Impact of Quad-Core Cray XT4 System and Software Stack on Scientific Computation

S.R. Alam, R.F. Barrett, H. Jagode, J.A. Kuehn, S.W. Poole, and R. Sankaran

Oak Ridge National Laboratory, Oak Ridge, TN 37831, USA
{alamsr,barrett,jagode,kuehn,poole,sankaran}@ornl.gov

Abstract. An upgrade from dual-core to quad-core AMD processor on the Cray XT system at the Oak Ridge National Laboratory (ORNL) Leadership Computing Facility (LCF) has resulted in significant changes in the hardware and software stack, including a deeper memory hierarchy, SIMD instructions and a multi-core aware MPI library. In this paper, we evaluate impact of a subset of these key changes on large-scale scientific applications. We will provide insights into application tuning and optimization process and report on how different strategies yield varying rates of successes and failures across different application domains. For instance, we demonstrate that the vectorization instructions (SSE) provide a performance boost of as much as 50% on fusion and combustion applications. Moreover, we reveal how the resource contentions could limit the achievable performance and provide insights into how application could exploit Petascale XT5 system's hierarchical parallelism.

1 Introduction

Scientific productivity on the emerging Petascale systems is widely attributed to the system balance in terms of processor, memory, network capabilities and the software stack. The next generations of these Petascale systems are likely to be composed of processing elements (PE) or nodes with 8 or more cores on single or multiple sockets, deeper memory hierarchies and a complex interconnection network infrastructure. Hence, the development of scalable applications on these systems cannot be achieved by a uniformly balanced system; it requires application developers to develop a hierarchical view where memory and network performance follow a regular but non-uniform access model. Even the current generation of systems with peak performance of hundreds of Teraflops such as the Cray XT and IBM Blue Gene series systems offer 4 cores or execution units per PE, multiple levels of unified and shared caches and a regular communication topology along with support for distributed computing (message-passing MPI) and hybrid (MPI and shared-memory OpenMP or pthreads) programming models [Dagnum98, Snir98, BGL05, BGP08, XT3a-b, XT4a-b]. As a result, it has become extremely challenging to sustain let alone to improve performance efficiencies or scientific productivity on the existing systems as we demonstrate in this paper. At the same time however, these systems serve as test-beds for applications targeting Petascale generation systems that are composed of hundreds of thousands of processing cores.

H. Sips, D. Epema, and H.-X. Lin (Eds.): Euro-Par 2009, LNCS 5704, pp. 334–344, 2009.

We have extensive experience of benchmarking and improving performance efficiencies of scientific applications on the Cray XT series systems, beginning from the first-generation, single-core AMD based ~26 Teraflops Cray XT3 system to the latest quad-core based ~263 Teraflops Cray XT4 system. During these upgrades, a number of system software and hardware features were modified and replaced altogether such as migration from Catamount to Compute Node Linux (CNL) operating system, network capabilities and support for hybrid programming models and most importantly multi-core processing nodes [Kelly05]. A complete discussion of individual features are beyond the scope of this paper, however we do attempt to provide a comprehensive overview of the features updated in the latest quad-core upgrade and how these features impact performance of high-end applications. We provide an insight by using a combination of micro-benchmarks that highlight specific features and then provide an assessment of how these features influence overall performance of complex, production-level applications and how performance efficiencies are improved on the target platform.

In this paper, we focus on micro-architectural characteristics of the quad-core system particularly the new vectorization units and the shared level 3 cache. This study also enables us to identify features that are likely to influence scientific productivity on the Petascale Cray XT5 system. Hence, a unique contribution of this paper is that it not only evaluates performance of a range of scientific applications on one of the most powerful open-science supercomputing platform but also discusses how the performance issues are addressed during the quad-core upgrade. The Cray XT5 system shares a number of features including the processor and the network infrastructure with its predecessor, the quad-core XT4 system. However, the XT5 system has some distinct characteristics; most importantly, hierarchical parallelism within a processing node since an XT5 PE is composed of two quad-core processors thereby yielding additional resource contentions for memory, network and file I/O operations.

