CHAPTER 7

Networking, I/O, and Serialization

Most of this book focuses on optimizing computational aspects of application performance. We have seen numerous examples, such as tuning garbage collection, parallelizing loops and recursive algorithms, and even by coming up with better algorithms to reduce runtime costs.

For some applications, optimizing only the computational aspect results in limited performance gains, because the performance bottleneck lies in I/O work, such as network transfers or disk accesses. In our experience, a considerable portion of performance problems encountered in the field is not caused by an unoptimized algorithm or excessive CPU utilization, but is due to an inefficient utilization of the system's I/O devices. Let us consider two scenarios in which optimizing I/O can result in performance gains:

- An application might incur a significant computational (CPU) overhead due to inefficient use of I/O, which comes at the expense of useful work. Worse, this overhead might be so high that it becomes the limiting factor to realizing the full potential capacity of the I/O device.
- The I/O device might be under-utilized or its capacity is wasted because of inefficient usage patterns, such as making many small I/O transfers or by failing to keep the channel fully utilized.

This chapter discusses strategies for improving I/O performance in general and network I/O performance in particular. In addition, we cover serialization performance and compare several serializers.