Intel Woodcrest: An Evaluation for Scientific Computing

Philip C. Roth and Jeffrey S. Vetter
Computational Science and Mathematics Division
Oak Ridge National Laboratory
One Bethel Valley Road
Oak Ridge, TN 37831
{rothpc,vetter}@ornl.gov

Abstract

Intel recently began shipping its Xeon[1] 5100 series processors, formerly known by their "Woodcrest" code name. To evaluate the suitability of the Woodcrest processor for high-end scientific computing, we obtained access to a Woodcrest-based system at Intel and measured its performance first using computation and memory micro-benchmarks, followed by full applications from the areas of climate modeling and molecular dynamics. In most cases, the Woodcrest showed excellent performance compared to a test system managed by our research group that uses Opteron processors from Advanced Micro Devices, even when taking differences in clock speed into account. Our evaluation suggests the Woodcrest to be a compelling foundation for future leadership class systems for scientific computing.

1 Introduction

In late June 2006, Intel began shipping its Xeon 5100 series processors. These processors, code-named "Woodcrest," are dual-core parts using the company's Core architecture. To evaluate the suitability of the Woodcrest processor for scientific computing, we obtained access to a system at Intel with two Woodcrest processors through an early access program. We benchmarked the system following the methodology used in past early evaluations of HPC systems at Oak Ridge National Laboratory [3, 9], adapted slightly to reflect differences in the scales of the systems under consideration. In our evaluation, we used not only computation and memory micro-benchmarks, but also three full scientific applications from the areas of climate modeling and molecular dynamics. Because the AMD Opteron processor is featured prominently in current systems from HPC vendors like Cray Inc., we compared the Woodcrest system's performance with that of an Opteron-based system in our Experimental Computing Laboratory (ExCL) with a similar socket organization. Our evaluation suggests the Woodcrest to be a compelling foundation for building leadership class systems for scientific computing.

2 The Woodcrest Processor

The Intel Woodcrest processor contains two processor cores and a cache hierarchy. Each processor core employs Intel's next generation Core microarchitecture [7]. Cores using this microarchitecture can sustain execution of four instructions per clock cycle, compared to three instructions per cycle with previous Intel microarchitectures. Also, cores with this microarchitecture can execute 128-bit SIMD instructions at a rate of one per cycle; the previous generation microarchitecture could produce one 128-bit SIMD result every two cycles. Furthermore, the microarchitecture includes both a floating-point multiply unit and a floating-point add unit, each of which can operate on two packed double-precision values each cycle using 128-bit SIMD instructions. Thus, each Woodcrest core is capable of producing four double-precision floating-point results per clock cycle. In contrast, current generation AMD Opteron processors can produce at most two double-precision floating point results per clock cycle [1].

The Woodcrest processor memory hierarchy is implemented with a combination of on-chip memory caches, an off-chip memory controller, and Fully Buffered Dual Inline Memory Modules (FB-DIMMs). Each processor core has its own L1 caches (16KB for instructions and 16KB for data), but a 4MB unified L2 cache is shared by both cores. Memory accesses that are not satisfied in the processor caches are transferred via front-side bus (FSB) to a

[1] Product names are trademarks of their respective owners.

memory controller. The Woodcrest processor line uses FB-DIMM memory that provides better bandwidth than DDR and DDR2 memory technology. The Intel memory hierarchy design, especially for Woodcrest, differs substantially from the AMD approach. In current AMD dual-core processors, cache is not shared between cores. Furthermore, a memory controller is located on the processor chip with some system memory being local to each processor socket, a design that AMD argues will scale better than one with an off-chip memory controller as the number of processors increases. However, the AMD approach results in a Non-Uniform Memory Access (NUMA) architecture, where the time required to access a memory location differs depending on whether the location is local to the processor performing the access. On such architectures, a good process placement can be critical for achieving good performance. In contrast, the Intel design results in a Uniform Memory Access (UMA) architecture that may be more forgiving to bad process placement than a NUMA design. However, process placement cannot be ignored on systems with Woodcrest processors because the Woodcrest cores share an L2 cache.

