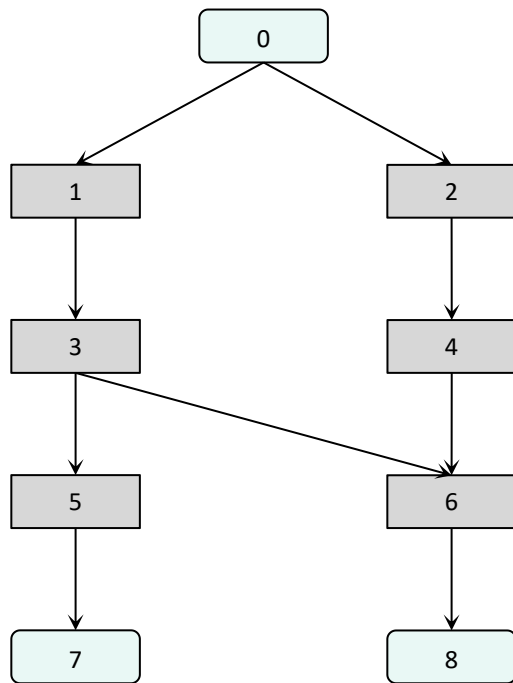


数据集一

数据集特征：

- 1、四个算力单元性能一致；
- 2、算力单元间通信开销全为1；
- 3、计算任务资源消耗各不相同；
- 4、仅一组样本数据；
- 5、9个计算任务部署在4个算力单元中，需要确保内存资源合理占用，并关注任务间依赖关系导致的通信开销；

input1.txt (computations, 计算图, 每个节点为一个计算任务)



计算任务ID	任务类型ID	计算量	所需内存	后续任务
0	0	5	1	1, 2
1	1	10	2	3
2	1	20	2	4
3	2	20	2	5, 6
4	2	20	1	6
5	3	20	1	7
6	3	15	3	8
7	3	5	1	
8	3	10	2	

sample.txt 输入样本

样本ID	任务类型ID	贡献度
0	0	1
0	1	2
0	2	3
0	3	3

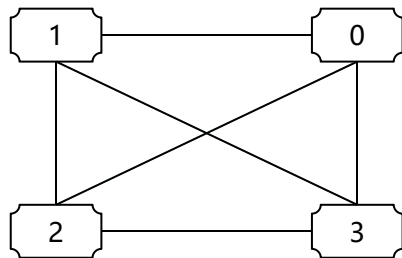
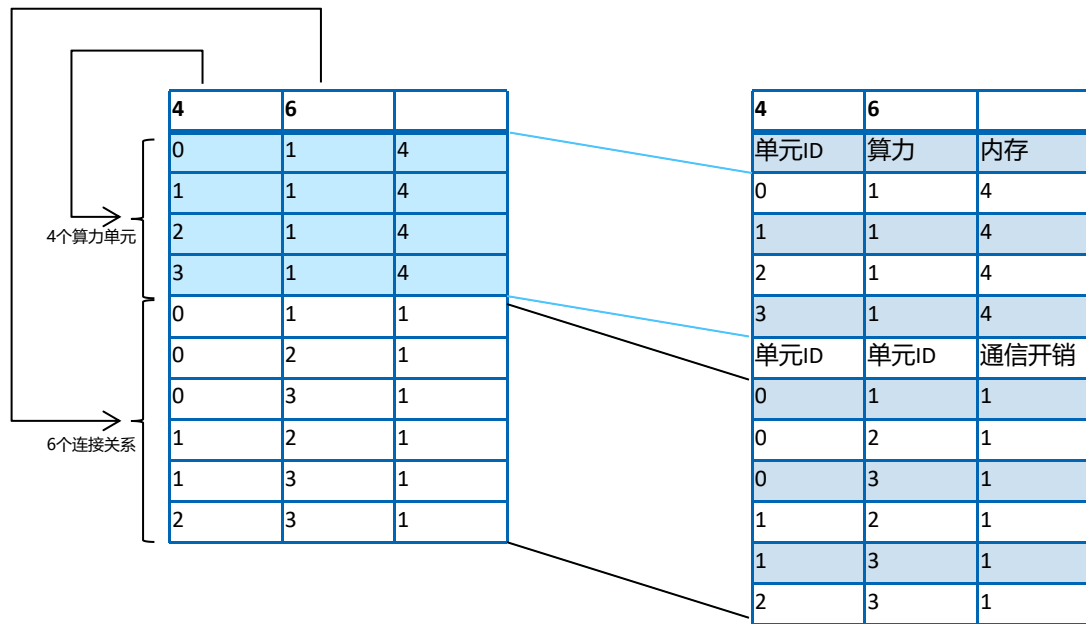
样本文件中的条目通过任务类型与计算任务匹配。

