CHARACTERIZATION OF SPEC BENCHMARKS

1P. Anuradha, 2Hemalatha Rallapalli, 3G.Narsimha
1 Department of Electronics & Communication Engg.,
S R Engineering College Warangal,
Telangana, India.
2 Department of Electronics & Communication Engg.,
University College of Engineering, Osmania University,
Hyderabad, India
3 Computer Science & Engineering,
JNTUH College of Engineering Sultanpur,
Sultanpur Telangana, India.
lanu.rnitw99@gmail.com
April 29, 2018

Abstract
The architecture of processors is being developed and it is vital to progress expedient benchmarks that evaluate their performance. The SPEC CPU 2006 is one such benchmark that uses extensive workloads based on different real applications and is also the successor of the SPEC CPU 2000 benchmark. In the present work, workload characterization of SPEC CPU2006(C/C++) benchmark programs handling on Intel core2 Duo E4500 processor is performed. There is a need to understand the requirements of a memory system that came into view in workloads for a rational amount of time. SPEC CPU2006 consists of vast memory workloads which necessitate cache size with more than 4MB for better performance.
1 Introduction:

An integrated circuit is said to be a multi-core processor when two or more processors have been combined for better performance, reduced energy consumption and more efficient simultaneous processing of various tasks. Two processors installed in the same computer plugged into the same socket and the connection between them is faster in a dual-core processor. Generally, a dual-core processor is more powerful than a single core processor. In practice, performance gain for the dual-core processor is twice than that of a single core processor. Multi-core processing is growing rapidly which reach limits of complexities and speed. Systems with more number of processor cores are referred to as multi-core systems.

Benchmark is a standard of evaluation or measurement used to measure the performance of processors. A computer benchmark is generally a computer program which performs a set of operations, a workload and gives some form of results, a metric. It describes the performance of the tested computer. Computer benchmark metrics typically measure performance; how fast the workload finished, which is also called throughput and how many numbers of workloads which are completed in unit time. A comparison can be made by running the same computer benchmark on several computers. Ideally, own application with workload can be the best correlation test for frameworks. Utilizing own application with own workload it is unworkable to get a wide range of repeatable reliable and comparable measurements for various systems. Problems may be money, time, generation of a good test case, and difficulty ensuring comparable conditions, confidentiality concerns.

SPEC CPU2006 concentration on compute-intensive speed or performance which means that benchmarks mainly focus on the performance of the compilers, the memory architecture, and the computer processor (CPU). This paper gives a thorough analysis of the SPEC CPU2006 benchmark programs running on Intel Core II Duo processor [3] and emphasizes its memory system behavior.
and workload characteristics. SPEC CPU2006 benchmarks have longer execution time and larger input dataset than those of SPEC CPU2000. SPEC CPU2006 contains two suites which mainly focus on two different types of compute-intensive performance: The CFP2006 suite measures compute-intensive floating point performance. The CINT2006 suite measures compute-intensive integer performance, and each of the benchmark suites can be used to measure throughput or Speed \cite{2} using compilation method. CFP2006 has 17 benchmarks: 4 uses C++, 3 use C, 6 uses FORTRAN, and 4 uses a mixture of C and FORTRAN. CINT2006 contains 12 benchmarks: 9 use C and 3 uses C++. Some SPEC CPU2006 benchmarks are 999.specrand, 458.sjeng, 470.lbm, 429.mcf, 401.bzip2, 456.hammer, 482.sphnix, 410.bwaves, 445.gobmk, 435.gromacs. In addition, to build, compile, and run SPEC CPU 2006 benchmarks, it can also simulate using SimpleScalar; SimpleScalar \cite{6} is a set of different tools that give a virtual computer system with CPU, Memory and Cache hierarchy. Using the simple scalar different tools, modeling applications that simulate real benchmark programs running on a collection of modern processors and systems can be built.

The toolset includes simple simulators extend from a fast functional simulator (FFS) to a detailed, dynamically scheduled processor model. It supports non-blocking caches, branch prediction, and speculative execution. It also includes a collection of microarchitecture simulators which emulates the microprocessor at different levels of detail. Different SimpleScalar simulators are Sim-Fast, Sim-Profile, Sim-Safe, Sim-Outorder, Sim-Bpred, and Sim-Cache. Sim-Fast is a functional simulation optimized for speed. Sim-Profile is a Program Profiler that gives detailed profiles, by symbol and by its address. Sim-safe is a functional simulation which checks for instruction errors. Sim-Outorder is very difficult and detailed simulator which supports out-of-order issue and the execution. Sim-Bpred Simulate diverse branch prediction mechanisms and generate prediction hit and miss rate reports. It does not simulate the effect of branch prediction on total execution time. Sim-Cache is used for cache simulation, ideal for fast simulation of caches. SimpleScalar is very flexible, Extensible, and portable and gives better Performance. The specifications of Intel Core2 Duo machine is shown in table 1.

Table 1: Specifications of system
2 Methodology

Computer architectures are analyzed by running a workload on the processor and measuring the execution time. Nowadays modern computers are also designed in a similar manner [5]. However, as the computer doesn’t exist yet, it is not possible to execute the workload. This is where the workload characterization comes into the picture. The objective of workload characterization is to explain the properties of a given workload in terms of abstract performance or speed metrics which are called workload characteristics that predict the final performance [1]. There are, however, various aspects of the behavior of a workload and each of these aspects must be characterized. These aspects include the data and instruction memory access patterns, the amount of instruction level parallelism, the predictability of branch instructions, the types of ALU operations, etc.

