First 12-cabinets Cray XC30 System at CSCS: Scaling and Performance Efficiencies of Applications

Sadaf R. Alam, Themis Athanassiadou, Timothy W. Robinson, Gilles Fourestey, Andreas Jocksch, Luca Marsella, Jean-Guillaume Piccinali, Jeff Poznanovic, Benjamin Cumming, Dominik Ulmer
Swiss National Supercomputing Centre (CSCS), Lugano, Switzerland

***Abstract*—CSCS has recently deployed one of the largest Cray XC30 systems, which is composed of 6 groups or 12 cabinets of dual-socket Intel Sandy Bridge processors, and the new Aries network ASICs with a dragonfly topology. With respect to earlier Cray XT and XE series platforms, the Cray XC30 has several unique features that have the potential to affect application performance: (1) Intel Xeon vs. AMD Opteron based nodes; (2) Aries (XC30) vs. Gemini (XE/XK) vs. SeaStar (XT) network and router ASIC; (3) PCIe vs. Hypertransport interface to the network ASIC; (4) dragonfly vs. 3D torus topology; (5) mixed optical and copper vs. all copper cables; (6) growing number of compute nodes per communication NIC; (7) Hyperthreading enabled nodes; and (8) compute cabinet layouts. In this report, we compare scaling and performance efficiencies of a range of applications on CSCS's Cray XC30 and Cray XE6 platforms.**

Keywords-Cray XC30; Cray XE; Cray XT; applications performance; scalability; network topology; hyperthreading

I. INTRODUCTION

The Swiss National Supercomputing Centre (CSCS) has recently deployed the first generation of the Cray Cascade platform called XC30. The system is composed of 2,256 compute nodes and 24 service nodes. Each compute node contains a dual-socket, 8-core Intel Sandy Bridge processor and 32 Gbytes of memory. Four compute nodes on a blade are connected to an Aries network ASIC, which is a proprietary, high radix network router chip from Cray [5][7]. The network ASIC chips are interconnected in a dragonfly topology [8]. Altogether, the Cray XC30 system brings a range of unique features as compared to the previous generation of the Cray XT and XE series platforms from processor to node architectures to the network architecture and topology, to the system configuration. We therefore compare application performance efficiencies on this system to a previous generation Cray XE6 system, which is composed of AMD Interlagos processors and the Gemini interconnect [3][4]. We summarize the effects on application performance of key architectural, operating and programming environment features of the Cray XC30 and XE6 platforms installed at CSCS.

A Cray XC30 system called Piz Daint became available to CSCS users from April 2013. Therefore it has been critical to measure and analyze efficiency and scaling characteristics of a range of representative applications. In this report, we compare and contrast performance of representative applications from science domains such biological sciences and chemistry (CP2K, Gromacs and NAMD), earth sciences (SPECFEM3D) and climate (ECHAM and COSMO). Our target applications are implemented using MPI only or hybrid MPI and OpenMP programming models. In order to evaluate the high-global bandwidth dragonfly network, we target input configuration and test cases that are likely to stress the communication capabilities of the system. Moreover, we evaluate impact of hyperthreading for hybrid MPI and OpenMP applications, by mapping tasks and threads to physical cores as well as hyperthreads.

The observations that are being made using our target applications on the two systems enable us to correlate several distinctive features of the Cray XC30 platform with achievable performance and scaling of applications. A subset of these features include Intel Xeon vs. AMD Opteron based nodes, Aries vs. Gemini network and router chip, PCIe vs. Hypertransport interface to the network chip, dragonfly vs. 3D torus topology and the number of NUMA regions per node. Based on our experimental evaluation, we also identify areas of improvement and further development for the Cray XC30 platform.

The paper is organized as follows: an overview of the distinctive architectural and programming features of the two systems is provided in Section II. Section III contains results and observations for our target applications. Analysis and discussion of the results is provided in Section IV. Section V summarizes the key observations and findings and outlines the next steps for further improving the efficiency and scalability of applications.

II. CRAY XC30 AND CRAY XE6 PLATFORMS AT CSCS

At the time of writing, both the Cray XC30 (called Piz Daint) and Cray XE6 (called Monte Rosa) are production machines for the user program at CSCS. The mission of CSCS is to provide compute and storage facilities for researchers in HPC-based sciences for scientists at all academic institutions located in Switzerland. The Cray XC30 system is composed of 12 cabinets or 6 electrical groups, with a peak performance of 750 TFLOP/s and 72 TBytes memory while the Cray XE6 platform offers over 400 TFLOP/s and 47 TB of memory. The following subsections highlight distinctive features of the node and the network architecture as well as the programming and execution environments. A brief description of the scratch

file systems is also included as there are some distinctions both in terms of the file system layout and technologies. This report however does not focus on comparative evaluation of the scratch file systems of our target platforms.

A. Compute Node Architecture

One of the major distinctions between the Cray XC30 and Cray XE6 platforms is the processor architecture. Both systems have dual-socket processing nodes and 32 GBytes of DDR3-1600 memory. The CPU of an XC30 is an Intel Xeon Sandy Bridge processor (E5-2670) with 2.6 GHz clock frequency, 8 physical cores and 8 hyperthreads. The CPU of an XE6 system is an AMD Opteron 6272 processor, which has 8 core modules and 16 cores, operating at 2.1 GHz. While both CPUs support clock boost and NUMA memories and have similar instruction sets, there are some key differences. For example, the Opteron system supports both FMA4 and AVX. The Intel system has 2 NUMA memories connected through QPI while the AMD system has 4 NUMA memories connected via Hypertransport. These distinctions are important for users for on-node optimizations such as tuning for the appropriate vector lengths and OpenMP memory locality. Both nodes can execute up to 32 MPI tasks or OpenMP threads. A Cray XC30 will launch 16 hyperthreads to support 32 threads/processes per node while on the AMD node, 1 core per core module is assigned to each thread/processor.