The Woodcrest processor line includes parts clocked from 1.6 GHz (Xeon 5110) to 3.0 GHz (Xeon 5160). The Thermal Design Point (TDP) for the entire line except the 5160 model is 65W; the TDP for that part is 80W.

3 Test System Configurations

We used two systems in this evaluation: one with Intel Woodcrest processors and one with AMD Opteron processors. At the time our evaluation, both systems had two dual-core processors for a total of four cores per system.

The Woodcrest system contained two Xeon 5160 dual-core processors. The processor clock rate was 3.0 GHz and the front-side bus rate was 1333 MHz. The system contained 8 GB of Fully Buffered DIMM (FB-DIMM) memory. We obtained access to the evaluation Woodcrest system through the Intel Remote Access Service. This service provides access to the latest Intel technologies and tools via the Internet for evaluation, validation, and optimization activities.

The Opteron system contained two Opteron 275 dual-core processors, running at 2.2 GHz. This test system contained 4 GB DDR-400 memory. The system is located in a machine room with other ExCL systems

Both systems used a Linux 2.6 kernel. On the Woodcrest system, we used the Intel Fortran and C/C++ compilers (version 9.1), the Intel Math Kernel Library (version 8.1), and Intel MPI (version 2.0). Unlike Intel, AMD does not develop its own compiler, relying on products from companies like Portland Group, Pathscale, and even Intel for commercial compiler support. On our Opteron test system, we used the Portland Group Fortran and C compilers (version 6.1-6), the AMD Core Math Library (version 3.1), and the OpenMPI MPI implementation (version 1.1). We chose to use the Portland Group compiler on our Opteron test system because of its position as the preferred compiler on Opteron-based systems from Cray Inc.

Clearly, there are substantial differences in both hardware and software between the systems we considered. For example, AMD has traditionally downplayed differences in processor clock speed, a position recently adopted by Intel, so we struggle to isolate the performance differences in our benchmarking observations from both the perspective of raw performance and the relative to differences in test system hardware. Likewise, we felt it best to use the software suite (compilers and math libraries) that were likely to be used for scientific computing on each test system rather than arbitrarily choosing commercial compiler for all tests or using a free alternative like the GNU Compiler Collection. Whenever possible, we tried to use comparable optimization flags between compilers, but we cannot guarantee that we found the best platform- and compiler-specific optimization flags or that we spent exactly the same amount of time optimizing for each platform.

4 Computation Performance

To evaluate Woodcrest's floating point capabilities, we measured the performance of the Woodcrest system when executing three important floating point operations: double-precision matrix-matrix multiply (DGEMM), finding the LU factorization of a matrix, and forming the Fast Fourier Transform of a vector. We measured the performance of these operations as implemented by the Intel Math Kernel Library (MKL) and the AMD Core Math Library (ACML) on the Woodcrest and Opteron, respectively. We used OpenMP-enabled versions of both libraries.

