University College Dublin

Optimization of Multithreaded Data-parallel Applications on Modern Multicore CPUs For Performance and Energy Using Application-level Decision Variables

Semyon Khokhriakov

UCD Student Number: 15204508

This thesis is submitted to University College Dublin in fulfilment of the requirements for the degree of

Doctor of Philosophy in Computer Science

School of Computer Science

Head of School: Assoc. Prof. Chris Bleakley

Research Supervisor: Assoc. Prof. Alexey Lastovetsky

Co-Supervisor: Dr. Ravi Reddy Manumachu

September 2019

Abstract

Performance and energy are two most important objectives for optimization on modern parallel platforms such as supercomputers, high performance computing (HPC) clusters, and cloud computing infrastructures. These platforms are now ubiquitously equipped with multicore CPUs to address the twin critical concerns of performance and energy efficiency. The multicore CPUs feature tight integration of tens of cores organized in one or more sockets with multi-level cache hierarchy. Such tight integration, however, leads to several inherent complexities. The complexities are: a). Severe resource contention for shared on-chip resources such as last level lache (LLC), interconnect (For example: Intel's Quick Path Interconnect, AMD's Hyper Transport), and DRAM controllers; b). Non-uniform memory access (NUMA) where the time for memory access between a core and main memory is not uniform and where main memory is distributed between locality domains or groups called NUMA nodes; c). Dynamic power management (DPM) of multiple power domains (CPU sockets, DRAM).

The inherent complexities in these CPUs pose difficult challenges to solution methods solving the single- and bi-objective optimization problems of multithreaded data-parallel applications for performance and energy on such platforms. Recent researches demonstrate that performance and energy profiles of data-parallel applications executed on modern multicore CPUs to

manifest drastic variations and these variations are the principal cause for low average performance.

This thesis studies the influence of three-dimensional decision variable space on single- and bi-objective optimizations of applications for performance and energy on multicore CPUs. The three decision variables are: a). The number of identical multithreaded kernels (threadgroups) involved in the parallel execution of an application; b). The number of threads in each threadgroup; and c). The workload distribution between the threadgroups.

The thesis demonstrates the *workload distribution* to be an important decision variable that can no longer be ignored in performance optimization problem of data-parallel applications on modern multicore CPUs. The solution methods using *workload distribution* as a decision variable are proposed in this thesis. These methods employ model-based parallel computing technique and use load-imbalancing data partitioning.

The thesis proposes methods for single-objective optimization for performance and energy on modern multicore CPUs that use the *threadgroups* and the *number of threads* in each threadgroup as decision variables. The workload distribution is fixed so that a given workload is always partitioned equally between the threadgroups.

One of the key findings of this thesis is that energy proportionality of computing does not hold true for multicore CPUs thereby affording an opportunity for bi-objective optimization for performance and energy. Based on this finding, this thesis proposes the first application-level method for solving the bi-objective optimization problem for performance and energy on a single multicore CPU. The method uses two decision variables, the number of identical multithreaded kernels (*threadgroups*) executing the application in

parallel and the *number of threads* in each threadgroup. The workload distribution is not a decision variable. It is fixed so that a given workload is always partitioned equally between the threadgroups.

Finally, this thesis proposes a predictive dynamic energy model based on a non-negative linear regression and employing performance monitoring counters (PMCs) as predictor variables to explain the Pareto-optimal solutions determined by the solution method proposed in this thesis for modern multicore CPUs.

Contents

Chapter 1

Introduction

High-performance computing (HPC) has received lots of attention from the science and business industry with the advent of multi-core and cloud computing. HPC is essential in physical simulations, weather forecasting, quantum mechanics, data analytics, artificial intelligence (AI), etc., where large-scale problems need to be solved requiring massive computations to be performed. HPC gathers together a wide range of modern homogeneous and heterogeneous platforms (supercomputers [1], Grid'5000 [2]) to deliver higher performance. Multicore CPUs are the mandrel of such system, and any optimization focusing on the objectives such as performance and energy consumption of multicore CPUs, will optimize these objectives for the overall system.

Reviewing the history of computers, for more than three decades prior to mid-2000s called the single-core era, performance doubled every 18 months due to Moore's law [3] and Dennard scaling ([4]). Moore's law states that the number of transistors per square inch on integrated circuits doubles every year since the integrated circuit was invented. Dennard scaling is a scaling model whereby the power density of a transistor based processor of a unit area remains constant due to voltage and current scaling down with the length of the transistor. However, since 2004, designers of processors started facing physical constraints of the integrated circuit containing the transistors. Both power dissipation and power density trends have essentially required

designers to remain within a particular power budget and density requirements. All these limitations, associated with voltage supply scaling, threshold scaling, and clock frequency scaling, along with design complexity, forced companies to look for an alternative to the single core paradigm [5]. Thus, in 2005, AMD released their first dual-core processor (Athlon 64 X2) and from that time onwards, microprocessor architecture entered multicore era. Multicore processors integrate many cores into one chip to overcome the physical constraints of uniprocessor architecture and deliver high computing power with a single chip.

Modern parallel platforms are composed of tightly integrated multicore CPUs with a hierarchical arrangement of cores into sockets with multi-level cache hierarchy. This tight integration has resulted in the cores contending for various shared on-chip resources such as Last Level Cache (LLC) and interconnect (For example: Intel's Quick Path Interconnect [6], AMD's Hyper Transport [7]), leading to resource contention and non-uniform memory access (NUMA). NUMA happens where the time for memory access between a core and main memory is not uniform and where main memory is distributed between locality domains or groups called NUMA nodes. Figure 1.1 shows the most general architecture of multicore CPUs. It comprises of two sockets (NUMA node 0 and NUMA node 1) with four physical cores each.

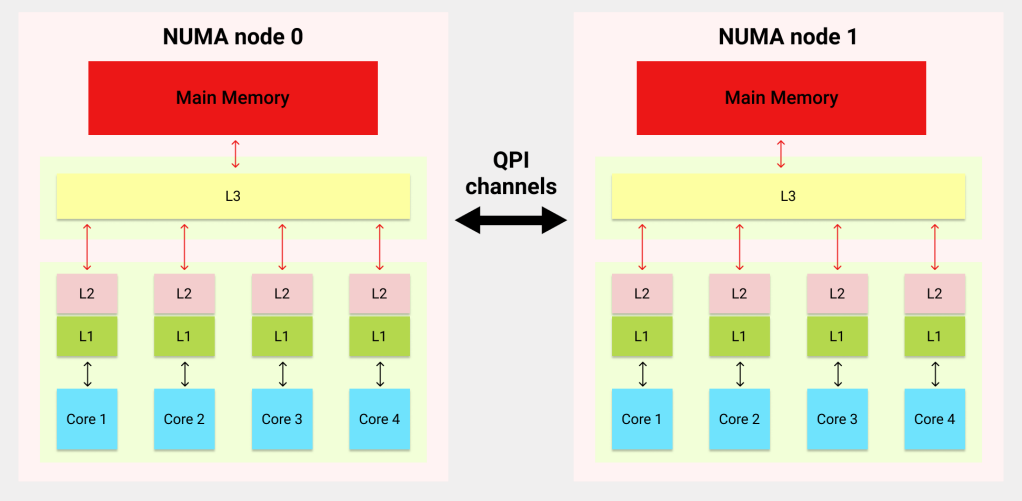

Figure 1.1: The most general architecture of processors nowadays.

Each core has its own L1 and L2 caches. All the cores in a socket share the last level cache (L3). The time taken to access a data item depends on where it is in the multi-level cache and memory hierarchy. The closer the memory to the core, the less the access time. For example: time to access data in the L1 cache is considerably less than that for L2 and L3 caches. Time is longer for access to the memory of the neighbour NUMA node since in this case the slow on-chip interconnect is used. Furthermore, all cores share the same last level cache (L3) leading to severe resource contention for it between threads. Efficient portable parallel programming on platforms composed of such multicore CPUs must address daunting challenges posed by the inherent complexities.

1.1 Motivation Behind This Thesis

To explain the motivation of this thesis, the author elucidates the challenges posed by the inherent complexities in multicore CPU platforms to solving single-objective optimization of data-parallel applications for performance and energy, and bi-objective optimization for performance and energy on such platforms. The challenges are illustrated using two well-known highly optimized scientific kernels, matrix-matrix multiplication (DGEMM) and 2D fast Fourier transform (2D-FFT).

1.1.1 Performance Optimization on Modern Multicore CPUs

This section presents the challenges posed to performance optimization on modern multicore CPUs. This is followed by explanation why the state-of-the-art dominant technique of load balancing fails to address the challenges. Finally, it proposes solution methods to address the challenges.

Figure 1.2 shows the performance profile of multithreaded matrix-matrix multiplication employing DGEMM routine provided by the Intel Math Kernel Library v.2017. The application computes the matrix product ($C = \alpha \times A \times B + \beta \times C$) of two dense square matrices A and B of size $N \times N$. It is executed on a modern Intel Haswell server consisting of 36 cores. The number of threads

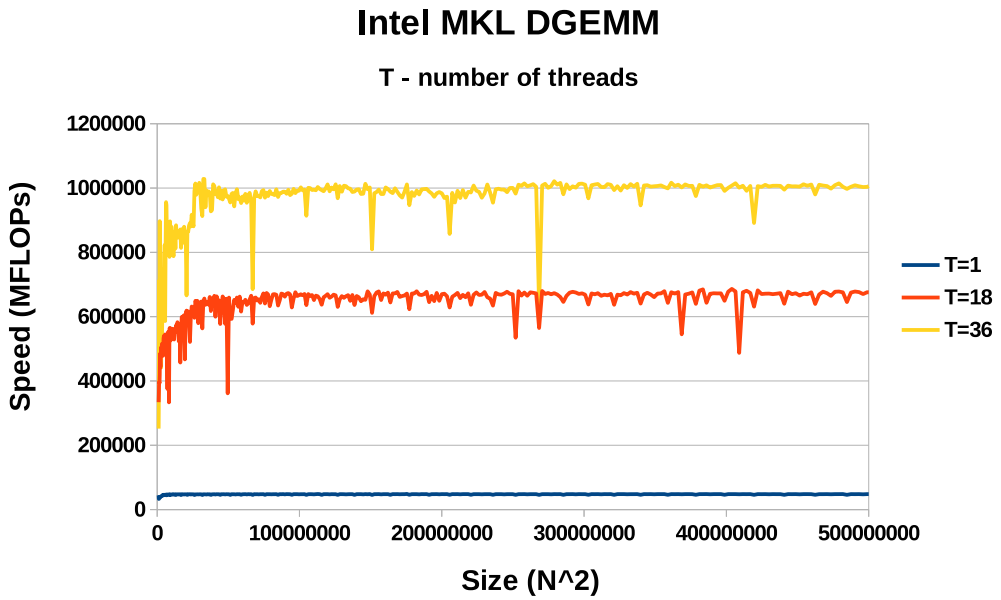

Figure 1.2: Speed function of IMKL DGEMM application executing varying number of threads (T) on the Intel Haswell server.

employed during the execution of the DGEMM routine is configurable.