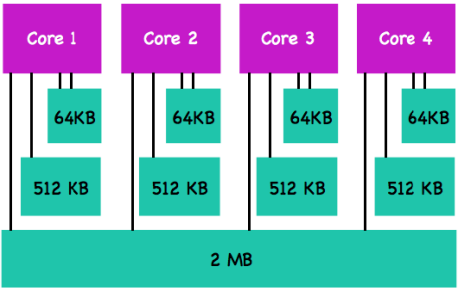
The outline of the paper is as follows: background and motivation for this research along with a description of target system hardware and software features is provided in section 2. In section 3, we briefly outline micro-benchmarks and high-end scientific applications that are targeted for this study. Details of experiments and results relating to each feature that we focus on in this paper are presented in section 4. Conclusions and future plans are outlined in section 5.

2 Motivation and Background

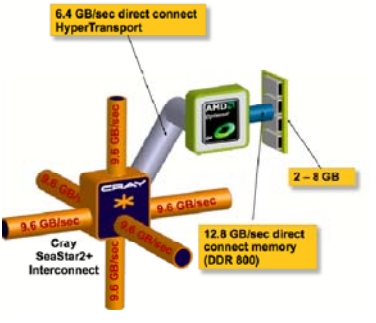
The Cray XT system located at ORNL is the second most powerful computing capability for the Department of Energy's (DOE) Office of Science, and in fact represents one of the largest open science capability platforms in the United States. Named Jaguar, it is the primary leadership computer for the DOE Innovative and Novel Computational Impact on Theory and Experiment (INCITE) program, which supports computationally intensive, large-scale research projects. The 2008 program awarded over 140 million processor hours on Jaguar to groups investigating a broad set of science questions, including global climate dynamics, fusion, fission, and combustion energy, biology, astrophysics, and materials.

In order to support this scale of computing, Jaguar has been upgraded from a 119 Teraflops capability to 262 Teraflops (TFLOPS). Several fundamental characteristics of the architecture have changed with this upgrade, which have a wide-ranging impact across different application domains. This motivates our research of identifying and quantifying the impact of these new architectural and system software stack features on leadership scale applications.

The current incarnation of Jaguar is based on an evolutionary improvement beginning with the XT3, Cray's third-generation massively parallel processing system, building on the T3D and T3E systems. Based on commodity AMD Opteron processors, most recently for instance the quad-core Barcelona system, a Cray custom interconnect, and a light-weight kernel (LWK) operating system, the XT3 was delivered in 2005. Each node consisted of an AMD Opteron model 150 (single core) processor, running at 2.4 GHz with 2 GBytes of DDR-400 memory. The nodes were connected by a SeaStar router through HyperTransport, in a 3-dimensional torus topology, and running the Catamount operating system. With 5,212 compute nodes, the peak performance of the XT3 was just over 25 TFLOPS [XT3a-c].



(a) Block diagram of Barcelona processor



(b) Cray XT4 node architecture [Courtesy of Cray Inc.]

Fig. 1. Internals of a Quad-core based Cray XT4 node

Jaguar processors were upgraded to dual-core Opteron model 100 2.6 GHz processors in 2006, with memory per node doubled in order to maintain 2 GBytes per core. It was again upgraded April, 2007, with three major improvements: 6,296 nodes were added; memory on the new nodes was upgraded to DDR2-667, increasing memory bandwidth from 6.4 GBytes per second (GB/s) to 10.6 GB/s; and the SeaStar2 network chip connected the new nodes, increasing network injection bandwidth (of those nodes) from 2.2 GB/s to 4GB/s and increasing the sustained network performance from 4GB/s to 6GB/s. Thus with 23,016 processor cores, this so-called XT3/XT4 hybrid provided a peak performance of 119 TFLOPS [XT4a-b].

In spring 2008, Jaguar was again upgraded: 7,832 quad-core processors replace the 11,508 dual-core (illustrated in Figure 1, the interconnect is now fully SeaStar2, and the LWK is a customized version of Linux named Compute-Node Linux (CNL). Each compute node now contains a 2.1 GHz quad-core AMD Opteron processor and 8

GBytes of memory (maintaining the per core memory at 2 GBytes). As before, nodes are connected in a 3-dimensional torus topology, now with full SeaStar2 router through HyperTransport (see Figure 1(b)). This configuration provides 262 TFLOPS with 60 TBytes of memory.

