miniVite Weak Scaling

GeekPie_HPC, ShanghaiTech

There are some limitations of miniVite.

- Number of processes must be power of 2. If oversubscription is not used, we can only test the number of processes from 1 to 256. Because the maximum number of processes is $20 \times 20 = 400$.
- Number of vertices must be perfectly divisible by number of processes.

In weak scaling, both number of processors and number of vertices are increased. As a result, the number of vertices per processor is fixed. Because the complexity of Louvain method around $O(n \log n)$, the workload of each processor is almost fixed.

Assuming that the function of processors scaling is p and the work load scaling is w, to achive the weak scaling we must make the $w \propto p$. However, we can only change the input of vertices number which is not actually the work load. According to the Louvain method's complexity, the work load w is proportional to $n \log n$. As a result, we must vary the n to satisfy $n \log n \propto p$, resulting in the $n \propto \frac{p}{\log p - \log \log p + \log \log \log p \ldots}$, which is similar to p.

Therefore, to meet the requirements of weak scaling, instead of increasing the number of vertices by the number of nodes, we increase the number of vertices by the number of processes.

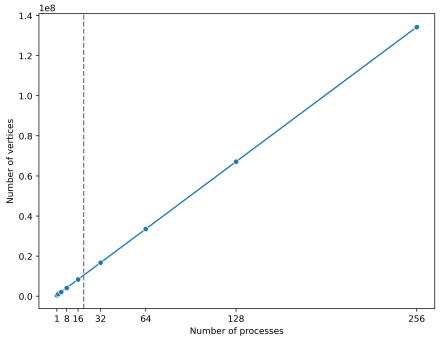
Results

We run miniVite on $1, 2, 4, 8, 16, 32, 64, 128, 256$ processes with 524288 vertices per process.

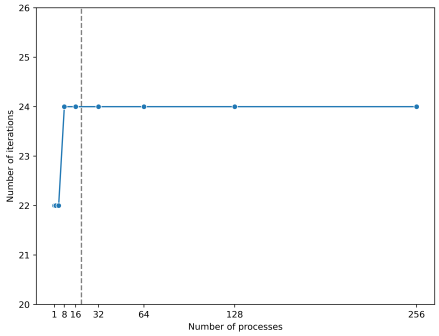
The maximum number of vertices is 134217728. The minimum number of vertices is 524288.

- MPI arguments: `mpirun -np {np} --hostfile hostfile -map-by core --bind-to core`
- miniVite arguments: `miniVite -n {vertices} -l -w`

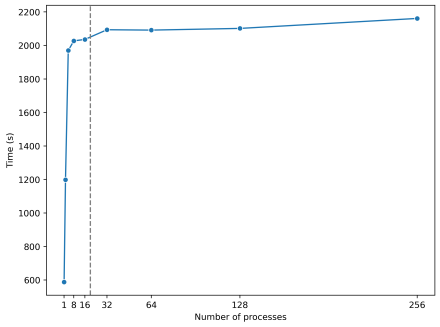
We also try to run miniVite with `-p 5` to randomly generated between processes.



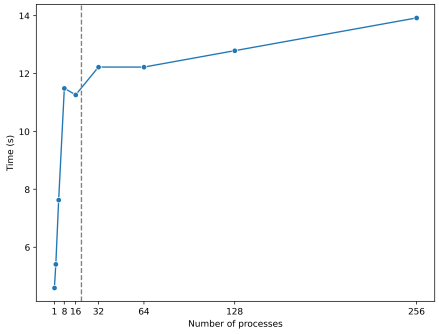
(a) Number of vertices



(b) Number of Iterations

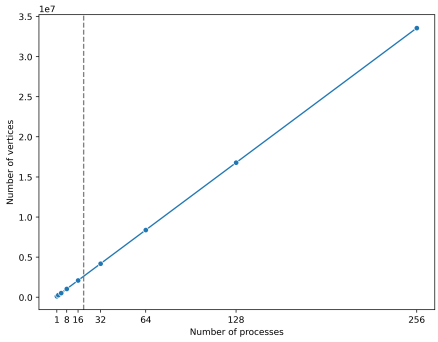


(c) Time to generate

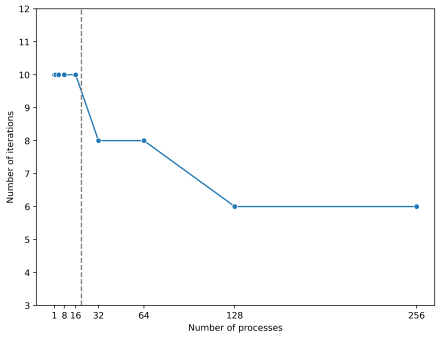


(d) Time to perform clustering

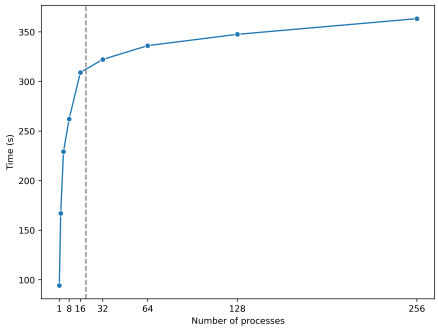
Figure 1: Weak Scaling Running Time With `-l -w`



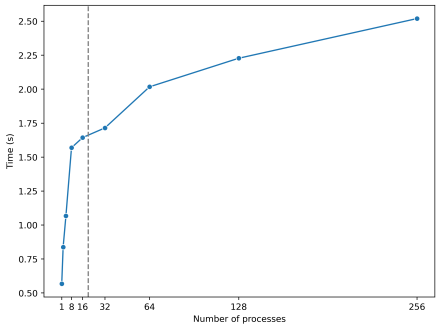
(a) Number of vertices



(b) Number of Iterations



(c) Time to generate



(d) Time to perform clustering

Figure 2: Weak Scaling Running Time With `-l -w -p 5`

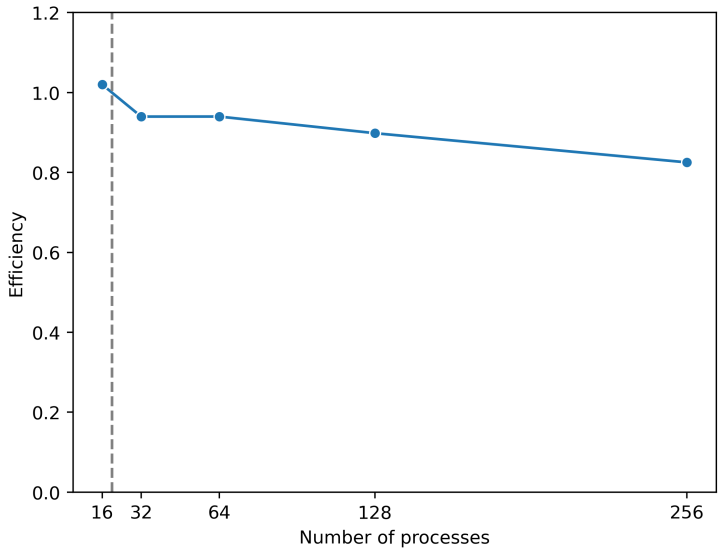


Figure 3: Efficiency of weak scaling

Analysis

As the number of processes changes from 16 to 32, the communication overhead increases. We compute the scaling efficiency of miniVite and draw the following figure.