An Energy Aware Edge Priority-based Scheduling Algorithm for Multiprocessor Environments

Authors: Ashish Kumar Maurya and Anil Kumar Tripathi

Conference: PDPTA'18, Las Vegas, USA

Paper ID: PDP3408

Presented by:

Anil Kumar Tripathi

Professor

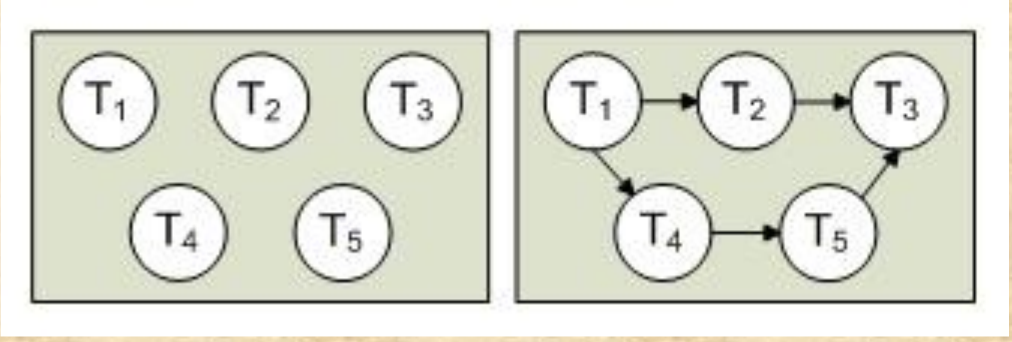
Dept. of Computer Science & Engineering,

Indian Institute of Technology (BHU), Varanasi, INDIA

Contents

Introduction

- Multiprocessors
- New challenges in multiprocessors
- Dynamic Voltage Frequency Scaling (DVFS)
- A parallel application
- Proposed an algorithm called EAEPS
  - It reduces the energy consumption by zeroing the edges of high priorities.
  - It is an energy aware version of our EPS algorithm.



Introduction

Objective

“To develop an energy-aware scheduling algorithm which aims to reduce power consumption by exploiting DVFS technique.“

Contributions

- The idea of edge prioritization leads to meaningful clustering
- The work fruitfully achieves minimization of energy consumption
- A simulation study is performed for benchmark task graphs

Related Work

Task scheduling algorithms

- List scheduling algorithms
- Duplication-based scheduling algorithms
- Clustering-based scheduling algorithms

Energy aware scheduling algorithms

System Model and Problem Formulation

Energy Model

- Processors have several voltage and frequency levels, and a scheduling algorithm may choose the appropriate voltage and frequency to save energy.

$$P_{dynamic} = A \times C_L \times V_{dd}^2 \times f$$

$$\xi = \sum_{\Delta t} (\eta \times V_{dd}^2 \times f \times \Delta t)$$

Processor Model

- Processors are enabled with software controlled DVFS

$$f = \bigcup_{1 \leq i \leq K} \{f_i\} \qquad V = \bigcup_{1 \leq i \leq K} \{V_i\}$$

Application Model

- Directed Acyclic Graph (DAG)

The Proposed Algorithm

Edge prioritization

- Concept of edge priority in Sarkar's algorithm
- Our concept of edge prioritization

$$p(e_{i,j}) = \frac{CT(e_{i,j})}{ET(T_i) + ET(T_j)}$$

Clustering

- Merging of two tasks or clusters
- The order among tasks within a cluster is decided by their bottom-levels.

The Proposed Algorithm

Voltage scaling of non-critical tasks

- Slack of a task is defined as the amount of time the task can be delayed without delaying the makespan.

$$slack(T_i) = LFT(T_i) - EST(T_i) - ET(T_i)$$

where LFT is latest finish time, EST is earliest start time and ET is execution time of the task

- New execution time of the task: $ET_{slack}(T_i) = ET(T_i) + slack(T_i)$
- When a processor executes a non-critical task, it first calculates its slack and then its new execution time. After that it attempts to scale its operating frequency

$$f_k(T_i) = f_{max} ET(T_i) / ET_{slack}(T_i)$$

The Proposed Algorithm

EAEPS Algorithm

```
Initially one cluster for each of the tasks is formed
Calculate initial energy consumption
Compute priority of each edge
Sort all edges in non-increasing order by their priorities and make a list
repeat
    for all edges in the sorted list do
        Zero an edge if energy consumption reduces
        When two clusters are grouped, the order among tasks is decided by comparing their bottom-levels with each other
        if bottom-level of one task is equal to the bottom-level of other task then
            Both tasks are ordered according to their topological-order in the cluster
        end if
        Update energy consumption
        break
    end for
    Remove an edge from sorted list by which energy consumption reduces.
until energy consumption decreases
```

Experimental Results

- Performed simulation for some benchmark task graphs given by Davidovic and Crainic.
- Results of EAEPS are compared with EADUS & TEBUS, Energy reduction algorithm, ECS and PATC.
- Frequency and supply voltages used in this work

Frequency (GHz)	Supply Voltage (V)
0.8	0.90
1.0	1.00
1.2	1.05
1.4	1.10
1.6	1.15
1.8	1.20

Energy aware scheduling algorithms	Minimum energy saving (%)
EADUS & TEBUS	11.4
Energy reduction algorithm	18.5
ECS	29.4
PATC	30.5
EAEPS	31.4

Conclusions

- Proposed an energy-aware version of our EPS algorithm called EAEPS that aims at reducing energy consumption by exploiting DVFS technique and outperforms the four energy-aware scheduling algorithms.
- The simulation results show that EAEPS achieves up to 31.4 % energy saving.

