JCTC Journal of Chemical Theory and Computation

Article

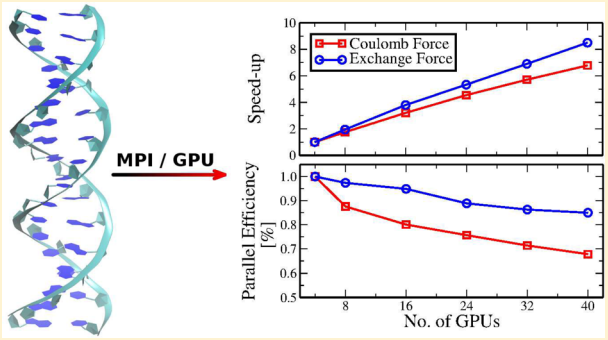
pubs.acs.org/JCTC

Preselective Screening for Linear-Scaling Exact Exchange-Gradient Calculations for Graphics Processing Units and General Strong-Scaling Massively Parallel Calculations

Jörg Kussmann and Christian Ochsenfeld*

Department of Chemistry and Center for Integrated Protein Science (CIPSM), University of Munich (LMU), D-81377 München, Germany

ABSTRACT: We present an extension of our recently presented PreLinK scheme (*J. Chem. Phys.* **2013**, *138*, 134114) for the exact exchange contribution to nuclear forces. The significant contributions to the exchange gradient are determined by preselection based on accurate shell-pair contributions to the SCF exchange energy *prior* to the calculation. Therefore, our method is highly suitable for massively parallel electronic structure calculations because of an efficient load balancing of the significant contributions only and an unhampered control flow. The efficiency of our method is shown for several illustrative calculations on single GPU servers, as well as for hybrid MPI/CUDA parallel calculations with the largest system comprising 3369 atoms and 26952 basis functions.



I. INTRODUCTION

In recent years, the use of graphics processing units (GPUs) for quantum chemical calculations has been explored by several groups.[1−11] It has been shown that significant speedups can be gained for Hartree−Fock (HF) and density functional theory (DFT) calculations by careful consideration of the underlying GPU architecture. In particular, the data arrangement as proposed by Ufimtsev and Martínez[3,5] has proven to be highly suitable for processing on GPUs. Whereas their original work was limited to minimal basis sets,[7] GPU-based calculations can nowadays also be performed with medium to large basis sets.[10]

However, the efficient adaption of the rate-determining integral-evaluation steps to the GPU architecture also means that linear-scaling methods, which were developed over the past several decades for central processing units (CPUs; see, e.g., ref 12 for a recent review), cannot be directly transferred to GPUs. The reasons for this problem are the more involved control flow, bookkeeping, and random memory access of these linear-scaling algorithms such as the continuous fast multipole method (CFMM)[13] for the Coulomb matrix or the linear exchange K (LinK) method[14] for the exchange matrix.

The first linear-scaling method designed to perform efficiently on GPUs is the PreLinK scheme.[10] It allows an $O(N)$ formation of the exchange matrix by an accurate preselection of the significant elements $K_{\mu\nu}$ *prior* to the integral evaluation, thus allowing an unhampered control flow on GPUs. Because the formation of **K** is far more demanding than Coulomb evaluation, the impact of PreLinK screening on the overall performance is significant.[10] Furthermore, the PreLinK scheme also allows the scattering of the exchange matrix into a sparse storage format[15−17] by directly using the sparse pattern of the screening matrix for **K**.

In this work, we present a linear-scaling method for the efficient evaluation of the exchange contribution to the nuclear gradient of the energy[18] based on the PreLinK scheme. The scaling and efficiency of our approach are shown for illustrative calculations on a single GPU server where the largest system contains 3369 atoms and 26952 basis functions. Furthermore, we analyze a more general advantage of our PreLinK scheme that can also benefit massively parallel MPI (Message Passing Interface) calculations on pure CPU architectures: The predetermination of significant exchange matrix elements enables a straightforward and efficient load balancing that is demonstrated for hybrid MPI/CUDA (Compute Unified Device Architecture) calculations on up to 10 GPU servers and a total of 40 GPU devices.

