Towards Fault-Tolerant Energy-Efficient High Performance Computing
in the Cloud

Kurt L. Keville
Mass. Inst. of Technology
Cambridge, MA, USA
Email: kkeville@mit.edu

Rohan Garg
Northeastern University
Boston, MA, USA
Email: rohgarg@ccs.neu.edu

David J. Yates
Bentley University
Waltham, MA, USA
Email: dyates@bentley.edu

Kapil Arya*, Gene Cooperman*
Northeastern University
Boston, MA, USA
Email: {kapil,gene}@ccs.neu.edu

Abstract—In cluster computing, power and cooling represent a significant cost compared to the hardware itself. This is of special concern in the cloud, which provides access to large numbers of computers. We examine the use of ARM-based clusters for low-power, high performance computing. This work examines two likely use-modes: (i) a standard dedicated cluster; and (ii) a cluster of pre-configured virtual machines in the cloud. A 40-node department-level cluster based on an ARM Cortex-A9 is compared against a similar cluster based on an Intel Core 2 Duo, in contrast to a recent similar study on just a 4-node cluster. For the NAS benchmarks on 32-node clusters, ARM was found to have a power efficiency ranging from 1.3 to 6.2 times greater than that of Intel. This is despite Intel's approximately five times greater performance. The particular efficiency ratio depends primarily on the size of the working set relative to L2 cache. In addition to energy-efficient computing, this study also emphasizes fault tolerance: an important ingredient in high performance computing. It relies on two recent extensions to the DMTCP checkpoint-restart package. DMTCP was extended (i) to support ARM CPUs, and (ii) to support checkpointing of the Qemu virtual machine in user-mode. DMTCP is used both to checkpoint native distributed applications, and to checkpoint a network of virtual machines. This latter case demonstrates the ability to deploy pre-configured software in virtual machines hosted in the cloud, and further to migrate cluster computation between hosts in the cloud.

I. Introduction

High performance computing is facing a power crisis. The largest supercomputers today consume power and cooling comparable in electricity use to a small city. The ARM architecture is promising as a low-power alternative that supports cluster computing.

While the ultimate target for ARM-based cluster computing will be the 64-bit ARMv8 low-power CPU (see discussion in Section II), we can estimate the benefits of such a low-power cluster by examining an ARM-based cluster using today's 32-bit Cortex-A9 CPU.

We analyze the performance of a cluster of 32-bit 1 GHz ARMv7 dual-core CPUs based on Texas Instruments OMAP4 CPUs. For a cluster of up to 40 OMAP4 ARM

CPUs, we show a performance-to-power ratio that is from 1.3 to 6.2 times better than a similar cluster of 3 GHz Intel Core 2 Duo CPUs running in 32-bit mode. The Intel Core 2 Duo consumes approximately 70 Watts, while the ARM Cortex-A9 consumes approximately 5 Watts. Nevertheless, in raw speed, the Intel Core 2 Duo is approximately 5 times faster than the ARM Cortex-A9.

This is a preliminary study based on the 32-bit architectures. Much of the speed advantage of Intel appears due to the three-times higher clock rate of the Intel CPU — a ratio that will change with the next generation of ARM CPUs. According to ARM roadmaps, at the 64-bit level (ARMv8), ARM will have increased their performance, allowing them to come closer to Intel speeds, while maintaining their current power usage. The added performance will come from an increased ARM clock rate (2 GHz and higher), the 64-bit architecture, and more cores per chip. Current Intel CPUs also show a performance improvement over the Intel Core 2 Duo, but not to as great an extent. Nevertheless, even at the 32-bit level, the ARM architecture demonstrates attractive possibilities.

We present a comparison that tests two different domains of high performance computing (HPC):

1) standard high performance computing on a dedicated cluster; and
2) a cluster of Qemu virtual machines running in the cloud.

Performance and energy measurements are presented for ARM and for Intel in both domains. Further, times for checkpoint-restart are presented.

In order to support these experiments, DMTCP was extended in two ways. First, it was extended to support the ARM architecture (new with version 1.2.5; released in June, 2012). Second, an internal version of DMTCP was extended to checkpoint the Qemu virtual machine. We do not know of any other checkpoint-restart package that has demonstrated this capability.

A further contribution of this work is to test scalability of ARM and Intel Core 2 Duo for realistic clusters scaling up to 40 nodes. Two other recent studies had only considered clusters of 4 nodes [1] and of 5 nodes [2].