The workload characteristics which are strongly interconnected. Characterization of SPEC CPU2006 benchmark suite based on performance considered all the integer and floating point programs. The complete details of the applications in the benchmark suite can be found in [2]. For implementing the SPEC CPU 2006 benchmarks the following steps are involved.

2.1 Compilation and run of SPEC CPU2006

SPEC also provides few tools to build or run the benchmarks on the system and to create desired binaries and all tools are located in the bin folder. We use those tools to build and run the applications. Using runspec tool SPEC CPU 2006 benchmarks can be run based on ref and test data sets provided by SPEC.
2.2 Simulation using SimpleScalar

Here simulation results have been shown with Sim-Outorder simulator using SimpleScalar along with the Simplesim-3.0 tool for the SPEC CPU2006 suite. SimpleScalar can be configured to simulate either PISA (Portable ISA) or Alpha ISA [7]. In this paper, simulations are carried out targeting alpha architecture. Other architectures can also be added to the SimpleScalar. The toolset takes binaries of different programs, compiled for the simplescalar architecture and simulates their execution on one of several provided simplescalar simulators. The system which runs for simplescalar is called the host machine or host while the ISA, the one that is targeting such as PISA or Alpha is called target. Microarchitecture events such as Instruction per Cycle (IPC), simulation speed, branches, loads and stores, il1 hits, il1 misses, il1 miss rate, dl1 hits, dl1 misses, dl1 miss rate, ul2 hits, ul2 misses, ul2 miss rate, itlb misses, itlb hits, itlb miss rate etc. have been measured.

3 SPEC CPU2006 BENCHMARKS RESULTS

CPU2006 benchmark programs have input dataset which is larger and also longer execution time. The execution time for CPU2000 benchmark programs ranges from 56-170 seconds, and for CPU2006 benchmarks range from 463-1610 seconds execution time on the Intel Core II Duo system. Compilation and run results using runspec tool with ref input as shown in table 2 and 3 based on base and peak, and also shown in figure 1 and 2. Each benchmark inputs, CINT/CFP type, and C/C++ language mentioned in table.4

Table 2: Compilation and run results (base)

<table>
<thead>
<tr>
<th>Benchmark</th>
<th>BenchRef</th>
<th>Run Time (s)</th>
<th>Speed Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>gzip</td>
<td>908</td>
<td>887</td>
<td>10.0</td>
</tr>
<tr>
<td>Mem</td>
<td>923</td>
<td>802</td>
<td>11.5</td>
</tr>
<tr>
<td>Spreadsheet</td>
<td>426</td>
<td>618(993)</td>
<td>63.66</td>
</tr>
<tr>
<td>Lbm</td>
<td>12740</td>
<td>1099</td>
<td>11.2</td>
</tr>
<tr>
<td>Spg</td>
<td>12100</td>
<td>977</td>
<td>12.4</td>
</tr>
</tbody>
</table>

Table 3: Compilation and run results (peak)
Comparisons between different computer hardware can be made by the source code of benchmarks provided by SPEC. SPEC permits two types of compilation, base metrics and peak metrics [2].

Table 4: Reference data inputs of SPEC CPU 2006 CINT/CFP benchmarks (C/C++ Language)

<table>
<thead>
<tr>
<th>Benchmarks</th>
<th>Input data set</th>
</tr>
</thead>
<tbody>
<tr>
<td>bzip2</td>
<td>Source</td>
</tr>
<tr>
<td>gzip</td>
<td>changelog</td>
</tr>
<tr>
<td>mem</td>
<td>compression</td>
</tr>
<tr>
<td>perl</td>
<td>output</td>
</tr>
<tr>
<td>Mad</td>
<td></td>
</tr>
<tr>
<td>Speed</td>
<td>Control</td>
</tr>
<tr>
<td>Lbm</td>
<td>Benchmark</td>
</tr>
<tr>
<td>Stc</td>
<td>ref.txt</td>
</tr>
<tr>
<td>Hmmer</td>
<td>mem tamaño</td>
</tr>
<tr>
<td>Browsie</td>
<td>solo warn</td>
</tr>
<tr>
<td>Wf</td>
<td>snoopy warn</td>
</tr>
</tbody>
</table>

The base metrics are required for all reported results and have stricter guidelines for compilation whereas the peak metrics are optional and have less strict requirements. Execution time for lbm
is more since it has larger input data sets. The Instruction per cycle (IPC) detail is shown in figure 3 using SimpleScalar. From the figure, it can be observed that mcf have low IPC. Figure 4 represents branch prediction accuracy of SPEC CPU 2006. CPU2006 benchmarks like bzip2 and gcc have comparatively high percentage branch prediction accuracy. The measurement and analysis of dl1 cache misses, il1 cache misses were carried out. Figure 5 and 6 indicates dl1 cache misses, il1 cache misses of CPU2006. In CPU2006 benchmarks, the lbm has highest dl1 cache misses. This is one of the significant reasons for its low IPC. Thus from the results analyzed so far we can conclude that the CPU 2006 benchmarks have larger data sets and require longer execution time. Researchers in computer architecture area show strong interests in performance characterization of CPU2006.

Figure 3. Instruction per cycle (IPC) of SPEC CPU 2006 Benchmarks

Figure 4. Represent branch prediction accuracy SPEC CPU 2006
4 CONCLUSION

In this paper, we have examined the emerging CPU2006 benchmark programs on Intel core II Duo processor. According to the obtained results, it shows that CPU2006 benchmark programs have longer execution time because of the larger input dataset. In addition to that, it also shows improved performance in L2 cache miss rate and branch miss rate for most of the programs because of its specific optimizations on Intel Core architecture.

References


