International Journal of Electrical and Computer Engineering (IJECE)
Vol. 13, No. 3, June 2023, pp. 3079~3087
ISSN: 2088-8708, DOI: 10.11591/ijece.v13i3.pp3079-3087

Performance analysis of multicore processors using multi-scaling techniques

Jwan Mohammed, Diary R. Sulaiman
Department of Electrical Engineering, College of Engineering, Salahaddin University, Erbil, Iraq

Article Info

Article history:

Received Aug 6, 2022
Revised Sep 10, 2022
Accepted Oct 1, 2022

Keywords:

Core scaling
Dynamic voltage frequency scaling
Performance
Power constraint

ABSTRACT

Integrating more cores per chip enables more programs to run simultaneously, and more easily switch from one program to another, and the system performance will be improved significantly. However, this current trend of central processing unit (CPU) performance cannot be maintained since the budget of power per chip has not risen while the consumption of power per core has slowly reduced. Generally, the processor's maximum performance is proportional to the product of the number of their cores and the frequency they are running at. However, this is usually limited by constraints of power. In this study, first, the voltage/frequency adjustment of the running cores has been analyzed for several programs to improve the processor's performance within the constraint of power. Second, the impact of dynamically scaling the number of running cores is summarized for additional performance improvements of the active programs and applications. Finally, it has been verified that scaling the number of the running cores and their voltage/frequency simultaneously can improve the processor's performance for a higher power dissipation or under power constraints. The performance analysis and improvements are obtained in a real-time simulation on a Linux operating system using a GEM5 simulator. Results indicated that performance improvement was attained at 59.98%, 33.33%, and 66.65% for the three scenarios, respectively.

Corresponding Author:

Jwan Mohammed
Department of Electrical Engineering, College of Engineering, Salahaddin University
Kirkuk Road, Erbil, Iraq
Email: jwan.ali@epu.edu.iq, jwanawayyes@gmail.com

1. INTRODUCTION

In the past decades, Moore's Law has always been the major factor leading high-performance computing. The continuous evolution of complementary metal-oxide-semiconductor (CMOS) technology increases the performance of single-core processors in a linear manner; also, it doubles the density of transistors for very-large-scale integration (VLSI) systems each 18 to 24 months [1]. However, the era of increasing frequency and performance without additional power density is over as the threshold voltage stops scaling along with the lithographic size of transistors [2], [3]. Since 2005, the industry of semiconductor has switched to multi-core and many-core processors for efficient utilization of the enormous number of per-chip transistors. The potential of many-core processors with network-on-chip interconnects has been proven for high-performance and energy-efficient computing [4], [5].

Years before, the major improvements in computer system performance have been accomplished in two different ways: i) integrating additional hardware resources (i.e., cores, on-chip interconnects/caches1, and off-chip channels of memory) and ii) running these resources faster with higher voltage/frequency ratio [6]–[9] with some optimizations at software-level for better exploitation of the parallel architecture of

systems [10]–[14]. Here are some cases of scalable resources: the level of voltage and frequency, the number of operating cores, the size of the queues, buffers, and registers, the number of execution units, and the size of the cache. The scaling algorithm's job is to decide how much to scale each scalable resource in order to maximize the effectiveness of the application that is currently being executed [15]. However, these computer systems also have a high power consumption [16], [17]. This basically limits additional improvement in these systems' performance by just adding more hardware resources and running them at higher frequencies [18]–[20]. The performance of programs scales in a linear manner with the number of cores that are available and/or the frequency at which they are running. However, for illustration, due to the power constraint, approximately only 75% of the total number of cores can operate simultaneously at a specific frequency. In this instance, the reduction of the voltage/frequency of the processor's cores enables the system to run all the available cores within its power constraint. This could enhance the performance of the programs since the performance is roughly proportional to the product of the number of running cores and the frequency [21]. The product is more likely to increase with a higher number of cores running at lower voltage/frequency within the power constraint. This is due to the reduction in power consumption from lower voltage/frequency, which exceeds the increase of power consumption when a higher number of cores are running. However, due to the parallelism of programs and/or the bandwidth of on-chip interconnect/cache and the channels of off-chip memory, the performance of other programs does not scale with the number of running cores and their voltage/frequency [22]. The power constraint also restricts on-chip interconnects/caches1 and off-chip memory channels from raising their bandwidth. For instance, if a program's thread count is not sufficiently high when compared to the number of running cores, adding more cores will not enhance the performance. In this case, increasing voltage/frequency while decreasing the number of cores (e.g., halving the number of running cores) can enhance the performance within the power constraint [23]. Moreover, the performance of programs with high rates of memory accesses and/or inter-core interactions may not improve with either more running cores or a higher voltage/frequency. This is due to performance limitations by the bandwidth of off-chip memory channels and/or on-chip interconnects/caches1. For example, the system can increase the frequency of on-chip interconnects and caches to increase their bandwidth while concurrently decreasing the number of running cores and/or their frequency to fulfill the power constraint. This could enhance the performance of the system for those types of programs.

