**NAME:** GBENRO TIMOTHY

**MATRIC NO:** 15/ENG02/027

**COURSE:** CSC 410 – COMPUTER SYSTEM PERFORMANCE EVALUATION

**ASSIGNMENT:** Assignment 2

**QUESTION**

Based on the perspectives of the operational laws you have studied, discuss the probability of achieving an accurate performance report using the three major evaluation techniques

**ANSWER**

Computer performance is the efficiency of a given computer system, or how well the computer performs, when taking all aspects into account. A computer performance evaluation is the process by which a computer system’s resources and outputs are assessed to determine whether the system is performing at an optimal level. To evaluate the performance of a system, established benchmarks are used to see if it is performing correctly. When evaluating the system results, several parameters are used to determine the result. Examples are latency, speed, throughput etc. Several laws have been derived which establish relationships between throughput, response time, device utilization, space-time products and various other parameters related to computer system performance. Standards, or points of reference, are then used against the parameters, and an assessment is given. This process is known as benchmarking. It is not an easy task to create benchmarks for assessing a systems performance. The primary challenge is that technological characteristics are constantly changing. This means that benchmarking must be constantly updated, too. This makes computer evaluation a complex process and techniques need to be adopted to ensure that the performance report of the assessed system is accurate.

Take for example, the complexity in processor design. Processors have increased its number of basic components (transistors) by a factor of two about every 18 months. Besides, in the past, the main parameter that designers cared about was execution time and area, but nowadays other performance metrics such as power consumption, power density, thermal characteristics and reliability are key performance metrics.

The most common approach to estimate the performance of a superscalar processor is through building a software model and simulating the execution of a set of benchmarks. The advantage of this method is that simulation results are often much better reproducible than direct measurement results and this approach provides a means of visualizing the effects of the performance evaluation. Since processors are synchronous systems, these simulators usually work at cycle-level granularity. To estimate execution time, they have a typical slowdown in the order of 105 - 106, which implies that simulating a single relative small program that takes 1 minute to execute requires about between 1 month and year to simulate. For estimating power consumption this slowdown is further increased. This slowdown is a main barrier for many types of experiments. Besides, the estimation of the delay and power consumption of each component is not trivial. In some cases, such as memory structures, relatively accurate analytical models have been developed. The analytical method demands a thorough understanding of the system. However, for some others, a layout synthesis of the corresponding circuit may be necessary. Because of the above, most new microarchitectural proposals are evaluated by simulating just a small section (100 million – 1 billion instructions) of several (around 10 – 20) benchmarks. This represents in the order of 10 seconds of activity of a real computer, which is certainly an important limitation if one considers the huge variety of applications that are run on computers. Besides, new applications are continuously being developed. This makes the task of choosing a small subset of representative applications to be very difficult, if not impossible. The other main problem that processor designers are facing is the relative simplicity of the performance evaluation tools. This is a result of the relatively small effort that research teams usually devote to develop performance evaluation tools. In most of the cases, research groups rely on public domain simulators that some groups have made available, but due to the continuous evolution of processor’s microarchitecture, these simulators become soon outdated. A related problem is the difficulty to reproduce results. Even though many groups share the same baseline simulation infrastructure, it is very difficult for a research group to reproduce results generated by another group, since usually this baseline simulator is modified in different ways by the different groups to make it more up to date. This shows that to an extent the results of evaluation reports will be prone to errors.

If we accept that we cannot have a small workload (in the order of seconds of execution time due to the typical slowdown of a simulator) that is representative of the dynamic evolving and large variety of programs that are run on a computer, then the conclusions we may obtain just by arbitrarily selecting a tiny section of a small subset of programs may be completely unrepresentative of the reality. The reliance on key metrics obtained using measurement techniques will be a better approach. The idea would be to define a set of benchmarks that represent the different possible types of codes that may be run on a computer. For that, we should first do the exercise of identifying which are the different workload parameters that have the most important effect on processor performance. Examples of these parameters are: size of the data and instruction working set, frequency of branches, predictability of branches, length and composition of the dependence chains of some critical instructions such as loads, etc. We could build synthetic benchmarks that would represent these scenarios and would be relatively small. Because of that, we could consider thousands of different scenarios. By the adoption of the three evaluation techniques it becomes obvious that where one fails, the other succeeds and a reliance on their combined strengths will prove the result of the performance evaluation to be of a greater degree of accuracy.