3 Micro-benchmark and Application Details

3.1 HPCC Benchmark Suite

We used High Performance Computing Challenge (HPCC) benchmark suite to confirm micro-architectural characteristics of the system. HPCC benchmark suite [HPCCa-b] is composed of benchmarks measuring network performance, node-local performance, and global performance. Network performance is characterized by measuring the network latency and bandwidth for three communication patterns. The node local and global performance are characterized by considering four algorithm sets, which represent four combinations of minimal and maximal spatial and temporal locality: DGEMM/HPL for high temporal and spatial locality, FFT for high temporal and low spatial locality, Stream/Transpose (PTRANS) for low temporal and high spatial locality, and RandomAccess (RA) for low temporal and spatial locality. The performance of these four algorithm sets are measured in single/serial process mode (SP) in which only one processor is used, embarrassingly parallel mode (EP) in which all of the processors repeat the same computation in parallel without communicating, and global mode in which each processor provides a unique contribution to the overall computation requiring communication.

3.2 Application Case Studies

The application test cases are drawn from the workload configurations that are expected to scale to large number of cores and that are representative of Petascale problem configurations. These codes are large with complex performance characteristics and numerous production configurations that cannot be captured or characterized adequately in the current study. The intent is rather to provide a qualitative view of system performance using these test cases to highlight how the quad-core system upgrade has influenced the performance as compared to the preceding system configurations.

3.2.1 Fusion Application (AORSA)

The two- and three-dimensional All-ORders Spectral Algorithm (AORSA [AORSA08]) code is a full-wave model for radio frequency heating of plasmas in fusion energy devices such as the International Thermonuclear Experimental Reactor5 (ITER) and the National Spherical Torus Experiment (NSTX) [Jaerger06-07]. AORSA operates on a spatial mesh, with the resulting set of linear equations solved for the Fourier coefficients. A Fast Fourier Transform algorithm converts the problem to a frequency space, resulting in a dense, complex- valued linear system. Parallelism is centered on the solution of the dense linear system, currently accomplished using a locally modified version of HPL [Dongarra90, Longau07]. Quasi-linear diffusion coefficients are then computed, which serve as an input to a separate application (Fokker-Plank solver) which models the longer term behavior of the plasma.

3.2.2 Turbulent Combustion Code (S3D)

Direct numerical simulation (DNS) of turbulent combustion provides fundamental insight into the coupling between fluid dynamics, chemistry, and molecular transport in reacting flows. S3D is a massively parallel DNS solver developed at Sandia National Laboratories. S3D solves the full compressible Navier-Stokes, total energy, species, and mass continuity equations coupled with detailed chemistry. It is based on a high-order accurate, non-dissipative numerical scheme and has been used extensively to investigate fundamental turbulent chemistry interactions in combustion problems including auto-ignition [Chen06], premixed flames [Sankaran07], and non-premixed flames [Hawkes07].

The governing equations are solved on a conventional three-dimensional structured Cartesian mesh. The code is parallelized using a three-dimensional domain decomposition and MPI communication. Spatial differentiation is achieved through eighth-order finite differences along with tenth-order filters to damp any spurious oscillations in the solution. The differentiation and filtering require nine and eleven point centered stencils, respectively. Ghost zones are constructed at the task boundaries by non-blocking MPI communication among nearest neighbors in the three-dimensional decomposition. Time advance is achieved through a six-stage, fourth-order explicit Runge-Kutta (R-K) method [Kennedy00].

4 Quantitative Evaluation and Analysis of Selected Features

4.1 Vector (SSE) Instructions

The Cray XT4 system is upgraded from dual-core Opteron to a single-chip, native quad-core processor called Barcelona. One of the main features of the quad-core system was quadrupling the floating point performance using a wider, 32-byte instruction fetch, and the floating-point units can execute 128-bit SSE operations in a single clock cycle (including the Supplemental SSE3 instructions Intel included in its Core-based Xeons). In addition, the Barcelona core has relatively higher bandwidth in order to accommodate higher throughput—internally between units on the chip, between the L1 and L2 caches, and between the L2 cache and the north bridge/memory controller.

In order to measure the impact of the new execution units with 128-bit vectorization support, we ran two HPCC benchmarks that represent scientific computation: DGEMM and FFT, in single processor (SP) where a single instance of an application runs on a single core and embarrassingly parallel (EP) where all cores execute an application without communicating with each other. We also measure performance per socket to estimate overall processor efficiencies. The quad-core XT4 has 4 cores per socket while the dual-core XT4 has two cores per socket. The results are shown in Figure 2. We observe a significant increase in per core performance for the dense-matrix computation benchmark (DGEMM), which is able to exploit the vector units. The FFT benchmarks on the other hand showed a modest increase in performance. Results in the EP mode when all four cores execute the same program revealed the impact of the shared L3 cache as the FFT performance slows down at a much higher rate for the quad-core system as compared to the dual-core and single-core XT platforms. The L3 behavior is detailed in the next section.

