Energy-Efficient Computational Chemistry: Comparison of x86 and ARM Systems

Kristopher Keipert,[†] Gaurav Mitra,[‡] Vaibhav Sunriyal,[§] Sarom S. Leang,[†] Masha Sosonkina,[§] Alistair P. Rendell,[‡] and Mark S. Gordon*,[†]

[†]Department of Chemistry and Ames Laboratory, Iowa State University, Ames, Iowa 50011-3111, United States
[‡]Research School of Computer Science, Australian National University, Acton, Australian Capital Territory 0200, Australia
[§]Department of Modeling and Simulation, Old Dominion University, Norfolk, Virginia 23529, United States

S *Supporting Information*

ABSTRACT: The computational efficiency and energy-to-solution of several applications using the GAMESS quantum chemistry suite of codes is evaluated for 32-bit and 64-bit ARM-based computers, and compared to an x86 machine. The x86 system completes all benchmark computations more quickly than either ARM system and is the best choice to minimize time to solution. The ARM64 and ARM32 computational performances are similar to each other for Hartree−Fock and density functional theory energy calculations. However, for memory-intensive second-order perturbation theory energy and gradient computations the lower ARM32 read/write memory bandwidth results in computation times as much as 86% longer than on the ARM64 system. The ARM32 system is more energy efficient than the x86 and ARM64 CPUs for all benchmarked methods, while the ARM64 CPU is more energy efficient than the x86 CPU for some core counts and molecular sizes.
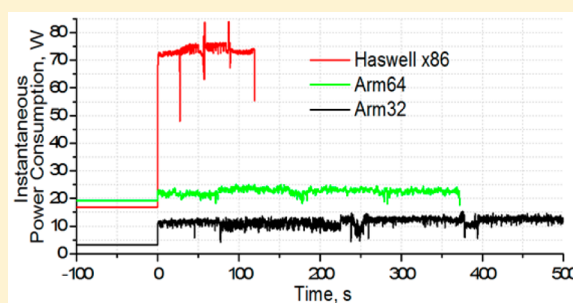
1. INTRODUCTION

It is widely recognized that energy usage is a major bottleneck in the pursuit of improving computational performance. This reflects in part the demise of Dennard scaling[1,2] but also fundamental limitations on the energy that can be provided to a single chip regardless of the transistor count. Consequently, computational application developers and users will increasingly need to consider both speed and energy and the interplay between these two metrics. Clear evidence of this trend is seen in the rapid rise of energy-optimized accelerators and co-processors and in the availability of advanced power management facilities on modern processors.

In the pursuit of new energy-efficient hardware designs, low-power mobile computing driven by ARM-based system-on-chip (SoC) processors has aroused significant interest within the high performance computing (HPC) community. These systems are designed with energy efficiency in mind, typically utilizing 32-bit CPUs that are optimized for 32-bit arithmetic. This may be adequate for mobile applications, but for quantum chemistry applications large memory and double precision floating point arithmetic is usually required. And while ARM-based devices are now being used in servers, it is not yet clear whether either 32-bit ARM-based SoC computers or more recent 64-bit ARM CPUs are viable for HPC workloads.

The popular GAMESS[3] quantum chemistry package is used on a wide range of HPC architectures and is therefore a useful test bed for assessing the performance of novel architectures. The present work focuses on measuring performance and energy-to-solution of GAMESS workloads on two ARM-based systems, a 32-bit NVIDIA Jetson TK1 and a 64-bit APM Xgene1 X-C1. The two ARM systems are also compared to a 64-bit Haswell x86 Intel Xeon-E5 processor. A set of commonly used computational chemistry techniques are evaluated, namely, Hartree−Fock (HF) self-consistent field (SCF),[4] density functional theory (DFT),[5−7] and second-order Møller−Plesset (MP2)[8,9] energy and gradient calculations.

2. COMPUTATIONAL DETAILS

The GAMESS performance evaluations used a benchmark set of molecules that contains 99−509 basis functions when using the 6-31G(d)[10−12] basis set. Power measurements were obtained for DFT, HF SCF, MP2 energy, and MP2 gradient calculations. The PBE0 functional[13−15] was used in all DFT calculations. All parallel computations were performed by the creation of compute and data server processes for each physical CPU core via the distributed data interface[16−18] (DDI) in GAMESS. In all benchmarks, two-electron integrals were calculated at each (direct) SCF step.