The crucial observation is that for one thread the profile is smooth. However, drastic variations in the performance can be observed with increasing number of threads. The variation is related to the difference of speeds between two subsequent local minima (s_1) and maxima (s_2) and is defined as: $variation(\%) = \frac{|s_1 - s_2|}{\min(s_1, s_2)} \times 100$. The maximum width of variations with 36 threads is more than 40%. There are several sizes where the width of variations reaches more than 20%.

Figure 1.3 illustrates the performance profile of 2D-FFT offered by the same Intel Math Kernel Library v.2017. The 2D-FFT application is executed with 36 threads on the same Intel Haswell server. It computes the 2D-DFT of the signal matrix of size $N \times N$. The number of threads employed during the execution of the 2D-FFT routine is also configurable. The variations happen for the whole range of problem sizes. The maximum width of variations is around 89%. The detailed study of performance profiles of the 2D-FFT application using three vendor packages, FFTW-2.1.5, FFTW-3.3.7 and IMKL

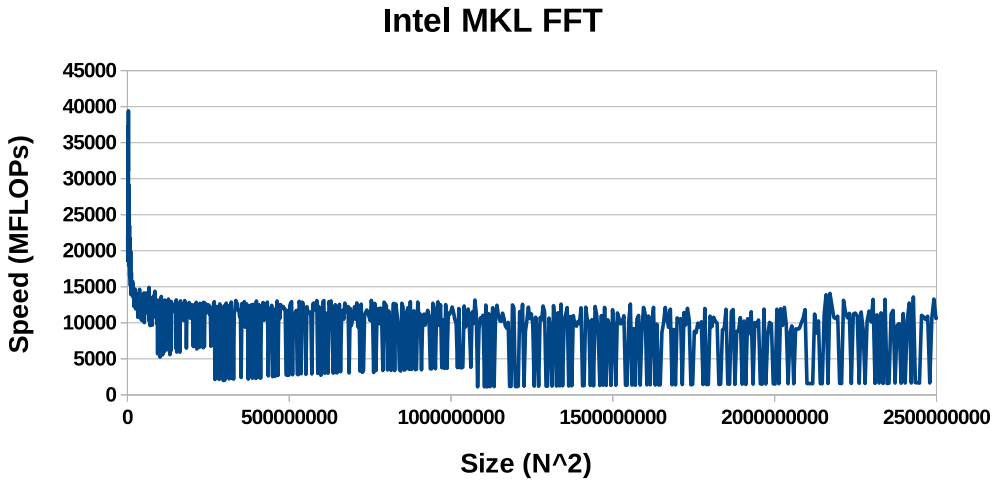

Figure 1.3: Speed function of IMKL FFT application executing with 36 threads on the Intel Haswell server.

FFT, can be found in the Appendix **??**, where also is show that the FFT routines in the packages demonstrate low average performance due to these variations.

To make sure the experimental results are reliable and not noise, a statistical methodology described in Appendix **??** is used. Briefly, for every data point in the functions, the automation software executes the application repeatedly until the sample mean lies in the 95% confidence interval with precision of 0.025 (2.5%).

The variations cannot be explained by the constant and stochastic fluctuations due to OS activity or a workload executing in a node in common networks of computers. In such networks, a node is persistently performing minor routine computations and communications by being an integral part of the network. Examples of such routine applications include e-mail clients, browsers, text editors, audio applications, etc. As a result, the node will experience constant and stochastic fluctuations in the workload. This changing transient load will cause a fluctuation in the speed of the node in the sense that the speed will vary for different runs of the same workload. One way to represent these inherent fluctuations in the speed is to use a speed

band rather than a speed function. The width of the band characterizes the level of fluctuation in the speed due to changes in load over time [8], [9], [10]. For a node with uniprocessors, the width of the band has been shown to decrease as the problem size increases. For a node with a very high level of network integration, typical widths of the speed bands were observed to be around 40% for small problem sizes and narrowing down to 3% for large problem sizes. Therefore, as the problem size increases, the width of the speed band is observed to decrease. Therefore, for long running applications, one would observe the width to become quite narrow (3%). However, this is not the case for variations in the presented graphs. Hence, these variations are consequences of the inherent complexities posed by the tight integration which has resulted in the cores contending for various shared on-chip resources such as Last Level Cache (LLC) and interconnect (NUMA). They pose a daunting challenge to performance optimization of multi-threaded applications on modern multicore CPUs.

Load balancing is a well known and still the dominant technique for performance optimization of scientific applications on parallel platforms. Load balancing algorithms can be classified as static or dynamic. Static algorithms (for example, those based on data partitioning) [11], [12] require a priori information about the parallel application and platform. Dynamic algorithms (such as task scheduling and work stealing) [13]–[15] balance the load by moving finegrained tasks between processors during the calculation. Dynamic algorithms do not require a priori information about execution but may incur significant communication overhead due to data migration.

The most advanced load balancing algorithms use functional performance models (FPMs), which are application-specific and represent the speed of a processor by continuous function of problem size but satisfying some assumptions on its shape [9]. These FPMs capture accurately the real-life behavior of applications executing on nodes consisting of uniprocessors (single-core CPUs). The assumptions require them to be smooth enough in order to guarantee that optimal solutions minimizing the computation time are always load balanced. However, as can be seen from the figures 1.2 and 1.3, due to complex nodal architectures with a highly hierarchical arrangement

and tight integration of cores the shape of the performance profiles of real scientific applications on the modern multicore CPUs is not smooth and may deviate significantly from the shapes that allowed traditional and state-of-the-art load balancing algorithms to find optimal solutions.

Lastovetsky et al. [16], [17] study the variations in performance profile for a real-life data-parallel scientific application, Multidimensional Positive Definite Advection Transport Algorithm (MPDATA), on a Xeon Phi co-processor. They geometrically prove the limitations of the FPM-based load balancing algorithms to modern performance profiles executed on multicore CPUs. Based on FPMs, the authors propose a novel optimization technique that distributes workload among cores unequally but gaining better performance in comparison with traditional load balancing. Furthermore, Lastovetsky et al. in [18] propose new model-based methods and algorithms for minimization of time and energy of computations for the most general shapes of performance and energy profiles of data parallel applications observed on the modern homogeneous multicore clusters.

The methods [16]–[18] show that workload distribution has become an important decision variable for performance optimization on modern multicore CPUs. The methods are, however, theoretical works and target homogeneous clusters of multicore CPUs and not a single multicore CPU.

There are three solution approaches that can be employed to remove the performance variations.

Manual code optimization is typically the first approach adopted to improve the performance of an application. The roofline model [19] is used to visually depict the trend of performance gains accrued from code tuning towards the theoretical peak performance of a multicore processor. Using this model, the highly optimized scientific applications such as Intel Math Kernel Library (IMKL) (BLAS, FFT) consistently demonstrate the superior performance of their codes for new platforms.

However, manual code optimization is a time-consuming process and programmers who can program such techniques are rare because they should be experts in both hardware and software domain. This approach involves different techniques such as loop transformation, use of pointers,

use of SIMD registers, blocking etc. [20], [21], [22], to avoid the unprofitable use of cache resources and improve CPU utilization that in turn leads to higher performance. For this, data from performance monitoring counters (PMCs) is required, that demands additional knowledge about hardware specific architecture. PMCs are special-purpose registers provided in modern microprocessors to store the counts of software and hardware activities. We will use the acronym PMCs to refer to software events, which are pure kernel-level counters such as *page-faults*, *context-switches*, etc. as well as micro-architectural events originating from the processor and its performance monitoring unit called the hardware events such as *cache-misses*, *branch-instructions*, etc.

Besides, PMCs in some cases are not reliable based on additivity test proposed in [23]. Moreover, such efficient tuning for one architecture can be inefficient for the other that damages code portability. Some vendors such as Intel do not disclose the source code of their applications which makes code modification impossible at the kernel level.

The second approach constructs solutions for an input workload size by employing solutions to larger workload sizes with better performance. From the figures 1.2 and 1.3 can be seen that two subsequential workload sizes have different performance where sometimes a larger problem size has better performance. The basic idea is to increase the input workload size (by padding, for example) to a bigger workload size with better performance, solve the padded workload size, and use its solution to construct the solution for the input workload size. This is a portable approach.

Finally, the third approach is optimization using model-based parallel computing method [16]–[18]. The key idea behind this approach is to design and implement a parallel version of the application that can be executed using identical abstract processors named threadgroups in parallel. The performances of the threadgroups are represented by realistic and accurate performance models of computation. The models are input to a data partitioning algorithm to determine the optimal workload distribution maximizing the performance during the parallel execution of the application. The main advantages of this approach are:

- It is portable when the performance models of computation used in the data partitioning algorithms do not use architecture-specific parameters.
- It does not require source code modification of the optimized package.
- The programming effort is less time-consuming, which is to distribute the workload between identical already optimized and well-tested multithreaded routines (abstract processors) and execute them in parallel.

This thesis proposes novel single-objective optimization methods specifically designed for performance optimization of 2D fast Fourier transform based on FFTW and IMKL (PFFT) and dense matrix-matrix multiplication written using OpenBLAS DGEMM and IMKL (PMM).

The solution methods employ workload distribution as the decision variable and are based on model-based parallel computing method using load-imbalancing data partitioning technique. The technique determines optimal solutions (workload distributions) that may not load-balance the application in terms of execution time. The methods take as inputs, the discrete functions of the performance of the processors against problem size. Based on the experiments conducted on a dual-socket Intel Haswell CPU consisting of 36 physical cores, the average and maximum speedups observed for *PFFT* using *FFTW-3.3.7* are 2.3x and 9.4x and the average and maximum speedups observed using *IMKL FFT* are 1.4x and 5.9x. The average and maximum speedups observed for *PMM* using *OpenBLAS DGEMM* are 1.2x and 1.4x and the average and maximum speedups observed using *IMKL DGEMM* are 1.1x and 1.3x.

