Co-Array Fortran Experiences with Finite Differencing Methods

Cray User Group 2006, Lugano, Switzerland
May 10, 2006

Richard F. Barrett

Oak Ridge National Laboratory
Oak Ridge, TN 37831

http://www.csm.ornl.gov/ft
http://www.nccs.gov

OAK RIDGE
National Laboratory

Goal of study

- To discover “natural” effective ways to use CAF for Finite Differencing.
  - With eye towards unstructured, semi-structured or dynamic grids.

- *What it isn’t:*
  - Create optimized version of FD.
  - Productivity study (though coding effort will be mentioned).
  - X1E compiler study (though performance results presented).

CAF syntax

```
REAL*8, ALLOCATABLE :: A(:,:)[:]

ALLOCATE( A(m,n)[*] )
```

– Local view; user manages decomposition.

```
A(i,j) = B(i,j)[img_loc]

A(i,j) = B(i,j)
```

CAF syntax *cont'd*

- If co-arrays to be of *different length*, create derived type containing the (locally [allocatable]) array.

```
grid[img_loc]%A(i,j)
```

- A few *intrinsics:*

```
sync_<all,team,memory>(<arg>)
```

Reductions proposed.

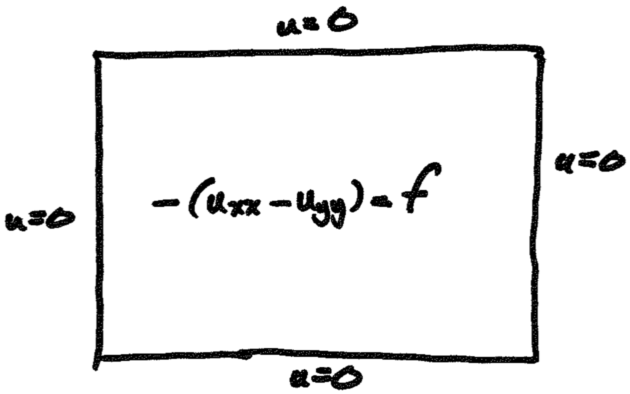
X1E at ORNL: Phoenix

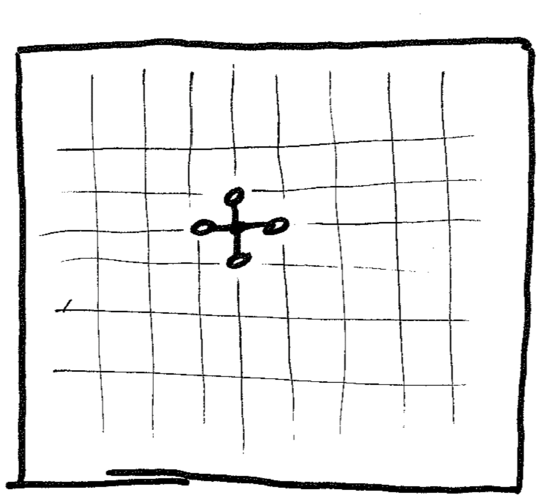
- 1024 Multistreaming vector processors (MSP)
- Each MSP
  - 4 Single Streaming Processors (SSP)
  - 4 scalar processors (400 MHz)
  - Memory bw is roughly half cache bw.
  - 2 MB cache
  - 18 GFLOP peak (~18.5 TFLOPS)
- 4 MSPs form a node
  - 8 GB of shared memory.
  - Inter-node load/store across network. 56 cabinets

Memory Latency

Memory location	*Relative access time*
D-cache	1X
E-cache	2X
Local (node) memory	7X
Remote (off node) memory	10X-32X

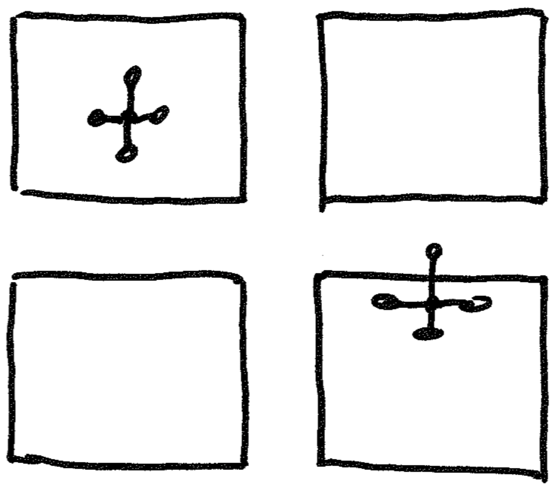
Continuous PDE to discrete form for Finite Difference Stencils





```
DO J = 2, LCOLS+1
   DO I = 2, LROWS+1

      GRID2(I,J) = (
                              GRID(I-1,J) +
                GRID(I,J-1) + GRID(I,J) + GRID(I,J+1 ) +
                              GRID(I+1,J)  ) / 5

   END DO
END DO
```

Parallel Processing

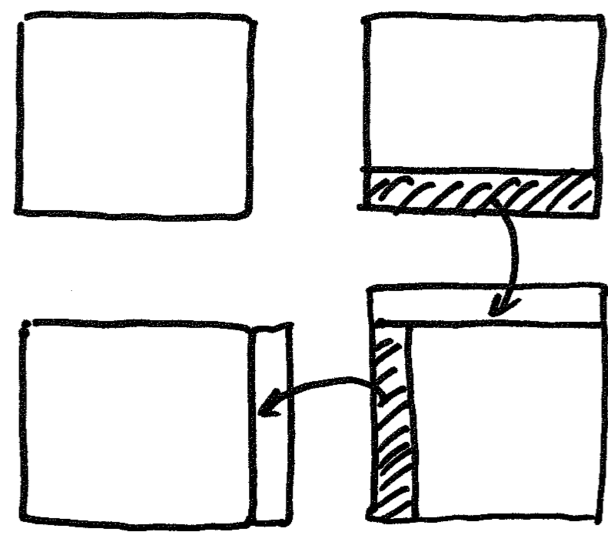


Parallel Processing

MPI Code:

