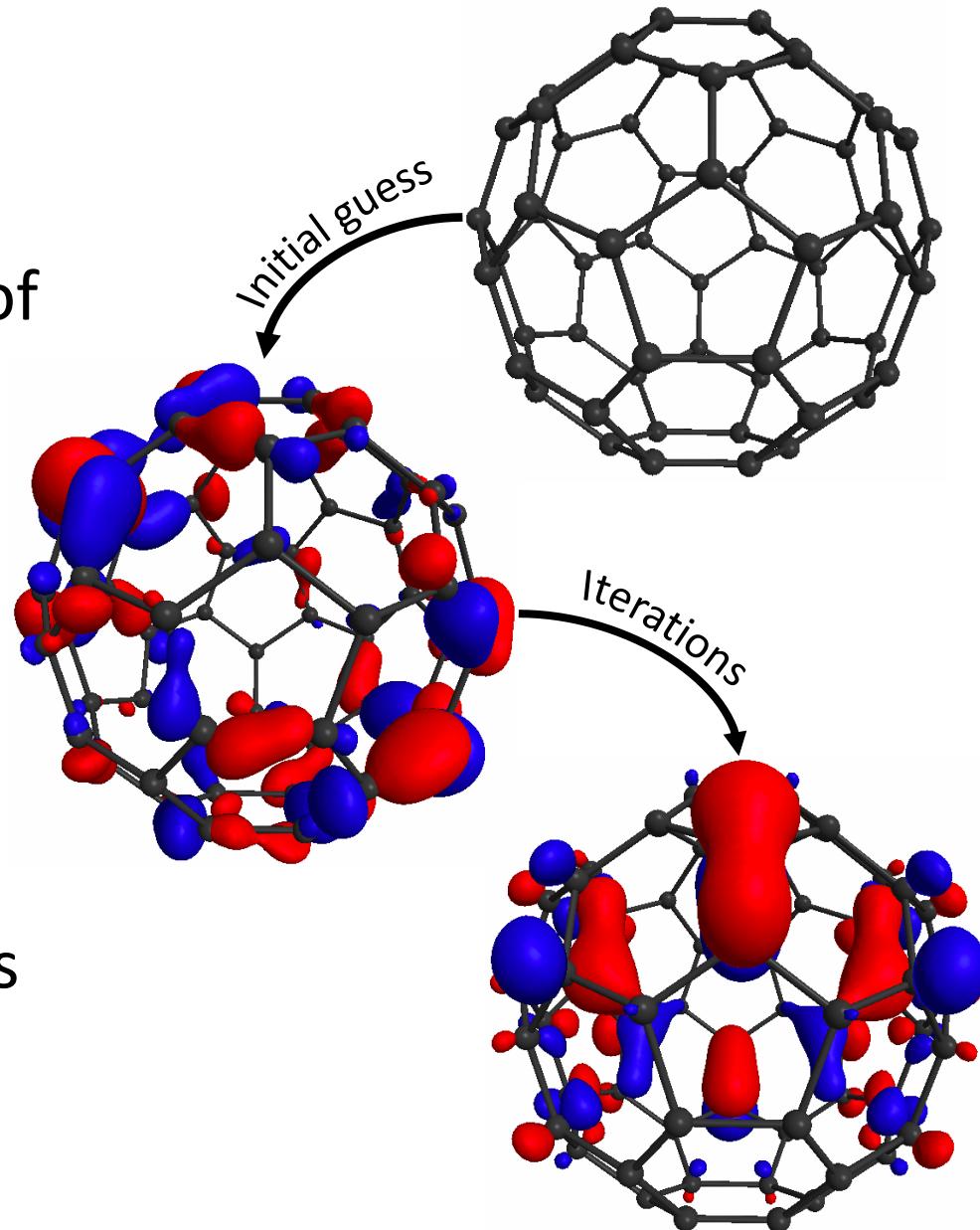
Scaling Behavior	Method
$N^4$	Hartree-Fock (HF), Density Function Theory (DFT)
$N^5$	Møller-Plesset Second-Order Perturbation Theory (MP2)
$N^7$	Coupled-Cluster with Singles and Doubles and Perturbative Triples excitations (CCSD(T))
$N^8$	Coupled-Cluster with Singles and Doubles and Triples excitations (CCSDT)
$N!$	Full Configuration-Interaction (Full-CI)

# Initial Molecular Orbital Guess

Lead: Daniel Del Angel (Ames National Laboratory)

# Initial Guess

- Electronic structure described in terms of molecular orbitals (MOs)
- MOs optimized through an iterative procedure
- Starting set of MOs (initial guess)
- Matrix operations used for initial guess generation
- Acceleration approach: use GPU libraries (e.g., cuBLAS and cuSOLVER)



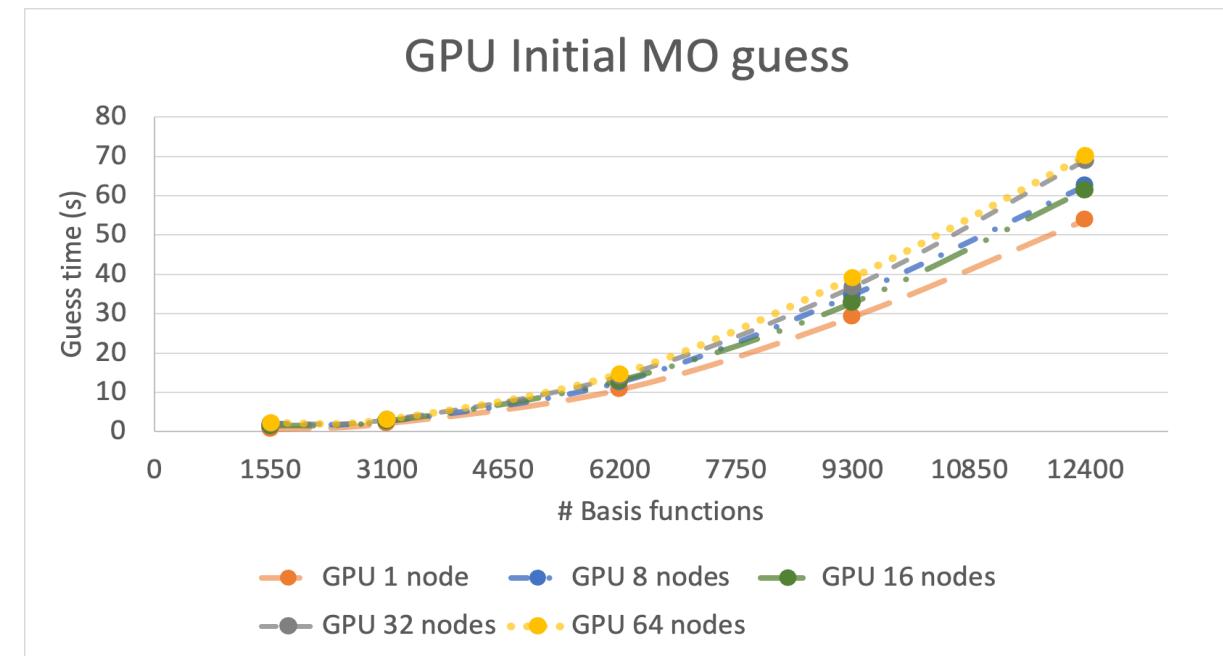
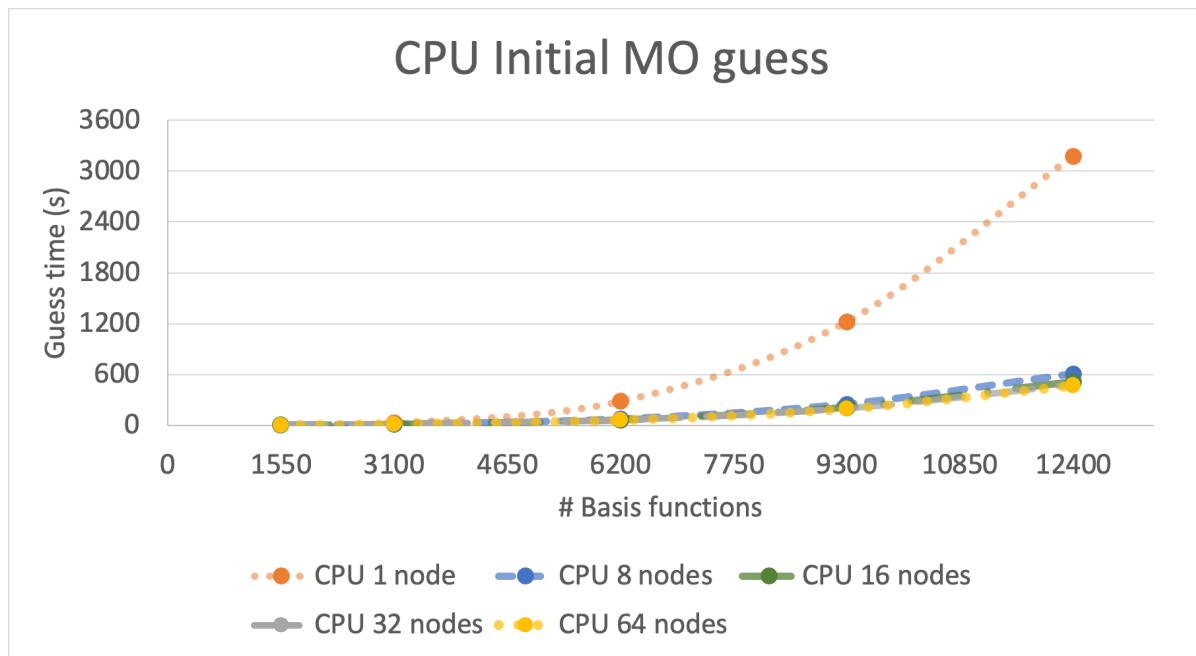
# Offloaded vs. Non-Offloaded Code