In this article, the simultaneous and dynamic scaling of the number of the running cores for the processor, as well as the variation of voltage/frequency, has been studied. The achieved results prove the impact of combining the core scaling technique with the voltage/frequency scaling technique. It also confirms the validity of the theoretical fundamentals for high-performance processors. The main originality is using combined multi-scaling techniques to study, analyze and enhance the performance of a multicore processor. First, the voltage and frequency scaling are used dynamically and in real-time for a processor of four running cores. The system was analyzed for several programs to improve the processor's performance within the constraint of power. Second, the dynamic scaling of running cores' numbers is studied and applied for additional performance improvements of the active programs and applications using different benchmark programs. Finally, both scaling techniques are combined and simultaneously applied to exploit the total performance improvements. It is confirmed that the proposed idea of combining techniques for processor performance has a great impact on a processor with a higher power dissipation or under power constraints. Results indicated that the performance improvements attained were 59.98%, 33.33%, and 66.65% for the three scenarios, respectively.

2. THEORETICAL ANALYSIS

The performance of a multicore processor executing compute-bound programs is proportional to the product of the number of running cores and their clock frequency [15], [24].

$$Performance \propto F * N$$

where F is the clock frequency of the core and N is the number of the running cores executing the threads. For measuring performance, the instruction per second (IPS) is a better indicator of central processing unit (CPU) performance is used. Higher instructions per second indicate higher performance [25]:

$$IPS = N * IPC * F \quad (1)$$

where *IPC* is the number of instructions that are executed per cycle for each core while executing the threads. Performance can be improved by raising the number of cores, frequency, or both, although doing so is limited by the system's constraint of power consumption (P_{max}).

2.1. Methodology and problem statement

The methodology followed in this article (unlike recent power reduction techniques) integrates the most recent and efficient power-aware techniques that are applied for high-performance multicore processors. This study developed a novel power-aware integrated voltage, frequency, and the number of operating cores scaling techniques. When these techniques are combined, the power consumption of multicore processing systems can be reduced in real-time while executing any kind of tasks, whether they are hard, soft, periodic, or aperiodic. The key characteristics of the combined techniques are validated through the formal proof of the mathematical model and theoretical principles and also through a series of comparisons with recent achievements of power-aware scaling techniques.

The problem formulation is an important step. It introduces the importance of the approach being studied and specifies the article's objectives. When a domain with a variable voltage/frequency ratio (*V/F*) is available, the system's power consumption (*P*) is basically represented by (2),

$$P\ (N,V,F) = N * (C_{EFF} * F(V) *\ V^2 +\ I_{LEAK}(V) * V) \quad (2)$$

where I_{LEAK} and C_{EFF} are the effective switching leakage current and capacitance, while $F(V)$ and $I_{LEAK}(V)$ represent the frequency and leakage current that is dependent on the voltage (V). The methodology of the power model in [26] has been followed, assuming that leakage power represents 30% [27]. C_{EFF} and I_{LEAK} depend on the activities of the circuit (i.e., the features of the program while executing it); however, worst-case of power consumption should be taken into account since the firmware of the system and the operating system are unable to track the immediate power consumption changes due to rapid changes in the activity of the circuit during execution time. Another assumption is that power gating is utilized to disable the cores that are not in use (i.e., they are not consuming any power). *V/F* and *N* could be adjusted for the purpose of enhancing the performance of a multicore processor within a specified maximum power constraint (P_{max}).

The objective function is $Maximize\ (Performance\ (N,V))$, while the constraints are explained in (3),

$$\begin{aligned} &P\ (N,V)\ \leq\ P\ (Nnom,Vnom)\ =\ Pmax \\ &N\ \leq\ N_{max}, \text{and}\ V_{min}\ \leq\ V\ \leq\ V_{max} \end{aligned} \quad (3)$$

where performance (N, V) is the performance of the multi-core processor as a function of the number of cores (N) and running voltage (V); N_{nom} is the nominal number of running cores, V_{nom} is the nominal voltage for the running cores (N_{nom}); P_{max} is equal to P (N_{nom}, V_{nom}), V_{min} is the system's minimal voltage which is constrained by on-chip memory element failures, and V_{max} is the maximum voltage limited by the processor's transistors' gate-oxide reliability. The mathematical model in [28] had been followed for adjusting the voltage within the desired clock frequency range.

$$V_{nom}\ (F) =\ V_{max} *\ \sqrt{F/F_{max}} \quad (4)$$

where $V_{nom}(F)$ is the required nominal voltage (V_{nom}) at any frequency F. V_{max} and F_{max} are the maximum supply voltage and frequency, respectively, defined within the hardware parameters of the model. This accurate voltage selection will enhance performance even more while maintaining power constraints.