Then an application-level method, SOPPETG, for solving performance optimization problem on a single multicore CPU is proposed. The method uses two decision variables, the number of identical multithreaded kernels (threadgroups) executing the application in parallel and the number of threads in each threadgroup. The workload distribution is not a decision variable. It is fixed so that a given workload is always partitioned equally between the threadgroups. Based on the experiments conducted on a single-socket Intel

Skylake CPU consisting of 22 physical cores, the average and maximum performance improvements of SOPPETG using *OpenBLAS DGEMM* are 7% and 26.3% and the average and maximum performance improvements using *IMKL DGEMM* are 4.1% and 6.5%. The average and maximum performance improvements of SOPPETG using *IMKL FFT* are 7% and 13% and using *FFTW-3.3.7* are 25% and 51% respectively.

On a dual-socket Intel Haswell CPU consisting of 36 physical cores, the average and maximum performance improvements of SOPPETG using *OpenBLAS DGEMM* are 19% and 31.7% and the average and maximum performance improvements using *IMKL DGEMM* are 7% and 42.1%. The average and maximum performance improvements of SOPPETG using *FFTW-3.3.7* are 85% and 90%.

1.1.2 Energy Optimization on Modern Multicore CPUs

Reducing energy consumption is of paramount concern to the HPC community since its pervasiveness in data centers and cloud computing infrastructures. Energy in HPC is now an environment concern not only because of the maintenance cost of HPC systems but also of high carbon footprint which affects environmental sustainability as modern data centers already can rival cities in power consumption. This was not an issue in the past since until now we have followed Moore's Law enhancements in photolithography techniques which are proportional reductions in dynamic power consumption per transistor and consequent improvements in clock frequency at the same level of power dissipation. However, below 90 nm, the static power dissipation can be greater than the dynamic power dissipation. This effect summons clock frequency freezing in order to stay within thermal power emission limits [24].

The optimization of energy consumption of multicore CPUs is more complex than that of a single- or dual-core CPUs. The new complexities such as tight integration with severe contention on shared resources (Last level caches (LLC), main memory, PCI-E links, etc.) and NUMA pose tremendous challenges to the energy optimization of data-parallel applications on modern

multicore CPUs.

In contrast to single-core optimization, where energy profiles follow the fully polynomial-time scheme for task partitioning, i.e. energy consumption with a higher workload is larger than that with a lower workload [25], the energy profiles of real scientific applications executed on modern multicore CPUs demonstrate highly non-linear relationship between workload size and energy consumption.

As an example, figure **??** depicts the dynamic energy consumption profile of 2D-FFT employing IMKL FFT on the Intel Xeon Platinum server consisting of 56 cores. The dynamic energy consumption is measured with Yokogawa WT310 power meter. It can be seen that the graph is highly non-linear. The maximum width of variations can be up to 73%. It represents the maximum amount of energy savings possible.

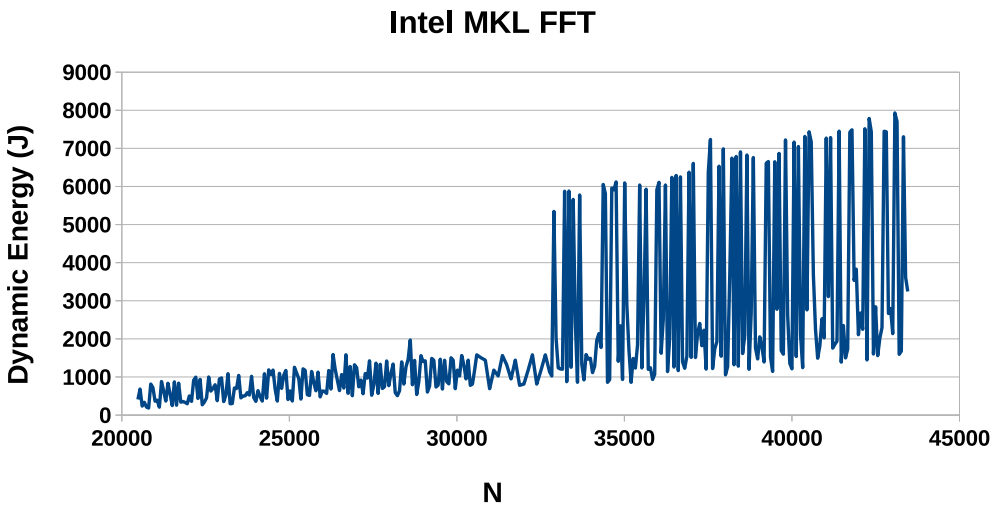

Figure 1.4: Dynamic Energy Consumption of IMKL FFT application executing with 56 cores on the Intel Xeon Platinum server.

The research works [26], [27] propose model-based data partitioning methods to minimize the total dynamic energy consumption during the execution of a data-parallel application on homogeneous clusters of multicore CPUs. They take as input discrete dynamic energy functions with no shape assumptions (for example, the discrete profile in the Figure 1.4), which

accurately and realistically account for resource contention and NUMA inherent in modern multicore CPU platforms. The research works are theoretical demonstrating energy improvements based on simulations of clusters of homogeneous nodes containing multicore CPUs.

This thesis proposes an application-level method, SOPPETG, for solving energy optimization problem on a single multicore CPU. The method uses two decision variables, the number of identical multithreaded kernels (threadgroups) executing the application in parallel and the number of threads in each threadgroup. The workload distribution is not a decision variable. It is fixed so that a given workload is always partitioned equally between the threadgroups. Based on the experiments conducted on a single-socket Intel Skylake CPU consisting of 22 physical cores, the average and maximum energy savings of SOPPETG using *OpenBLAS DGEMM* are 7.9% and 30% and the average and maximum energy savings using *IMKL DGEMM* are 35.7% and 67%. The average and maximum energy savings of SOPPETG using *FFTW-3.3.7* are 30% and 63%.

On a dual-socket Intel Haswell CPU consisting of 24 physical cores, the average and maximum energy savings of SOPPETG using *OpenBLAS DGEMM* are 10% and 24.5% and the average and maximum energy savings using *IMKL DGEMM* are 13% and 67%. The average and maximum energy savings of SOPPETG using *FFTW-3.3.7* on a dual-socket Intel Skylake CPU consisting of 56 cores are 23% and 43%.

1.1.3 Bi-Objective Optimization for Performance and Energy

Energy proportionality is the key design goal pursued by architects of modern multicore CPU platforms [28]. One of its implications is that optimization of an application for performance will also optimize it for energy. Modern multicore CPUs however have several inherent complexities, which are: a) Severe resource contention due to tight integration of tens of cores organized in multiple sockets with multi-level cache hierarchy and contending for shared on-chip resources such as last level lache (LLC), interconnect (For example:

Intel's Quick Path Interconnect, AMD's Hyper Transport), and DRAM controllers; b) Non-uniform memory access (NUMA) where the time for memory access between a core and main memory is not uniform and where main memory is distributed between locality domains or groups called NUMA nodes; and c) Dynamic power management (DPM) of multiple power domains (CPU sockets, DRAM). This thesis shows that due to these complexities, energy proportionality does not hold true for multicore CPUs. This finding creates the opportunity for bi-objective optimization of applications for performance and energy.

Solution methods solving the bi-objective optimization problem for performance and energy BOPPE can be broadly classified into *system-level* and *application-level* categories. System-level methods aim to optimize performance and energy of the environment where the applications are executed. The methods employ application-agnostic models and hardware parameters as decision variables. They are principally deployed at operating system (OS) level and therefore require changes to either the OS or the hardware. The key decision variable employed is Dynamic Voltage and Frequency Scaling (DVFS).

In the second category, solution methods optimize applications rather than the executing environment. The methods use application-level decision variables and predictive models for performance and energy consumption of applications to solve BOPPE. The dominant decision variables include the number of threads, loop tile size, workload distribution, etc. Following the principle of energy proportionality, a dominant class of such solution methods aim to achieve optimal energy reduction by optimizing for performance alone. Definitive examples are scientific routines offered by vendor-specific software packages that are extensively optimized for performance. For example, Intel Math Kernel Library [29] provides extensively optimized multithreaded basic linear algebra subprograms (BLAS) and 1D, 2D, and 3D fast Fourier transform (FFT) routines for Intel processors. Open source packages such as [30]–[32] offer the same interface functions but contain portable optimizations and may exhibit better average performance than a heavily optimized vendor package [33], [34]. The optimized routines in these software packages allow

employment of one key decision variable, which is the number of threads. A given workload is load-balanced between the threads.

The works [26], [27], [35] propose model-based data partitioning methods that take as input discrete performance and dynamic energy functions with no shape assumptions, which accurately and realistically account for resource contention and NUMA inherent in modern multicore CPU platforms. Using a simulation of the execution of a data-parallel matrix multiplication application based on OpenBLAS DGEMM on a homogeneous cluster of multicore CPUs, [26] show that optimizing for performance alone results in average and maximum dynamic energy reductions of 24% and 68%, but optimizing for dynamic energy alone results in performance degradations of 95% and 100%. For a 2D fast Fourier transform application based on FFTW, the average and maximum dynamic energy reductions are 29% and 55% and the average and maximum performance degradations are both 100%. Research work [35] proposes a solution method called *ALEPH* to solve BOPPE on homogeneous clusters of modern multicore CPUs. ALEPH is shown to determine a diverse set of globally Pareto-optimal solutions whereas existing solution methods give only one solution when the problem size and number of processors are fixed. The methods target homogeneous HPC platforms. Khaleghzadeh et al. [36] propose a solution method solving the bi-objective optimization problem on heterogeneous processors. The authors prove that for an arbitrary number of processors with linear execution time and dynamic energy functions, the globally Pareto-optimal front is linear and contains an infinite number of solutions out of which one solution is load balanced while the rest are load imbalanced. A data partitioning algorithm is presented that takes as an input discrete performance and dynamic energy functions with no shape assumptions. The research works [26], [27], [35], [36] are theoretical demonstrating performance and energy improvements based on simulations of clusters of homogeneous and heterogeneous nodes.

All these works done on bi-objective optimization for performance and energy do not consider the optimization on a single multicore CPU. Furthermore, the works [26], [27], [35], [36] are theoretical and use only workload distribution as a decision variable. However, one of the findings of

this thesis is that modern multicore CPUs are not energy proportional and a trade-off between energy and performance can be found on such platforms. This finding opens an opportunity for bi-objective optimization for performance and energy on a single multicore CPU and makes it meaningful. To the best of author's knowledge, this is the first work studying bi-objective optimization for performance and energy consumption on a single multicore CPU.