IEEE computer society

In the rest of this paper, Section II provides background on the roadmap of the ARM CPU architecture, while Section III reviews DMTCP. Section IV briefly reviews the power measurements. Section V presents the experimental results and analysis. Section VI describes related work, and Section VII presents the conclusions.

II. BACKGROUND: ARM CPU ARCHITECTURE

ARM is the most widely used general purpose CPU based on number of units. ARMv7 is the latest 32-bit ARM architecture, and ARMv8 is the previously announced 64-bit ARM architecture. The ARM Limited corporation does not manufacture CPUs, but instead licenses their ARM designs to other companies.

32-bit ARMv7: Within the ARMv7 architecture, the current generation for high performance computing is the 32-bit Cortex-A9. The TI OMAP4 CPU used here is an implementation of the Cortex-A9. In late 2012, the next generation of ARM processors, the Cortex-A15, will be available.

64-bit ARMv8: Up to a year later (in late 2013), current roadmaps project availability in quantity of the follow-on generation (64-bit ARMv8 architecture). In one example, AppliedMicro projects in their roadmap a 3 GHz ARMv8 CPU consuming 2 Watts per core [3].

Figure 1 provides a view of the TI PandaBoard cluster, along with a chair placed beside it in order to show the scale.

Figure 1.   TI PandaBoard cluster used for experiments (photo by Lisa DeLacey)

III. BACKGROUND: DMTCP

DMTCP [4] (Distributed Multi-Threaded CheckPointing) is open source software. It supports *transparent* checkpoint-restart. In particular, it checkpoints the several dialects of MPI directly at the socket level, without the use of any MPI-specific hooks. Figure 2 shows the architecture of DMTCP. The three core commands of DMTCP are:

```
dmtcp_command a.out
dmtcp_command --checkpoint
dmtcp_restart ckpt_a.out_*.dmtcp
```

When the user invokes `a.out`, via the command `dmtcp_command`, the LD_PRELOAD environment variable causes a DMTCP-specific "hijack" library to be loaded prior to executing the end user's "main" routine. This library creates a checkpoint thread and creates a signal handler for `SIGUSR2`, the intra-process checkpoint signal. The library also creates wrappers for several system calls. The checkpoint thread then connects to the DMTCP coordinator, and waits for a checkpoint command.

Upon receiving a checkpoint request from the coordinator, the checkpoint thread quiesces each of the user threads by sending the `SIGUSR2` checkpoint signal. Next, DMTCP "drains" any data from the user sockets. To do this the DMTCP checkpoint thread sends a "cookie" through the sender end of each socket, and then reads from the receiver end of each socket until seeing the "cookie." In-transit data in socket buffers is thus saved in user space. Next, the checkpoint thread interrogates the kernel for information such as the open sockets, ptys, various file descriptors, mapped memory regions, etc. Information such as socket options and the current file offset for open files are also saved.
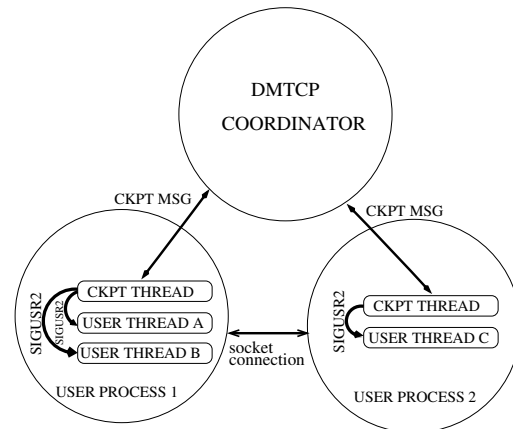

Figure 2.   Architecture of DMTCP

DMTCP is also "inherited" within and between processes. Any calls to `pthread_create()` cause DMTCP to note the new thread. Any calls to `fork()` cause the DMTCP "hijack" library to be loaded into the child process. Similarly, remote shell calls are detected, and the "hijack" library is also loaded into the remote process when it starts up.