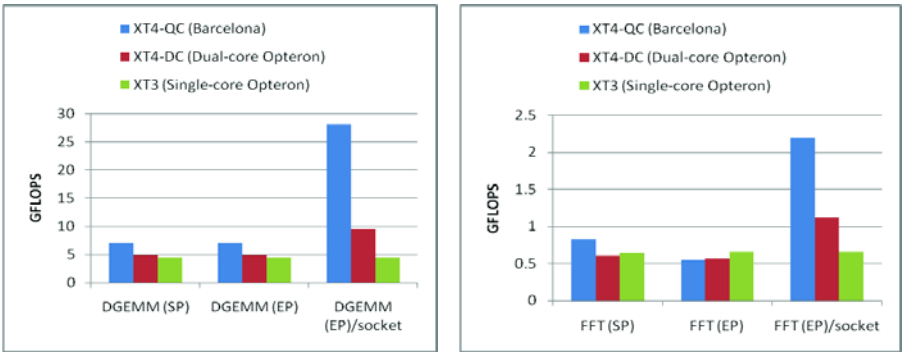
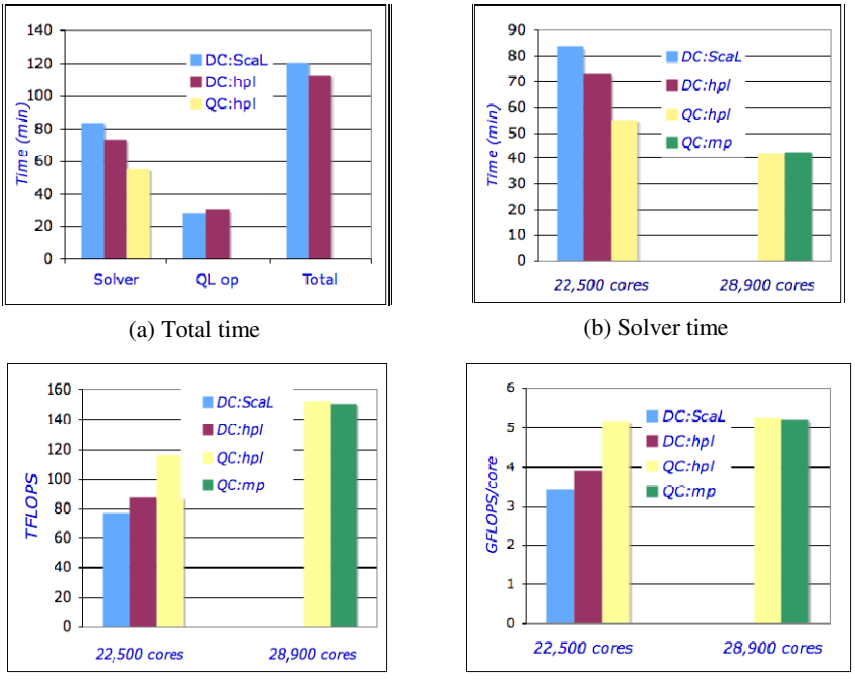


Fig. 2. HPCC DGEMM and FFT performance on different generations of Cray XT systems

Although there is a slowdown for FFT in the EP mode, we observe that per socket performance of the quad-core processor is significantly higher than that of the dual-core processor. We conclude that the significant performance boost per core brings in additional requirements for code development and generation for the quad-core processors. In other words, a misaligned and non-vector instruction could result in a code achieving less than a quarter of total achievable performance. Our two target applications highlighted the need for optimizing these vector operations.

