Iterative methods

Marco Barbone

Summary

1. Jacobi method
2. Blocks methods
3. Asynchronous methods
4. Overlapping methods
5. Experimental results
6. Conclusions

Jacobi point method

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix}, \qquad \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, \qquad \mathbf{b} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}.$$

$$A = D + R \qquad \text{where} \qquad D = \begin{bmatrix} a_{11} & 0 & \cdots & 0 \\ 0 & a_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & a_{nn} \end{bmatrix} \text{and } R = \begin{bmatrix} 0 & a_{12} & \cdots & a_{1n} \\ a_{21} & 0 & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & 0 \end{bmatrix}.$$

$$\mathbf{x}^{(k+1)} = D^{-1}(\mathbf{b} - R\mathbf{x}^{(k)}),$$

$$x_i^{(k+1)} = \frac{1}{a_{ii}} \left(b_i - \sum_{j \neq i} a_{ij} x_j^{(k)} \right), \quad i = 1, 2, \ldots, n.$$

One row at the time!

Jacobi Method

function JACOBI(A, b, ϵ, K)
  $x^{(0)} \leftarrow$ initial guess
  $k \leftarrow 0$
  repeat
    $error \leftarrow 0$
    for i=0 to M **do**
      $\beta \leftarrow 0$
      for j=0 to M **do**
        if $i \neq j$ **then**
          $\beta \leftarrow \beta + a_{ij} x_j^{(k)}$
      $x_i^{(k+1)} \leftarrow \frac{b_i - \beta}{a_{ii}}$
    $error \leftarrow \frac{\| x^{(k+1)} - x^{(k)} \|_1}{N}$
    $k \leftarrow k + 1$
  until $error > \epsilon$ and k $< K$
  return $x^{(k)}$

Input:
  A linear system in matrix form Ax=b of size MxM
  $\boldsymbol{\epsilon}$ the maximum accepted error
  K the maximum number of iterations
Output:
  $\boldsymbol{x}^{(k)}$ the solution vector

Block method

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} & a_{16} \\ a_{21} & a_{22} & a_{23} & a_{24} & a_{25} & a_{26} \\ a_{31} & a_{32} & a_{33} & a_{34} & a_{35} & a_{36} \\ a_{41} & a_{42} & a_{43} & a_{44} & a_{45} & a_{46} \\ a_{51} & a_{52} & a_{53} & a_{54} & a_{55} & a_{56} \\ a_{61} & a_{62} & a_{63} & a_{64} & a_{65} & a_{66} \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \end{bmatrix}$$

$A_1 \quad A_2 \quad A_3$

$$x_j^{i+1} = A_{jj}^{-1}\left(b_j - \sum_{\substack{s=1 \\ s \neq j}}^{n} A_{js} x_s^{(i)}\right) \quad \text{j= 0 … n}$$

Overlapping Methods

Overlapping method

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} & a_{16} \\ a_{21} & a_{22} & a_{23} & a_{24} & a_{25} & a_{26} \\ a_{31} & a_{32} & a_{33} & a_{34} & a_{35} & a_{36} \\ a_{41} & a_{42} & a_{43} & a_{44} & a_{45} & a_{46} \\ a_{51} & a_{52} & a_{53} & a_{54} & a_{55} & a_{56} \\ a_{61} & a_{62} & a_{63} & a_{64} & a_{65} & a_{66} \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \end{bmatrix}$$

A_1 A_2 A_3

$$x_j^{i+1} = E_J[A_{jj}^{-1}\left(b_j - \sum_{\substack{s=1 \\ s \neq j}}^{n} A_{js} x_s^{(i)}\right)] \quad \text{j= 0 … n}$$

What is E?

Splitting matrix: $$\begin{bmatrix} 1 & & & & & \\ & {}^{1}/_{2} & & & 0 & \\ & & {}^{1}/_{3} & & & \\ & & & {}^{1}/_{3} & & \\ & 0 & & & 0 & \\ & & & & & 0 \end{bmatrix}$$

Asynchronous Methods

Jacobi Method

function JACOBI(A, b, ϵ, K)
    $x^{(0)} \leftarrow$ initial guess
    $k \leftarrow 0$
    repeat
        $error \leftarrow 0$
        for i=0 to M **do**
            $\beta \leftarrow 0$
            for j=0 to M **do**
                if $i \neq j$ **then**
                    $\beta \leftarrow \beta + a_{ij} x_j^{(k)}$
            $x_i^{(k+1)} \leftarrow \frac{b_i - \beta}{a_{ii}}$
        $error \leftarrow \frac{\|x^{(k+1)} - x^{(k)}\|_1}{N}$
        $k \leftarrow k+1$
    until $error > \epsilon$ and k $< K$
    return $x^{(k)}$

Input:
- **A** linear system in matrix form Ax=b of size MxM
- $\boldsymbol{\epsilon}$ the maximum accepted error
- **K** the maximum number of iterations

Output:
- $\boldsymbol{x}^{(k)}$ the solution vector

Local error checking

Remove the synchronization point

Minor optimization

- LDLT Cholesky factorizations in order to avoid the inverse
- Vectorization

Methods

- Parallel
- Parallel Async
- Blocks
- Blocks Async
- Block Optimized
- Overlapping
- Overlapping Async
- Overlapping optimized

Datasets

- Random matrices of size 4096, 16384, 32768, 65536
- Equilibrium distribution of a Markov chain transition matrix of size 910596

Metrics

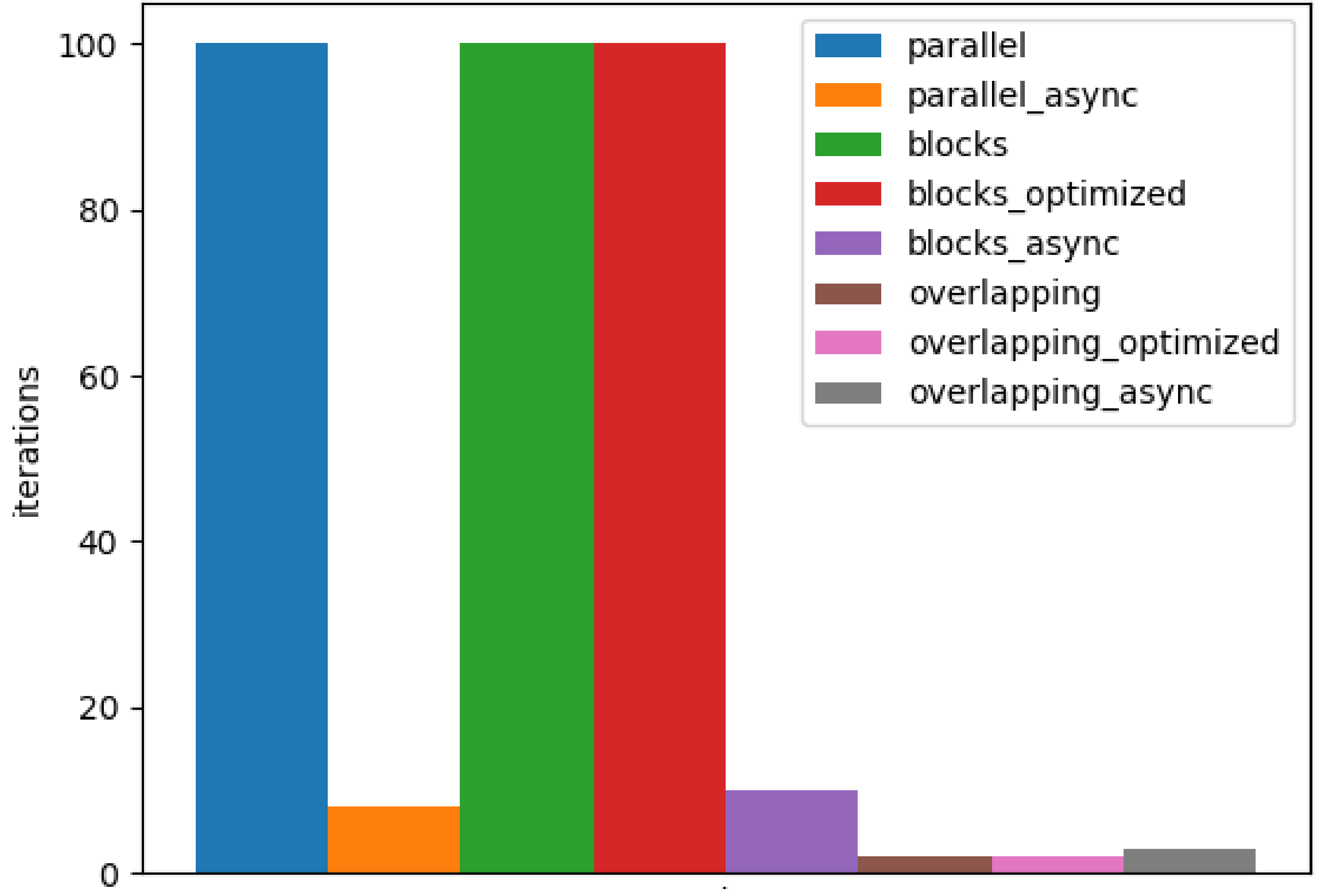
- Scalability
- Wall clock time
- Benchmark

Parameters

- Max number of iterations 100
- Number of physical cores 24
- Error tolerated 10^{-10}

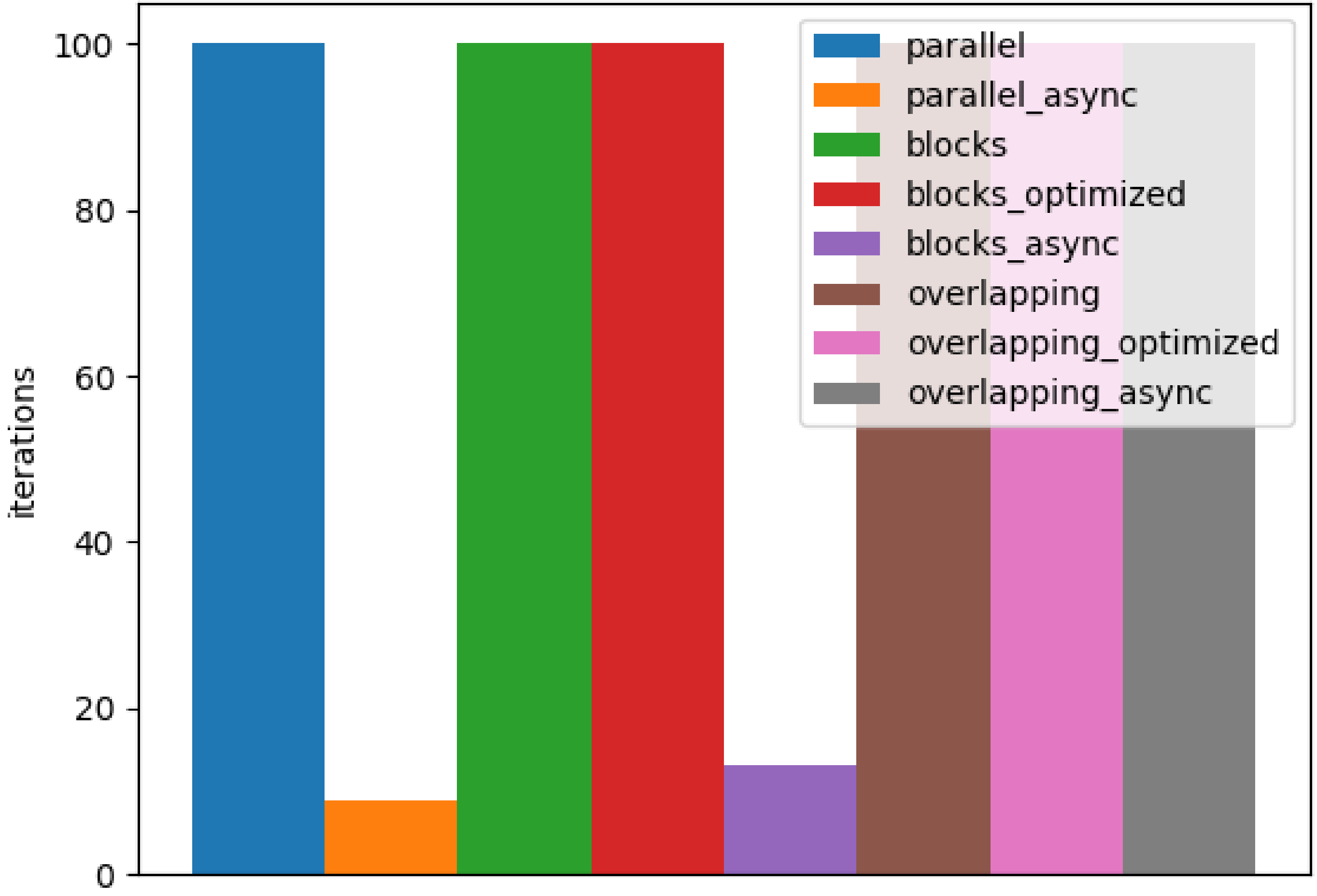
Convergence

Random sparse matrix



Size	65536
Block Size	256
Workers	24

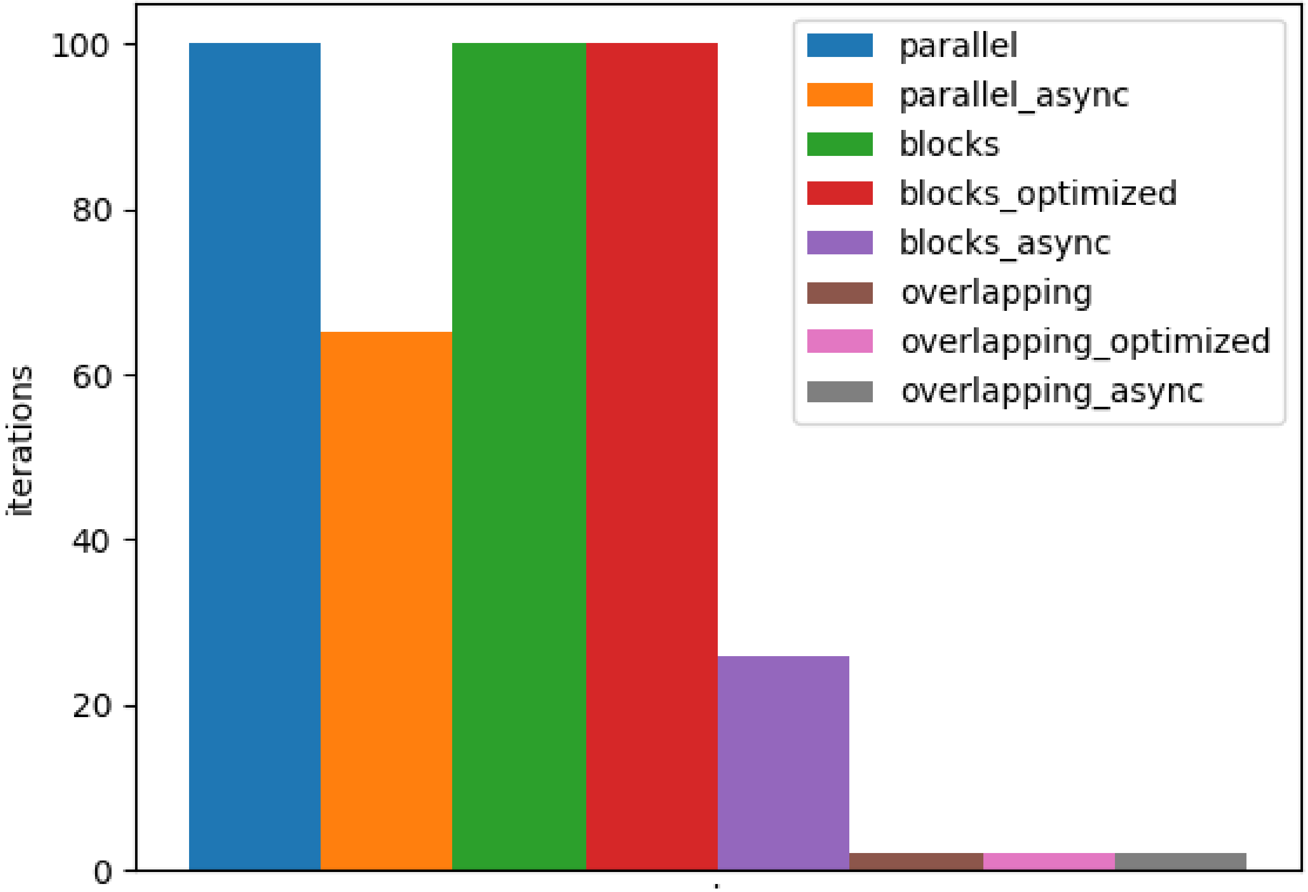
Random sparse matrix



Size	16384
Block Size	128
Workers	24

Blocks too big!
Slow convergence!