以样本0为例，如下图所示：

样本0在类型为0的计算任务上执行时，产生1点贡献度，
在类型为2的计算任务上执行时，产生3点贡献度。

计算任务ID	任务类型ID	计算量	所需内存	后续任务
0	0	5	1	1, 2
1	1	10	2	3
2	1	20	2	4
3	2	20	2	5, 6
4	2	20	1	6
5	3	20	1	7
6	3	15	3	8
7	3	5	1	
8	3	10	2	

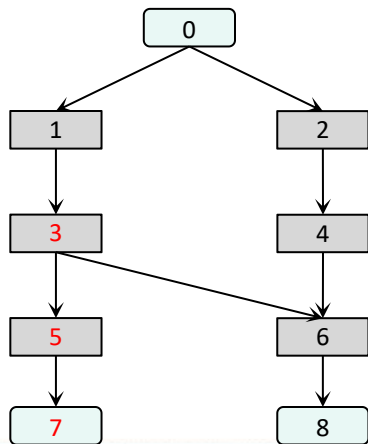
input2.txt 算力单元描述



output.txt

计算任务ID	算力单元ID	计算模式
0	0	0
1	0	0
2	1	0
3	1	0
4	2	0
5	3	0
6	2	0
7	3	0
8	3	0

计算任务3、5、7 共消耗4单位的内存，可以部署在同一个算力单元，局部通信开销比当前方案更优，但是任务6消耗内存为3，只能与任务0或任务4部署在同一个算力单元，总开销不变。



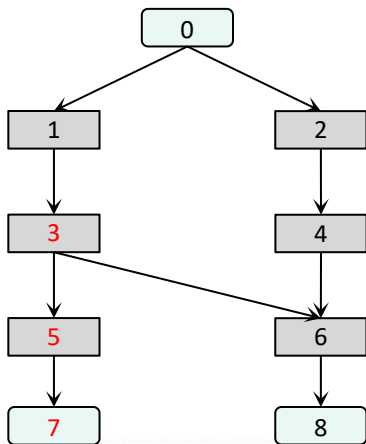
算力单元	0		1		2		3		
计算任务	0	1	2	3	4	6	5	7	8
时间0	样本0								
时间1		样本0	0->2传递						
时间2			样本0	1->3					
时间3				样本0	2->4				
时间4					样本0	3->6	3->5		
时间5						样本0	样本0		
时间6								样本0	6->8
时间7									样本0

output.txt

计算任务ID	算力单元ID	计算模式
0	0	0
1	0	0
2	1	0
3	2	0
4	3	0
5	2	0
6	3	0
7	2	0
8	1	0

计算任务3、5、7 共消耗4单位的内存，可以部署在同一个算力单元，局部通信开销比当前方案更优，但是任务6消耗内存为3，只能与任务0或任务4部署在同一个算力单元，总开销不变。

算力单元	0		1		2			3	
计算任务	0	1	2	8	3	5	7	4	6
时间0	样本0								
时间1		样本0	0->2传递						
时间2			样本0		1->3				
时间3					样本0			2->4	
时间4						样本0		样本0	3->6
时间5							样本0		样本0
时间6				6->8					
时间7				样本0					