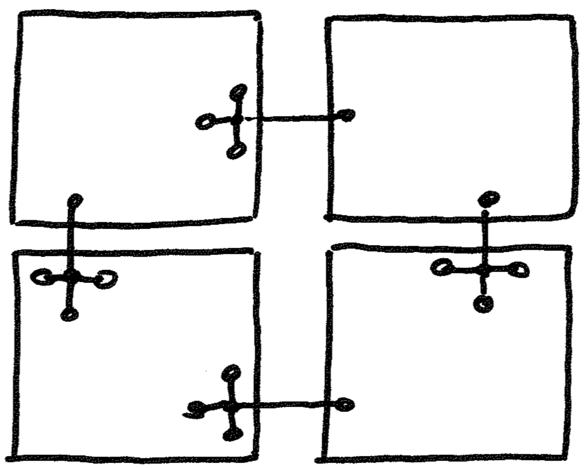
! Exchange dΩ

! Compute



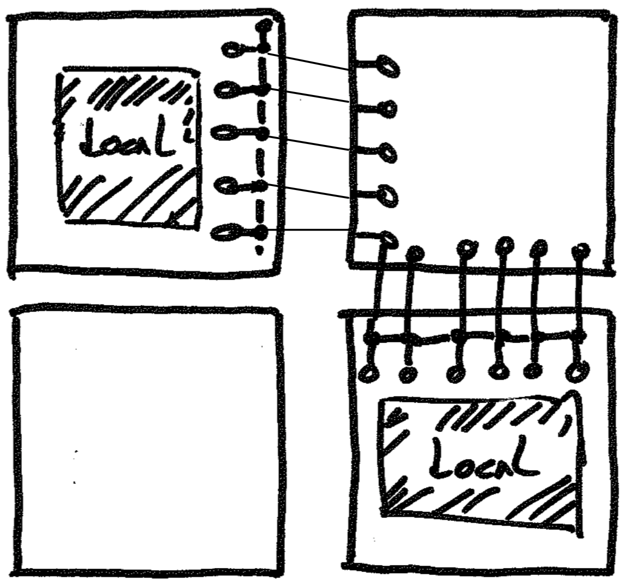
CAF

Load it when you need it



```
DO J = 2, LCOLS+1
   DO I = 2, LROWS+1

      LEFT     = GRID1(II(I,J-1),JJ(I,J-1))[IMG_LOC(I,J-1)]

      TOP      = GRID1(II(I-1,J),JJ(I-1,J))[IMG_LOC(I-1,J  )]

      CENTER   = GRID1(II(I,J  ),JJ(I,J))[IMG_LOC(I,J  )]

      BOTTOM   = GRID1(II(I+1,J),JJ(I+1,J))[IMG_LOC(I+1,J  )]

      RIGHT    = GRID1(II(I,J+1),JJ(I,J+1))[IMG_LOC(I,J+1)]

      GRID2(I,J) = ( LEFT + TOP + CENTER + BOTTOM + RIGHT ) / 5

   END DO
END DO
```

CAF Segmented
One loop per boundary



- Sweep over each boundary (4 loops)
  - *Hint to compiler?*
- Loop over "interior" region.
- Eliminates the indirection from CAF.

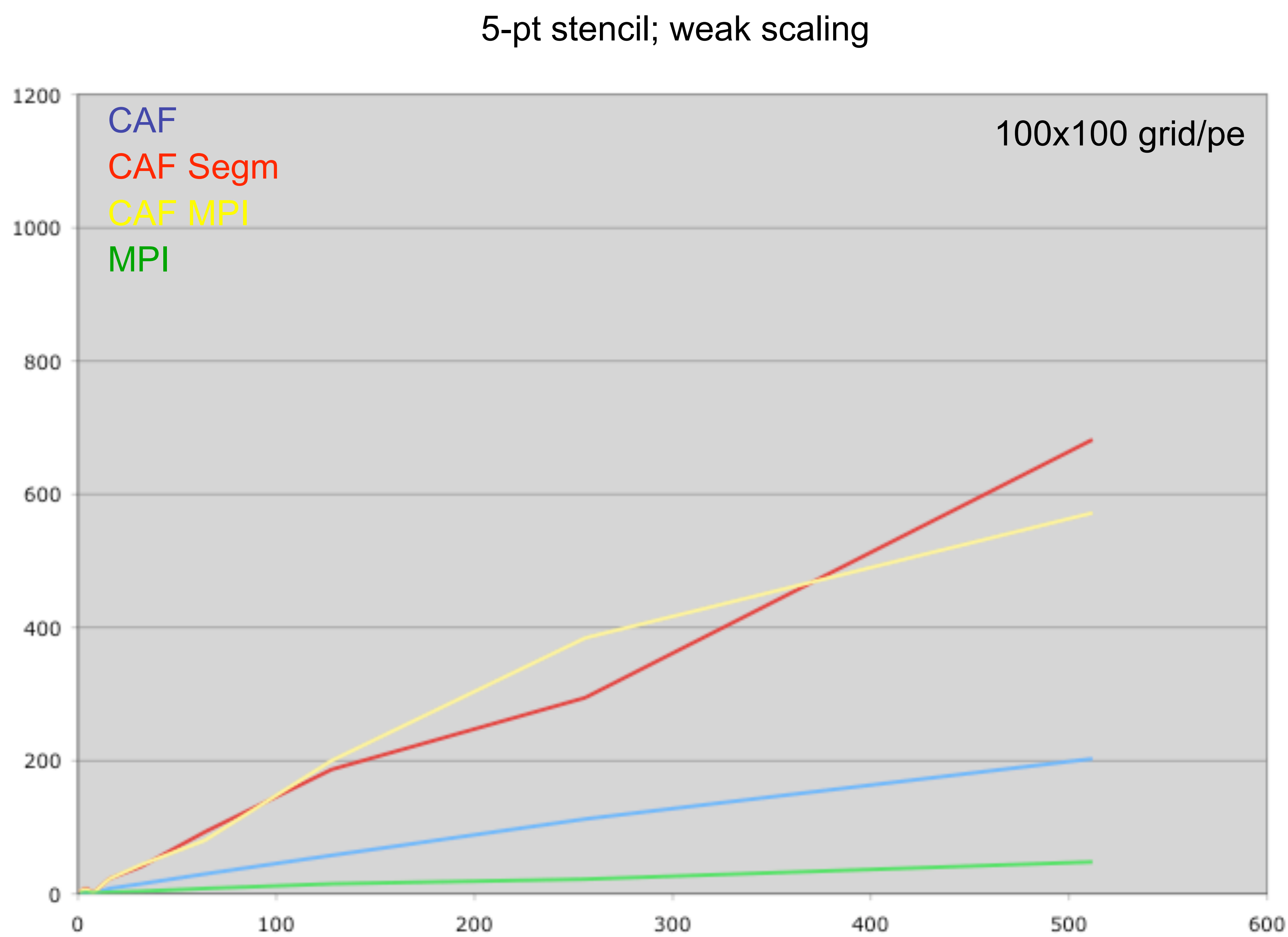
```
DO J = 2, LCOLS-1

      GRID2(1,J) =  (                                              &
                        GRID1(LROWS,J)  [NEIGH(NORTH)] + &
         GRID1(1,J-1) + GRID1(1,J) + GRID1(1,J+1) +            &
                        GRID1(2,J) )                                  &
                * FIFTH

END DO
```