II. THEORY

For the evaluation of the exchange gradient, we use the same shell-pair data arrangement as for the exchange matrix evaluation;[5,10] i.e., for a given l-quantum-number combination, we first sort according to the first index μ, and for a given index, we sort the batch $[\mu...|$ with respect to decreasing Schwarz integral[19] values. A single, primitive exchange integral $K_{\mu\nu}$ is then evaluated from the two data sets $[\mu...|$ and $[\nu...|$ as described in refs 5 and 10, exploiting only one integral symmetry, namely, $K_{\mu\nu} \Leftrightarrow K_{\nu\mu}$.

For the evaluation of the exchange gradient, we choose to retain the algorithmic structure. Thus, the nuclear derivative

Received: October 4, 2014
Published: February 27, 2015

DOI: 10.1021/ct501189u
J. Chem. Theory Comput. 2015, 11, 918−922

of the exchange energy E_K with respect to the coordinates of nucleus A is given as

$$\nabla_A E_K = -\frac{1}{2}\sum_{\mu\nu\lambda\sigma} P_{\mu\nu}[([\nabla_A\mu]\lambda|\nu\sigma) + (\mu\lambda|[\nabla_A\nu]\sigma)]P_{\lambda\sigma} \quad (1)$$

For the integral evaluations, Rys quadrature is employed, which has proven to be efficient for GPUs because of its low computational resource demands. Here, the explicit derivatives of μ and ν are calculated and directly contracted with the density matrix elements $P_{\mu\nu}$ and $P_{\lambda\sigma}$. The final summation over primitives and scattering into the derivative vector is done on the CPU.

As discussed in ref 10, a linear-scaling formation by a method such as LinK[14,20] or ONX (order-N exchange)[21,22] is not possible because of the on-the-fly screening within the inner loops of the algorithm, so that a preselective screening approach is necessary to ensure good performance on GPUs. For the evaluation of the exchange matrix, we proposed preselection with respect to

$$Q'_{\mu\nu} = \sum_{\lambda\sigma} \sqrt{(\mu\lambda|\mu\lambda)}\sqrt{(\nu\sigma|\nu\sigma)}\,|P_{\lambda\sigma}| \geq K_{\mu\nu} \quad (2)$$

where the matrix $\mathbf{Q}'$ can be simply determined by two sparse matrix multiplications, $\mathbf{Q}' = \mathbf{Q} \times |\mathbf{P}| \times \mathbf{Q}$. A straightforward extension to the gradient calculation would be to use

$$Q^{\nabla'}_{\mu\nu} = \sum_{\lambda\sigma} |P_{\mu\nu}|\sqrt{(\mu\lambda|\mu\lambda)}\sqrt{(\nu\sigma|\nu\sigma)}\,|P_{\lambda\sigma}| \quad (3)$$

to determine whether the contribution from $[\mu...|$ and $[\nu...|$ can be discarded. However, our study shows that the use of absolute values of the density matrix twice is inaccurate, being either ineffective for tight thresholds or inaccurate for less tight thresholds.

Thus, we suggest that the size of the contribution of $[\mu...|$ and $[\nu...|$ to the nuclear gradient be estimated by considering its contribution to the exchange energy, which can be written as

$$E_{exx} = -\frac{1}{2}\sum_{\mu\nu} P_{\mu\nu}K_{\mu\nu} \quad (4)$$

Therefore, for exchange-gradient preselection, we use the quantity

$$\mathbf{Q}^{\nabla'} = \mathbf{P}\circ\mathbf{K} \quad (5)$$

which can be evaluated as the Hadamard product of the density matrix and the exchange matrix, where the latter can be obtained from the preceding self-consistent-field (SCF) calculation. Thus, we discard the pair $[\mu...|$ and $[\nu...|$ if $Q^{\nabla'}_{\mu\nu}$ is less than ϑ^{∇}_{pre}.

