Supporting Information

Accelerating the Generalized Born with Molecular Volume and Solvent Accessible Surface Area Implicit Solvent Model Using Graphics Processing Units

Xiping Gong [1], *Mara Chiricotto* [1], *Xiaorong Liu* [1], *Erik Nordquist* [1], *Michael Feig* [2], *Charles L. Brooks III* [3] *and Jianhan Chen* [1,4*]

[1] Department of Chemistry, University of Massachusetts, Amherst, Amherst, MA 01003, United States
[2] Department of Biochemistry and Molecular Biology, Michigan State University, East Lansing, Michigan 48824, United States
[3] Department of Chemistry, University of Michigan, Ann Arbor, MI, 48109. United States
[4] Department of Biochemistry & Molecular Biology, University of Massachusetts, Amherst, Amherst, MA 01003, United States

*Corresponding Author: Phone: (413) 545-3386; Email: jianhanc@umass.edu

Electrostatic solvation energies

The electrostatic solvation energies in a low concentration of salt are described as follows,

$$\Delta G^{\text{elec}} = -\frac{1}{2}\sum_{i,j}\tau_{ij}\frac{q_i q_j}{f_{ij}^{\text{GB}}},$$
$$\tau_{ij} = \left(1/\varepsilon_{\text{solute}} - \exp\left(-\kappa f_{ij}^{\text{GB}}\right)/\varepsilon_{\text{solvent}}\right), \qquad \text{(Eq. S1)}$$
$$f_{ij}^{\text{GB}} = \sqrt{\mathbf{R}_{ij}^2 + R_i^{\text{GB}} R_j^{\text{GB}} \exp\left(-\mathbf{R}_{ij}^2/K_s R_i^{\text{GB}} R_j^{\text{GB}}\right)}.$$

where q_i and R_i^{GB} the partial charge and Born radius of i^{th} atom, $\mathbf{R}_{ij}$ is a distance vector between two atoms, K_s is usually set to 8 for GPU-GBMV2/SA electrostatic calculations, $\varepsilon_{\text{solute}}$ and $\varepsilon_{\text{solvent}}$ are the dielectric constant of solute and solvent, respectively, and κ is a Debye-Huckel screening parameter.

In GBMV2/SA model, the Born radii are related to the molecular volume by considering the numerical Coulomb and high-order correction terms.

$$R_i^{GB} = \frac{P_1}{a_0 G_i^0 + a_1 G_i^1} + P_2,$$
$$G_i^0 = \frac{1}{R_i^{\text{eff}}} - \sum_n w_n^0 V\left(\mathbf{r}_n + \mathbf{R}_i\right),\ G_i^1 = \left(\frac{1}{4\left(R_i^{\text{eff}}\right)^4} - \sum_n w_n^1 V\left(\mathbf{r}_n + \mathbf{R}_i\right);\right)^{1/4}, \qquad \text{(Eq. S2)}$$

where the parameters of Born radii are P_1 = 0.9085, P_2 = -0.102 Å, $a_0 = 1 - 1/\sqrt{2}$, and a_1 = 1, $\mathbf{r}_n$ are the coordinates of grid points, $\mathbf{R}_i$ are the atomic coordinates, w_n^0 are the grid weights of the CFA term, and w_n^1 are the grid weights of the correction term, and R_i^{eff} are the effective atomic radii used for the quadrature integrals.

The molecular volume has a complicated expression.

$$V\left(\mathbf{r}_n + \mathbf{R}_i\right) = \frac{1}{1 + \exp\left[\beta\left(S\left(\mathbf{r}_n + \mathbf{R}_i\right) - \lambda\right)\right]},$$
$$S\left(\mathbf{r}_n + \mathbf{R}_i\right) = S_0 X_1 \frac{X_2}{\left(\mathbf{X}_3\right)^2} + 2X_4,\ \mathbf{t}_{nij} = \mathbf{r}_n + \mathbf{R}_i - \mathbf{R}_j, \qquad \text{(Eq. S3)}$$

and four intermediate volumes are written as follows,

$$X_1\left(\mathbf{r}_n+\mathbf{R}_i\right)=\sum_j F_{\mathrm{MV2}}\left(\left|\mathbf{t}_{nij}\right|\right),$$