CAF MPI-style
(Actually one-sided model)

```
CALL SYNC_TEAM ( NEIGHBORS )

IF ( NEIGHBORS(SOUTH) /= MY_IMAGE ) &
  GRID1( LROWS+2, 2:LCOLS+1 ) = GRID1( 2,2:LCOLS+1 )[NEIGHBORS(SOUTH)]

IF ( NEIGHBORS(NORTH) /= MY_IMAGE ) &
  GRID1( 1, 2:LCOLS+1 )  = GRID1( LROWS+1,2:LCOLS+1 )[NEIGHBORS(NORTH)]

IF ( NEIGHBORS(WEST) /= MY_IMAGE ) &
  GRID1( 2:LROWS+1, 1 )  = GRID1( 2:LROWS+1, LCOLS+1)[NEIGHBORS(WEST)]

IF ( NEIGHBORS(EAST) /= MY_IMAGE ) &
  GRID1( 2:LROWS+1, LCOLS+2 ) = GRID1( 2:LROWS+1, 2 )[NEIGHBORS(EAST)]
```

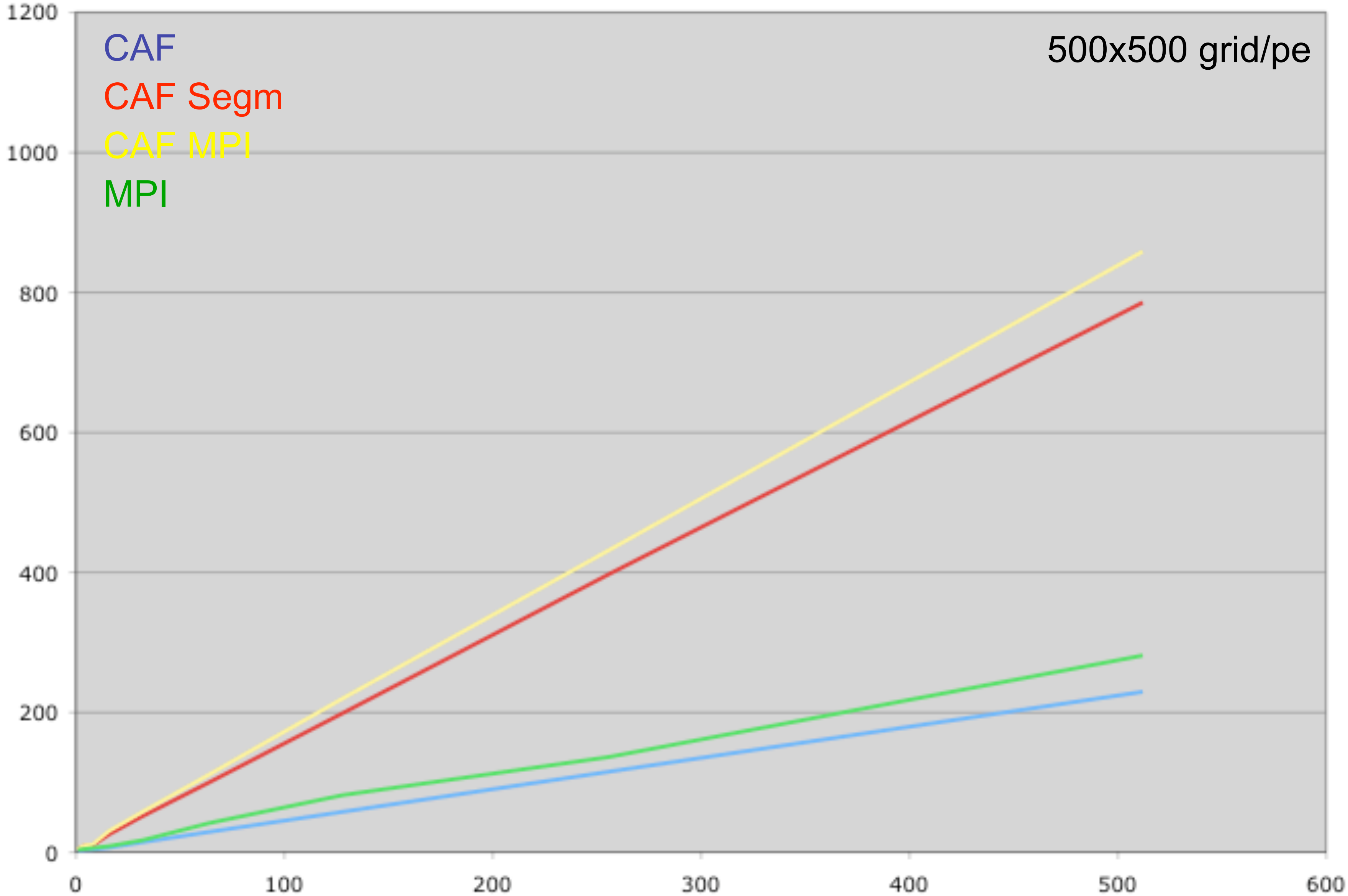
Performance on X1E

5-point difference stencil

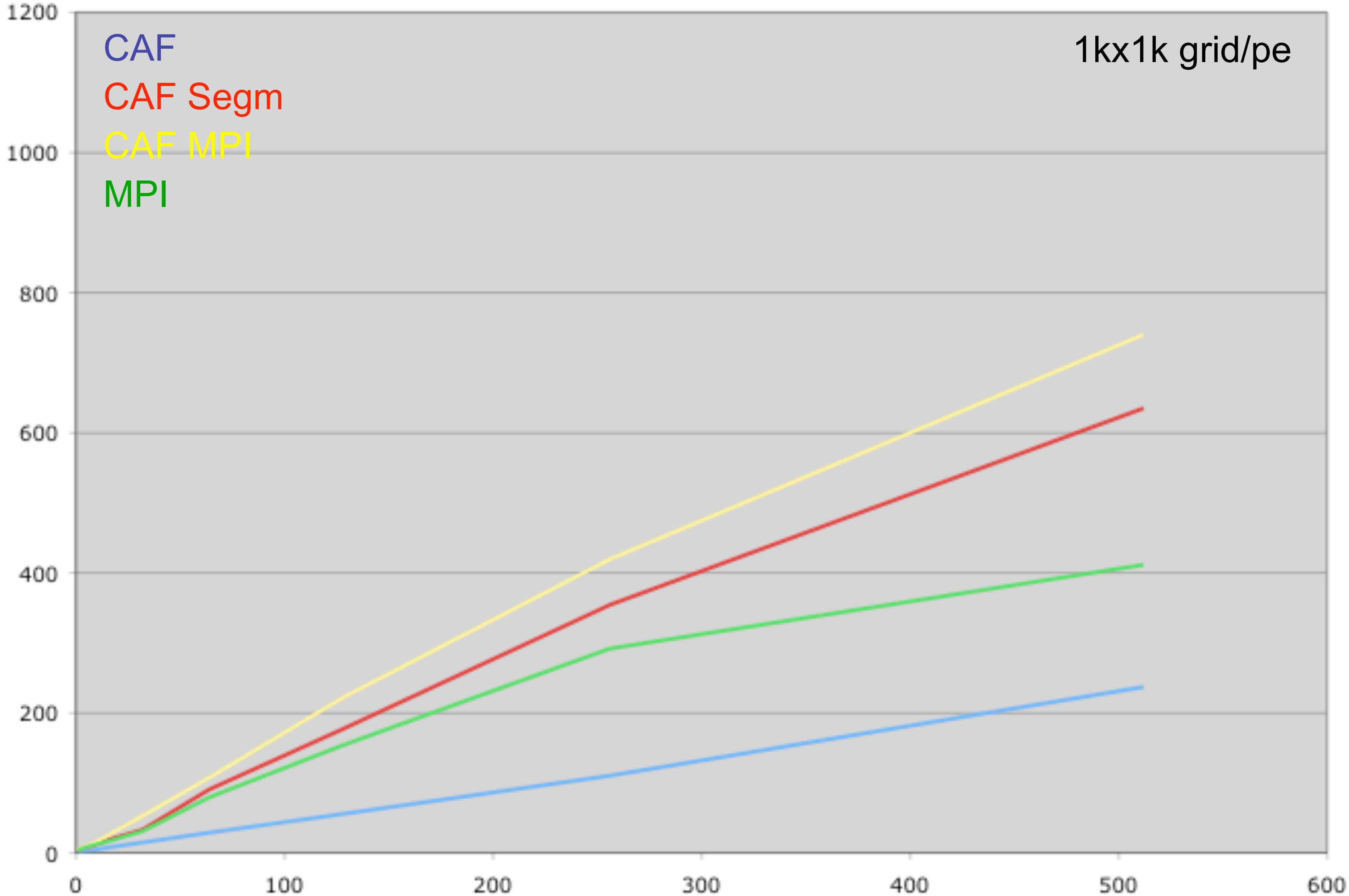


5-pt stencil; weak scaling
CAF
CAF Segm
CAF MPI
MPI
100x100 grid/pe
1200
1000
800
600
400
200
0
0
100
200
300
400
500
600