AORSA, a fusion application, has a distinguished history running on the XT-series, allowing researchers to conduct experiments at resolutions previously unattainable [XT4a-c] executing at unprecedented computational scales. For example, the first simulations of mode conversion in ITER were run on the single-core XT3 [Jaeger06] on a 350 x 350 grid. On the dual-core XT3/XT4, this feat was again achieved, at increased resolution (500 x 500 grid), with the linear solver achieving 87.5 TFLOPS (74.8% of peak) on 22,500 cores [Jaeger07]. This same problem run on the quad-core XT increased this performance to 116.5 TFLOPS, and when run on 28,900 cores performance increased to 152.3 TFLOPS. Performance results for this scale are shown in Figure 3. Results are shown for the dual-core (DC) and quad-core (QC) processors with ScaLAPACK (Scal) and HPL (hpl) based solver. Moreover, experimental mixed-precision (mp) results are also shown in the figure. While impressive, relative to the theoretical peak performance has decreased from 74.8% to 61.6%. Although this is not unexpected due to the decreased clock speed and other issues associated with the increased number of cores per processor, we are pursuing further improvements. However, the time-to-solution (the relevant metric of interest) dropped from 73.2 minutes to 55.0 minutes, a decrease of 33%.

We expect performance of the solver phase to increase based on planned improvements to the BLAS library and the MPI implementation. In addition, we are experimenting with a mixed-precision approach [Langou07]. This capability is currently included in the Cray math library (-libsci) as part of the Iterative Refinement Toolkit (IRT). While this technique shows promise, it is not providing an improvement at the relevant problem scales. Although the condition of the matrix increases with resolution, this does not appear to be an issue. More likely is the use of the ScaLAPACK factorization routine within IRT compared with the HPL version: at 22,500 cores on the dual-core Jaguar, ScaLAPACK achieved 48 TFLOPS, whereas HPL achieved 87 TFLOPS.



(a) Total time

(b) Solver time

(c) Solver TFLOPS

(d) Solver GFLOPS per core

Fig. 3. AORSA performance on XT4 systems

The turbulent combustion application, S3D, is parallelized using a three-dimensional domain decomposition and MPI communication. Each MPI process is responsible for a piece of the three-dimensional domain. All MPI processes have the same number of grid points and the same computational load. Inter-processor communication is only between nearest neighbors in a logical three-dimensional topology. A ghost-zone is constructed at the processor boundaries by non-blocking MPI sends and receives among the nearest neighbors in the three-dimensional processor topology. Global communications are only required for monitoring and synchronization ahead of I/O. A comparison of dual-core and quad-core performance is shown in Table 1.

The initial port (Table 1) showed a decrease in performance, though less than that attributable to only the decrease in clock speed. This suggests that vectorization is occurring, though not as aggressively as desired. Special effort was applied to the computation of reaction rates, which consumed approximately 60% of overall runtime. Table 2 shows the effects of the compiler when able to vectorize code. Although for each category the number of operations increases, the proportion of operations occurring in vector mode increased by 233%, resulting in a decrease in runtime of this computation by over 20%.

Table 1. S3D single processor performance (weak scaling mode). The amount of work is constant for each process. "MPI mode" refers to the number of MPI processes and how they are assigned to each node: -n is the total number of processes, -N is the number of processes assigned to each quad-core processor node. Time is wall clock in units of seconds; "cost" is defined as micro-sec per grid point per time step. The "vec" columns show the performance after the code was reorganized for stronger vectorization.

		Dual-core		Quad-core			
Problem Size	MPI mode	Time	Cost	Time		Cost	
					vec		vec
30 × 30 × 30	–n 1 –N 1	404	150	415	333	154	123
60 × 30 × 30	–n 2 –N 2	465	172	430	349	159	129
60 × 60 × 30	–n 4 –N 4	n/a	n/a	503	422	186	156

Table 2. S3D reaction rate computation counters. (Values from dual-core Jaguar).

Counters	Before	After
	x 10^9 operations	
Add	182	187
Multiply	204	210
Add + Mult	386	397
Load/Store	179	202
SSE	91	212

4.2 Deeper Memory Hierarchy (L3 Cache)

Another distinctive feature of the quad-core processor is the availability of an L3 cache that is shared among all four cores. There was no L3 cache in the predecessor Opteron processors. L3 serves as a victim cache for L2. L2 caches (not shared) are filled with victims from the L1 cache (not shared) i.e. after the L1 fills up rather than sending data to memory it sits in L2 for reuse. Hence, data-intensive applications could benefit from this L3 cache only if the working set is within cache range.