B. Network Architecture

Our two target systems differ in terms of the network to node connectivity, network topology, design of the network and router chip and the routing schemes. The proprietary network and router chip of a Cray XC30 system is called Aries while its predecessor, the Cray XE6 system high-speed network, is composed of the Gemini chip. Each Aries chip is connected to four compute processing nodes via a PCI 3.0 x16 bus, which is an I/O bus. The Gemini chip, and its predecessor, the SeaStar chip, is connected to the processing nodes on a memory bus called Hypertransport [11]. The injection bandwidth of an Aries chip is considerably higher, however, there has not been a proportional reduction in the inter-Aries chips latencies.

The Gemini chips are interconnected with a 3D torus topology. A dragonfly topology has been implemented for the Cray XC30 platform. This unique topology has several distinct features. First, each cabinet of an XC30 system, which is twice as dense in terms of the number of compute nodes as compared to the XE6, is paired with another cabinet to provide an all-to-all electrical network. Each two cabinets form an electrical group. Each electrical group is then connected using optical cables to all other electrical groups. Optical connectivity is customizable. For example, the CSCS current system has populated about 25% of the optical cabling between groups. Hence, the global bandwidth of the system is about 25% of the maximum possible bandwidth that could be achieved by fully populating all connections. Still, the CSCS Cray XC30 system has over 4 times higher bisection bandwidth compared to the Cray XE6 system.

Another major difference is the way routing has been implemented for the Cray XC30 system, which has considerably higher all-to-all bandwidth within an electrical group and among electrical groups. The full adaptive routing and the network topology result in much reduced sensitivity for application placements as compared to the Cray XE6 platform. Our micro-benchmarking results, not included in this paper, confirmed improvements for all-to-all communication benchmarks on the Cray XC30 platform over the Cray XE6 system.

C. Programming and Execution Environment

Users porting applications from our Cray XE6 to the Cray XC30 platform observed minor differences in the early access phase of the system, primarily due to the integration of tools and libraries for different compilers. All compilers available on the Cray XE6 platforms are available on the Cray XC30 system, except for PGI. The numerical libraries, MPI and performance tools work seamlessly, with minor linking issues associated the Intel programming environment and numerical libraries. There are some differences in terms of CLE and version of the compilers, with the latest versions being the default on the XC30 platform. Hyperthreading can be enabled on the nodes with (-j) flag. By default, it is equal to one i.e. hyperthreading is disabled by default. Core specialization, i.e. the ability to offload operating system workload to dedicated cores, can be turned on and we observed improvements for MPI micro-benchmarking test cases and some user applications. CSCS ported SLURM to the latest ALPS/BASIL interface. Users with MPI only applications where they use 32 cores per node on the Cray XE6 platform may have to adjust to 16 cores per node, if they prefer using physical cores only on the Cray XC30 nodes. Memory per MPI task would double for from 1 GB per task to 2 GB per task for MPI only applications when one MPI task per physical core is used.

D. File System

The file system for the Cray XC30 platform at CSCS is provisioned according to its peak computing capabilities with an external Lustre file system. The version of Lustre is different on the two platforms. Both the throughput and metadata capabilities of the Cray XC30 system are considerably higher than the Cray XE6 platform. A Sonexion 1600 based file system has been deployed with 10 SSUs as an external file system. CSCS's Cray XE6 platform has an internal file system. We do not include any file system results because the system is still being tuned in order to obtain high efficiencies for the parallel file systems benchmarks.

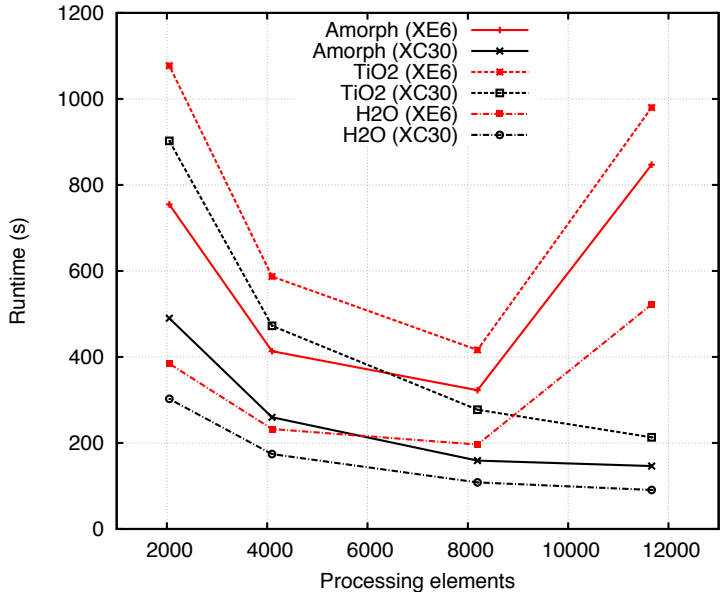

Figure 1: Time to solution as a function of the number of MPI processes (processing elements) for three CP2K benchmark systems (see text). The results from the XE6 system are shown in red and the XC30 results in black.

III. Applications Results

In addition to our target Cray platforms, the Cray XE6 (Monte Rosa) and Cray XC30 (Piz Daint), we collected data on an InfiniBand FDR cluster (Pilatus), which comprises dual-socket Sandy Bridge nodes (with an identical CPU to Piz Daint) and has a non-blocking tree topology network. This system was deployed at CSCS prior to the installation of the Cray XC30 platform in order to prepare migration of user applications from an AMD Opteron based system to an Intel Xeon based platform. In the following sections we present the results from a series of application benchmark studies.