2.2. Impact of *V/F* and *N* on the performance

Assuming a multicore processor has a maximum number of running cores (N_{max}), only (N_{nom}) cores can simultaneously run at F and V_{nom} (i.e., V at F) to meet the power constraint. Reducing V decreases the power (P) at least cubically (2). Since the power (P) is reduced, the number of the running cores (N_{nom}) can rise to N_{max} as long as $P \leq P_{max}$ is maintained.

Taking into consideration that performance is usually proportional to the product of F and N, reducing *V/F* while increasing N could enhance the performance only if the performance is constrained by (1) the on-chip interconnects/caches1 and off-chip memory bandwidth or (2) threads' number (i.e., size of the problem). Consider that in order to run more cores, *V/F* cannot be decreased endlessly as it will be constrained by (1) the maximum number of cores N_{max} and/or (2) the requirement of minimum voltage (V_{min}) that is restricted by the failures of on-chip memory since these failures increase in an exponential manner as V decreases. When the performance of a processor for certain programs is not improved by using a higher N value, it can be enhanced more effectively with a smaller N and greater *V/F*. Keep in mind the super-linear increase in (P) as a result of running at greater F, $F \times N$ at higher *V/F* is considerably smaller than $F \times N$ at lower *V/F*. Thus, for a given power constraint, N reduces much more.

For example, if a program requires running at most 2 cores of a 4 cores processor because of the limited threads' number, increasing the number of running cores to more than 2 will not enhance the

performance. In this situation, running the program at a higher F (instead of a higher N) will enhance the performance due to the fast execution of the threads. Note that decreasing the number of the running cores to N=2 enables the processor to run at higher F without violating the constraint of power. Meanwhile, the performance of some programs could decrease when running with higher N because of increasing the contentions of on-chip interconnects/caches1 and the memory controller accesses. In this case, reducing N (while increasing F) usually decrease these contentions. This will enhance the performance further than the case of running the processor with higher N and lower F. Remember that the frequency of on-chip interconnects/caches1 and the memory controller accesses is also in proportion with $F \times N$.

3. SIMULATION SETUP

In this study, the performance of a multicore computer system has been analyzed using a GEM5 simulator on Linux operating system. Figure 1 shows the architecture of the multicore model. Table 1 shows the specifications of hardware parameters for the suggested model. The essential parameters as per the simulation process are identified as the CPU type, CPU frequency, number of cores, memory type, number of memory channels, number of memory ranks per channel, and the physical memory size. For the specified model, these parameters are adjusted to build an ARM processor with 4 HPI type cores. The power supply has a maximum clock frequency of 120 MHz at 1.8 to 3.6 volts [29]. DDR3_1600_8x8 RAM, 2 memory channels, and 2 GB physical memory size. It is observable that there are two levels of caches, L1 (private) and L2 (shared). Each line has a 64 bytes cache.

