在非集中式算法中定位接收站点有下述三种方案（定位发送站点与其类似）：

➤ 随机定位策略

随机定位策略是最简单的定位策略，因为它在决定将一个任务转移到何处时不需要在各场地间交换状态信息，不需要使用远程状态信息，任务被随机地选择发送到一个站点，因此不必彼此交换信息。但这样会带来一些问题，特别是当全系统的负载比较平均时，随机定位策略可能会使得所有站点都忙于转移任务而无暇去执行它们。为此，必须限制轮询的次数。最简单的定位策略是根本不使用（全局或局部）状态信息的策略。随机定位策略采用随机方式选择一个目的场点，被选定为要转移的任务将转移到该场点。目的场点按下述方式处理转移来的任务：如果该场点的等待队列的长度小于它所允许接收任务的阈值，那么它就像处理该场点上的任务一样处理转移来的任务；否则，它再用随机策略将该任务转移到其他场点。

这种策略可能会引起系统的不稳定性，即系统最终可能进入这样一种状态，其中的场点把它们的时间都花在任务的转移上而没有时间去执行任务。这种不稳定性可由下述的简单控制策略予以克服：利用一个静态转移限值 L_v 来限制一个任务的转移次数。一旦某任务已转移了第 $L_v - 1$ 次，那么转移 L_v 次的那个场点必须处理此任务，而不管该场点上的状态如何。

➤ 阈值指定定位策略

阈值定位策略先用随机方式选定一个目的场点，接着探查一下转移一任务到此场点是否会超越其规定的阈值。若没超越则将任务转移给它，若超越，则重复所述过程。当任务实际到达时，目的场点必须处理这个任务而不管其状态如何。继续这个过程直至找到一个合适的目的场点，或者探查的次数超过了静态探查限值 L_s 。若为后者，则原来的场点必须处理此任务。为了减少额外开销，必须限制轮询的次数。

显然，这种策略只需要使用目的场点上少量信息，同时避免了许多“无用的”任务转移。当然，这种策略不可能为一任务选择“最好的”目的场点。利用固定的探查限值而不采用广播通信方式主要考虑到降低执行负载共享算法时的开销。事实证明，这种策略使用较小的探查值（如 3 或 5）时的性能，差不多与使用较大的探查值（如 20）时的性能一样好。

➤ 最短队列优先定位策略

最短队列优先定位策略需要附加的系统状态并试图进行“最好的”选择。它先用随机方式选择 L_p 个不同的目的场点，然后分别考虑它们的等待队列的长度，最后将任务转移给其等待队列长度最短的那个目的场点。若此目的场点的等待队列长度大于或等于规定的阈值，则该任务应在原来的场点上处理，否则，当任务实际到达时，这个目的场点必须处理此任务而不管其状态如何。

最短队列优先定位策略不仅能决定哪些站点可以接收任务，还能决定哪些站点接收任务更好些。与前两种定位策略一样，它也要轮询，但却是选择具有最短任务队列的站点作为转移对象。