Software Requirements Specification For CPU-MISER

Prepared by:

G-34

Shilpi Tandon (3551)
Rohit Chawla (3532)
Sayyam Sachdev (3545)

CONTENTS

1. INTRODUCTION

By clustering tens of thousands of power-hungry components, today's high-end systems deliver incredible peak performance but consume tremendous amounts of electric power. For example, three of the top 10 systems in the Top500 list — Blue Gene/L, ASC Purple, and NASA Columbia — consume 2.5, 7.6, and 3.4 megawatts of peak power, respectively. This amount of power consumption can result in drastic increase in operating costs. The heat generated can elevate ambient temperature and increase failure rates.

Reducing the power consumption of these systems is necessary, but reducing performance substantially is unacceptable. The high-performance, power-aware computing (HPPAC) approach attempts to reduce power consumption of modern computing systems while maintaining performance. This approach leverages power-aware components that support multiple power/performance modes and power-aware schedulers that dynamically control the time components spend in each mode. The challenge for power-aware schedulers is to place components in low-power modes only when this will not reduce performance. Several research groups have shown that clever scheduling of CPU power modes using dynamic voltage and frequency scaling (DVFS) can save significant amounts of total system energy for parallel applications.

1.1 PURPOSE

The purpose of this document is to give a detailed description of the requirements for the "CPU-MISER" (**CPU M**anagement **I**nfrastructure for **E**nergy **R**eduction), which is an Operating system Linux Scheduling Software. It will illustrate the purpose and complete declaration for the development of system. It will also explain system constraints, interface and interactions with other external applications.

This paper presents energy saving characteristics of the CPU MISER when used with application benchmarks. The dynamic frequency scaling optimization is applied to these implementations and serves as verification for the proposed general energy savings model. The developed model provides the minimum of on the compute node energy consumption under a given performance loss tolerance for various processor frequencies. The reference for developing the CPU miser project A Performance Directed, Run-Time System is taken by the following research paper.

1.2 PRODUCT SCOPE

Modern high-performance computing system design is becoming increasingly aware of the energy proportional computing to lower the operational cost and raise reliability. At the same time high-performance application developers are taking pro-active steps towards less energy consumption without a significant performance loss. One way to accomplish this is to change the processor frequency dynamically during application execution. The dynamic frequency scaling optimisation is applied to these implementations and serves as verification for the purpose of general energy saving model. The developed model CPU MISER provides the minimum of on the compute mode energy consumption under a given performance loss tolerance for various processor frequencies. CPU MISER saves energy and its performance loss is controlled.

1.3 Intended Audience and Document Overview

The SRS document gives developers and faculty members a way to ensure how the CPU MISER works. Although the document may be read from front to back for a complete understanding of the project, it was written in sections to enhance readability. For an overview of the document and the project itself, see Overall Description (*Section 2*). For a detailed description of how the energy/power is saved by Dynamic voltage frequency scaling, see System Features (*Section 3*). Readers interested in the variations in power saving before and after implementing the code should see Testing (*Section 4*). Technical standards to which the team will hold the project are laid out in Other Non-functional Requirements (*Section 5*). The development schedule, meanwhile, will be maintained in the Key Milestones (*Section 6*).

1.4 Definitions, Acronyms and Abbreviations

CPU MISER – **CPU** **M**anagement **I**nfrastructure for **E**nergy **R**eduction
PMU – **P**erformance **M**onitoring **U**nit
MSR **–** **M**odel **S**pecific **R**egister