$$X_2\left(\mathbf{r}_n+\mathbf{R}_i\right)=\sum_j \left|\mathbf{t}_{nij}\right|^2 F_{\mathrm{MV2}}^2\left(\left|\mathbf{t}_{nij}\right|\right),$$

$$\mathbf{X}_3\left(\mathbf{r}_n+\mathbf{R}_i\right)=\sum_j \mathbf{t}_{nij} F_{\mathrm{MV2}}\left(\left|\mathbf{t}_{nij}\right|\right),$$

$$X_4\left(\mathbf{r}_n+\mathbf{R}_i\right)=\sum_j F_{\mathrm{vdW}}\left(u_{nij}\right),$$

$$F_{\mathrm{vdw}}\left(u_{nij}\right)=\begin{bmatrix} 1 & u_{nij}\le 0 \\ 1+u_{nij}^3\left[u_{nij}\left(15-6u_{nij}\right)-10\right] & 1>u_{nij}>0 \\ 0 & u_{nij}\ge 1 \end{bmatrix}, u_{nij}=\frac{\left|\mathbf{t}_{nij}\right|^2-\left(R_j^{\mathrm{vdW}}+t_-^{\mathrm{vdW}}\right)^2}{\left(R_j^{\mathrm{vdW}}+t_+^{\mathrm{vdW}}\right)^2-\left(R_j^{\mathrm{vdW}}+t_-^{\mathrm{vdW}}\right)^2}$$

$$F_{\mathrm{MV2}}\left(\left|\mathbf{t}_{nij}\right|\right)=\begin{bmatrix} F_{\mathrm{MV2}}^*\left(\left|\mathbf{t}_{nij}\right|\right)\left(1-F_{\mathrm{vdW}}\left(\left|\mathbf{t}_{nij}\right|\right)\right) & R_j^{\mathrm{vdW}}+t_-^{\mathrm{vdW}}<\left|\mathbf{t}_{nij}\right|<=R_j^{\mathrm{vdW}}+t_+^{\mathrm{vdW}} \\ F_{\mathrm{MV2}}^*\left(\left|\mathbf{t}_{nij}\right|\right) & R_j^{\mathrm{vdW}}+t_+^{\mathrm{vdW}}<\left|\mathbf{t}_{nij}\right|<=R_j^{\mathrm{vdW}}+t_-^{\mathrm{MV2}} \\ F_{\mathrm{MV2}}^*\left(\left|\mathbf{t}_{nij}\right|\right)F_{\mathrm{vdw}}\left(\left|\mathbf{t}_{nij}\right|\right) & R_j^{\mathrm{vdW}}+t_-^{\mathrm{MV2}}<\left|\mathbf{t}_{nij}\right|<=R_j^{\mathrm{vdW}}+t_+^{\mathrm{MV2}} \end{bmatrix}, \qquad \text{(Eq. S4)}$$

$$F_{\mathrm{MV2}}^*\left(\mathbf{t}_{nij}\right)=\frac{\left(C_j\right)^2}{\left(C_j+\left|\mathbf{t}_{nij}\right|^2-\left(R_j^{\mathrm{vdW}}\right)^2\right)^2}, C_j=c_1+c_2R_j^{\mathrm{vdW}}, \text{ or } F_{\mathrm{MV2}}^*\left(\mathbf{t}_{nij}\right)=\exp\left[\alpha\left(\left|\mathbf{t}_{nij}\right|-R_j^{\mathrm{vdW}}\right)\right].$$

where,

the parameters of molecular volume are β = -12, λ= 0.5, S_0 = 0.65, and α = -1.98 1/Å,

the parameters of approximated function are c_1= 0.45 Å^2, and c_2 =1.25 Å,

the parameters of atomic volume function are t_-^{vdW} = -0.125 Å, t_+^{vdW} = 0.25 Å, t_-^{MV2} = 1.90 Å, and t_+^{MV2} = 2.10 Å, respectively, R_i^{vdW} are the atomic vdW or input radii,