- Original CPU code uses BLAS calls for matrix operations
- Offloaded code contains a Fortran interface to the C-based cuBLAS and cuSOLVER libraries
- Conditional check to determine if CPU or GPU approach is taken

```
module offloaded_guess_mod
...
subroutine offloaded_DGEMM(matrices)
use iso_c_binding,omp_lib
...
interface to cuBLAS
...
!$omp target data map (tofrom: matrices)
    call cuBLAS_DGEMM(matrices)
!$omp end target data
```

```
subroutine M0_GUESS
use offloaded_guess_mod
...
if (offloading_enabled) then
    call offloaded_DGEMM(matrices)
else
    call DGEMM(matrices)
end if
...
```

# Offloading Results (Initial Guess)



- Calculations performed on Perlmutter with 128 logical cores per node
- GPU runs done with 4 MPI ranks per node, 16 threads per rank
- CPU runs done with 8 MPI ranks per node, 8 threads per rank

# Resolution-of-Identity Møller-Plesset Second-Order Perturbation Theory (RI-MP2)

Lead: Buu Q. Pham (Ames National Laboratory)

Compute:  $N^5$   
Memory:  $N^4$

# Møller-Plesset Second-Order Perturbation Theory (MP2)

The MP2 correlation energy is:

$$E^{(2)} = \sum_{i>j}^{\text{occ.}} (2 - \delta_{ij}) \sum_{ab}^{\text{vir.}} \frac{(ia|jb)[(ia|jb) - (ib|ja)]}{\varepsilon_i + \varepsilon_j - \varepsilon_a - \varepsilon_b}$$

where the MO-based four-index two-electron repulsion integrals (4-2ERIs) are,

$$(ia|jb) = \sum_{\sigma}^N C_{\sigma b} \sum_{\nu}^N C_{\nu a} \sum_{\lambda}^N C_{\lambda j} \sum_{\mu}^N C_{\mu i} (\mu\nu|\lambda\sigma)$$

and AO-based 4-2ERIs are,

$$(\mu\nu|\lambda\sigma) = \iint d\mathbf{r}_1 d\mathbf{r}_2 \phi_{\mu}^*(\mathbf{r}_1) \phi_{\nu}(\mathbf{r}_1) \frac{1}{|\mathbf{r}_1 - \mathbf{r}_2|} \phi_{\lambda}^*(\mathbf{r}_2) \phi_{\sigma}(\mathbf{r}_2)$$

# Resolution-of-Identity MP2 (RI-MP2)

$$(ia|jb) \approx \sum_P^X B_{ia}^P B_{jb}^P$$

$$B_{ia}^P = \sum_R^X (ia|R) W_{RP}^{-\frac{1}{2}}$$

$$(ia|P) = \sum_{\mu}^{NB} C_{\mu i} \sum_{\nu}^{NB} C_{\nu a} (\mu\nu|P)$$

$$(\mu\nu|P) = \iint d\mathbf{r}_1 d\mathbf{r}_2 \phi_{\mu}^*(\mathbf{r}_1) \phi_{\nu}(\mathbf{r}_1) \frac{1}{|\mathbf{r}_1 - \mathbf{r}_2|} \alpha_P(\mathbf{r}_2)$$

$$W_{PQ} = \iint d\mathbf{r}_1 d\mathbf{r}_2 \alpha_P(\mathbf{r}_1) \frac{1}{|\mathbf{r}_1 - \mathbf{r}_2|} \alpha_Q(\mathbf{r}_2)$$

- 4-2ERI is approximated with 2-2ERIs and 3-2ERIs
- Four main wall-time components
  - Evaluation of 2-2ERIs
  - Evaluation of 3-2ERIs
  - Contracting 3-2ERI with Cholesky vectors of the inverse 2-2ERI matrix
  - Distributed memory operations and data transfers

# Fortran Interfaces to Accelerated Libraries

```
1 module cublasf
2   use, intrinsic :: iso_c_binding
3   type(c_ptr)    :: cublas_handle
4   enum, bind(c)
5     enumerator :: CUBLAS_OP_N = 0
6     enumerator :: CUBLAS_OP_T = 1
7     enumerator :: CUBLAS_OP_C = 2
8   end enum
9   integer(c_int) function cublasXtDgemm &
10     (handle,transa,transb,           &
11      m,n,k,alpha,dA,ldda,dB,lddb,&
12      beta,dC,lddc)           &
13      bind(c, name="cublasXtDgemm")
14   type(c_ptr), value :: handle
15   integer(c_int), value :: transa
16   integer(c_int), value :: transb
17   integer(c_int), value :: m
18   integer(c_int), value :: n
19   integer(c_int), value :: k
20   real(c_double) :: alpha
21   real(c_double),dimension(*) :: dA
22   integer(c_int), value :: ldda
23   real(c_double),dimension(*) :: dB
24   integer(c_int), value :: lddb
25   real(c_double) :: beta
26   real(c_double),dimension(*) :: dC
27   integer(c_int), value :: lddc
28   end function cublasXtDgemm
29 end module
```

```
1 !$omp target data map(alloc:QVV)
2 !$omp loop over all slices of BI, BJ
3 !$omp target data map(to:BI,BJ)
4 !$omp target data use_device_ptr(BI,BJ,QVV)
5   cublas_return = cublasXtDgemm &
6     (cublas_handle,CUBLAS_OP_T,CUBLAS_OP_N, &
7      NVIR*iQVV,NVIR,NAUXBASD, &
8      one,BI,NAUXBASD, &
9      BJ,NAUXBASD, &
10     zer,QVV,NVIR*iQVV)
11   cublas_return = cudaDeviceSynchronize()
12 !$omp end target data
13 !$omp end target data
14 !$omp end target data
```