Finally, we briefly contrast our screening scheme with the conventional approach, which determines whether the integral $(\mu\lambda|\nu\sigma)$ is significant using the expression

$$\vartheta_{int} \leq \sqrt{(\mu\lambda|\mu\lambda)}\sqrt{(\nu\sigma|\nu\sigma)} \times \max[\tilde{P}_{\mu\nu}\tilde{P}_{\lambda\sigma}, \tilde{P}_{\mu\sigma}\tilde{P}_{\lambda\nu}] \quad (6)$$

where $\tilde{P}_{\mu\nu}$ is the maximum absolute value of the density matrix for a given shell pair $\mu\nu$. Note that the screening step is within an inner loop for a given shell pair $\mu\lambda$; that is, the straightforward algorithm is $O(N^2)$ because of the screening step. An overall linear scaling is obtained by more involved algorithms such as LinK[14,20] or ONX[21,22] that involve a considerable amount of bookkeeping. Furthermore, it has to be stressed that the Schwarz integrals are used to estimate the significance of the interal $(\mu\lambda|\nu\sigma)$ to the exchange energy.

In contrast, our approach in eq 5 uses the correct—within the given numerical accuracy of the SCF solution—maximum contribution of the shell pair $\mu\nu$ to the exchange energy by introducing virtually no additional computational effort even for large systems with thousands of atoms. The consequences regarding, for example, the choice of the preselection threshold are discussed in section III.

II.A. Multi-GPU Exchange-Gradient Algorithm. In this section, we briefly describe the design of the GPU-based exchange-gradient routine, including multi-GPU parallelization and its extension to a hybrid MPI/CUDA scheme. The shell-pair data are arranged as described for the SCF exchange potential by Ufimtsev and Martínez,[5,10] that is

(1) Sort shell pairs according to l-quantum-number combinations (ss, ps, sp, pp, ds, sd, ...).
(2) Sort each l-quantum-number batch according to first index $[\mu...|$.
(3) Sort each $[\mu...|$ batch according to the Schwarz integral $[\mu\lambda|$.

Note that, in the case of an MPI/CUDA calculation, the shell-pair data are present on each node. The exchange-gradient algorithm is computed as depicted in the following scheme. The algorithm for OpenMP/CUDA and MPI/CUDA calculations are basically the same; MPI-specific steps are indicated accordingly.

- MPI: Broadcast density $\mathbf{P}$ and screening matrix $\mathbf{Q}^{\nabla'}$ to all nodes.
- Loop the following steps over bra and ket l-quantum-number combinations:
  - Preselect significant pairs $[\mu...|...\nu]$ for which $Q^{\nabla'}_{\mu\nu} \geq \vartheta_{pre}$.
  - Statically distribute batches of significant $[\mu...|... \nu]$ to GPUs (OpenMP/MPI).
  - For a single batch/GPU device, do the following:
    - Collect shell-pair data and submatrices of density with all occurring $[...\lambda|\sigma...]$ and $[\mu...|...\nu]$, respectively.
    - Copy data to GPU device.
    - Execute integral kernel by using Rys quadrature to evaluate the integral derivatives in eq 1 and directly contracting with $P_{\mu\nu}$ and $P_{\lambda\sigma}$
    - Copy back exchange-gradient contributions to nuclei related to μ and ν.
    - Scatter results into gradient vector.
- MPI: Reduce gradient vector.

The design of the GPU kernels is similar to that of the kernels for the SCF exchange potential evaluation. Note that we use a straightforward static parallelization because only significant shell pairs are distributed over the compute nodes. Thus, the different MPI nodes can be run independently.

III. ILLUSTRATIVE CALCULATIONS

All calculations were performed with the FermiONs++ program package,[10] in which our PreLinK gradient algorithm is implemented. The GPU servers all contained two Intel Xeon E5-2620@2.1 GHz CPUs with six cores each and four NVidia GTX Titan cards.