IV. POWER MEASUREMENTS

For the Intel Core 2 Duo, power measurements per node were taken from the Green500 list [5], as being more representative. Specifically, number 115 on the June, 2008 list of the Green500 appears as the largest cluster using Intel Core 2 Duo nodes. That cluster consisted of 1024 dual-core nodes, consuming 71.93 kW total power (excluding the network switch and disks). Based on this, the power per

node used in this paper for Intel nodes is 71930/1024 = 70.24 Watts per node. Note that the Core 2 Duo from June, 2008 was running at 1.5 GHz. Our own newer cluster was running at 3.0 GHz. We conservatively continue to use the value of 70.24 Watts for Intel, thereby giving the benefit of doubt to the Intel CPU.

Power consumption on the ARM CPUs was found to vary from 3 Watts when idle to 5 Watts under load. Direct power measurements were taken periodically to compute performance-to-energy ratios. Specifically, power measurements were taken every 5 seconds. Power was found to vary by about 15% during a single benchmark run. An average of the power during the benchmark was used.

The TI OMAP4 CPU has the unusual feature of being able to separately turn individual cores on and off. Anecdotally, HPCC/Linpack consumes 23 Watts on a six-node single-core configuration. This increases to only 30 Watts for a six-node dual-core configuration.

V. Experimental Results

A. Configuration

We compare nodes on a cluster of ARM OMAP4 processors against Intel Core 2 Duo processors, a recent dual-core CPU architecture, with 6 MB of L2 cache. The ARM CPU is a TI OMAP 4430 with 1 MB L2 cache on a PandaBoard motherboard. The Intel architecture is part of a Dell Optiplex 960 computer. Both CPUs are dual-core. The TI OMAP4 runs at 1 GHz, with 1 GB of RAM, while the Intel Core 2 Duo runs at 3 GHz, with 4 GB of RAM.

We present ARM and Intel results for 32 to 40 nodes below. In our experiments, The ARM computers were using gcc-4.7 under Ubuntu 12.04, while the Intel computers were using gcc-4.5 under Ubuntu 9.10. Both use 100 Mbps Switched Fast Ethernet.

As described in Section IV, the assumed power usage was taken as 70.24 Watts per CPU core for the Intel processors, based on values for the Intel Core 2 Duo from the Green500 list [5]. This was considered fairer for Intel measurements, since the local computers used for time measurements were not designed for low power.

Power measurements do not include the power of the Ethernet switch. This was measured separately at 23 Watts for a 48-port switch in the ARM cluster. Hence, power used by the cluster dominates power used by the switch. The ARM computers were run from local SD media.

The benchmarks used were from the NAS Parallel Benchmarks for MPI [6], [7]. MPICH-2 was used as the MPI dialect in all tests. To demonstrate native distributed computation, six NAS benchmarks were run using 32 nodes. In addition, NAS/LU (Lower-Upper Gauss-Seidel Solver) was used in experiments to test scaling our ARM cluster to 40 nodes in dedicated cluster and cloud-based modes.

The choice of Intel Core 2 Duo to represent the x86 architecture is motivated by the work of Keys et al. [2].

They showed the Intel Core 2 Duo to have performance per unit of power better than many other Intel and AMD CPUs, including several server processors.

Researchers at Aalto University [1] have shown that the performance to power ratio of a 4-node ARM-based cluster of PandaBoards is superior to that of a similar 4-node x86-based cluster by a factor of 1.2 for video transcoding and up to a factor of 9.5 for an in-memory database.

B. Dedicated Cluster Results

Table I shows time and energy performance for the NAS Parallel Benchmarks [6] running on dedicated ARM- and Intel-based clusters, without virtual machines.

Benchmark	ARM		Intel		Ratio
	Time (s)	Energy (KJ)	Time (s)	Energy (KJ)	(ARM/Intel)
NAS/CG.A	7.79	1.18	3.27	7.35	6.2
NAS/EP.A	13.00	1.97	1.12	2.52	1.3
NAS/FT.A	13.25	2.01	3.61	8.12	4.0
NAS/IS.A	3.37	0.51	1.16	2.61	5.1
NAS/LU.A	138.98	21.08	12.03	27.04	1.3
NAS/MG.A	4.50	0.68	0.68	1.53	2.3

Table I
NATIVE APPLICATIONS (NO VM). CLUSTERS FOR ARM AND INTEL BOTH CONTAIN 32 NODES. THE ARM/INTEL RATIO COMPARES THE PERFORMANCE-TO-ENERGY RATIOS FOR THE TWO ARCHITECTURES. THE ENERGY IS THE READING FROM A POWER METER, INTEGRATED OVER THE TIME OF THE BENCHMARK FOR THE 32 NODES. AN ARM/INTEL RATIO GREATER THAN 1.0 FAVORS ARM.

