In this way, we get the speed of compiled classes even within the standard Java Applet Sandbox, and so users can be reassured that no matter what JPC is running, the JVM is enforcing the basic restrictions on executing code that make the activity safe in a dangerous world.

A nice side effect of the remote compiler is that as time goes on and many JPC instances make use of it, it builds up a library of compiled classes that it can share among the JPC instances. Thus the compiler server rapidly becomes a simple web server for previously compiled classes. Moreover, by counting the number of times each compiled class is requested, the server knows which classes are most used, and therefore which code to spend more time optimizing. Although this latter feature has not yet been implemented, we believe such focused optimization could offset the execution speed penalty suffered by JPC applet clients introduced by the network delay in loading classes. Thus, JPC applet clients might perform as well as normal JPC application clients where compilation is carried out locally.[‡]

## It Feels Better the Second Time Around

**Everybody knows that it is always possible to do a thing better the second time.**

—*Henry Ford*, My Life and Work

Developing in an academic environment comes with its own challenges, which are somewhat different from those in a commercial setting. In an academic environment, performance targets are mostly self imposed, which is both a blessing and a curse. Discipline is required on the part of the developers to keep the project on track and to prevent its focus from shifting. However, a free environment also allows ideas to be rapidly developed and tested to confirm or disprove their benefits. For the most creative and ambitious projects, this type of atmosphere is critical to eventual success.

The fact that the architecture of JPC has progressed as far as it has with such a small team of developers[§] is the result of an overriding attitude toward coding. As Figure 9-13 shows, in the life of the project over 500,000 lines of code have been written. Of this, only 85,000 survive to this day. There have been numerous rewrites of various sections of the emulation, including one complete purge and rewrite from scratch.

[‡] Indeed, when the compiler server and JPC Applet are connected via a decent 100 MBit LAN, there is virtually no perceived performance penalty due to network issues. The advantage of having compilation carried out elsewhere frees up CPU resources locally, and this seems to balance the network delay.

[§] The average team size during the major 30-month development phase was 2.5 programmers, not all Java experts.