The molecules used for benchmarking are listed in Table 1. The molecular geometries used for all benchmark calculations

Table 1. Benchmark Molecule Specifications

molecule	chemical formula	number of basis functions
pentane	C_5H_{12}	99
asparagine	$C_4H_8N_2O_3$	151
nicotine	$C_{10}H_{14}N_2$	208
trinitrotoluene (TNT)	$C_7H_5N_3O_6$	250
indigo	$C_{16}H_{10}N_2O_2$	320
tetrahydrocannabinol (THC)	$C_{21}H_{30}O_2$	405
adenosine triphosphate (ATP)	$C_{10}H_{16}N_5O_{13}P_3$	509

were obtained by HF/cc-PVDZ[19,20] optimizations. The coordinates and the total wall times are provided in the Supporting Information.

2.1. Hardware. The 32-bit ARM machine is an NVIDIA Jetson TK1, configured with a quad-core 2.35 GHz ARM Cortex-A15 CPU (ARMv7-A architecture) paired with 2 GB of LP-DDR3 RAM. The 64-bit ARM machine is an AppliedMicro (APM) X-Gene X-C1 with an 8-core 2.4 Ghz APM883208-X1 CPU (ARMv8-A architecture) and 16 GB of

DDR3 memory. The Haswell x86 machine utilizes an 18-core Intel Xeon E5-2699 v3 CPU clocked to the maximum turbo frequency of 3.6 GhZ with hyperthreading disabled and 32 GB of DDR4 memory.

To consider the impact on computational performance due to the different system memory types used in each machine, the DRAM read and write bandwidths were measured with the LMBench[21] performance analysis suite. For small memory transactions, 1.05MB in size, the read/write bandwidth is 18.0/12.6 GB/s for the x86 system, 5.9/9.5 GB/s for the ARM64 system, and 4.3/9.2 GB/s for the ARM32 system. For larger memory transactions, 67.11 MB in size, the read/write bandwidth is 10.6/7.6 GB/s for the x86 system, 5.1/9.3 GB/s for the ARM64 system, and 1.2/3.2 GB/s for the ARM32 system. The 32-bit x86 4 GB physical memory capacity limitation is expanded to 1 TB for the 32-bit ARMv7-A architecture via 40-bit physical memory address space support. Also, note that on the 32-bit ARM system double precision numbers are moved between the CPU registers and system memory locations in two 4-byte segments, while on the 64-bit CPUs the entire 8-byte number can be moved to memory with a single instruction.
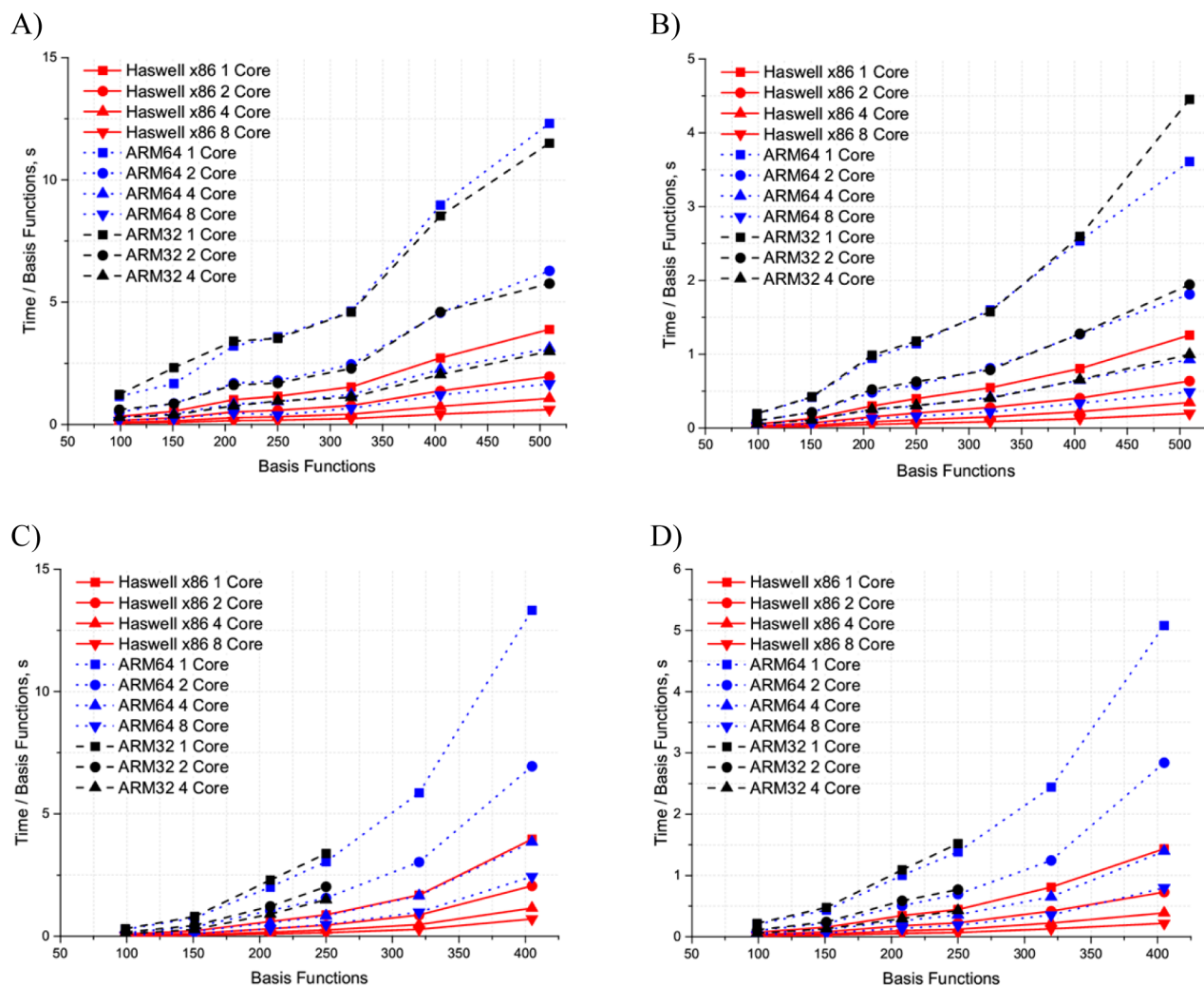
Figure 1. Computation times per basis function for (A) DFT energy, (B) HF SCF energy, (C) MP2 energy, and (D) MP2 gradient benchmark calculations.