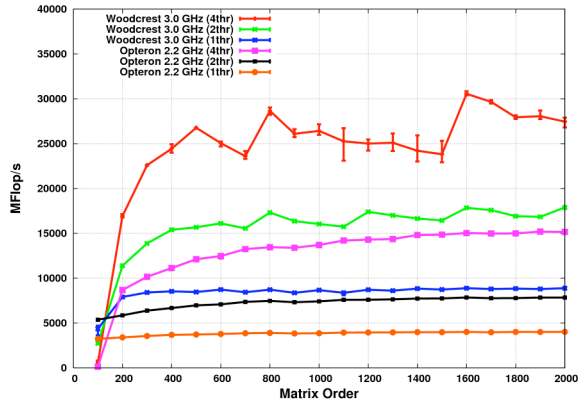

Figure 1: DGEMM performance

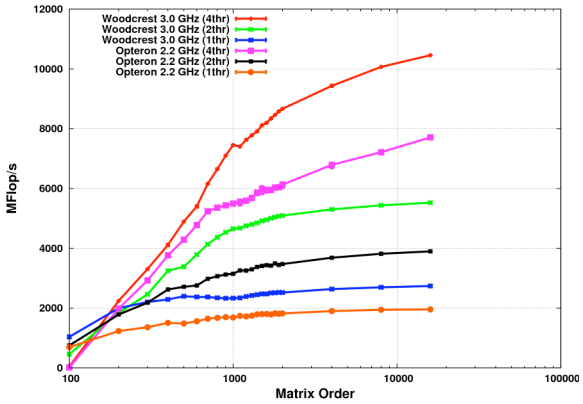

Figure 2: LU factorization performance

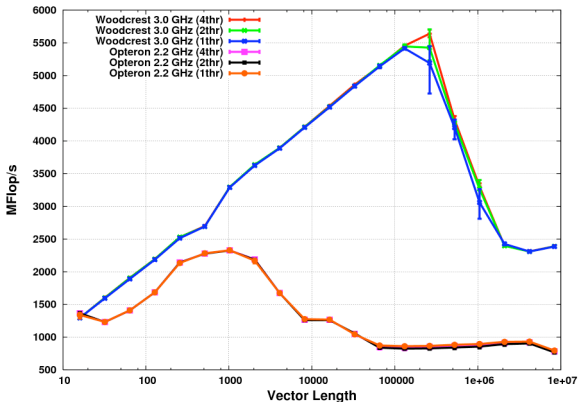

Figure 3: 1D complex FFT performance

4.1 Matrix Multiply

To evaluate DGEMM performance, we measured the elapsed time (i.e., wall clock time) required to complete a DGEMM operation on square matrices with order ranging from 100 to 2000 using one, two, and four OpenMP threads. We converted these elapsed times into throughput values and charted them in Figure 1. Each data point represents the average of three runs, with bars indicating the minimum and maximum observed value at each matrix order.

In absolute terms, the Woodcrest DGEMM performance is exceptional. On one core, the Woodcrest system achieved just over 8.9 GFlop/s; with four OpenMP threads on all four cores, we observed 30.8 GFlop/s. In comparison, the Opteron system achieved a maximum of just over 15 GFlop/s with four OpenMP threads. Clock speed differences between our test systems may account for some of this gap, but the fact that each Woodcrest core is theoretically capable of producing twice as many floating point results per clock tick as the Opteron is certainly another significant factor. However, based on the clock speeds of the two systems and the theoretical computational rate, we observed the Opteron to run at higher floating-point efficiency than the Woodcrest (86% versus 64% with four threads).

4.2 LU Factorization

Factorization of a matrix into lower and upper triangular matrices is another common operation, useful when solving linear systems. We measured the system throughput when executing the BLAS LU function from the Intel and AMD math libraries. Our measurements are charted in Figure 2. The performance difference between the Woodcrest and Opteron is not as pronounced as it was for DGEMM, but the Woodcrest does show a substantial performance advantage at each thread count.

4.3 Fast Fourier Transform (FFT)

Fast Fourier Transform (FFT) is the third computational micro-benchmark we considered in our Woodcrest evaluation. As with DGEMM and LU, we timed the implementation provided in the vendor's processor-specific math library, and converted them to floating point throughput values. Our 1D FFT measurements are charted in Figure 3.

The performance profiles for the FFT operation are dramatically different between the Woodcrest and the Opteron. The MKL FFT operation shows good scalability for vectors smaller than 10^5 elements, but its performance drops sharply thereafter. The performance of the ACML FFT function peaks with much smaller vectors, but degrades more gracefully than the MKL version on Woodcrest. Varying the number of OpenMP threads had negligible effect on both platforms except at the knee in the Woodcrest performance curve, where OpenMP threads allowed the FFT scalability to continue beyond the point where single-thread performance degraded.

5 Memory Hierarchy Performance

In contrast to the Opteron dual-core processors that do not share cache between cores, Woodcrest processors have a 4MB shared L2 cache. Also, Woodcrest systems use Fully Buffered DIMM (FB-DIMM) memory, whereas the Opteron uses DDR memory. These differences in memory hierarchy contribute to different memory access profiles for the two systems.

Figure 4, Figure 5, and Figure 6 show the bandwidth observed when accessing vectors of increasing length, as measured by the CacheBench memory benchmark [6]. Although CacheBench is not multi-threaded, measurements were taken when one, two, and four copies of the CacheBench program were running to try to capture the effect of memory contention. For Woodcrest, an error in the script that ran the experiment overwrote the two-thread results, hence Woodcrest results are shown only for one and four threads.