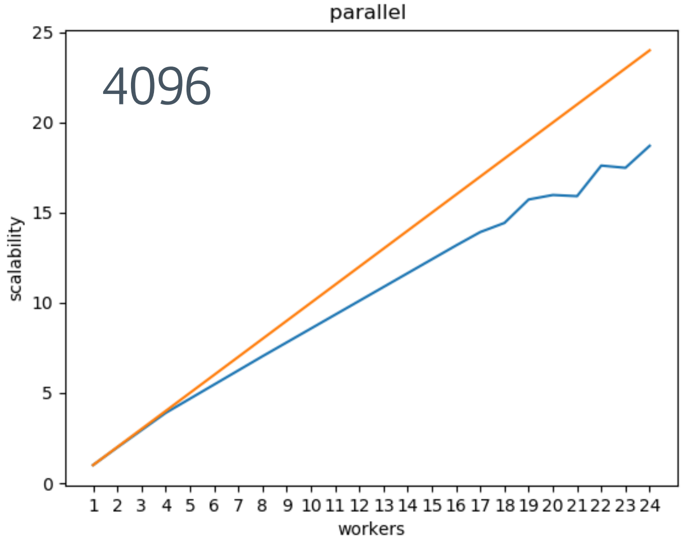
Markov transition matrix

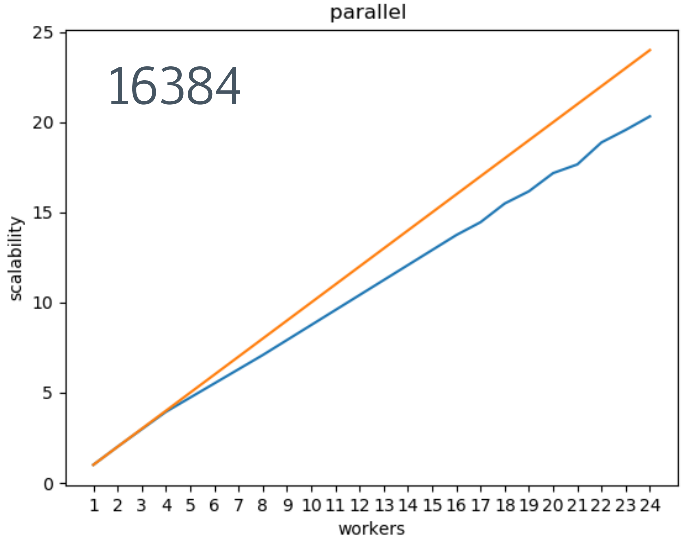


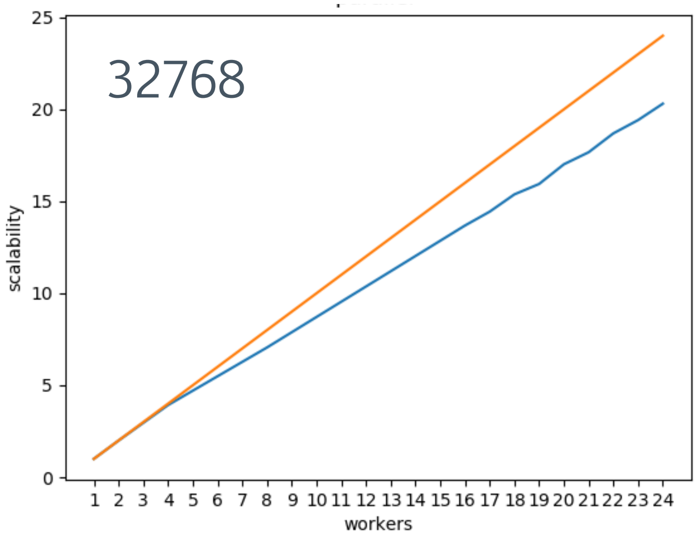
Size	910596
Block Size	512
Workers	24

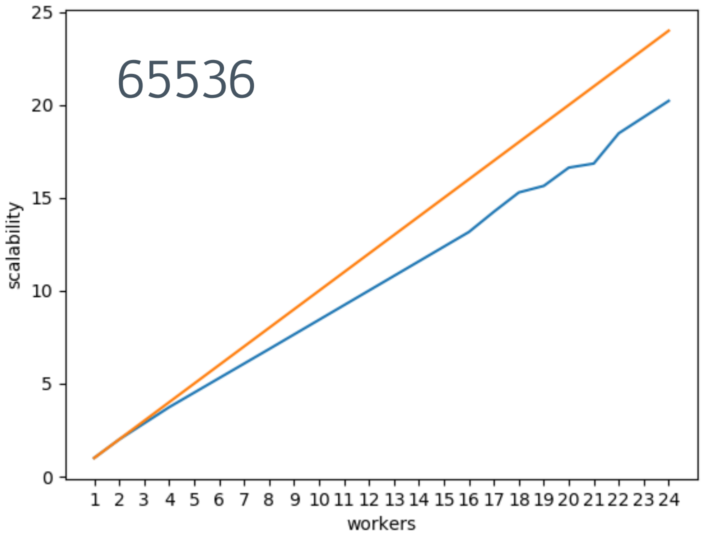
Scalability

Parallel

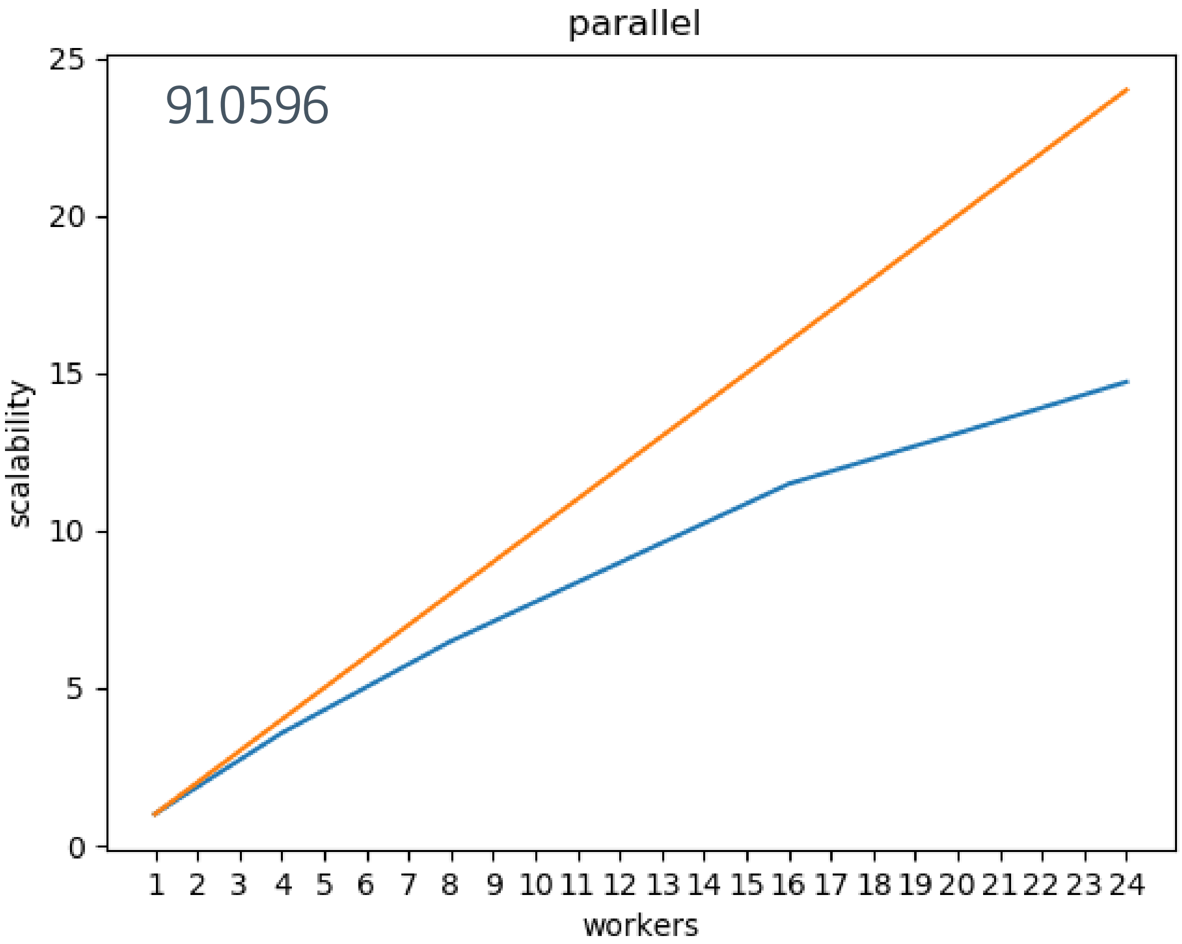

parallel
4096
scalability
0 5 10 15 20 25
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24
workers


parallel
16384
scalability
0 5 10 15 20 25
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24
workers

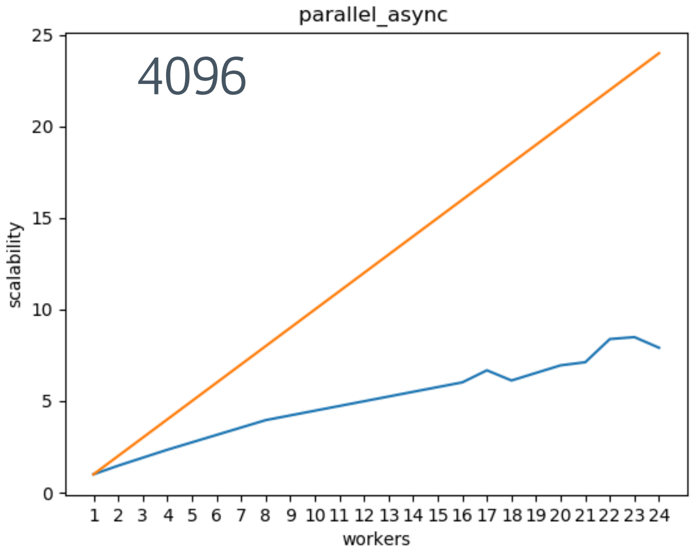

32768
scalability
0 5 10 15 20 25
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24
workers

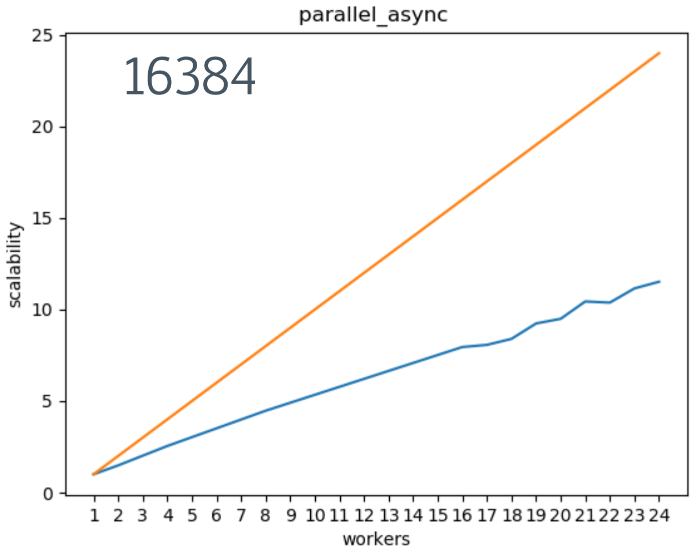

65536
scalability
0 5 10 15 20 25
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24
workers

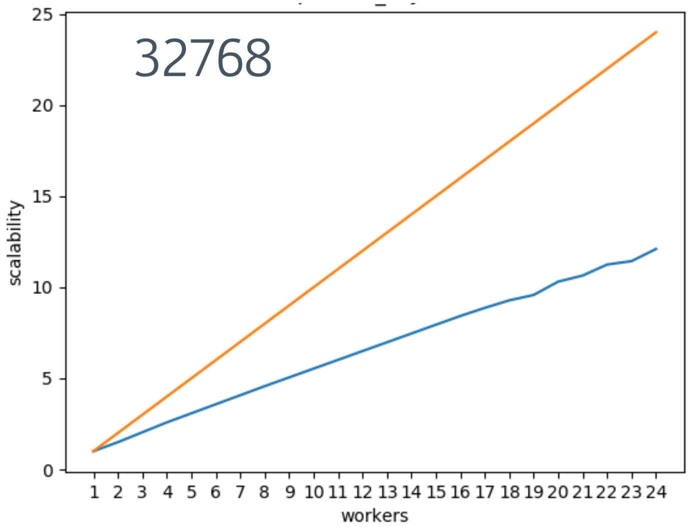
Parallel

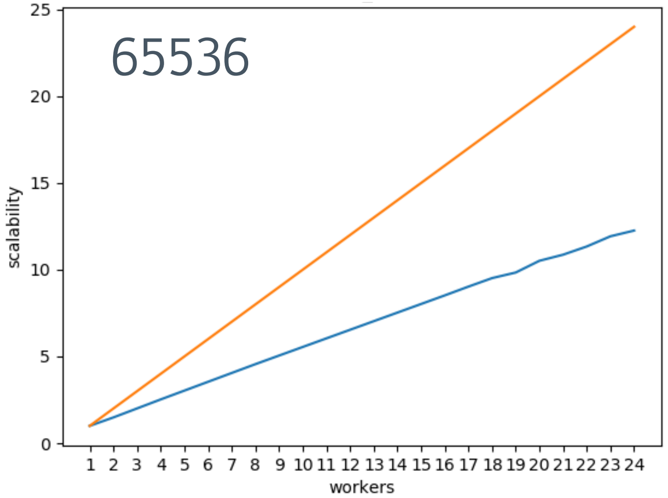

parallel
910596
scalability
workers
0
5
10
15
20
25
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24