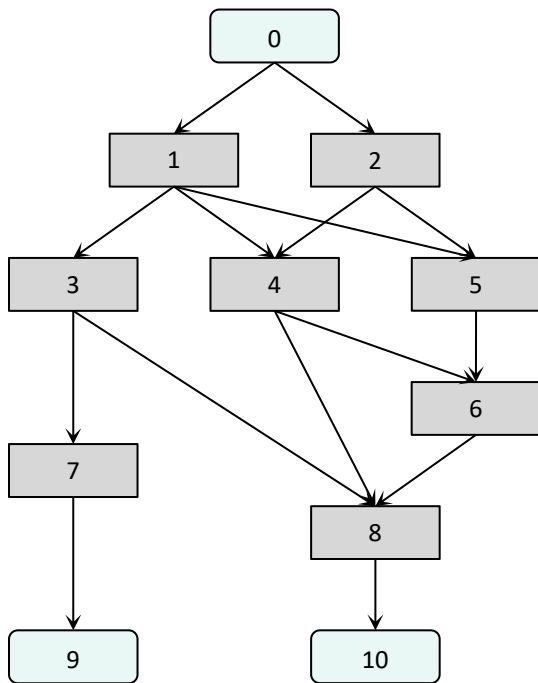


数据集二

数据集特征：

- 1、算力单元按性能分三类；
- 2、算力单元间通信开销有不同；
- 3、计算任务资源消耗各不相同；
- 4、支持包括剪枝在内的所有计算模式；
- 5、三组样本数据，需要注意计算任务调度顺序；
- 6、数据集包含多种影响编排的因素：
 - 算力单元间存在通信开销不同的情况；
 - 多组样本数据导致计算任务排队的情况需要注意调度顺序；

input1.txt (computations, 计算图, 每个节点为一个计算任务)



计算任务ID	任务类型ID	计算量	所需内存	后续任务
0	0	5	1	1, 2
1	1	10	2	3, 4
2	1	25	2	4, 5
3	2	25	2	7, 8
4	2	20	1	6, 8
5	2	20	1	6
6	3	45	3	8
7	3	45	3	9
8	3	45	3	10
9	4	30	1	
10	4	30	1	

sample.txt 输入样本

样本文件中的条目通过任务类型与计算任务匹配。

以样本0为例，如下图所示：

样本0在类型为0的计算任务上执行时，产生1点贡献度，
在类型为2的计算任务上执行时，产生3点贡献度。

样本ID	任务类型ID	贡献度
0	0	1
0	1	2
0	2	3
0	3	3
0	4	2
1	0	1
1	1	2
1	2	3
1	3	3
1	4	2
3	0	1
3	1	2
3	2	3
3	3	3
3	4	2

计算任务ID	任务类型ID	计算量	所需内存	后续任务
0	0	5	1	1, 2
1	1	10	2	3, 4
2	1	25	2	4, 5
3	2	25	2	7, 8
4	2	20	1	6, 8
5	2	20	1	6
6	3	45	3	8
7	3	45	3	9
8	3	45	3	10
9	4	30	1	
10	4	30	1	

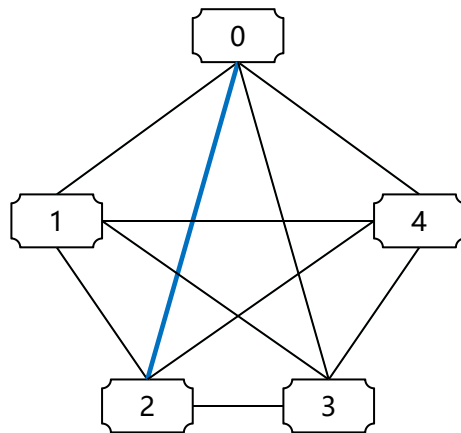
input2.txt 算力单元描述

5	10	
0	1	10
1	1	10
2	2	20
3	2	20
4	4	40
0	1	1
0	2	2
0	3	1
0	4	1
1	2	1
1	3	1
1	4	1
2	3	1
2	4	1
3	4	1

5个算力单元

10个连接关系

5	10	
单元ID	算力	内存
0	1	10
1	1	10
2	2	20
3	2	20
4	4	40
单元ID	单元ID	通信开销
0	1	1
0	2	2
0	3	1
0	4	1
1	2	1
1	3	1
1	4	1
2	3	1
2	4	1
3	4	1