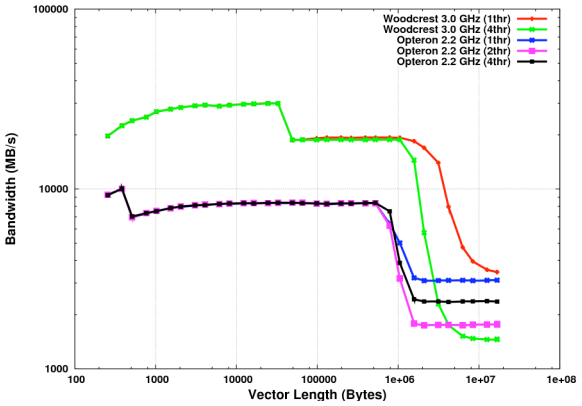

Figure 4: Cache read bandwidth

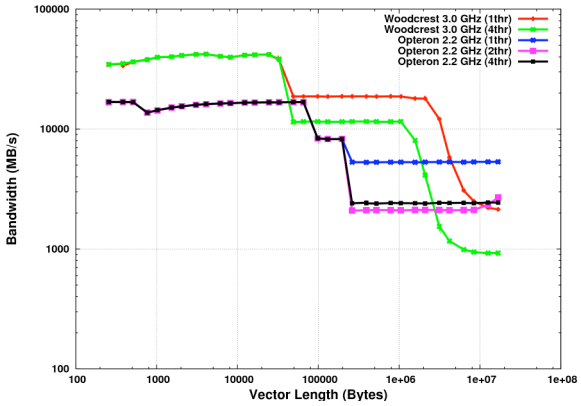

Figure 5: Cache write bandwidth

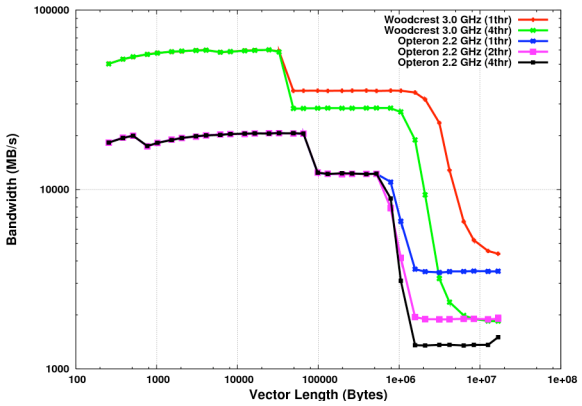

Figure 6: Cache read-modify-write bandwidth

Table 1: STREAM Triad bandwidth.
(All measurements expressed as MB/s.)

Threads	Woodcrest	Opteron
1	4217	4750
2	7355	7403
4	7494	9420

The observed memory bandwidth for the Woodcrest processors is superior compared to the Opteron test system in most cases, with the notable exception for very large vectors accessed from memory. In particular, accesses that hit in either level of the Woodcrest's cache provide superior bandwidth to the Opteron except for a small range of vector lengths with the CacheBench write operation. This superior performance could be due to several factors, including the higher clock rate of the Woodcrest and its FB-DIMM memory.

For both Opteron and Woodcrest, the effects of contention are apparent from the figures, but note that the Woodcrest shows the effects of contention in accesses to the L2 cache with writes, whereas the Opteron results show no sign of contention until vector lengths cause accesses to be satisfied from main memory. Note also the interesting inversion in the two- and four-thread case for the Opteron; we are unsure of the reason why bandwidth with four threads to main memory was observed to be better than bandwidth with two threads, and will require further experimentation to explain the effect.

The STREAM benchmark [5] provides another perspective on the memory performance of the Woodcrest processor. The STREAM micro-benchmark measures memory bandwidth when performing simple operations (copy, add, multiply, and a combination of the three called 'Triad') on long vectors. The vectors must be long enough that the benchmark measures bandwidth to main memory instead of to any level of data cache. Our observed STREAM Triad results for one, two, and four OpenMP threads on the Woodcrest are shown in Table 1. We suspect that slightly higher bandwidths may be possible with further optimization; we performed a minimal amount of tuning of vector lengths and used a collection of compiler flags that had proven to give good performance in our other experiments. For comparison, we also include STREAM Triad results from our 2.2 GHz Opteron system. The Opteron system memory bandwidth is quite high compared to that of our test Woodcrest system, perhaps due to the Opteron processors' on-chip memory controllers.

