Author's Reply

February 7, 2017

Overview

I appreciate these quite thorough reviews. I appreciate the feedback and suggestions.

Reviewer

<< Reviewer's comments to the author(s) >>

The SEDONA is based on the probability that any daemons run during the measurement. And cutoff time is based on the actual measurement time. The proposed approach is interesting, but I cannot understand why the author employs such statistics and heuristic way, and how efficient the proposed approach will work. Please refer to my comments, and consider to revise the manuscript and re-submit .

Your concerns are well understood. Indeed, our statistics and heuristics greatly help us identify what daemons are infrequent, long-running on our system (as described in the fourth paragraph in the left column on page 3). Applying such statistics and heuristics can be done to any Linux system, which make our protocol generic. Furthermore, the experimental results show that our protocol is more efficient than the traditional method relying on elapsed time.

Note that my responses to your comments are incorporated into the revision and marked in red. There are some new and modified contents in the revision, which are marked in blue for better distinction.

I now respond to individually below.

Comments: (1) In the last of the manuscript, the author describes "Our plan is to integrate SEDONA into the query timing protocol [8]." If the research goal is the combination of the SEDONA and the query timing protocol, the proposed approach may be suitable. However, the goal is not clearly described in the manuscript. The readers will recognize the SEDONA is a generic protocol. If the research goal is the combination of this work and the prior work, the author should mention so.

I understand your confusion. The previously described plan was suggested as one of the possibilities in the future work. But that is not the research goal of this paper. As you already recognized, the goal of this paper is to propose a *generic, better* timing protocol against the conventional approach of utilizing elapsed time, to enable better timing results. This goal is successfully achieved by significantly reducing variability in timing as demonstrated in our various experiments. I now clearly state this goal, marked in red, at the last sentence in Section 1.

To avoid confusion, I removed the last mention. Also, the first paragraph on page 2, which used to provide an explanation of a possible combination of the query timing protocol with SEDONA, was removed.

(2) If the proposed approach is for a generic purpose, there is a strong doubt about usability. The improvement of the SEDONA is little. And the extent of the each benchmarks improvement described in Table 3 varies widely. Thus, the experimental results does not confirm whether the SEDONA is widely practical or is useful. The description to clarify the SEDONAs characteristics is needed.

Your concern is understood, but it is hard to agree with the comment that the improvement is little, for the following reasons. First, for a certain workload such as `434`, our protocol reduced the variability was reduced up to by about 10x (from 75 ms to 7 ms), which is *not* small. (Please refer to the last sentence in the first paragraph in the second column on page 4.) There were also some other workloads such as `410` and `445` in which the improvement reached about 3x, respectively. Furthermore, every benchmark's standard deviation by our protocol was less than that of the conventional method. The relative error of the benchmarks was on average improved by 1.5x as well.

Second, our protocol scaled well over increasing execution time. (Please refer to the second paragraph in the right column on page 4.) The relative error of our protocol was still smaller than that of the ORG method for a long-running benchmark like `436` and `454` as well as for a medium-length or a short-length one. Such scalability was still observed in our new experiment, which I describe shortly, as shown in Fig. 5.

Note also that the ET results by the ORG method were also based on the settings taking care of several identified timing factors as described in the third paragraph in Section 2. Under such a well-configured timing environment, our protocol further reduced the variability of execution time compared to the ORG method in such various benchmarks, which actually demonstrated its general purpose and wide usability.

Lastly, the results from a new experiment provide another strongly evidence that our protocol is widely practical and useful. In the new experiment we evaluated the performance of our protocol on some real-world programs like insertion sort and matrix multiplication (in column major). The experimental results showed that the performance gap between the original and our methods was increasingly widened, e.g., up to by 6x, as workload level increased.

Therefore, I strongly claim that these various experiments confirm the wide acceptance and usability of SEDONA as an alternative to the ET-based measurement technique.

(3) The improvement of the SEDONA is little. And the extent of the each benchmarks improvement described in Table 3 varies widely. Thus, the experimental results does not confirm whether the SEDONA is widely practical or is useful.

This review is basically the same as (2). Please refer to our response of (2) for this comment. One note. You may feel that the extent of the improvement varies widely, because the SPEC CPU2006 includes a set of very different compute-bound benchmarks. But even if so, we found that all the benchmarks in the suite benefited from our timing protocol, and in addition, for some workloads, our protocol significantly reduced their variability up to by 10x.

(4) Since the explanation of "dual-PUT" is not enough, I can not well understand the algorithm shown in Fig. 3. Similarly, what is 1st Halfs ET (ms) attached at Fig. 4(b), 4(c), 4(d) ? I understood only that the label is related with dual-PUT. These words are important in the manuscript. So, please keep in mind to write understandably.

Your point is well taken. The term of "dual-PUT" represents a pair of two consecutive samples of a run of a (regular) PUT. (The task length of that regular PUT is virtually doubled, extended to the corresponding dual-PUT.) More descriptions of the term "dual-PUT" are provided with a simple example in the second paragraph in the right column on page 2. In addition, the 1st half indicates the first element of the pair whereas the 2nd half the second element of the same pair. The label of "1st Half's ET (ms)" is renamed as "Odd Samples' ET(ms)," for better clarification. Please refer to the subsequent (third) paragraph.