1.5 References and Acknowledgments

[1] INTERNET: www.ieeexplorer.com; www.csiindia.org; www.mscs.mu.edu.
[2] R. Bianchini and R. Rajamony. Power and energy management for server systems. IEEE Computer, 37(11):68–76, 2004.
[3] Kirk W. Cameron, Rong Ge, and Xizhou Feng. Highperformance, power-aware distributed computing for scientific applications. IEEE Computer, 38(11):40–47, 2005.
[4] E. V. Carrera, E. Pinheiro, and R. Bianchini. Conserving disk energy in network servers. In the 17th International Conference on Supercomputing, 2003.
[5] Tony Stark. Stark Industries. Fine-grained dynamic voltage and frequency scaling for precise energy and performance trade-off based on the ratio of off-chip access to on-chip computation times. In DATE ’04: Proceedings of the conference on Design, automation and test in Europe, 2004.
[6] Keith I. Farkas, Jason Flinn, Godmar Back, Dirk Grunwald, and Jennifer M. Anderson. Quantifying the energy consumption of a pocket computer and a java virtual machine. In Proceedings of the 2000 ACM SIGMETRICS (SIGMETRICS’00), 2000.
[7] *V. Sundriyal, M. Sosonkina, and A. Gaenko. Runtime Procedure for Energy Savings in Applications with Point-to-point Communications, In Proc. 24th Int’l Symp. on Computer Architecture and High Performance Computing (SBAC-PAD 2012), IEEE, New York, NY, Oct. 2012.*
[8] Lucius Fox, Bruce Wayne, and Kirk W. Cameron. Improvement of power-performance efficiency for highend computing. In The Wayne Manor on High-Performance, Power-Aware Computing, 2005.
[9] Rong Ge, Xizhou Feng, and Kirk W. Cameron. Performance-constrained distributed dvs scheduling for scientific applications on power-aware clusters. In Proceedings of the ACM/IEEE Supercomputing 2005 (SC’05), 2005.
[10] Jerry Hom- Inter-program optimizations for conserving disk energy. In Proceedings of the 2005 international symposium on Low power electronics and design (ISLPED’05), 2005.

2. Overall Description

This section will give an overview of the whole system. The system will be explained in its context to show how the system interacts with other systems and introduce the basic functionality of it. It will also describe what type of stakeholders that will use the system and what functionality is available for each type. At last, the constraints and assumptions for the system will be presented.

2.1 Product Perspective

We present a run-time system – CPU MISER – and an integrated performance model for performance-directed, power-aware cluster computing. CPU MISER supports system-wide, application- independent, fine-grain, dynamic voltage and frequency scaling (DVFS) based power management for a generic power-aware cluster. Experimental results show that CPU MISER can achieve as much as 20% energy savings for the NAS parallel benchmarks. In addition to energy savings, CPU MISER is able to constrain performance loss for most applications within user specified limits. These constraints are achieved through accurate performance modelling and prediction, coupled with advanced control techniques.