A. CP2K

CP2K is a predictive quantum simulations tool based on Density Functional Theory and is used in materials science and chemistry [13]. The code is written mostly in Fortran (parts in C) and uses MPI and OpenMP for parallelization. The code can scale to tens of thousands of processing cores [12].

In our first benchmarking study we provide results obtained for three molecular systems: the energy computation for replicated H2O and TiO2 unit cells, and the electronic structure of an amorphous hole conducting material used in solar cells. The simulations make use of the order N (or Cannon) algorithm implemented in CP2K. On both the Cray XC30 and XE6 systems, 16 MPI tasks were used per node. On the Cray XE6, each MPI task spawns two OpenMP threads, such that there is one MPI task per Interlagos module and one OpenMP thread per processing core (2 cores are in a module, which share execution units and L2 cache).

The results for the three systems under investigation are shown in Figure 1. We first consider per node performance of CP2K for three test cases. Note that a processing node of the Cray XE6 platform delivers 268.8 GFLOP/s while the theoretical compute capability of Intel Sandy Bridge nodes is 332.8 GFLOP/s. Our results demonstrate that a higher fraction of the peak performance is achieved on the Cray XC30 platform as compared to the Cray XE6 platform for majority of our experiments. Furthermore, the differences in Cray XC30 and Cray XE6 runtimes grow with the number of nodes. We could attribute these findings to the peak floating-point capabilities of the system, fewer NUMA memories or simplified memory hierarchy of the Cray XC30 platform and a higher injection bandwidth of processing nodes.

An analysis of scaling results reveal similar trends for each benchmark case: the speed up observed on the XC30 as compared to the XE6 is between 1.2 and 3, for small and large node counts, respectively. Note that a Cray XC30 nodes peak floating-point performance is 1.2x of a Cray XE6 node. In the case that the number of processes is not a power of two (11,664 processes) the performance of the XE6 is very poor. No such anomalous behavior is observed for the XC30. This confirms one of the key requirements of the CP2K Cannon's algorithm. The global bandwidth requirement grows with the number of processing tasks and without a careful mapping onto the 3D torus network, the network latencies can dominate overall runtimes of the application. We plan on profiling the application to characterize the impact of network features, particularly,

high injection bandwidth to the network chip and the high global bandwidth within and across electrical groups on our target platforms.

Our second benchmark study compares the performance of the pure MPI version of CP2K on XC30 with hyperthreading enabled and disabled, and we compare the results to those obtained on the XE6 machine. We have used a water benchmark of 1,024 and 2,048 molecules, and the results are shown in Figure 2 and Figure 3, respectively. In all cases XC30 outperforms the XE6 in terms of both performance and scalability, no matter whether hyperthreading is enabled or disabled. In terms of the value of hyperthreading, the results are less clear. For the smaller test case, 1,024 water molecules, there is an overhead for using hyperthreading on the XC30 node. Since it is a smaller benchmark and does not scale to a large number of nodes, we conclude that similar workloads may not benefit from hyperthreading on the Cray XC30 system.

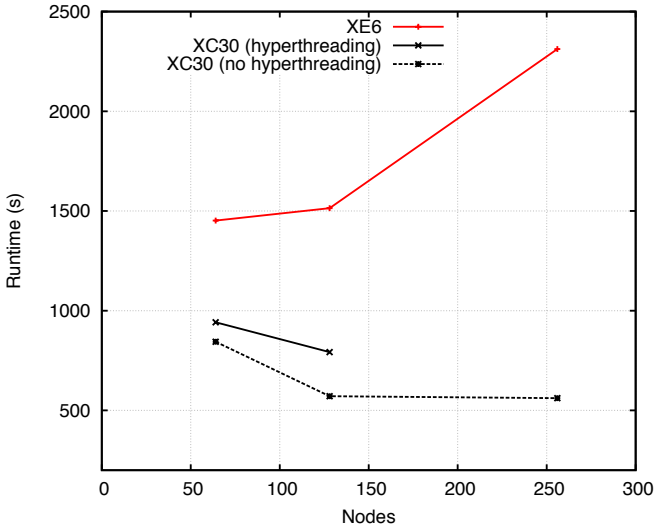

Figure 2: Time to solution as a function of the number of nodes for a 1024 water molecule benchmark using CP2K. The results for XE6 are compared to that of XC30 with and without hyperthreading enabled.

When we double the problem size, we observe a different effect, i.e. hypethreading could improve performance of the application on the same number of nodes. Smaller problem sizes, for example, 256 water molecules, exhibit performance characteristics similar to 1,024 water molecules for experiments with and without hyperthreading. Since the impact of hyperthreading on CP2K is unclear at the time of writing this report, we continue to investigate hyperthreading of MPI and hybrid MPI and OpenMP versions of CP2K for problems sizes that are typically used on the CSCS systems for production level simulations.

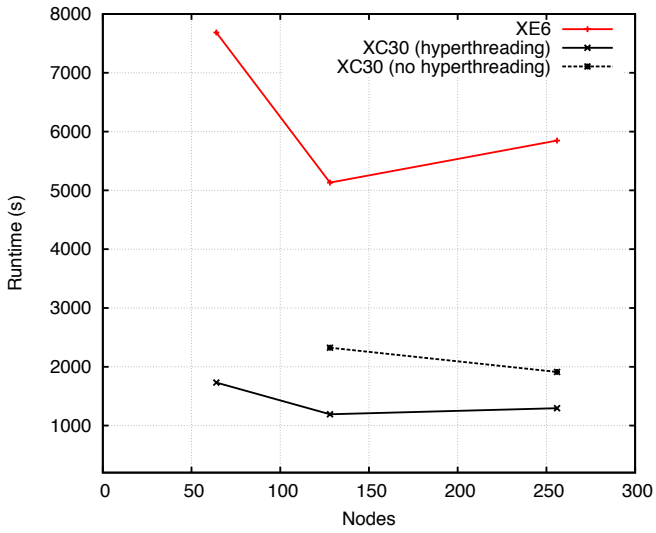

Figure 3: Time to solution as a function of the number of nodes for a 2048 water molecule benchmark using CP2K. The results for XE6 are compared to that of XC30 with and without hyperthreading enabled.

