DOME Microserver: Performance Evaluation of suitable processors

R. Clauberg and R.P. Luijten

THE DOME MICROSERVER PROJECT WANTS TO IMPLEMENT A HIGH DENSITY HIGH ENERGY EFFICIENCY MICROSERVER FOR THE PLANNED SQUARE KILOMETER ARRAY RADIO TELESCOPE. THE PRESENT ARTICLE DESCRIBES THE PERFORMANCE EVALUATION OF SUITABLE PROCESSORS FOR THIS PROJECT. SPECIAL EMPHASIS IS ON THE ABILITY TO PROCESS WORKLOADS RANGING FROM SIMPLE WEB-SERVER TYPE TO COMPLEX CLUSTER TYPES WITH HIGH ENERGY EFFICIENCY.

……The DOME project [1] is a collaboration between IBM and ASTRON, the Netherlands Institute for Radio Astronomy, to develop a computing solution to the exascale requirements for data processing of the international "Square Kilometer Array" radio telescope. One sub-project is the development of low cost low energy consumption microservers for processing the huge amount of data generated every day. To achieve these goals we are developing a high density microserver enabled by a water cooling system [2] and a highly modular system to cover different classes of workloads [2] [3]. The system is built of node cards and baseboards. Node cards cover compute nodes, accelerator nodes, storage nodes, and switching nodes. Baseboards offer different interconnect fabrics for different classes of workloads at different cost levels. Figure 1 shows a block diagram of the microserver system.

To achieve our low energy consumption goal we are looking specifically at system on a chip (SOC), multi-core processor solutions. SOCs reduce energy expensive inter-chip signaling and multi-core processors offer increased energy efficiency by running multiple cores at low frequency instead of one core at high frequency. However, the need to handle commercial applications requires processors with a 64-bit operating system.

Present candidate for the compute nodes of our microserver is the T4240 CPU from Freescale™[i]. This provides 12 cores with 2 threads per core. The T4240 SOC integrates all major parts of a motherboard except DRAM, NOR-boot flash, and power conversion logic.

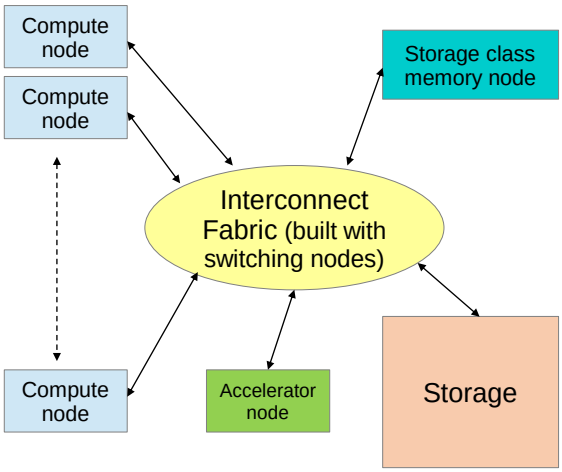


Figure 1: Block diagram of microserver system

Freescale products aim for the embedded system market and use custom operating systems. With the help of Freescale we were able to use the 64 bit Fedora® [4]operating system on the Freescale processors and to run major commercial workloads as e.g. Hadoop®, Apache™ webserver, and IBM® DB2®. We also run benchmark suites like SpecBench CPU2006 [5] which includes large sets of workloads from specific fields of business and high performance computing (HPC).

Performance benchmarking tools

For a performance evaluation we consider multiple choices of CPUs (cores per node, threads per core).

The benchmark tools used are:

- CoreMark [6] from EEMBC®, for performance of the processor cores
- CPU2006 [5] from SpecBench, for system performance
- Stream [7] , for memory access bandwidth analysis

CoreMark focusses on the performance of the core, has a total binary size of not more than 16KB using the GNU GCC compiler on an x86 machine and fits in the L1 cache of embedded processors. Hence, memory access bandwidth and other system aspects have no or negligible impact on the CoreMark value.