Parallel Async

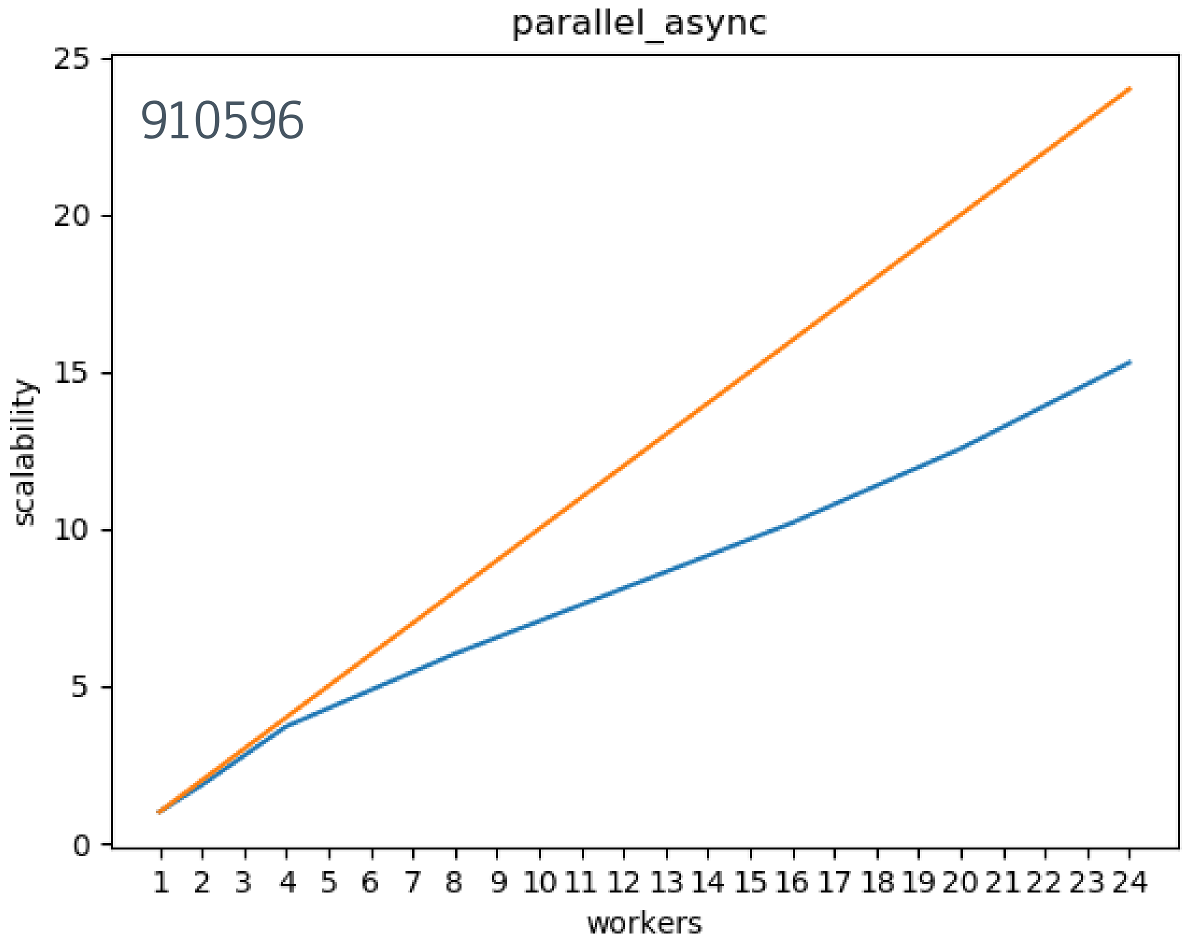




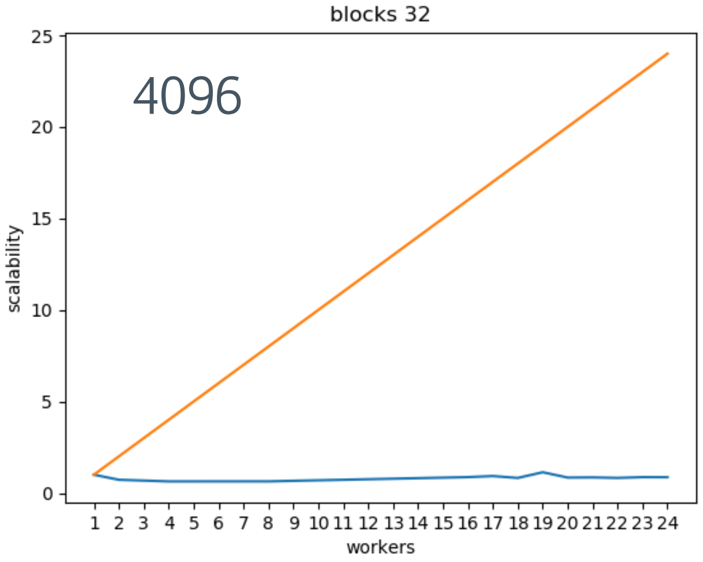


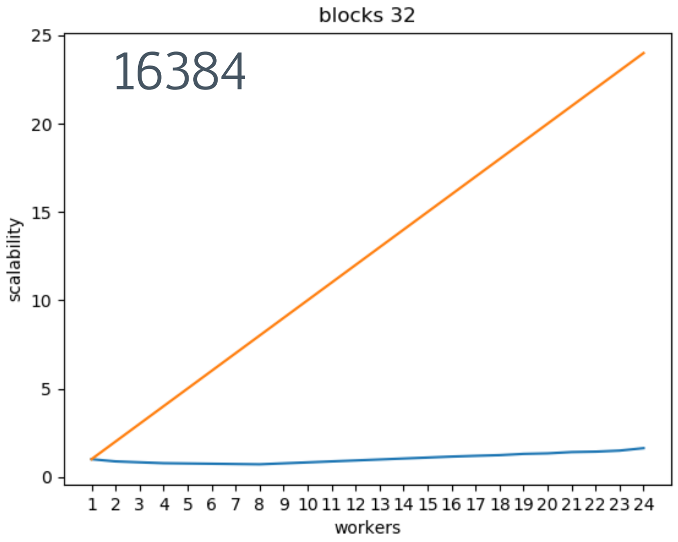


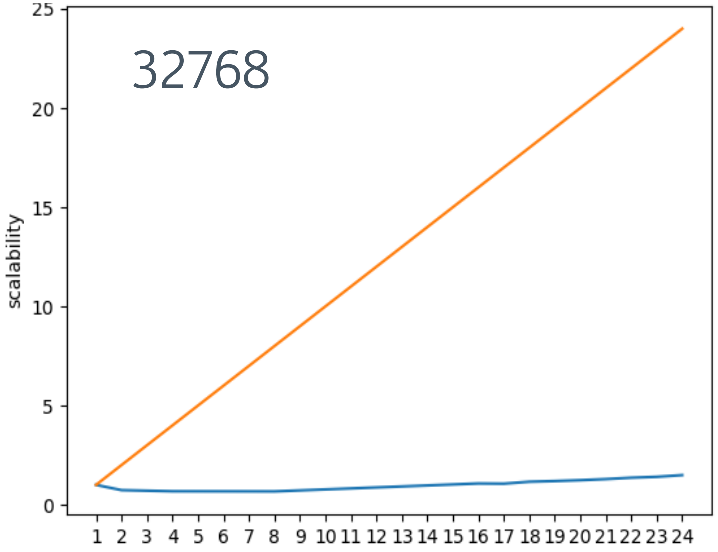
Parallel Async

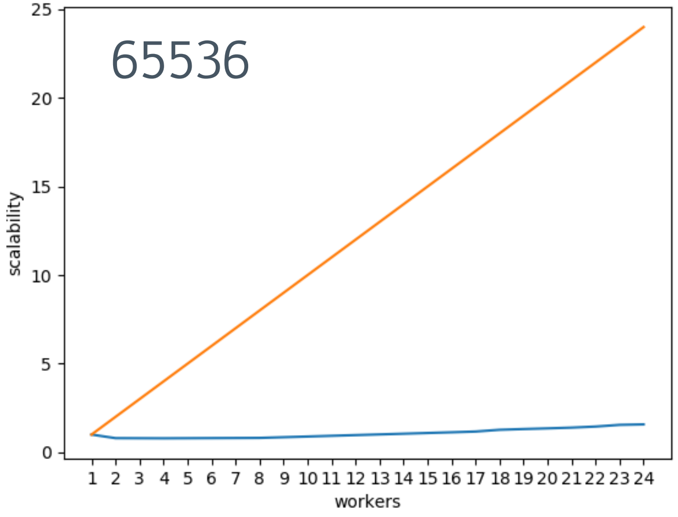


Blocks

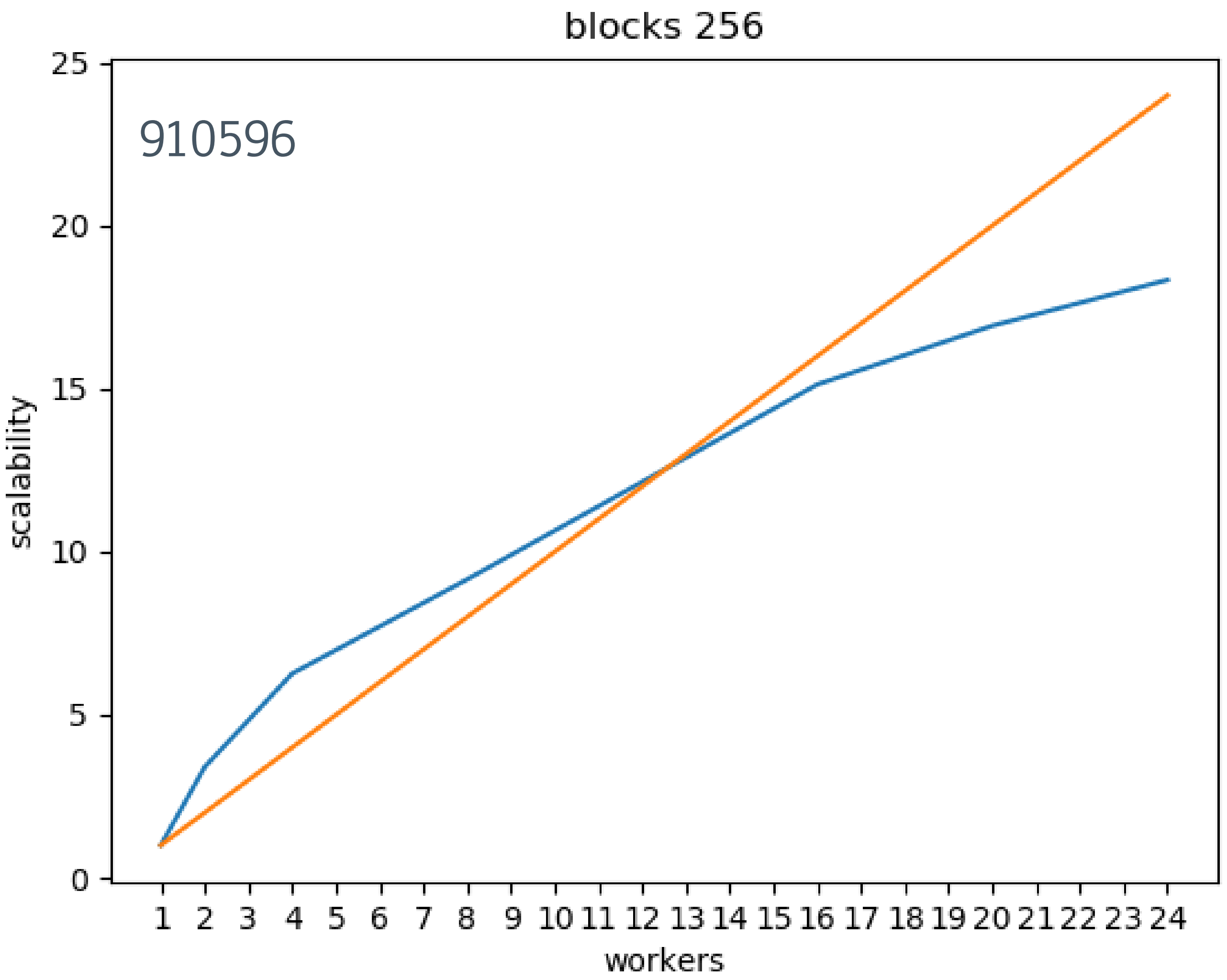

blocks 32
4096
scalability
workers


blocks 32
16384
scalability
workers

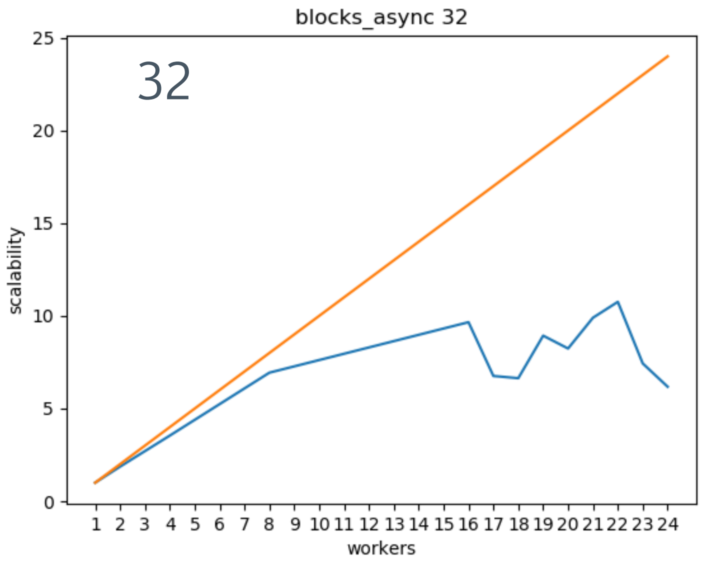

32768
scalability

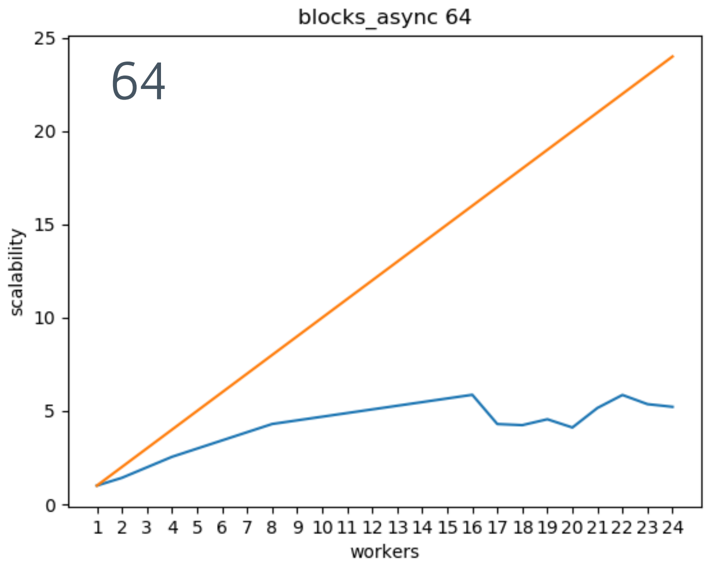

65536
scalability
workers

Blocks


blocks 256
910596
scalability
workers
0
5
10
15
20
25
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24

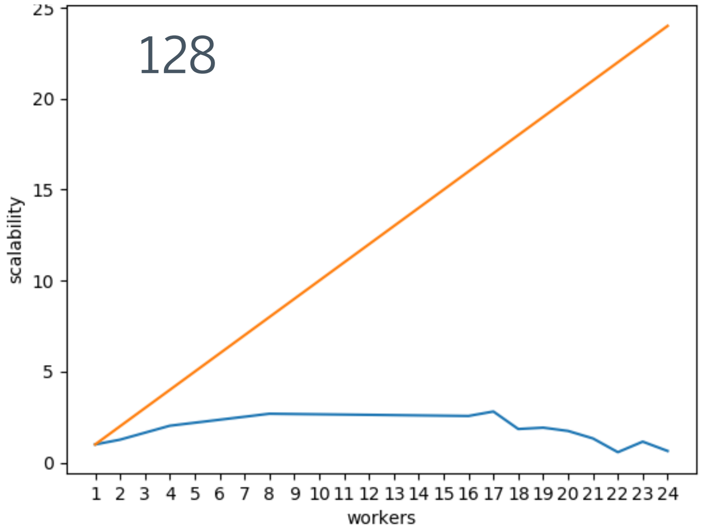
Blocks Async

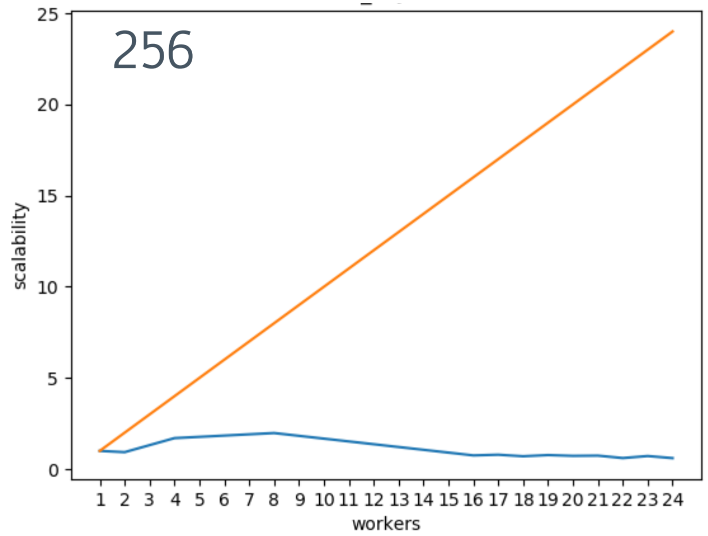




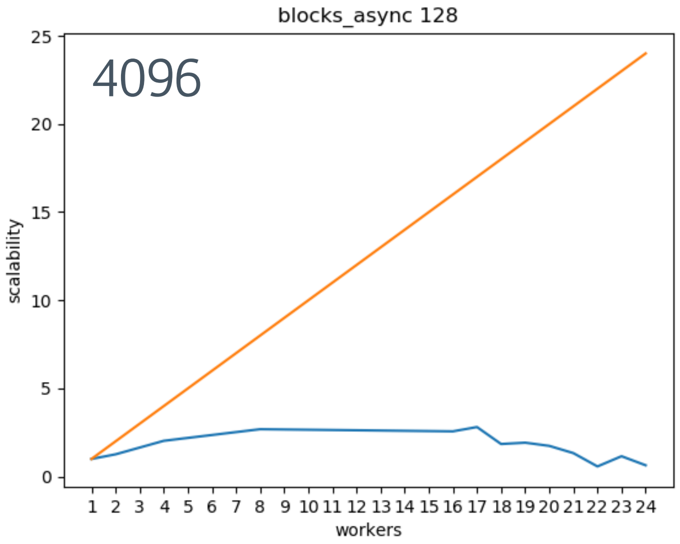
4096

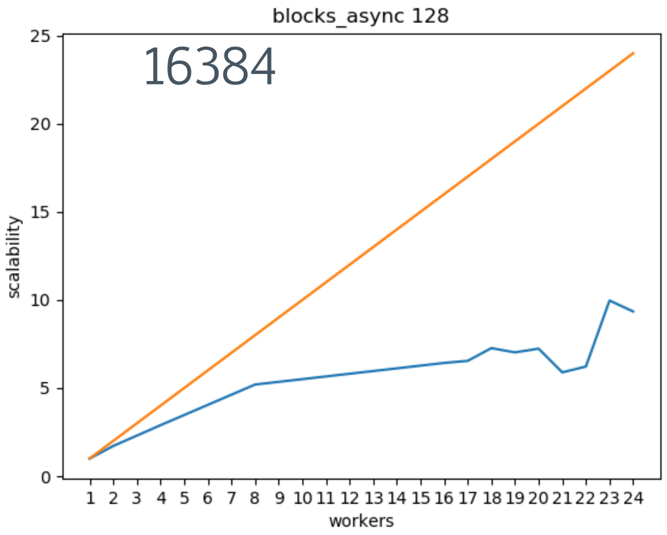
Blocks too big!