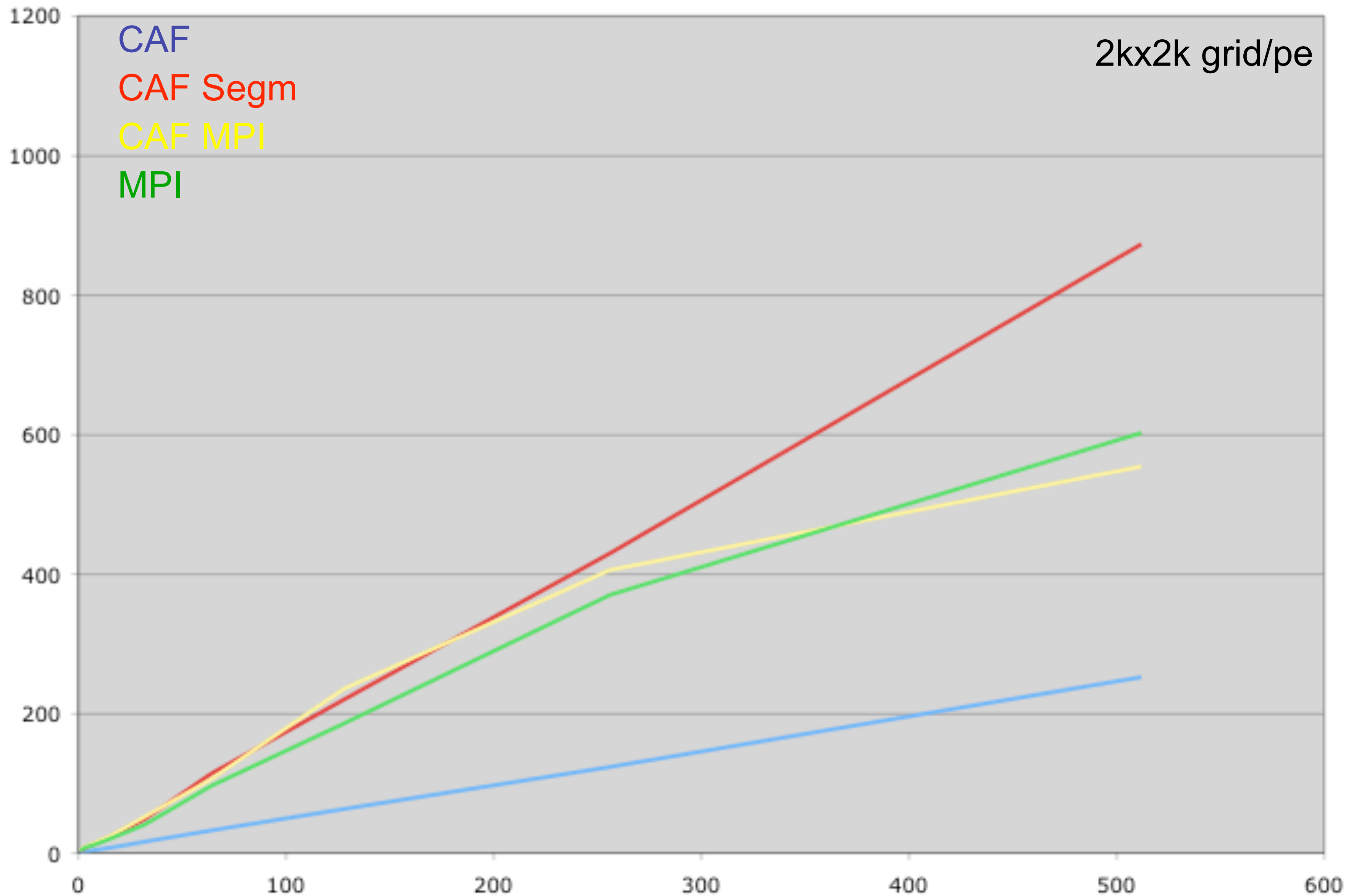
5-pt stencil; weak scaling



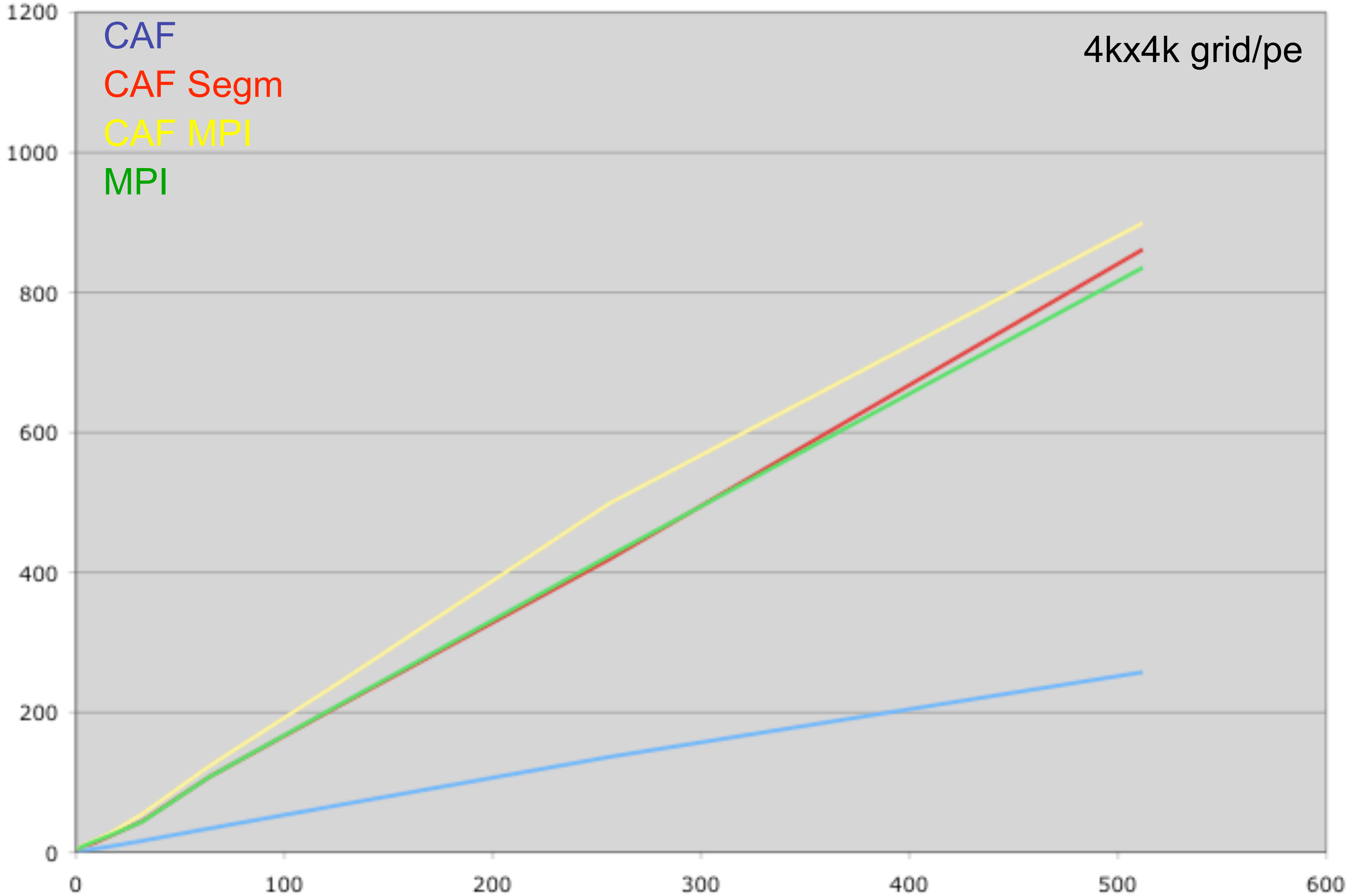
5-pt stencil; weak scaling