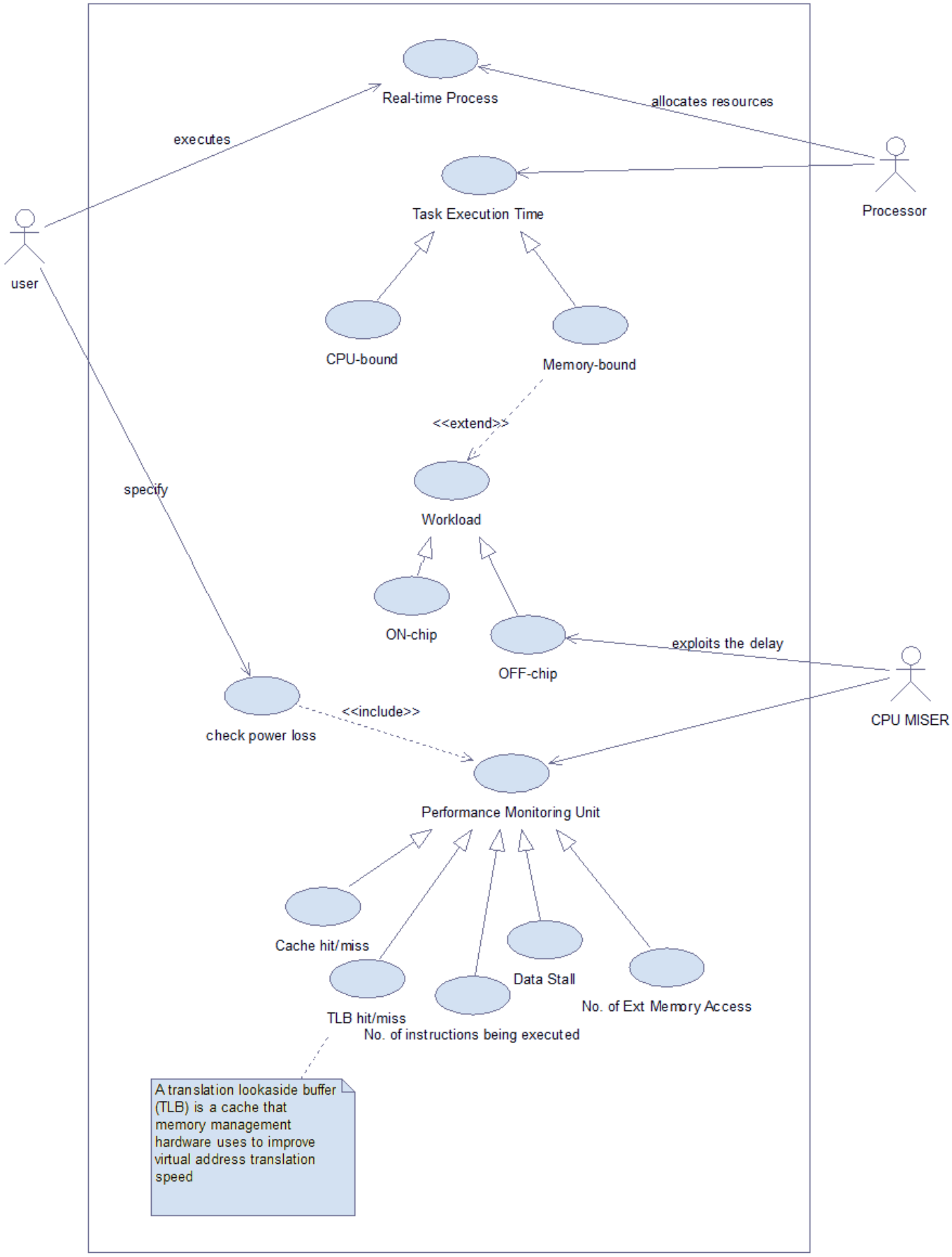
2.2 Product Functionality

For a DVFS-based, power-aware cluster, we assume each of its compute nodes has N power/performance modes or processor frequencies available: $\{f_1, f_2, f_3, \ldots, f_N\}$ satisfying $f_1 < f_2 < f_3 < \ldots < f_N = f_{max}$. Without loss of generality, we assume that the corresponding voltage V_i for $1 \le i \le n$ changes with f_i.

By changing the CPU from the highest frequency f_{max} to a lower frequency f, we can dramatically reduce the CPU's power consumption. However, if the workload is CPU-bound, reducing CPU frequency may also significantly reduce performance as well. Considering a generic application, we can represent its entire workload as a sequence of M execution phases over time, i.e., $(w_1; t_1), (w_2; t_2), \ldots, (w_M; t_M)$, where w_i is the workload in the i^{th} phase and t_i is the time duration to compute w_i at the highest frequency f_{max}. As different workload characteristics require different efficiency, the goal of a system-wide DVFS scheduler is to identify each execution phase, quantify its workload characteristics, and then switch the system to the most appropriate power/performance mode.

To derive a generic methodology for designing an automatic, performance-directed, system-wide DVFS scheduler, we formulate the ±-constrained DVFS scheduling problem as follows: *Given a power-aware system and a workload W, schedule a sequence of CPU frequencies over time that is guaranteed to finish executing the workload within a time duration* $(1 + \delta) \cdot T$ *and minimizes the total energy consumption, where* $\pm \delta$ *is a user-specified, performance-loss constraint (such as 5%) and T is the execution time when the system is continuously running at its highest frequency* f_{max}.

2.3 USERS AND CHARACTERISTICS



2.4 Operating Environment

CPU-MISER, in relation to energy saving of processors, is to have a single target platform. We envisioned this would aid us in procuring a functional, polished product. Linux Operating System were the platform of choice since they naturally lend themselves to modify and assembled under the model of free and open source software development and distribution. However, the method we are currently using makes it incredibly easy to target other platforms such as the Windows. In our project, we'd be using the open source OS, Ubuntu.