Electrostatic solvation forces

The electrostatic solvation forces in terms of atomic positions are expressed as follows,

$$F_a^{\text{elec}} = -\frac{\partial \Delta G^{\text{elec}}}{\partial \mathbf{R}_a} = -\left(\sum_{ij} \frac{\partial \Delta G^{\text{elec}}}{\partial \mathbf{R}_{ij}} \frac{\partial \mathbf{R}_{ij}}{\partial \mathbf{R}_a} + \sum_i \frac{\partial \Delta G^{\text{elec}}}{\partial R_i^{\text{GB}}} \frac{\partial R_i^{\text{GB}}}{\partial \mathbf{R}_a} \right),$$

$$F_a^{\text{elec,1}} = -\sum_{ij} \frac{\partial \Delta G^{\text{elec}}}{\partial \mathbf{R}_{ij}} \frac{\partial \mathbf{R}_{ij}}{\partial \mathbf{R}_a}$$

$$= \sum_i \left(\tau_{ia} - \frac{\kappa \exp\left(-\kappa f_{ia}^{\text{GB}}\right) f_{ia}^{\text{GB}}}{\varepsilon_{\text{solvent}}} \right) \frac{q_i q_a \left[1 - \exp\left(-\mathbf{R}_{ia}^2 / K_s R_i^{\text{GB}} R_a^{\text{GB}} \right) / K_s \right]}{\left(f_{ia}^{\text{GB}} \right)^3} \left(\mathbf{R}_i - \mathbf{R}_a \right), \quad \text{(Eq. S5)}$$

$$F_a^{\text{elec,2}} = -\sum_i \frac{\partial \Delta G^{\text{elec}}}{\partial R_i^{\text{GB}}} \frac{\partial R_i^{\text{GB}}}{\partial \mathbf{R}_a} = -\left(F_a^{\text{elec,2a}} + F_a^{\text{elec,2b}} \right),$$

$$F_a^{\text{elec,2a}} = \frac{\partial \Delta G^{\text{elec}}}{\partial R_a^{\text{GB}}} \sum_j \sum_n \left(\frac{\partial R^{\text{GB}}}{\partial \mathbf{R}} \right)^{naj}, \quad F_a^{\text{elec,2b}} = \sum_i \sum_n \left(-\frac{\partial \Delta G^{\text{elec}}}{\partial R_i^{\text{GB}}} \right) \left(\frac{\partial R^{\text{GB}}}{\partial \mathbf{R}} \right)^{nia},$$

where,

$$\frac{\partial \Delta G^{\text{elec}}}{\partial R_i^{\text{GB}}} = \frac{1}{2} \sum_j \left(\tau_{ij} - \frac{\kappa \exp\left(-\kappa f_{ij}^{\text{GB}}\right) f_{ij}^{\text{GB}}}{\varepsilon_{\text{solvent}}} \right) \frac{q_i q_j \exp\left(-\mathbf{R}_{ij}^2 / K_s R_i^{\text{GB}} R_j^{\text{GB}} \right)}{\left(f_{ij}^{\text{GB}} \right)^3} \left(R_j^{\text{GB}} + \frac{\mathbf{R}_{ij}^2}{K_s R_i^{\text{GB}}} \right),$$

$$\left(\frac{\partial R^{\text{GB}}}{\partial \mathbf{R}} \right)^{nij} = \frac{P_1 \beta \left(a_0 w_n^0 + \dfrac{a_1 w_n^1}{4 \left(G_i^1 \right)^3} \right)}{\left(a_0 G_i^0 + a_1 G_i^1 \right)^2} \frac{\exp\left[\beta \left(S_{ni} - \lambda \right) \right]}{\left(1 + \exp\left[\beta \left(S_{ni} - \lambda \right) \right] \right)^2}$$

$$\left(\frac{\partial X_{1,ni}}{\partial \mathbf{R}_j} \frac{S_0 X_{2,ni}}{\left(\mathbf{X}_{3,ni} \right)^2} + \frac{\partial X_{2,ni}}{\partial \mathbf{R}_j} \frac{S_0 X_{1,ni}}{\left(\mathbf{X}_{3,ni} \right)^2} - \frac{\partial \mathbf{X}_{3,ni}}{\partial \mathbf{R}_j} \frac{\mathbf{X}_{3,ni} 2 S_0 X_{1,ni} X_{2,ni}}{\left(\mathbf{X}_{3,ni} \right)^4} + 2 \frac{\partial X_{4,ni}}{\partial \mathbf{R}_j} \right).$$

