BerkeleyGW

Mauro Del Ben

Computational Research Division, Lawrence Berkeley National Laboratory

July 3, 2019

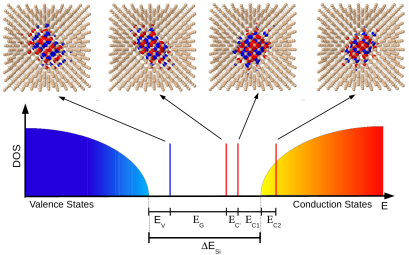
GPUs for Science Day

Accurate Optical and Electronic Properties of Complex Materials

Complex Materials: unique electronic and optical triggered by symmetry breaking

Important implications in many fields:

- Quantum Computing
- Energy Storage/Conversion
- Photovoltaics
- Nanoelectronics
- Catalysis



Example: schematic representation of the electronic structure of a divacancy in crystalline Silicon

Accurate predictions requires:

- Accuracy beyond standard (DFT) approaches $\rightarrow GW$
- System size beyond conventional simulations $\rightarrow$ Thousands of atoms

The GW Method

GW method represents the state of the art most effective and accurate approach to predict excited-state properties in a wide range of materials

Solve Dyson's equation:

$$\left[-\frac{1}{2}\nabla^2 + V_{\mathrm{Nuc}} + V_{\mathrm{H}} + \Sigma(E_n)\right]\phi_n = E_n\phi_n, \tag{1}$$

$\Sigma(E_n) \rightarrow$ self-energy (non-Hermitian, non-local, energy-dependent operator)

Application of GW to thousands atoms systems still a challenge $\rightarrow O(N^4)$

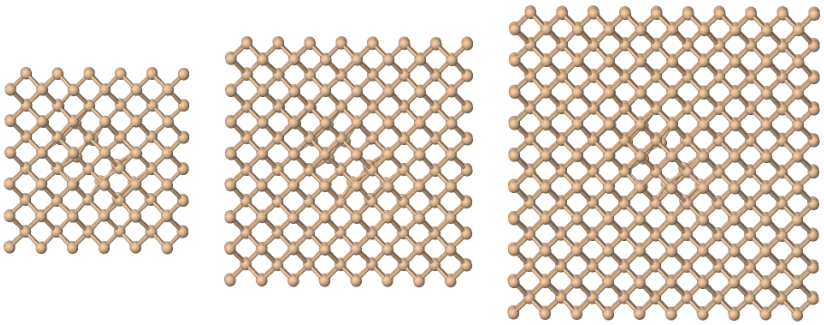
- Develop methods to reduce prefactor and scaling with system size
- Improve single node performance and parallel scalability

BerkeleyGW

BerkeleyGW is a massively parallel computational package for electron excited-state properties employing the *ab-initio* GW and GW plus Bethe-Salpeter equation

- Programming language: Mainly Fortran (over 100,000 lines of code)
- Parallelization: Hybrid MPI/OpenMP/GPU (Cuda)
- Libraries: BLAS, LAPACK, ScalaPACK, FFTW, ELPA, PRIMME, HDF5, cuBLAS, cuFFT
- Code structure: five major steps in the workflow
  - Each step has its own computational complexity / memory requirements
  - Five stand alone executable: `parabands`, `epsilon`, `sigma`, `kernel` and `absorption`
- Basic algorithmic kernels:
  - Large distributed matrix multiplication between tall and skinny matrices
  - Large distributed linear algebra (direct / iterative eigenproblems, matrix factorization, matrix inversion etc...)
  - Non-distributed fast Fourier transformations (FFT)
  - Dimensionality reduction and low-rank approximations