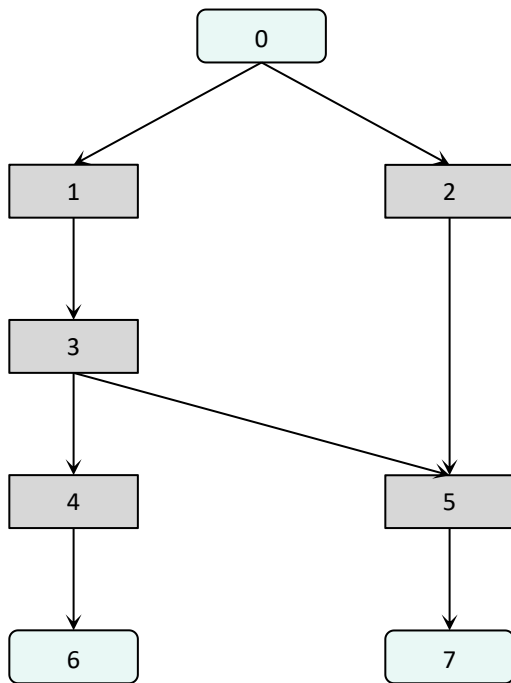


数据集三

数据集特征：

- 1、算力单元按性能分两类；
- 2、两组算力单元间通信开销为2，组间通信开销为1；
- 3、计算任务资源消耗各不相同；
- 4、仅一组样本数据；
- 5、注意通信开销和任务所需的内存资源，不一定要用完所有算力单元；

input1.txt (computations, 计算图, 每个节点为一个计算任务)



计算任务ID	任务类型ID	计算量	所需内存	后续任务
0	0	6	1	1, 2
1	1	10	2	3
2	1	20	1	5
3	2	20	2	4, 5
4	2	20	2	6
5	3	20	1	7
6	3	6	3	
7	3	16	1	

sample.txt 输入样本

样本ID	任务类型ID	贡献度
0	0	1
0	1	2
0	2	3
0	3	3

样本文件中的条目通过任务类型与计算任务匹配。

以样本0为例，如下图所示：

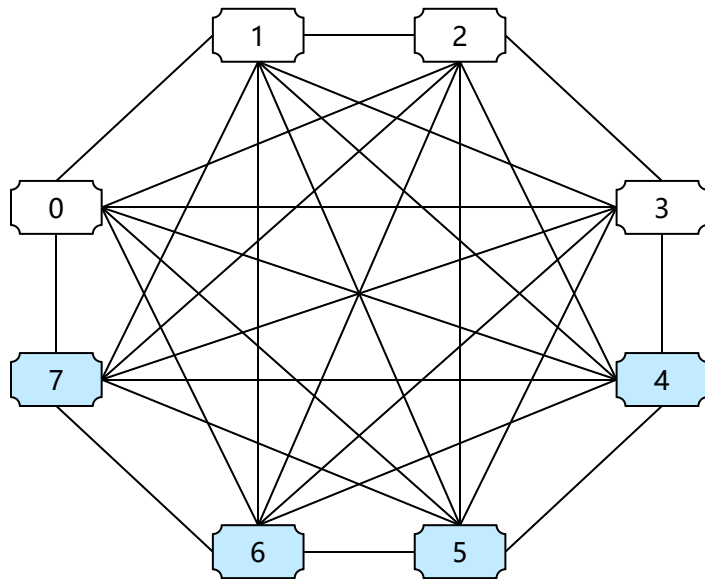
样本0在类型为0的计算任务上执行时，产生1点贡献度，
在类型为2的计算任务上执行时，产生3点贡献度。

计算任务ID	任务类型ID	计算量	所需内存	后续任务
0	0	6	1	1, 2
1	1	10	2	3
2	1	20	1	5
3	2	20	2	4, 5
4	2	20	2	6
5	3	20	1	7
6	3	6	3	
7	3	16	1	

input2.txt 算力单元描述

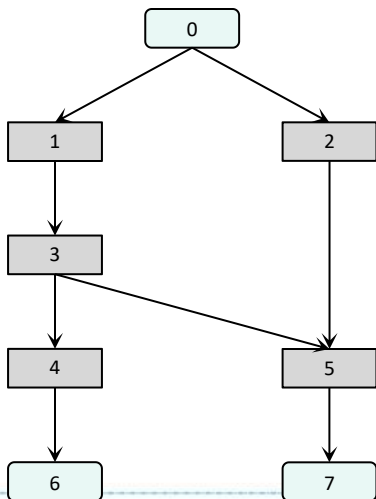
8	28				
0	1	2	1	6	2
1	1	2	1	7	2
2	1	2	2	3	1
3	1	2	2	4	2
4	2	4	2	5	2
5	2	4	2	6	2
6	2	4	2	7	2
7	2	4	3	4	2
0	1	1	3	5	2
0	2	1	3	6	2
0	3	1	3	7	2
0	4	2	4	5	1
0	5	2	4	6	1
0	6	2	4	7	1
0	7	2	5	6	1
1	2	1	5	7	1
1	3	1	6	7	1
1	4	2			
1	5	2			