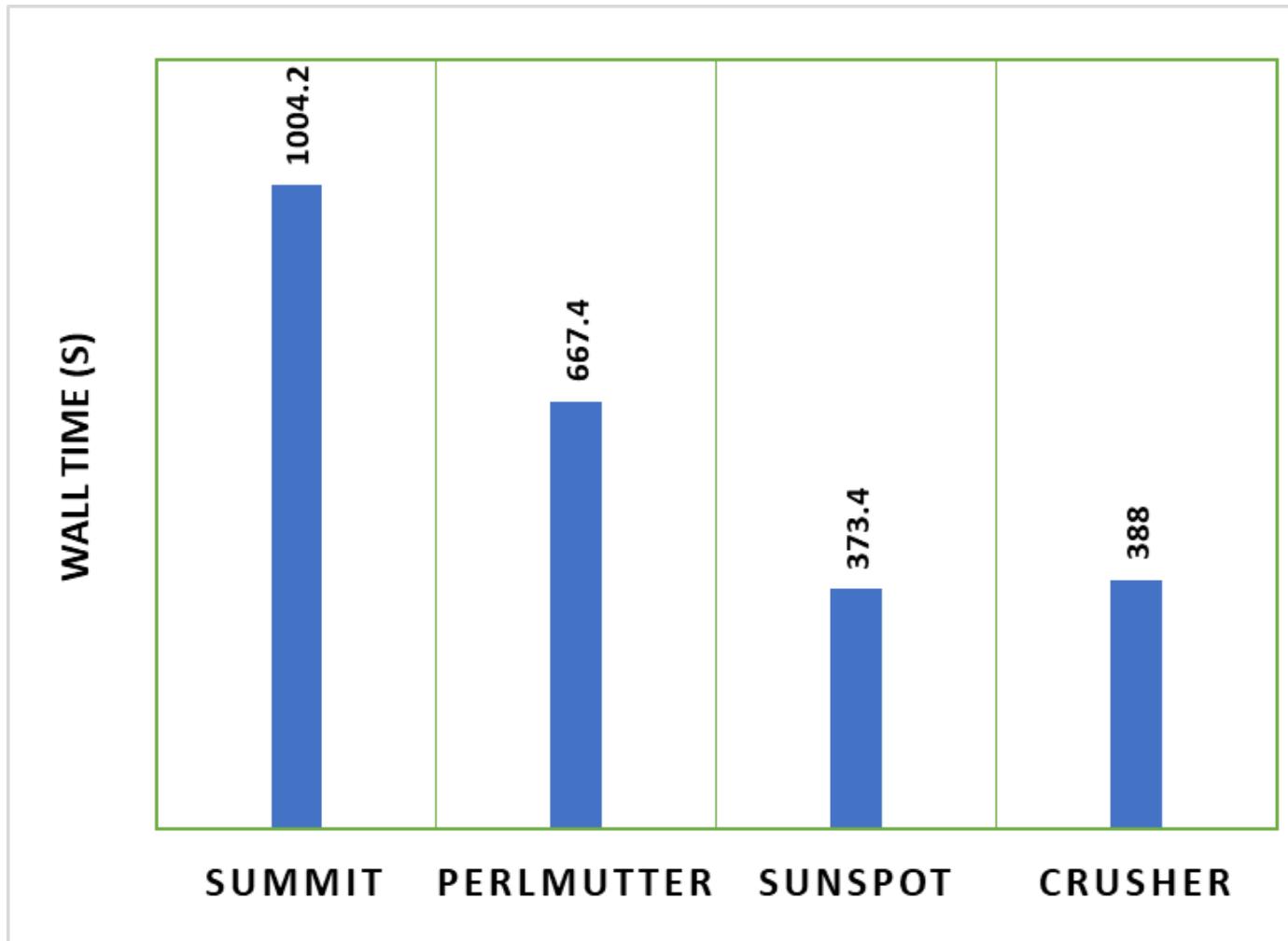
```
1 !$omp target data map(alloc:QVV)
2 !$omp loop over all slices of BI, BJ
3 !$omp target data map(to:BI, BJ)
4 !$omp target data use_device_ptr(BI,BJ,QVV)
5   call hipblasCheck(hipblasDgemm(
6     HIPBLAS_handle,      & ! type(c_ptr), value
7     HIPBLAS_OP_T,        & ! integer(kind(hipblas_op_n)), value
8     HIPBLAS_OP_N,        & ! integer(kind(hipblas_op_n)), value
9     NVIR*iQVV,          & ! integer(c_int), value
10    NVIR,                & ! integer(c_int), value
11    NAUXBASD,            & ! integer(c_int), value
12    one,                 & ! real(c_double)
13    c_loc(BI(1)),        & ! type(c_ptr), value
14    NAUXBASD,            & ! integer(c_int), value
15    c_loc(BJ(1)),        & ! type(c_ptr), value
16    NAUXBASD,            & ! integer(c_int), value
17    zero,                & ! real(c_double)
18    c_loc(QVV(1,1,1)),   & ! type(c_ptr), value
19    NVIR*iQVV           & ! integer(c_int), value
20  ))
21   call hipCheck(hipDeviceSynchronize())
22 !$omp end target data
23 !$omp end target data
24 !$omp end target data
```

# Single Node Capability

- Summit (512 GB/node):
  - ~5,000 atomic orbitals && ~15,000 auxiliary basis functions
- Perlmutter (256 GB/node):
  - ~4,000 atomic orbitals && ~12,000 auxiliary basis functions
- Aurora (1024 GB/node)\*:
  - >>5000 atomic orbitals && >>15,000 auxiliary basis functions
- Frontier (512 GB/node):
  - >5000 atomic orbitals && >15,000 auxiliary basis functions

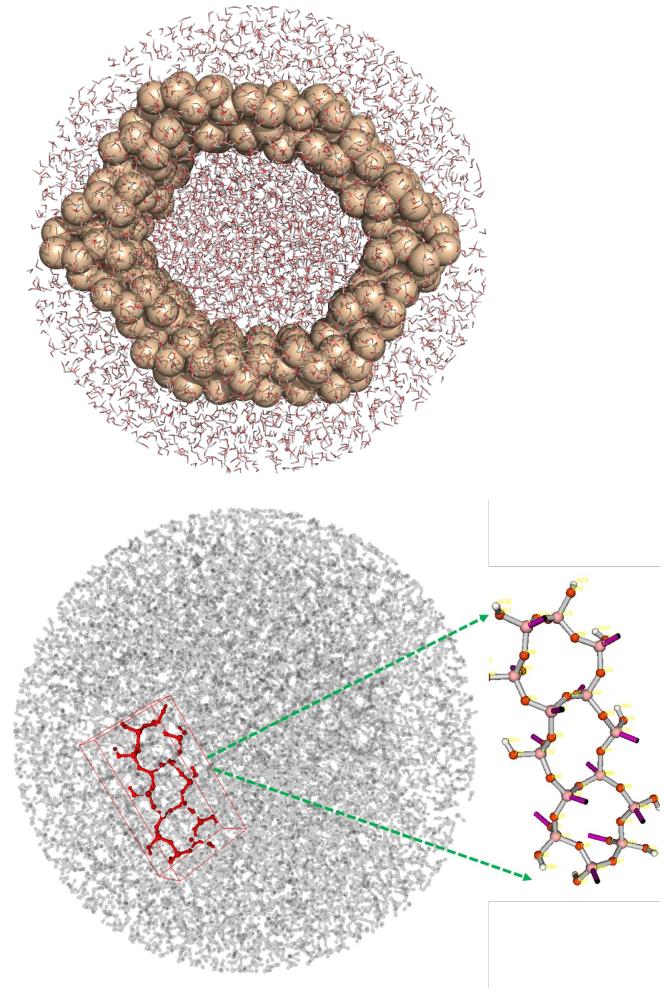
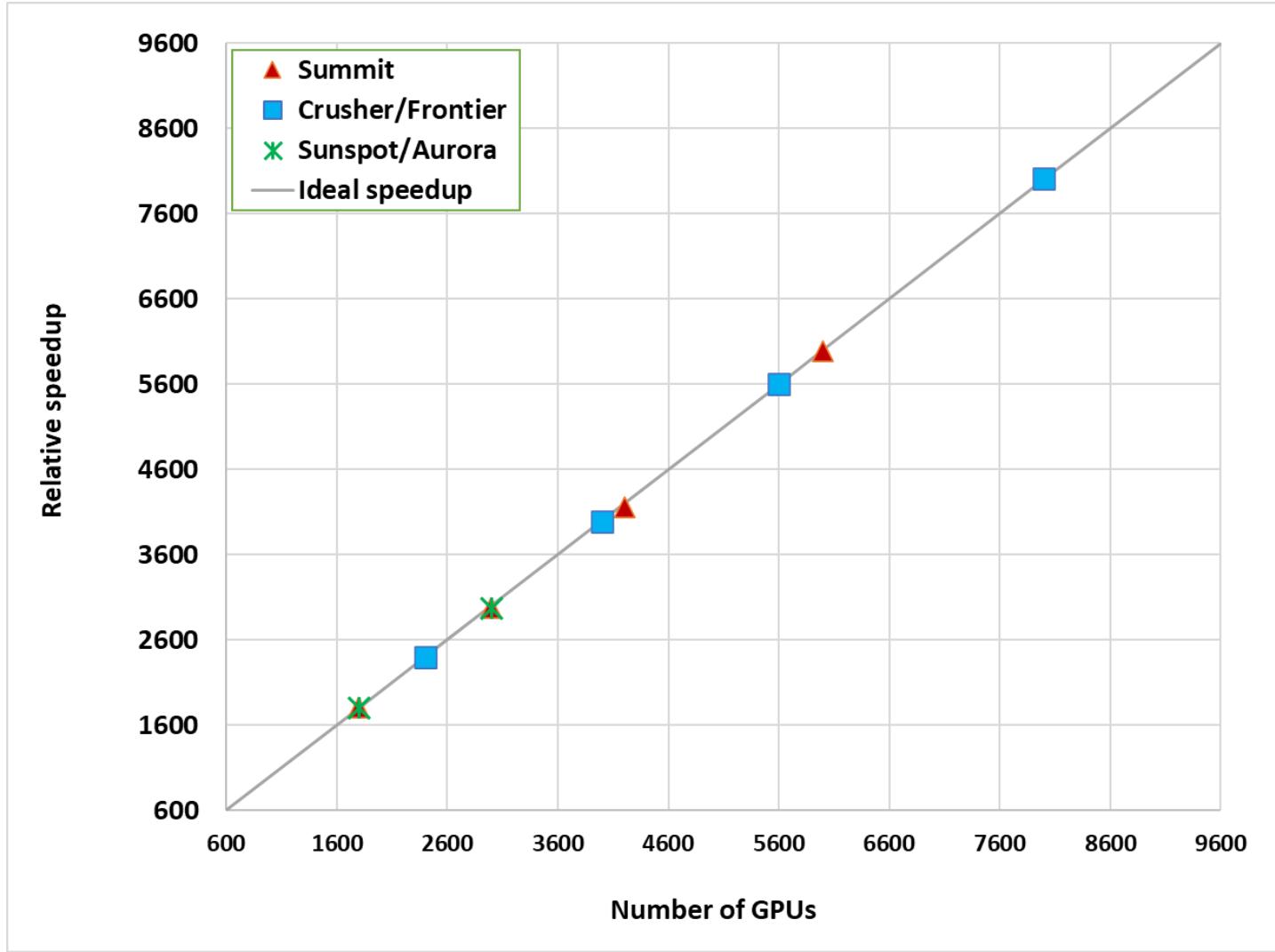
\*<https://www.alcf.anl.gov/aurora>