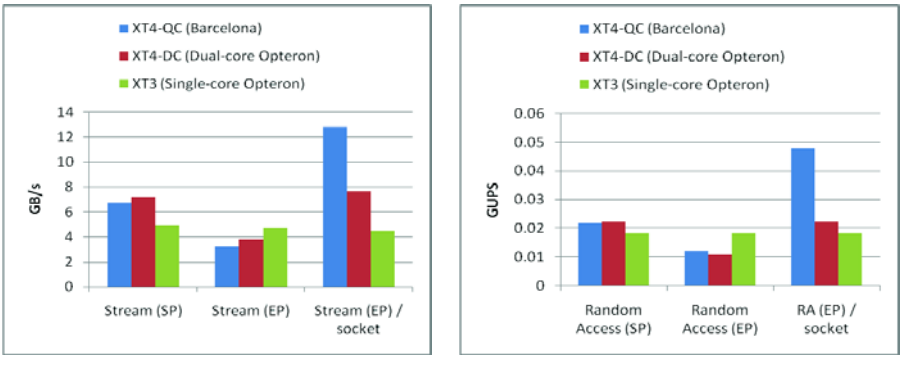


Fig. 4. HPCC memory performance benchmark results

Two HPCC memory performance benchmarks, stream and random access, were targeted to quantitatively evaluate performance of the memory sub-system. We compared the quad-core performance with the dual-core (XT4-DC) and single-core (XT3) AMD processors that preceded the latest quad-core (XT4-QC) processing nodes as shown in Figure 4. Both memory benchmarks highlight the effect of using a single core (SP mode) as compared to using all four cores simultaneously (EP mode) both for regular, single-strided (stream) access and random access benchmarks. Random memory access benchmarks highlight this cache behavior. We note that the shared resources in memory sub-system do account for slowdown in the EP mode, however this slowdown is less than by a factor of 4. In fact on the quad-core system, we have a relatively high per socket performance as compared to the dual-core system, which can be attributed to the shared L3 cache.

We have multiple applications that show slowdown in the quad-core or virtual node mode (VNM) modes as compared to single-core (SMP). In VNM mode 16 XT nodes are used while in SMP mode 64 XT nodes (256 cores) are reserved but only one core per node is used for 64 MPI tasks altogether. The S3D application has about a 25% slowdown in the mode where all four cores contribute to a calculation as compared to only single-core per processor. We collected hardware counter data using the PAPI library that confirms our findings [PAPI00]. L3 cache (shared between all four cores) behavior is measured and computed using the following PAPI native events. The L3 miss rate shows how frequently a miss occurs for a set of retired instructions. The L3 miss ratio indicates the portion of all L3 accesses that result in misses. Our results confirm that the L3 cache miss and request rate increase by a factor of two when using 4 cores per node versus using 1 core per node mode.

The most distinctive feature of the Petaflops XT5 system is the dual-socket, quad-core nodes as compared to a single-core socket node. In other words, there could be an additional level of memory and communication hierarchies that could be exposed to the application developers that are familiar with the quad-core XT4 memory sub-system. Although the optimization for the wide vector units would be beneficial for the XT5 system, the issues of memory sub-system are likely to become more complex since there will be 8 cores sharing the Hypertransport link on the XT5 node as compared to 4 cores on the XT4 node.

5 Conclusions and Future Plans

We have demonstrated how individual features of a system's hardware and software stack could influence performance of high-end applications on over a 250 Teraflops scale supercomputing platform. Our capability of comparing and contrasting performance and scaling of applications on multiple generations of Cray XT platforms, which share many system software and hardware features, enable us to not only identify the strategies to improve efficiencies on the current generation system but also prepare us to target the next-generation Petascale system. Since only a selection of features is studied in detail for this study, we plan on expanding the scope of this research by including application that have hybrid programming models to study the impact of within and across nodes and sockets. We are in process of working with application groups that have a flat MPI hierarchy models to explore and incorporate alternate work decomposition strategies on the XT5 platform.

Acknowledgements

This work was supported by the United States Department of Defense and used resources of the Extreme Scale Systems Center and the National Center for Computational Sciences at Oak Ridge National Laboratory, which is supported by the Office of Science of the Department of Energy under Contract DE-ASC05-00OR22725.

