Appendix A

Perfmon2 Interface

Perfmon2 [122] is an interface developed for monitoring the performance of programs running on computing systems. This interface provides APIs and tool to access the hardware performance counters of the processors from user-mode. The perfmon2 interface has been developed for linux kernel version 2.6, and is available on the web [133] for download. The interface is uniform across all hardware platforms, i.e. it offers the same level of software functionality on each platform. The nature of the captured data depends solely on the capabilities of the underlying hardware. Performance monitoring using hardware performance counters of the processor does not require modification or instrumentation or recompilation of the program to be monitored.

The perfmon2 distribution comes with three software components as mentioned below, which can be downloaded from the perfmon2 project site on web [133].

- Patch of perfmon2 interface for linux kernel: This patch is to be applied on linux kernel source. The linux source could be downloaded from linux kernel project site on web [125]. After applying the patch the linux kernel need to be compiled and installed on the system. Before compiling the kernel, the hardware performance monitoring related options need to be enabled via make menuconfig command.

- Library for perfmon2, libpfm: This is a user space library. Once the system is booted with perfmon2 enabled kernel as mentioned in previous step, the library can be built and installed on it.

- Pfm tool for monitoring: After installing the libpfm as above, the Pf- mon tool can be built and installed on the system.

The output of Pfmon utility on Intel quad-core Xeon X5482 processor based
Experimental platform is shown in figure A.1. The operating system on the platform is Linux-2.6.30 kernel with perfmon2 interface [122]. The output is shown for command $ls$. The output shows counts of four events for command $ls$:

- **INSTRUCTIONS RETIRED**: It gives number of instructions retired.
- **UNHALTED_CORE_CYCLES**: It gives number of cycles consumed.
- **LAST_LEVEL_CACHE_REFERENCES**: It gives number of references to last level (L2) cache.
- **L2_LINES_IN_SELF_ANY**: It gives number of misses and prefetches to last level (L2) cache.

The events which can be measured on a processor are described in processor manuals [55]. The information about the syntax of command-lines for using the Pfmon tool is available on perfmon2 project site in the form of pfmon user-guide [134].
Appendix B

WEKA Machine Learning Workbench

WEKA (Waikato Environment for Knowledge Analysis) [112] is a suite of machine learning algorithms. It was developed at the University of Waikato, New Zealand. The algorithms can either be applied directly to a data-set or called from Java code. WEKA contains tools for data pre-processing, classification, regression, clustering, association rules, and visualization.

The WEKA distribution can be downloaded from the WEKA project site on web [113]. It provides mainly four types of interfaces – Explorer, Experimenter, KnowledgeFlow and Commandline. The WEKA manual provided with the distribution describes the utility of the interfaces.

The example output of WEKA Explorer is shown in figure B.1. The output shows prediction accuracy results of M5P (i.e. M5’ ) algorithm for predicting solo-run last level cache stress. The train-data-set has been generated from hardware performance counter data collected from Intel Xeon X5482 processor based experimental platform (specifications described in table 3.3 on page 45)). The test-data-set has been generated from hardware performance counter data collected from Intel Core2 6300 processor based experimental platform (specifications described in table 3.4 on page 45).

The Attributes: 5 shown in Classifier output panel of the figure B.1, refers to the five-tuple data instance i.e. four attributes plus the class variable solo-run last level cache stress (mentioned at page 51).
Figure B.1: Output of WEKA Explorer. The output shows prediction accuracy results of M5P algorithm for predicting solo-run last level cache stress. The train-data-set has been generated from hardware performance counter data collected from Intel Xeon X5482 processor based experimental platform. The test-data-set has been generated from hardware performance counter data collected from Intel Core2 6300 processor based experimental platform.