$$\frac{\partial X_{1,ni}}{\partial \mathbf{R}_j} = \frac{\partial F_{\text{MV2}}\left(\left| \mathbf{t}_{nij} \right| \right)}{\partial \left| \mathbf{t}_{nij} \right| \left| \mathbf{t}_{nij} \right|} \mathbf{t}_{nij},$$

$$\frac{\partial X_{2,ni}}{\partial \mathbf{R}_j} = 2 F_{\text{MV2}}\left(\left| \mathbf{t}_{nij} \right| \right) \left(F_{\text{MV2}}\left(\left| \mathbf{t}_{nij} \right| \right) + \frac{\partial F_{\text{MV2}}\left(\left| \mathbf{t}_{nij} \right| \right)}{\partial \left| \mathbf{t}_{nij} \right| \left| \mathbf{t}_{nij} \right|} \left| \mathbf{t}_{nij} \right|^2 \right) \mathbf{t}_{nij},$$

$$\frac{\partial \mathbf{X}_{3,ni}}{\partial \mathbf{R}_j} = F_{\text{MV2}}\left(\left| \mathbf{t}_{nij} \right| \right) + \frac{\partial F_{\text{MV2}}\left(\left| \mathbf{t}_{nij} \right| \right)}{\partial \left| \mathbf{t}_{nij} \right| \left| \mathbf{t}_{nij} \right|} \left(\mathbf{t}_{nij} \right)^T \mathbf{t}_{nij},$$

$$\frac{\partial X_{4,ni}}{\partial \mathbf{R}_j} = \frac{\partial F_{\text{vdW}}\left(\left| \mathbf{t}_{nij} \right| \right)}{\partial \left| \mathbf{t}_{nij} \right| \left| \mathbf{t}_{nij} \right|} \mathbf{t}_{nij},$$

$$\frac{\partial F_{\text{vdW}}\left(\left| \mathbf{t}_{nij} \right| \right)}{\partial \left| \mathbf{t}_{nij} \right| \left| \mathbf{t}_{nij} \right|} = \frac{60 u_{nij}^2 \left(u_{nij} \left(2 - u_{nij} \right) - 1 \right)}{\left(R_j^{\text{vdW}} + t_+^{\text{vdw}} \right)^2 - \left(R_j^{\text{vdW}} + t_-^{\text{vdw}} \right)^2},$$

$$\frac{\partial F_{\text{MV2}}\left(\left| \mathbf{t}_{nij} \right| \right)}{\partial \left| \mathbf{t}_{nij} \right| \left| \mathbf{t}_{nij} \right|} = \begin{bmatrix} \dfrac{\partial F_{\text{MV2}}^*\left(\left| \mathbf{t}_{nij} \right| \right)}{\partial \left| \mathbf{t}_{nij} \right| \left| \mathbf{t}_{nij} \right|} \left(1 - F_{\text{vdw}}\left(\left| \mathbf{t}_{nij} \right| \right) \right) - F_{\text{MV2}}^*\left(\left| \mathbf{t}_{nij} \right| \right) \dfrac{\partial F_{\text{vdW}}\left(\left| \mathbf{t}_{nij} \right| \right)}{\partial \left| \mathbf{t}_{nij} \right| \left| \mathbf{t}_{nij} \right|} \\ \dfrac{\partial F_{\text{MV2}}^*\left(\left| \mathbf{t}_{nij} \right| \right)}{\partial \left| \mathbf{t}_{nij} \right| \left| \mathbf{t}_{nij} \right|} \\ \dfrac{\partial F_{\text{MV2}}^*\left(\left| \mathbf{t}_{nij} \right| \right)}{\partial \left| \mathbf{t}_{nij} \right| \left| \mathbf{t}_{nij} \right|} F_{\text{vdw}}\left(\left| \mathbf{t}_{nij} \right| \right) + F_{\text{MV2}}^*\left(\left| \mathbf{t}_{nij} \right| \right) \dfrac{\partial F_{\text{vdW}}\left(\left| \mathbf{t}_{nij} \right| \right)}{\partial \left| \mathbf{t}_{nij} \right| \left| \mathbf{t}_{nij} \right|} \end{bmatrix}, \quad \text{(Eq. S6)}$$