This thesis studies the influence of three-dimensional decision variable space on bi-objective optimization of applications for performance and energy on multicore CPUs. The three decision variables are: a). The number of identical multithreaded kernels (threadgroups) involved in the parallel execution of an application; b). The number of threads in each threadgroup; and c). The workload distribution between the threadgroups. The author focuses exclusively on the first two decision variables in this work. The number of possible workload distributions increases exponentially with increasing number of threadgroups employed in the execution of a data-parallel application and it would require employment of threadgroup-specific performance and energy models to reduce the complexity. It is a subject of future work.

The thesis proposes the first application-level method for bi-objective optimization of multithreaded data-parallel applications on a single multicore CPU for performance and energy. The method uses two decision variables, the number of identical multithreaded kernels (threadgroups) executing the application in parallel and the number of threads in each threadgroup. The workload distribution is not a decision variable. It is fixed so that a given workload is always partitioned equally between the threadgroups. The method allows full reuse of highly optimized scientific codes and does not require any changes to hardware or OS.

Based on the experiments conducted on a dual-socket Intel Skylake CPU consisting of 56 cores, it was observed that the number of Pareto optimal solutions can be up to 11 for *FFTW-3.3.7.*. Figure 1.5 shows these solutions for problem size $m = n = 30464$. One can observe, choosing the best configuration for performance (g,t)=(1,96), increases the dynamic energy

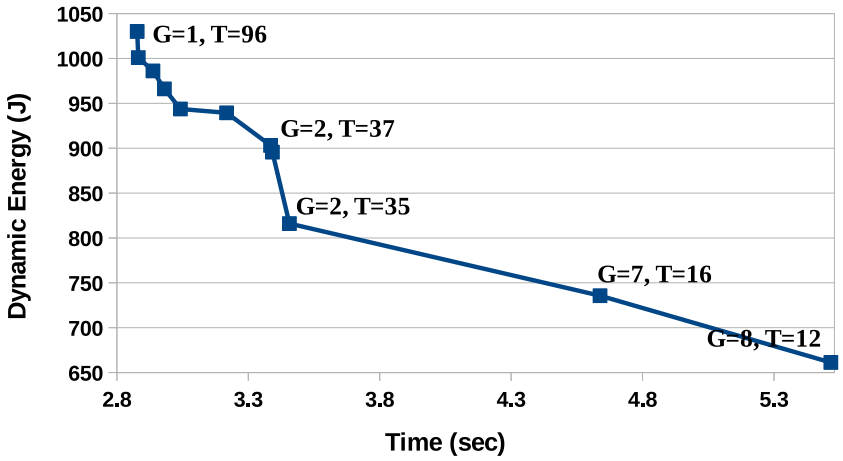

Figure 1.5: Pareto frontier of FFTW PFFTTG application on HCLServer4 (S4) for workload size $m = n = 30464$.

consumption by 35% in comparison with the optimal configuration for energy (8,12), and choosing the optimal configuration for energy (8,12), degrades the performance by 49% in comparison with the optimal configuration for performance (1,96). The average number of globally Pareto-optimal solutions for *FFTW-3.3.7* is 3. On a single-socket Intel Skylake CPU consisting of 22 physical cores, the average and the maximum number of globally Pareto-optimal solutions for *IMKL DGEMM* and *IMKL FFT* are (2.3,3) and (2.6,3).

Finally, this thesis proposes a predictive dynamic energy model based on non-negative linear regression and employing performance monitoring counters (PMCs) as predictor variables to explain the Pareto-optimal solutions determined by solution method proposed in this thesis for multicore CPUs.

1.2 Thesis Contributions

The main contributions of this thesis are the following:

1. Demonstration of the challenges posed by inherent complexities in modern multicore CPUs such as severe resource contention and

NUMA to the performance of multi-threaded data-parallel applications executing on such platforms.

2. Studying the performance profiles of multithreaded 2D FFT and matrix-matrix multiplication provided in highly optimized packages, FFTW-3.3.7, IMKL FFT, OpenBALS DGEMM and IMKL DGEMM on a modern Intel Haswell multicore processor consisting of thirty-six cores. It is shown that all routines demonstrate drastic performance variations and that their average performances therefore are considerably lower than their peak performances.

3. Three novel optimization methods specifically designed for optimization of 2D-FFTW, 2D-FFT-IMKL, OpenBLAS-DGEMM and IMKL-DGEMM for performance. The methods employ workload distribution as the decision variable and are based on model-based parallel computing method using load-imbalancing data partitioning technique. The technique determines optimal solutions (workload distributions) that may not load-balance the application in terms of execution time.

4. Application-level methods for single-objective optimization of multithreaded data-parallel applications for performance and energy. The method uses two decision variables, the number of identical multithreaded kernels (threadgroups) and the number of threads in each threadgroup.

5. Detection and demonstration of that the energy proportionality does not hold true for multicore CPUs thereby affording an opportunity for bi-objective optimization for performance and energy.

6. The first application-level method for bi-objective optimization of multithreaded data-parallel applications for performance and energy. The method uses two decision variables, the number of identical multithreaded kernels (threadgroups) and the number of threads in each threadgroup. The method is demonstrated using four highly optimized data-parallel applications. It is shown that the proposed

method determines good numbers of globally Pareto-optimal configurations of the applications allowing for a better balance between performance and energy consumption.

7. Predictive dynamic energy model based on linear regression and employing PMCs as predictor variables to explain the Pareto-optimal solutions determined by the method proposed in this thesis for dual-socket multicore CPUs.

1.3 Thesis Structure

The rest of the thesis is organized as follows: chapter 2 covers the review of state-of-the-art methods of single-objective optimization for performance and energy, bi-objective optimization for performance and energy on modern multicore CPUs, and performance and energy models of computation. Chapter 3 presents novel methods for single-objective optimization performance and energy using three decision variables - workload distribution, the number of threadgroups and the number of threads in each threadgroup. Chapter 4 proposes bi-objective optimization for performance and energy on modern multicore CPUs using the number of threadgroups and the number of threads per threadgroup as decision variables. The conclusion of this thesis is in chapter 5.

Bibliography

[1] Intel Corporation, *Top500 list*, July, 2019. [Online]. Available: https://www.top500.org/lists/2019/06/ (cit. on p. 2).

[2] Scientific interest group (GIS), *Grid5000:home*, 2019. [Online]. Available: https://www.grid5000.fr/w/Grid5000:Home (cit. on p. 2).

[3] G. E. Moore, "Gramming more components onto integrated circuits," *Electronics*, vol. 38, p. 8, 1965 (cit. on p. 2).

[4] Dennard, *Dennard scaling*, 1974. [Online]. Available: https://en.wikipedia.org/wiki/Dennard_scaling (cit. on p. 2).

[5] J. Parkhurst, J. Darringer, and B. Grundmann, "From single core to multi-core: Preparing for a new exponential," in *Proceedings of the 2006 IEEE/ACM international conference on Computer-aided design*, ACM, 2006, pp. 67–72 (cit. on p. 3).

[6] QPI. (2008). Intel quickpath interconnect, [Online]. Available: https://en.wikipedia.org/wiki/Intel_QuickPath_Interconnect (cit. on p. 3).

[7] AMDHT. (2001). Hypertransport, [Online]. Available: https://en.wikipedia.org/wiki/HyperTransport (cit. on p. 3).

[8] A. L. Lastovetsky and R. Reddy, "Data partitioning with a realistic performance model of networks of heterogeneous computers," in *Parallel and Distributed Processing Symposium, 2004. Proceedings. 18th International*, IEEE, 2004, p. 104 (cit. on p. 7).

[9] A. Lastovetsky and R. Reddy, “Data partitioning with a functional performance model of heterogeneous processors,” *International Journal of High Performance Computing Applications*, vol. 21, no. 1, pp. 76–90, 2007 (cit. on p. 7).

[10] A. Lastovetsky and J. Twamley, “Towards a realistic performance model for networks of heterogeneous computers,” in *High Performance Computational Science and Engineering*, Springer, 2005, pp. 39–57 (cit. on p. 7).

[11] A. Lastovetsky and R. Reddy, “Data partitioning with a functional performance model of heterogeneous processors,” *The International Journal of High Performance Computing Applications*, vol. 21, no. 1, pp. 76–90, 2007 (cit. on p. 7).

[12] Y. Ogata, T. Endo, N. Maruyama, and S. Matsuoka, “An efficient, model-based CPU-GPU heterogeneous FFT library,” in *Parallel and Distributed Processing, 2008. IPDPS 2008. IEEE International Symposium on*, IEEE, 2008, pp. 1–10 (cit. on p. 7).

[13] M. D. Linderman, J. D. Collins, H. Wang, and T. H. Meng, “Merge: A programming model for heterogeneous multi-core systems,” in *ACM SIGOPS operating systems review*, ACM, vol. 42, 2008, pp. 287–296 (cit. on p. 7).

[14] G. Quintana-Ortí, F. D. Igual, E. S. Quintana-Ortí, and R. A. Van de Geijn, “Solving dense linear systems on platforms with multiple hardware accelerators,” in *ACM Sigplan Notices*, ACM, vol. 44, 2009, pp. 121–130 (cit. on p. 7).

[15] C. Augonnet, S. Thibault, and R. Namyst, “Automatic calibration of performance models on heterogeneous multicore architectures,” in *European Conference on Parallel Processing*, Springer, 2009, pp. 56–65 (cit. on p. 7).

[16] A. L. Lastovetsky, L. Szustak, and R. Wyrzykowski, “Model-based optimization of MPDATA on Intel Xeon Phi through load imbalancing,” *CoRR*, vol. abs/1507.01265, 2015 (cit. on pp. 8, 9).

[17] A. Lastovetsky, L. Szustak, and R. Wyrzykowski, "Model-based optimization of EULAG kernel on Intel Xeon Phi through load imbalancing," *IEEE Transactions on Parallel and Distributed Systems*, (cit. on pp. 8, 9).

[18] A. Lastovetsky and R. R. Manumachu, "New model-based methods and algorithms for performance and energy optimization of data parallel applications on homogeneous multicore clusters," *IEEE Transactions on Parallel and Distributed Systems*, vol. 28, no. 4, pp. 1119–1133, 2017 (cit. on pp. 8, 9).