8	28	
单元ID	算力	内存
单元ID	单元ID	通信开销



output.txt

计算任务ID	算力单元ID	计算模式
0	0	0
1	4	0
2	1	0
3	5	0
4	6	0
5	2	0
6	7	0
7	3	0



计算任务1和3一起部署在算力单元4上效果更好。

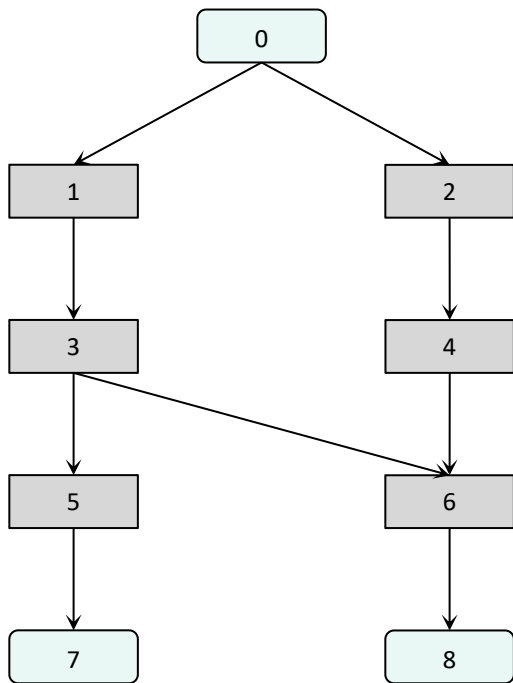
算力单元	0	1	2	3	4	5	6	7
计算任务	0	2	5	7	1	3	4	6
时间?	样本0							
时间?		0->2传递			0->1传递			
时间?		样本0			0->1传递			
时间?			2->5传递		样本0			
时间?						1->3传递		
时间?						样本0		
时间?			3->5传递				3->4传递	
时间?			3->5传递				样本0	
时间?			样本0					4->6传递
时间?				5->7传递				样本0
时间?				样本0				

数据集四

数据集特征：

- 1、仅两个算力单元；
- 2、算力单元间通信开销为2；
- 3、计算任务资源消耗各不相同；
- 4、仅两组样本数据；
- 5、注意任务组合策略，9个任务要分配在两个算力单元，保证内存足够；

input1.txt (computations, 计算图, 每个节点为一个计算任务)



计算任务ID	任务类型ID	计算量	所需内存	后续任务
0	0	6	1	1, 2
1	1	10	2	3
2	1	20	2	4
3	2	20	2	5, 6
4	2	20	1	6
5	3	20	1	7
6	3	16	2	8
7	3	6	1	
8	3	10	2	

sample.txt 输入样本

样本文件中的条目通过任务类型与计算任务匹配。

以样本0为例，如下图所示：

样本0在类型为0的计算任务上执行时，产生1点贡献度，
在类型为2的计算任务上执行时，产生3点贡献度。

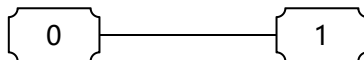
样本ID	任务类型ID	贡献度
0	0	1
0	1	2
0	2	3
0	3	3
1	0	1
1	1	2
1	2	3
1	3	3

计算任务ID	任务类型ID	计算量	所需内存	后续任务
0	0	6	1	1, 2
1	1	10	2	3
2	1	20	2	4
3	2	20	2	5, 6
4	2	20	1	6
5	3	20	1	7
6	3	16	2	8
7	3	6	1	
8	3	10	2	