CPU2006 is a suite of programs from the business and HPC areas and measures overall system performance. The performance measures CINT and CFP are defined as the ratio of CPU-time against the CPU-time on a reference machine for integer and floating point programs, respectively. The total SPECint and SPECfp ratios are the geometric mean about the CINT and CFP ratios of all integer and floating point programs, respectively.

The Stream benchmark measures memory access bandwidth by implementing three arrays, each substantially larger than all the caches used by the processors and measuring the time it takes to access the array values and perform operations on these values. Four benchmark values are provided:

1. Copy : a(i) = b(i) - no floating point operations (FLOPs)
2. Scale : a(i) = q * b(i) - 1 FLOP per iteration
3. Sum : a(i) = b(i) + c(i) - 1 FLOP per iteration
4. Triad : a(i) = b(i) + q*c(i) - 2 FLOPs per iteration

Measured Performance

A single program can exploit multiple cores/threads efficiently only if it is parallelizable. However, multiple programs may be running concurrently on a processor, each on a different thread even if they are not parallelizable. This situation is shown in Figure 2. There have been multiple studies in the area of performance analyzes of various aspects of multi-core computer systems [8] [9]. Here we focus on the change of performance with the number of active threads exploited by the system. Accordingly, many comparisons use normalized data. For CPU2006 the values are estimates in the way that most of the simulations are based on a single run and not the 3 runs required by the standard. However, all optimizations parameters are equal for all programs of the same type and the same compiler.

Analyzed systems

- Freescale™ P5020 : 2 cores, single thread per core, 2.0 GHz, TDP (Thermal Design Power)=30W
- Freescale™ T4240 : 12 cores, 2 threads per core, 1.7 GHz, TDP=25W
- Intel® E3-1220L v3 : 2 cores, 2 threads per core, 1.1 GHz, TDP=13W
- Intel® E3-1230L v3 : 4 cores, 2 threads per core, 1.8 GHz, TDP=15W, turbo: 25W
- Intel® E3-1240L v3 : 4 cores, 2 threads per core, 2.0 to 3.0 GHz, TDP=15W, turbo: 25W
- Intel® E3-1265L v3 : 4 cores, 2 threads per core, 2.5 GHz, TDP=45W
- Intel® E3-1245 v3 : 4 cores, 2 threads per core, 3.4 GHz, TDP=84W

- IBM Power8® S824 : 4 nodes (=chips), 3 active cores per node, 8 threads per core, 8 4-channel memory controllers
- OpenPower™ evaluation board from Tyan® : 4 cores, 8 threads per core, single 4-channel memory controller, TDP=145W

all running a version of the Fedora operating system, except the Power8 systems which use the Ubuntu® operating system. We will show results only for our choice of T4240 for the microprocessor and some of the other systems.

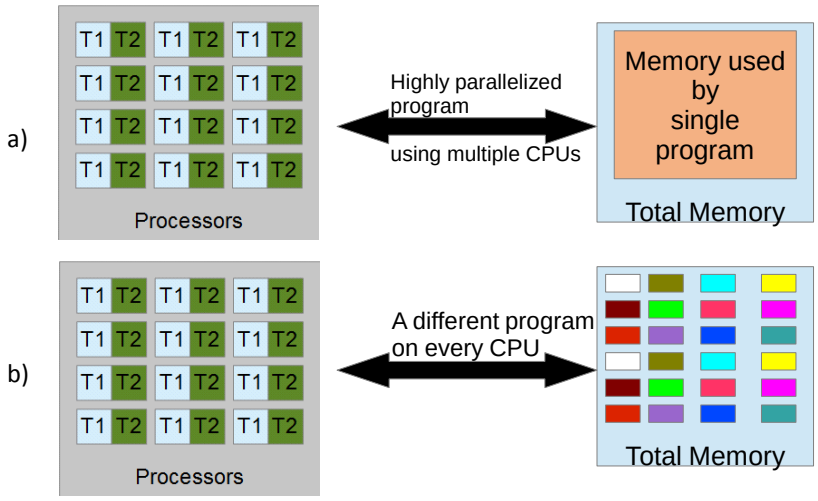


Figure 2: Exploiting multi-core processors

The analyses cover no network aspects since all the results are measured on regular test boards for the corresponding CPUs, not the final microprocessor node boards or chassis.