B. GROMACS

GROMACS is a versatile package used to perform molecular dynamics, i.e. simulating the Newtonian equations of motion for systems with hundreds to millions of particles [2]. It is primarily designed for biochemical molecules like proteins, lipids and nucleic acids that have a considerable number of complicated bonded interactions, but since GROMACS is extremely fast at calculating the non-bonded interactions (that usually dominate simulations) many research groups are also using it for research on non-biological systems, e.g. polymers. The version of GROMACS used to perform the benchmark was version 4.6, compiled with the Intel 2013 compiler on the XC30, and GCC 4.70 on the Cray XE6. We have also conducted experiments at CSCS on the Sandy Bridge processor to evaluate performance difference between the Intel and GNU compilers for GROMACS when allowing optimization with AVX intrinsics. We found virtually no discernable differences between the Intel and GNU compilers on the Intel Sandy Bridge platform.

We have performed a weak scaling experiment using water molecules, starting with a box of 32K water molecules at standard conditions on a single node and increasing the size of the system proportionally with the number of nodes, up to 33M water molecules for the 1024 nodes system. For each data point, the system was equilibrated for 100 ps using a time steps of 2 fs, and the production time for the experiment was 40 ps. Coulomb interactions were treated with Particle Mesh Ewald (PME). Berendsen temperature coupling was set at a reference temperature of 300.0 K and Berendsen isotropic pressure coupling was set with the compressibility for water at 1 atm and 300 K at 4.5×10^{-5} bar.

The benchmark compares the performance of the hybrid MPI/OpenMP code on the Cray XC30 (with hyperthreading enabled) and the Cray XE6. The jobs were run using 16 MPI tasks per node and 2 OpenMP threads per task, selecting numa_node affinity for CPUs with aprun. In other words, 32 MPI tasks and OpenMP threads per cores (16 x 2) are used on both Cray XE6 and Cray XC30 processing nodes. The numa_node affinity ensure that MPI processes use the NUMA memory affinities as there are 4 NUMA memories on a Cray XE6 processing node, which can lead to memory contention.

Results of the weak scaling experiments using the Gromacs application are shown Figure 4. Like CP2K results, we quickly observe higher efficiencies when performing a node-to-node comparison of the two systems. The XC30 shows an improvement in performance compared to the XE6 by a factor of about 1.7 for one and two nodes, increasing to a factor of about 2.2 for 256 nodes.

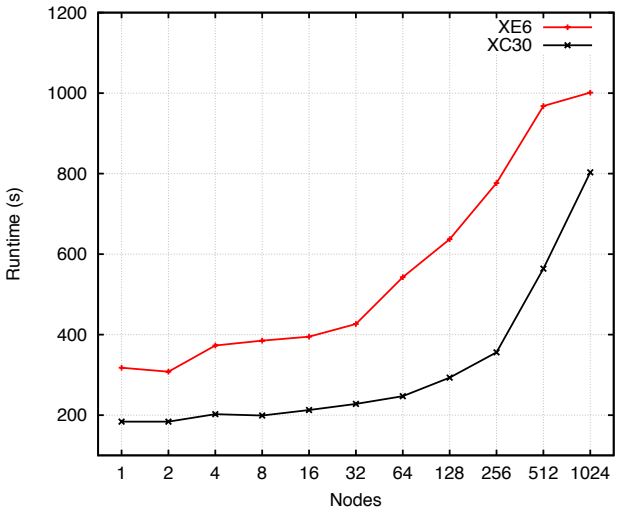

Figure 4: Weak scaling Gromacs using a box of 32K water molecules at standard conditions on a single node and increasing the size of the system proportionally with the number of nodes on the Cray XE6 and Cray XC30 platforms.

A careful analysis of network times and mapping onto logical to physical mapping of MPI tasks onto the processing nodes is necessary to explain the behavior shown in the figure. We can however highlight that experiments on 256 processing nodes on the Cray XC30 system can be performed within a single electrical group. There are 388 compute nodes within an electrical group, which have higher bandwidth and lower latencies as compared to inter-group communications that take place on the optical interconnect. As indicated in the previous subsection, only about a quarter of the optical links are populated in the CSCS Cray XC30 system.

This increasing performance improvement suggests that the dragonfly topology and Aries ASIC makes a significant difference for medium sized node counts; However, as seen in Figure 4, when the number of nodes used increases beyond a single group, then the performance gain decreases: to about a factor of about 1.7 on 512 nodes (two groups) and about 1.25 on 1024 nodes (three groups).

In particular, we need to understand the behavior for 1,024 nodes experiments. On the Cray XC30 system, 1,024 nodes can be spread over 3 or more electrical groups out of the 6 groups in total. This intergroup connectivity may have an impact on the weak scaling of the Gromacs test runs. We plan on collecting communication profile data on both systems, with and without hyperthreading, to understand which factors influence performance and scaling efficiencies of Gromacs on the Cray XC30 platform.