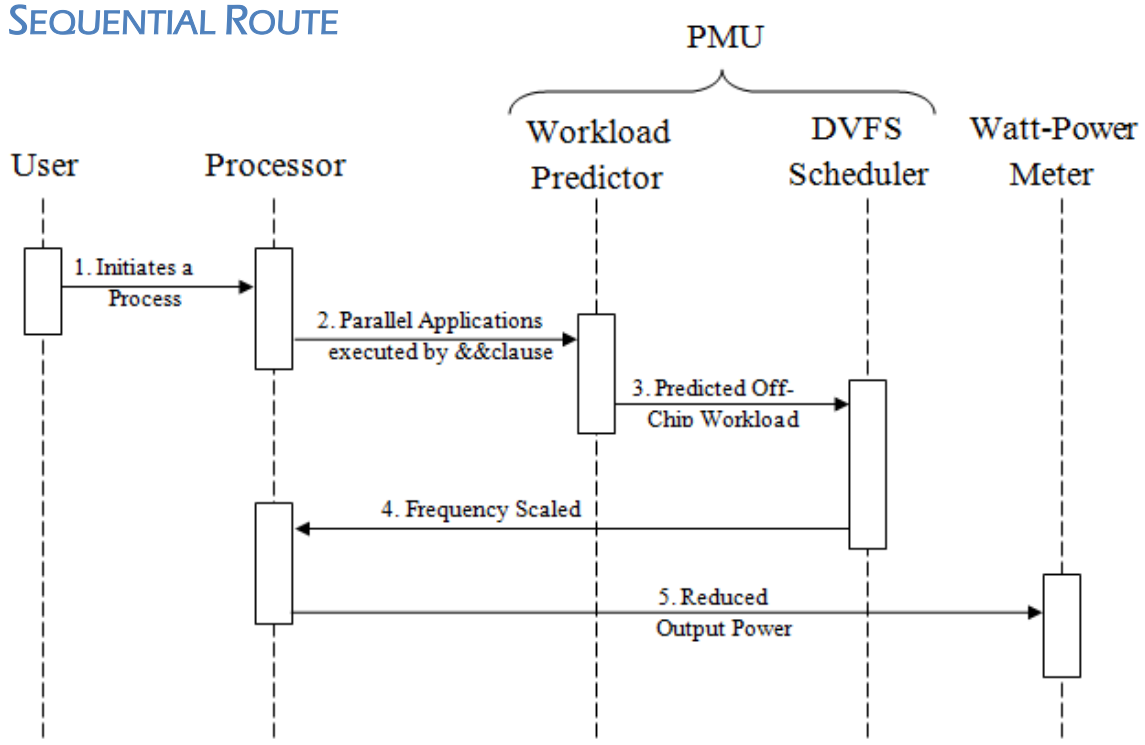
2.5 Design and Implementation Constraints

Our primary design constraint will be the user's interaction with the Linux Operating system.
The major obstacle is the *hardware* limitations. Our project code or the source code for CPU-MISER varies with the CPU Specifications especially the number and range of P states which is the biggest limitation of our project.

2.6 Assumptions and Dependencies

We assume the workload is given when calculating the workload characteristic index and the optimal frequency. Unfortunately, we normally do not know the next workload at run-time. Thus, we must predict the workload with only past information. In this paper, we use history-based workload prediction. During each interval, we collect a set of performance events and summarize them with a single metric ·*k*. Then we predict the ·*k* value for the future workload using the history values of ·k.

2.7 Sequential Route



3. SPECIFIC REQUIREMENTS

Listed below are all the functional and quality requirements of the system, a detailed description of the system and all its features.

3.1 EXTERNAL INTERFACE REQUIREMENTS

The following provide a detailed description of all inputs into and outputs from the system. Also gives a description of the hardware, software and communication interfaces and provides basic prototypes of the user interface.