Stream benchmark

Stream uses OpenMP® [10] to exploit multiple processor threads. Figure 3 shows that memory access bandwidth increases steadily for the Freescale T4240 but jumps already close to the maximum value with the second thread for Intel E3-1230L v3 and the Tyan evaluation board. A more complicated picture evolves for the 4 chip Power S824 system. Here, memory access bandwidth increases with the number of active threads until this number equals the number of memory controllers, i.e. 8. The maximum value is at 11 threads close to 12 the number of active cores. The important difference between the processors is that the T4240 uses 3 independent

memory controllers, the E3-1230L v3 and the Power8 Tyan board use a single memory controller – dual channel for the E3 and quad channel for the Power8 Tyan board, and the Power S824 uses 8 quad channel memory controllers.

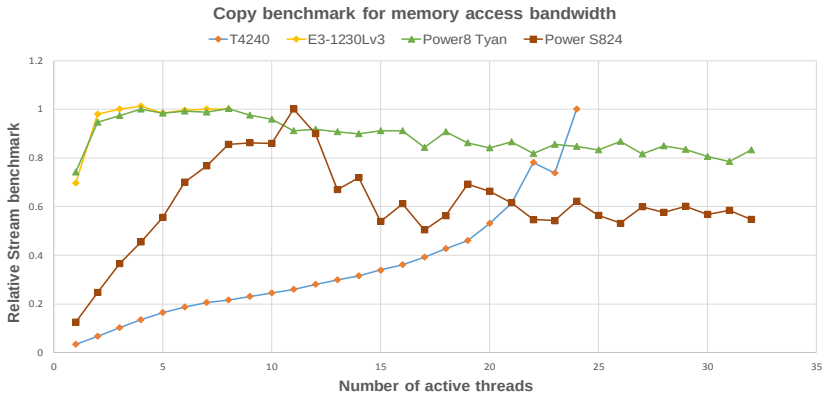


Figure 3: Stream copy benchmark

CoreMark

CoreMark runs multiple copies of the program in parallel as in Figure 2(b). The single core CoreMark value for the E3-1230L v3 is 12959 or about 3 times 4080, the value of the T4240. But the CoreMark value for the maximum number of cores in Table I is about 40% higher for the T4240.

Specbench CPU2006

From the CPU2006 benchmark only program Libquantum can exploit multiple threads as in Figure 2(a). Libquantum is a program-set from the CPU2006 SpecBench which simulates a quantum computer, running Shor's polynomial-time factorization algorithm. All programs of CPU2006 can mimic situation (b) of Figure 2 by concurrently running multiple copies of a program.

Figure 4 compares the the change of the CINT base ratio with the number of active threads for program Libquantum on the T4240 and the Power8 board from Tyan. For T4240 the value improves steadily with increasing degree of parallelization up to the maximum number of 24 threads. Parallelization was controlled with compiler option "-ftree-parallelize-loops=n" with n the number of active threads. For Tyan board we have 4 cores, 32 threads in total. While the maximum performance is reached with the maximum number of threads as for the T4240, we here see that already 89% of the maximum performance is reached when the number of threads equals the number of cores. Also, there is a drop to about 71% when the number of threads exceeds the number of cores.

