

## 第5章、 微机集群的性能优化

经过前面几章的学习和实践，我们已经建起规模或大或小的微机集群，并且已经通过了一些基本的、必要的测试。作为一个尝试性质的微机集群，比如两台微机的集群，学习并掌握前面几章的内容就已经表明，读者已经有能力自己组建超级计算机，并可以用作并行计算。但是，按前面三章内容组建起来的微机集群只是一个初级的并行计算系统。

这是因为微机集群中的微机和连接微机的网络等不是为并行计算的用途而发展、设计的。用微机建造微机集群具有传统超级计算机不具备的优势，比如部件的标准性、通用性、开放性和兼容性。使得微机集群的所有部件都具备多供应渠道，容易替代，容易扩充，配置灵活，因而也容易维护的特点。但是，微机集群的发展仍然受诸多因素，特别是普通网络设备通讯速度的制约。这使大型微机集群的发展遇到很大的困难。除特殊或专项应用外，用 100M 交换机集成 16 台以上规模的微机组成的集群就可能遇到通讯堵塞，有些计算任务甚至 8 台微机组成的集群都会引起通讯堵塞。因此，提高通讯速度对大型微机集群就非常重要。

此外，大型微机集群面临的另一个问题就是日常管理和维护。集成的规模一大，管理和维护就会成为一个突出的问题。这是因为每个结点并不是一个完全独立的系统，每个结点必须以一定的方式与其他所有结点发生联系。实际上，对传统超级计算机来说也会遇到同样问题。但是，传统超级计算机在设计时就可以考虑采用各种手段，专门解决这个问题。而微机本身不是为建造并行系统而设计的，这个问题就更加突出。

因此，如何在通讯速度上、在管理上接近或达到传统超级计算机的性能是大型微机集群发展必须解决的关键问题。针对这两个困难，作者将结合自己的实践，作为提高部分，在本章和下一章中介绍一些方案来解决。可以看到，这些都是在现有的硬件基础上的性能价格比较好的方案。

前面几章都是基本内容，我们的叙述尽可能详尽，使得没有学过 Linux 的读者，也能在一步一步手把手教的过程中，学会自己动手装超级计算机。而作为提高部分的内容，我们的叙述将比较扼要，需要读者对 Linux 比较熟悉。

当然，和其他独立的计算机一样，微机集群也有系统或软件的升级、用户管理等任务。如果微机集群使用了网络信息服务 (NIS)，那么用户管理就和独立的计算机一样；服务器上的系统或软件的升级这类管理也是与独立的计算机相同的，而对于微机集群所有的节点机也需使用的软件，如果使用了网络文件系统 (NFS)，一般可以将这类软件安装在共享的硬盘上，那么它们的升级也只需在服务器上进行，然后通过 NFS 共享，所以也与独立的计算机上的安装和升级一样。

服务器的系统管理可以参见前面两章的相关内容。SuSE Linux 有一个很好用的管理工具 YaST，系统大部分的管理工作都可以通过 YaST 完成。我们这里介绍的主要是 YaST 管理以外的一些内容，和一些基本的管理手段，包括开机和关机，等等。就是对于一些较小的集群，这样的管理也可以起到事半功倍的效果。

大部分管理工具都是一些 shell 脚本文件，如果读者对此不太熟悉也没有关系，可以