[19] S. WiLLiAmS, A. WAteRmAn, and D. PAtteRSon, "The roofline model offers insight on how to improve the performance of software and hardware.," *CommunicAtionS of the Acm*, vol. 52, no. 4, 2009 (cit. on p. 8).

[20] S. Ghose and J. Tse, "Cs 5220: Project 1 tuning the matrix multiply algorithm," (cit. on p. 9).

[21] K. Datta, M. Murphy, V. Volkov, S. Williams, J. Carter, L. Oliker, D. Patterson, J. Shalf, and K. Yelick, "Stencil computation optimization and auto-tuning on state-of-the-art multicore architectures," in *Proceedings of the 2008 ACM/IEEE conference on Supercomputing*, IEEE Press, 2008, p. 4 (cit. on p. 9).

[22] J. Huang and R. A. Van de Geijn, "Blislab: A sandbox for optimizing gemm," *ArXiv preprint arXiv:1609.00076*, 2016 (cit. on p. 9).

[23] A. Shahid, M. Fahad, R. Reddy, and A. Lastovetsky, "Additivity: A selection criterion for performance events for reliable energy predictive modeling," *Supercomputing Frontiers and Innovations*, vol. 4, no. 4, pp. 50–65, 2017 (cit. on p. 9).

[24] S. Kamil, J. Shalf, and E. Strohmaier, "Power efficiency in high performance computing," in *2008 IEEE International Symposium on Parallel and Distributed Processing*, IEEE, 2008, pp. 1–8 (cit. on p. 11).

[25] H. Hajimiri, P. Mishra, and S. Bhunia, “Dynamic cache tuning for efficient memory based computing in multicore architectures,” in *2013 26th International Conference on VLSI Design and 2013 12th International Conference on Embedded Systems*, IEEE, 2013, pp. 49–54 (cit. on p. 12).

[26] A. Lastovetsky and R. Reddy, “New model-based methods and algorithms for performance and energy optimization of data parallel applications on homogeneous multicore clusters,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 28, no. 4, pp. 1119–1133, 2017 (cit. on pp. 12, 15).

[27] R. Reddy Manumachu and A. L. Lastovetsky, “Design of self-adaptable data parallel applications on multicore clusters automatically optimized for performance and energy through load distribution,” *Concurrency and Computation: Practice and Experience*, vol. 0, no. 0, e4958, (cit. on pp. 12, 15).

[28] L. A. Barroso and U. Hölzle, “The case for energy-proportional computing,” *Computer*, no. 12, pp. 33–37, 2007 (cit. on p. 13).

[29] Intel Corporation, *Intel MKL FFT - fast fourier transforms*, 2018. [Online]. Available: `https://software.intel.com/en-us/mkl/features/fft` (cit. on p. 14).

[30] OpenBLAS, *Openblas: An optimized BLAS library*, 2016. [Online]. Available: `http://www.openblas.net/` (cit. on p. 14).

[31] FFTW, *Fastest fourier transform in the west*, 2018. [Online]. Available: `http://www.fftw.org/` (cit. on p. 14).

[32] H. Khaleghzadeh, Z. Zhong, R. Reddy, and A. Lastovetsky., *Zzgemmooc: Multi-GPU out-of-core routines for dense matrix multiplization*, 2019. [Online]. Available: `https://git.ucd.ie/hcl/zzgemmooc.git` (cit. on p. 14).

[33] H. Khaleghzadeh, Z. Zhong, R. Reddy, and A. Lastovetsky, "Out-of-core implementation for accelerator kernels on heterogeneous clouds," *The Journal of Supercomputing*, vol. 74, no. 2, pp. 551–568, 2018 (cit. on p. 14).

[34] S. Khokhriakov, R. R. Manumachu, and A. Lastovetsky, "Performance optimization of multithreaded 2d fast fourier transform on multicore processors using load imbalancing parallel computing method," *IEEE Access*, vol. 6, pp. 64 202–64 224, 2018 (cit. on p. 14).

[35] R. R. Manumachu and A. Lastovetsky, "Bi-objective optimization of data-parallel applications on homogeneous multicore clusters for performance and energy," *IEEE Transactions on Computers*, vol. 67, no. 2, pp. 160–177, 2018 (cit. on p. 15).

[36] H. Khaleghzadeh, M. Fahad, A. Shahid, R. Reddy, and A. Lastovetsky, "Bi-objective optimization of data-parallel applications on heterogeneous hpc platforms for performance and energy through workload distribution," *CoRR*, vol. abs/1907.04080, 2019. arXiv: 1907.04080. [Online]. Available: http://arxiv.org/abs/1907.04080 (cit. on p. 15).

[37] N. Ding, S. Xu, Z. Song, B. Zhang, J. Li, and Z. Zheng, "Using hardware counter-based performance model to diagnose scaling issues of hpc applications," *Neural Computing and Applications*, vol. 31, no. 5, pp. 1563–1575, 2019.

[38] J.-P. Lehr, "Counting performance: Hardware performance counter and compiler instrumentation," *Informatik 2016*, 2016.

[39] B. Zhou, A. Gupta, R. Jahanshahi, M. Egele, and A. Joshi, "Hardware performance counters can detect malware: Myth or fact?" In *Proceedings of the 2018 on Asia Conference on Computer and Communications Security*, ACM, 2018, pp. 457–468.

[40] L. Uhsadel, A. Georges, and I. Verbauwhede, "Exploiting hardware performance counters," in *2008 5th Workshop on Fault Diagnosis and Tolerance in Cryptography*, IEEE, 2008, pp. 59–67.

[41] D. Dauwe, E. Jonardi, R. D. Friese, S. Pasricha, A. A. Maciejewski, D. A. Bader, and H. J. Siegel, "Hpc node performance and energy modeling with the co-location of applications," *The Journal of Supercomputing*, vol. 72, no. 12, pp. 4771–4809, 2016.

[42] Z. Zhong, V. Rychkov, and A. Lastovetsky, "Data partitioning on multicore and multi-gpu platforms using functional performance models," *IEEE Transactions on Computers*, vol. 64, no. 9, pp. 2506–2518, 2014.

[43] A. Lastovetsky and R. R. Manumachu, "New model-based methods and algorithms for performance and energy optimization of data parallel applications on homogeneous multicore clusters," *IEEE Transactions on Parallel and Distributed Systems*, vol. 28, no. 4, pp. 1119–1133, 2016.

[44] A. Lastovetsky, L. Szustak, and R. Wyrzykowski, "Model-based optimization of eulag kernel on intel xeon phi through load imbalancing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 28, no. 3, pp. 787–797, 2016.

[45] K. Goto and R. A. Geijn, "Anatomy of high-performance matrix multiplication," *ACM Transactions on Mathematical Software (TOMS)*, vol. 34, no. 3, p. 12, 2008.

[46] P. Balaprakash, J. Dongarra, T. Gamblin, M. Hall, J. K. Hollingsworth, B. Norris, and R. Vuduc, "Autotuning in high-performance computing applications," *Proceedings of the IEEE*, no. 99, pp. 1–16, 2018.

[47] J. Demmel, J. Dongarra, V. Eijkhout, E. Fuentes, A. Petitet, R. Vuduc, R. C. Whaley, and K. Yelick, "Self-adapting linear algebra algorithms and software," *Proceedings of the IEEE*, vol. 93, no. 2, pp. 293–312, 2005.

[48] S. Koliai, S. Zuckerman, E. Oseret, M. Ivascot, T. Moseley, D. Quang, and W. Jalby, "A balanced approach to application performance tuning," in *International Workshop on Languages and Compilers for Parallel Computing*, Springer, 2009, pp. 111–125.

[49] S. Williams, J. Carter, L. Oliker, J. Shalf, and K. Yelick, "Optimization of a lattice boltzmann computation on state-of-the-art multicore platforms," *Journal of Parallel and Distributed Computing*, vol. 69, no. 9, pp. 762–777, 2009.

[50] PeXL, *Maqao (modular assembly quality analyzer and optimizer)*, 2004. [Online]. Available: `http://www.maqao.org`.

[51] M. Hashimoto, M. Terai, T. Maeda, and K. Minami, "Cca/ebt: Code comprehension assistance tool for evidence-based performance tuning," 2018.

[52] M. Rajagopalan, B. T. Lewis, and T. A. Anderson, "Thread scheduling for multi-core platforms.," in *HotOS*, 2007.

[53] D. Chandra, F. Guo, S. Kim, and Y. Solihin, "Predicting inter-thread cache contention on a chip multi-processor architecture," in *11th International Symposium on High-Performance Computer Architecture*, IEEE, 2005, pp. 340–351.

[54] P. Radojkovic, V. Cakarevic, J. Verdu, A. Pajuelo, F. J. Cazorla, M. Nemirovsky, and M. Valero, "Thread assignment of multithreaded network applications in multicore/multithreaded processors," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 12, pp. 2513–2525, 2013.

[55] E. Ebrahimi, R. Miftakhutdinov, C. Fallin, C. J. Lee, J. A. Joao, O. Mutlu, and Y. N. Patt, "Parallel application memory scheduling," in *Proceedings of the 44th Annual IEEE/ACM International Symposium on Microarchitecture*, ACM, 2011, pp. 362–373.

[56] M. K. Jeong, D. H. Yoon, D. Sunwoo, M. Sullivan, I. Lee, and M. Erez, "Balancing dram locality and parallelism in shared memory cmp systems," in *IEEE International Symposium on High-Performance Comp Architecture*, IEEE, 2012, pp. 1–12.

[57] M. De Vuyst, R. Kumar, and D. M. Tullsen, “Exploiting unbalanced thread scheduling for energy and performance on a cmp of smt processors,” in *Proceedings 20th IEEE International Parallel & Distributed Processing Symposium*, IEEE, 2006, 10–pp.

[58] Y. Wen, Z. Wang, and M. F. O’boyle, “Smart multi-task scheduling for opencl programs on cpu/gpu heterogeneous platforms,” in *2014 21st International Conference on High Performance Computing (HiPC)*, IEEE, 2014, pp. 1–10.

[59] H. Khaleghzadeh, H. Deldari, R. Reddy, and A. Lastovetsky, “Hierarchical multicore thread mapping via estimation of remote communication,” *The Journal of Supercomputing*, vol. 74, no. 3, pp. 1321–1340, 2018.