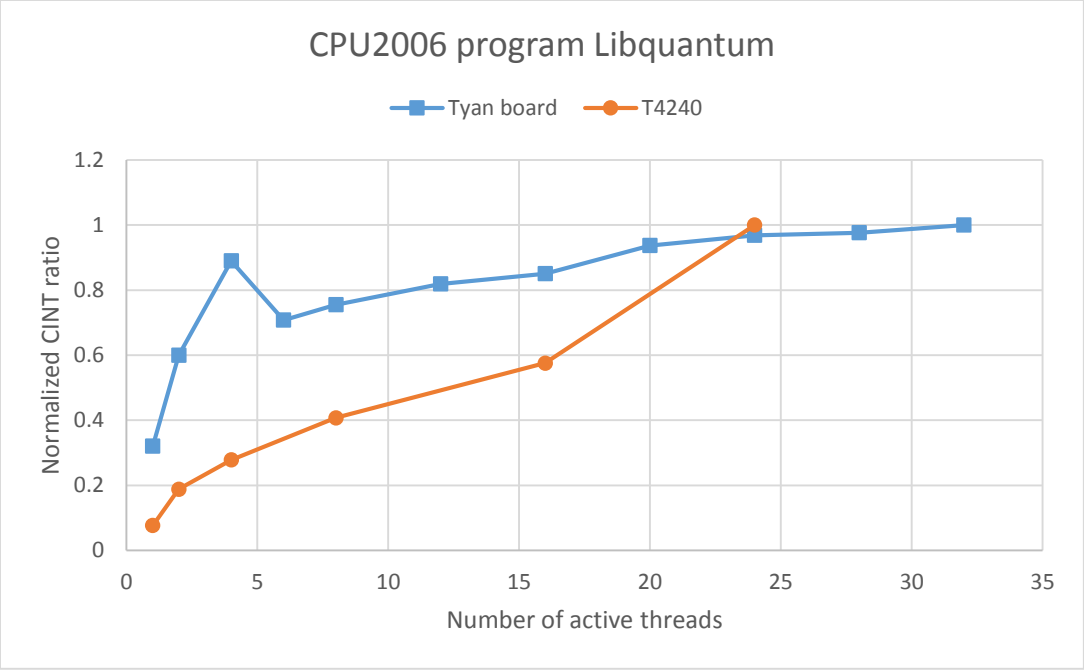


Figure 4: CPU2006 program Libquantum

$$\text{Norm. perf.} = \begin{cases} Copies/(F * Cores) \,;\, Copies \le Cores \\ \frac{Copies}{Threads} \,;\, Copies > Cores \end{cases} \qquad F=1.25$$

Figure 5: E3-1230L v3 with simple model

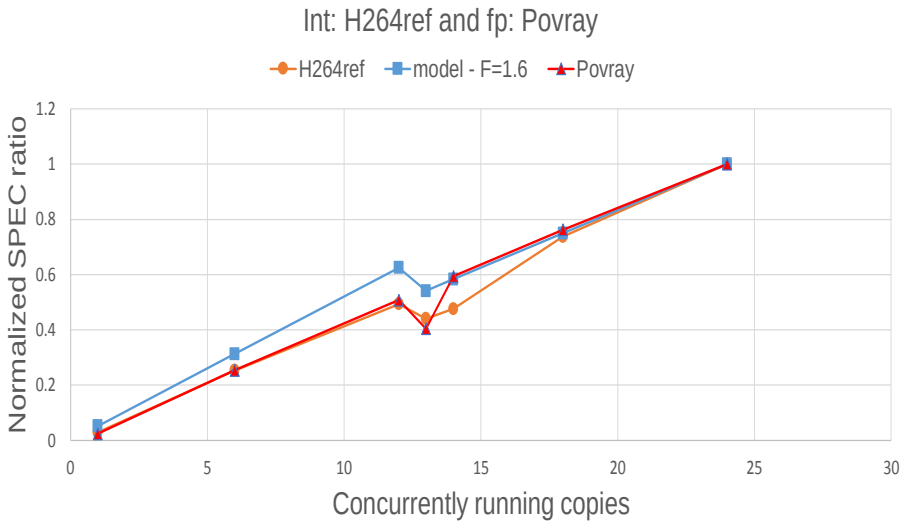
Figure 5 shows the entire SPECint base ratio as a function of the concurrently running copies of the programs on the E3-1230L v3. The behavior can be explained by a simple model in which every core takes the same time for the job and the 2nd thread per core multiplies core performance by F (F between 1 and 2). This model is based on the fact that CPU2006 measures performance P by measuring the time it takes to complete the simulation of all running copies of

the program. Simple model factor is F=1.25 for this plot, i.e. the 2nd threat gives a 25% improvement above the single thread.