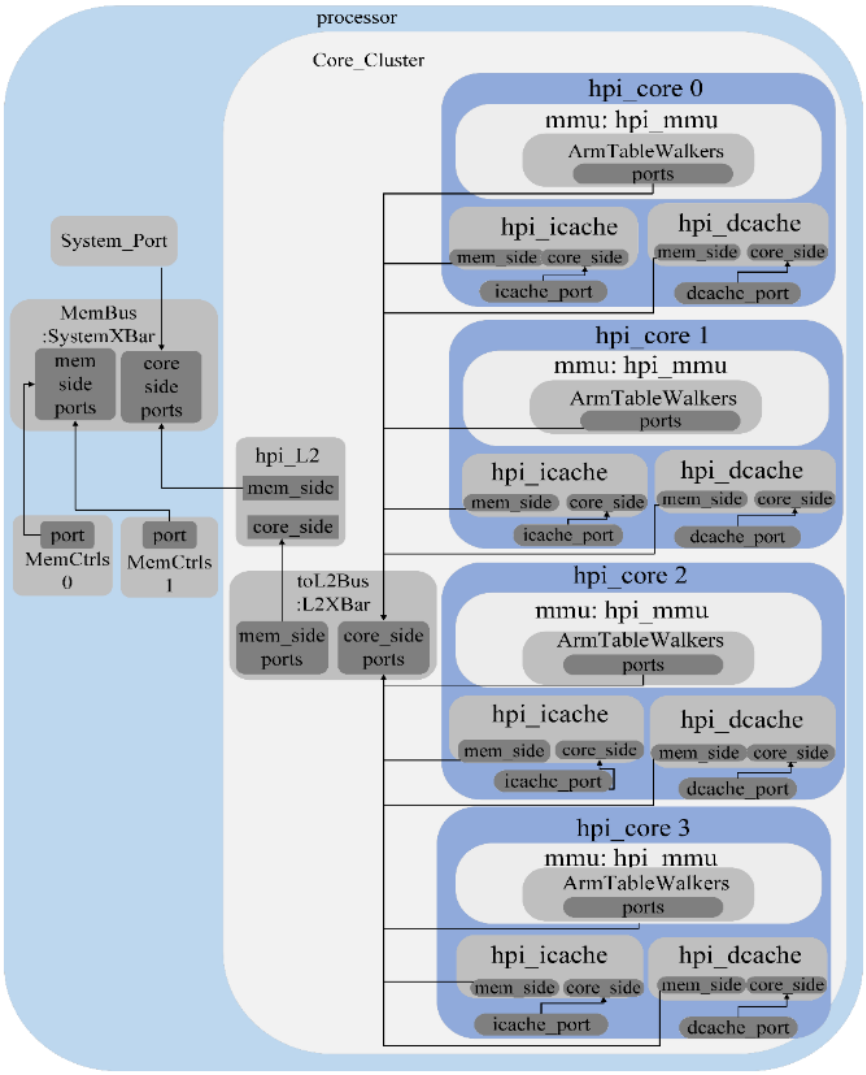

Figure 1. The system model of the targeted multicore processor

Table 1. Hardware parameters for the suggested ARM model

Hardware parameter	Description
processor type	HPI
number of cores (N_{max})	4 cores
clock frequency	Max 120 MHz
supply voltage	1.8 volts to 3.6 volts
RAM	DDR3_1600_8x8
number of memory channels	2 channels
physical memory size	2 GB
caches	L1 (private) and L2 (shared)
cache size	64 bytes for each line

In order to observe the impact of the approach and the processor's capabilities, five benchmarks have been executed on the system model which are standard against the performance metrics [30], [31]. *Specbzip*, *speclibm* and *specmcf* belong to SPEC CPU2006 Benchmarks, while *Bubblesort* and *FloatMM* belong to Stanford benchmarks. Table 2 shows the executed benchmarks with their specifications. The model has been simulated in different execution environments. Scaling the voltage/frequency (*V/F*) levels, the number of active cores (*N*), or both (*V/F* and (*N*) simultaneously has been used to create various execution environments. The efficiency of the technique has been observed by studying the performance behavior under these different circumstances.

Table 2. Benchmarks specifications

Benshmarks	Specification
Specbzip	Forces the processor to compress and decompress different types of files
Speclibm	Which executes the "Lattice Boltzmann Method" to simulate a 3D incompressible fluid
Specmcf	Developed from MCF, a software tool for scheduling single-depot vehicles in public transportation
Bubblesort	A simple algorithm for sorting a collection of elements in ascending or descending sequence
FlaotMM	Uses single-precision arithmetic. Floating point composite result (matrix multiplication, FFT). FFT is an algorithm used for the determination of discrete Fourier transform for the input with significantly less time as compared with direct computation

4. SIMULATION AND RESULTS

The clock frequency was adjusted and scaled from 30 to 120 MHz in order to observe how the performance will behave. The voltage value (V) corresponding to any clock frequency (F) within the adjusted range has been obtained by applying (4) in order to perform precise simulation steps. Applying (4) has produced Table 3. It includes the precise voltage value (V) for every clock frequency (F) between 30 and 120 MHz. Figure 2 illustrates the gradual rise in voltage (V) when the clock frequency is increased (F).

Table 3. The clock frequency (*F*) with the precise corresponding voltage (*V*)

#	F/MHz	V/volts
1	30	1.8
2	52	2.4
3	70	2.7
4	75	2.8
5	80	2.9
6	90	3.1
7	100	3.3
8	110	3.4
9	120	3.6

Figure 3 plots the voltage *V* and the maximum possible number of running cores *N* versus the clock frequency (*F*) while the P_{max} constraint is satisfied for certain values of the frequency (F) and the voltage (*V*); in the plot *N* and *V* have been normalized to *N*=3 at *V*=3.1 V and *F*=90 MHz. For instance, reducing *F* to 30 MHz along with scaling *V* enables the processor to rise *N* to N_{max}=4 without compromising the power constraint. Also, reducing *N* to 2 enables the processor to rise *F* to 120 MHz. The detailed performance analysis is summarized in the next sections. The model is studied for scaling V/F and N individually versus the performance analysis when scaling both V/F and N simultaneously.