# Relative Timing Across Different GPUs



- $(H_2O)_{139}$
- 6-31G(d,p)
- Computed on one node
  - Summit: 6 V100s
  - Perlmutter: 4 A100s
  - Sunspot: 6 Intel GPUs
  - Crusher: 4 MI250Xs (8 GCDs)

# EFMO/RIMP2 Strong Scaling



- $MSN @ (H_2O)_{4597}$
- 6-31G/cc-pVDZ
- 74,277 basis functions
- Rcut=1.0
- 9,556 dimers
- 13,791 atoms

# Fragmentation Methods

Fragment Molecular Orbital (FMO)

Effective Fragment Potential (EFP)

Effective Fragment Molecular Orbital (EFMO)

Leads: Peng Xu and Tosaporn Sattasathuchana (Ames National Laboratory)

# Fragment Molecular Orbital (FMO)

# Fragment Molecular Orbital (FMO) Method

$$E \cong \sum_I^{n_F} E_I + \sum_{I < J}^{n_F} \Delta E_{IJ} + \sum_{I < J < K}^{n_F} \Delta E_{IJK} + \dots$$

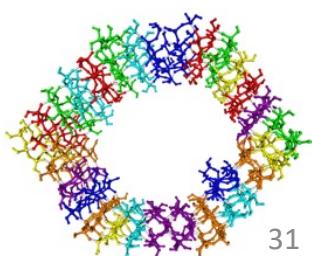
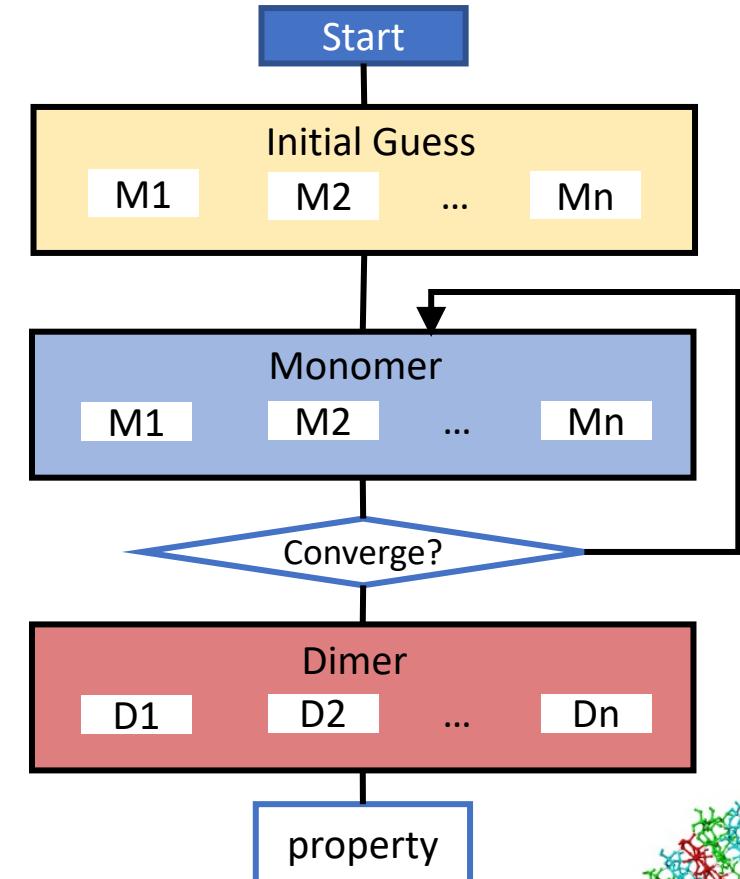
**Many-body expansion of the total energy**

$$\Delta E_{IJ} = E_{IJ} - E_I - E_J$$

**E(2-body interaction)**

$$\begin{aligned}\Delta E_{IJK} \\ &= E_{IJK} - E_I - E_J - E_K - (E_{IJ} - E_I - E_J) \\ &\quad - (E_{JK} - E_J - E_K) - (E_{IK} - E_I - E_K)\end{aligned}$$

**E(3-body interaction)**



# Effective Fragment Potential (EFP)

# Effective Fragment Potential (EFP) Method

- *ab initio* force field method designed to capture intermolecular interactions
- $E^{interaction} = E^{Coul} + E^{Pol} + E^{Disp} + E^{EXREP} + E^{CT}$
- Includes many-body polarization: induced dipoles are iterated to self-consistency

EFP components	Required Parameters	
Coulomb ( <i>Coul</i> )	Multipole moments up to octopoles located at atom centers and bond midpoints	
Polarization ( <i>Pol</i> )	Static dipole polarizability tensors distributed at localized MO (LMO) centroids	⇐ CPHF
Dispersion ( <i>Disp</i> )	Imaginary frequency dependent polarizability tensors distributed at LMO centroids	⇐ TDHF
Exchange Repulsion ( <i>EXREP</i> )	LMOs and LMO centroid coordinates, Fock matrix in LMO basis	
Charge Transfer ( <i>CT</i> )	Occupied + virtual molecular orbitals, eigenvalues of Fock matrix	

# Effective Fragment Molecular Orbital (EFMO)

# Effective Fragment Molecular Orbital (EFMO) Method

$$E^{EFMO} = \sum_I E_I^0 + \sum_{I>J}^{R_{IJ} < R_{cut}} [(E_{IJ}^0 - E_I^0 - E_J^0) - E_{IJ}^{pol}] + \sum_{I>J}^{R_{IJ} > R_{cut}} (E_{IJ}^{Coul} + E_{IJ}^{disp} + E_{IJ}^{ExRep} + E_{IJ}^{CT}) + E_{tot}^{pol}$$

- Evaluate monomer energy in vacuo
- QM dimers: proximate dimer treated by QM method
- EFP dimers: distant dimer treated by EFP
- Many-body effect captured by EFP polarization

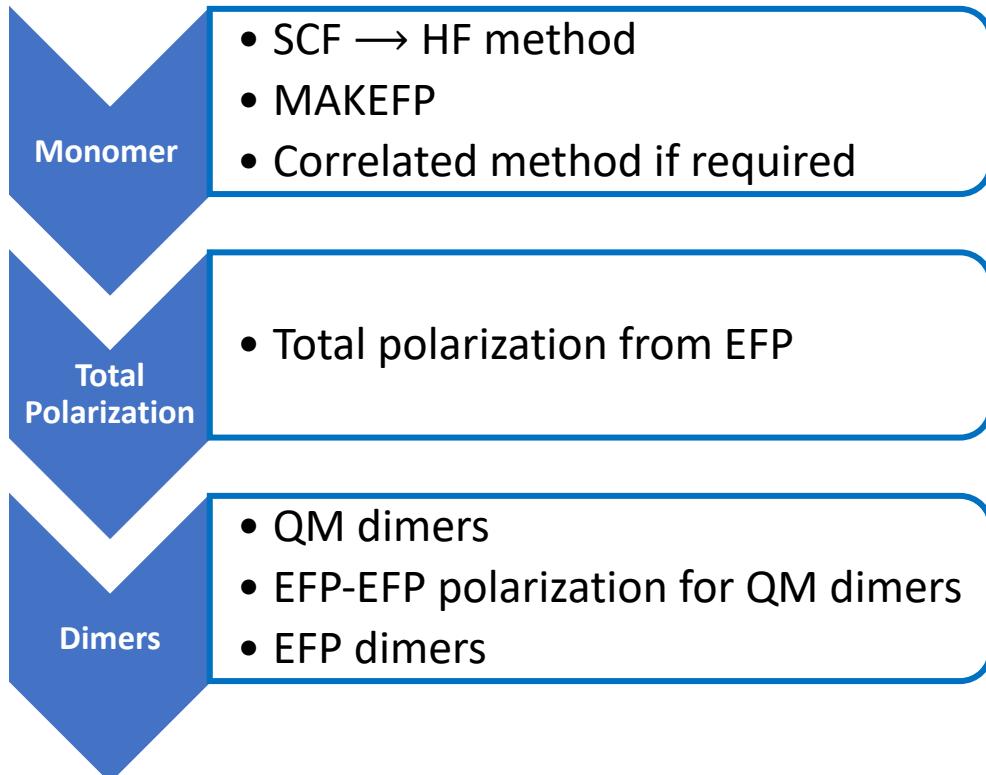
$$R_{IJ} = \min_{i \in I, j \in J} \left\{ \frac{|\vec{r}_i - \vec{r}_j|}{r_i^{\nu dw} + r_j^{\nu dw}} \right\}$$