[60] O. Franek, “A simple method for static load balancing of parallel fdtd codes,” in *Electromagnetics in Advanced Applications (ICEAA), 2016 International Conference on*, IEEE, 2016, pp. 587–590.

[61] R. L. Cariño and I. Banicescu, “Dynamic load balancing with adaptive factoring methods in scientific applications,” *The Journal of Supercomputing*, vol. 44, no. 1, pp. 41–63, 2008.

[62] J. A. Martínez, E. M. Garzón, A. Plaza, and I. García, “Automatic tuning of iterative computation on heterogeneous multiprocessors with ADITHE,” *J. Supercomput.*, vol. 58, no. 2, Nov. 2011.

[63] G. Cybenko, “Dynamic load balancing for distributed memory multiprocessors,” *Journal of parallel and distributed computing*, vol. 7, no. 2, pp. 279–301, 1989.

[64] J. M. Bahi, S. Contassot-Vivier, and R. Couturier, “Dynamic load balancing and efficient load estimators for asynchronous iterative algorithms,” *IEEE transactions on parallel and distributed systems*, vol. 16, no. 4, pp. 289–299, 2005.

[65] J. Bahi, R. Couturier, and F. Vernier, “Synchronous distributed load balancing on dynamic networks,” *Journal of Parallel and Distributed Computing*, vol. 65, no. 11, pp. 1397–1405, 2005.

[66] F. Liu, Y. Chen, and W. S. Wong, “An asynchronous load balancing scheme for multi-server systems,” in *Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON), IEEE Annual*, IEEE, 2016, pp. 1–7.

[67] P. K. Smolarkiewicz, “Multidimensional positive definite advection transport algorithm: An overview,” *International Journal for Numerical Methods in Fluids*, vol. 50, no. 10, pp. 1123–1144, 2006.

[68] A. Lastovetsky and R. Reddy, “New model-based methods and algorithms for performance and energy optimization of data parallel applications on homogeneous multicore clusters,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 28, no. 4, pp. 1119–1133, 2017.

[69] R. Reddy and A. Lastovetsky, “Bi-objective optimization of data-parallel applications on homogeneous multicore clusters for performance and energy,” *IEEE Transactions on Computers*, vol. 64, no. 2, pp. 160–177, 2017.

[70] H. Khaleghzadeh, R. R. Manumachu, and A. Lastovetsky, “A novel data-partitioning algorithm for performance optimization of data-parallel applications on heterogeneous HPC platforms,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 29, no. 10, pp. 2176–2190, 2018. DOI: 10.1109/TPDS.2018.2827055.

[71] L. Niu and G. Quan, “Reducing both dynamic and leakage energy consumption for hard real-time systems,” in *Proceedings of the 2004 international conference on Compilers, architecture, and synthesis for embedded systems*, ACM, 2004, pp. 140–148.

[72] S. Mittal, “A survey of techniques for improving energy efficiency in embedded computing systems,” *ArXiv preprint arXiv:1401.0765*, 2014.

[73] K. O’brien, I. Pietri, R. Reddy, A. Lastovetsky, and R. Sakellariou, “A survey of power and energy predictive models in hpc systems and applications,” *ACM Computing Surveys (CSUR)*, vol. 50, no. 3, p. 37, 2017.

[74] T. Heath, B. Diniz, E. V. Carrera, W. Meira Jr, and R. Bianchini, "Energy conservation in heterogeneous server clusters," in *Proceedings of the tenth ACM SIGPLAN symposium on Principles and practice of parallel programming*, ACM, 2005, pp. 186–195.

[75] D. Economou, S. Rivoire, C. Kozyrakis, and P. Ranganathan, "Full-system power analysis and modeling for server environments," International Symposium on Computer Architecture-IEEE, 2006.

[76] X. Feng, R. Ge, and K. W. Cameron, "Power and energy profiling of scientific applications on distributed systems," in *Parallel and Distributed Processing Symposium, 2005. Proceedings. 19th IEEE International*, IEEE, 2005, pp. 34–34.

[77] N. Vijaykrishnan, M. Kandemir, M. J. Irwin, H. S. Kim, and W. Ye, "Energy-driven integrated hardware-software optimizations using simplepower," *ACM SIGARCH Computer Architecture News*, vol. 28, no. 2, pp. 95–106, 2000.

[78] P. Gschwandtner, M. Knobloch, B. Mohr, D. Pleiter, and T. Fahringer, "Modeling cpu energy consumption of hpc applications on the ibm power7," in *2014 22nd Euromicro International Conference on Parallel, Distributed, and Network-Based Processing*, IEEE, 2014, pp. 536–543.

[79] R. Zamani and A. Afsahi, "Adaptive estimation and prediction of power and performance in high performance computing," *Computer Science-Research and Development*, vol. 25, no. 3-4, pp. 177–186, 2010.

[80] A. Shahid, M. Fahad, R. R. Manumachu, and A. Lastovetsky, "Improving the accuracy of energy predictive models for multicore cpus using additivity of performance monitoring counters," in *International Conference on Parallel Computing Technologies*, Springer, 2019, pp. 51–66.

[81] D. C. Snowdon, S. Ruocco, and G. Heiser, "Power management and dynamic voltage scaling: Myths and facts," in *Proceedings of the 2005 workshop on power aware real-time computing*, vol. 12, 2005, pp. 1–7.

[82] Q. Deng, D. Meisner, A. Bhattacharjee, T. F. Wenisch, and R. Bianchini, "Coscale: Coordinating cpu and memory system dvfs in server systems," in *Proceedings of the 2012 45th Annual IEEE/ACM International Symposium on Microarchitecture*, IEEE Computer Society, 2012, pp. 143–154.

[83] Z. Lai, K. T. Lam, C.-L. Wang, J. Su, Y. Yan, and W. Zhu, "Latency-aware dynamic voltage and frequency scaling on many-core architectures for data-intensive applications," in *2013 International Conference on Cloud Computing and Big Data*, IEEE, 2013, pp. 78–83.

[84] G. Chen, K. Huang, and A. Knoll, "Energy optimization for real-time multiprocessor system-on-chip with optimal dvfs and dpm combination," *ACM Transactions on Embedded Computing Systems (TECS)*, vol. 13, no. 3s, p. 111, 2014.

[85] A. K. Datta and R. Patel, "Cpu scheduling for power/energy management on multicore processors using cache miss and context switch data," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 5, pp. 1190–1199, 2013.

[86] N. B. Rizvandi, J. Taheri, and A. Y. Zomaya, "Some observations on optimal frequency selection in dvfs-based energy consumption minimization," *Journal of Parallel and Distributed Computing*, vol. 71, no. 8, pp. 1154–1164, 2011.

[87] S. Yang, R. A. Shafik, G. V. Merrett, E. Stott, J. M. Levine, J. Davis, and B. M. Al-Hashimi, "Adaptive energy minimization of embedded heterogeneous systems using regression-based learning," in *2015 25th International Workshop on Power and Timing Modeling, Optimization and Simulation (PATMOS)*, IEEE, 2015, pp. 103–110.

[88] F. P. Miller, A. F. Vandome, and J. McBrewster, "Advanced configuration and power interface: Open standard, operating system, power management, cross-platform, intel corporation, microsoft, toshiba,... sleep mode, hibernate (os feature), synonym," 2009.

[89] L. Benini, A. Bogliolo, and G. De Micheli, “A survey of design techniques for system-level dynamic power management,” *IEEE transactions on very large scale integration (VLSI) systems*, vol. 8, no. 3, pp. 299–316, 2000.

[90] W. L. Bircher and L. K. John, “Analysis of dynamic power management on multi-core processors,” in *Proceedings of the 22nd annual international conference on Supercomputing*, ACM, 2008, pp. 327–338.

[91] K. Huang, L. Santinelli, J.-J. Chen, L. Thiele, and G. C. Buttazzo, “Adaptive power management for real-time event streams,” in *Proceedings of the 2010 Asia and South Pacific Design Automation Conference*, IEEE Press, 2010, pp. 7–12.

[92] E.-Y. Chung, L. Benini, and G. De Micheli, “Dynamic power management using adaptive learning tree,” in *Proceedings of the 1999 IEEE/ACM international conference on Computer-aided design*, IEEE Press, 1999, pp. 274–279.

[93] A. Beloglazov and R. Buyya, “Energy efficient resource management in virtualized cloud data centers,” in *Proceedings of the 2010 10th IEEE/ACM international conference on cluster, cloud and grid computing*, IEEE Computer Society, 2010, pp. 826–831.

[94] W.-K. Lee, S.-W. Lee, and W.-O. Siew, “Hybrid model for dynamic power management,” *IEEE Transactions on Consumer Electronics*, vol. 55, no. 2, pp. 656–664, 2009.

[95] C. Imes and H. Hoffmann, “Minimizing energy under performance constraints on embedded platforms: Resource allocation heuristics for homogeneous and single-isa heterogeneous multi-cores,” *ACM SIGBED Review*, vol. 11, no. 4, pp. 49–54, 2015.

[96] J. Trajkovic, A. V. Veidenbaum, and A. Kejariwal, “Improving sdram access energy efficiency for low-power embedded systems,” *ACM Transactions on Embedded Computing Systems (TECS)*, vol. 7, no. 3, p. 24, 2008.

[97] S. Song, C.-Y. Su, R. Ge, A. Vishnu, and K. W. Cameron, “Iso-energy-efficiency: An approach to power-constrained parallel computation,” in *2011 IEEE International Parallel & Distributed Processing Symposium*, IEEE, 2011, pp. 128–139.

[98] J. H. Ahn, J. Leverich, R. Schreiber, and N. P. Jouppi, “Multicore dimm: An energy efficient memory module with independently controlled drams,” *IEEE Computer Architecture Letters*, vol. 8, no. 1, pp. 5–8, 2008.

[99] A. R. Lebeck, X. Fan, H. Zeng, and C. Ellis, “Power aware page allocation,” *ACM Sigplan Notices*, vol. 35, no. 11, pp. 105–116, 2000.

[100] Q. Deng, D. Meisner, L. Ramos, T. F. Wenisch, and R. Bianchini, “Memscale: Active low-power modes for main memory,” in *ACM SIGARCH Computer Architecture News*, ACM, vol. 39, 2011, pp. 225–238.

[101] J. Lin, H. Zheng, Z. Zhu, E. Gorbatov, H. David, and Z. Zhang, “Software thermal management of dram memory for multicore systems,” *ACM SIGMETRICS Performance Evaluation Review*, vol. 36, no. 1, pp. 337–348, 2008.