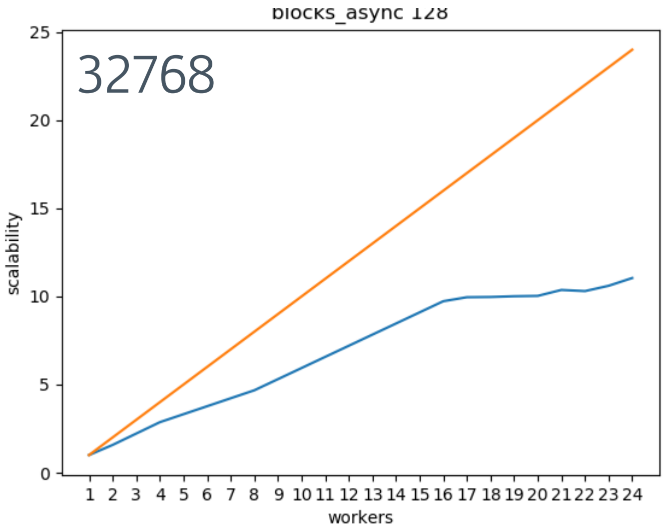


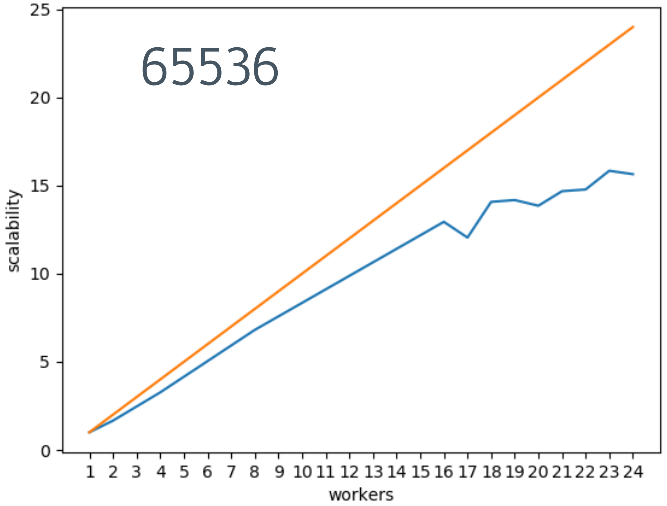
Blocks Async



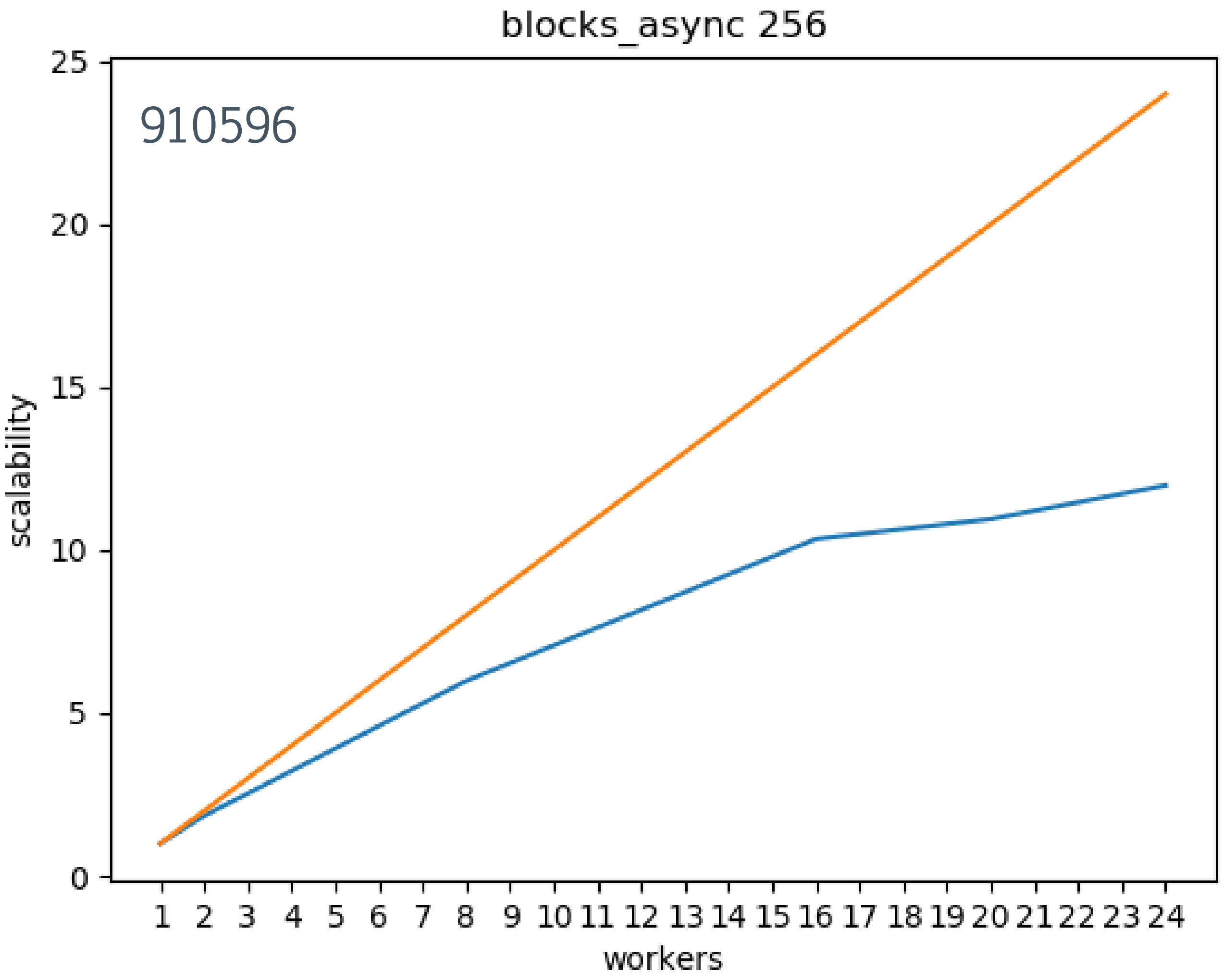


128

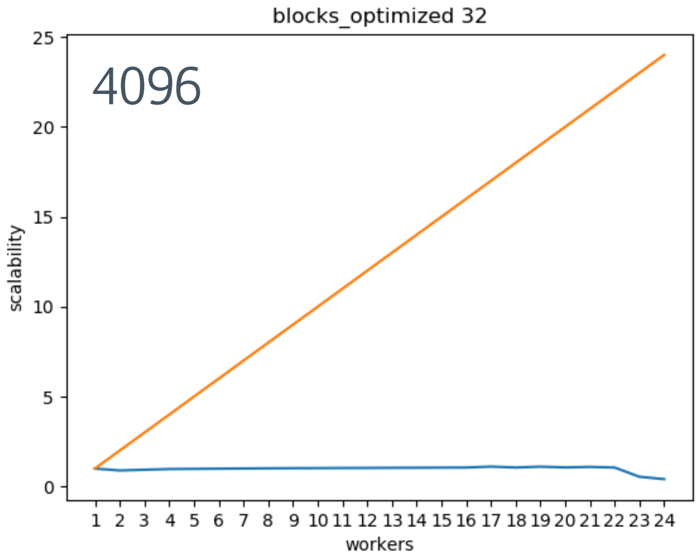


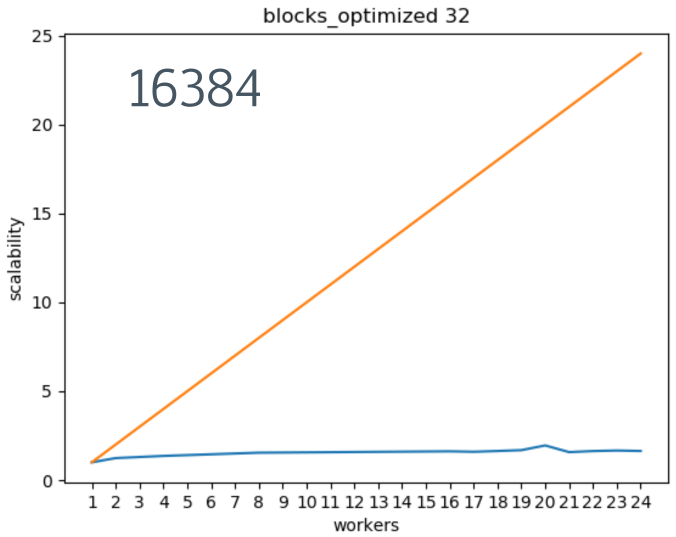


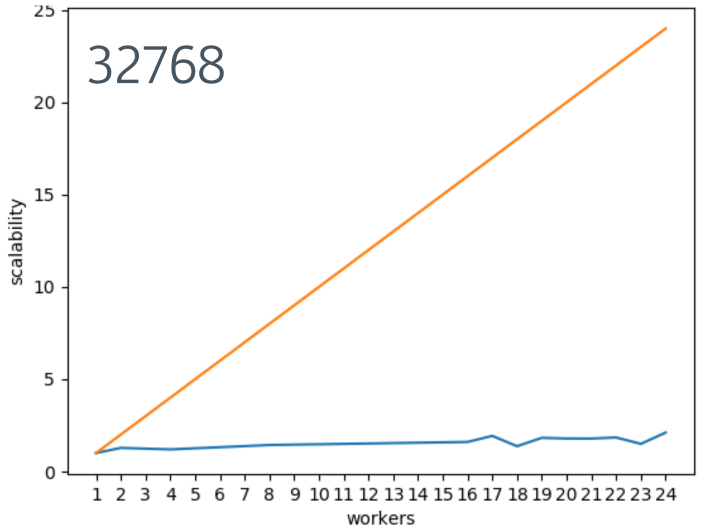
Blocks Async

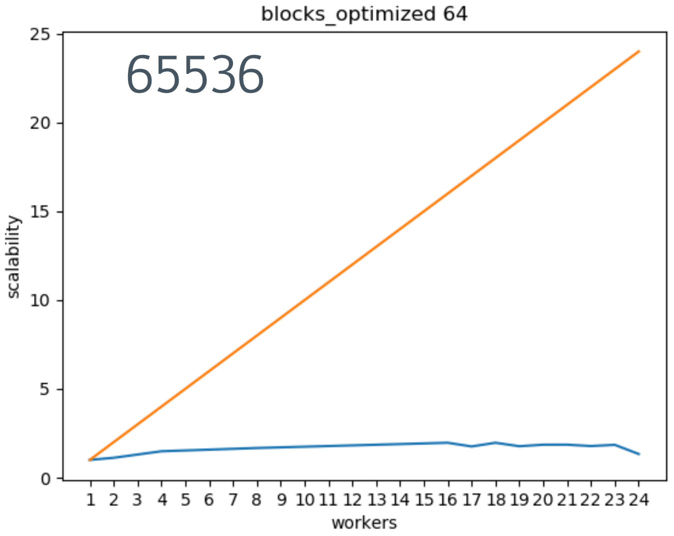


Blocks Optimized

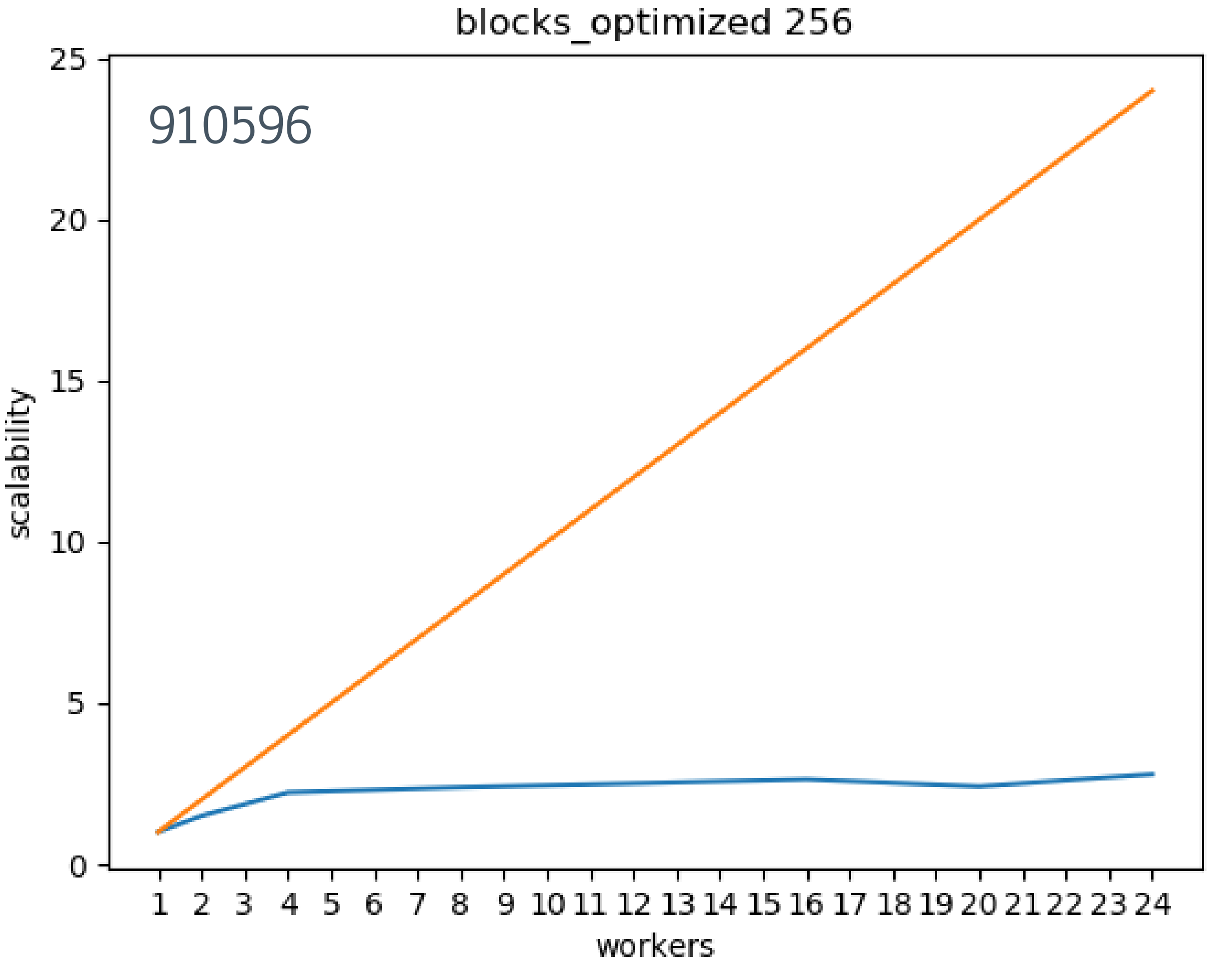




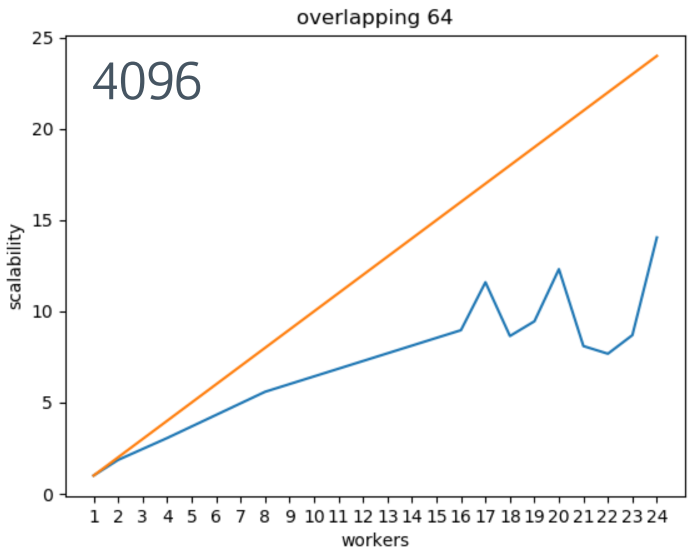


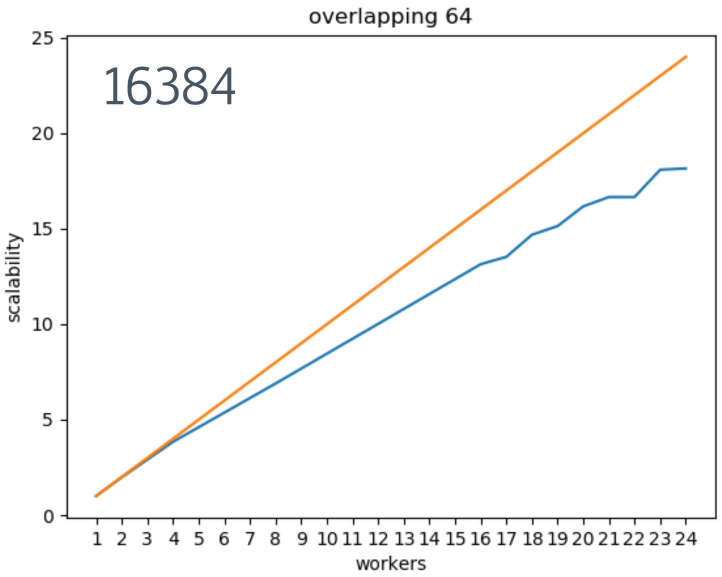


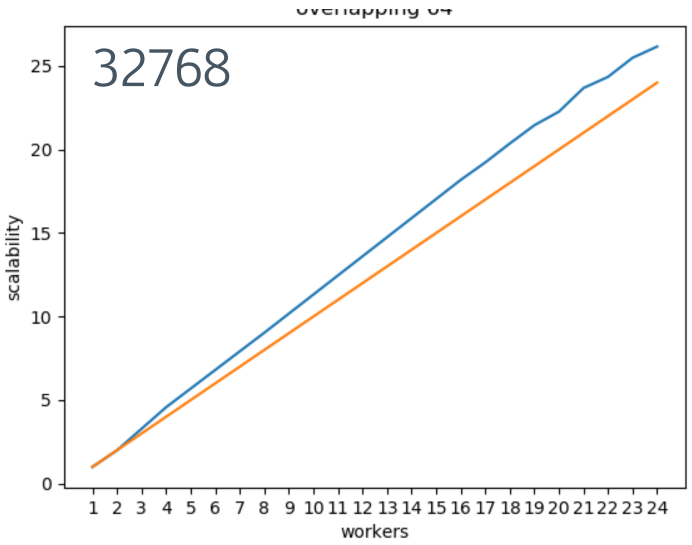
Blocks Optimized

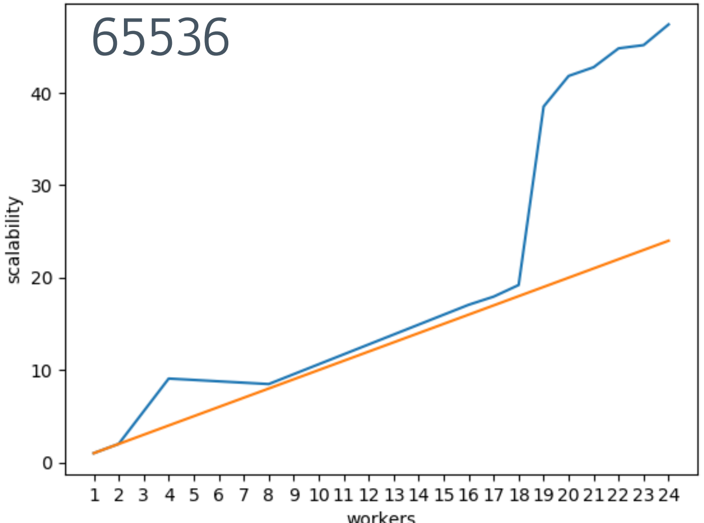


Overlapping

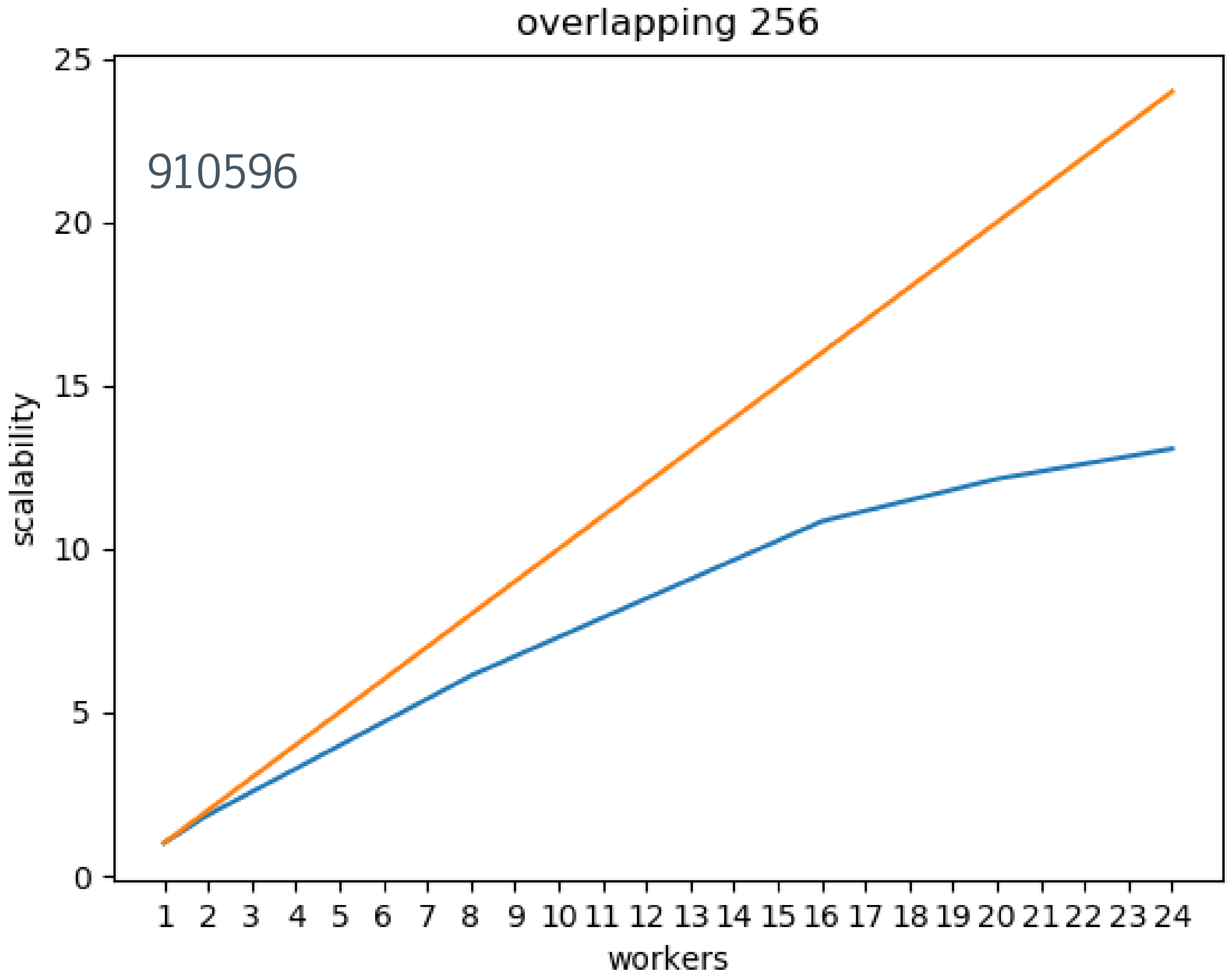




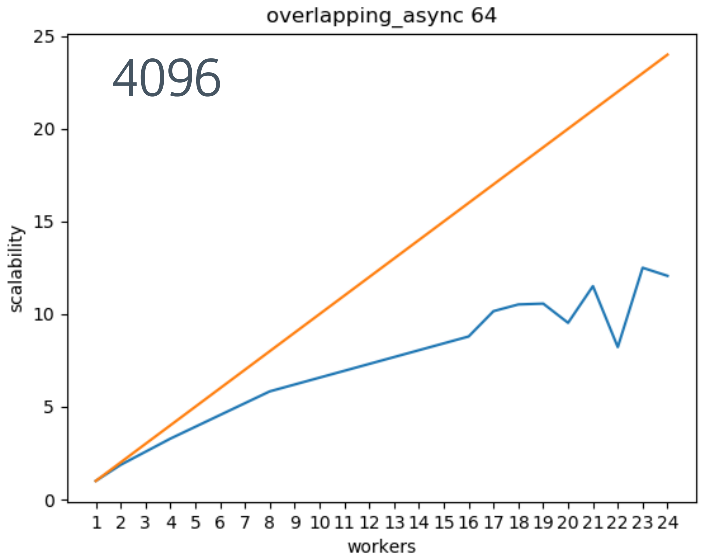