Relative minimum distance between fragments I and J

$R_{cut}$ : A user-defined, distance-related, dimensionless threshold value. The larger the  $R_{cut}$ , the more dimers are treated as QM dimers

# Effective Fragment Molecular Orbital (EFMO) Method

## EFMO Calculation Overview



## Approach

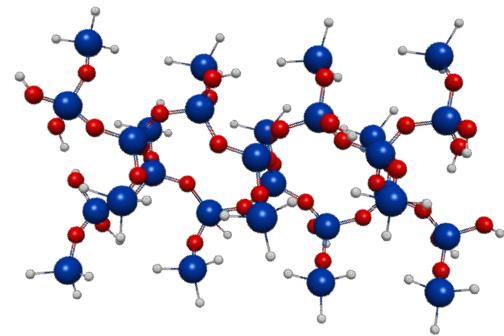
- Offload EFMO bottlenecks
  - SCF
  - MAKEFP (CPHF and TDHF)
  - Correlated method (e.g., RI-MP2)
- Keep on the CPU
  - EFP dimer
  - EFP polarization

# Offloading HF/MAKEFP (CPHF and TDHF)

- CPHF and TDHF
  - Common to HF and MAKEFP
- Bottleneck
  - Evaluation of ERIs
    - Some integral subroutines contain >1,000 lines of code.
    - Contraction with density matrix (DA) to form Fock matrix (FA)
- Modernize and restructure Fortran77 code
  - Remove common blocks
  - Remove large intermediate arrays

```
!$omp target distribute parallel do &
!$omp shared(FA,DA) private(ERIs) &
!$omp reduction(+:nintn)
do i,j,k,l
    call integral_j01(i,j,k,l,ERIs)
    call dirfck_rhf(i,j,k,l,ERIs,nintn,DA,FA)
```

```
subroutine dirfck_rhf(i,j,k,l,ERIs,nintn,DA,FA)
!$omp atomic update
    FA(IJ) = FA(IJ)+4*ERIs(ijkl)*DA(KL)
!$omp atomic update
    FA(KL) = FA(KL)+4*ERIs(ijkl)*DA(IJ)
!$omp atomic update
    FA(IK) = FA(IK)-ERIs(ijkl)*DA(JL)
!$omp atomic update
    FA(JL) = FA(JL)-ERIs(ijkl)*DA(IK)
!$omp atomic update
    FA(IL) = FA(IL)-ERIs(ijkl)*DA(JK)
!$omp atomic update
    FA(JK) = FA(JK)-ERIs(ijkl)*DA(IL)
```



# Offloading Results (HF/MAKEFP)

Using 1GPU	Time from miniERIs (an iteration of RHF method) (s)						
	Summit		Perlmutter		Frontier		Sunspot
Kernel	xlf/ 16.1.1-10	nvhpc/ 22.5	nvhpc/ 22.7	cce/ 15.0.1	amd/ 5.4.3	cce/16.0.0 rocm5.5.1	oneAPI
j01	0.26	0.03	0.02	0.01	0.02	0.01	0.01
j02	1.53	0.16	0.09	0.07	0.24	0.09	0.06
j03	1.98	0.15	0.08	0.2	0.72	0.16	0.12
j04	3.88	0.24	0.15	1.98	0.85	4.74	0.35
j05	16.96	1.09	0.54	5.95	15.8	20.88	1.98
j06	79.33	8.92	4.52	5.77	22.44	19.32	3.00
<b>Sum</b>	<b>103.94</b>	<b>10.59</b>	<b>5.4</b>	<b>13.98</b>	<b>40.07</b>	<b>45.2</b>	<b>5.52</b>

# Portability of Offloaded Code

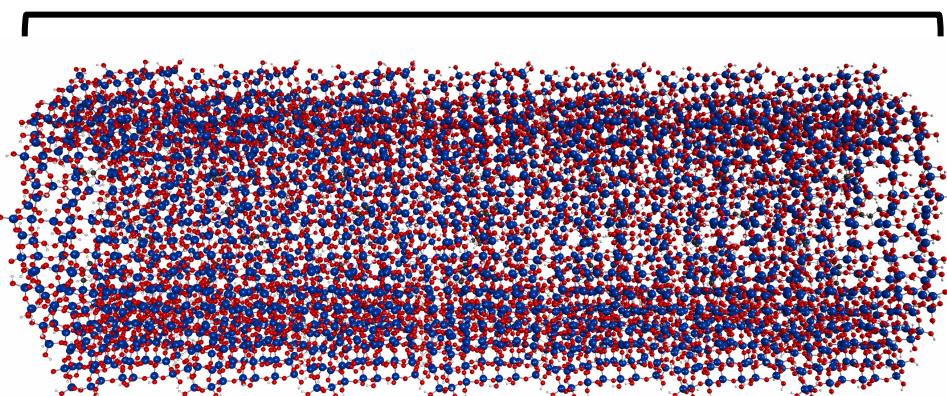
<b>Accelerator</b>	V100	A100	MI250X	PVC
<b>Compiler</b>	NVHPC	NVHPC CCE	CCE	oneAPI
<b>Center</b>	OLCF Summit	ALCF Polaris	OLCF Frontier	ALCF Aurora

# ECP Science Challenge

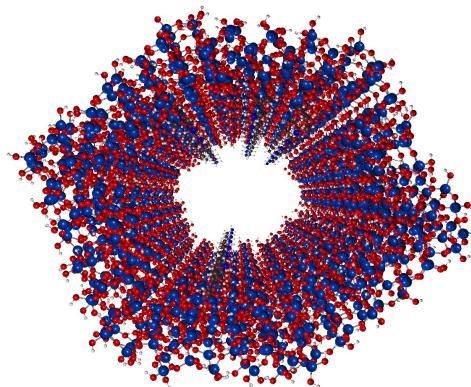
Leads: Peng Xu, Tosaporn Sattasathuchana, and Buu Q. Pham (Ames National Laboratory)

# ECP Science Challenge

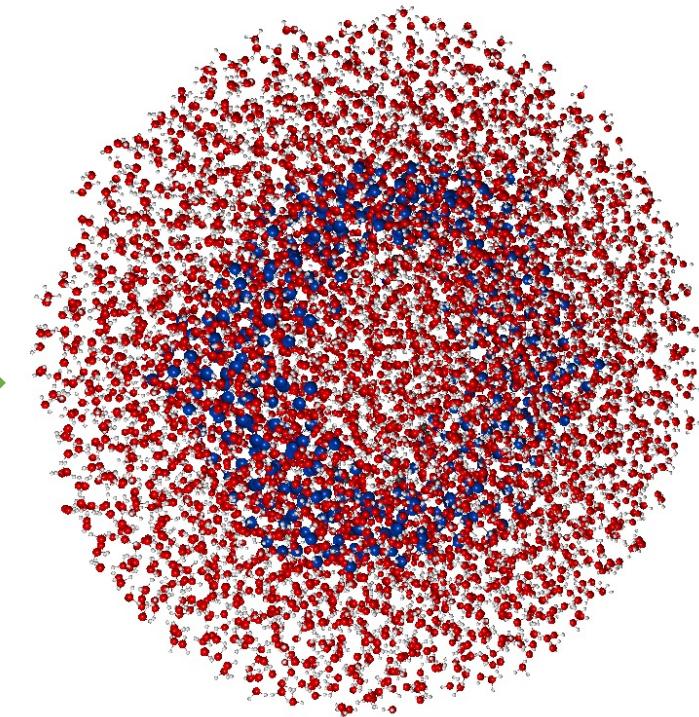
Stretch problem ~150,000 basis functions