2.2. Software. GAMESS was compiled for the x86 and ARM32 systems with GCC v4.8 and with GCC v5.1 on the ARM64 system (v5.1 is the first version with compiler tuning capabilities for the X-Gene1 CPU). BLAS routines were provided by the ATLAS v3.11 math library,[22] natively built for each machine to take advantage of automatic tuning of BLAS routines for each hardware type. The Red Hat GNU/Linux operating system was used on the x86 system with kernel version 3.10.0. The Ubuntu GNU/Linux operating system was used for both ARM systems. The kernel versions were 3.13.0 for the ARM64 system and 3.10.40 for the ARM32 system.

2.3. Energy/Power Measurements. High-accuracy energy measurements were obtained by uniquely adapting the measurement method for each system. The running average power limit (RAPL)[23] software interface which reads energy consumption information from model-specific registers on an x86 CPU was used to measure the DC power consumption of the 18-core Haswell CPU. RAPL measurements were reported every 0.2 s.

The DC power consumption of the 64-bit ARM CPU was measured by placing a Fluke i1010 AC/DC current clamp around the wire from the power supply unit (PSU) that supplies power to the CPU. The current clamp was connected to a multimeter which stored current measurements every 0.5 s on a remote server.

The current used by the ARM32 Jetson system was measured using a uCurrent Gold high-precision current measurement tool and an mbed LPC1768 microcontroller with a 12-bit analog-to-digital (ADC) converter, ranging from 0 to 3.3 V. To measure the current, a 0 V supply line for the system was routed through the current side of the uCurrent Gold. The ADC was then connected across the voltage output pins of the uCurrent Gold. Serial connections were used to send start and stop signals from the Jetson to the measuring device and to send the measurements from the measuring device to the measuring computer.

The power measurements reported for both the x86 Haswell and ARM64 systems are only for the CPU. The RAPL interface used for measurements of the x86 system provides energy consumption information for the isolated CPU socket. The current clamp used for ARM64 measurements probes the +12 V wire from the ATX power supply unit that powers the CPU only. The ARM32 Jetson uses an AC adapter that has a single power supply output. ARM32 energy measurements are for the entire system and include power consumption for components such as the fan and memory in addition to the CPU.

3. RESULTS AND DISCUSSION

3.1. Computational Efficiency. For performance comparisons of different computer systems, the same number of cores is used on each system. The CPU wall clock times for the various methods on the different platforms are shown in Figure 1 normalized according to the number of basis functions in each molecule. [A set of figures that provide an alternative view of the same data is presented in the Supporting Information.] For all methods employed there is an increase in computational time per basis function as the system size increases. This reflects the worse-than-linear scaling of all methods.

For the DFT energy computations in Figure 1A, the x86 single-core performance is consistently better by a factor of ∼3 than both ARM CPUs, with little change in the ratio as

the system size increases. The x86 performance relative to ARM64 decreases to a factor of ∼2.8 when 8 cores are used. The performance of the ARM32 and ARM64 CPUs are within ±10% of each other.