Figure 6 shows the CINT base ratio for integer program H264ref and the CFP ratio for floating point program Povray as a function of the concurrently running copies of the programs on the T4240RDB.

Simple model factor is F=1.6 for this plot, i.e. the second thread gives a 60% improvement above a single thread. We are definitely not CPU limited in these cases. Taking the stream benchmark results into account, it is clear that for the CPU2006 SpecBench tests the T4240 is mainly limited by memory access bandwidth for the integer as well as the floating point program.

T4240RDB processor : 12 cores (3 clusters at 4 cores), 2 threads per core
binding = 0,1,2,3,8,9,10,11,16,17,18,19,4,5,6,7,12,13,14,15,20,21,22,23



$$\text{Norm. perf.} = \begin{cases} Copies/(F * Cores) \,;\, Copies \leq Cores \\ \frac{Copies}{Threads} \,;\, Copies > Cores \end{cases}$$

Figure 6: T4240 with simple model

Figure 7 shows the CPU2006 SPECint_base ratio as well as the CINT_base ratios for two selected CPU2006 programs as function of the number of concurrently running copies of the programs on the Power8 S824 system.

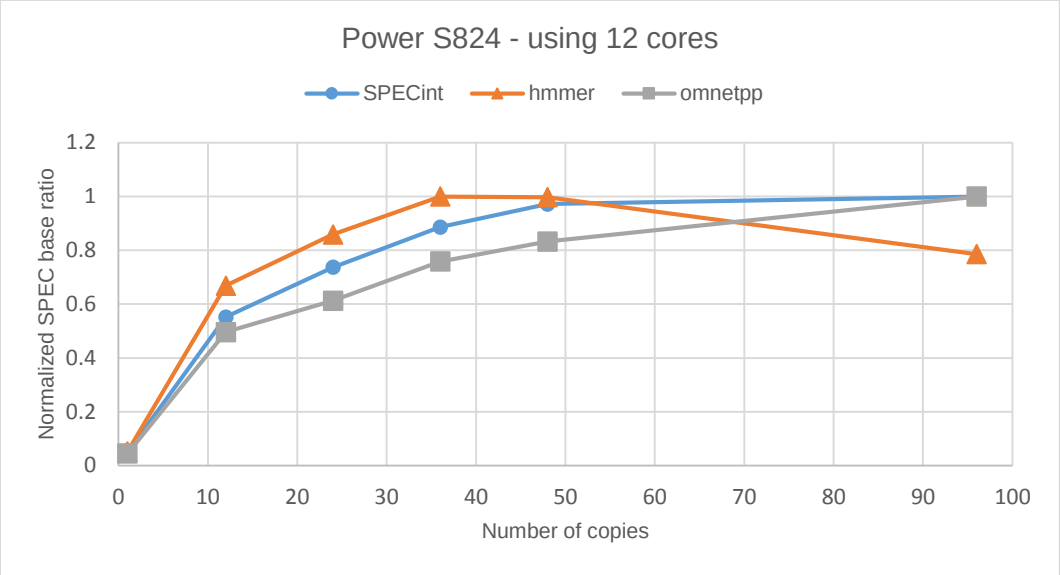


Figure 7: Normalized CPU2006 base ratios on Power S824 system

There is a clear difference to the T4240 behavior. While the T4240 benchmarks show a strong improvement still at large numbers of concurrently running program copies, the SPECint value for the Power8 system comes close to its maximum at about 48 copies. This number corresponds to about 4 copies per core since the S824 uses 12 cores. Figure 7 also shows the behavior of two selected examples of CPU2006 programs. One with the largest growth still at high numbers of copies and one with the earliest drop in performance with increasing number of copies. Comparing the results for T4240 against the E3 as well as the 2 Power8 systems, it is remarkable that for all benchmarks used here, the T4240 shows a steady increase in performance with the number of active threads up to the maximum value while the other systems usually show a strong increase in performance only up to the point where the number of active threads equals the number of cores. The E3 processors come close to maximum performance already when one thread is used on each core and the Power8 processors when about 4 threads are used per core.

