1. 从算法角度优化拓扑

BAMMA 算法在计算机簇群上可分为 4 个过程：

- 乘法计算
- 节点间必要数据交换
- 归约化及加和（不一定有）
- 收集结果

其中乘法计算过程被认为与簇上的拓扑结构完全无关，所以针对拓扑的 BAMMA 算法设计主要关注后三个过程：

- 节点间必要数据交换 $\Longrightarrow$ 针对$[\![\text{多个环嵌套}]\!]$的优化
- 归约化及加和 $\Longrightarrow$ 针对$[\![\text{多个二叉树嵌套}]\!]$的优化
- 收集结果 $\Longrightarrow$ 针对$[\![Broadcast\ \&\ Gather]\!]$的优化

因此，如果将一给定的 BAMMA 算法中的拓扑结构拆分为：

$$\tau_{BAMMA} = \left(\bigcup_{i=1}^{\alpha} Ring_i\right) \bigcup \left(\bigcup_{j=1}^{\beta} Tree_j\right)$$

则我们期待最适应该算法的拓扑结构为：

$$\{\tau_{cluster} | \tau_{BAMMA} \subset \tau_{cluster}\}$$

及其之中直径与平均距离最小的拓扑。

1.1. 针对环的优化方案

任何 BAMMA 算法都基于 P（核数）组（个）3D hypercubes 构建，因而其一定拥有相对两个矩阵缓存的 $(ml + ln)$ 个、拥有至少一个信号源、且面积为 1 的环拓扑。

而这些环可以：

- 合并为少于或等于$(ml + ln)$ 个、拥有至少一个信号源、且面积大于 1 的环；
- 拆分为多个只拥有一个信号源、且面积大于 1 的通信环；

因此，当节点间带宽充足时，我们将第一步合并后的环优先沿长度切分，即获得更多、面积较大、但长度更短的通信环，使得其适应拓扑更为灵活，$\tau_{cluster}$ 选择更多。

否则优先沿面积切分，获得更多高效、但是长度较长，不易于适应簇拓扑的通信环。

1.2. 针对二叉树的优化方案

二叉树结构本身即与环结构相悖，基于树结构构成的环，其长度与节点层数一致（节点最大连接数 $k = 3$ 时）。所幸，BAMMA 算法的 hypercube 结构中，z 向离散度和 x 向离散度和 y 向离散度间的积是有上限的：

$$variance_z \cdot variance_x \cdot variance_y \leq P \cdot (ml + ln)$$

而且用于归约的二叉树的深度约为：$\log_2 variance_z$，一般远小于$variance_z$。

所以一般在环与二叉树间寻找平衡点一般是可能的。

2. 从拓扑角度优化算法

从给定拓扑角度优化算法则遵循相似的思路：将已知拓扑拆分为多个环与二叉树，在从其中择取并合并为τ_{BAMMA}，再基于此构建 BAMMA 类算法。

从该角度出发的优势有：

1. 方便从二叉树中计算 z 向离散度；
2. 方便从环中计算 x 向及 y 向离散度；
3. 方便从三个离散度得知，在给定缓存空间限制[1]的情况下，算法的效能上限及优化方向在何处。

但同时其劣势有：

1. 计算集群的拓扑往往比较复杂，拆分环（或回路）会有过多解；
2. 难以将合适的子图相互独立开来。

[1] 无限制的话，选择到其它点总距离最小的一个节点作为根节点即可。