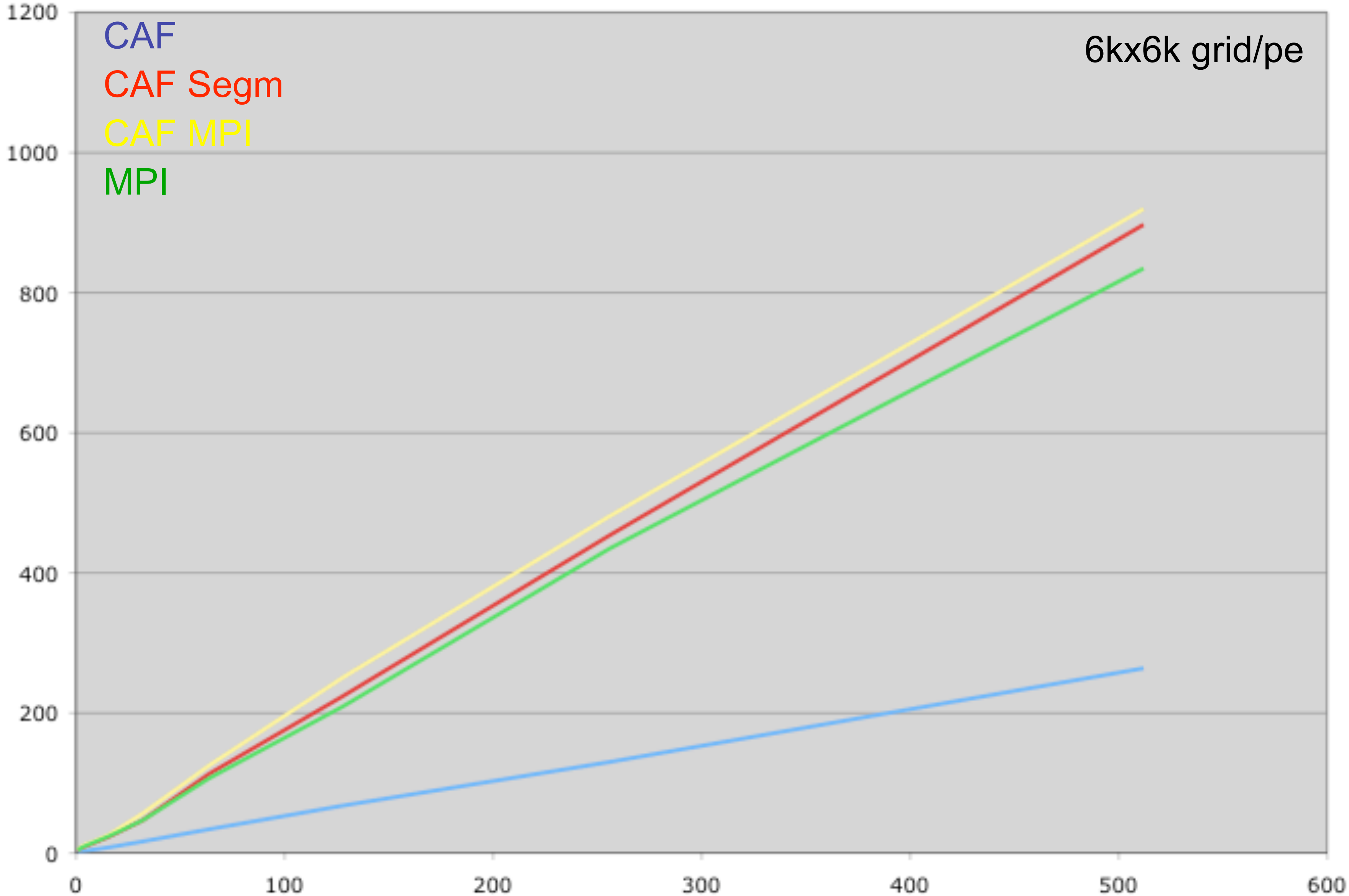
5-pt stencil; weak scaling

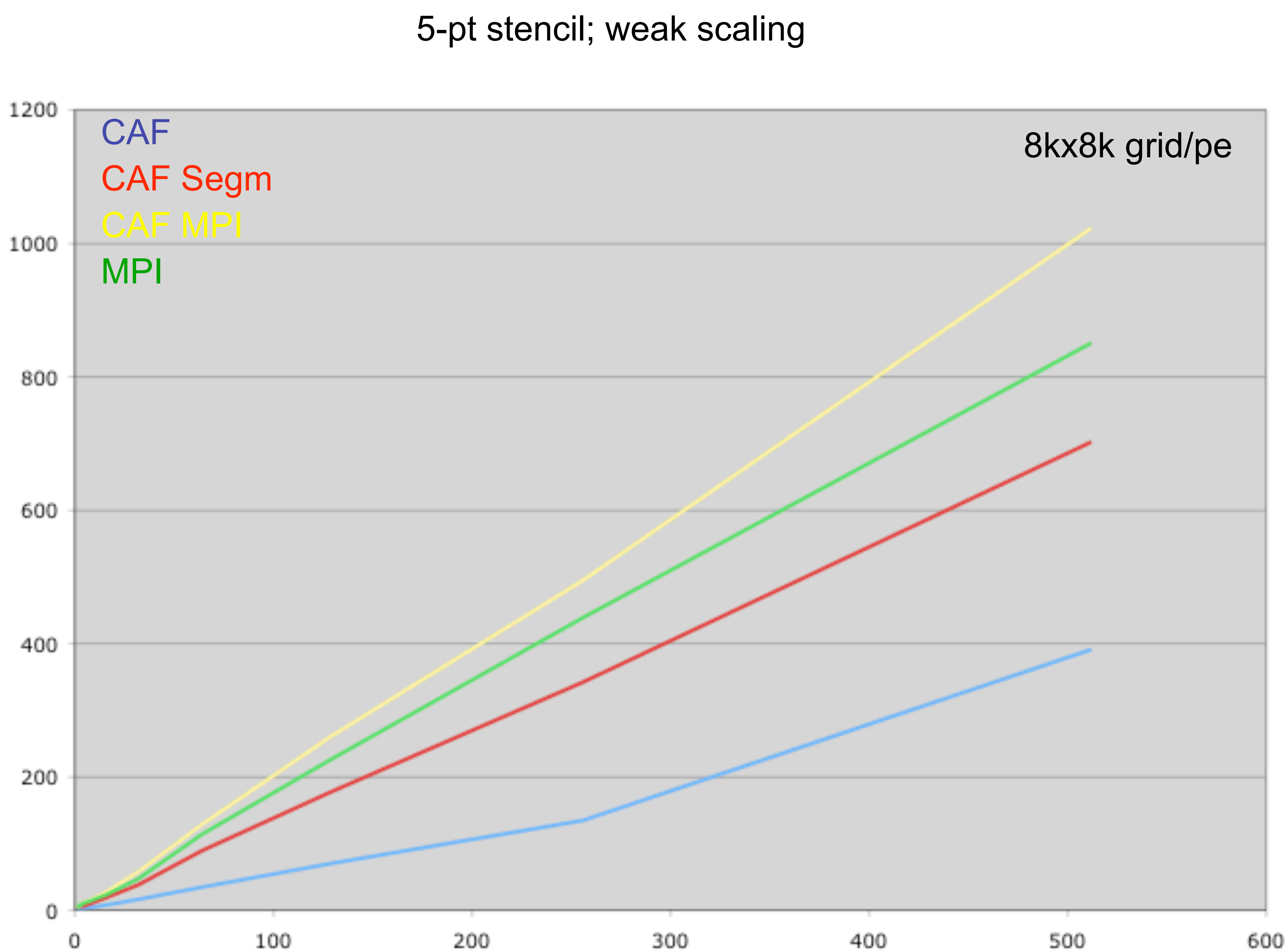


5-pt stencil; weak scaling