input2.txt 算力单元描述

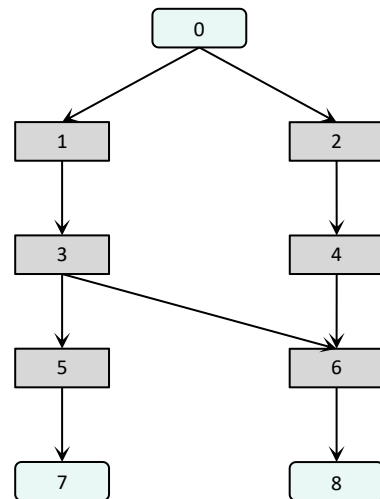
2	1	
0	1	6
1	2	8
0	1	2

2	1	
单元ID	算力	内存
0	1	6
1	2	8
单元ID	单元ID	通信开销
0	1	2



output.txt

计算任务ID	算力单元ID	计算模式
0	1	0
1	0	0
2	1	0
3	0	0
4	1	0
5	0	0
6	1	0
7	0	0
8	1	0



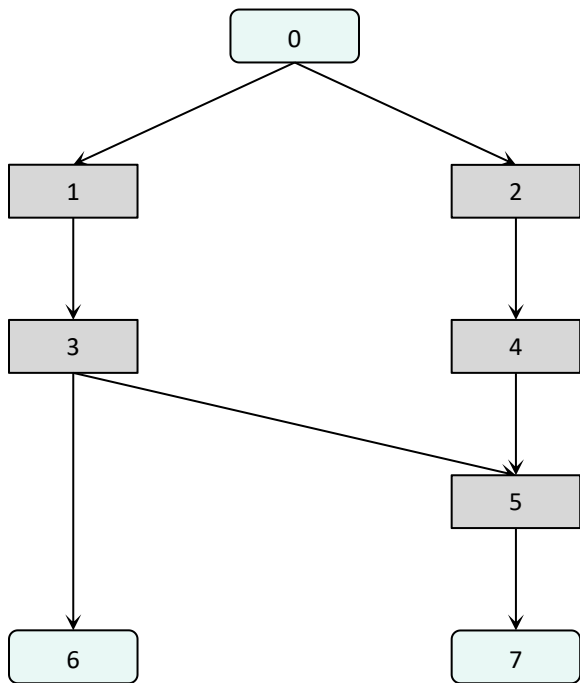
算力单元	1					0			
计算任务	0	2	4	6	8	1	3	5	7
时间?	样本0								
时间?		样本0				0->1传递			
时间?			样本0			样本0			
时间?							样本0		
时间?				3->6传递				样本0	
时间?				样本0					样本0
时间?					样本0				

数据集五

数据集特征：

- 1、8个算力单元性能相同；
- 2、算力单元间通信开销为0；
- 3、计算任务资源消耗各不相同，存在计算开销大且贡献度低的任务；
- 4、仅一组样本数据；
- 5、演示量化操作的应用场景；

input1.txt (computations, 计算图, 每个节点为一个计算任务)



计算任务ID	任务类型ID	计算量	所需内存	后续任务
0	0	10	1	1, 2
1	0	10	1	3
2	1	10	1	4, 5
3	1	40	1	4, 6
4	2	40	1	7
5	3	20	1	7
6	3	20	1	
7	3	10	1	

sample.txt 输入样本

样本文件中的条目通过任务类型与计算任务匹配。

以样本0为例，如下图所示：

样本0在类型为0的计算任务上执行时，产生1点贡献度，
在类型为2的计算任务上执行时，产生3点贡献度。

样本ID	任务类型ID	贡献度
0	0	300
0	1	400
0	2	100
0	3	500

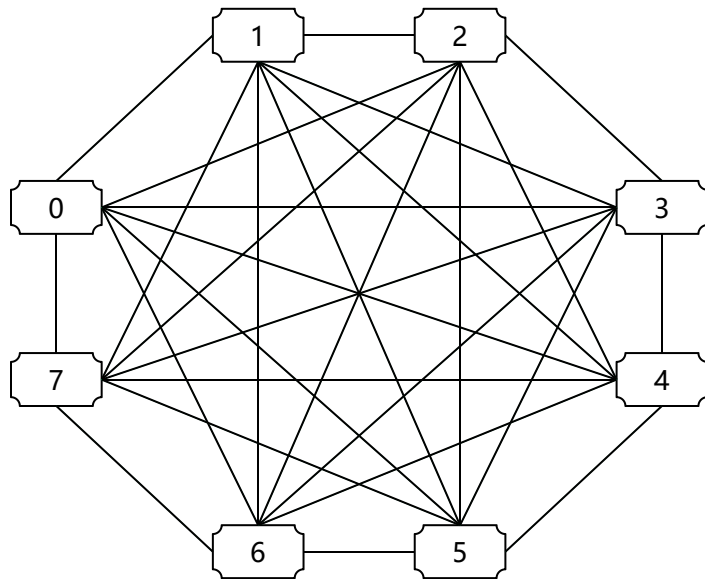
计算任务ID	任务类型ID	计算量	所需内存	后续任务
0	0	10	1	1, 2
1	0	10	1	3
2	1	10	1	4, 5
3	1	40	1	4, 6
4	2	40	1	7
5	3	20	1	7
6	3	20	1	
7	3	10	1	