Our memory micro-benchmarks measure performance of memory accesses satisfied at all levels of the memory hierarchy. We emphasize that neither CacheBench nor STREAM gives a complete picture regarding memory hierarchy performance, especially since both use synthetic memory access patterns that are rarely encountered in full applications.

6 Application Performance

Our micro-benchmark results suggest the Woodcrest is well suited for systems designed for computational science. To further evaluate its suitability, we ran three full scientific applications on the Woodcrest system: the Large-scale Atomic/Molecular Massively Parallel Simulator (LAMMPS) [8], the Climate Atmosphere Model (CAM) [2], and the Parallel Ocean Program [4]. LAMMPS and POP use MPI for communication between parallel tasks, whereas CAM was built to use only OpenMP. Although the test systems were not distributed memory systems, we considered MPI-only applications because of the likelihood that MPI performance will remain important in the near future, due to portability concerns and development team inertia.

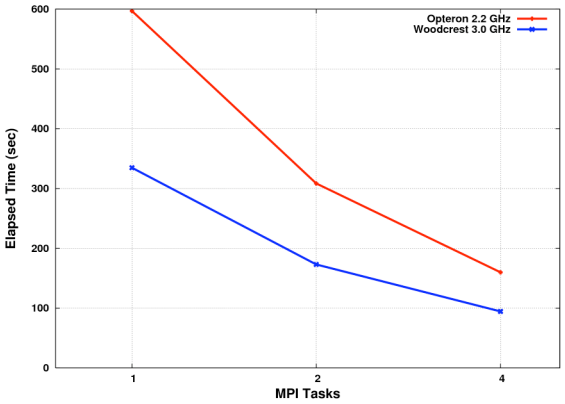

Figure 7: LAMMPS performance, Eam benchmark

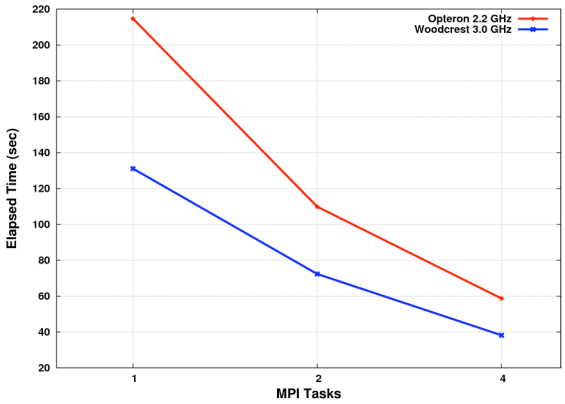

Figure 8: LAMMPS performance, Lj benchmark

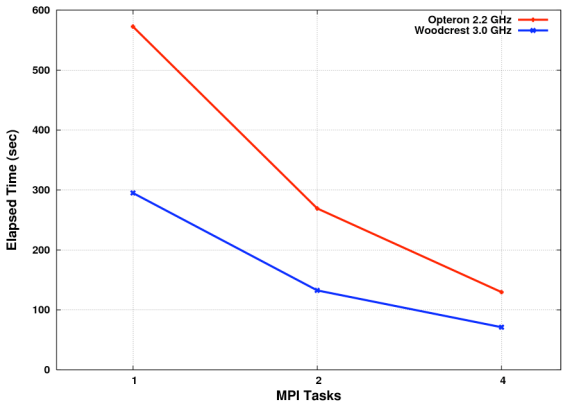

Figure 9: LAMMPS performance, Rhodo benchmark

6.1 LAMMPS

LAMMPS is a C++ program that uses MPI for communication. We used the "12Apr06" version of LAMMPS for our timings. The LAMMPS distribution includes several benchmark problems; we chose three for our timings. We ran each in weak scaling mode. We used the Intel MPI library on the Woodcrest system, and used the OpenMPI library, version 1.1, built using the PGI compilers. The timings from our LAMMPS experiments with one, two, and four MPI tasks are shown in Figure 7 (Eam benchmark), Figure 8 (Lj benchmark), and Figure 9 (Rhodo benchmark). Each charts plots the elapsed time required to compute 100 LAMMPS timesteps, so lower values are better than higher values. The Eam and Lj problems were run with 1,024,000 atoms, whereas the Rhodo problem was run with 128,000 atoms.