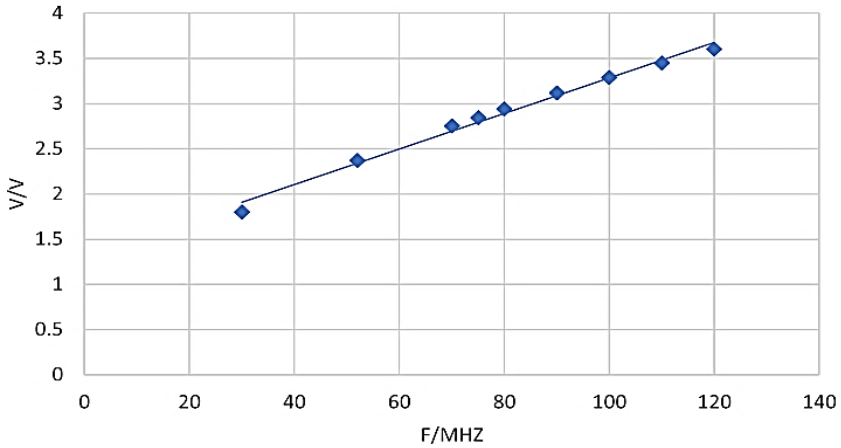

Figure 2. The gradual rise in voltage (*V*) when the clock frequency is increased (*F*)

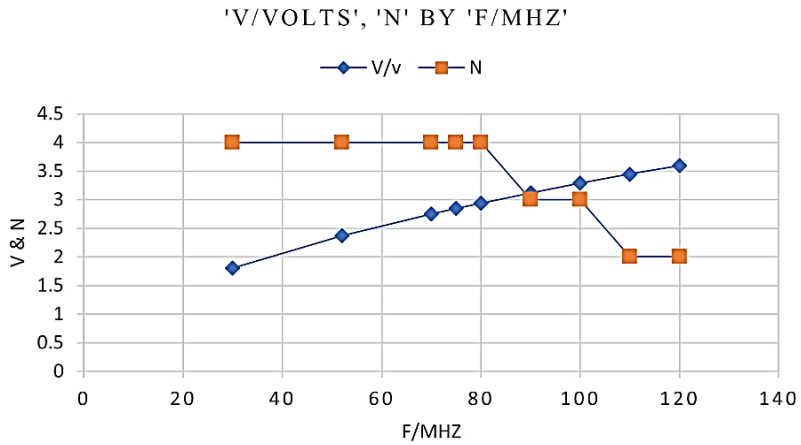

Figure 3. *V* and *N* versus *F* while P_{max} is satisfied for certain values of *F* and *V*

4.1. The performance observation with V/F scaling

Figure 4 plots the performance of various benchmarks measured with millions of instructions per second (MIPS) versus *V* and *F*, where *F* determines *V* following (4). The benchmarks have been run with *V/F* adjusted to the values mentioned in Table 3 while keeping N unchanged at *N*=2. Performance observation started with 1.8 V/30 MHz, then moved to the other three values of *V/F* mentioned in Table 3.

- Step 1 is changing the values of *V/F* from 1.8 V/30 MHz to 2.4 V/52 MHz improved the performance. Maximum improvement was 26.82% for *Bubblesort* and *FloatMM* benchmarks. The minimum improvement was 23.27% for the *specbzip* benchmark.
- Step 2 is increasing the values of V/F to 2.8 V/75 MHz improved the performance by a maximum of 42.85% for the *Bubblesort* benchmark. The minimum improvement was 40.63% for the *specmcf* benchmark.
- In step 3, finally, at V/F=3.6 V/120 MHz maximum improvement is 59.98% for the *Bubblesort* benchmark. The minimum improvement was 56.63% for the *specmcf* benchmark.
- So, up to 59.98%, performance improvement could be obtained without violating power constraints by scaling V/F.

4.2. The performance observation with *N* scaling

Figure 5 plots the performance (MIPS) of various benchmarks versus *N*. First, the benchmarks have been run with *N*=2 and *V/F* adjusted to 2.4 V/52 MHz. To examine how the performance behaves, a second run has been performed with N=4 and *V/F* remaining constant at 2.4 V/52 MHz. The performance was enhanced by changing *N* from 2 to 4 by up to 33.33% for the *Bubblesort* and *FloatMM* benchmarks, and by a minimum of 32.45% for the *specmcf* benchmark.