5-pt stencil; weak scaling



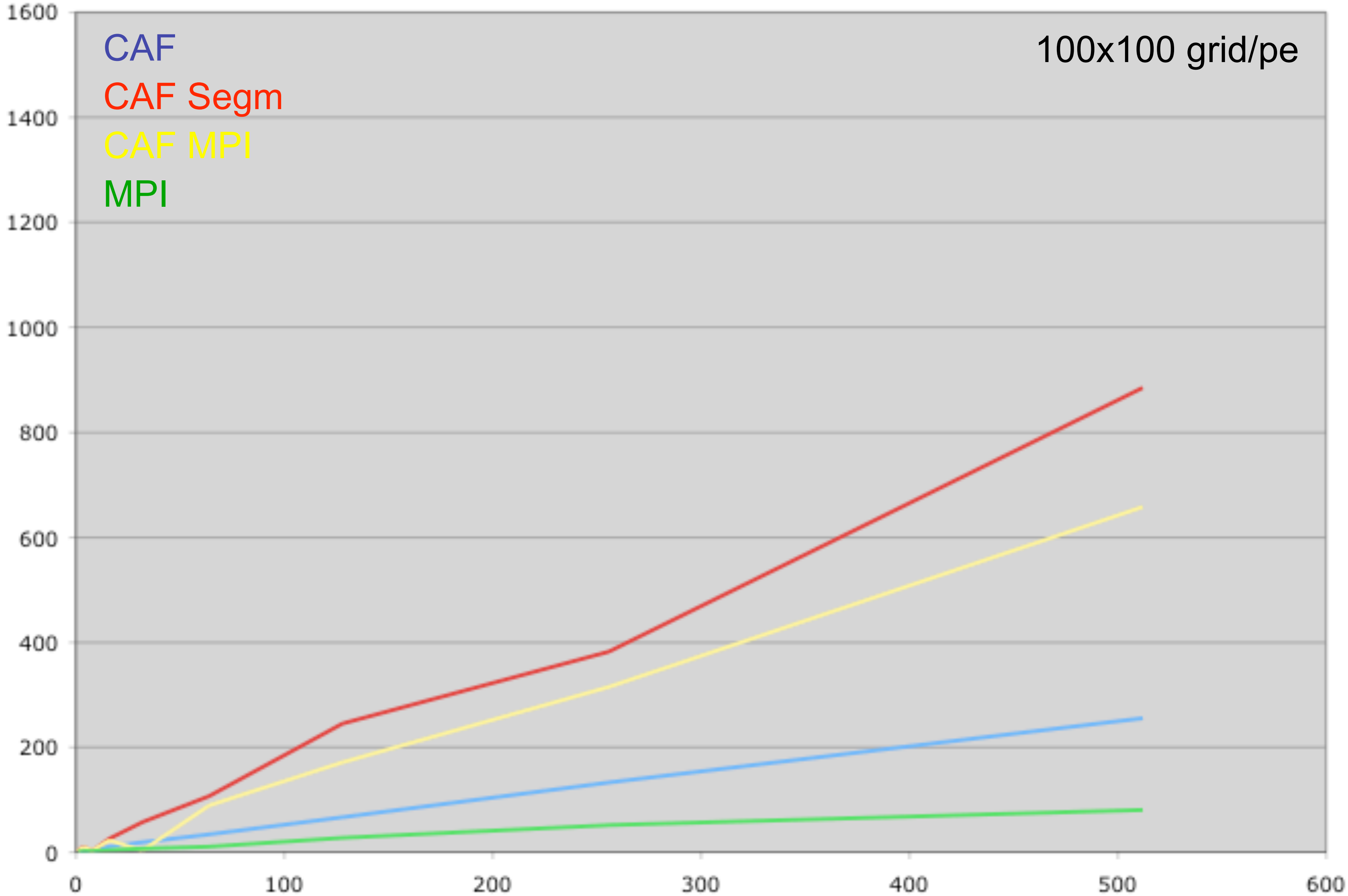

5-pt stencil; weak scaling
CAF
CAF Segm
CAF MPI
MPI
8kx8k grid/pe
1200
1000
800
600
400
200