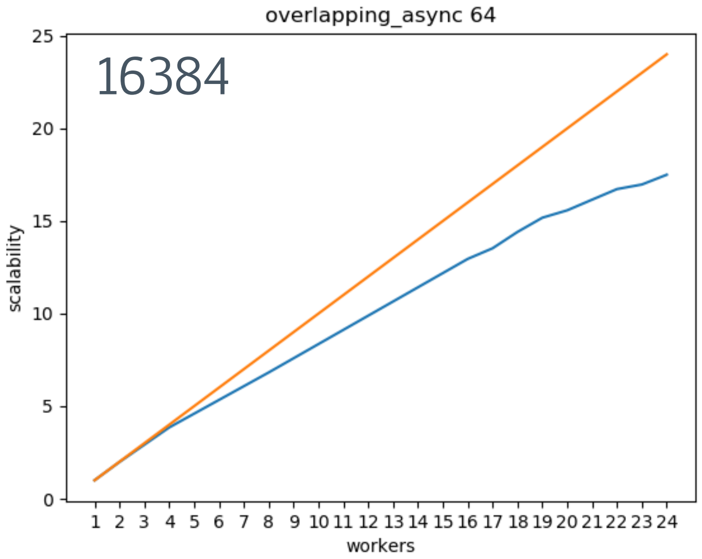


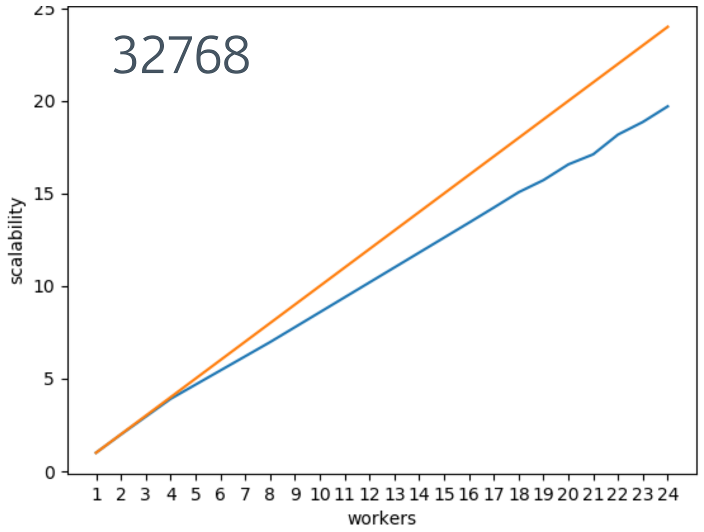
Overlapping

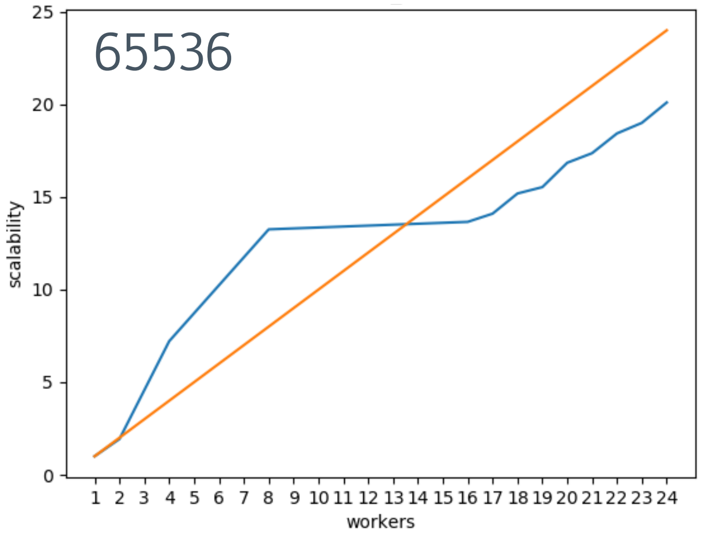


Overlapping Async

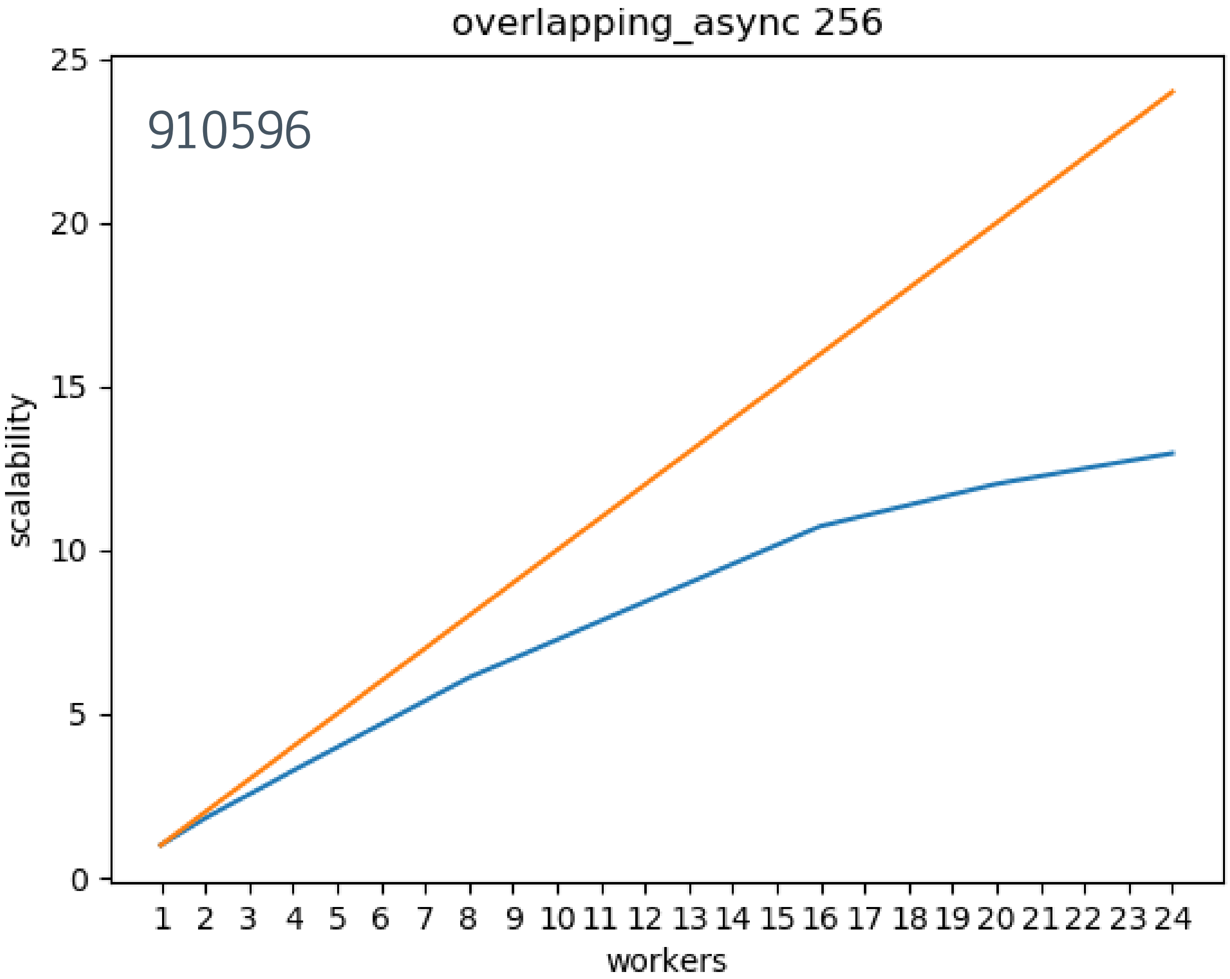




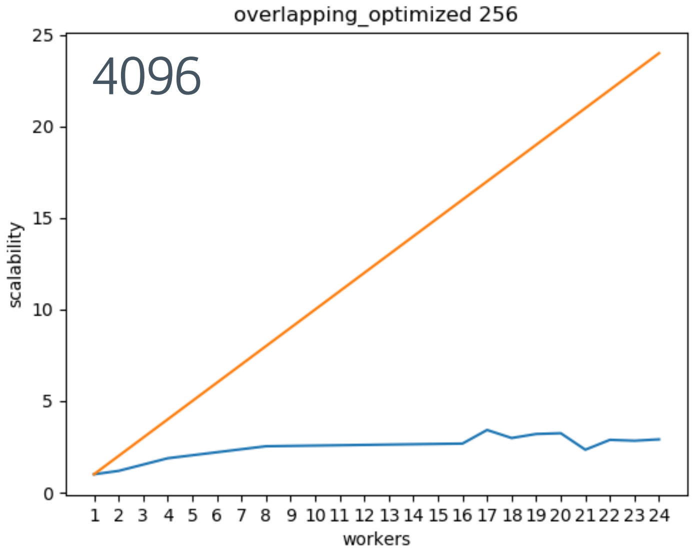


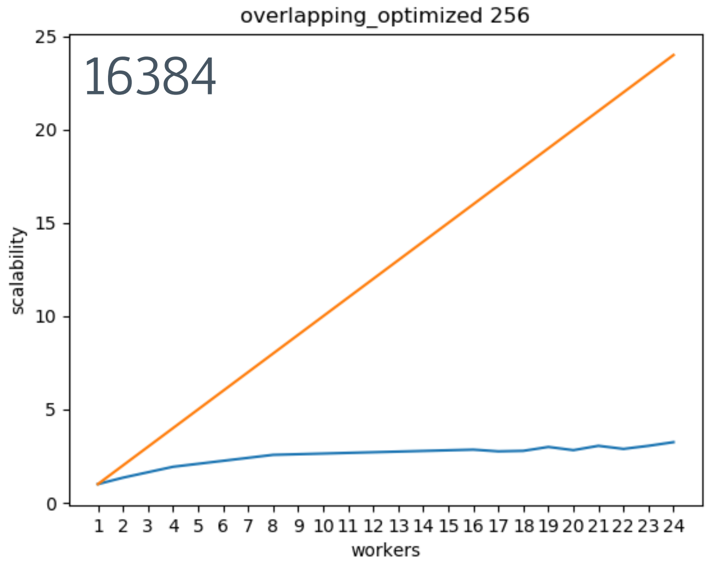


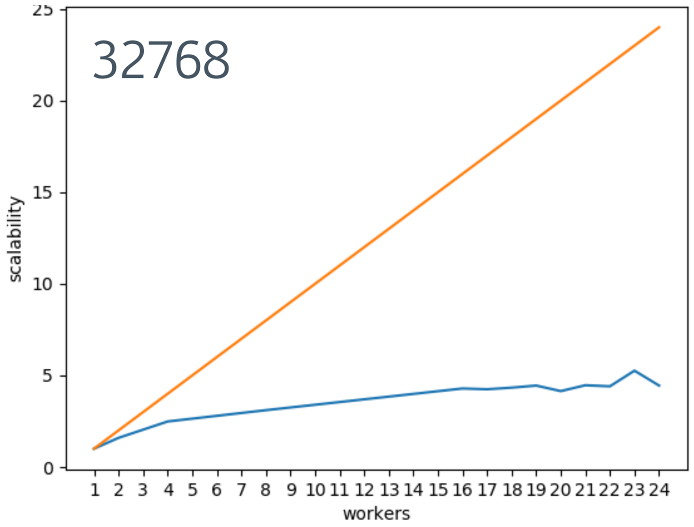
Overlapping Async

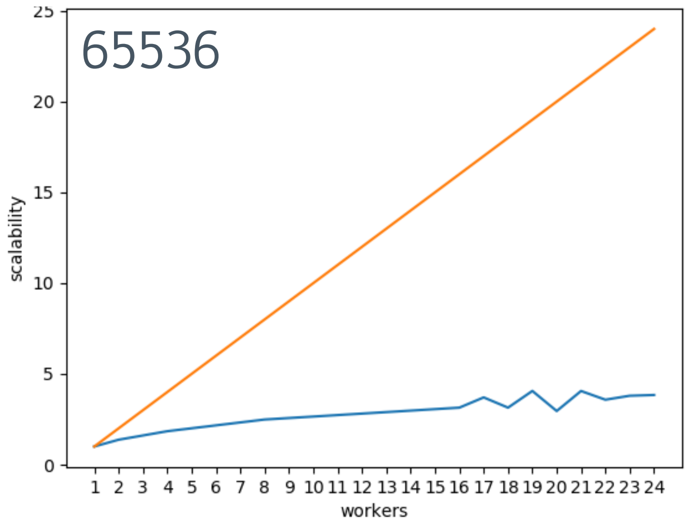


Overlapping Optimized

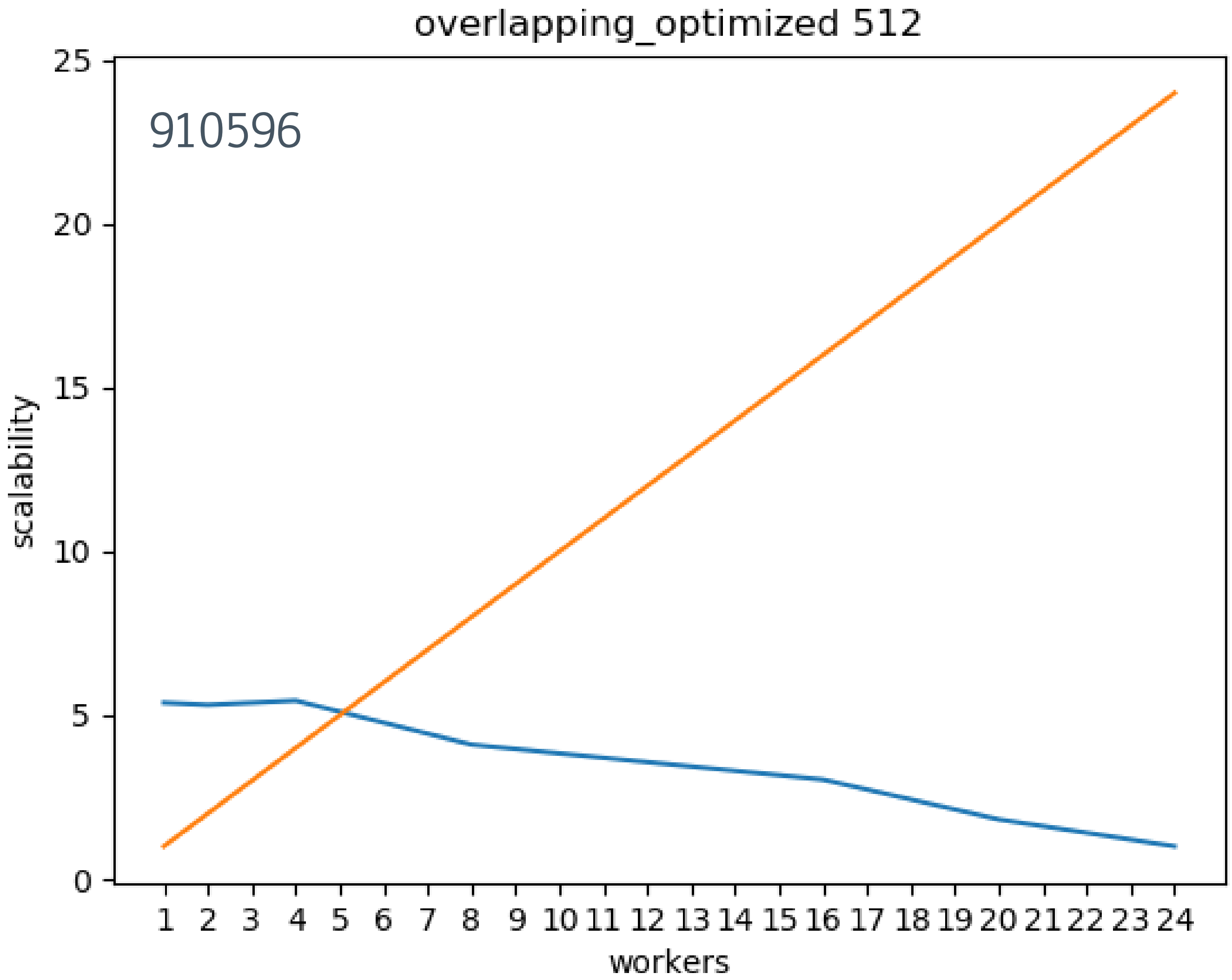








Overlapping Optimized

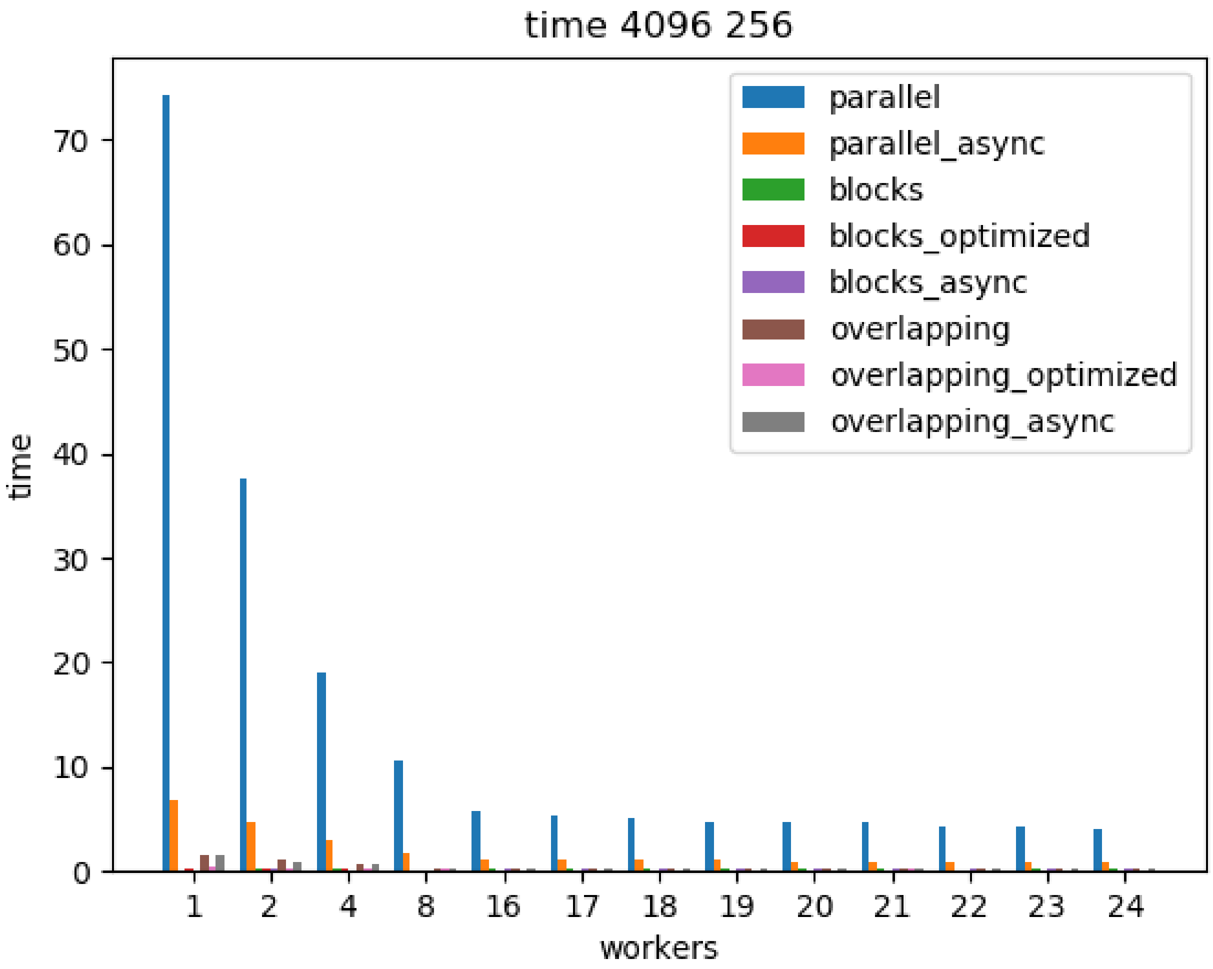


Time

Parameters

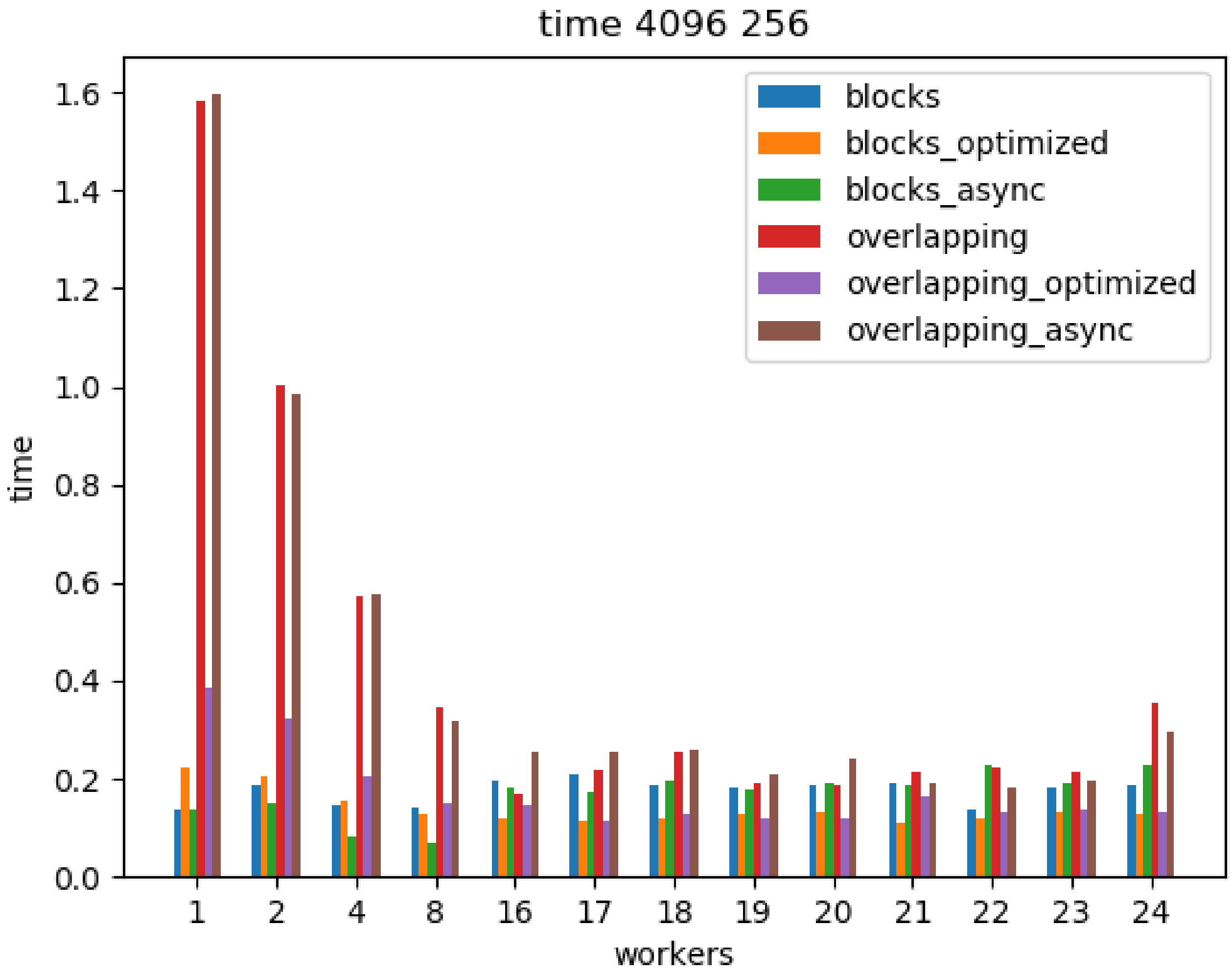
- Max number of iterations 100
- Number of physical cores 24
- Error tolerated 10^{-10}
- Block size 256