First, the impact of the PreLinK screening was analyzed by performing gradient calculations for several test systems[23] with

DOI: 10.1021/ct501189u
J. Chem. Theory Comput. 2015, 11, 918−922

Table 1. Effects of Conventional (STD) and Preselective (PreLinK) Screening on the Final SCF Gradient Vector for the Exchange-Gradient Calculation for Several Test Systems and Thresholds ϑ with HF/SVP[a,b]

		$\vartheta/\vartheta^{\nabla}_{pre}$				
		10^{-12}	10^{-11}	10^{-10}	10^{-9}	10^{-8}
amylose$_2$	STD	−0.004238	0.040176	0.308974	2.463321	−24.072653
	PreLinK	−1.655131	−1.852719	2.528770	60.356595	−150.506051
	wall time (s)	8.2	7.5	6.5	5.3	4.0
amylose$_4$	STD	−0.006749	0.040304	−0.358826	2.369706	−26.073794
	PreLinK	−0.447571	0.893235	2.461994	−67.711728	−158.076769
	wall time (s)	25.5	21.8	17.8	13.8	9.8
amylose$_8$	STD	−0.005842	−0.040340	−0.307553	2.344300	−25.771049
	PreLinK	0.405359	−1.837842	−47.871945	−74.586705	−139.610151
	wall time (s)	60.5	50.6	39.9	29.8	20.7
A−T$_1$	STD	−0.003654	0.028778	0.301787	2.508772	30.425719
	PreLinK	−0.022693	1.873725	−2.396069	154.256091	246.260224
A−T$_2$	STD	−0.005970	−0.040067	0.435669	−3.283286	28.899120
	PreLinK	0.456695	−0.705472	−24.109745	−23.053629	260.735671
$(H_2O)_{142}$	STD	0.024697	−0.052436	−0.361967	3.885328	−33.986588
	PreLinK	1.061366	−11.863304	13.255226	34.081778	166.089639
beta-carotene	STD	0.006395	−0.047957	0.416887	4.006672	26.282741
	PreLinK	0.113173	0.872039	2.838692	−24.625888	89.138311
angiotensin	STD	−0.004418	0.046072	−0.373452	−3.037332	35.681078
	PreLinK	0.055042	2.609798	−5.839585	−189.551402	−308.152918
graphite$_{54}$	STD	−0.002604	0.025349	0.445970	−2.480177	16.390383
	PreLinK	−0.017499	−0.363789	−1.149061	9.581187	−2886.582154
diamond$_{102}$	STD	0.008664	0.083976	0.700345	−5.672850	−29.031070
	PreLinK	0.032198	−3.442257	36.730925	53.315107	−242.855130
LiF_{72}	STD	0.005474	0.045975	−0.555997	5.231158	−39.033935
	PreLinK	−0.033582	−1.068375	3.412783	−5.336652	149.741714
$(S_8)_5$	STD	0.010877	0.027477	0.233002	−3.120980	−11.007260
	PreLinK	0.135188	0.691967	−2.855973	17.091663	89.495817

[a]Maximum deviation in the corresponding gradient vector from the reference values ($\vartheta = 10^{-15}$) given in 10^{-6} au/a_0 ($\mu H/a_0$). [b]Wall times for the PreLinK algorithm given for the amylose fragments.

