Question: Why do you think some measurements are so erratic?

> Because it is impossible to know when a thread gets interrupted and for how long, some measurements take longer than others for no logical reason.

Question: Why is the speedup of low intensity runs with iteration-level synchronization the way it is?

> With iteration-level synchronization each thread locks the aggregate variable a lot more than thread-level synchronization, because aggregate variable is shared by all threads. Every time a thread gets a result value from a integration function call, it has to lock access to the aggregate variable used to increment the total result. Increasing the intensity increases the total work each thread must do in proportion to the number of times it must lock the aggregate variable., thus increasing the intensity eventually increases speedup of iteration-level synchronization.

Question: Compare the speedup of iteration-level synchronization to thread-level synchronization. Why is it that way?

> Unlike iteration-level synchronization, thread-level synchronization does not have to lock the total results variable every iteration since it has a local variable. Thread-level synchronization is more efficient than iteration-level synchronization because it only forces other threads to pause when a thread adds on it's partial result to the total result at the very end.