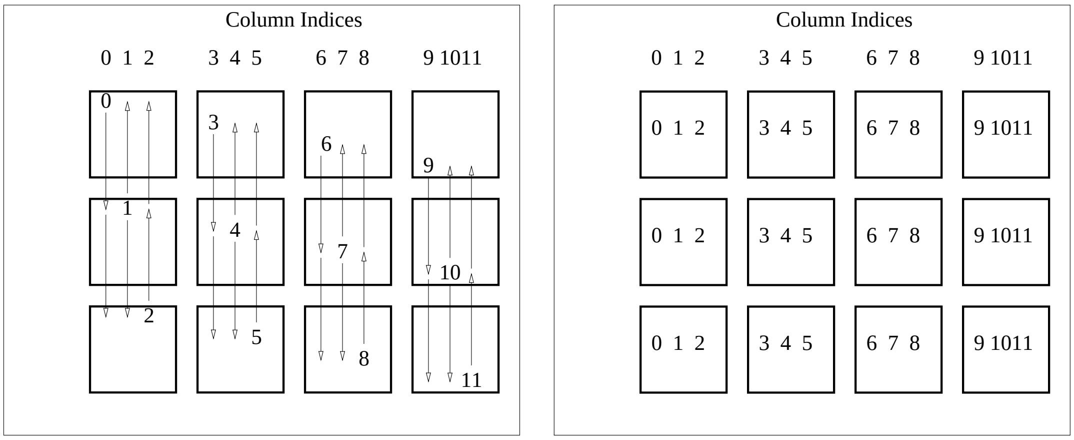

Figure 1.4
Spreading a vector within columns of nodes (or redistributing a vector like a duplicated vector, projected onto rows of nodes).

vector distribution, followed by a spreading of the vector within rows of nodes. Notice that this requires a **scatter** within columns of nodes, followed by a **collect** within rows of nodes. Spreading a matrix column within columns of nodes can be accomplished similarly.

It is important to note that the above observations expose a systematic approach to the required data movement. It naturally exposes the inducing vector distribution as an intermediate step through which redistribution of rows and columns of matrices can be implemented in a building-block approach. More on this in Chapter 5.

1.4.3 Reducing vectors, matrix rows, and matrix columns

Notice that after a vector, matrix row, or matrix column has been spread, multiple copies of the data exist. Often, as part of an operation, multiple instances of data exist, but each instance holds a contribution to a global result. These contributions must then be *reduced* (typically summed) into the global result. Often, the reduction needs only be performed (simultaneously) within one or all row(s) or column(s) of the node mesh. Looking ahead slightly, consider the situation in Figure 1.5(c). Each column of nodes holds a column vector, which must be summed *element-wise* into a single vector. The collective communication requires is a reduction that leaves a sub-vector of the result on each node. We call this a *distributed reduction.* We will discuss reduction (consolidation) of data again in Chapter 5.