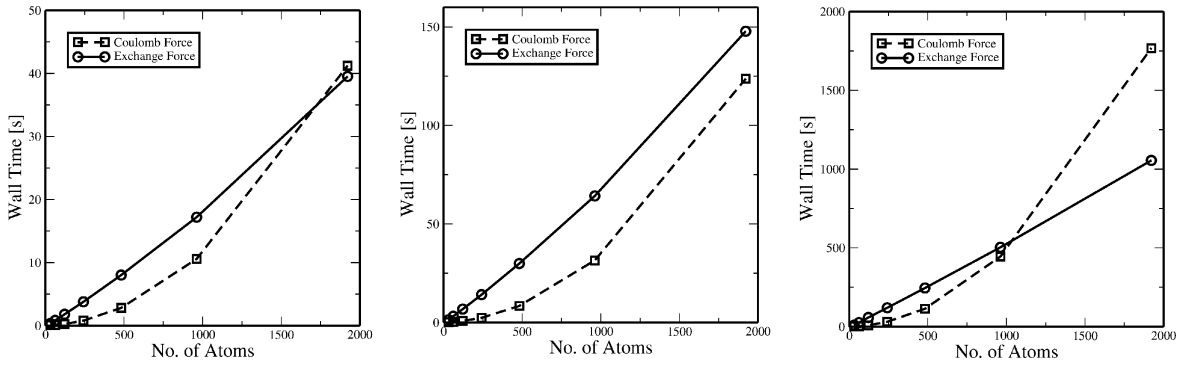


Figure 1. Wall times (in seconds) for Coulomb- and exchange-gradient calculations for a series of linear alkanes C_nH_{2n+2} (n = 10, 20, 40, 80, 160, 320, 640) with HF/SV (left), HF/SVP (middle), and HF/TZVP (right) using four GPUs ($\vartheta_{int} = 10^{-10}$, $\vartheta^{\nabla}_{pre} = 10^{-10}$). The largest system is $C_{640}H_{1282}$, comprising 1922 atoms and 8324 basis functions.

HF/SVP ($\vartheta_{int} = 10^{-10}$ for integrals and $\vartheta_{conv} = 10^{-7}$ for convergence). In Table 1, the largest absolute deviation in the gradient vector is given for a series of gradient thresholds with conventional screening (eq 6) and with our proposed PreLinK prescreening algorithm. Note that the error is, in general, roughly an order of magnitude larger for PreLinK than for the conventional screening in eq 6, as can be expected by considering that, in eq 6, maximum absolute values of the density matrix elements and the Schwarz integral as an upper bound to the true integral value ($\mu\lambda|\nu\sigma$) are used. In contrast, the PreLinK estimate is based on accurate quantities, namely, the density **P** and exchange matrix **K** from the preceding SCF calculation, so that more integrals are discarded for a given threshold. For the amylose fragments, we also give the wall times for the calculations employing different thresholds ϑ^{∇}_{pre} in Table 1. As expected, one can see a steep decrease of the wall time with a decreasing threshold ϑ^{∇}_{pre}. Based on these results, the integral threshold ϑ_{int} appears to be a reasonable and safe choice for ϑ^{∇}_{pre}. Thus, we employed a preselection threshold of $\vartheta^{\nabla}_{pre} = 10^{-10}$ for the remaining calculations in this work.

To analyze the scaling behavior of our algorithm, we computed the gradient of a series of linear alkanes C_nH_{2n+2}

DOI: 10.1021/ct501189u
J. Chem. Theory Comput. 2015, 11, 918−922

Table 2. Wall Times (in Seconds) Using Four GPUs for the Calculation of the Exchange and Coulomb Gradients for a Series of Linear Alkanes[a−c]

	HF/SV				HF/SVP				HF/TZVP			
	dK/dx		dJ/dx		dK/dx		dJ/dx		dK/dx		dJ/dx	
N_A	t (s)	$O(N^x)$	t (s)	$O(N^x)$	t (s)	$O(N^x)$	t (s)	$O(N^x)$	t (s)	$O(N^x)$	t (s)	$O(N^x)$
32	0.3	—	0.1	—	1.3	—	0.1	—	10.03	—	1.19	—
62	0.8	1.3	0.1	0.2	3.1	1.3	0.3	1.2	26.05	1.4	2.99	1.4
122	1.8	1.1	0.2	1.4	6.7	1.1	0.8	1.4	58.33	1.2	8.80	1.6
242	3.7	1.1	0.8	1.7	13.6	1.1	2.4	1.6	120.05	1.0	31.55	1.9
482	7.6	1.1	2.8	1.8	28.1	1.0	8.4	1.8	245.54	1.0	113.79	1.9
962	15.8	1.1	10.6	1.9	57.9	1.0	31.5	1.9	503.83	1.0	445.12	2.0
1922	33.9	1.1	41.2	1.9	122.6	1.1	123.7	2.0	1055.24	1.1	1767.40	2.0