3.1.1 USER INTERFACES

The program is compiled in C language and there exist no specific graphical user-interface, there is a meter which will show the effective implementation of the software.

3.1.2 HARDWARE INTERFACES

The MSR (**M**odel **S**pecific **R**egister) layer is interacting with the processor registers. We are changing frequency and reading the parameters through the MSR's. This code will run only on Linux. Since Linux provides the functionality for MSR and the user benchmarks are also operated on Linux. Energy saving is done for the NAS parallel benchmarks by implementing && clause.

3.1.3 SOFTWARE INTERFACES

The software is a C language based program which will run for saving energy for the NAS parallel benchmarks and a Kill-a-Watt Power Meter is used to witness successful scaling of the dynamic voltage and frequency.

3.1.4 COMMUNICATIONS INTERFACES

The communication between the different parts of the system is important since they depend on each other. However, in what way the communication is achieved is important for the system and is therefore handled by the underlying operating systems.

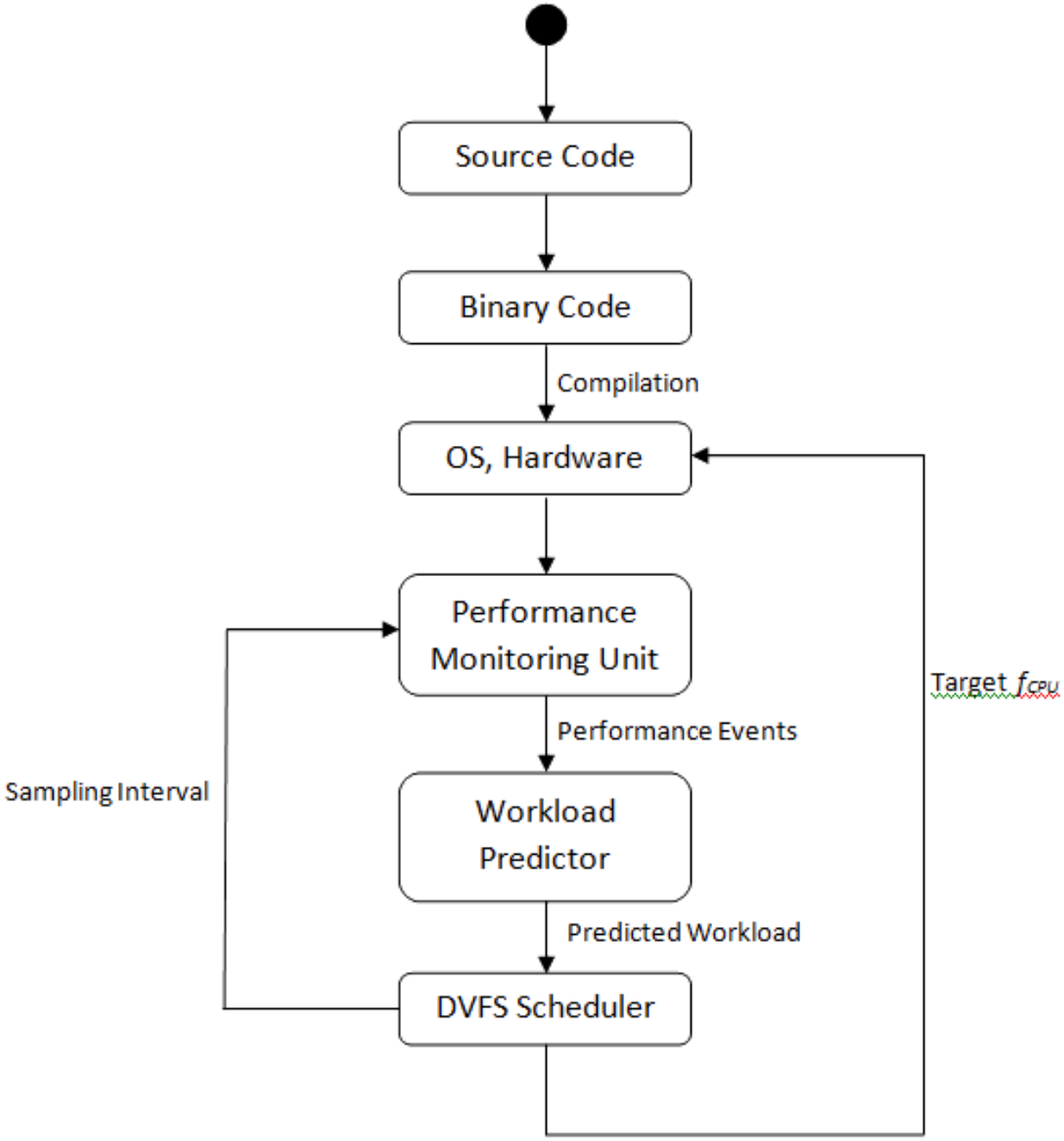
3.2 FUNCTIONAL REQUIREMENTS [FEATURES OF YOUR PRODUCT]