C. NAMD

NAMD is a classical molecular dynamics code designed for the simulation of large biomolecular systems, developed at the Theoretical and Computational Biophysics group at the University of Illinois [10]. We have run four biological systems from a whitepaper benchmark study produced by STFC Daresbury laboratory [9]. The four systems benchmarked are as follows, in order of increasing size: Crambin, a small seed storage protein from the Abyssinian cabbage (ca. 20K atoms); the glutamine binding protein GlnBP (ca. 60K atoms); the eipidermal growth factor receptor (EGFR; HER1 in humans) dimer, doubly ligated, on a POPC membrane bilayer (ca. 465K atoms); and two HER1 standing “proud” on the POPC membrane (ca. 1.4M atoms).

All calculations were preformed using 32 MPI processes per node on Rosa and 16 processes per node on Daint. The results obtained for the four biological systems are shown in Figures 5, 6, 7 and 8 respectively. Note that the numbers of nodes are indicated in the figures as we are interested in time to solution per node. The number of MPI tasks are doubled on the Cray XE6 (Rosa) platform for each test run. This may result in scaling efficiencies, especially for smaller systems. This phenomenon is illustrated for smaller systems, where Rosa and Daint performance start diverging at a smaller node count. In all cases, and at all node counts, the absolute performance per node on the XC30 exceeds that on the XE6. Moreover, the gap in performance between the two platforms increases markedly with increasing node counts for all system sizes.

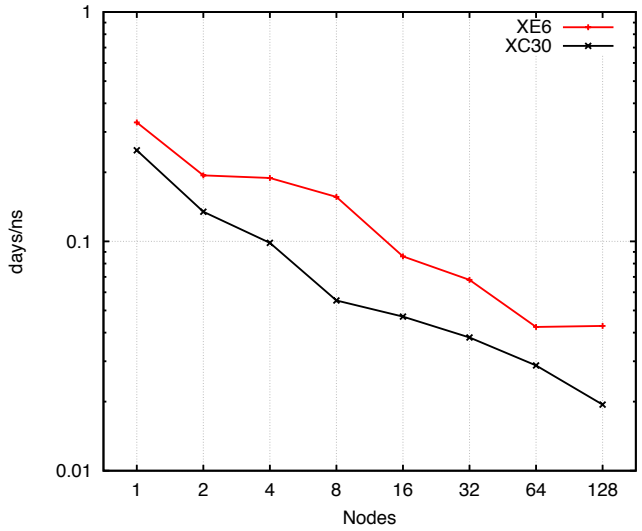

Figure 5: NAMD benchmark for 20K atoms system. Performance is shown as days/ns (less is better).

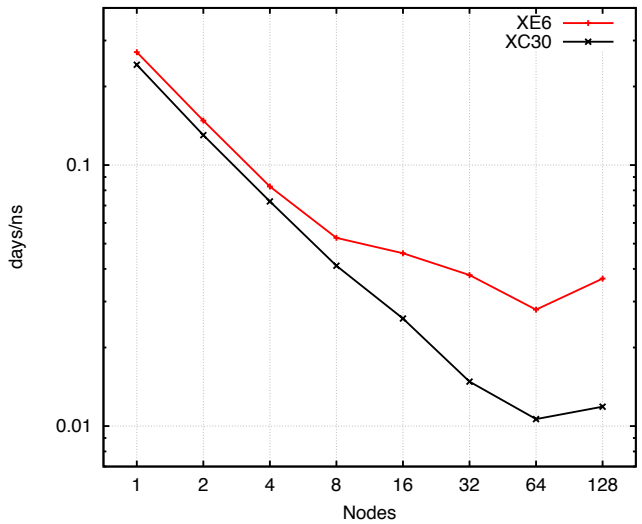

Figure 6: NAMD benchmark for 60K atoms system. Performance is shown as days/ns (less is better).

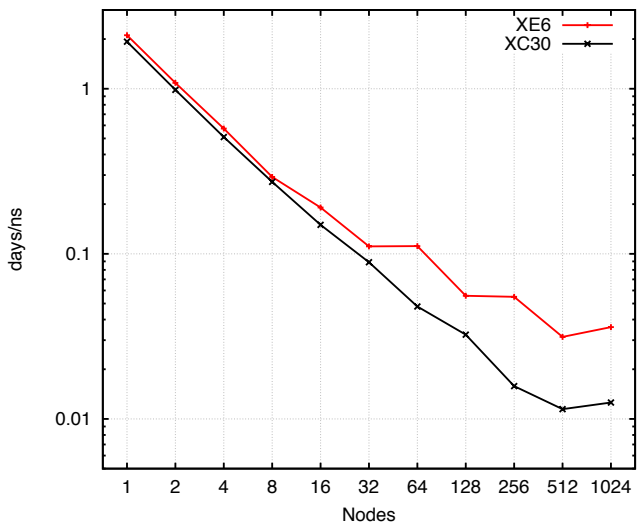

Figure 7: NAMD benchmark for 0.5M atoms system. Performance is shown as days/ns (less is better).

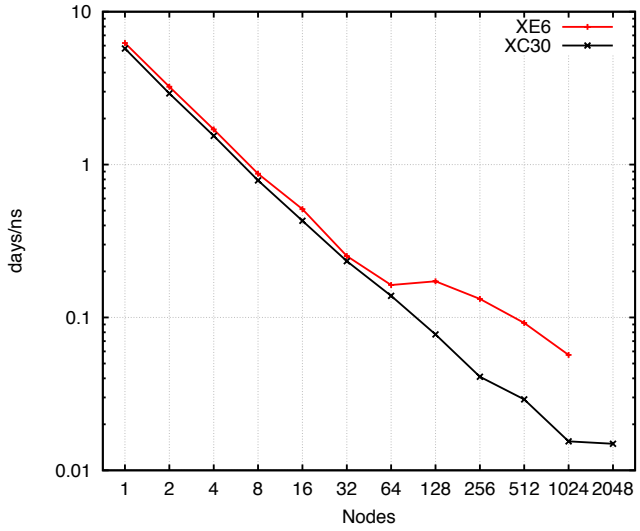

Figure 8: NAMD benchmark for 1.4M atoms system. Performance is shown as days/ns (less is better).