[a]All calculations performed with a conservative integral threshold of $\vartheta_{int} = 10^{-10}$ and a gradient preselection theshold of $\vartheta^{\nabla}_{pre} = 10^{-10}$. [b]$N_A$ denotes the number of atoms; $O(N^x)$ denotes the scaling exponent. [c]Largest system listed is $C_{640}H_{1282}$.

Table 3. Wall Times (in Seconds) Using Four GPUs for the Calculation of the Exchange and Coulomb Gradients for a Series of DNA Fragments and Water Clusters Using HF/SVP[a−c]

	DNA					water			
	dK/dx		dJ/dx			dK/dx		dJ/dx	
N_A	t (s)	$O(N^x)$	t (s)	$O(N^x)$	N_A	t (s)	$O(N^x)$	t (s)	$O(N^x)$
62	7.3	—	0.6	—	204	18.5	—	18.8	—
128	46.0	2.4	3.1	2.3	426	58.4	1.6	59.8	1.9
260	169.0	1.8	13.7	2.0	855	158.2	1.4	163.3	2.1
524	409.9	1.3	55.7	2.0	1707	402.1	1.3	423.0	2.0
1052	862.3	1.1	223.0	2.0	3369	976.3	1.3	1057.4	2.0

[a]All calculations performed with a conservative integral threshold of $\vartheta_{int} = 10^{-10}$ and a gradient preselection theshold of $\vartheta^{\nabla}_{pre} = 10^{-10}$. [b]$N_A$ denotes the number of atoms; $O(N^x)$ denotes the scaling exponent. [c]Largest systems listed are DNA_{16} (16 A−T base pairs) and $(H_2O)_{1123}$.

(n = 10, 20, 40, 80, 160, 320, 640) with HF/SV, HF/SVP, and HF/TZVP on a single GPU server with four GPU devices, with the largest system containing 1922 atoms and 27532 basis functions (TZVP). The wall times are shown in Figure 1, and one can see that the linear-scaling regime is reached with $C_{40}H_{122}$. Again, it has to be stressed that the Coulomb evaluation is very fast even with the $O(N^2)$ J-engine algorithm. This performance gap is far more pronounced for more realistic system such as DNA fragments or water clusters. However, for the larger TZVP basis and $C_{640}H_{1282}$, the exchange gradient is evaluated significantly faster than the Coulomb gradient (1355 vs 1767 s) as compared to the timings for the SCF matrices, where the Coulomb matrix takes 689 s and the exchange matrix takes 1457 s within the SCF calculation. The improved performance of the exchange gradient is due to the reduced resource efforts regarding shared memory on the GPU device, because only two gradient contributions are stored independent of the l quantum numbers of the μ and ν shells.

The wall times and scaling exponents x [$O(N^x)$] for the series of linear alkanes and for DNA fragments and water clusters obtained using HF/SVP are reported in Tables 2 and 3, respectively. Note that we also obtained an asymptotically linear scaling for the water clusters and DNA fragments.

As a final example, we performed hybrid MPI/CUDA calculations for the largest DNA fragment containing 1052 atoms with HF/SVP and up to 10 GPU servers. The results for the wall times to form the nuclear derivatives of the Coulomb and exchange matrices are shown in Figure 2. When 10 computing nodes were used instead of 1, the wall times were strongly reduced from 863 to 102 s and from 220 to 32 s for the exchange and Coulomb forces, respectively. To elucidate the scaling with respect to the number of processes, we also