The results for the HF SCF energy shown in Figure 1B are similar to those for the DFT energy. That is, on average the ARM32 computation is 3.3% slower than the ARM64 computation while the ARM64 system takes on average 3.17×/3.16×/2.90×/2.69× longer than the x86 system for HF calculation execution time with 1/2/4/8 cores.
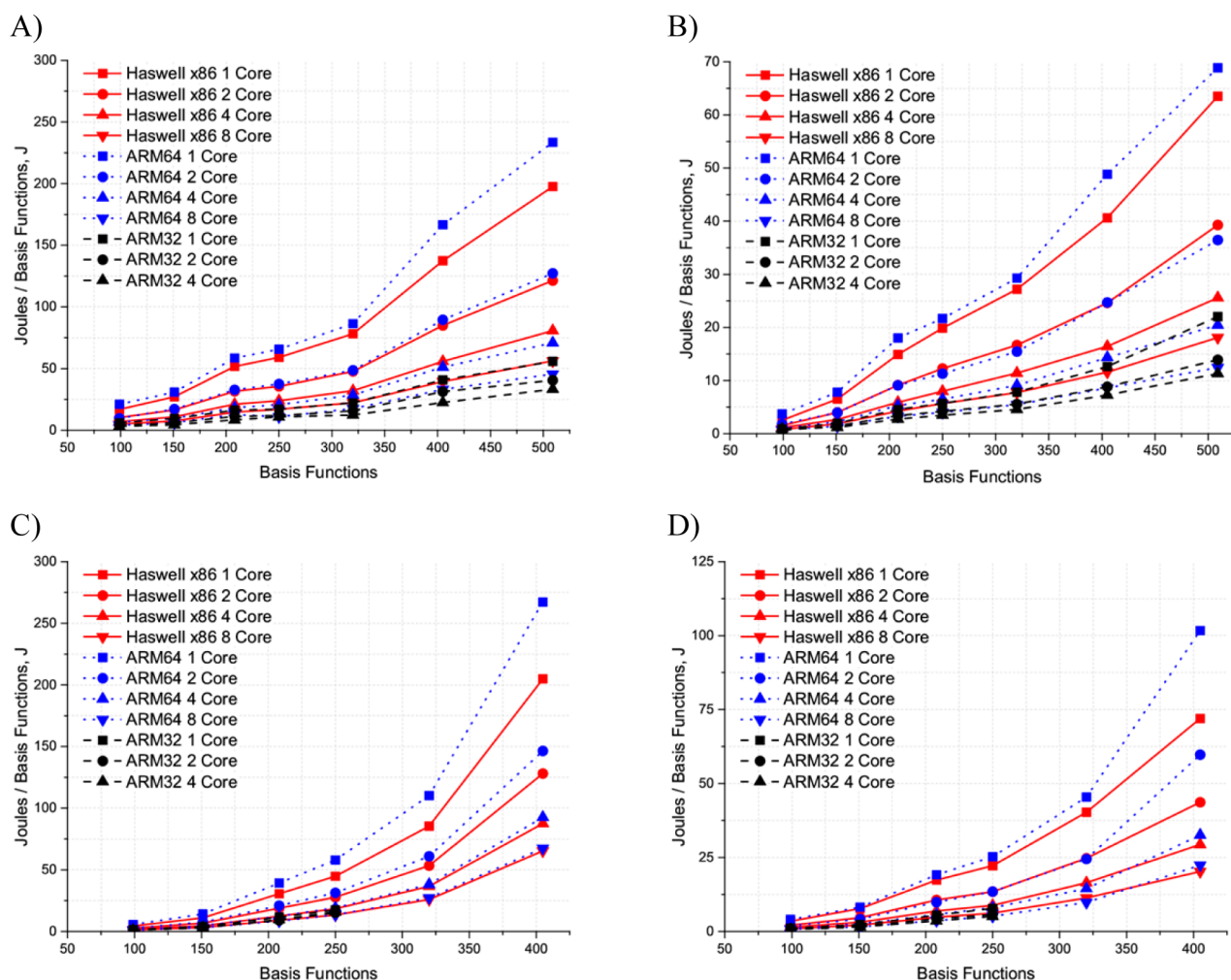
For the MP2 energy and gradient calculations, memory requirements limit the calculations on the ARM32 system to four molecules that contain 99−250 basis functions, while other restrictions due to the DDI implementation limit the calculations to six molecules in the range of 99−405 basis functions on the ARM64 and x86 systems.

The computation times for the MP2 energy calculations (Figure 1C) show the largest difference in performance between ARM32 and ARM64 among all analyzed computation types. Furthermore, the performance degradation of the ARM32 CPU relative to ARM64 worsens with increasing system size and the number of cores used. For example, the MP2 energy computation time for the smallest system, pentane, is 8.6%/13.6%/20.7% greater on 1/2/4 ARM32 cores compared to the same number of ARM64 cores, and 10.0%/22.8%/44.1% greater on 1/2/4 ARM32 cores than the same number of ARM64 cores for the largest molecule that can be run on ARM32 (TNT). By contrast no such correlation is found between the system size or the number of active cores and the relative computational performance when comparing the x86 system to the ARM64 system. On average, the MP2 energy calculations take 3.27×/3.37×/3.32×/3.25× more execution time on the ARM64 system than the x86 system for the 1/2/4/8 cores.