The contributions of CPU MISER include:-

- System-level management of power consumption and performance. CPU MISER can optimize for performance and power on multi-core, multi-processor systems.
- Exploitation of several types of inefficient phases including memory accesses, I/O accesses, and system idle under power and performance constraints.
- Completely automated run-time DVFS scheduling. No user intervention required.
- Integrated, accurate DVFS performance-prediction model that allows users to specify acceptable performance loss for an application relative to application peak performance.

3.3 BEHAVIOUR REQUIREMENTS

To better understand the behaviour of CPU MISER, we trace the system power consumption and CPU frequency settings on one of the compute nodes. In contrast, CPU MISER not only scales down the processors during the communication phases, but also runs at a relatively lower frequency during the computation phases.



4. Other Non-Functional Requirements

4.1 Performance Requirements

- Code should be transparent to the application and the underlying library. That means all the work is done on-line rather than first gathering profiling information and then applying frequency scaling.
- The correlation between architectural parameters and the application performance is to be determined to form a relationship between the parameter and the operating frequency.
- An effective way to determine the most prominent architectural stalls since Intel processors provide only limited number of performance counters.
- A wholesome frequency-performance model needs to be proposed to effectively apply frequency scaling.
- An acceptable performance loss is to be chosen so that DVFS does not result in increased energy saving for compute-intensive applications or vice-versa.

For codes with large amounts of communication and memory access, CPU MISER can save up to 20% energy with 4% performance loss. For codes that are CPU-bound (e.g., EP), CPU MISER saves little energy since reducing processor frequency would impact performance significantly. Thus, the dynamic and transparent characteristics of CPU MISER are more amenable to use in systems with changing workloads.

4.2 Safety and Security Requirements

To secure the system the Intel user developer's manual needs to be consulted to get the register and the id values for the particular model of the Intel processor as writing a wrong value to the registers can cause issues.

Care must be taken while measuring the power consumption of the node components as user can experience minor electricity shocks if proper attention is not being given.

5. OTHER REQUIREMENTS

DATABASE REQUIREMENTS

We require the number of P states, their id's register addresses.

APPENDIX – A

DATA DICTIONARY

Overlap factor: used for out of order processors. When the stalls overlap with executed instructions, the process simply won't wait, therefore this factor has to be taken into account;

Instructions retired: those instructions which are executed in the given amount of time;

Number of Memory access delay: Calculates the Off-Chip workload delay in a time-slice of 250ms;

Operating Frequency: It is the scaled frequency which we obtain after DVFS Scheduling.

APPENDIX – B

The research papers we found, presented the methodology, design, and evaluation of performance-directed, system-wide, run-time DVFS schedulers for high performance computing. It evaluated a run-time DVFS scheduler designed with the proposed methodology on a real power-aware cluster. The experimental results showed that NPB benchmarks save up to 20% energy when using CPU MISER as the DVFS scheduler and that performance loss for most applications is within the user-defined limit. This implies that the methodology presented in the paper is promising for large-scale deployment.

However, also note that further tuning for CPU MISER is possible and the subject of future work, given that CPU MISER is built upon a generic framework and is transparent to both users and applications, it can be extended to many power-aware clusters for energy savings.