$$\frac{\partial F_{\text{MV2}}^*\left(\mathbf{t}_{nij} \right)}{\partial \left| \mathbf{t}_{nij} \right| \left| \mathbf{t}_{nij} \right|} = \frac{-4 \left(C_j \right)^2}{\left(C_j + \left| \mathbf{t}_{nij} \right|^2 - \left(R_j^{\text{vdW}} \right)^2 \right)^3}, \text{ or } \frac{\partial F_{\text{MV2}}^*\left(\mathbf{t}_{nij} \right)}{\partial \left| \mathbf{t}_{nij} \right| \left| \mathbf{t}_{nij} \right|} = \frac{\alpha \exp\left[\alpha \left(\left| \mathbf{t}_{nij} \right| - R_j^{\text{vdW}} \right) \right]}{\left| \mathbf{t}_{nij} \right|}.$$

Nonpolar solvation energies and forces

Based on the expression of nonpolar energy, the forces, the derivatives in terms of the atomic position, are expressed as follows,

$$\Delta G_{\mathrm{np}} = \sum_i \gamma_i A_i$$
$$= \sum_i \gamma_i 4\pi \left(R_i^{\mathrm{vdW}} + R_{\mathrm{w}}\right)^2 \sum_m w_m f\left(\overline{V}_i\left(\hat{\mathbf{r}}_m\left(R_i^{\mathrm{vdW}} + R_{\mathrm{w}}\right) + \mathbf{R}_i\right)\right), \quad \text{(Eq. S7)}$$

$$F_a^{\mathrm{np}} = -\frac{\partial \Delta G_{\mathrm{np}}}{\partial \mathbf{R}_a}, \text{ and}$$

$$\frac{\partial \Delta G_{\mathrm{np}}}{\partial \mathbf{R}_a} = \sum_i \gamma_i 4\pi \left(R_i^{\mathrm{vdW}} + R_{\mathrm{w}}\right)^2 \sum_m w_m f'\left[\overline{V}_i\left(\hat{\mathbf{r}}_m\left(R_i^{\mathrm{vdW}} + R_{\mathrm{w}}\right) + \mathbf{R}_i\right)\right]$$
$$\sum_{j \neq i} 4 f'\left(u_{mij}\right) \frac{\left(\hat{\mathbf{r}}_m\left(R_i^{\mathrm{vdW}} + R_{\mathrm{w}}\right) + \mathbf{R}_i - \mathbf{R}_j\right)}{\left(R_j^{\mathrm{vdW}} + t_+^{\mathrm{SA}}\right)^2 - \left(R_j^{\mathrm{vdW}} + t_-^{\mathrm{SA}}\right)^2}\left(\delta_{ia} - \delta_{ja}\right). \quad \text{(Eq. S8)}$$

where the $f'\left(u\right)$ is the derivative of exposed function, and $\hat{\mathbf{r}}_m$ is the unit vector of grid points.

In order to implement the nonpolar energy and forces in one kernel, whose calculation was divided into two parts and then effectively avoid the conflicts of blocks.