For the MP2 gradient (Figure 1D), there is a weak correlation between the number of CPU cores used and the relative system performance for ARM64 vs ARM32, but no such correlation is observed for molecule size. On average, the ARM64 system executes MP2 gradient calculations in 8.5%/10.3%/10.4% less time than the ARM32 system for 1/2/4 cores. The performance benefits of the x86 system relative to the ARM64 system decrease when the number of cores used for the computation is increased. With the exception of the largest molecule (THC; 405 basis functions) the MP2 gradient calculation using the x86 system is on average 2.95×/2.89×/2.80×/2.67× faster than the ARM64 system with 1/2/4/8 cores. No consistent correlation between system size and relative performance of x86 vs ARM64 is observed, but the relative advantage in computational speed for the x86 machine relative to the ARM64 system for the MP2 gradient calculation is greatest for the largest molecule: 3.54×/3.90×/3.61×/3.65× with 1/2/4/8 cores. In general, the ARM32 system performance is worse than the ARM64 system performance for MP2 calculations in contrast to the similar performance observed for the less memory-intensive HF SCF and DFT energy calculations. This degradation in performance may be due to the relatively low read and write bandwidths which were measured for the LPDDR3 RAM of the ARM32 device.

3.2. Energy Consumption. Figure 2 shows the energy consumption per basis function for (A) the DFT energy, (B) the HF SCF energy, (C) the MP2 energy, and (D) the MP2 gradient calculations measured for the x86, ARM64, and

Figure 2. Energy consumption per basis function for (A) DFT energy, (B) HF SCF energy, (C) MP2 energy, and (D) MP2 gradient benchmark calculations.

ARM32 systems. For the DFT calculations averaged over all molecules the ARM32 system requires 31.8%/36.5%/44.3% of the energy consumed by the x86 CPU for 1/2/4 core jobs, while the ARM64 CPU requires 116.2%/102.9%/89.5%/79.5% of the x86 CPU energy for calculations on 1/2/4/8 cores. The HF SCF energy calculations (Figure 2B) exhibit similar trends for the x86 and ARM64 CPUs for all core counts; that is, the x86 calculation is always slightly more energy efficient for all benchmark molecules on 1 core and always less efficient than the ARM64 CPU on 4 and 8 cores. The ARM32 system consumes an average of 31.1% of the x86 CPU energy for 1 core, 36.3% for 2 cores, and 48.6% for 4 cores.

The MP2 energy efficiency results are shown in Figure 1C. The ARM64 calculations on average and using 1/2 cores consumes 29.1%/11.1% more energy than 1/2 x86 cores; when using 4 or 8 cores the x86 machine falls within ±3% of the analogous results obtained on the ARM64 machine. The ARM32 system is the most energy efficient, but this energy efficiency rapidly diminishes when increasing the number of cores. On 1/2/4 cores the MP2 energy calculations on the ARM32 system and averaged over all molecules uses 36.5%/48.4%/71.8% of the energy required for the equivalent calculations on the x86 system.

For the MP2 gradient computations, the energy consumption of the ARM64 CPU averaged over all molecules is 117%/102%/90%/86% of the x86 CPU energy used for the same computations on 1/2/4/8 cores. On the ARM32 system the 99−250 basis function computations consume on average 31.9%/40.0%/51.0% of the energy used by the x86 CPU for 1/2/4 cores, similar to the relative energy consumption for the DFT energy and HF SCF computations.

3.3. Busy/Idle Core Energy Usage. When running a calculation on less than the total number of CPU cores, the unused cores consume energy in the idle state. To examine the efficiency of running parallel versus multiple copies of sequential code, and in order to estimate the energy consumed by busy and idle cores, the energy usage was measured for MP2 gradient calculations on TNT performed using varying levels of CPU core saturation. The energies and times used per basis function are shown in Table 2. The 1-core values correspond to single 1-core computations while all remaining cores are idle. The 8-core values correspond to 8 cores used for a single computation running in parallel. This fully saturates the available ARM64 cores but leaves 10 idle cores for the x86 CPU. Also shown is the energy usage for running 8 × 1-core jobs simultaneously.

