Process placement is a technique widely used on parallel machines with heterogeneous interconnections to reduce the overall communication time. For instance, two processes which communicate frequently are mapped close to each other. Finding the optimal mapping between threads and cores in a shared-memory environment (for example, OpenMP and Pthreads) is an even more complex task due to implicit communication. In this work, we examine data sharing patterns between threads in different workloads and use those patterns in a similar way as messages are used to map processes in cluster computers. We evaluated our technique on two state-of-the-art multi-core processors and achieved moderate improvements in the common case and considerable improvements in some cases, reducing execution time by up to 45%.