~



Solvated



Base problem ~15,000 basis functions

Calculations of real heterogeneous catalysts  
at HF/RI-MP2/RI-CCSD(T) level of accuracy

# ECP Science Challenge

Stretch problem ~150,000 basis functions

High scaling methods

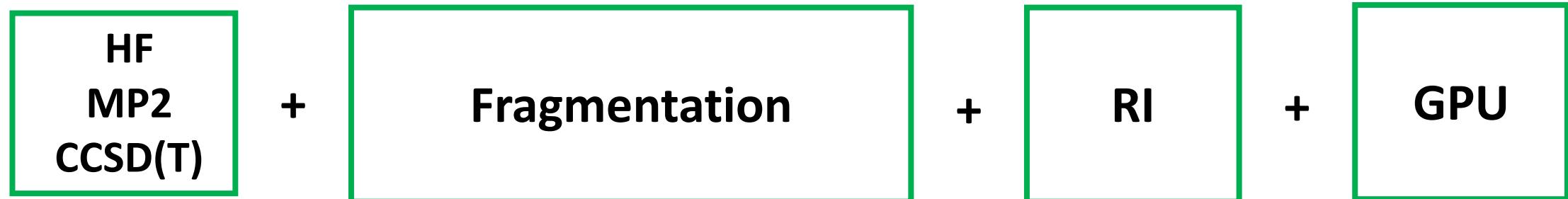
- Huge computational costs
- Enormous memory demands
- Scalability in practice (handling fault tolerance)

Base problem ~150 basis functions

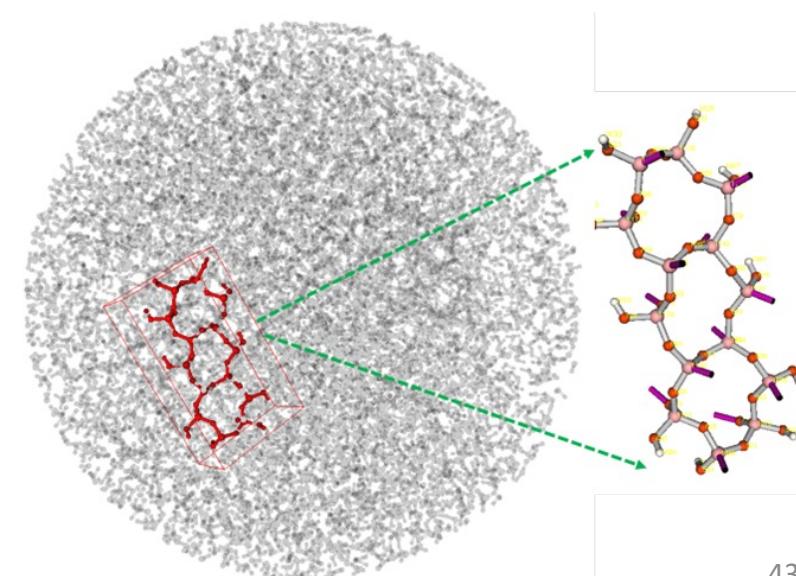
Solvated

Calculations of real heterogeneous catalysts  
at HF/RI-MP2/RI-CCSD(T) level of accuracy

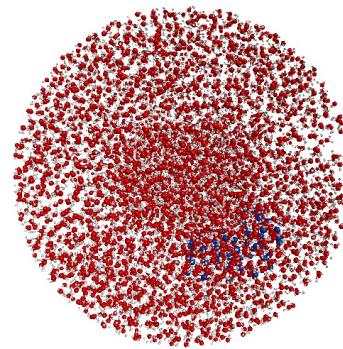
# Approach



- Capture relevant QM effects
- Provide ideal scaling framework
- Preserve accuracy of underlying QM methods
- Reduce memory footprint
- Provide ideal data structure for GPU offloading
- Handling large numerical operation
- Enhance accuracy



# Offloading Results (Frontier)



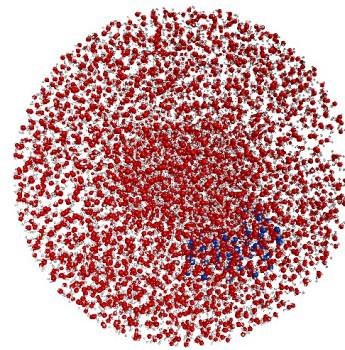
EFMO/RI-MP2/6-31G/cc-pVDZ-ri

**CPU and CPU+GPU timings comparisons for MSN5-HYD on Frontier (128 nodes)**

Run	Rcut	Best FMO Energy (Hartree)	Total Wall Time (min)	Speed-up (x)
CPU	1	-382235.178290952	133.4	
CPU+GPU	1	-382235.178290833	29.3	4.6
CPU	2	-382236.589264343	174.0	
CPU+GPU	2	-382236.589266623	35.5	4.9

cpe/22.12, cce/15.0.0, rocm/5.3.0  
128 nodes, 16 MPI ranks per node, 6 threads per rank

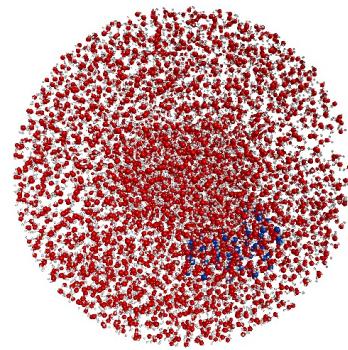
# Kernel Timings on Frontier (ROCPROF)



EFMO/RI-MP2/6-31G/cc-pVDZ-ri

Kernel and HIP API timing statistics for ranks 0-7 of a 128 node run of MSN5-HYDR on Frontier							
Rank	Timings (s)						
	Kernels					HIP API Calls	
	RHF	CPHF	TDHF	RIMP2	DGEMM	HtoD	DtoH
0	114.9	50.6	620.9	10.4	44.5	44.5	27.7
1	111.5	53.7	662.2	13.1	50.1	50.1	27.9
2	107.3	52.3	633.4	16.2	51.2	51.2	27.9
3	93.2	44.2	506.0	18.8	59.4	59.4	28.1
4	96.7	44.9	505.2	21.2	57.1	57.1	26.4
5	100.8	46.5	550.5	22.7	61.9	61.9	27.4
6	98.7	45.4	512.0	24.1	65.8	65.8	26.7
7	101.5	50.5	563.1	25.8	69.5	69.5	27.5
Average	103.1	48.5	569.2	19.0	57.4	57.4	27.5

# Kernel Timings on Frontier (ROCPROF)



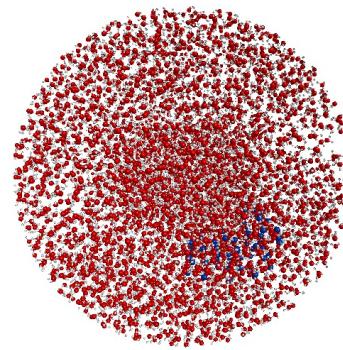
EFMO/RI-MP2/6-31G/cc-pVDZ-ri

Kernel and HIP API timing statistics for ranks 0-7 of a 128 node run of MSN5-HYDR on Frontier							
Rank	Timings (s)						
	Kernels					HIP API Calls	
	RHF	CPHF	TDHF	RIMP2	DGEMM	HtoD	DtOH
0	620.9	10.4	44.5	44.5	27.7		
1	662.4	18.1	50.1	50.1	27.9		
2	512.3	31.2	51.2	51.2	27.9		
3	597.4	59.4	59.4	59.4	28.1		
4	571.1	57.1	57.1	57.1	26.4		
5	100.8	46.5	550.5	22.7	61.9	61.9	27.4
6	98.7	45.4	512.0	24.1	65.8	65.8	26.7
7	101.5	50.5	563.1	25.8	69.5	69.5	27.5
Average	103.1	48.5	569.2	19.0	57.4	57.4	27.5

## GPU Utilization

- 1760.7 s for non-profiled run on 128 nodes
- 45% of total wall-time spent in kernels (797 s)
- 51% of total wall-time if DT time is included

# Offloading Results (Aurora) Preliminary

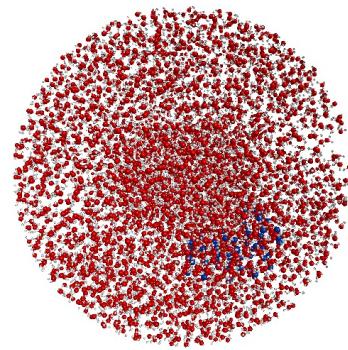


EFMO/RI-MP2/6-31G/cc-pVDZ-ri

CPU and CPU+GPU timings comparisons for MSN5-HYD on Aurora (128 nodes)				
Run	Rcut	Best FMO Energy (Hartree)	Total Wall Time (min)	Speed-up (x)
CPU	1	-382235.178290892	48.7	
CPU+GPU	1	-382235.178290870	15.0	3.2
CPU	2	-382236.589265166	62.4	
CPU+GPU	2	-382236.589265446	18.5	3.4