[102] H. David, C. Fallin, E. Gorbatov, U. R. Hanebutte, and O. Mutlu, “Memory power management via dynamic voltage/frequency scaling,” in *Proceedings of the 8th ACM international conference on Autonomic computing*, ACM, 2011, pp. 31–40.

[103] M. Banikazemi, D. Poff, and B. Abali, “Pam: A novel performance/power aware meta-scheduler for multi-core systems,” in *SC’08: Proceedings of the 2008 ACM/IEEE Conference on Supercomputing*, IEEE, 2008, pp. 1–12.

[104] A. Merkel, J. Stoess, and F. Bellosa, “Resource-conscious scheduling for energy efficiency on multicore processors,” in *Proceedings of the 5th European conference on Computer systems*, ACM, 2010, pp. 153–166.

[105] V. Petrucci, O. Loques, D. Mossé, R. Melhem, N. A. Gazala, and S. Gobriel, "Energy-efficient thread assignment optimization for heterogeneous multicore systems," *ACM Transactions on Embedded Computing Systems (TECS)*, vol. 14, no. 1, p. 15, 2015.

[106] J. Qian, H. Jiang, W. Srisa-An, S. Seth, S. Skelton, and J. Moore, "Energy-efficient i/o thread schedulers for nvme ssds on numa," in *2017 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID)*, IEEE, 2017, pp. 569–578.

[107] C. Hankendi and A. K. Coskun, "Reducing the energy cost of computing through efficient co-scheduling of parallel workloads," in *2012 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, IEEE, 2012, pp. 994–999.

[108] W. Wang, P. Mishra, and S. Ranka, "Dynamic cache reconfiguration and partitioning for energy optimization in real-time multi-core systems," in *2011 48th ACM/EDAC/IEEE Design Automation Conference (DAC)*, IEEE, 2011, pp. 948–953.

[109] G. Chen, B. Hu, K. Huang, A. Knoll, D. Liu, and T. Stefanov, "Automatic cache partitioning and time-triggered scheduling for real-time mpsocs," in *2014 International Conference on ReConFigurable Computing and FPGAs (ReConFig14)*, IEEE, 2014, pp. 1–8.

[110] V. Delaluz, M Kandemir, A. Sivasubramaniam, M. J. Irwin, and N. Vijaykrishnan, "Reducing dtlb energy through dynamic resizing," in *Proceedings 21st International Conference on Computer Design*, IEEE, 2003, pp. 358–363.

[111] K. T. Sundararajan, V. Porpodas, T. M. Jones, N. P. Topham, and B. Franke, "Cooperative partitioning: Energy-efficient cache partitioning for high-performance cmps," in *IEEE International Symposium on High-Performance Comp Architecture*, IEEE, 2012, pp. 1–12.

[112] Y. Liu, H. Yang, R. P. Dick, H. Wang, and L. Shang, "Thermal vs energy optimization for dvfs-enabled processors in embedded

systems," in *8th International Symposium on Quality Electronic Design (ISQED'07)*, IEEE, 2007, pp. 204–209.

[113] M. Huang, J. Renau, S.-M. Yoo, and J. Torrellas, "The design of deetm: A framework for dynamic energy efficiency and temperature management," *Journal of Instruction-Level Parallelism*, vol. 3, pp. 1–31, 2002.

[114] K. Skadron, M. R. Stan, W. Huang, S. Velusamy, K. Sankaranarayanan, and D. Tarjan, "Temperature-aware microarchitecture," in *30th Annual International Symposium on Computer Architecture, 2003. Proceedings.*, IEEE, 2003, pp. 2–13.

[115] A. Cohen, F. Finkelstein, A. Mendelson, R. Ronen, and D. Rudoy, "On estimating optimal performance of cpu dynamic thermal management," *IEEE Computer Architecture Letters*, vol. 2, no. 1, pp. 6–6, 2003.

[116] R. Ayoub, R. Nath, and T. Rosing, "Jetc: Joint energy thermal and cooling management for memory and cpu subsystems in servers," in *IEEE International Symposium on High-Performance Comp Architecture*, IEEE, 2012, pp. 1–12.

[117] S. Zhuravlev, J. C. Saez, S. Blagodurov, A. Fedorova, and M. Prieto, "Survey of energy-cognizant scheduling techniques," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 7, pp. 1447–1464, 2012.

[118] P.-A. Tsai, C. Chen, and D. Sanchez, "Adaptive scheduling for systems with asymmetric memory hierarchies," in *2018 51st Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*, IEEE, 2018, pp. 641–654.

[119] J. C. Saez, M. Prieto, A. Fedorova, and S. Blagodurov, "A comprehensive scheduler for asymmetric multicore systems," in *Proceedings of the 5th European conference on Computer systems*, ACM, 2010, pp. 139–152.

[120] X. Fan, Y. Sui, and J. Xue, “Contention-aware scheduling for asymmetric multicore processors,” in *2015 IEEE 21st International Conference on Parallel and Distributed Systems (ICPADS)*, IEEE, 2015, pp. 742–751.

[121] T. Li, D. Baumberger, D. A. Koufaty, and S. Hahn, “Efficient operating system scheduling for performance-asymmetric multi-core architectures,” in *SC’07: Proceedings of the 2007 ACM/IEEE conference on Supercomputing*, IEEE, 2007, pp. 1–11.

[122] Y. Wang, X. Wang, and Y. Chen, “Energy-efficient virtual machine scheduling in performance-asymmetric multi-core architectures,” in *2012 8th international conference on network and service management (cnsm) and 2012 workshop on systems virtualiztion management (svm)*, IEEE, 2012, pp. 288–294.

[123] F. A. Bower, D. J. Sorin, and L. P. Cox, “The impact of dynamically heterogeneous multicore processors on thread scheduling,” *IEEE micro*, vol. 28, no. 3, pp. 17–25, 2008.

[124] J. Demmel, A. Gearhart, B. Lipshitz, and O. Schwartz, “Perfect strong scaling using no additional energy,” in *2013 IEEE 27th International Symposium on Parallel and Distributed Processing*, IEEE, 2013, pp. 649–660.

[125] J. W. Choi, D. Bedard, R. Fowler, and R. Vuduc, “A roofline model of energy,” in *2013 IEEE 27th International Symposium on Parallel and Distributed Processing*, IEEE, 2013, pp. 661–672.

[126] F. Alessi, P. Thoman, G. Georgakoudis, T. Fahringer, and D. S. Nikolopoulos, “Application-level energy awareness for openmp,” in *International Workshop on OpenMP*, Springer, 2015, pp. 219–232.

[127] V. R. Silva, A. Furtunato, K. Georgiou, K. Eder, and S. Xavier-de Souza, “Energy-optimal configurations for single-node hpc applications,” *ArXiv preprint arXiv:1805.00998*, 2018.

[128] H. Wang, V. Sathish, R. Singh, M. J. Schulte, and N. S. Kim, "Workload and power budget partitioning for single-chip heterogeneous processors," in *Proceedings of the 21st international conference on Parallel architectures and compilation techniques*, ACM, 2012, pp. 401–410.

[129] R. CHIŞ, A. Florea, C. Buduleci, and L. VINŢAN, "Multi-objective optimization for an enhanced multi-core sniper simulator," 2018.

[130] B. Subramaniam and W.-c. Feng, "Statistical power and performance modeling for optimizing the energy efficiency of scientific computing," in *Proceedings of the 2010 IEEE/ACM Int'l Conference on Green Computing and Communications & Int'l Conference on Cyber, Physical and Social Computing*, IEEE Computer Society, 2010, pp. 139–146.

[131] H. F. Sheikh and I. Ahmad, "Dynamic task graph scheduling on multicore processors for performance, energy, and temperature optimization," in *2013 International Green Computing Conference Proceedings*, IEEE, 2013, pp. 1–6.

[132] H. Lei, R. Wang, T. Zhang, Y. Liu, and Y. Zha, "A multi-objective co-evolutionary algorithm for energy-efficient scheduling on a green data center," *Computers & Operations Research*, vol. 75, pp. 103–117, 2016.

[133] N. K. Sharma and G. R. M. Reddy, "Multi-objective energy efficient virtual machines allocation at the cloud data center," *IEEE Transactions on Services Computing*, vol. 12, no. 1, pp. 158–171, 2016.

[134] J. Dong, X. Jin, H. Wang, Y. Li, P. Zhang, and S. Cheng, "Energy-saving virtual machine placement in cloud data centers," in *2013 13th IEEE/ACM International Symposium on Cluster, Cloud, and Grid Computing*, IEEE, 2013, pp. 618–624.

[135] M. Mezmaz, N. Melab, Y. Kessaci, Y. Lee, E.-G. Talbi, A. Zomaya, and D. Tuyttens, "A parallel bi-objective hybrid metaheuristic for energy-aware scheduling for cloud computing systems," *Journal of*

Parallel and Distributed Computing, vol. 71, no. 11, pp. 1497 –1508, 2011.

[136] H. M. Fard, R. Prodan, J. J. D. Barrionuevo, and T. Fahringer, "A multi-objective approach for workflow scheduling in heterogeneous environments," in *Proceedings of the 2012 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (Ccgrid 2012)*, ser. CCGRID '12, IEEE Computer Society, 2012, pp. 300–309.

[137] A. Beloglazov, J. Abawajy, and R. Buyya, "Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing," *Future Generation Computer Systems*, vol. 28, no. 5, pp. 755 –768, 2012, Special Section: Energy efficiency in large-scale distributed systems.

[138] Y. Kessaci, N. Melab, and E.-G. Talbi, "A pareto-based metaheuristic for scheduling hpc applications on a geographically distributed cloud federation," *Cluster Computing*, vol. 16, no. 3, pp. 451–468, Sep. 2013.

[139] J. J. Durillo, V. Nae, and R. Prodan, "Multi-objective energy-efficient workflow scheduling using list-based heuristics," *Future Generation Computer Systems*, vol. 36, pp. 221 –236, 2014.

[140] J. Kołodziej, S. U. Khan, L. Wang, and A. Y. Zomaya, "Energy efficient genetic-based schedulers in computational grids," *Concurr. Comput. : Pract. Exper.*, vol. 27, no. 4, pp. 809–829, Mar. 2015.

[141] V. W. Freeh, D. K. Lowenthal, F. Pan, N. Kappiah, R. Springer, B. L. Rountree, and M. E. Femal, "Analyzing the energy-time trade-off in high-performance computing applications," *IEEE Transactions on Parallel and Distributed Systems*, vol. 18, no. 6, pp. 835–848, 2007.