Additionally, the results on XC30 generally show far less sensitivity to specific node numbers and, in particular, far less variability upon successively doubling the node count: this phenomenon is most clearly demonstrated in Figure 7, whereupon a doubling of node counts beyond 32 nodes alternates between significant performance gain and almost no performance gain on the two target platforms. As indicated earlier, a careful analysis of communication profile for NAMD would reveal how the MPI communication times scale on two systems. We are also interested in finding out why NAMD does not show higher sensitivity to the node floating-point performance capabilities as we have observed for CP2K and Gromacs. We would therefore conduct analysis of computation and memory profile of NAMD for different problem sizes to understand this behavior and report findings to users who use this application for their production level simulations.

D. ECHAM6

ECHAM6 is a global climate model developed by the Max Planck Institute [1]. We used ECHAM6 svn revision r735 for our benchmarks, with an executable built with the following compilers: On Pilatus (Sandy Bridge) using GCC version 4.7.1, on XE6 using Intel version 12.1.2, on XC30 using Intel version 13.0.1. The model was run with a resolution of T63L47GR15 for 1 month (10/1961) on between 32 and 1056 cores using 32 MPI tasks per node, and the results are shown in Figure 9. The best overall

performance was obtained on the XC30, which outperformed the XE6 by about 50% on all node counts. Moreover, the benchmark case scaled to 512 cores or 16 nodes (with very small speedup) on the XC30 but only to 256 cores or 8 nodes on the XE6 and Infiniband cluster. On the Cray XE6 platform, 2 MPI tasks are mapped onto a core module or an MPI task is mapped onto a single core. On the Cray XC30 and Pilatus dual-socket Sandy Bridge nodes, 32 MPI tasks are mapped onto a node with 16 physical cores and 16 hyperthreads. We observe speedup for using hyperthreads on the Sandy Bridge processors. The difference between the Sandy Bridge cluster performance and Cray XC30 can be attributed to the compiler optimization and the way in which CPU and memory bindings are implemented on the Cray XC30 nodes. The scaling on the Pilatus system is unexpected since it has an InfiniBand FDR interconnect with a fully non-blocking tree topology. We plan on investigating both the node and scaling efficiencies on the InfiniBand cluster.

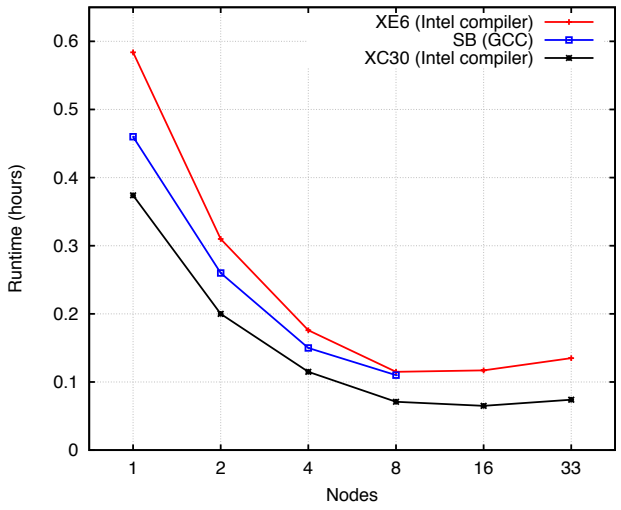

Figure 9: ECHAM6 benchmark case T63L47GR15. Performance is shown as time to solution in hours (less is better).

It is possible on the CRAY XC30 to specify how many CPUs to use per compute unit, i.e., hyperthreading enabled or disabled. An aprun command specifying "-j 0" will use two CPUs per compute unit, whereas "-j 1" will use one. Note that the use of the "-j 0" flag will reduce the number of nodes used by half: it is therefore worth investigating the relative performance of the two available modes. In Table 1 we present results for the ECHAM6 simulation comparing the performance of the two "-j" modes, with and without hyperthreading. When using up to eight compute nodes there is a small performance increase (up to 11%) when running with "-j 0". However, above eight nodes there is a very large performance decrease when using "-j 0": up to 65% for 32 nodes. Note that the ECHAM code stops scaling after 32 nodes and network cost could be higher for compute intensive applications.

Table 1: ECHAM6 runtime (hours) when using one ("-j 1") and two ("-j 0") CPUs per compute unit or core on XC30.

Compute nodes	"-j 1" time	"-j 0" time	Percentage improvement -j0/-j1
2	0.225	0.200	11%
4	0.126	0.115	9%
8	0.073	0.071	3%
16	0.048	0.065	-36%
32	0.045	0.074	-65%

E. SPECFEM3D

SPECFEM3D simulates three-dimensional seismic wave propagation through the Earth using the spectral-element method. We have performed a benchmark study comparing the performance of SPECFEM3D executables built with a variety of compilers on our XC30, XE6 and Sandy Bridge Infiniband cluster. The results are presented in Figure 10. In all cases the scaling is nearly perfect.