$$\frac{\partial \Delta G_{\mathrm{np}}}{\partial \mathbf{R}_a} = \sum_m \gamma_a 4\pi \left(R_a^{\mathrm{vdW}} + R_{\mathrm{w}}\right)^2 w_m f'\left[\overline{V}_a\left(\hat{\mathbf{r}}_m\left(R_a^{\mathrm{vdW}} + R_{\mathrm{w}}\right) + \mathbf{R}_a\right)\right]$$
$$\left[\sum_{j \neq a} 4 f'\left(u_{maj}\right) \frac{\left(\hat{\mathbf{r}}_m\left(R_a^{\mathrm{vdW}} + R_{\mathrm{w}}\right) + \mathbf{R}_a - \mathbf{R}_j\right)}{\left(R_j^{\mathrm{vdW}} + t_+^{\mathrm{SA}}\right)^2 - \left(R_j^{\mathrm{vdW}} + t_-^{\mathrm{SA}}\right)^2}\right] + \quad \text{(Eq. S9)}$$
$$\sum_m \frac{4 w_m}{\left(R_a^{\mathrm{vdW}} + t_+^{\mathrm{SA}}\right)^2 - \left(R_a^{\mathrm{vdW}} + t_-^{\mathrm{SA}}\right)^2}\left[\sum_{j \neq a} f'\left[\overline{V}_j\left(\hat{\mathbf{r}}_m\left(R_j^{\mathrm{vdW}} + R_{\mathrm{w}}\right) + \mathbf{R}_j\right)\right]\right.$$
$$\left.\gamma_j 4\pi \left(R_j^{\mathrm{vdW}} + R_{\mathrm{w}}\right)^2 f'\left(u_{mja}\right)\left(\mathbf{R}_a - \hat{\mathbf{r}}_m\left(R_j^{\mathrm{vdW}} + R_{\mathrm{w}}\right) - \mathbf{R}_j\right)\right].$$

CUDA algorithm for computing the electrostatic solvation energies

Two important steps are used to calculate the electrostatic solvation energies. First step is to calculate the Born radius of each atom. Besides looping over all atoms, it is necessary to loop each numerical integration grid and then all neighbor atoms at each grid point (as given by the lookup table). The major cost is to compute the molecular volume at each grid point (Eq. S3), which has four intermediate volumes (Eq. S4) that can be attributed to the neighbor atoms. The pseudocode is given below.

Each block loops the atoms (i)
    Assign the shared memory (size = # the numerical grids)
  Loop the numerical grids (n) using 256 threads (optimal)
    Initialize the V, S, X_1, X_2, X_3, and X_4
    Using the lookup table array to locate the neighbor atoms ($\mathbf{r}_n + \mathbf{R}_i \Rightarrow \mathbf{R}_j$)
    Loop the neighbor atoms
      Calculate the F_{vdW}, F_{MV2}, and X_1, X_2, X_3, and X_4 (Eq. S4)
      Calculate the S and V (Eq. S3), and save them into shared memory
      Save S, X_1, X_2, X_3, and X_4 into global memory for the calculations of forces
    Loop the numerical grids (n) using 1 thread
      Do the sum reduction (Eq. S2) by extracting data from the shared memory
    Save the sum into the Born radius of each atom
END

After obtaining the Born radii, the existing kernel was used for computing the electrostatic solvation energies (Eq. S1).

CUDA algorithm for computing the electrostatic solvation forces

Computing the electrostatic solvation forces are much more complicated than that of energies. The forces in terms of the coordinates can be calculated using a similar algorithm as implemented in the GPU-GBSW plugin. For the forces in terms of the Born radii, the computation of atomic forces is divided into two parts, in order to avoid the conflict of blocks. The algorithm is summarized in the following pseudo code.

Each block loops the atoms (i)
    Assign the shared memory (size = # the numerical grids)
    # First part of Eq. S5: $F^{\text{elec, 2a}}$
  Loop the numerical grids (n) using 256 threads (optimal)
    Using the lookup table array to locate the neighbor atoms ($\mathbf{r}_n + \mathbf{R}_i \Rightarrow \mathbf{R}_j$)
     Access the global arrays (S, X_1, X_2, X_3, and X_4)
     Loop the neighbor atoms
        Calculate the derivatives of F_{vdW}, F_{MV2}, and X_1, X_2, X_3, and X_4 (Eq. S6)
            Calculate derivatives of Born radii (Eq. S6)
        Do the sum reduction and save into the shared memory
    Loop the numerical grids (n) using 1 thread
        Do the sum reduction (Eq. S2) by extracting data from the shared memory
    Save the sum into the atomic forces array
    # Second part of Eq. S5: $F^{\text{elec, 2b}}$
  Loop the numerical grids (n) using 256 threads (optimal)
    Using the lookup table array to locate the neighbor atoms ($\mathbf{r}_n + \mathbf{R}_i \Rightarrow \mathbf{R}_j$)
     Loop the neighbor atoms
       Access the global arrays (S, X_1, X_2, X_3, and X_4)
        Calculate the derivatives of F_{vdW}, F_{MV2}, and X_1, X_2, X_3, and X_4 (Eq. S6)
            Calculate derivatives of Born radii (Eq. S6) using S, X_1, X_2, X_3, and X_4
        Do the sum reduction and save into the shared memory
    Loop the numerical grids (n) using 1 thread
        Do the sum reduction (Eq. S2) by extracting data from the shared memory
    Save the sum into the atomic forces array
END

