Test Your Parallelization Skills

Personal Data:

Your name:

Your company:

Your Email Address:

Your Phone Number:

Tell us a bit about yourself and your interest in parallelization:

1. What can be said about multicore processors?

□ Parallel processing improves performance

□ Using multicore processors increases power consumption

□ Multiple cores improve multitasking

□ Cache coherence makes programming simple

2. Is it reasonable to expect that sequential (non-parallel) programs to keep getting faster on multicore processors?

□ Yes – single-core performance is still improving at a steady pace

□ In general, no – it is more fruitful to improve performance through parallelism

□ No – single-core performance will degrade the more cores are added to a processor

3. Suppose you parallelized a program and managed to reduce its runtime from 10 s to 4 s. What is the resulting speedup?

□ 4

□ 2.5

□ 2

□ 0.4

4. If a program is 90 % parallel and 10 % sequential, what is, theoretically, the best speedup you can hope to achieve?

□ Unlimited

□ 90

□ 10

□ 9

5. Suppose you have just spent a lot of time on parallelizing a program to run on two cores. Is it possible to avoid spending the same amount of effort when adapting the program to run on four, eight, or more cores?

□ Yes – by trying to decompose the program into as many independent tasks as possible

□ No – a program must be parallelized with a specific number of cores in mind

6. When would you consider a parallel program to be scalable?

□ When it runs faster with more processors or cores

□ When using more processors or cores allows you to increase the problem size, and the program doesn't get slower

□ When the speedup with N processors or cores is exactly N

7. What are possible reasons for less than perfect speedups of parallel programs?

□ Communication and synchronization among the processes/threads

□ Many more independent tasks than there are processors/cores

□ An uneven distribution of tasks

□ A large sequential task that cannot be parallelized

8. What is a race condition?

□ A race condition is when multiple processes/threads compete against each other to finish first

□ A race condition is a serious error where the timing or ordering of events affects a program's correctness

9. Can parallelization introduce race conditions?

□ Yes – parallel programs are nondeterministic by construction

□ Yes – when using incorrect or insufficient synchronization

□ No – parallel programs are deterministic by construction

10. Describe how you would prevent multiple processes/threads from making conflicting changes to a shared resource.

Please save the interactive PDF on your computer and send it as attachment to contact@emmtrix.com.