For all LAMMPS problems we considered, the Woodcrest showed better performance than the Opteron, although the performance gap narrowed as the number of MPI tasks was increased. Furthermore, the gap was large enough that differences in clock speed alone between the processors in the Woodcrest test system and our Opteron system does not account for it.

6.2 CAM

CAM consists mainly of Fortran code, and can be built to use MPI, OpenMP, or both for expressing parallelism. To ensure that we considered one example of a "pure" OpenMP application, we built CAM to use OpenMP only. We used CAM version 3.0p1 for our timings. After determining that CAM is extremely sensitive to the OpenMP per-thread stack size, we were able to run it on both the Woodcrest and Opteron test systems and measured the time required to complete ten simulation days of the T42 benchmark problem. The results of our CAM timings are shown in Figure 10. The trend shown in the chart is similar to that shown by the LAMMPS timings: the Woodcrest provided superior performance for the CAM code over the Opteron, but here it is not as clear that the performance difference goes beyond differences in processor clock speed.

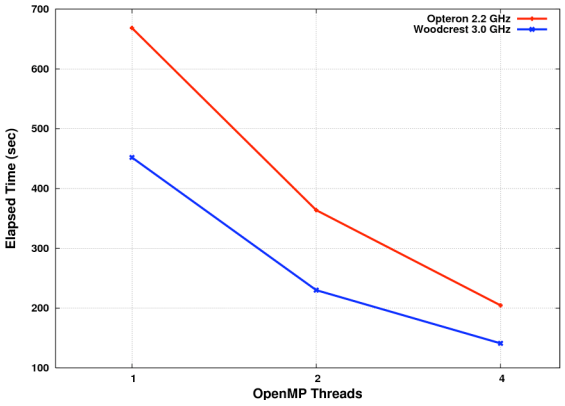

Figure 10: CAM performance for ten simulation days

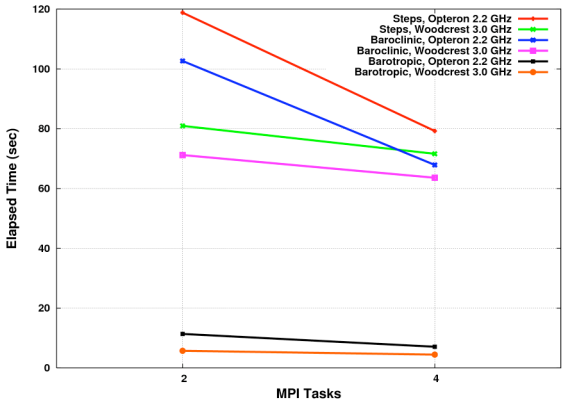

Figure 11: POP performance for ten simulation timesteps

6.3 POP

POP is a Fortran 90 code that, like CAM, can be built using MPI, OpenMP, or both. We only have experience using POP with MPI, we built and ran the MPI-only version of POP. We used POP version 1.4.3 for our timings. On the Woodcrest system, we used the Intel MPI library, and on the Opteron system we used OpenMPI 1.1. We tried to use the OpenMPI libraries built with the PGI compilers that we used for LAMMPS, but these libraries caused an error in the MPI routines dealing with communicator topologies used by POP. In the interest of time, we linked POP against an MPI installation built using the GCC compilers for these timings. We collected timings for runs of ten simulation timesteps of the x1 (i.e., one degree) problem. The timings are shown in Figure 11; each data point in the figure represents the average of three runs. Curves are shown for the total time spent in timesteps (labeled "Steps") and the two main phases of POP execution (labeled "Baroclinic" and "Barotropic").