Contact Colleen Bertoni (Argonne) for run details

# Offloading Results (Polaris)



EFMO/RI-MP2/6-31G/cc-pVDZ-ri

**CPU and CPU+GPU timings comparisons for MSN5-HYD on Polaris (128 nodes)**

Run	Rcut	Best FMO Energy (Hartree)	Total Wall Time (min)	Speed-up (x)
CPU	1	-382235.178290867	121.0	
CPU+GPU	1	-382235.178290870	17.8	6.8

nvhpc/23.3, CUDA 11.8

128 nodes, 8 MPI ranks per node, 6 threads per rank

# Density Functional Theory

Lead: Federico Zahariev (Ames National Laboratory)

# Electronic Schrödinger Equation (Revisited)

$$\hat{H}_{elec} \Psi_{elec} = E_{elec} \Psi_{elec}$$

$$\hat{H}_{elec} = -\frac{1}{2} \sum_{i=1}^N \nabla_i^2 - \sum_{i=1}^N \sum_{A=1}^M \frac{Z_A}{r_{iA}} + \sum_{i < j}^N \frac{1}{r_{ij}}$$

- $\Psi_{elec}$  is a function of  $4N$  variables where  $N$  is the number of electrons in system
- Hohenberg-Kohn theorem states that the ground state electronic energy is determined completely by the electron density (a function of 3 spatial variables),

$$\rho(\mathbf{r}) = N \int \int d\mathbf{s} d\mathbf{x}_2 \dots d\mathbf{x}_N |\Psi_{elec}(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N)|^2$$

The energy is a functional of the electronic density

R.G. Parr and W. Yang Density  
Functional Theory of Atoms and  
Molecules. Oxford University Press,  
New York, 1989

# Density Functional Theory

- Hohenberg-Kohn theorem establishes relationship between energy and electron density through the existence of a universal functional
- Universal functional is not known and gives rise to the development of density functional approximations (e.g., M06, PBE0,  $\omega$ B97-X)
- Kohn-Sham formalism of DFT enables leveraging existing machinery (HF) in quantum chemistry codes to perform a DFT calculation
  - Hamiltonian includes an exchange-correlation potential,

$$v_{xc} = \frac{\delta E_{xc}}{\delta \rho}$$

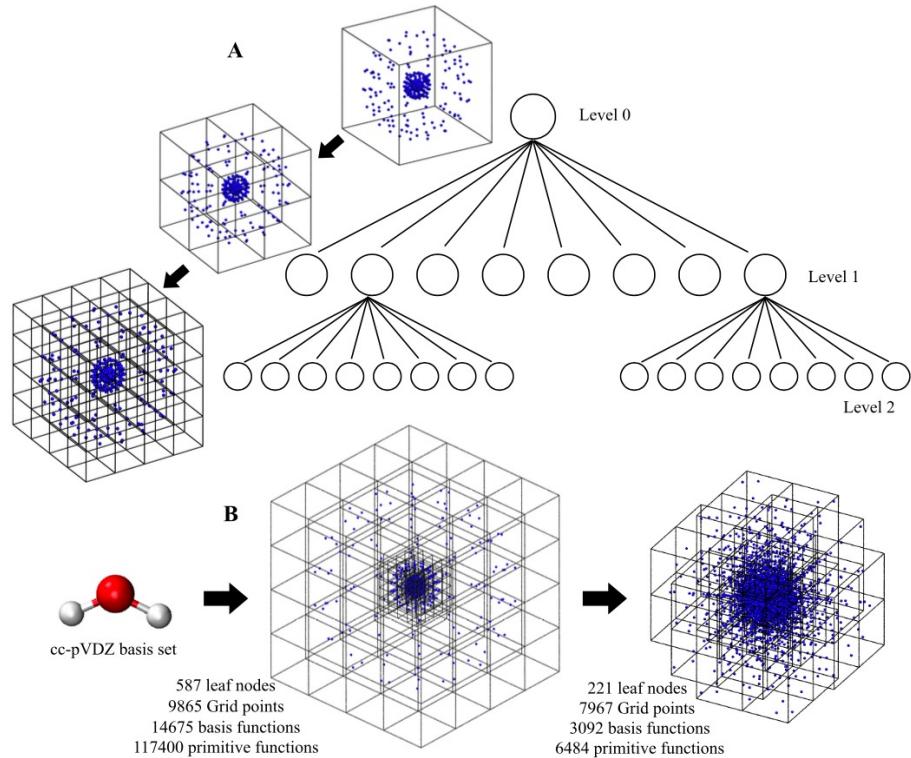
where,

$$E_{xc} = \underbrace{\int \rho(\mathbf{r}) f_{xc} (\rho_\alpha, \rho_\beta, \gamma_{\alpha\alpha}, \gamma_{\alpha\beta}, \gamma_{\beta\beta}, \tau_\alpha, \tau_\beta) d\mathbf{r}}_{\text{Density Functional Approximation}}$$

Solved using  
numerical  
quadrature

# Offload Exchange-Correlation (XC) Component of the Fock Matrix

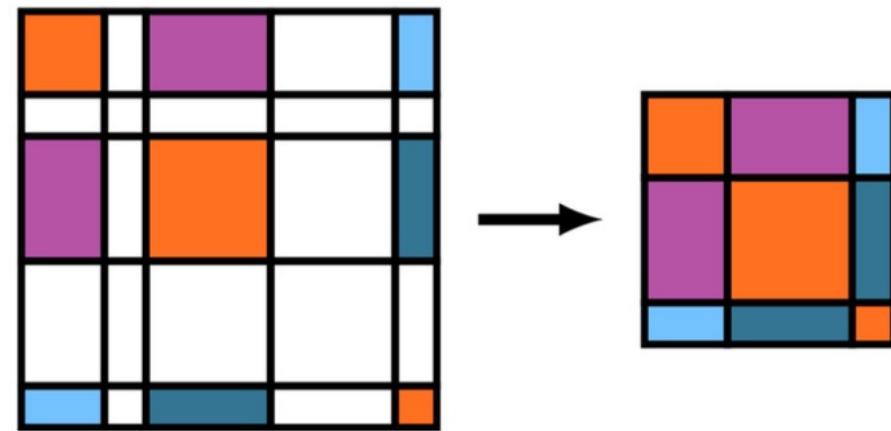
**Recursive partitioning of the XC grid:**



Data prep for DGEMM and DSYR2K to minimize DT

The figure is taken from M. Manathunga, Y. Miao, D. Mu, A. W. Götz, K. M. Merz, *J. Chem. Theory Comput.* 16, 4315 (2020).

**Batch compression of matrices due to the sparse grid:**



Batched (with variable dimensions) versions of DGEMM and DSYR2K (Magma library with HIP)

The figure is taken from D. B. Williams-Young, W. A. de Jong, H. J. J. van Dam and C. Yang, *Front. Chem.* 8, 581058 (2020).