The 8 × 1-core parallel and 8-core schemes have similar energy consumption and calculation times for all computation

Table 2. Energy Consumption and Computation Time Per Basis Function of x86 and ARM64 CPUs for TNT (250 Basis Functions) MP2 Gradient Calculation Steps for 1-Core and 8-Core Calculations and for 8 and 18 1-Core Calculations in Parallel

		energy/basis function, J	time/basis function, s
HF SCF energy	x86 1-core	19.849	0.397
	x86 8-core	5.658	0.062
	x86 8 × 1-core, parallel	5.665	0.063
	x86 18 × 1-core, parallel	4.139	0.029
	ARM64 1-core, serial	21.676	1.144
	ARM64 8-core	4.003	0.158
	ARM64 8 × 1-core, parallel	4.155	0.155
MP2 energy	x86 1-core	44.784	0.871
	x86 8-core	13.362	0.145
	x86 8 × 1-core, parallel	14.597	0.157
	x86 18 × 1-core, parallel	13.352	0.097
	ARM64 1-core	57.932	3.040
	ARM64 8-core	13.435	0.480
	ARM64 8 × 1-core, parallel	13.855	0.476
MP2 gradient	x86 1-core	22.217	0.444
	x86 8-core	6.190	0.069
	x86 8 × 1-core, parallel	6.133	0.068
	x86 18 × 1-core, parallel	3.884	0.032
	ARM64 1-core	25.247	1.384
	ARM64 8-core	5.132	0.188
	ARM64 8 × 1-core, parallel	5.281	0.187

steps for both x86 and ARM64 CPUs with the exception of the MP2 energy on the x86 system. That is, it is as efficient to run 8 identical single-core calculations simultaneously as it is to run one calculation in parallel using 8 cores, and then to repeat that calculation 8 times. For the x86 MP2 energy there is a slight difference: the 8 × 1-core parallel scheme consumes 9.2% more energy and takes 8.3% more execution time per basis function compared to the 8-core computation. Overall the results suggest that the HF SCF, MP2 energy, and MP2 gradient algorithms in GAMESS do not have significant computational cost overhead for parallel task coordination. Also, there is no significant off-chip memory or I/O contention when running 8 compute processes in parallel.

To calculate the power consumption of individual busy and idle cores, their energy usage is approximated using eqs 1 and 2, respectively. $core_{max}$ is the number of physical cores per CPU: 18 for x86 and 8 for ARM64. In eq 1, the "saturated" subscript indicates the value for $core_{max}$ jobs running simultaneously, each using 1 core. This corresponds to the "x86 18 × 1-core, parallel" and "ARM64 8 × 1-core, parallel" values (Table 2). In eq 2, the "n-core" subscript indicates the value for a single job running on n cores. The value $n = 1$ is chosen for idle core calculations in this study and corresponds to the "x86 1-core" and "ARM64 1-core" values in Table 2.

$$\text{busy core power} = \frac{\text{energy}_{\text{saturated}}/\text{time}_{\text{saturated}}}{\text{core}_{\text{max}}} \tag{1}$$

$$\text{idle core power}$$
$$= \frac{(\text{energy}_{n\text{-core}}/\text{time}_{n\text{-core}}) - (n \times \text{busy core power})}{\text{core}_{\text{max}} - n} \tag{2}$$

The calculated power consumption per busy core during the HF SCF/MP2 energy/MP2 gradient calculations is 7.93 W/ 7.65 W/6.74 W for the x86 CPU and 3.35 W/3.64 W/3.53 W

for the ARM64 CPU. The calculated power consumption per idle core during the HF SCF/MP2 energy/MP2 gradient calculations is 2.47 W/2.57 W/2.55 W for the x86 CPU and 2.62 W/2.21 W/2.10 W for the ARM64 CPU. Extrapolating the average idle core power consumption during the three calculation types to the $core_{max}$ value, the calculated total power consumption for an idle CPU is 45.57 W for the x86 CPU and 18.46 W for the ARM64 CPU.

For comparison power usage was measured experimentally for both CPUs in the idle state over a period of 1 h. It was found that while on the ARM64 the average measured value of 19.10 W agreed well with the derived value of 18.46 W, the measured value of 16.83 W on the x86 CPU is significantly less than the derived value of 45.57 W. This 2.7× reduction in power usage presumably reflects the fact that the Haswell x86 CPU includes the C7 sleep state feature to lower idle core power consumption when the entire CPU is idle.

In terms of ideal energy efficiency for the quantum chemistry algorithms analyzed, the results clearly demonstrate that it is much more important to saturate all available cores regardless of the number of cores per computation than it is to choose between parallel and back-to-back serial computation executions. This is particularly true for the Haswell architecture, which incurs a relatively large incremental energy cost when left in the completely idle CPU state. This is not observed for the ARM64 CPU.