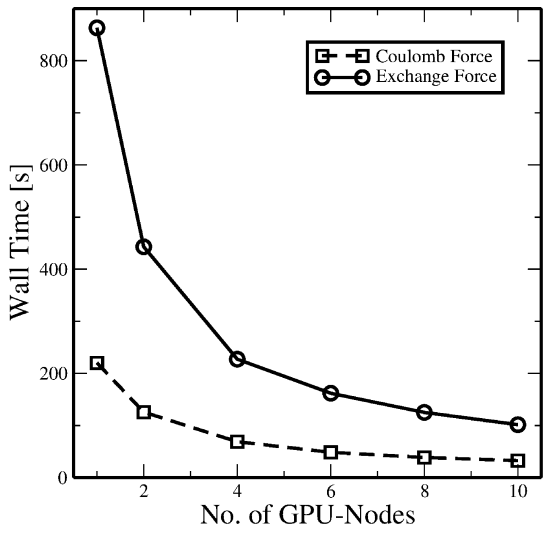


Figure 2. Wall times (in seconds) for Coulomb- and exchange-gradient calculations for a DNA fragment containing 16 A−T base pairs (1052 atoms, 11230 basis functions) with HF/SVP using up to 10 GPU nodes ($\vartheta_{int} = 10^{-10}$, $\vartheta^{\nabla}_{pre} = 10^{-10}$).

show the speedups and parallel efficiency[24] in Figure 3. It can be seen that the speedup for the exchange gradient with up to 10 GPU servers is still approximately 8.5, that is, a parallel efficiency of 0.85. Considering that the DNA fragment contains only 1052 atoms, the performance can still be significantly enhanced by steeply increasing the number of computing units, which indicates a strong scaling behavior. Note again that 10 GPU servers corresponds to the use of 40 GPU cards and, in turn, more than 100000 CUDA cores in total. Here, it should

DOI: 10.1021/ct501189u
J. Chem. Theory Comput. 2015, 11, 918−922

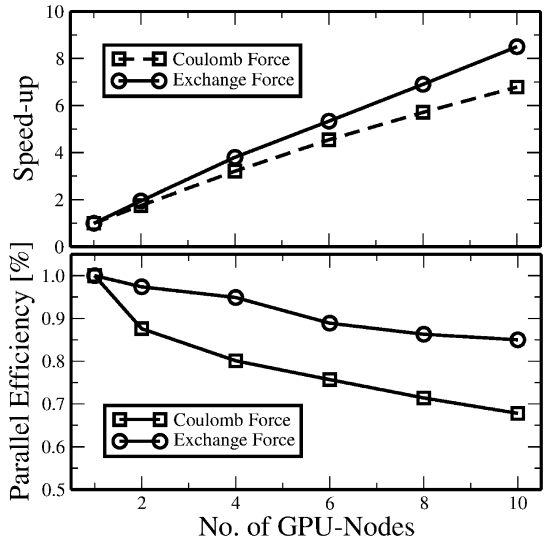


Figure 3. Speedup (top) and parallel efficiency (bottom) for Coulomb- and exchange-gradient calculations for a DNA fragment containing 16 A−T base pairs (1052 atoms, 11230 basis functions) with HF/SVP using up to 10 GPU nodes ($\vartheta_{int} = 10^{-10}$, $\vartheta^{\nabla}_{pre} = 10^{-10}$).

be stressed that this strong scaling behavior can of course also be exploited within massively parallel calculations using CPUs only by employing the fine-grained shell-pair data arrangement and the simple and efficient load balancing due to the predetermination of the significant contributions to the exchange matrix and its nuclear derivatives.

IV. CONCLUSIONS

We have presented a preselective screening scheme for the evaluation of the exchange contribution to the nuclear gradient of the energy. It is shown that this approach is highly suitable not only for GPU architectures but also, in general, for massively parallel computing systems. This is due to the fine-grained data arrangement that allows for efficient parallelization over many computing units, as well as the simple and efficient load balancing over significant contributions only. Regarding the latter point, it should be noted that this approach contrasts with the on-the-fly screening of conventional algorithms, where the load for a single node cannot be accurately determined beforehand, so that significant imbalances can, in principle, arise.