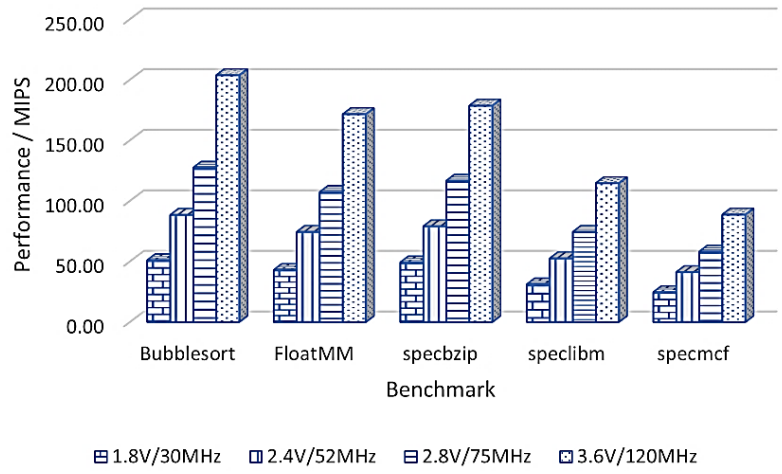

Figure 4. The performance of various benchmarks at versus *V* and *F*

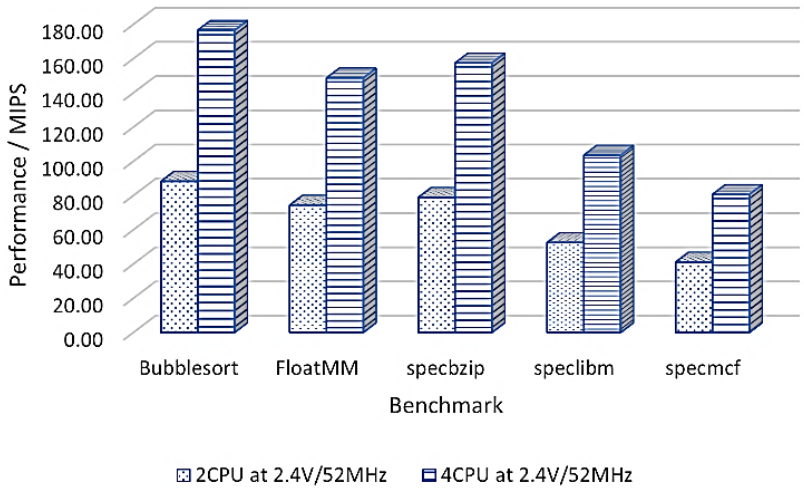

Figure 5. The performance of various benchmarks versus *N*

4.3. The performance observation with *V/F* and *N* scaling

Figure 6 plots the performance of various benchmarks versus *V/F* and *N*. The benchmarks have been run while scaling both *N* and *V/F* values simultaneously. The performance has been observed in these various environments. The values of the adjusted *N* and *V/F* are listed:

- *N*=1 and *V/F*=3.6 V/120 MHz
- *N*=2 and *V/F*=1.8 V/30 MHz
- *N*=2 and *V/F*=2.4 V/52 MHz
- *N*=2 and *V/F*=2.8 V/75 MHz
- *N*=2 and *V/F*=3.6 V/120 MHz
- *N*=4 and *V/F*=1.8 V/30 MHz
- *N*=4 and *V/F*=2.4 V/52 MHz
- *N*=4 and *V/F*=2.8 V/75 MHz

Minimum performance was obtained at (*N*=2 and *V/F*=1.8 V/30 MHz), while maximum performance was obtained at (*N*=4 and *V/F*=2.8 V/75 MHz). The improvement in performance started with a minimum value of 0.02% for Bubblesort benchmark. This minimum value was obtained when scaling (*N*=1 and *V/F*=3.6 V/120 MHz) to (*N*=4 and *V/F*=1.8 V/30 MHz). Maximum performance was at *N*=4 and V/F=2.8 V/75 MHz. The maximum improvement in performance value was 66.65% for *Bubblesort* and *FloatMM* benchmarks when scaling (*N*=2 and *V/F*=1.8 V/30 MHz) to (*N*=4 and *V/F*=2.8 V/75 MHz). Table 4 shows the precise minimum and maximum improvement in the performance for each of the benchmarks within the simulation.

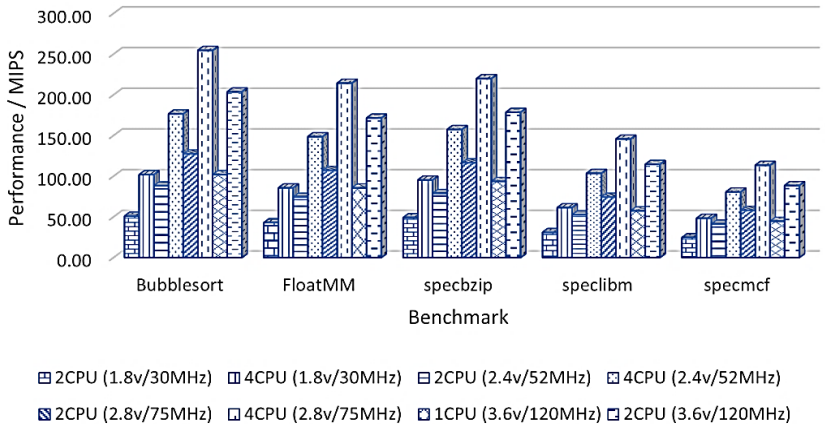

Figure 6. The performance of various benchmarks versus *V/F* and *N*

Table 4. Minimum and maximum improvement in the performance for all benchmarks within the simulation