3.4. Energy Usage Trace. To explore whether energy usage changes significantly during the course of the calculations, Figure 3 shows a trace of the instantaneous power consumption of the x86 and ARM64 CPUs and ARM32 system during an MP2 gradient calculation on TNT running on four CPU cores. The average idle energy consumption over a 1 h measurement is plotted in Figure 3 for the x86, ARM64, and ARM32 systems, indicated by times from −100 to 0 s.
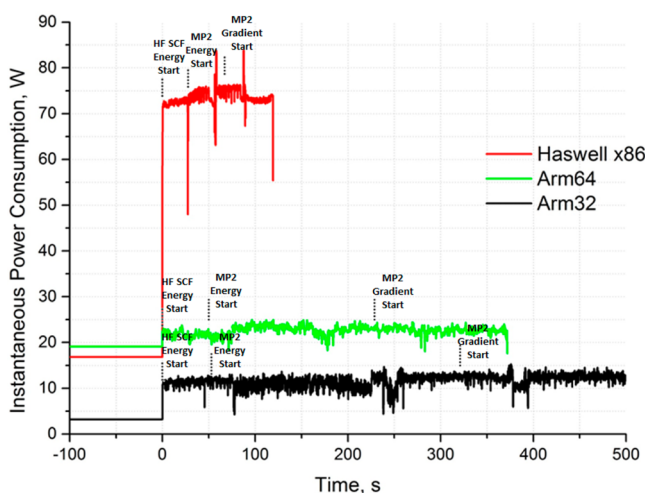
Figure 3. Power trace during TNT MP2 gradient calculation for ARM64 and x86 CPU, ARM32 systems, and four CPU cores.

The average x86 idle CPU power consumption of 16.83 W is initially lower than the 19.10 W average of the ARM64 CPU, but within 1 s of the HF SCF calculation, power consumption increases by 71.95 W for the x86 CPU, but only to 23.12 W for the ARM64 CPU. The ARM32 system uses less power than either ARM64 or x86, with an average idle power consumption of 3.21 W which increases to 10.58 W after 1.0 s has elapsed in the HF SCF calculation.

Table 3 shows the mean, standard, and relative standard deviations of the x86, ARM64, and ARM32 systems during the CPU power trace calculation. On all machines, once the computation has begun, fluctuations in power usage are relatively small. For the x86 and ARM64 CPUs, the mean power consumption is highest for the MP2 energy calculation, followed by the MP2 gradient and the HF SCF calculations. The ARM32 MP2 gradient calculation consumes slightly more power in the gradient step, followed by the HF SCF energy and the MP2 energy calculation. The standard deviation of CPU power consumption is highest for the x86 CPU for each calculation step of the power trace at 1.57 W for the HF SCF step, 2.17 W for the MP2 energy step, and 2.52 W for the MP2 gradient step. The relative standard deviation, which takes the magnitude of the average power consumption into account, is lowest for the x86 CPU at 2.52% for the HF SCF step, 2.17% for the MP2 energy step, and 1.57% for the MP2

gradient step. The ARM64 CPU power consumption is the most consistent between calculation steps with a standard deviation of 0.86 W for the HF SCF step, 0.98 W for the MP2 energy step, and 0.81 W for the MP2 gradient step.

4. CONCLUSIONS

Supercomputers capable of exascale level computations will greatly extend the complexity of feasibly solvable problems in computational sciences. The most significant barrier to exascale supercomputers is the relatively poor energy efficiency of modern computer hardware. To reach the exascale, it is therefore imperative that improvements in CPU technology address both computational throughput and energy efficiency. This work has explored these issues in the context of a widely used quantum chemistry package running on ARM32, ARM64, and x86 processors. For all methods and molecules considered the x86 CPU is the clear choice in terms of minimizing time to solution, in the order of x86 < ARM64 < ARM32. Although the 32-bit architecture limits the utility of the ARM32 system for quantum chemistry calculations, it offers the best performance in terms of energy efficiency with a general ordering of energy-to-solution of ARM32 < x86 < ARM64. It appears that the transition from ARM32 to ARM64 technology comes at a significant cost to energy usage without a significant increase in performance. Whether the latter is in part a reflection on the immaturity of the ARM64 compiler and runtime remains to be seen.

■ ASSOCIATED CONTENT

Ⓢ Supporting Information

The Supporting Information is available free of charge on the ACS Publications website at DOI: 10.1021/acs.jctc.5b00713.

> Benchmark moderate geometries, calculation total wall clock times, and calculation computation times and calculation energy consumption normalized to x86 1-core (PDF)

■ AUTHOR INFORMATION

Corresponding Author
*E-mail: mark@si.msg.chem.iastate.edu.

