Reviewer 1:

1. We agree that when task creation is parallelized it does not become such an important bottleneck. Our results demonstrate this case in the bodytrack application that introduces nested parallelism and does not benefit as much from our proposal. However, currently most applications support only one level of parallelism. This can be seen in the PARSEC benchmark suite that we use; from all the applications that we evaluate only bodytrack uses nested parallelism that parallelizes task creation. Even in this scenario, where task creation is highly parallel, TaskGenX manages to improve the baseline by up to 37%.
2. TaskGenX runtime system also maintains a decentralized TDG. Dependency tracking is being held in a distributed manner and whenever one task finishes execution, it is responsible for updating the TDG and also informing its successor tasks about the data that it has produced. It was not the authors’ intention to assume that dependency tracking is being held globally. However we understand that this is not clearly mentioned in the paper and we plan to modify the text accordingly.
3. We will clarify in our text that the task graphs in TaskGenX are completely dynamic. As tasks are being created the prior created tasks might be executing, finishing or deleted. The applications that we use do not take the core count in a direct way. Instead we select the appropriate block sizes in order to generate the appropriate amount of tasks that can exploit parallelism on 512 cores.

Thank you for the detailed corrections, we will address them in the paper.

Reviewer 2:

1. The tracing overhead of our simulator is less than 10%. Specifically our simulator is very similar to the one used in this paper: [1]. This publication evaluates the overhead and the accuracy of a trace-driven approach very similar to the one we have considered in our paper. We will update the text accordingly to give this information to the reader. This approach is accurate as long as there is no contention in the shared memory resources on a real system. We consider that analyzing performance degradation due to these contention effects is beyond the scope of this paper and has widely been studied. We consider that computationally-wise our study of the requirements of the specialized hardware is realistic. Beyond that, depending on the memory model there can be various different outcomes.

Thank you for pointing out the missing applications from the repository. We have already asked the developers for the updated link and we will correct it in the text.

Reviewer 3:

Thank you for your comments, we will rectify the text accordingly so that our contributions become clear from an earlier point in the paper.

Reviewer 4:

Thank you for your comments.