Benchmark	Min Improvement	Max Improvement
Bubblesort	0.02%	66.65%
FloatMM	0.03%	66.65%
specbzip	0.91%	63.49%
speclibm	3.20%	64.91%
specmcf	4.04%	64.44%

5. CONCLUSION

The performance assessment according to the power constraints in multicore processors is analyzed and studied. Utilizing the scaling of both the number of processor's running cores and their voltages/frequencies separately and simultaneously. First, the impact of the voltage/frequency of the cores on the processor's performance under a power constraint was studied. The performance analysis and improvements are obtained in a real-time simulation on a Linux OS using a GEM5 simulator. It is achieved that selecting an appropriate ratio voltage/frequency of the cores for a given program can improve the performance of a processor by up to 59.98%. Second, it was determined that scaling the number of running cores for a given program increases the performance of a processor by up to 33.33%. Third and finally, scaling both the number of running cores and their voltages/frequencies within the power constraint increases the processor's performance by up to 66.65%. Hence, using the third approach that combines both scaling techniques of the running cores number and the processor's voltage/frequency dynamically and simultaneously based on the power constraints has a great impact on improving the processor's performance and reducing the power dissipated in high-performance systems.

REFERENCES

[1] G. Moore, "Cramming more components onto integrated circuits (1965)," in *Ideas That Created the Future*, The MIT Press, 2021, pp. 261–266.
[2] D. Sulaiman, I. Hamarash, and M. Ibrahim, "Microprocessors optimal power dissipation using combined threshold hopping and voltage scaling," *IEICE Electronics Express*, vol. 14, no. 24, pp. 20171046–20171046, 2017, doi: 10.1587/elex.14.20171046.
[3] D. Sulaiman, I. Hamarash, and M. Ibrahim, "Adaptive supply and body voltage control for ultra-low power microprocessors," *IEICE Electronics Express*, vol. 14, no. 12, pp. 20170306–20170306, 2017, doi: 10.1587/elex.14.20170306.
[4] D. N. Truong *et al.*, "A 167-processor computational platform in 65 nm CMOS," *IEEE Journal of Solid-State Circuits*, vol. 44, no. 4, pp. 1130–1144, Apr. 2009, doi: 10.1109/JSSC.2009.2013772.
[5] S. R. Vangal *et al.*, "An 80-Tile Sub-100-W TeraFLOPS Processor in 65-nm CMOS," *IEEE Journal of Solid-State Circuits*, vol. 43, no. 1, pp. 29–41, Jan. 2008, doi: 10.1109/JSSC.2007.910957.
[6] A. M. M. Aalsaud, "Investigation into runtime workload classification and management for energy-efficient many-core systems," Newcastle University, 2019.
[7] G. Martin, "Overview of the MPSoC design challenge," *2006 43rd ACM/IEEE Design Automation Conference*, 2006, pp. 274-279, doi: 10.1145/1146909.1146980.
[8] S. Borkar, "Thousand core chips: a technology perspective," in *Proceedings of the 44th annual Design Automation Conference (DAC '07)*, 2007, pp. 746–749, doi: 10.1145/1278480.1278667.
[9] M. K. Faeq and S. S. Omran, "Cache coherency controller for MESI protocol based on FPGA," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 11, no. 2, pp. 1043–1052, Apr. 2021, doi: 10.11591/ijece.v11i2.pp1043-1052.
[10] X. Chen, "Performance and power management for multi-core processors," Dissertation, Georgia Institute of Technology, 2018.
[11] F. Xia *et al.*, "Voltage, throughput, power, reliability, and multicore scaling," *Computer*, vol. 50, no. 8, pp. 34–45, 2017, doi: 10.1109/MC.2017.3001246.