As with our other test applications, the Woodcrest system showed a clear performance advantage over the Opteron-based system for our benchmark POP problem. In our initial experiments, we observed large variability in POP run times on the Woodcrest system with two processes. With help from Intel performance engineers, we determined that with the default MPI configuration on the system it was possible for both POP processes to contend for the same processor. When we used process affinity controls provided by the Intel MPI implementation, the variability in run times dropped to negligible levels. These updated results were taken from a slightly different Woodcrest system than our initial experiments (with newer processor version and different BIOS settings) but the observed run times on the newer system were approximately the same as the minimum run times from our initial experiments on the original test Woodcrest system.

7 Power and Thermal Performance

Power and heat have become important issues in contemporary microprocessor-based system design, and is one of the main factors driving microprocessor vendors toward multi-core processors. The Thermal Design Point (TDP) for each Woodcrest processor found in the Intel evaluation system is 80W and the TDP for each Opteron 275 processors 95W. Using only TDP values, Woodcrest exhibited a superior performance/power ratio than the Opteron system in our tests. However, this analysis of Woodcrest performance/power ratio is far from complete. A full analysis should consider system-wide power draw to capture the differences between power required for the Woodcrest system's FB-DIMM memory and the Opteron system's DDR memory. Because we accessed the Woodcrest system remotely, we had no capability to measure its system-wide power consumption. Also, a more complete performance/power analysis could also consider that Woodcrest parts with lower clock speeds have smaller power requirements, and that Opteron offers low-power versions of its Opteron processor line. Such processors might be more attractive candidates for use in HPC systems with a large number of processors and dense packaging.

8 Summary

We evaluated the dual-core Xeon 5160 processor, code-named Woodcrest, by considering the suitability of a Woodcrest-based system for scientific computing. With a limited time window for accessing the system, we

performed a preliminary evaluation using computational and memory micro-benchmarks and several full applications. In most cases, the Woodcrest showed excellent performance compared to an Opteron-based test system managed by our research group, even when taking differences in clock speed into account. DGEMM performance of the four-core test system was especially striking using the Intel Math Kernel Library optimized specifically for the Woodcrest processor. However, we caution that there were many hardware and software differences between the systems, and further comparison is needed to understand the reasons for the performance differences we observed. Also, the Woodcrest evaluation system was described by Intel representatives as an "Alpha level platform" and that the expected performance of shipping Woodcrest-based systems should exceed the performance of the evaluation system due to processor and chipset revisions in shipping systems. Nevertheless, our measurements suggest the Intel Woodcrest processor will be a formidable competitor to the Opteron as the commodity processor in high-end computing systems for scientific computing.

References

[1] Advanced Micro Devices, Inc., *Software Optimization Guide for AMD64 Processors*, http://www.amd.com/us-en/assets/content_type/white_papers_and_tech_docs/25112.PDF, 2005.

[2] W.D. Collins, P.J. Rasch *et al.*, "The Formulation and Atmospheric Simulation of the Community Atmosphere Model: CAM3," *Journal of Climate*, 2006.

[3] T.H. Dunigan, Jr., J.S. Vetter *et al.*, "Performance Evaluation of the Cray X1 Distributed Shared Memory Architecture," *IEEE Micro*, 25(1):30-40, 2005.

[4] P.W. Jones, P.H. Worley *et al.*, "Practical performance portability in the Parallel Ocean Program (POP)," *Concurrency and Computation: Experience and Practice*(in press), 2004.

[5] J.D. McCalpin, *Sustainable Memory Bandwidth in Current High Performance Computers*, http://home.austin.rr.com/mccalpin/papers/bandwidth/bandwidth.html, 1995.

[6] P.J. Mucci, K. London, and J. Thurman, "The CacheBench Report," University of Tennessee, Knoxville, TN 1998.

[7] S. Muchmore, "Inside the Intel Core Microarchitecture," in *Intel Developer Forum*. Cairo, Egypt, 2006, http://idfemea.intel.com/2006/cairo/download/DE03.pdf.

[8] S.J. Plimpton, "Fast Parallel Algorithms for Short-Range Molecular Dynamics," in *Journal of Computational Physics*, vol. 117, 1995

[9] J.S. Vetter, S.R. Alam *et al.*, "Early Evaluation of the Cray XT3," Proc. IEEE International Parallel & Distributed Processing Symposium (IPDPS), 2006.