The computational bottleneck is to calculate the second part, because frequent access of the global arrays (S, X_1, X_2, X_3, and X_4) is expensive. Additionally, these global arrays plus the lookup table array takes up most of the global memory, which should be optimized by minimizing the effective size.

Structure analysis of key GPU-GBMV2/SA kernels

In the most time-consuming reduceGBMVForce kernel, each thread uses 54 registers (**Figure S1**), and each block uses 54 x 256 = 13,824 registers. Since each streaming multiprocessor (SM) provides 65,536 register on Titan X (Pascal), only 4 blocks (equivalently, 32 warps or 1024 threads) could run simultaneously on each SM. The analysis of the computeNonbonded kernel of OpenMM is also provided for reference. Even though the computeNonbonded kernel has a higher theoretical occupancy of 62.5%, the actual achieved occupancies are similar between these two kernels.

Variable	Achieved	Theoretical	Device Limit
Occupancy Per SM			
Active Blocks		4	32
Active Warps	29.14	32	64
Active Threads		1024	2048
Occupancy	45.5%	50%	100%
Warps			
Threads/Block		256	1024
Warps/Block		8	32
Block Limit		8	32
Registers			
Registers/Thread		54	65536
Registers/Block		14336	65536
Block Limit		4	32
Shared Memory			
Shared Memory/Block		3072	98304
Block Limit		32	32

Variable	Achieved	Theoretical	Device Limit
Occupancy Per SM			
Active Blocks		5	32
Active Warps	31.69	40	64
Active Threads		1280	2048
Occupancy	49.5%	62.5%	100%
Warps			
Threads/Block		256	1024
Warps/Block		8	32
Block Limit		8	32
Registers			
Registers/Thread		42	65536
Registers/Block		12288	65536
Block Limit		5	32
Shared Memory			
Shared Memory/Block		2048	98304
Block Limit		48	32

Figure S1. GPU utilization using the nvvp and nvprof tools for the reduceGBMVForce kernel in GBMV2/SA (left) and the computeNonbonded kernel of OpenMM (right). The profile results were obtained using protein 3GB1.

Multi-Core Performance of CPU GBMV2/SA

Table S1. Benchmarks of GBMV2/SA for GPU vs. parallel CPU calculations with 1, 2, 4, 8, 12 and 16 cores. The time step was set to 2-fs. The GPU and CPU calculations were done on one NVIDIA TITAN X (Pascal) and the Intel Xeon E5-2620 v4 2.10GHz CPU, respectively.

PDBID (#Atoms)	3GB1 (855)	P53-TAD (926)	1BVC (2459)	4AT5 (11766)
CPU-GBMV2/SA (ns/day)	0.7969 (1x)	0.8392 (1x)	0.2534 (1x)	0.0488 (1x)
Fast CPU-GBMV2/SA (1-core)	1.2614 (1.6x)	1.3168 (1.6x)	0.3826 (1.5x)	0.0728 (1.5x)
2-core / 1-core	2.0x	2.0x	2.0x	2.0x
4-core / 1-core	4.0x	4.0x	3.9x	3.5x
8-core / 1-core	5.1x	5.1x	5.2x	4.6x
12-core / 1-core	5.6x	5.6x	5.7x	4.8x
16-core / 1-core	6.4x	6.4x	6.7x	5.5x
GPU-GBMV2/SA (1-GPU)	46.9974 (59.0x)	48.9630 (58.3x)	15.9292 (62.9x)	3.5294 (72.3x)