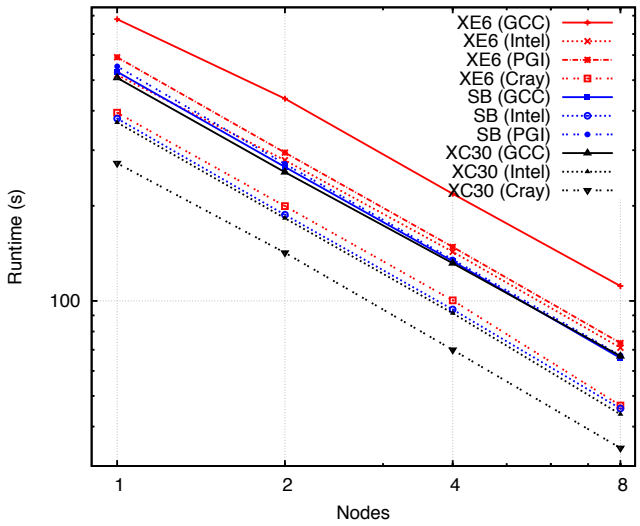

Figure 10: SPECFEM3D benchmark time to solution.

When comparing within a given compiler suite, the Sandy Bridge processor outperforms Interlagos by about a factor of 1.5, and in terms of absolute performance, the best results are achieved with XC30 and the Cray compiler. We note a strong dependence on the chosen compiler: Cray's fortran compiler performs the best, followed by ifort, and both of these compilers well outperform gfortran. The performance of the crayftn executable running on the XE6 even outperforms the gfortran version running on the XC30. Some further investigation showed that SPECFEM3D's performance was very dependent on the compiler's capability to vectorize the elastic forces hotspot. For up to 8 processor nodes, we observe near ideal speedup for SPECFEM3D on all our target platforms. We intend to

investigate scaling of SPECFEM3D on large number of nodes in future.

F. *COSMO*

COSMO is an atmospheric code used for both weather prediction (by DWD, Meteoswiss and others) and climate research (by ETHZ, KIT and others) [6]. It was originally implemented as a Fortran 90 code with flat MPI parallelization. There has been a significant effort invested in Switzerland (Meteoswiss, ETH, CSCS) to port this to hybrid multi-core and many-core systems. The results presented here have been obtained using the version of COSMO produced by this porting effort. The focus here is to understand characteristics of key application phases.

Figure 11 shows weak scaling on Cray XE6 and Cray XC30 platforms. Each node gets 128x128x60 sub-domain. Perform 12 hours of simulated weather with pseudo-random initial conditions. Plot shows average time to compute 1 hour of simulated weather. The "blip" at 16 nodes for Rosa is due to sub-optimal sub-domain mapping to nodes. We see that Daint comes out significantly faster (1.3 times faster at 256 nodes), which can be attributed to the higher clock frequency of the Cray XC30 Sandy Bridge processors.

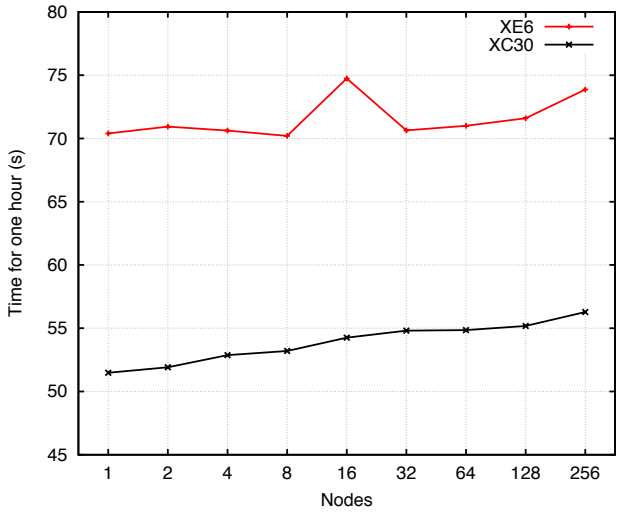

Figure 11: Weak scaling of COSMO with simulated weather with pseudo-random initial conditions

The new dynamical core of COSMO has been designed and optimized for the multi-core and many-core platforms. One of the auto-tuning features are introduced to exploit memory localities within a given memory hierarchy. We ran experiments to expose NUMA affinities on the AMD Interlagos processors (4 NUMA memories) versus Sandy Bridge nodes with 2 NUMA memories. Figure 12 shows strong scaling of the new dynamical core in COSMO. The time to perform 10 time steps is plotted against the size of the sub-domain. Tests performed with OpenMP on one NUMA domain (on Rosa this is one die with 8 cores, and on Daint this is one socket with 8 cores). In each case we see good strong scaling, with Daint being faster. Runtimes for one socket of Rosa are projected by providing an approximation of the socket time (1 socket of interlagos = 2 dies or NUMA memories).

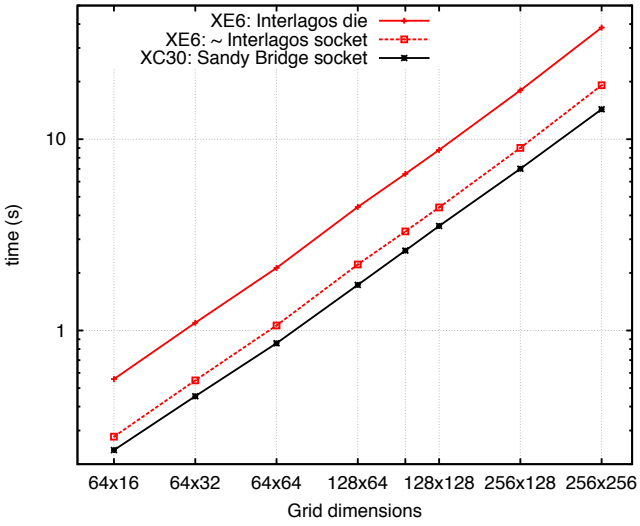

Figure 12: Measurement of NUMA sensitivity for the COSMO code

IV. Analysis and Discussion

The distinctive characteristics and strengths of the Cray XC30 node and network architecture has been confirmed by the results that we presented in the previous section for a range of CSCS representative applications and workloads. We have seen improvements in the node level performance as well as scaling efficiencies. The key findings and observations for characteristic features of the target platforms are as follows:

- *Floating-point Efficiency per node*: On daint, we observe d performance improvements over Rosa for Gromacs, CP2K and ECHAM. These applications demonstrate benefit of higher floating-point efficiencies per core.