Thus, our PreLinK scheme can be useful not only for GPU-based calculations, but also for calculations on conventional massively parallel supercomputing architectures.

AUTHOR INFORMATION

Corresponding Author

*E-mail: christian.ochsenfeld@uni-muenchen.de.

Notes

The authors declare no competing financial interest.

ACKNOWLEDGMENTS

C.O. acknowledges financial support from the Deutsche Forschungsgemeinschaft (DFG) funding proposal Oc35/4-1. Further financial support was provided by the SFB 749 “Dynamik und Intermediate molekularer Transformationen” (DFG) and the DFG cluster of excellence (EXC 114) Center for Integrated Protein Science Munich (CIPSM).

REFERENCES

(1) Yasuda, K. *J. Comput. Chem.* **2008**, *29*, 334−42.
(2) Yasuda, K. *J. Chem. Theory Comput.* **2008**, *4*, 1230−1236.
(3) Ufimtsev, I. S.; Martínez, T. J. *J. Chem. Theory Comput.* **2008**, *4*, 222−231.
(4) Ufimtsev, I. S.; Martínez, T. J. *Comput. Sci. Eng.* **2008**, *10*, 26−34.
(5) Ufimtsev, I. S.; Martínez, T. J. *J. Chem. Theory Comput.* **2009**, *5*, 1004−1015.
(6) Luehr, N.; Ufimtsev, I. S.; Martínez, T. J. *J. Chem. Theory Comput.* **2011**, *7*, 949−954.
(7) Isborn, C. M.; Luehr, N.; Ufimtsev, I. S.; Martínez, T. J. *J. Chem. Theory Comput.* **2011**, *7*, 1814−1823.
(8) Wu, X.; Koslowski, A.; Thiel, W. *J. Chem. Theory Comput.* **2012**, *8*, 2272−2281.
(9) Asadchev, A.; Allada, V.; Felder, J.; Bode, B.; Windus, T. L.; Gordon, M. S. *J. Chem. Theory Comput.* **2010**, *6*, 696−704.
(10) Kussmann, J.; Ochsenfeld, C. *J. Chem. Phys.* **2013**, *138*, 134114.
(11) Maurer, S.; Kussmann, J.; Ochsenfeld, C. *J. Chem. Phys.* **2014**, 051106.
(12) Kussmann, J.; Beer, M.; Ochsenfeld, C. *WIREs Comput. Mol. Sci.* **2013**, *3*, 614−636.
(13) White, C. A.; Johnson, B. G.; Gill, P. M. W.; Head-Gordon, M. *Chem. Phys. Lett.* **1994**, *230*, 8−16.
(14) Ochsenfeld, C.; White, C. A.; Head-Gordon, M. *J. Chem. Phys.* **1998**, *109*, 1663−1669.
(15) Challacombe, M. *J. Chem. Phys.* **1999**, *110*, 2332−2342.
(16) Kussmann, J.; Ochsenfeld, C. *J. Chem. Phys.* **2007**, *127*, 054103.
(17) Kussmann, J.; Ochsenfeld, C. *J. Chem. Phys.* **2007**, *127*, 204103.
(18) Pulay, P. *Mol. Phys.* **1969**, *17*, 197−204.
(19) Häser, M.; Ahlrichs, R. *J. Comput. Chem.* **1989**, *10*, 104−111.
(20) Ochsenfeld, C. *Chem. Phys. Lett.* **2000**, *327*, 216−223.
(21) Schwegler, E.; Challacombe, M. *J. Chem. Phys.* **1996**, *105*, 2726−2734.
(22) Schwegler, E.; Challacombe, M.; Head-Gordon, M. *J. Chem. Phys.* **1997**, *106*, 9708−9717.
(23) Structures are available online: http://www.cup.lmu.de/pc/ochsenfeld.
(24) Parallel efficiency is the ratio between measured and ideal speedup.

DOI: 10.1021/ct501189u
J. Chem. Theory Comput. 2015, 11, 918−922