Impact of different compilers and turbo-mode

Table I compares the Stream, CoreMark, and CPU2006 benchmarks for single thread and maximum number of threads of the most interesting processor versions analyzed by us. The benchmark results in parentheses in Table I are for specific compilers, the Intel Parallel Studio XE 2013 for the Intel processors and the IBM xlc Advanced Toolchain compiler on the Power systems. It is visible from the numbers that when using the GNU GCC compiler on all processors the single core performance of the E3 is substantially higher than for the T4240 but the situation turns around for SPECint and CoreMark if all threads of the T4240 can be exploited.

Table I: Comparison Table (GNU compiler and [Intel/IBM-compiler])

Processor	Stream [MB/s] (max)	CoreMark	CPU2006 (estimated base values)				
			CINT 1 thread	CINT max threads	CFP 1 thread	CFP max threads	CINT max per Watt
T4240	15'276	100'784	5.53	93.13	3.99	48.0	3.7
E3-1230L v3	21'144	63'597 [70'419]	20.07 [29.75]	77.57 [114.21]	20.44	77.31	7.6
E3-1240L v3 2.00 GHz fixed	21'246	[79'267]	22.13 [29.95]	83.56 [125.05]	22.56	81.96	8.3
E3-1240L v3 Turbo-mode up to 3 GHz	21'539 [21'571]	100'157 [108'735]	[41.75]	[151.55]	[44.76]	[128.98]	6.1
Power8 Tyan board (4.3 GHz)	17'707	62'167	20.9	93.8	19.3	95.0	0.7
Power S824 (copies distr. Onto 12 cores, 4.2 GHz)	167'208	377'470	25.6 [33.1]	536.1 [643.2]	27.1	527.2	1.7(*)

The performance per Watt values are based on the SPECint ratios with maximum number of threads divided by the power estimated as TDP (without Turbo mode if not used). The Power S824 with 370 W (250 W for P8 in turbo mode plus 8 memory buffer chips at 15W each). The measured Power S824 system uses 4 6-core P8 processors chips at 130 W each, plus 16 memory buffer chips, but only 3 cores per chip are active. The 1.7 value is an estimate for a single 12-core P8 chip which can have a maximum of 8 memory buffer chips.

However, the Intel processors show a substantially better SPECint base ratio when using Intel's compiler instead of the open source GNU compiler. Values for specific programs deviate strongly with a 1292% better CINT ratio for 8 copies of Libquantum and nearly no improvement for 8 copies of H264ref. For the simple Copy function of the Stream benchmark the effect is negligible, but for the Scale function, the Intel compiler achieves a 39% (single thread) to 63% (8 threads) better performance. The same improvement is achieved with the two other stream benchmarks (Sum and Triad). For CoreMark we see an improvement between 11% at 8 threads and 41% at a single thread when using the Intel compiler instead of the GNU compiler.

Comparing values for E3-1240L v3 with and without turbo-mode given in Table I shows that turbo-mode increases the SPECint_base value by about 39% for a single copy and by about 21% when 8 copies are running concurrently. From measurements with the Linux kernel program turbostat we know that power consumption for the E3-1240Lv3 chip package increases from 15W to 25W with turbo-mode while the CINT value for maximum number of threads increases from 125.1 to 151.6. Hence the energy efficiency measured as CINT per Watt drops from 8.3 to 6.1 or 26.5% when enabling the turbo-mode.

