Acceleration of computational quantum chemistry by AVX + CUDA on heterogeneous computer architectures

Yuki Furukawa, Ryota Koga : X-Ability Co.,Ltd.

Koji Yasuda : Nagoya University

Background

1. Biologically important macromolecules, such as proteins, are within the scope of electronic-structure calculation by using FMO because of the improvement of computational resources.
2. AVX is effective on Intel SandyBridge Architecture. This is a first step for the era of MIC(Many Integrated Cores) computation.
3. GPGPU computation by CUDA is successfully applied for many kinds of problems in high performance computation.

Purpose of this Research

Based on Yasuda [2008], We develop a software library (XA-AVX/CUDA-QM) to accelerate famous QM softwares (GAMESS-US etc...) using both NVIDIA GPU and Intel SandyBridge CPU. We show the excellent performance of XA-AVX/CUDA-QM together with FMO. This implemetation is a preliminary preparation for the era of MIC/GPGPU computation.

Coding Policy

1. Total energy and the gradient of under FMO approximation.
2. Most time-consuming steps (the evaluation of Coulomb, Hartree-Fock exchange, and exchange-correlation matrices…) are executed on GPUs. Remaining parts are calculated on CPUs.
3. Use single precision arithmetic as much as possible for the sake of effective speedup, without degradation of accuracy.
4. Take special care for task scheduling and host-GPU communication time to keep as many CPUs and GPUs busy.
5. Modular design applicable to many quantum chemistry softwares

Algorithms

Electrostatic potential (Coulomb or J-matrix) :

The two-electron integrals and the J-matrix in terms of primitive Hermite Gaussian basis are evaluated by GPU (Direct J Engine).

$$J_{ab} = \sum_{cd} (ab|cd) D_{cd} = \sum_{p} E_{ab}^{p} \sum_{q} (p|q) D_q$$

Schwarz upper bounds are used to reduce the number of ERIs to be calculated, but we don't use integral symmetry (e.g. (p|q)=(q|p)).
The McMurchie-Davidson algorism is used to evaluate Hermite ERIs.
The communication time between CPU and GPU to send / receive p, q, D_q, J_p was only about 10 % of GPU computation time.

The upper bound of each ERI is first evaluated and ERIs which are small in magnitude are calculated on GPU with single-precision. The rests (10% of total count of ERIs) are calculated on GPU with double precision.

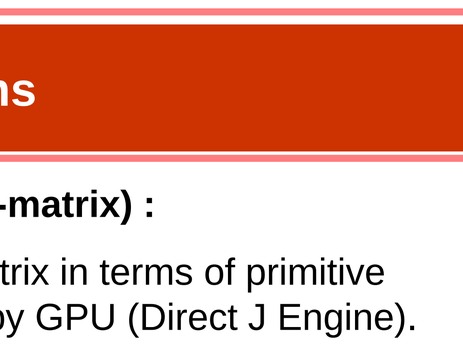
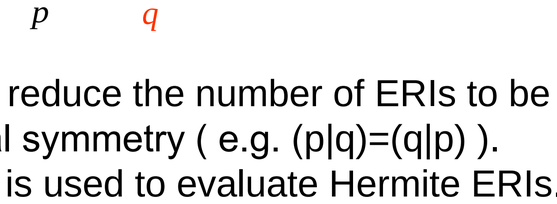


Fig 1. Valinomycin
(x 10^8, converged D)



Environmental Electrostatic Potential (ESP):

J-matrix algorithm is reused.
This is the mainly time-consuming step of FMO.

PRISM host code was re-written to use AVX as much as possible.

Four shell quartets are packed in an AVX vector and the 1- and 2-electron transformations in PRISM are fully vectorized. Some parts (evaluation of Boys function and summation of ERIs to Fock) are not vectorized yet.

Results & Discussion

Benchmark Environment :

(1) Intel Core i5 2500 @ 3.30GHz(4core), 6GB, CUDA 3.2,

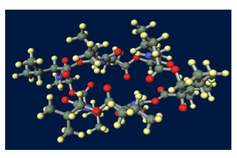
(2) Intel Core i7-3930K @3.2GHz (6 core) , GTX580 x 2, 32GB, CUDA 4.0

(3) Intel Core i7-3930K @3.2GHz (6 core), GTX580 x 4, 32GB, CUDA 4.0

using Intel Compiler XE 12.0 + MKL 10.3

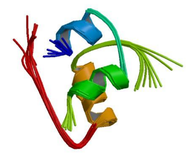
Table 1 : Ab-initio calculation time of Valinomycine by GAMESS 2010 R3 (RHF/3-21G, mainly Direct SCF) (1)

Software	Time(sec)	Energy[a.u.]
Original	476.24	-3750.9205018139
Original + XA-AVX-QM	149.69	-3750.9205017267



We achieved 3.18x acceleration of Valinomycine calculation using AVX PRISM algorithm.

Table 2-1 : FMO2 monomer energies of each residue of Insulin(2HIU) by GAMESS 2010 R3 (FMO-RHF/6-31G, a.u.) (2) + (3)



Residue	E(CPU)	ΔE(AVX+CUDA)
1(GLY)	-442.916776617	0.000000005
2(VAL)	-308.573606501	0.000000009
3(GLU)	-457.046042916	0.000000011
4(GLN)	-437.255799945	0.000000002
5(CYS)	-1254.889732125	0.000000004
6(CYS)	-1254.930631919	0.000000003
7(THR)	-344.327645726	0.000000007
8(SER)	-305.393476307	0.000000004
9(ILE)	-347.549172133	0.000000010

1. The value of dipole moment is exactly the same.
2. Total Monomer Energy
   E(Original) : -21635.4488652520
   E(AVX+CUDA) : -21635.4488649211
   ΔE : 3.31×10^{-7} a.u.
3. Calculation Time (sec) :
   Original : 3279.944
   GPGPU : 790.527

Table 2-2 : Calculation time of FMO2 environmental electrostatic potential of Insulin(2HIU) (sec) (2) + (3)

	Time(Original)	Time(AVX+CUDA)
FMO-ESP(CUDA : XA-CUDA-QM)	2571.490	170.897
FMO-ERI(AVX : XA-AVX-QM)	708.454	619.630

We achieved 4.15x acceleration of 44 residue protein calculation for total time by AVX+CUDA than the original GAMESS, and 15.0x acceleration for ESP alone. The calculation of ESP consumes predominant time (over 75%) so that FMO2 accelerates so much. The direct SCF method is unfavorable for GPGPU because fragments are so small that we can store all ERIs in a host memory. AVX was found to accelerate ERI slightly in this case. The total energy of FMO and the dipole moments were essentially the same, which means the accuracy of computation is enough.

Summary

Ab initio calculation by AVX becomes 3.18 time faster than GAMESS original one by multi-core CPUs. FMO calculation of environmental electrostatic potential by mainly CUDA becomes 15.0 times faster than multi-core CPUs calculation. AVX is found to be useful to accelerate the host-side ERI evaluation for FMO3 or higher.

Future Theme

1. Acceleration of ERI by AVX+CUDA including f or higher basis for metal protein
2. Parallel evaluation of target using RI-MP2 algorithm by GPGPU
3. Combination with molecular dynamics method
4. AVX vectorization of Boys function and summation of ERIs to Fock
5. Implementation on MIC architecture such as Knights Corner