input2.txt 算力单元描述

8	28				
0	1	100	1	6	0
1	1	100	1	7	0
2	1	100	2	3	0
3	1	100	2	4	0
4	1	100	2	5	0
5	1	100	2	6	0
6	1	100	2	7	0
7	1	100	3	4	0
0	1	0	3	5	0
0	2	0	3	6	0
0	3	0	3	7	0
0	4	0	4	5	0
0	5	0	4	6	0
0	6	0	4	7	0
0	7	0	5	6	0
1	2	0	5	7	0
1	3	0	6	7	0
1	4	0			
1	5	0			

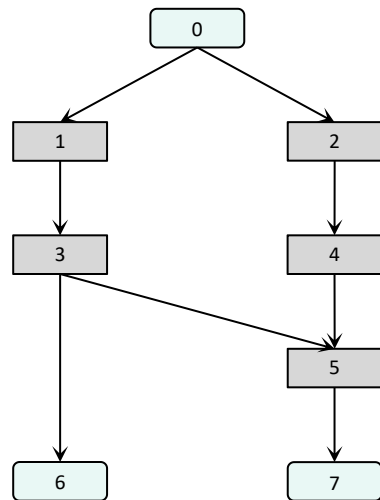
8	28	
单元ID	算力	内存
单元ID	单元ID	通信开销

算力单元性能相同，且无通信开销，
且每个算力单元因内存限制只能部署一个计算任务。
方便演示量化操作的应用场景。



output.txt

计算任务ID	算力单元ID	计算模式
0	0	0
1	1	0
2	2	0
3	3	0
4	4	0
5	5	0
6	6	0
7	7	0



最大贡献度 = 300 + 300 + 400 + 400 + 100 + 500 + 500 + 500 = 3000

计算模式全为 0 时:

运行时间 = 100 + 100 + 100 + 400 + 400 + 200 + 200 + 100 = 1600

贡献度比例 = 1

排行榜分数 = 1600 / 1 = 1600

计算模式	计算时间减少	内存占用减少	贡献度
0	0	0	100%
1	50%	50%	75%
2	75%	75%	50%
3	90%	90%	40%
4	100%	100%	0

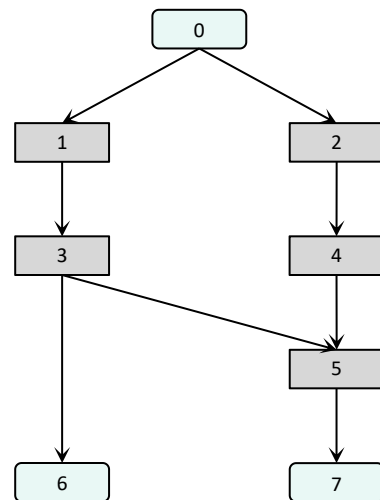
output.txt 包含量化操作

计算任务ID	算力单元ID	计算模式
0	0	0
1	1	0
2	2	0
3	3	1
4	4	3
5	5	0
6	6	0
7	7	0

注意：演示的量化方案
不一定是最优方案。

← 量化

← 量化



有通信开销的场景，减少
任务的内存占用后，可以
考虑合并多个计算任务到
同一个算力单元。

最大贡献度 = 300 + 300 + 400 + 400 + 100 + 500 + 500 + 500 = 3000

优化后贡献度 = 300 + 300 + 400 + **300** + **40** + 500 + 500 + 500 = 2840

未优化的运行时间 =

100 + 100 + 100 + 400 + 400 + 200 + 200 + 100 = 1600

运行时间 = 100 + 100 + 100 + **200** + **40** + 200 + 200 + 100 = 1040

贡献度比例 = 2840 / 3000

排行榜分数 = 1040 / (2840 / 3000) = 1098.592

(优于未优化时的排行榜分数 1600)

计算模式	计算时间减少	内存占用减少	贡献度
0	0	0	100%
1	50%	50%	75%
2	75%	75%	50%
3	90%	90%	40%
4	100%	100%	0