Benchmarks for Performance Measurement: Divacancy in Silicon

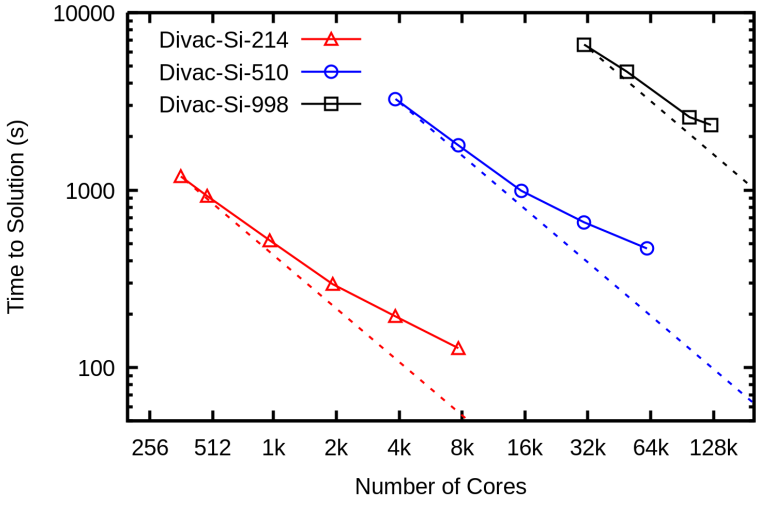


Three supercell, with 214, 510 and 998 atoms respectively allows for systematic assessment of performance across architectures in term of strong / weak scaling / time to solution and flop rate

Benchmarks for Performance Measurement: Problem Sizes

	Dvac-Si-214	Dvac-Si-510	Dvac-Si-998
N_G^{ψ}	31,463	74,653	145,837
N_G^{χ}	11,075	26,529	51,627
N_b	6,397	15,045	29,346
N_v	428	1,020	1,996
N_c	5,969	14,025	27,350
N_{eig}	3,500	7,000	14,000
N_ω	10	10	10
Epsilon Min PFlops	5.8	157.9	2335.7
Epsilon Min Memory (Tb)	0.6	7.7	57.5

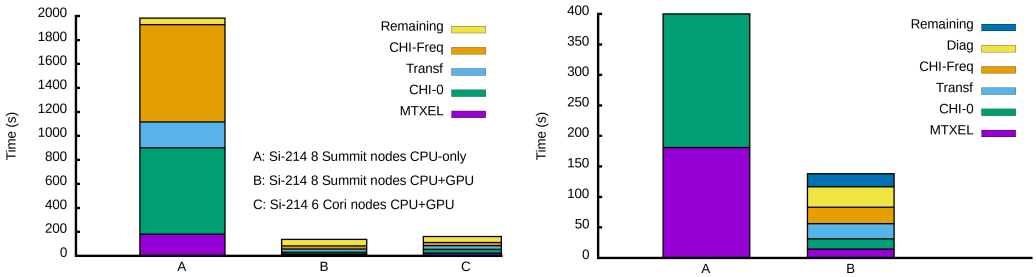
Baseline Performance: Epsilon on Edison@NERSC



Epsilon: Hybrid GPU-CPU Implementation

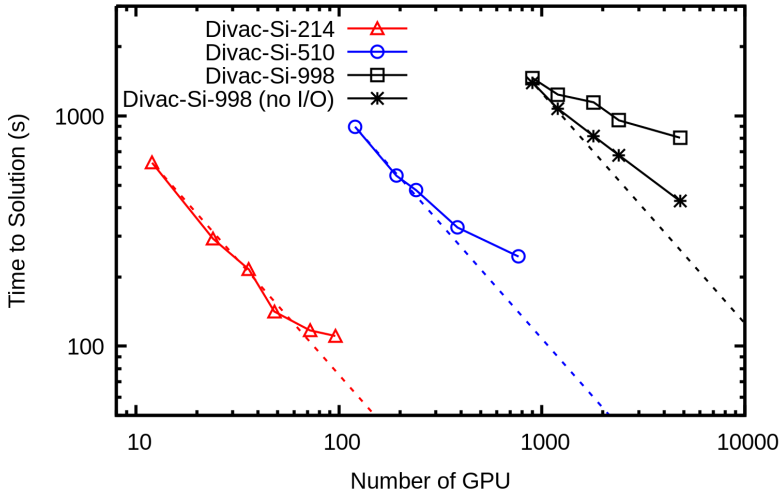
1. Matrix Elements (MTXEL): Unfavorable $O(N^3)$ vs $O(N^3 \log N)$ data/flops
   - cuFFT library $\rightarrow$ no benefit by just linking
   - Asynchronous data transfer $\rightarrow$ pinned host memory/data streams
2. Static Polarizability (CHI-0): Favorable $O(N^3)$ vs $O(N^4)$ data/flops
   - Use cuBLAS library $\rightarrow$ Asynchronous host to device data transfer
   - Non-blocking cyclic MPI communication scheme
   - Overlap CPU-communication/GPU-computation
3. Diagonalization (Diag): $O(N^3) \rightarrow$ ELPA
4. Basis Transformation (Transf): $O(N^4)$ memory bottlenecks for both host/device
   - Batch communication over eigenvectors $\rightarrow$ control host memory usage
   - Batch computation over wavefunctions $\rightarrow$ control device memory usage
5. Frequency Dependence (CHI-Freq): $O(N^4)$ multiple matrix multiplications
   - Smaller matrices ($N_G/N_b \simeq 5-10$) at multiple frequencies
   - Data streams over frequency index $\rightarrow$ allows for concurrent execution on device