Table I, above, presents results for a variety of NAS-MPI benchmarks. The tests show a performance to power advantage for ARM over Intel that ranges from a ratio of 1.3 to 6.2. The ratios correlate well with whether the working set fits in L2 cache. The Intel CPU has 6 MB of cache, while ARM has only 1 MB. CG (conjugate gradient: eigenvalue of a single matrix) and IS (integer sort using bucket sort) have relatively small working sets, while LU matrix factorization ($64 \times 64 \times 64$ for LU.A), and MG (MultiGrid; 3-D discrete Poisson equation) have larger working sets due to the problems being of higher dimension. The EP benchmark (embarassingly parallel) seems to be a special case due to its extensive use of square roots and logarithms. This takes advantage of Intel's CPU hardware support.

Table II, below, demonstrates the scalability of the ARM and Intel clusters running a native application as the number of nodes in the cluster varies. The NAS/LU.A benchmark used in Table II was chosen for scalability tests. NAS/LU.A was conservatively chosen because it favored the Intel CPU the most (aside from the EP special case). The energy efficiency ratios in Table II immediately highlight the two regimes of one to four nodes, and eight or more nodes. One to four nodes favor Intel due to the heavy pressure on the CPU cache, which is only 1 MB in the case of ARM. For eight nodes or more, network communication stress the computation to a greater extent, thus bringing the performance of ARM and Intel closer together.. We stress

again that LU was chosen to be conservative. The choice of NAS/CG or NAS/IS would very strongly favor ARM, as already seen by the large ratios in Table I.

Num. nodes	ARM		Intel		Ratio
	Time (s)	Energy (KJ)	Time (s)	Energy (KJ)	(ARM/Intel)
NAS-MPI/LU.A					
1	949.11	10.7	125.7	8.8	0.86
2	502.72	10.4	63.21	8.9	0.85
4	380.5	12.30	39.93	11.2	0.91
8	341.5	15.25	31.79	17.9	1.17
16	183.8	16.41	16.11	18.1	1.10
32	139.2	21.39	12.03	27.0	1.26
40	120.0	22.60	10.12	28.4	1.25

Table II
NATIVE APPLICATION (NO VM). SCALABILITY IS DEMONSTRATED AS THE NUMBER OF NODES IN A CLUSTER VARIES. THE NAS/LU.A BENCHMARK IS USED. THE RATIO IN THE LAST COLUMN IS DEFINED AS IN TABLE I.

Next, Table III shows the checkpoint times for the NAS/LU.A benchmark.

Num. nodes	ARM	Intel
	Ckpt (s)	Ckpt (s)
NAS-MPI/LU.A		
1	8.2	2.65
2	8.4	2.95
4	8.2	3.13
8	9.6	3.28
16	12.0	3.9
32	20.1	4.68
40	26.7	9.36

Table III
NATIVE APPLICATION (NO VM). SCALABILITY FOR CHECKPOINTING IS DEMONSTRATED AS THE NUMBER OF NODES IN A CLUSTER VARIES. CHECKPOINTS ARE TAKEN TO /TMP, RESIDING ON A LOCAL DISK (INTEL) OR A LOCAL SD CARD (ARM). FOR THE 40-NODE INTEL CASE, IT IS CONJECTURED THAT ONE NODE WAS PARTICULARLY SLOW.

Although writing to local disk (Intel) or an SD card (ARM) can contribute to the total time, operating systems hide this time by buffering disk writes. Further, DMTCP dynamically invokes gzip for compression. So, we believe the times to be limited largely by speed of RAM. The ARM clock speed is three times slower than that of Intel, which is consistent with checkpoint times for ARM being up to three times slower than Intel. Nevertheless the ARM CPU consumes less than a tenth the energy of the Intel CPU. This gives ARM an overall energy efficiency advantage.

C. Cloud-Based Cluster Results

Table IV shows the ability to use the Qemu virtual machine to run distributed computations in the cloud.

In Table IV, the times for checkpointing a virtual machine are now much closer between ARM and Intel. We believe this is because the memory footprint of a virtual machine is too large for the operating system to buffer most writes. Since checkpointing in both clusters is limited by disk-write speed, Intel loses much of its speed advantage, thereby yielding energy efficiency ratios heavily in favor of ARM.