[12] Z. Khan, M. Alam, and R. A. Haidri, "Effective load balance scheduling schemes for heterogeneous distributed system," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 7, no. 5, pp. 2757–2765, Oct. 2017, doi: 10.11591/ijece.v7i5.pp2757-2765.
[13] A. S. Al-Khalid and S. S. Omran, "Hybrid branch prediction for pipelined MIPS processor," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 10, no. 4, pp. 3476–3482, Aug. 2020, doi: 10.11591/ijece.v10i4.pp3476-3482.
[14] L. Das, S. Mohapatra, and D. P. Mohapatra, "Schedulability of rate monotonic algorithm using improved time demand analysis for multiprocessor environment," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 8, no. 1, pp. 429–440, Feb. 2018, doi: 10.11591/ijece.v8i1.pp429-440.
[15] M. Almatouq, *Performance and power optimization for multi-core systems using multi-level scaling*. University of California, Irvine, 2019.
[16] V. Kontorinis, A. Shayan, D. M. Tullsen, and R. Kumar, "Reducing peak power with a table-driven adaptive processor core," 2009, doi: 10.1145/1669112.1669137.
[17] B. Li, "Modeling and runtime systems for coordinated power-performance management," Dissertation, Virginia Polytechnic Institute and State University, 2019.
[18] K. M. Attia, M. A. El-Hosseini, and H. A. Ali, "Dynamic power management techniques in multi-core architectures: A survey study," *Ain Shams Engineering Journal*, vol. 8, no. 3, pp. 445–456, Sep. 2017, doi: 10.1016/j.asej.2015.08.010.
[19] K. Nagalakshmi and N. Gomathi, "Analysis of power management techniques in multicore processors," in *Artificial Intelligence and Evolutionary Computations in Engineering Systems*, Springer, 2017, pp. 397–418.
[20] K. R. Basireddy, "Runtime energy management of concurrent applications for multi-core platforms," PhD Thesis, University of Southampton, 2019.
[21] J. Lee, V. Sathisha, M. Schulte, K. Compton, and N. S. Kim, "Improving throughput of power-constrained GPUs using dynamic voltage/frequency and core scaling," in *2011 International Conference on Parallel Architectures and Compilation Techniques*, Oct. 2011, pp. 111–120, doi: 10.1109/PACT.2011.17.
[22] D. Culler, J. P. Singh, and A. Gupta, *Parallel computer architecture: a hardware/software approach*. Gulf Professional Publishing, 1999.
[23] R. Kumar, V. Zyuban, and D. M. Tullsen, "Interconnections in multi-core architectures: understanding mechanisms, overheads and scaling," in *32nd International Symposium on Computer Architecture (ISCA'05)*, 2005, pp. 408–419, doi: 10.1109/ISCA.2005.34.
[24] B. Liu, M. H. Foroozannejad, S. Ghiasi, and B. M. Baas, "Optimizing power of many-core systems by exploiting dynamic voltage, frequency and core scaling," in *2015 IEEE 58th International Midwest Symposium on Circuits and Systems (MWSCAS)*, Aug. 2015, pp. 1–4, doi: 10.1109/MWSCAS.2015.7282189.
[25] M. Ghasemazar, E. Pakbaznia, and M. Pedram, "Minimizing energy consumption of a chip multiprocessor through simultaneous core consolidation and DVFS," in *Proceedings of 2010 IEEE International Symposium on Circuits and Systems*, May 2010, pp. 49–52, doi: 10.1109/ISCAS.2010.5537096.
[26] J. Lee and N. S. Kim, "Optimizing total power of many-core processors considering voltage scaling limit and process variations," in *Proceedings of the 2009 ACM/IEEE International Symposium on Low Power Electronics and Design*, 2009, pp. 201–206, doi: 10.1145/1594233.1594283.
[27] K. Aygun, "Power delivery for high-performance microprocessors," *Intel Technology Journal*, vol. 9, no. 4, Nov. 2005, doi: 10.1535/itj.0904.02.
[28] M. Kadin and S. Reda, "Frequency and voltage planning for multi-core processors under thermal constraints," in *2008 IEEE International Conference on Computer Design*, Oct. 2008, pp. 463–470, doi: 10.1109/ICCD.2008.4751902.
[29] Arm, "Documentation," *Arm Developer*. https://developer.arm.com/documentation (accessed Jul. 01, 2022).
[30] "SPEC CPU® 2006," *Standard Performance Evaluation Corporation*. https://www.spec.org/cpu2006/ (accessed Jan. 09, 2018).
[31] J. Lowe-Power, "gem5 documentation learning gem5," *gem5*. https://www.gem5.org/documentation/ (accessed Apr. 08, 2022).

BIOGRAPHIES OF AUTHORS

Jwan Mohammed received a B.Sc. degree in electrical engineering from Baghdad University, Baghdad, Iraq. She is currently an M.Sc. student at Salahaddin University, Erbil, Iraq. Her current research interests include power and performance optimization in multicore processors using multi-scaling techniques. Her LinkedIn can be reached at https://www.linkedin.com/in/jwan-ali-a9283b242. She can be contacted at jwan.ali@epu.edu.iq or jwanawayyes@gmail.com.

Diary R. Sulaiman is a professor of Electronics and Computer Engineering, Department of Electrical Engineering, College of Engineering, Salahaddin University, Erbil, Iraq. He has been working in higher education for more than 30 years and he taught both undergraduate and postgraduate students at Department of Electrical Engineering. He has gained CISCO certifications and information system professional certifications. His current research interests include power/thermal management of microprocessors, advanced digital design, and CMOS circuit design. He has authored many publications on electronics and computer hardware design, CMOS circuit design, and microprocessors power/thermal management. He has more than 50 published articles in international journals and conferences. He can be reached on his academic website at https://academics.su.edu.krd/diary.sulaiman or on LinkedIn at https://www.linkedin.com/in/diariy-sulaiman/. He can be contacted at diary.sulaiman@su.edu.krd or diariy@gmail.com.