Random matrix 4096



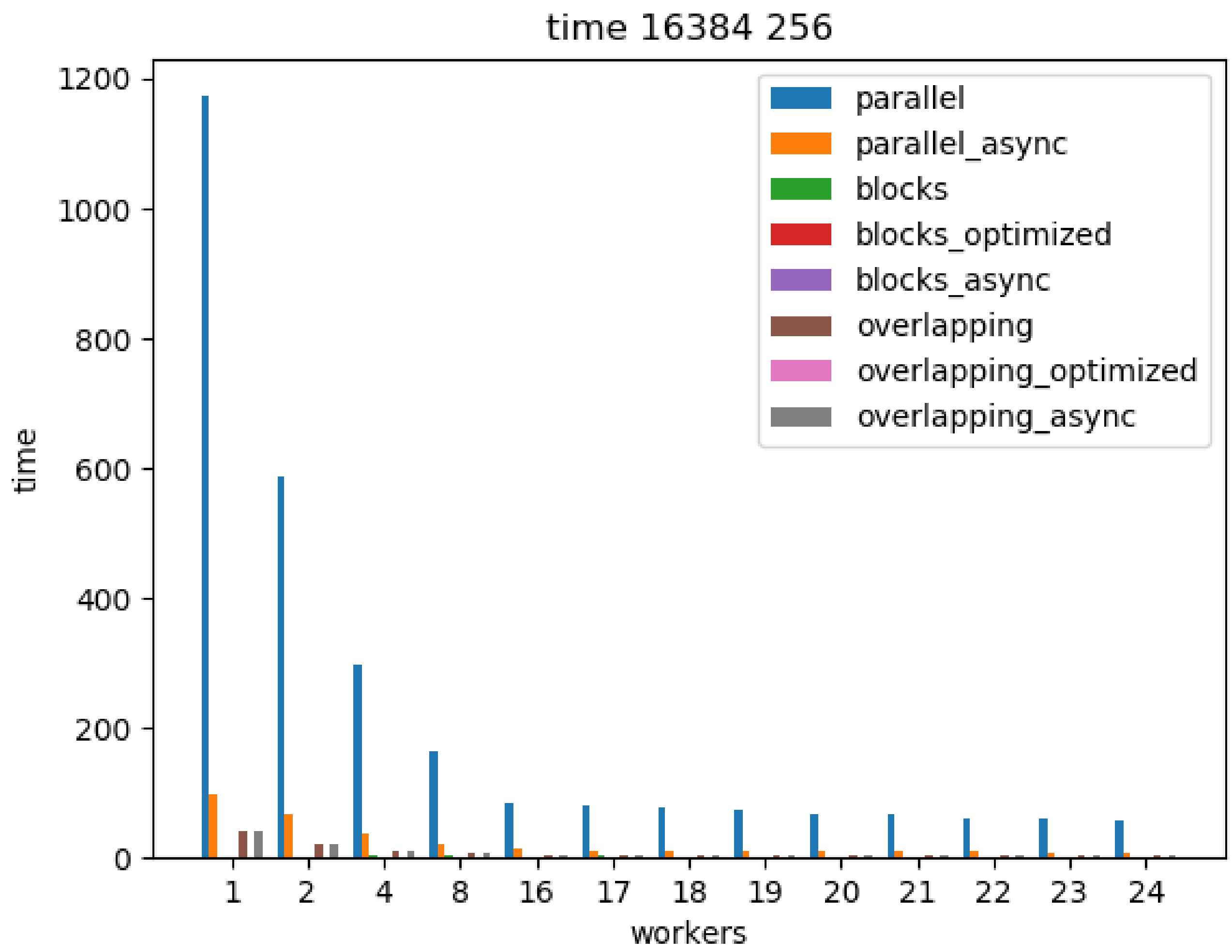
All algorithms

Random matrix 4096



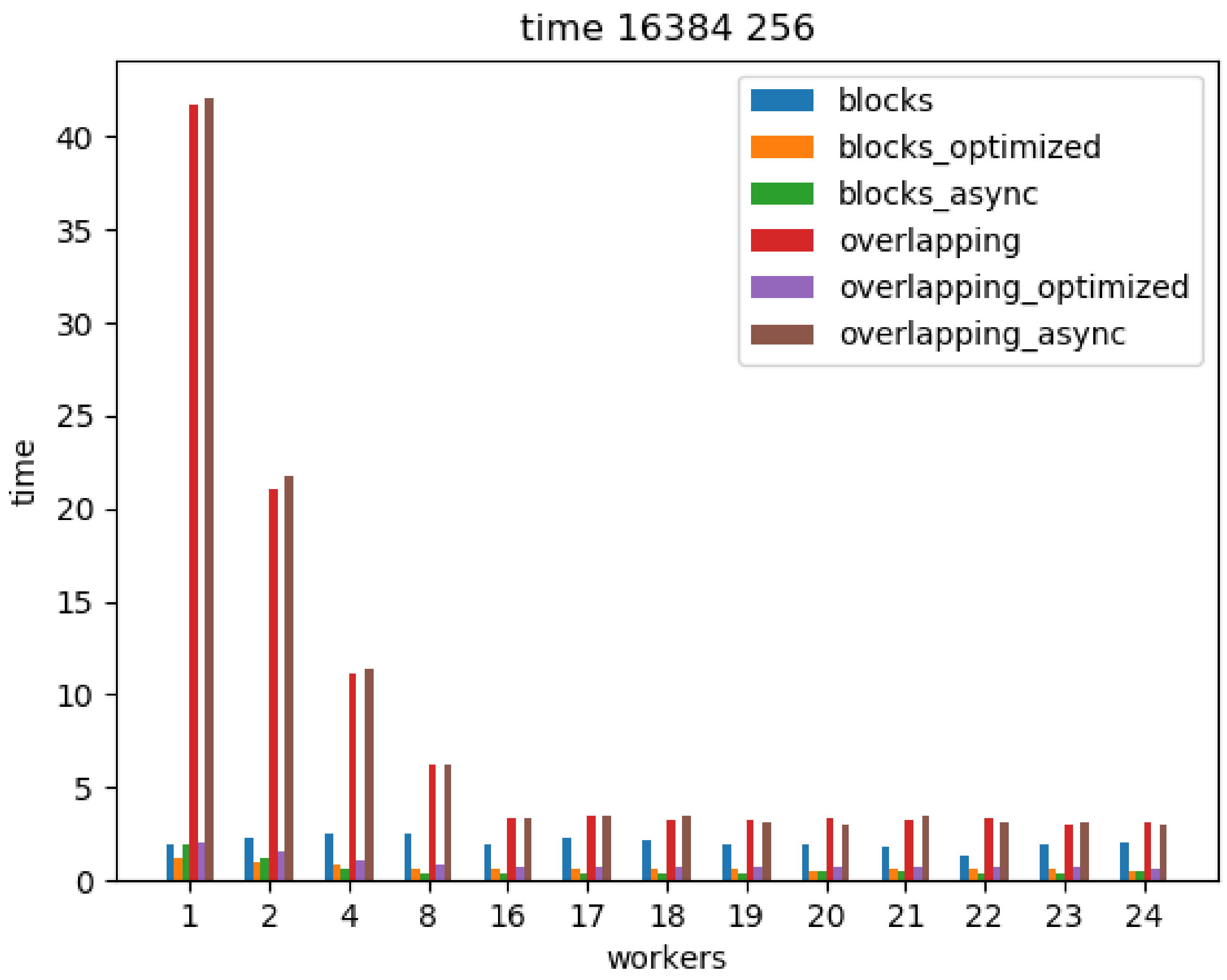
Blocks algorithms

Random matrix 16384



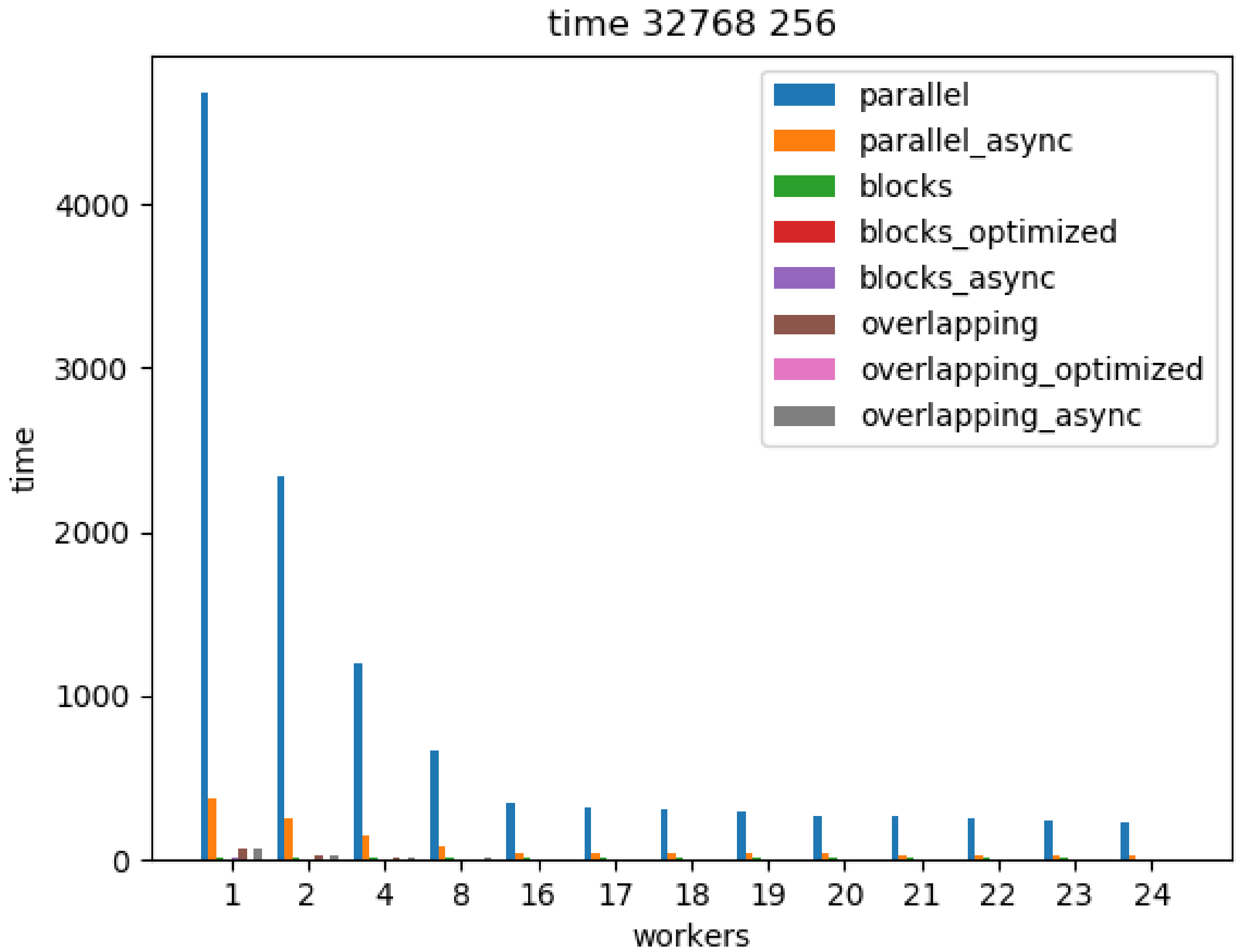
All algorithms

Random matrix 16384



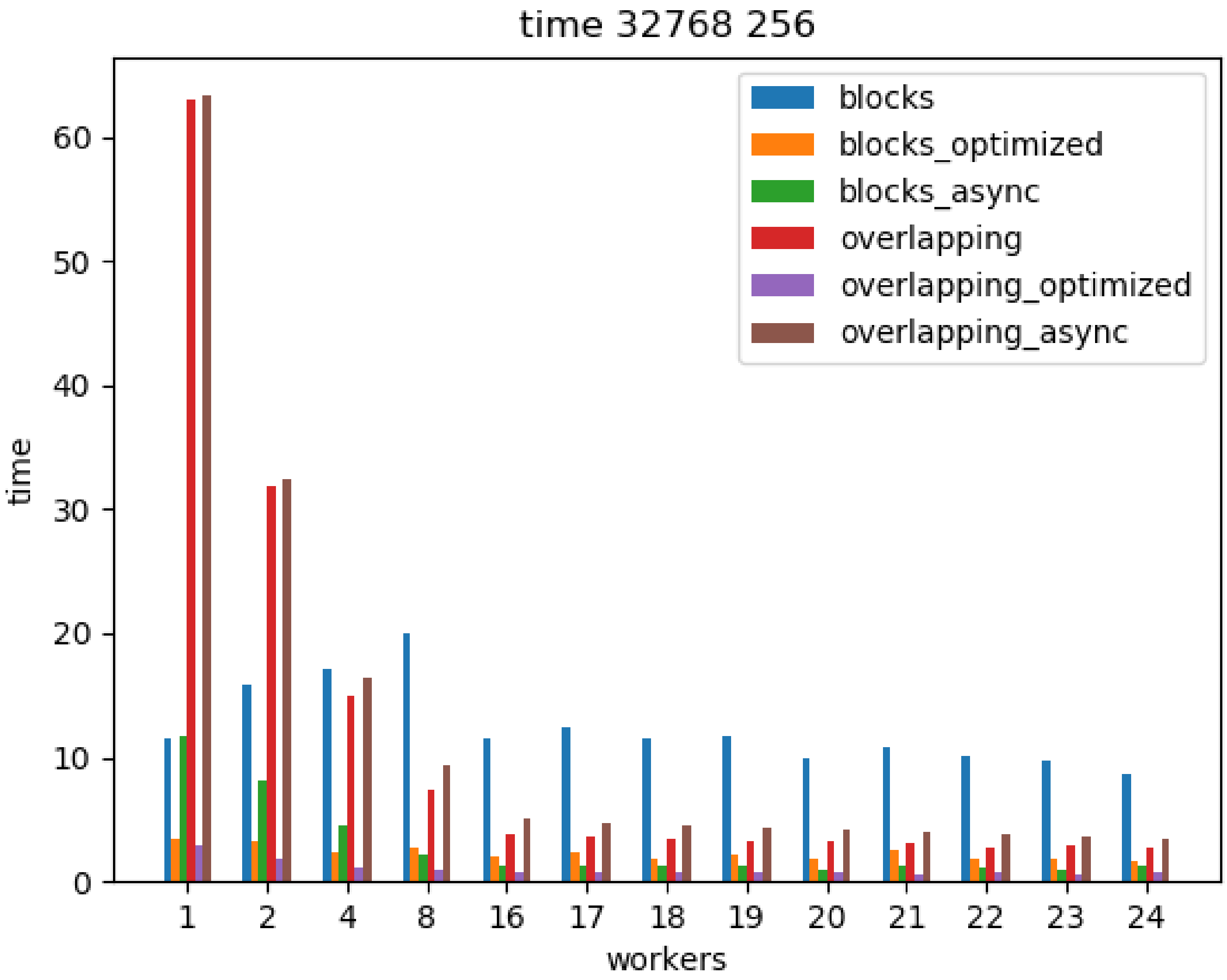
Blocks algorithms

Random matrix 32768



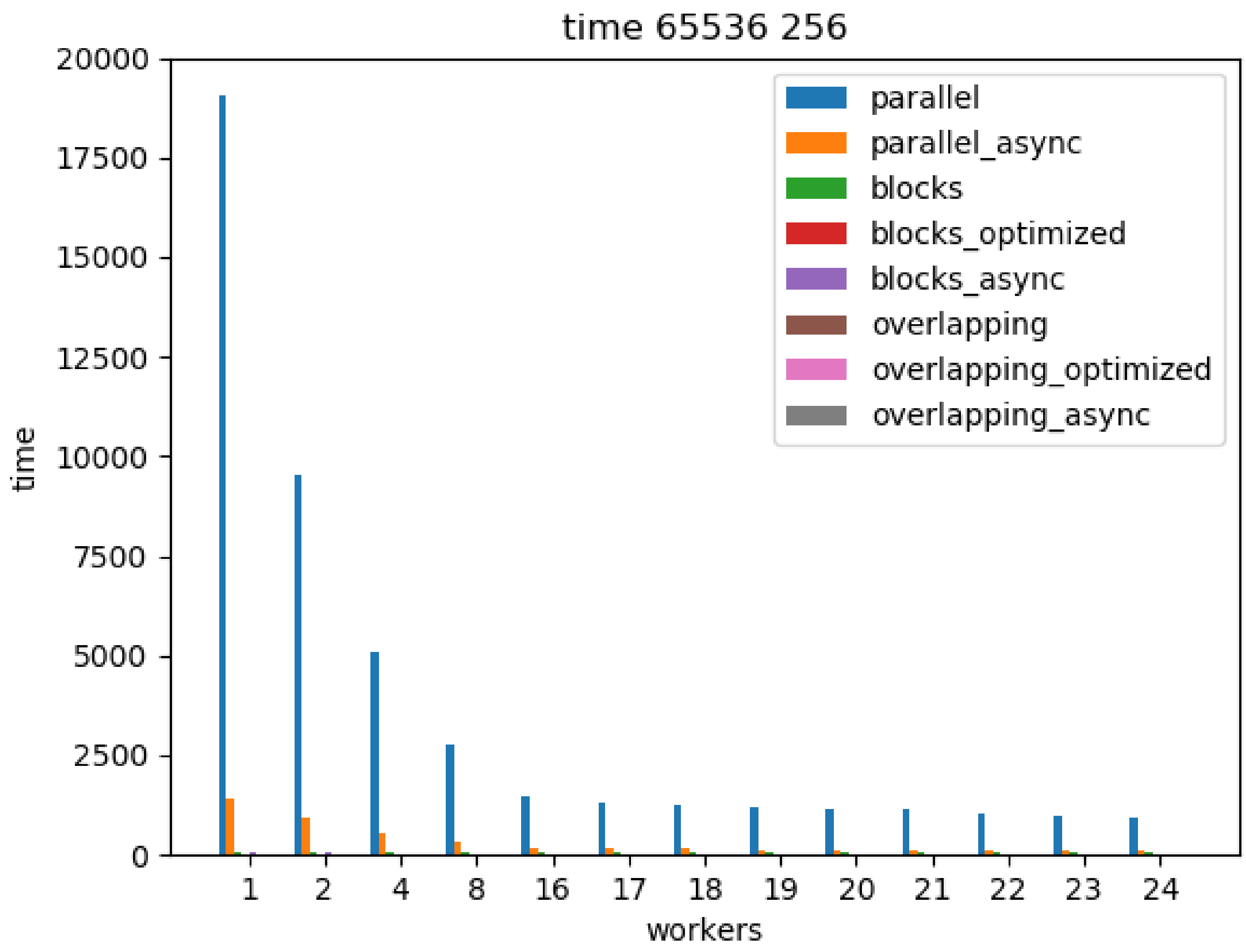
All algorithms

Random matrix 32768



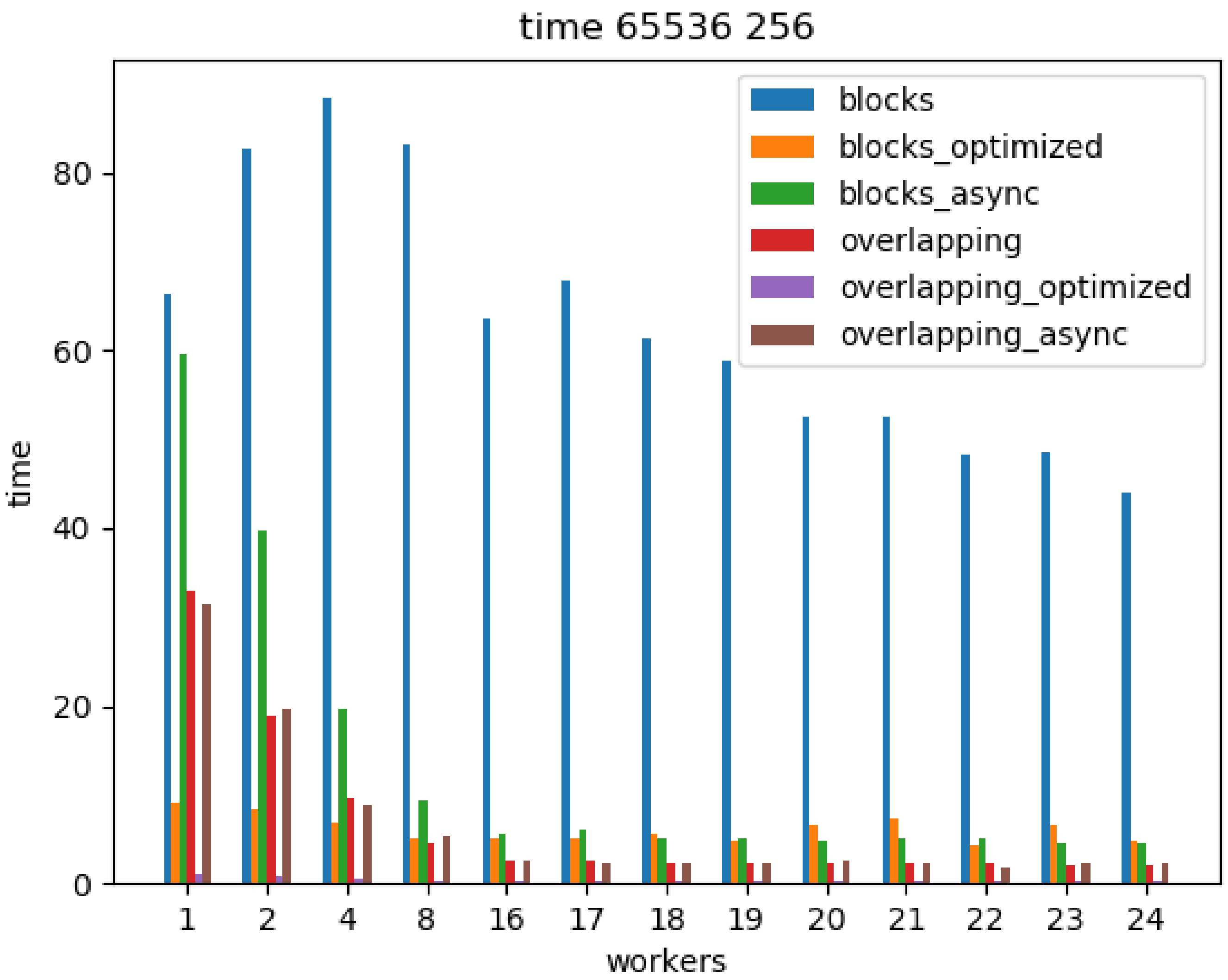
Blocks algorithms

Random matrix 65536