- *Hyperthreading*: We observed performance improvements for Gromacs, CP2K and ECHAM applications. CP2K and ECHAM however demonstrate slowdown for some instances, which we plan on investigating.

- *NUMA memories*: Although careful memory profiling is required to confirm the impact of a simplified memory hierarchy of the Cray XC30 platform, results from COSMO experiments and for applications in hybrid MPI and OpenMP mode have shown that memory locality sensitive applications can benefit from the Cray XC30 memory hierarchy.

- *Global bandwidth*: Several applications, particularly CP2K, show significant improvement in scaling efficiencies, which can be attributed to a high global bandwidth of the Cray XC30 platform.

- *Network injection bandwidth*: Cray XC30 platform can sustain a large injection bandwidth as compared to the XE6 system. For NAMD and CP2K, we observed that the Cray XC30 network was able to sustain scaling efficiencies for higher node count. Further analysis is required to confirm these findings.

- *Topology and adaptive routing*: The limitation of the 3D torus topology has been addressed by the dragonfly topology and an adaptive routing scheme that slows for efficient global and irregular communication. We observed effect of topology for the CP2K experiments, where the code was only able to scale for the power of 2 processor counts. On the Cray XC30 platform, there was no such restriction.

V. SUMMARY AND NEXT STEPS

Overall results demonstrate the strength of the Cray XC30 for a range of workloads. We are therefore confident that the application developers and end users will observe sustained and improved performance as they migrate from the Cray XE6 platform to the Cray XC30 system. Since these studies are conducted on an early access system, there are some instances where we did not observe expected performance and scaling efficiencies. Furthermore, in some instances, we were unable to confirm whether an observation can be attributed to a given feature. Additional profiling experiments and analysis is therefore needed to understand performance sensitivity of different applications on a given feature of our target platforms. We plan on investigating this further as the operating and programming environment on the system matures. Another important feature of the system, the parallel file system, is also going to be investigated for file I/O intensive workloads once the file system has been tuned for the Cray XC30 platform. We have begun investigation of interference of jobs interference on the system. The system is expected to run a range and variety of workload and we would like to understand the impact of a fully adaptively routed system in production environment.

REFERENCES

[1] ECHAM. http://www.mpimet.mpg.de/en/wissenschaft/modelle/echam.html.

[2] Gromacs. http://www.gromacs.org/

[3] Alam, S. R., Fourestey, G., Giuffreda, M. G., McMurtrie, C. Architectural Features of Cray XE6 and the Path Towards Exascale. Book chapter in Contemporary High Performance Computing: From Petascale toward Exascale, (Chapman & Hall/CRC Computational Science), 2013.

[4] Alverson, B., Roweth, D., Kaplan, L.: The Gemini System Interconnect. High Performance Interconnects Symposium. 2010.

[5] Alverson, B., Froese, E., Roweth, D., Kaplan, L.: Cray XC Series Network. Technical report WP-Aries01-112, Cray Inc. 2012.

[6] Doms, G., Schattler, U.: The non-hydrostatic limited-area model LM (LokalModell) of DWD Part I: scientific documentation. German Weather Service, Offenbach/M. 1999.

[7] Faanes, G., Bataineh, A., Roweth, D., Court, T., Froese, E., Alverson, B., Johnson, T., Kopnick, J., Higgins, M., Reinhard, J.: Cray cascade: a scalable HPC system based on a Dragonfly network. Proceedings of the International Conference on High Performance Computing, Networking, Storage and Anal- ysis (SC '12). 2012.

[8] Kim, J., Dally, W. J., Scott, S., Abts, D.: Technology-Driven, Highly-Scalable Dragonfly Topology. Proc. of the International Symposium of Com- puter Architecture (ISCA). 2008.

[9] Loeffler, Hannes H. and Winn, Martyn D., Large biomolecular simulation on HPC Platforms III. AMBER, CHARMM, GROMACS, LAMMPS and NAMD, STFC Daresbury Laboratory, 2012

[10] Phillips, James C., Braun, Rosemary, Wang, Wei, Gumbart, James, Tajkhorshid, Emad, Villa, Elizabeth, Chipot, Christophe, Skeel, Robert D., Kalé, Laxmikant, and Schulten, Klaus, Scalable molecular dynamics with NAMD, J. Comput. Chem., 26, 1781-1802, 2005

[11] Vaughan, C., Rajan, M., Barrett, R., Doerfler, Do., Pedretti, K.: Investigat- ing the Impact of the Cielo Cray XE6 Architecture on Scientific Application Codes. Proceedings of the 2011 IEEE International Symposium on Parallel and Distributed Processing Workshops and PhD Forum (IPDPSW '11). 2011.

[12] VandeVondele, J., Borstnik. U,. Hutter, J.: Linear scaling self-consistent field calculations for millions of atoms in the condensed phase. Journal of Chemcical Theory and Computation. 8(10): 3565-3573. 2012.

[13] VandeVondele, J., Krack, M., Mohamed, F., Parrinello, M., Chassaing, T., Hutter, J.: QUICKSTEP: Fast and accurate density functional calculations using a mixed Gaussian and plane waves approach. Computer Physics Communications. 167 (2): 103-128. 2005.