References

- [1] J. Jiang, Y. Lin, G. Xie, L. Fu, and J. Yang, “Time and energy optimization algorithms for the static scheduling of multiple workflows in heterogeneous computing system,” Journal of Grid Computing, vol. 15, no. 4, pp. 435-456, 2017.
- [2] L. Wang, S. U. Khan, D. Chen, J. Kołodziej, R. Ranjan, C-Z Xu, et al., “Energy-aware parallel task scheduling in a cluster,” Future Generation Computer Systems, vol. 29, no. 7, pp. 1661-1670, 2013.
- [3] A. K. Maurya, and A. K. Tripathi, “An edge priority-based clustering algorithm for multiprocessor environments,” Concurrency and Computation: Practice and Experience.
- [4] O. Sinnen, Task Scheduling for Parallel Systems, John Wiley & Sons; 2007.
- [5] A. K. Maurya, and A. K. Tripathi, “Performance comparison of HEFT, Lookahead, CEFT and PEFT scheduling algorithms for heterogeneous computing systems,” In Proceedings of the Seventh International Conference on Computer and Communication Technology 2017 (ICCCT’17), ACM, pp.128–132, 2017.
- [6] A. K. Maurya, and A. K. Tripathi, “On benchmarking task scheduling algorithms for heterogeneous computing systems,” The Journal of Supercomputing, 2018.
- [7] G. Terzopoulos, and H. D. Karatza, “Power-aware Bag-of-Tasks scheduling on heterogeneous platforms,” Cluster Computing, vol. 19, no. 2, pp. 615-631, 2016.
- [8] Z. Zong, A. Manzanares, X. Ruan, and X. Qin, “EAD and PEBD: two energy-aware duplication scheduling algorithms for parallel tasks on homogeneous clusters,” IEEE Transactions on Computers, vol. 60, no. 3, pp.360-374, 2011.
- [9] Y. C. Lee, and A. Y. Zomaya, “Minimizing energy consumption for precedence-constrained applications using dynamic voltage scaling,” 9th IEEE/ACM International Symposium on Cluster Computing and the Grid, 2009 (CCGRID'09), IEEE, 2009.
- [10] Y. Hu, C. Liu, K. Li, X. Chen, and K. Li, “Slack allocation algorithm for energy minimization in cluster systems,” Future Generation Computer Systems, vol. 74, pp.119-131, 2017.

References

- [11] G. Aupy, A. Benoit, and Y. Robert, “Energy-aware scheduling under reliability and makespan constraints,” 19th International Conference on High Performance Computing (HiPC 2012), IEEE, 2012.
- [12] H. Kimura, M. Sato, M., Y. Hotta, T. Boku, and D. Takahashi, “Emprical study on reducing energy of parallel programs using slack reclamation by dvfs in a power-scalable high performance cluster,” In IEEE International Conference on Cluster Computing, IEEE, 2006.
- [13] N. Kaur, S. Bansal, and R. K. Bansal, “Duplication-controlled static energy-efficient scheduling on multiprocessor computing system,” Concurrency and Computation: Practice and Experience, vol. 29, no. 12, 2017.
- [14] J. Mei, K. Li, and K. Li, “Energy-aware task scheduling in heterogeneous computing environments,” Cluster Computing, vol. 17, no. 2, pp. 537-550, 2014.
- [15] X. Tang, and W. Tan, “Energy-Efficient Reliability-Aware Scheduling Algorithm on Heterogeneous Systems,” Scientific Programming 2016, p.14, 2016.
- [16] M. Sharifi, Sa. Shahrivari, and H. Salimi, “PASTA: a power-aware solution to scheduling of precedence-constrained tasks on heterogeneous computing resources,” Computing, vol. 95, no. 1, pp. 67-88, 2013.
- [17] R. Ge, X. Feng, K. W. Cameron, “Performance-constrained distributed DVS scheduling for scientific applications on power-aware clusters,” in: SC, 2005, p. 34.
- [18] M. Y. Lim, V. W. Freeh and D. K. Lowenthal. “Adaptive, transparent frequency and voltage scaling of communication phases in MPI programs,” In proceedings of the ACM/IEEE SC 2006 conference, IEEE, pp. 14-14. 2006.
- [19] D. Kadamuddi, J.J. P. Tsai, “Clustering algorithm for parallelizing software systems in multiprocessors environment,”, IEEE Transactions on Software Engineering, vol. 26, no. 4, pp. 340-361, 2000.
- [20] T. Davidovic, T. G. Crainic, “Benchmark-problem instances for static scheduling of task graphs with communication delays on homogeneous multiprocessor systems,” Computers & Operations Research, vol. 33, no. 8, pp. 2155–2177, 2006.


Thank You!