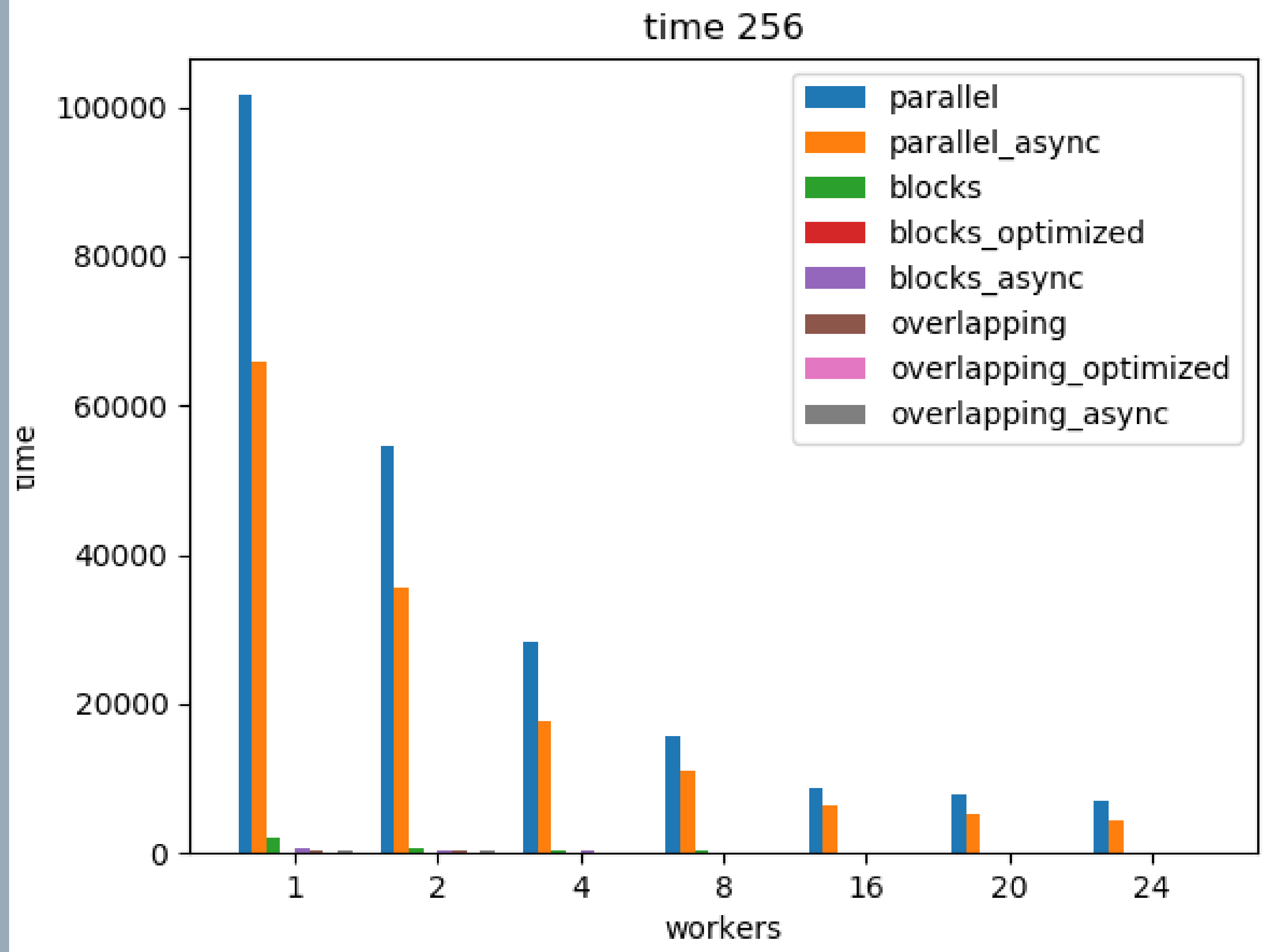
All algorithms

Random matrix 65536



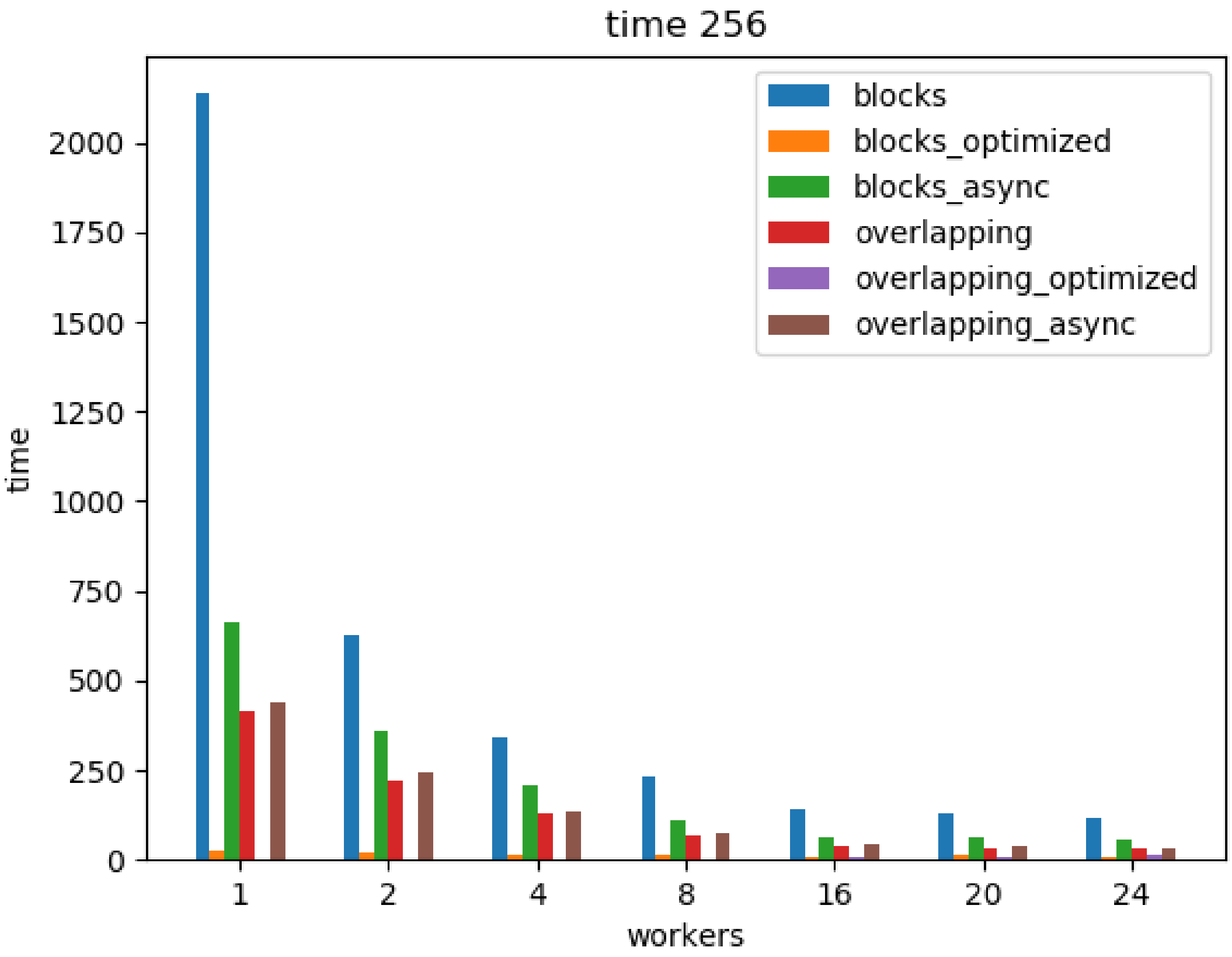
Blocks algorithms

Transition matrix 910596



All algorithms

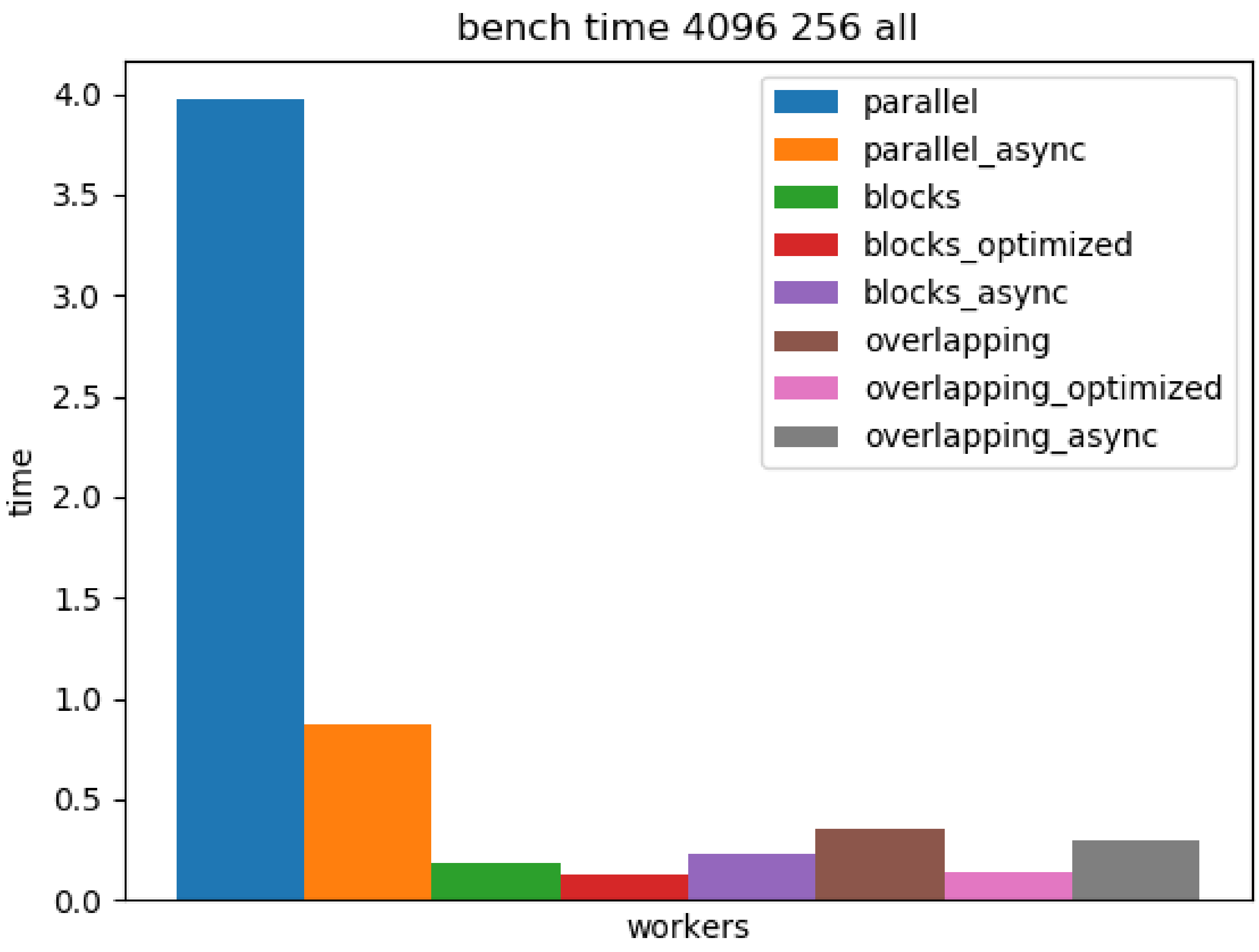
Transition matrix 910596



Blocks algorithms

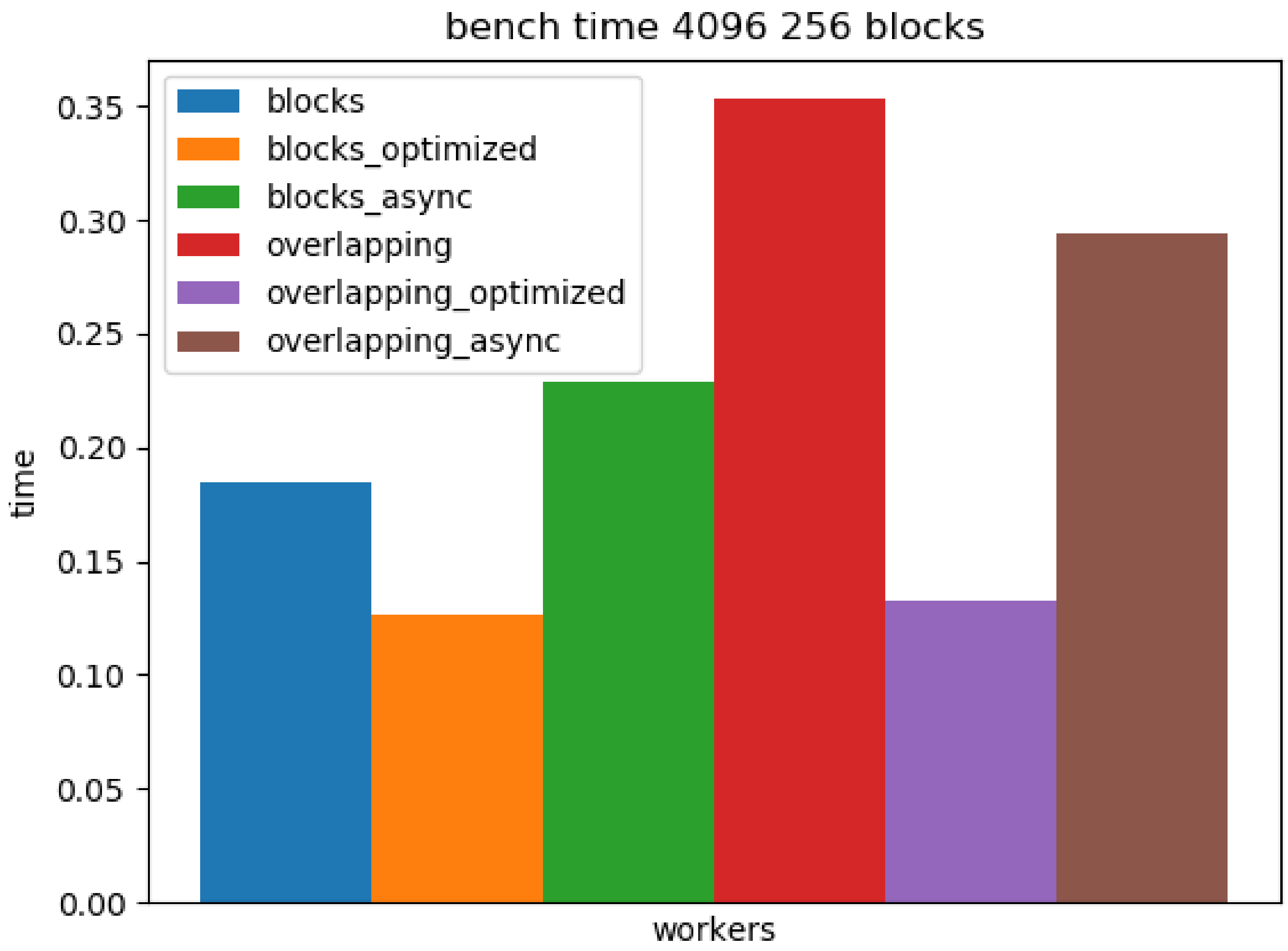
Benchmark

Random matrix 4096



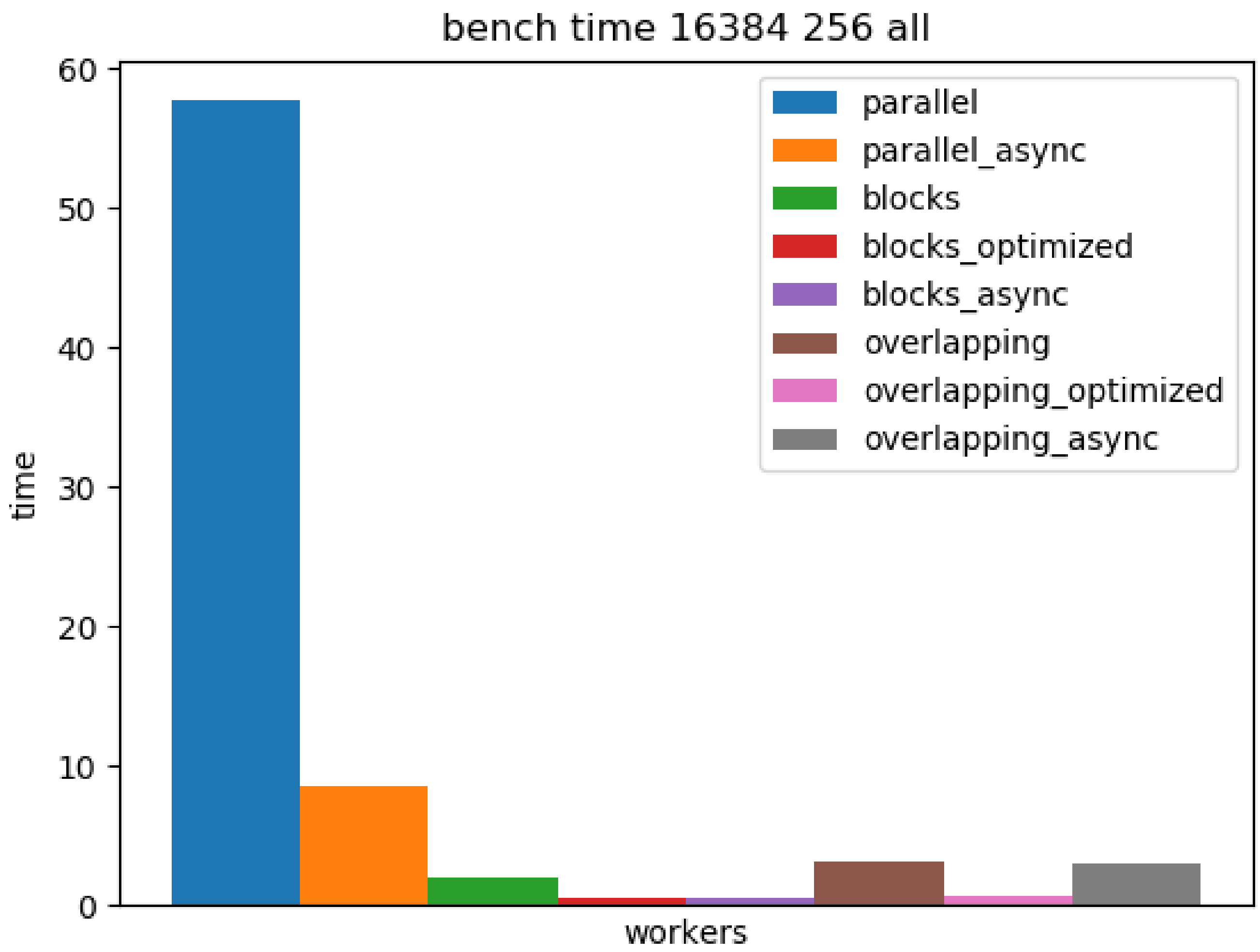
All algorithms

Random matrix 4096



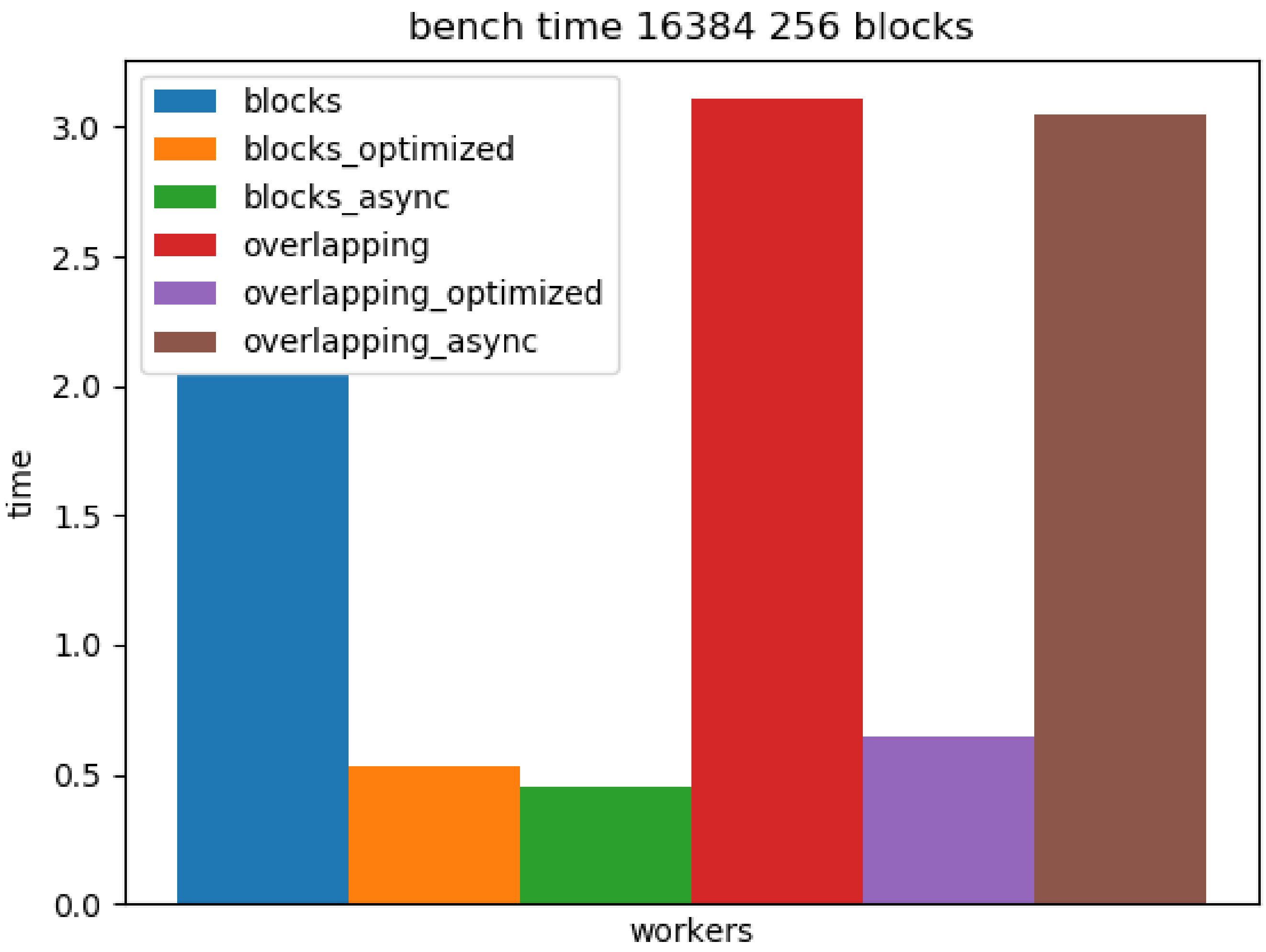
Blocks algorithms

Random matrix 16384



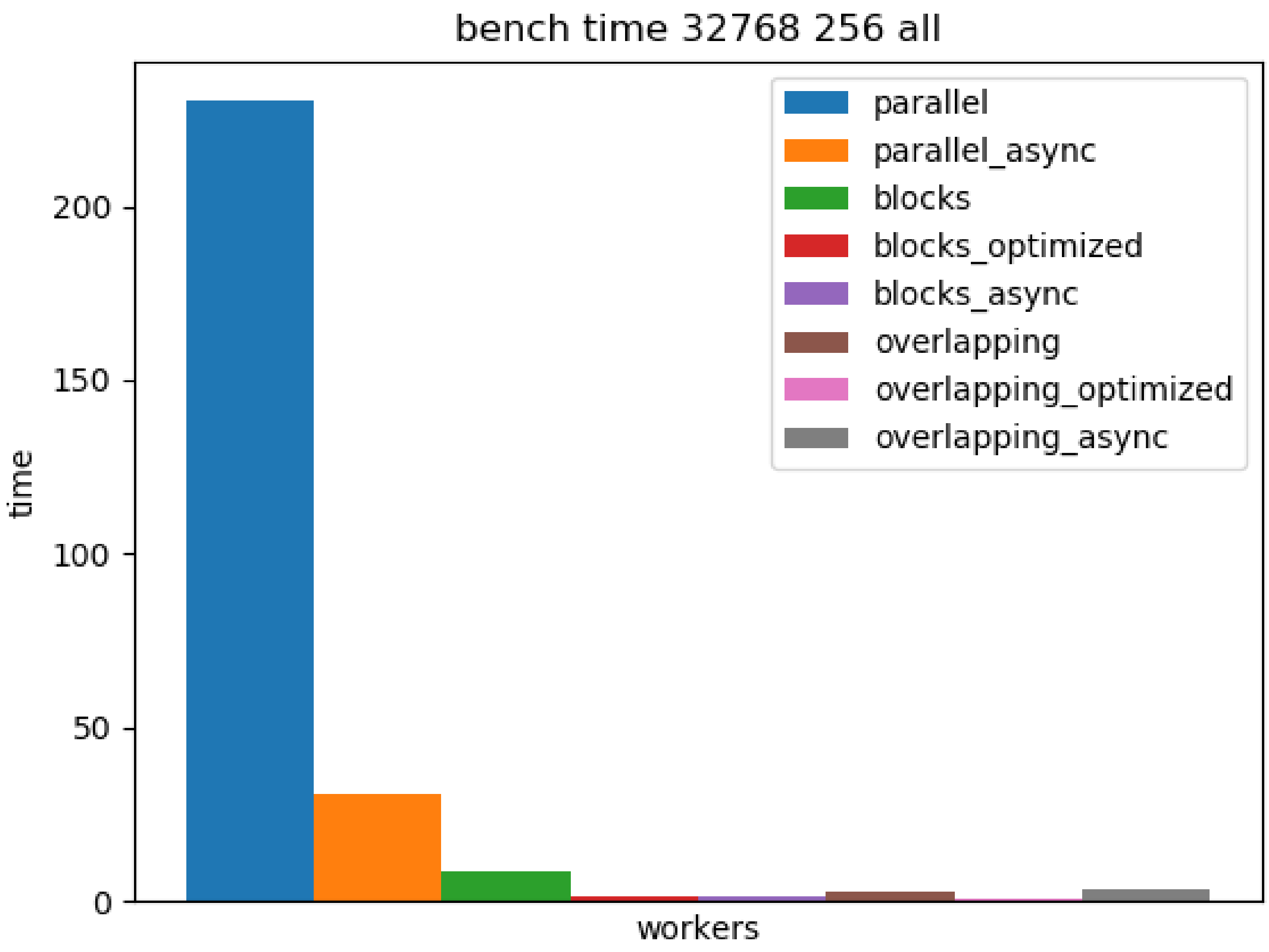