[142] A. Langer, H. Dokania, L. V. Kalé, and U. S. Palekar, "Analyzing energy-time tradeoff in power overprovisioned hpc data centers," in *2015 IEEE International Parallel and Distributed Processing Symposium Workshop*, IEEE, 2015, pp. 849–854.

[143] V. W. Freeh, D. K. Lowenthal, F. Pan, N. Kappiah, R. Springer, B. L. Rountree, and M. E. Femal, "Analyzing the energy-time trade-off in high-performance computing applications," *IEEE Trans. Parallel Distrib. Syst.*, vol. 18, no. 6, Jun. 2007.

[144] I. Ahmad, S. Ranka, and S. U. Khan, "Using game theory for scheduling tasks on multi-core processors for simultaneous optimization of performance and energy," in *Parallel and Distributed Processing, 2008. IPDPS 2008. IEEE International Symposium on*, 2008, pp. 1–6.

[145] J. W. Choi, D. Bedard, R. Fowler, and R. Vuduc, "A roofline model of energy," in *Parallel & Distributed Processing (IPDPS), 2013 IEEE 27th International Symposium on*, IEEE, 2013, pp. 661–672.

[146] J. Choi, M. Dukhan, X. Liu, and R. Vuduc, "Algorithmic time, energy, and power on candidate HPC compute building blocks," in *Parallel and Distributed Processing Symposium, 2014 IEEE 28th International*, IEEE, 2014, pp. 447–457.

[147] P. Balaprakash, A. Tiwari, and S. M. Wild, "Multi objective optimization of HPC kernels for performance, power, and energy," in *High Performance Computing Systems. Performance Modeling, Benchmarking and Simulation: 4th International Workshop, PMBS 2013, Denver, CO, USA, November 18, 2013. Revised Selected Papers*, A. S. Jarvis, A. S. Wright, and D. S. Hammond, Eds. Springer International Publishing, 2014, pp. 239–260.

[148] M. A. Aba, L. Zaourar, and A. Munier, "Approximation algorithm for scheduling a chain of tasks on heterogeneous systems," in *European Conference on Parallel Processing*, Springer, 2017, pp. 353–365.

[149] B. Subramaniam and W. C. Feng, "Statistical power and performance modeling for optimizing the energy efficiency of scientific computing," in *2010 IEEE/ACM Int'l Conference on Int'l Conference on Cyber, Physical and Social Computing (CPSCom)*, 2010.

[150] S. Song, C. Y. Su, R. Ge, A. Vishnu, and K. W. Cameron, "Iso-energy-efficiency: An approach to power-constrained parallel computation," in *Parallel Distributed Processing Symposium (IPDPS), 2011 IEEE International*, 2011, pp. 128–139.

[151] J. Demmel, A. Gearhart, B. Lipshitz, and O. Schwartz, "Perfect strong scaling using no additional energy," in *Parallel Distributed Processing (IPDPS), 2013 IEEE 27th International Symposium on*, 2013.

[152] M. Drozdowski, J. M. Marszalkowski, and J. Marszalkowski, "Energy trade-offs analysis using equal-energy maps," *Future Generation Computer Systems*, vol. 36, pp. 311–321, 2014.

[153] J. M. Marszalkowski, M. Drozdowski, and J. Marszalkowski, "Time and energy performance of parallel systems with hierarchical memory," *Journal of Grid Computing*, vol. 14, no. 1, pp. 153–170, 2016.

[154] K. M. Tarplee, R. Friese, A. A. Maciejewski, H. J. Siegel, and E. K. Chong, "Energy and makespan tradeoffs in heterogeneous computing systems using efficient linear programming techniques," *IEEE Transactions on Parallel and Distributed Systems*, vol. 27, no. 6, pp. 1633–1646, 2016.

[155] E. Gabaldon, J. L. Lerida, F. Guirado, and J. Planes, "Blacklist muti-objective genetic algorithm for energy saving in heterogeneous environments," *The Journal of Supercomputing*, vol. 73, no. 1, pp. 354–369, 2017.

[156] S. U. Khan, "A goal programming approach for the joint optimization of energy consumption and response time in computational grids," in *Performance Computing and Communications Conference (IPCCC), 2009 IEEE 28th International*, IEEE, 2009, pp. 410–417.

[157] G. Pinto, F. Castor, and Y. D. Liu, "Understanding energy behaviors of thread management constructs," in *ACM SIGPLAN Notices*, ACM, vol. 49, 2014, pp. 345–360.

[158] Y. Guo, "A scalable locality-aware adaptive work-stealing scheduler for multi-core task parallelism," PhD thesis, 2011.

[159] V. Kumar, D. Frampton, S. M. Blackburn, D. Grove, and O. Tardieu, "Work-stealing without the baggage," *ACM SIGPLAN Notices*, vol. 47, no. 10, pp. 297–314, 2012.

[160] H. Ribic and Y. D. Liu, "Energy-efficient work-stealing language runtimes," *ACM SIGARCH Computer Architecture News*, vol. 42, no. 1, pp. 513–528, 2014.

[161] A. Lastovetsky, L. Szustak, and R. Wyrzykowski, "Model-based optimization of eulag kernel on intel xeon phi through load imbalancing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 28, no. 3, pp. 787–797, 2017.

[162] H. Khaleghzadeh, R. R. Manumachu, and A. Lastovetsky, "A novel data-partitioning algorithm for performance optimization of data-parallel applications on heterogeneous hpc platforms," *IEEE Transactions on Parallel and Distributed Systems*, vol. 29, no. 10, pp. 2176–2190, 2018.

[163] D. Clarke, A. Lastovetsky, and V. Rychkov, "Dynamic load balancing of parallel computational iterative routines on highly heterogeneous HPC platforms," *Parallel Processing Letters*, vol. 21, pp. 195–217, 2011.

[164] A. Lastovetsky, R. Reddy, V. Rychkov, and D. Clarke, "Design and implementation of self-adaptable parallel algorithms for scientific computing on highly heterogeneous HPC platforms," *ArXiv preprint arXiv:1109.3074*, 2011.

[165] HCL, *Hclwattsup: API for power and energy measurements using WattsUp Pro Meter*, 2016. [Online]. Available: `http://git.ucd.ie/hcl/hclwattsup`.

[166] V. Petrucci, O. Loques, D. Mossé, R. Melhem, N. A. Gazala, and S. Gobriel, "Energy-efficient thread assignment optimization for heterogeneous multicore systems," *ACM Trans. Embed. Comput. Syst.*, vol. 14, no. 1, Jan. 2015.

[167] Y. G. Kim, M. Kim, and S. W. Chung, “Enhancing energy efficiency of multimedia applications in heterogeneous mobile multi-core processors,” *IEEE Transactions on Computers*, vol. 66, no. 11, pp. 1878–1889, 2017.

[168] W. Wang, P. Mishra, and S. Ranka, “Dynamic cache reconfiguration and partitioning for energy optimization in real-time multi-core systems,” in *2011 48th ACM/EDAC/IEEE Design Automation Conference (DAC)*, 2011, pp. 948–953.

[169] G. Chen, K. Huang, J. Huang, and A. Knoll, “Cache partitioning and scheduling for energy optimization of real-time mpsocs,” in *2013 IEEE 24th International Conference on Application-Specific Systems, Architectures and Processors*, 2013, pp. 35–41.

[170] S. Zhuravlev, J. C. Saez, S. Blagodurov, A. Fedorova, and M. Prieto, “Survey of scheduling techniques for addressing shared resources in multicore processors,” *ACM Comput. Surv.*, vol. 45, no. 1, Dec. 2012.

[171] J. Yang, X. Zhou, M. Chrobak, Y. Zhang, and L. Jin, “Dynamic thermal management through task scheduling,” in *ISPASS 2008 - IEEE International Symposium on Performance Analysis of Systems and software*, 2008, pp. 191–201.

[172] R. Z. Ayoub and T. S. Rosing, “Predict and act: Dynamic thermal management for multi-core processors,” in *Proceedings of the 2009 ACM/IEEE International Symposium on Low Power Electronics and Design*, ser. ISLPED ’09, ACM, 2009, pp. 99–104.

[173] T. Li, D. Baumberger, D. A. Koufaty, and S. Hahn, “Efficient operating system scheduling for performance-asymmetric multi-core architectures,” in *SC ’07: Proceedings of the 2007 ACM/IEEE Conference on Supercomputing*, 2007, pp. 1–11.

[174] E. Humenay, D. Tarjan, and K. Skadron, “Impact of process variations on multicore performance symmetry,” in *2007 Design, Automation Test in Europe Conference Exhibition*, 2007, pp. 1–6.

[175] IntelÂő Math Kernel Library (IntelÂő MKL), *Intel MKL FFT - fast fourier transforms*, 2019. [Online]. Available: `https://software.intel.com/en-us/mkl`.

[176] Z. Xianyi, *Openblas, an optimized blas library*, 2019. [Online]. Available: `http://www.netlib.org/blas/`.

[177] K. Miettinen, *Nonlinear multiobjective optimization*. Kluwer, 1999.

[178] E.-G. Talbi, *Metaheuristics: From design to implementation*. John Wiley & Sons, 2009, vol. 74.

[179] J. Treibig, G. Hager, and G. Wellein, "Likwid: A lightweight performance-oriented tool suite for x86 multicore environments," in *2010 39th International Conference on Parallel Processing Workshops*, IEEE, 2010, pp. 207–216.

[180] I. Kadayif, P. Nath, M. Kandemir, and A. Sivasubramaniam, "Reducing data tlb power via compiler-directed address generation," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 26, no. 2, pp. 312–324, 2007.

[181] V. Karakostas, J. Gandhi, A. Cristal, M. D. Hill, K. S. McKinley, M. Nemirovsky, M. M. Swift, and O. S. Unsal, "Energy-efficient address translation," in *2016 IEEE International Symposium on High Performance Computer Architecture (HPCA)*, 2016, pp. 631–643.

[182] V. Karakostas, J. Gandhi, F. Ayar, A. Cristal, M. D. Hill, K. S. McKinley, M. Nemirovsky, M. M. Swift, and O. Ünsal, "Redundant memory mappings for fast access to large memories," in *Proceedings of the 42Nd Annual International Symposium on Computer Architecture*, ser. ISCA '15, ACM, 2015, pp. 66–78.