Epsilon: CPU-Only *vs* Hybrid GPU-CPU (Summit@ORNL)

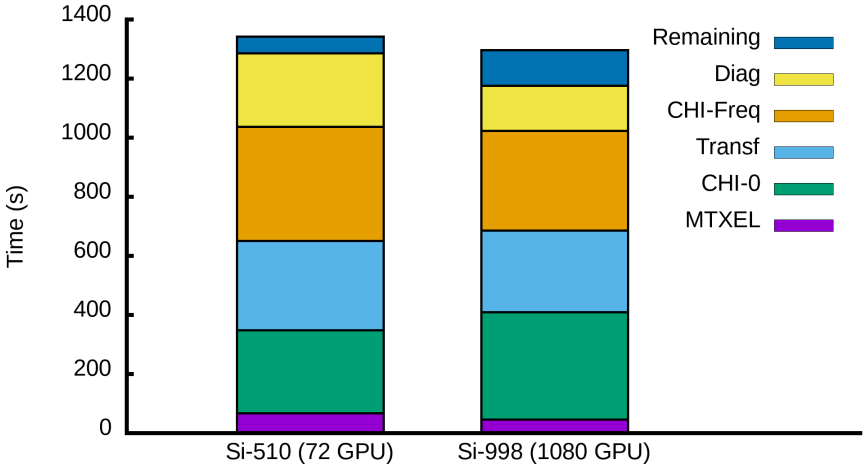


Summit node: 2 IBM POWER9 CPUs (21 cores each) and 6 NVIDIA V100 (Volta) GPUs, aggregate performance 42 TFlops. Cori-GPU node: 2 sockets of 20-core Intel Skylake + 8 NVIDIA Volta GPUs.

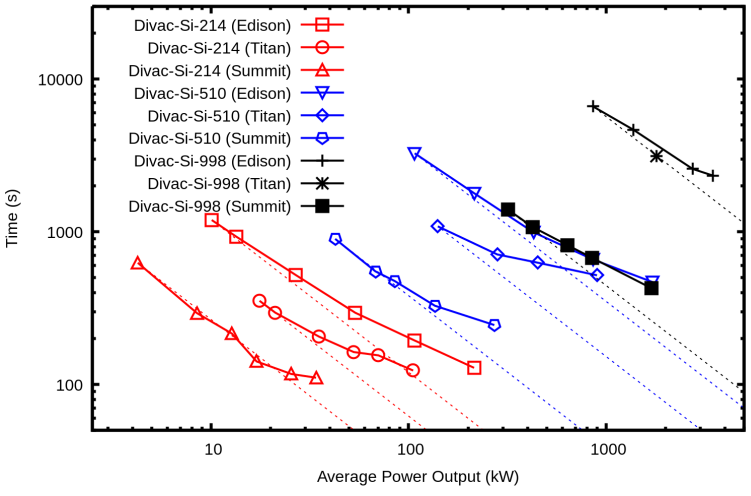
Summit@ORNL: Strong Scaling



Summit@ORNL: Weak Scaling



Comparison Across Architectures: Time *vs* Power



Average power per node (from Top500 website), Edison: 0.67 kW, Titan: 0.44 kW, Summit 2.12 kW.

Summary

GPU support in BerkeleyGW:

- $\times 10$ or more acceleration compare to CPU only runs
- Good strong / weak scaling
- Improved energy efficiency
- Excellent time to solution for systems made of thousands of atoms
- Working to extend portability to other BerkeleyGW modules