0
0
100
200
300
400
500
600

Performance on X1E

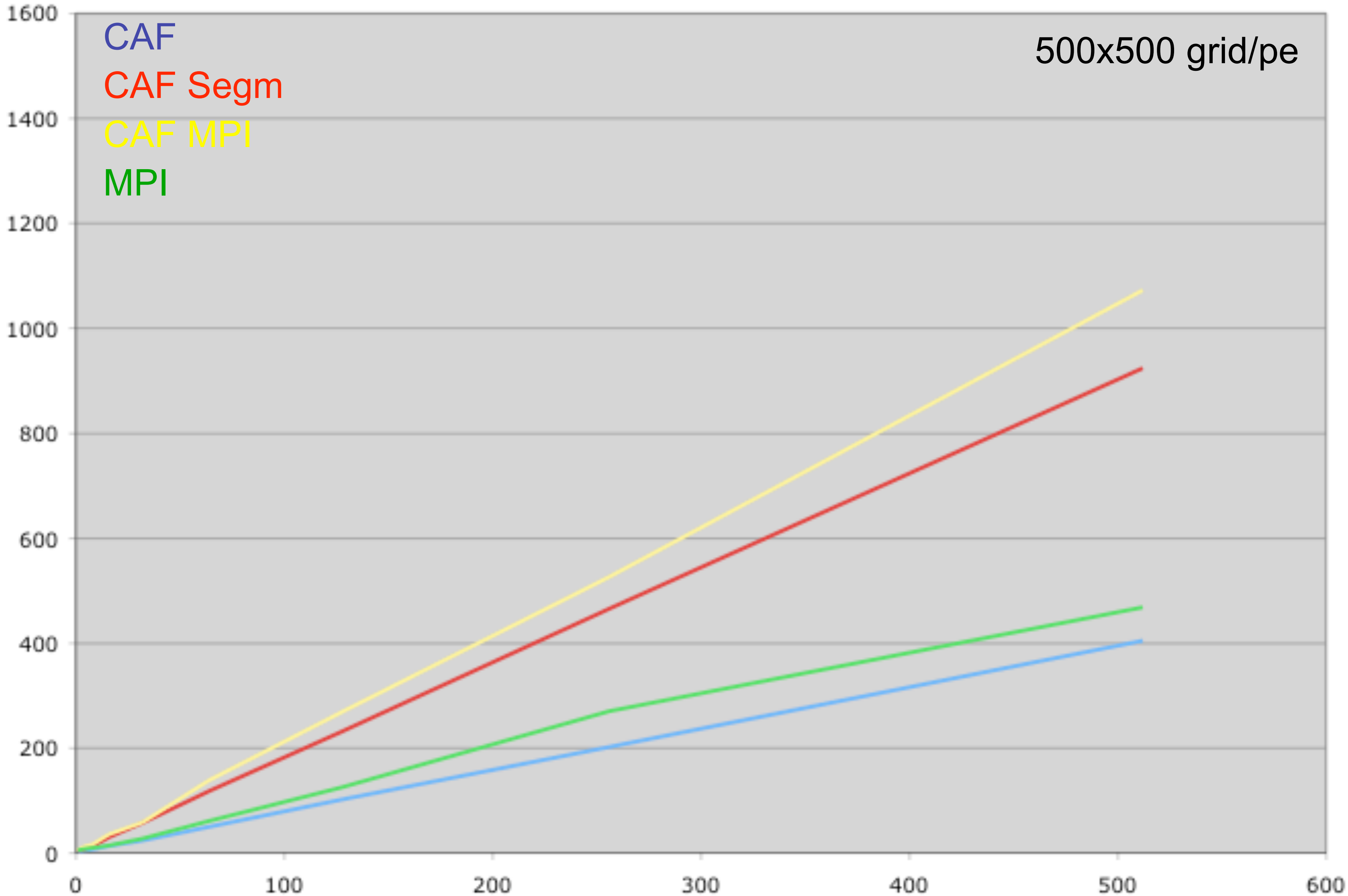
9-point difference stencil

Adds (up to) 4 new neighbors.