References

[Dagum98] Dagum, L., Menon, R.: OpenMP: An Industry-Standard API for Shared-Memory Programming. IEEE Computational Science & Engineering 5(1), 46–55 (1998)

[Snir98] Snir, M., Gropp, W.D., et al. (eds.): MPI – the complete reference (2-volume set), 2nd edn. MIT Press, Cambridge (1998)

[BGL05] Gara, A., et al.: Overview of the Blue Gene/L system architecture. IBM Journal of Research and Development, 49(2-3) (2005)

[BGP08] Vetter, J.S., et al.: Early Evaluation of IBM BlueGene/P. In: Proceedings of Supercomputing (2008)

[XT3a] Camp, W.J., Tomkins, J.L.: Thor's hammer: The first version of the Red Storm MPP architecture. In: Proceedings of Conference on High Performance Networking and Computing, Baltimore, MD (November 2002)

[XT3b] Vetter, J.S., Alam, S.R., et al.: Early Evaluation of the Cray XT3. In: Proc. IEEE International Parallel and Distributed Processing Symposium, IPDPS (2006)

[XT4a] Alam, S.R., Barrett, R.F., et al.: Cray XT4: An Early Evaluation for Petascale Scientific Simulation. In: Proceedings of the IEEE/ACM Conference on Supercomputing SC 2007 (2007)

[XT4b] Alam, S.R., Barrett, R.F., et al.: The Cray XT4 Quad-core: A First Look. In: Proceedings of the 50th Cray User Group (2008)

[Kelly05] Kelly, S., Brightwell, R.: Software architecture of the lightweight kernel, catamount. In: Proceedings of the 47th Cray User Group (2005)

[HPCCa] Luszczek, P., Dongarra, J., et al.: Introduction to the HPC Challenge Benchmark Suite (March 2005)

[HPCCb] High Performance Computing Challenge Benchmark Suite Website, `http://icl.cs.utk.edu/hpcc/`

[AORSA08] Barrett, R.F., Chan, T., et al.: A complex-variables version of high performance computing LINPACK benchmark, HPL (2008) (in preparation)

[Jaeger06] Jaeger, E.F., Berry, L.A., et al.: Self-consistent full-wave and Fokker-Planck calculations for ion cyclotron heating in non-Maxwellian plasmas. Physics of Plasmas (May 13, 2006)

[Jaeger07] Jaeger, E.F., Berry, L.A., et al.: Simulation of high power ICRF wave heating in the ITER burning plasma. In: Jaeger, E.F., Berry, L.A. (eds.) Proceedings of the 49th Annual Meeting of the Division of Plasma Physics of the American Physical Society, vol. 52. Bulletin of the American Physical Society (2007)

[Dongarra90] Dongarra, J.J., DuCroz, J., et al.: A set of level 3 basic linear algebra subprograms. ACM Trans.on Math. Soft. 16, 1–17 (1990)

[Langou07] Langou, J., Luszczek, P., et al.: Tools and techniques for exploiting the performance of 32 bit floating point arithmetic in obtaining 64 bit accuracy (revisiting iterative re_nement for linear systems). In: Proc. ACM/IEEE Supercomputing (SC 2006) (2006)

[Chen06] Chen, J.H., Hawkes, E.R., et al.: Direct numerical simulation of ignition front propagation in a constant volume with temperature inhomogeneities I. fundamental analysis and diagnostics. Combustion and flame 145, 128–144 (2006)

[Sankaran07] Sankaran, R., Hawkes, E.R., et al.: Structure of a spatially developing turbulent lean methane-air Bunsen flame. Proceedings of the combustion institute 31, 1291–1298 (2007)

[Hawkes07] Hawkes, E.R., Sankaran, R., et al.: Scalar mixing in direct numerical simulations of temporally evolving nonpremixed plane jet flames with skeletal CO-H2 kinetics. Proceedings of the combustion institute 31, 1633–1640 (2007)

[Kennedy00] Kennedy, C.A., Carpenter, M.H., Lewis, R.M.: Low-storage explicit Runge-Kutta schemes for the compressible Navier-Stokes equations. Applied numerical mathematics 35(3), 177–264 (2000)

[Scal] The ScaLAPACK Project, `http://www.netlib.org/scalapack/`

[Petit04] Petitet, A., Whaley, R.C., Dongarra, J.J., Cleary, A.: HPL: A portable high-performance LINPACK benchmark for distributed-memory computers (January 2004), `http://www.netlib.org/benchmark/hpl`

[PAPI00] Browne, S., Dongarra, J., et al.: A Scalable Cross-Platform Infrastructure for Application Performance Tuning Using Hardware Counters. In: Proceedings of Supercomputing (2000)