Evaluation summary

The present study shows the strong dependence of overall performance of the considered processors for our microserver on the number of exploitable processor cores and threads, the arrangement of memory controllers, and the choice of compilers. While the performance of the 12 cores, 24 threads Freescale T4240 processor is substantially below that of the 4 cores, 8 threads Intel E3-1230L v3 processor for single thread operation, the situation turns around for CoreMark and CPU2006 if the maximum number of threads can be exploited. For Stream the memory access bandwidth of T4240 still is below the E3 values. In general, the second thread per core on the T4240 nearly improves the performance of the T4240 by the same amount as an additional core. Considering performance per Watt we see from Table I that the Intel processors show the best values, followed by the T4240. The Power S824 system clearly is the performance leader, but nevertheless trails the Intel E3 as well as the T4240 processors in performance per Watt. This comparison is of course for the chip packages. It is important to remember that the T4240 is a SOC which contains a large number of I/O ports missing in the E3 and P8 processor packages and the corresponding energy consumption should be added to these systems for a realistic comparison with the T4240 SOC.

A second result is that it is important to consider the entire system under test going beyond the hardware capabilities. We observed problems with object code only applications on Fedora operating system due to the small time periods a specific version of Fedora is supported. Also, the available compilers for a specific processor showed a significant impact on the achievable performance.

Acknowledgments

We would like to acknowledge the excellent collaboration with all our colleagues working on the DOME microserver program, especially A. Doering, M. Cossale, and S. Paredes, as well as the support we received from Freescale.

References

[1] P. C. Broekema, A. J. Boonstra, V. Caparrós Cabezas, T. Engbersen, H. Holties, J. Jelitto, R. P. Luijten and et al., "DOME: towards the ASTRON & IBM center for exascale technology," in *Proc. ACM Workshop on High-Performance Computing for Astronomy (Astro-HPC'12), pp. 1-4*, 2012.

[2] R. P. Luijten, A. Doering and S. Paredes, "Dual function heat-spreading and performance of the IBM-Astron DOME 64-bit MicroServer demonstrator," in *IEEE International Conference on IC Design & Technology (ICICDT'2014)*, Austin, Tx, U.S.A., 2014.

[3] R. Luijten, D. Pham, R. Clauberg, M. Cossale, H. N. Nguyen and M. Pandya, "Energy Efficient MicroServer based on a 12-core 1.8GHz 188K Coremark 28nm Bulk CMOS 64-bit SoC for Big-Bata Applications with 159GB/s/liter Memory Bandwidth System Density," in *Proc. Internat. Solid-State Circuits Conf., pp. 76-78*, San Francisco, Ca, U.S.A., 2015.

[4] "Fedora Project," [Online]. Available: http://fedoraproject.org/.

[5] "CPU2006," [Online]. Available: http://www.spec.org/cpu2006/.

[6] "CoreMark," [Online]. Available: http://www.eembc.org/coremark/.

[7] J. D. McCalpin, "STREAM: Sustainable Memory Bandwidth in High Performance Computers," [Online]. Available: http://www.cs.virginia.edu/stream/.

[8] G. Blake, R. G. Dreslinski and T. Mudge, "A survey of multicore processors," *IEEE Signal Processing Magazine,* vol. 26, no. 6, pp. 26-37, 2009.

[9] S. Catalan, J. Gonzalez Dominguez, R. Mayo and E. S. Quintana Orti, "Analyzing the Energy Efficiency of the Memory Subsystem in Multicore Processors," in *IEEE International Symposium on Parallel and Distributed Processing with Applications (ISPA), pp. 10-14*, 2014.

[10] "The OpenMP API Specification for Parallel Programming," [Online]. Available: http://openmp.org/wp/.

[i] Freescale is a trademark of Freescale Inc, IBM, DB2, Power8 are trademarks of International Business Machines Corporation, Intel and XEON are trademarks of Intel Corporation. Trademarks of several standard organizations are used.