Num. nodes	ARM		Intel		Ratio
	Time (s)	Energy (KJ)	Time (s)	Energy (KJ)	(ARM/Intel)
NAS-MPI/LU.A					
1	19.3	0.22	13.3	0.93	4.2
2	22.3	0.46	16.6	2.32	5.0
4	22.2	0.65	20.8	5.84	9.0
8	24.1	1.08	21.3	11.97	11.1
16	29.9	2.67	21.8	24.50	9.2
32	35.2	5.34	25.2	56.64	10.6
40	60.2	11.34	40.8	113.65	10.0

Table IV
CHECKPOINTING EFFICIENCY ON ARM VERSUS INTEL USING THE QEMU VM. QEMU IS RUN IN USER-MODE. CLUSTERS ON ARM AND INTEL BOTH CONTAIN UP TO 40 NODES. THE ARM/INTEL RATIO IS DEFINED AS IN TABLE I.

VI. RELATED WORK

Our work on leveraging checkpoint-restart for fault-tolerant, energy-efficient high performance computing (HPC) builds on earlier work by others (e.g. `libckpt` [8], Condor [9], and Sun Grid Engine [10]) and also our own work [4]. Using DMTCP to run HPC applications in a cluster also complements work by others [11], which explores using DMTCP to implement speculative software parallelism for large-scale distributed memory applications. This study enables HPC applications linked with DMTCP, potentially running on user-mode Qemu virtual machines [12], to checkpoint, restart and migrate as required.

Using ARM-based systems as the building block of choice for our HPC cluster is motivated by the research of several groups. For example, the AppleTV cluster research at Ludwig-Maximilians-Universität [13] has demonstrated mobile or consumer ARM-based systems a feasible technology basis for future HPC system designs. The results in [1] improve on cluster computing energy-efficiency results based on processors for embedded systems [14], and also on processors for laptops [2].

Recently, there has also been an emphasis on NVIDIA GPUs to provide another low-power alternative for cluster computing, e.g. [15]. However, the NVIDIA architecture depends on special tuning of GPU kernels to accommodate a SIMD style of programming. Most of the software tested in Section V has not been ported to the NVIDIA GPU.

Finally, our work assumes that chip and board designs have an ideal configuration and clock speed that delivers the best performance per unit of power for most HPC applications. (Recall that the TI OMAP-based PandaBoard consumed 23 Watts for six cores running a representative HPC workload, but only 30 Watts for twelve cores.) We therefore argue that power- or energy-proportional designs [16] for HPC should adjust the aggregate performance of a cluster by running servers (or blades) in their ideal configuration and turning them on and off on demand. The merit of this idea has been demonstrated by others, e.g. [17], and can leverage DMTCP to consolidate and distribute tasks among nodes either within a cluster or between clusters, possibly hosted

in the cloud. This approach is in contrast to dynamic voltage and frequency scaling (DVFS) [7], [18], which recently has been shown to provide limited power savings [1].

VII. CONCLUSION

The 1 GHz dual-core ARM Cortex-A9 demonstrated an energy efficiency ratio 1.3 to 6.2 times greater than a 3 GHz Intel Core 2 Duo in 32-node cluster configurations. ARM was more energy efficient despite Intel having approximately five times greater performance. For those benchmarks with the largest working sets and smallest cluster sizes (just one, two, or four nodes), the ratio fell as low as 0.9, since this stressed primarily the CPU cache and not the network. The benchmarks favored ARM especially when the benchmark had a small working set, since the ARM CPU had only a 1 MB L2 cache, as opposed to Intel's 6 MB cache. A large cluster size also favored ARM, since the problem data could be distributed across the network. This stresses the network, but not the CPU cache. As the current ARM Cortex-A9 generation is replaced by by the 32-bit ARM Cortex-A15 and then the future 64-bit ARMv8, this may further bias the ratios toward ARM. In particular, AppliedMicro projects in their roadmap a 3 GHz ARMv8 CPU consuming 2 Watts per core [3]. The comparison here using the Cortex-A9 provides designers of energy-efficient systems with initial estimates of energy consumption on realistic benchmarks for department-level clusters, or in the cloud.

REFERENCES

[1] Z. Ou, B. Pang, Y. Deng, J. K. Nurminen, A. Ylä-Jääski, and P. Hui, "Energy- and cost-efficiency analysis of ARM-based clusters," in *Proc. of the Int'l Symp. on Cluster Computing and the Grid (CCGRID)*, Ottawa, Canada, 2012.

