2022年南京信息工程大学第十六届数学建模竞赛赛题

（请先阅读“南京信息工程大学校内数学建模竞赛论文格式规范说明”）

**C题 超级计算机路由策略的优化设计**

超级计算机是由高性能处理器结合高速通信网络并具备超高速运算能力的大型计算系统，通常也被称之为高性能计算集群。我国目前在全球高性能计算领域具有较高的水平，并在世界超算Top500榜单中占据着重要的地位。目前全球主要大国在超级计算领域正在向着E级(每秒百亿亿次)的里程碑迈进。超级计算机的通信网络可以分为两类网络，一类是非直连网络，即通过的交换机进行数据交换，另一类是直连网络，在这种架构中，计算节点不仅承担计算的功能，而且负责数据的转发。目前大部分超算采用非直连网络，例如我国的“天河2号”，“神威-太湖之光”均使用的Fat-tree（胖树）架构，但是随着世界各国竞争E级计算的制高点，传统非直连网络中的交换机难以满足众多计算节点海量的数据交换需求，因此直连网络成为了关注的对象。目前世界第一超算日本的“富岳”采用直连网络中典型Torus网络。此外，有部分资料表明，我国目前部分E级计算原型机开始采用直连网络结构。

图1给出了某高性能计算集群互联网络的拓扑图，在该网络中为实现节点1和节点7间的通信，采用最短路径下，需要通过节点6进行转发，那么1→6→7转发的过程被称之为网络的路由，而任意两节点间的转发路径则构成路由表。需要注意的是，整个网络中任意两节点间最短路的路径条数可能并不唯一，例如节点7和15之间就存在两条最短路径7→3→15和7→11→15。当前，随着高性能计算程序的复杂度越来越高，互联网络所承载的数据量也在不断增加。在进行大量数据交换时，若部分节点承担过重的转发负荷，其瓶颈将导致整个网络传输效率的下降。因此在设计路由表时，除了考虑最短路径（本题不考虑多条最短路径同时传输的情况），也要考虑负载均衡问题。

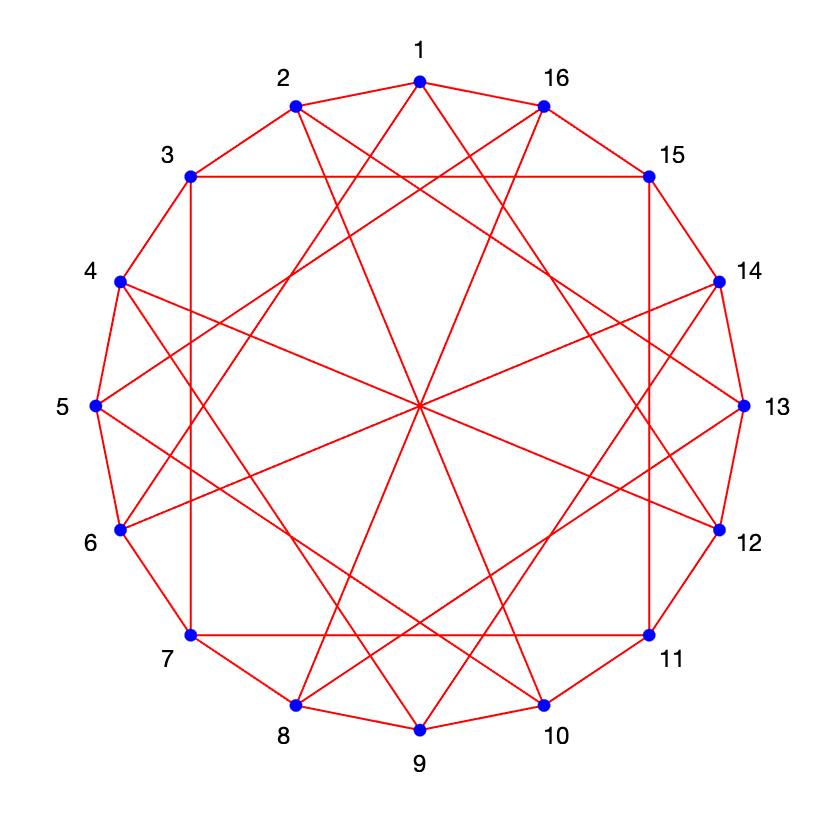


图1. 某集群通信网络对应的拓扑图

问题1：请计算图1拓扑中任意两节点间的所有最短路径，并给出最短路径条数最多的节点对，以及相应的最短路径条数。

问题2：在高性能计算程序中，通常采用MPI来实现并行计算程序在不同节点上数据交换。MPI不仅可以实现单点对单点的通信，也可以实现单点对多点、多点对多点的同时通信，即集合通信。在MPI集合通信的All-to-all通信模式（网络中任意两节点间同时进行数据交换，且数据量相等）下，请针对图1的拓扑网络，构建数学模型设计考虑负载均衡（只考虑对转发节点造成的影响，对接收和发送节点的影响不计）的路由策略，并将每个节点承载的路由条数填入下表。

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 节点编号 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
| 承载的路由条数 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

问题3：在大规模并行计算程序中，节点与节点之间通信的数据量并不都是一样的，但是可以通过分析代码或者抓包测定相应的数据量。“附件1.xlsx”给出了某程序发送节点与接收节点间通信数据量归一化后的统计结果，请结合图1所示的拓扑针对该程序设计相应的路由策略，使得程序运行过程中网络的负载保持均衡，并将每个节点承载的路由条数填入下表。

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 节点编号 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
| 承载的路由条数 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

问题4：高性能计算程序通过把任务划分为不同的程序块派给不同的计算节点来运行，从而实现并行加速的效果。程序块的编号在MPI编程中被称为MPI Rank，“附件1.xlsx”给出的是MPI Rank 1-16按顺序分配在1-16号节点时，不同MPI Rank之间的通信数据量（MPI Rank之间的通信数据量一般是跟网络拓扑无关的）。结合图1的拓扑网络，请设计一种最优的MPI Rank的分配方式（Task Mapping），使得程序运行的通信代价最小（通信数据量较大的MPI Rank对尽可能分配在距离较近的节点上），并将任务分配结果填入下表。

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| MPI Rank | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
| 分配节点编号 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |