

# 寻找最快的升级路径

## 解决思路

为尽可能找到最快的升级路径，需要在一定时间内可能升级更多的节点，可以将问题转化为每一时刻寻找可升级的最多节点数。

对于集群中的节点和应用程序可以使用矩阵描述，假设集群中节点数为 $n_N$ ，应用程序数目为 $n_A$ ，二维矩阵  $M[i, j], i \in [1, n_N], j \in [1, n_A]$  表示 $j$ 个应用程序在第 $i$ 个节点上的数目。如下所示

$$M = \begin{array}{c|ccc} & App_1 & \dots & App_{n_A} \\ \hline Node_1 & 0 & 0 & 0 \\ Node_2 & 0 & 0 & 0 \\ \dots & 0 & 0 & 0 \\ Node_{n_N} & 0 & 0 & 0 \end{array}$$

中断约束  $\vec{b}$  可以使用向量描述，向量长度为 $n_A$ 。

$$\vec{b} = [ App_1 \quad \dots \quad App_{n_A} ]$$

为求解每一时刻可升级的最多节点数，可转化为如下方程表示:

$$\begin{cases} \vec{W} * M \leq \vec{b} \\ \max \sum_{i=1}^{n_N} W_i \end{cases} \quad \text{其中, } \vec{W}_{n_N} = [W_1, W_2, \dots, W_{n_N}] \text{ 表示权重向量, 且对于任意 } W_i \text{ 取值为0或1.}$$

求解每一时刻的最大权重和，则表示每一时刻可升级的最多节点数，因此可以使用全组合(回溯法)方式求解、或者多维0-1背包方式（动态规划）、或者采用最优化（整数规划）方法求解精确解。但是考虑到节点数目较多的情况，计算的时间复杂度太高，很难在较短时间(10min)内得到全局最优解，因此，需要考虑采用非全局最优的解，在本例中使用贪心法，时间复杂度约为 $O(n)$ ，实验结果也较为满意。

经过测试后，总体求解算法为当  $n_N \geq 100$  时，采用贪心算法求解；当  $n_N \leq 100$  时，采用精确求解方法，本例使用最优化方法求解。对于5000台节点&4000个应用的情况下的计算基本在1min内（绝大多数5s左右）可以计算完成。

## 如何使用本项目

- 本机安装Python (3.8及以上)和pip
- 通过Git克隆本仓库至本机，并在终端上切换到该项目根目录下
- 运行命令 `pip install -r requirements.txt` 安装依赖包
- 运行 `python main.py`