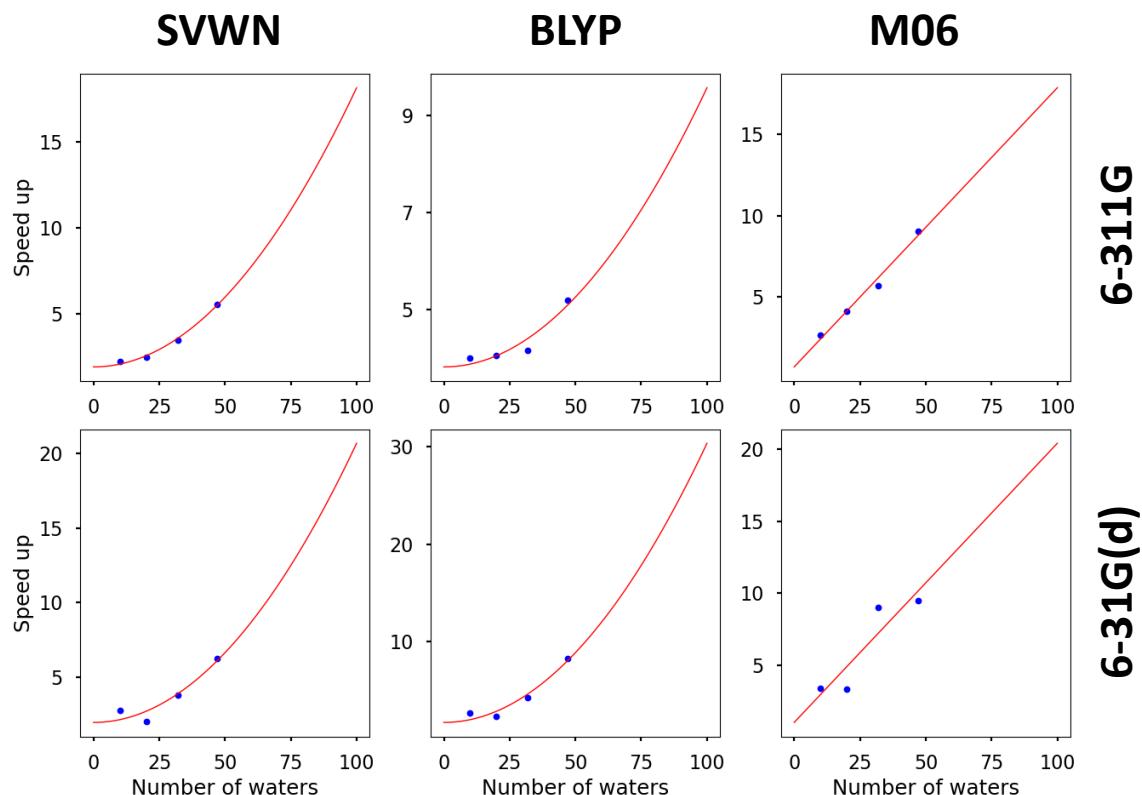
# Offloading Results (DFT) Preliminary

Speed-up vs. Number of Waters

Speed Up DFA/Basis Set	Number of Water Molecules			
	10	20	32	47
SVWN/6-311G	2.8	2.0	3.8	6.2
BLYP/6-311G	2.6	2.3	4.2	8.3
M06/6-311G	3.4	3.3	9.1	9.5
SVWN/6-31G(d)	2.2	2.5	3.4	5.6
BLYP/6-31G(d)	4.0	4.1	4.2	5.2
M06/6-31G(d)	2.6	4.1	5.7	9.1

Speed-up = Ratio of OpenMP and OpenMP Offload timing for XC component on 1 Summit node.

Extrapolations of the speed-ups



# Issues and ↗ Workarounds

# Issues and ↳ Workarounds

- Intel

- JIT is long (CMPLRLLVM-48607). ↳ Use AOT.
- Atomic update is very slow for Fortran (CMPLRLLVM-46117). ↳ Use C code.
- Issues with common blocks/modules (CMPLRLLVM-51444, incorrect result).  
↳ Pass common block as argument list

- CCE

- Only CCE needs '**simd**' clause to enable multithread parallelism. Code portability issue down the road.  
This will have conflict with Intel compiler.

- AMD flang

- NINT function is not implemented for GPU. ↳ Write the function manually.
- Can not do **reduction** for integer (compile error). ↳ Remove it as it does not affect the correctness of the answer.
- Can not have '**atomic update**' in a call.  
↳ manually inline or use an INCLUDE.

```
subroutine gpu_offload_rhf(FA,L1,L2)
use comm_B, only:: CO
use comm_NSHEL_sp, only: KMIN,KMAX

!$omp target distribute parallel do simd &
!$omp shared(CO,KMIN,KMAX) &
!$omp shared(FA) private(gpople) &
omp reduction(+:nintn)
do i,j,k,l
    call integral(i,j,k,l,gpople)
    call dirfck(i,j,k,l,gpople,nintn,FA)
```

Containing '\$omp atomic update'

```
call integral(i,j,k,l,gpople)
INCLUDE "dirfck.inc"
```

# Issues with Larger Kernels for SPD Integrals in GAMESS

- Intel
  - Issues with common blocks/modules (CMPLRLLVM-51444, incorrect result).  
  ~ passing as argument list.
  - Compile and link times using AOT > 2hours (CMPLRLLVM-48607).
  - Lots of spills for large kernels, which lead to incorrect answer (GSD-6123).  
  ~ IGC\_ForceOCLSIMDWidth=16
- CCE
  - Link time ~ 1 hour

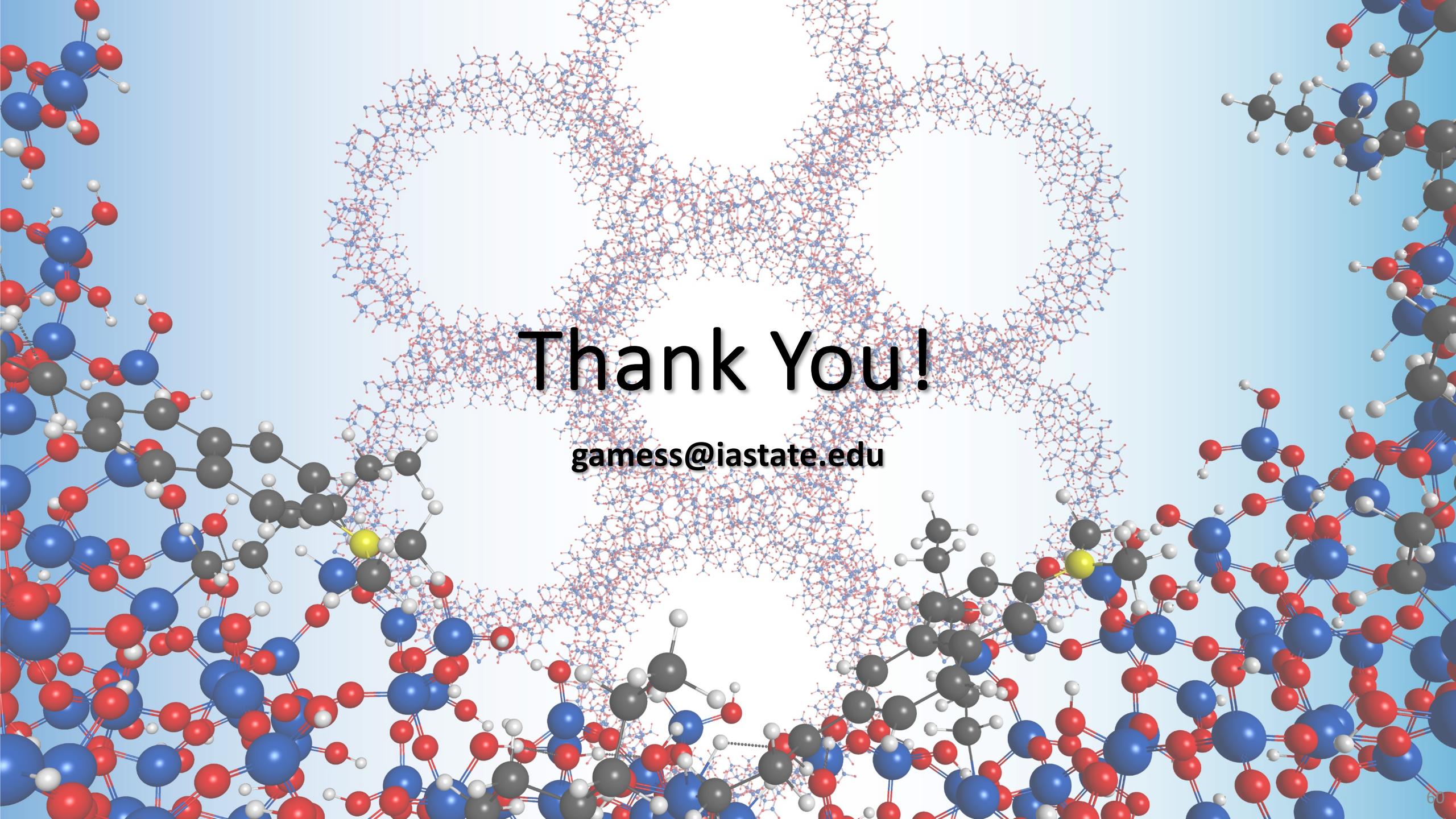
# Wish List

# Wish List

- OpenMP function to use shared '\$omp shared (...)'
- Dynamic scheduling for GPU offloading with OpenMP
- Reduction for arrays
- Unify across compiler vendors

# Outlook

- OpenMP offload focused GAMESS public release in November (SC23)
  - Summit, Polaris, Perlmutter, Frontier, Aurora support
  - <https://www.msg.chem.iastate.edu/gamess/download.html>
- Publications on OpenMP offload work coming soon!
  - RI-MP2
  - RI-CC (accepted)
  - EFMO
  - DFT



A circular arrangement of various molecular models, including protein-ligand complexes and DNA-like structures, set against a light blue background.

Thank You!

[gamess@iastate.edu](mailto:gamess@iastate.edu)

# Hidden Slides