Report:

From the

* For 8 threads the optimum speedup would be 8x, is that what we are seeing?

No, sometime, we can see 8 threads is slow than 7. It is because, sometime, when thread reach a special number, parts of threads may got more tasks than other, which leads totally speed slow down.

* Is there any particular reason why we'd see diminishing returns at certain thread counts? e.g. 2? 4? 6? 8?

It is because, when the thread number rise, the tasks, each thread got, is equal to (total tasks / threads number). In the case, the mathematic model is n / x which lead each thread’s task number decrease diminishing.

* Why is the speed increase more on some scenes than others?

It is because, when threads are separated tasks, parts of thread may got more tasks than other. Sometime, it also may happened that tasks’ number less than thread. In the case, speed increase may be less or zero.

* discuss any speed anomalies and suggest reasons, e.g.
  + In same scene and same threads, threads speed may different, which may because that other application or OS capture resource as well, which leads threads speed slow down
  + In same scene but threads number increase, the threads speed still slowdown, which is because part of threads have to get more calculate than others. However, in second time, because distribution is dynamic, speed may be increased due to more reasonable distribution.
* Are any stages when run with 1 thread faster than the base single-threaded code?

Yes, all of stages.

* Are any tests with less threads faster than the same test with more threads?

Not at all

- does the block size make any difference?

Haven’t too much different

- What's generally the best technique?

From the Average Graphs, we can see the best technique is stage 1. It is because stage 1 has the most continually buffer order when thread process it.