[2] L. Keys, S. Rivoire, and J. Davis, "The search for energy-efficient building blocks for the data center," in *Computer Architecture*, ser. LNCS, A. L. Varbanescu, A. Molnos, and R. van Nieuwpoort, Eds. Berlin, Germany: Springer, 2011, vol. 6161, pp. 172–182.

[3] AppliedMicro, "X-Gene overview: Press briefing," January 2012, slide 14 of http://www.apm.com/global/x-gene/docs/X-GeneOverview.pdf.

[4] J. Ansel, K. Arya, and G. Cooperman, "DMTCP: Transparent checkpointing for cluster computations and the desktop," in *Proc. of the Int'l Parallel and Distributed Processing Symp. (IPDPS)*. Rome, Italy: IEEE, 2009, pp. 1–12.

[5] Green500, "The Green500 list (June, 2008; #480; Core2Duo 1.5 GHz)," http://www.top500.org/system/performance/9162, accessed June 5, 2012.

[6] "NAS parallel benchmarks (MPI)." [Online]. Available: http://www.nas.nasa.gov/publications/npb.html

[7] V. W. Freeh, F. Pan, N. Kappiah, D. K. Lowenthal, and R. Springer, "Exploring the energy-time tradeoff in MPI programs on a power-scalable cluster," in *Proc. of the Int'l Parallel and Distributed Processing Symp. (IPDPS)*, vol. 1. Denver, CO: IEEE, 2005.

[8] J. S. Plank, M. Beck, G. Kingsley, and K. Li, "Libckpt: Transparent checkpointing under Unix," in *Proc. of the USENIX Winter Technical Conf.*, New Orleans, LA, 1995, pp. 213–223.

[9] M. Litzkow, T. Tannenbaum, J. Basney, and M. Livny, "Checkpoint and migration of UNIX processes in the Condor distributed processing system," University of Wisconsin, Computer Sciences Department, Tech. Rep. UW-CS-TR-1346, Apr. 1997.

[10] W. Gentzsch, "Sun Grid Engine: Towards creating a compute power grid," in *Proc. of the Int'l Symp. on Cluster Computing and the Grid (CCGRID)*, Brisbane, Australia, 2001, pp. 35–39.

[11] D. Ghoshal, S. R. Ramkumar, and A. Chauhan, "Distributed speculative parallelization using checkpoint restart," *Procedia Computer Science*, vol. 4, pp. 422–431, 2011.

[12] F. Bellard, "QEMU, a fast and portable dynamic translator," in *Proc. of the USENIX Annual Technical Conf.*, Anaheim, CA, 2005, pp. 41–46.

[13] K. Fürlinger, C. Klausecker, and D. Kranzlmüller, "Towards energy efficient parallel computing on consumer electronic devices," in *Information and Communication on Technology for the Fight against Global Warming*, ser. LNCS, D. Kranzlmüller and A. M. Toja, Eds. Berlin, Germany: Springer, 2011, pp. 1–9.

[14] D. G. Andersen, J. Franklin, M. Kaminsky, A. Phanishayee, L. Tan, and V. Vasudevan, "FAWN: A fast array of wimpy nodes," in *Proc. of the ACM Symp. on Operating Systems Principles (SOSP)*, Big Sky, MT, 2009, pp. 1–14.

[15] Z. Fan, F. Qiu, A. Kaufman, and S. Yoakum-Stover, "GPU cluster for high performance computing," in *Proc. of the ACM/IEEE Conf. on Supercomputing (SC)*, Pittsburgh, PA, 2004, pp. 47–58.

[16] L. A. Barroso and U. Hölzle, "The case for energy-proportional computing," *IEEE Computer*, vol. 40, no. 12, pp. 33–37, Dec. 2007.

[17] E. Pinheiro, R. Bianchini, E. V. Carrera, and T. Heath, "Dynamic cluster reconfiguration for power and performance," in *Compilers and Operating Systems for Low Power*, L. Benini, M. Kandemir, and J. Ramanujam, Eds. Norwell, MA, USA: Kluwer Academic Publishers, 2003, pp. 75–94.

[18] C.-h. Hsu and W.-c. Feng, "A power-aware run-time system for high-performance computing," in *Proc. of the ACM/IEEE Conf. on Supercomputing (SC)*, Seattle, WA, 2005, pp. 1–9.