Table 3. Mean, Standard Deviation, and Relative Standard Deviation of Instantaneous Power Consumption during 250 Basis Function MP2 Gradient Calculation for ARM64, x86 CPU, and ARM32 Systems

	mean, W	std dev, W	rel std dev, %
x86			
Hartree−Fock SCF energy	72.15	2.52	3.50
MP2 energy	74.58	2.17	2.91
MP2 gradient	72.98	1.57	2.15
ARM64			
Hartree−Fock SCF energy	21.57	0.86	3.99
MP2 energy	22.96	0.98	4.26
MP2 gradient	22.56	0.81	3.60
ARM32			
Hartree−Fock SCF energy	11.31	0.97	8.58
MP2 energy	10.61	1.71	16.09
MP2 gradient	11.96	0.84	7.06

■ REFERENCES

(1) Dennard, R. H.; Rideout, V. I.; Bassous, E.; LeBlanc, A. R. *IEEE J. Solid-State Circuits* **1974**, *9*, 256−268.

(2) The Impact of Dennard's Scaling Theory. *IEEE Solid-State Circuits Society News*, **2007**, *12* (1).

(3) Gordon, M. S.; Schmidt, M. W. *Theory and Applications of Computational Chemistry: The First Forty Years*; Dykstra, C. E., Frenking, G., Kim, K. S., Scuseria, G. E., Eds.; Elsevier: Amsterdam, 2005; Chapter 41, pp 1167−1189, DOI: 10.1016/B978-044451719-7/50084-6.

(4) Szabo, A.; Ostlund, N. S. *Modern Quantum Chemistry: Introduction to Advanced Electronic Structure Theory*; Dover: Mineola, NY, USA, 1996.

(5) Hohenberg, P.; Kohn, W. *Phys. Rev.* **1964**, *136*, 864−871.

(6) Kohn, W.; Sham, J. *Phys. Rev. A* **1965**, *140*, 1133−1138.

(7) Parr, R.; Yang, W. *Density-Functional Theory of Atoms and Molecules*; International Series of Monographs on Chemistry; Oxford University Press: New York, NY, USA, 1989.

(8) Møller, C.; Plesset, M. S. *Phys. Rev.* **1934**, *46*, 618−622.

(9) Bartlett, R. J. *Annu. Rev. Phys. Chem.* **1981**, *32*, 359−401.

(10) Ditchfield, R.; Hehre, W. J.; Pople, J. A. *J. Chem. Phys.* **1971**, *54*, 724−728.

(11) Hehre, W. J.; Ditchfield, R.; Pople, J. A. *J. Chem. Phys.* **1972**, *56*, 2257−2261.

(12) Rassolov, V. A.; Ratner, M. A.; Pople, J. A.; Redfern, P. C.; Curtiss, L. A. *J. Comput. Chem.* **2001**, *22*, 976−984.

(13) Perdew, J. P.; Burke, K.; Ernzerhof, M. *Phys. Rev. Lett.* **1996**, *77*, 3865−3868.

(14) Perdew, J. P.; Burke, K.; Ernzerhof, M. *Phys. Rev. Lett.* **1997**, *78*, 1396.

(15) Adamo, C.; Barone, V. *J. Chem. Phys.* **1999**, *110*, 6158−6170.

(16) Fletcher, G. D.; Schmidt, M. W.; Bode, B. M.; Gordon, M. S. *Comput. Phys. Commun.* **2000**, *128*, 190−200.

(17) Olson, R. M.; Schmidt, M. W.; Gordon, M. S.; Rendell, A. P. Enabling the Efficient Use of SMP Clusters: The GAMESS/DDI Model. In *Proceedings of the 2003 ACM/IEEE Conference on Supercomputing*, Phoenix, AZ, USA, Nov. 15−21, 2003; ACM: New York, NY, USA, 2003; DOI: 10.1145/1048935.1050191.

(18) Fedorov, D. G.; Olson, R. M.; Kitaura, K.; Gordon, M. S.; Koseki, S. *J. Comput. Chem.* **2004**, *25*, 872−880.

(19) Dunning, T. H., Jr. *J. Chem. Phys.* **1989**, *90*, 1007−1023.

(20) Woon, D. E.; Dunning, T. H., Jr. *J. Chem. Phys.* **1995**, *103*, 4572−4585.

(21) McVoy, L.; Staelin, C. lmbench: Portable tools for performance analysis. In *Proceedings of the 1996 Annual Conference on USENIX Annual Technical Conference* (ATEC '96), San Diego, CA, USA, Jan. 22−26, 1996; USENIX Association: Berkeley, CA, USA, 1996; p 23-23.

(22) Whaley, R. C.; Petitet, A.; Dongarra, J. J. *Parallel Comput.* **2001**, *27*, 3−35.

(23) David, H.; Gorbatov, E.; Hanebutte, Ulf, R.; Khanna, R.; Le, C. RAPL: Memory power estimation and capping. *ACM/IEEE International Symposium on Low-Power Electronics and Design*, Austin, TX, USA, Aug. 18−20, 2010; IEEE: New York, NY, USA, 2010; pp 189−194.