All algorithms

Random matrix 16384



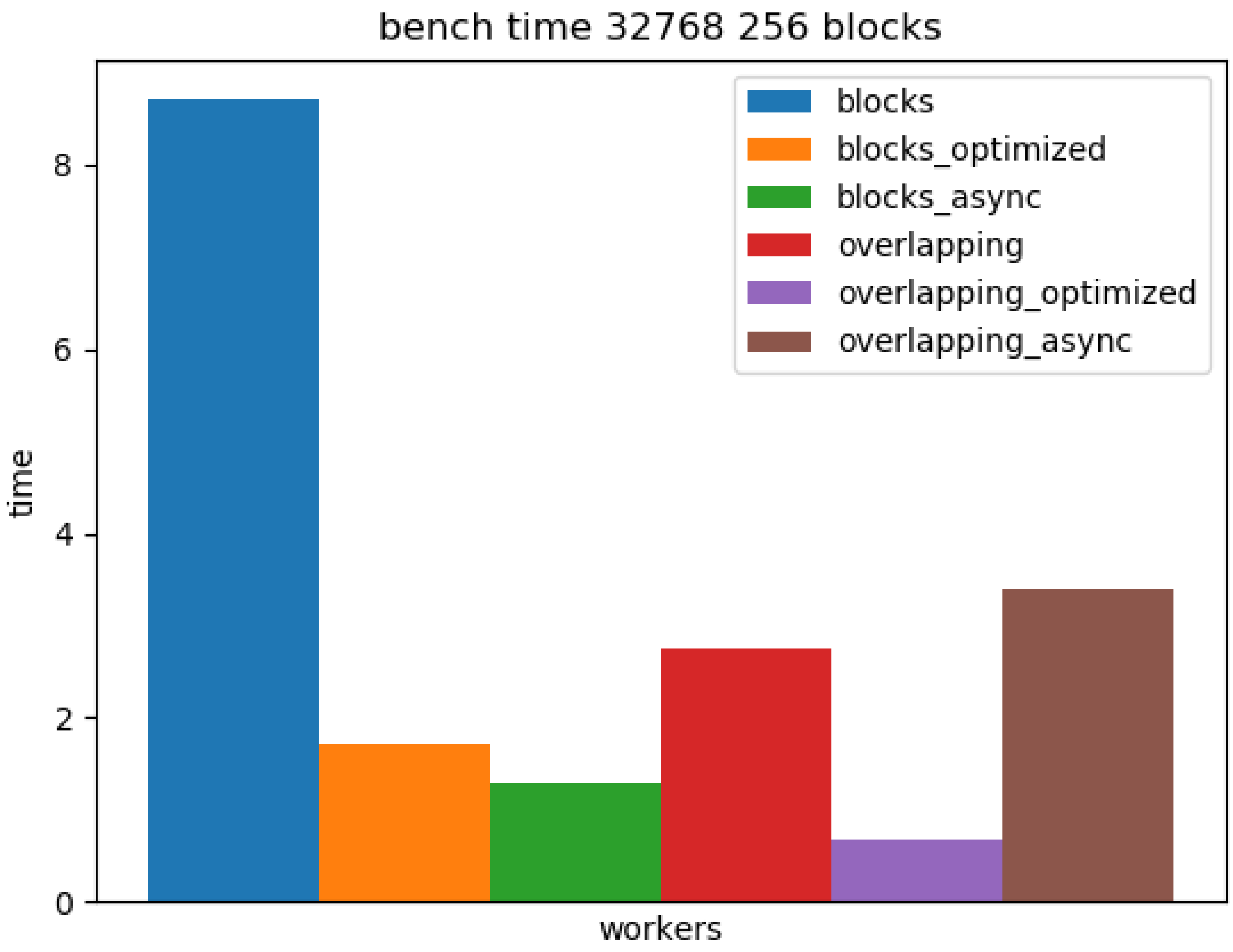
Blocks algorithms

Random matrix 32768



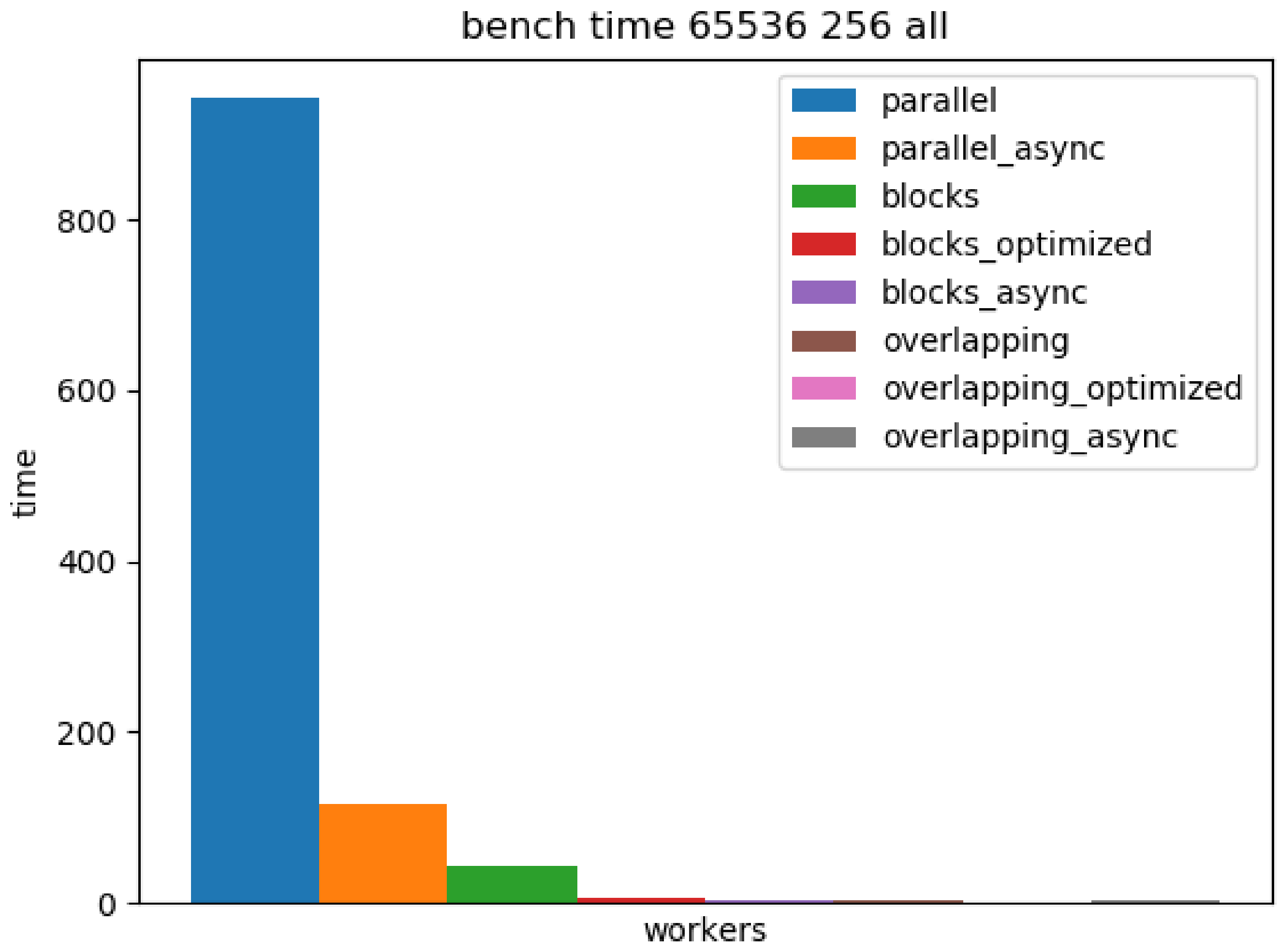
All algorithms

Random matrix 32768



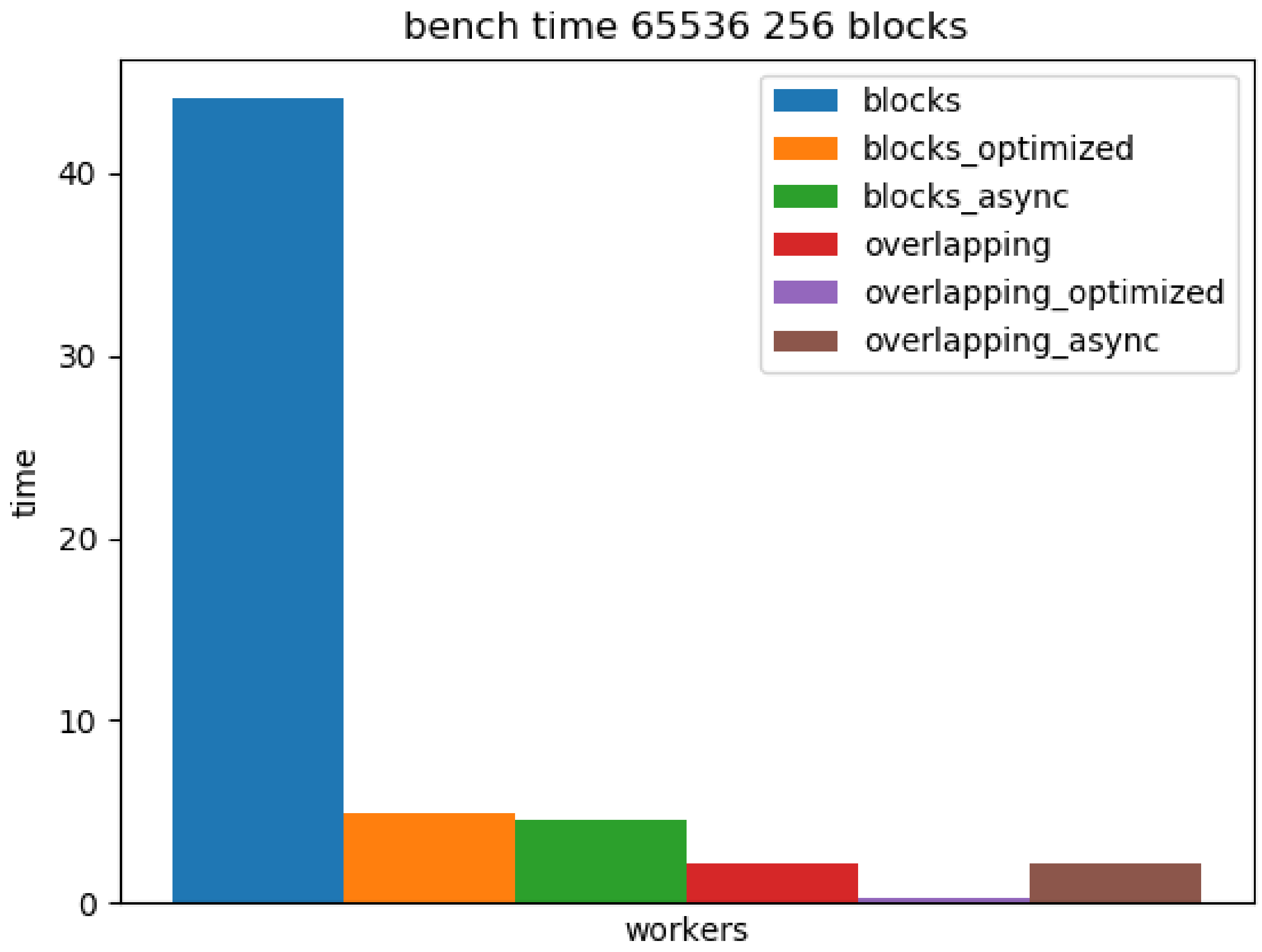
Blocks algorithms

Random matrix 65536



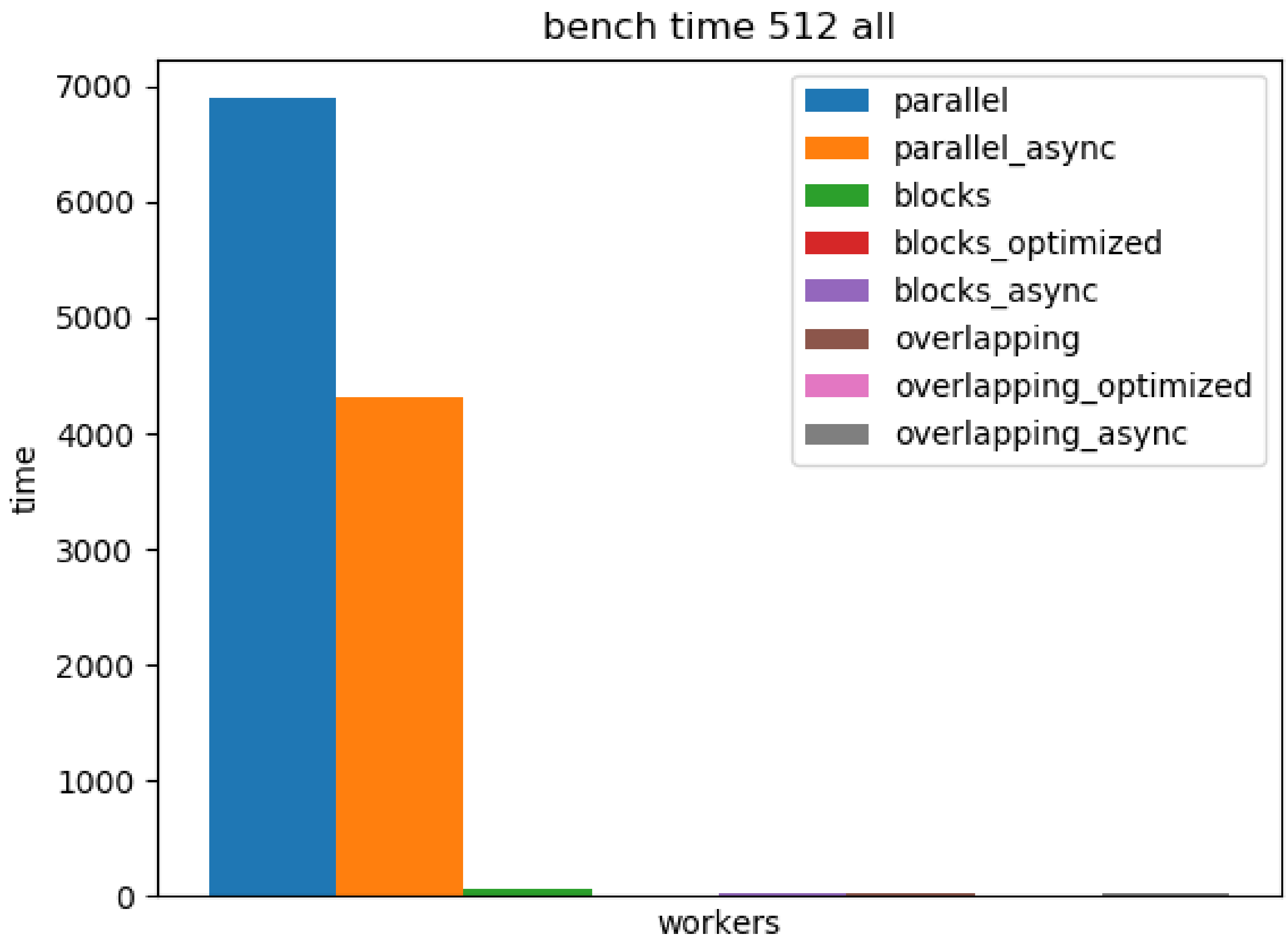
All algorithms

Random matrix 65536



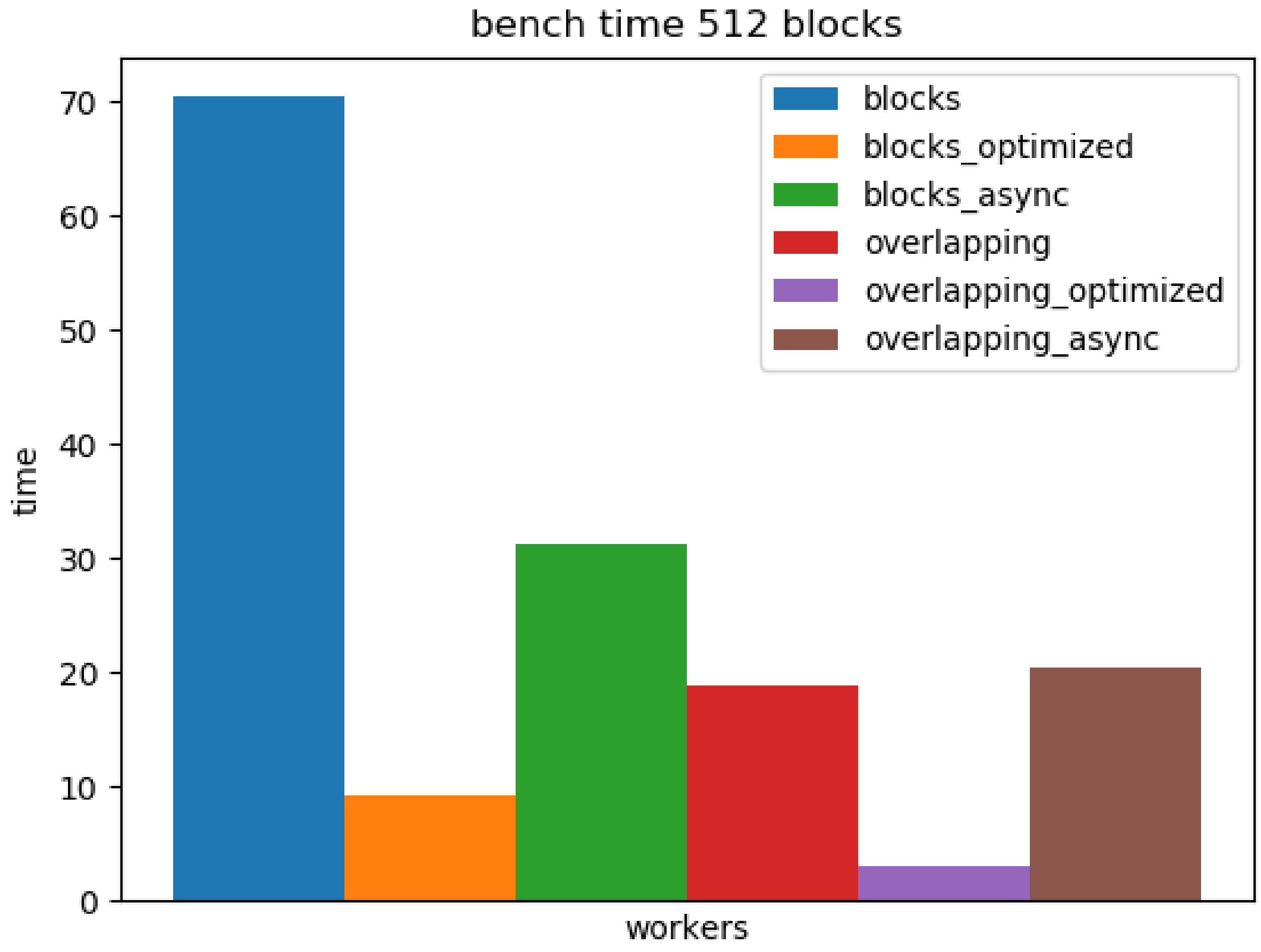
Blocks algorithms

Transition matrix 910596



All algorithms

Transition matrix 910596



Blocks algorithms

Conclusions