

Username: Pralay Patoria **Book:** Pro .NET Performance. No part of any chapter or book may be reproduced or transmitted in any form by any means without the prior written permission for reprints and excerpts from the publisher of the book or chapter. Redistribution or other use that violates the fair use privilege under U.S. copyright laws (see 17 USC107) or that otherwise violates these Terms of Service is strictly prohibited. Violators will be prosecuted to the full extent of U.S. Federal and Massachusetts laws.

CHAPTER 6



Concurrency and Parallelism

For many years, the processing power of computer systems increased exponentially. Processors have been getting faster with every model, and programs designed to challenge the hardware resources of an expensive workstation were being ported to laptops and handheld devices. This era came to an end several years ago, and today processors are not exponentially increasing in *speed*; they are exponentially increasing in *number*. Writing programs to take advantage of multiple processing cores hasn't been easy when multi-processor systems were rare and expensive, and it hasn't turned easy today, when even smartphones ship with dual- and quad-core processors.

In this chapter, we shall embark on a whirlwind tour through the world of modern parallel programming in .NET. Although this modest chapter cannot begin to describe all the APIs, frameworks, tools, pitfalls, design patterns, and architectural models that parallel programming is today, no book on performance optimization would be complete without discussing one of the apparently cheapest ways to improve application performance—namely, scaling to multiple processors.