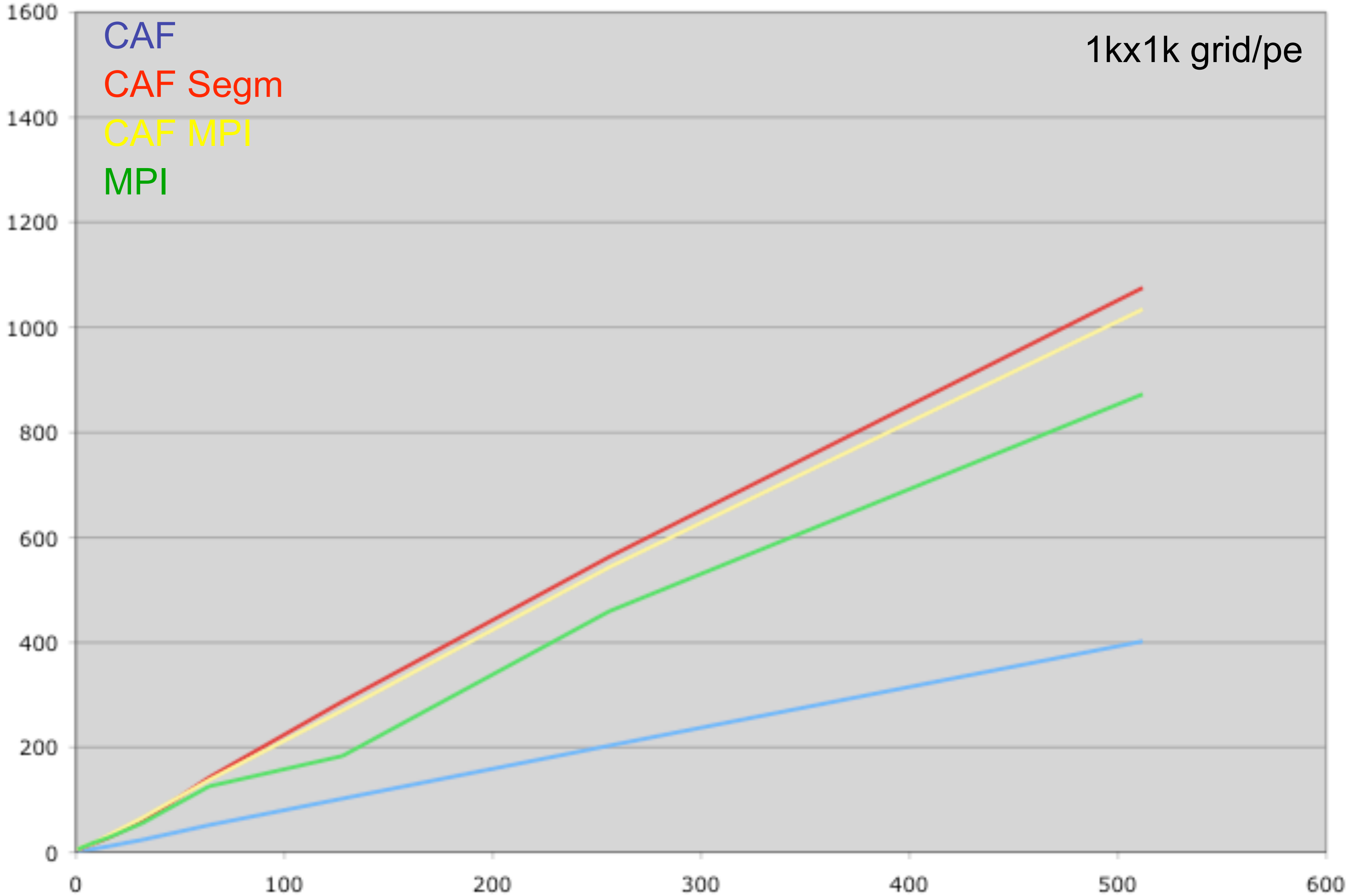
9-pt stencil; weak scaling



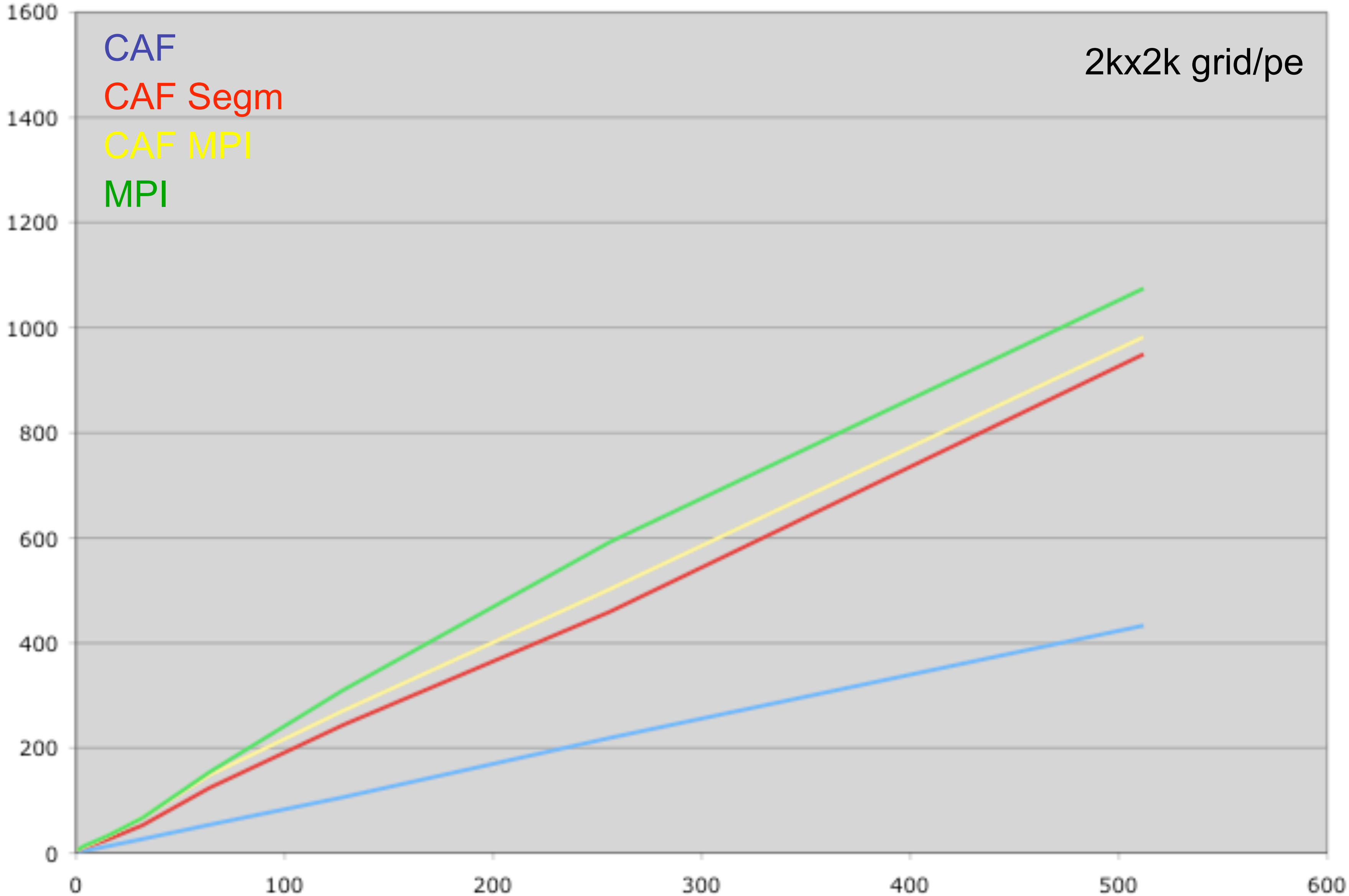
9-pt stencil; weak scaling



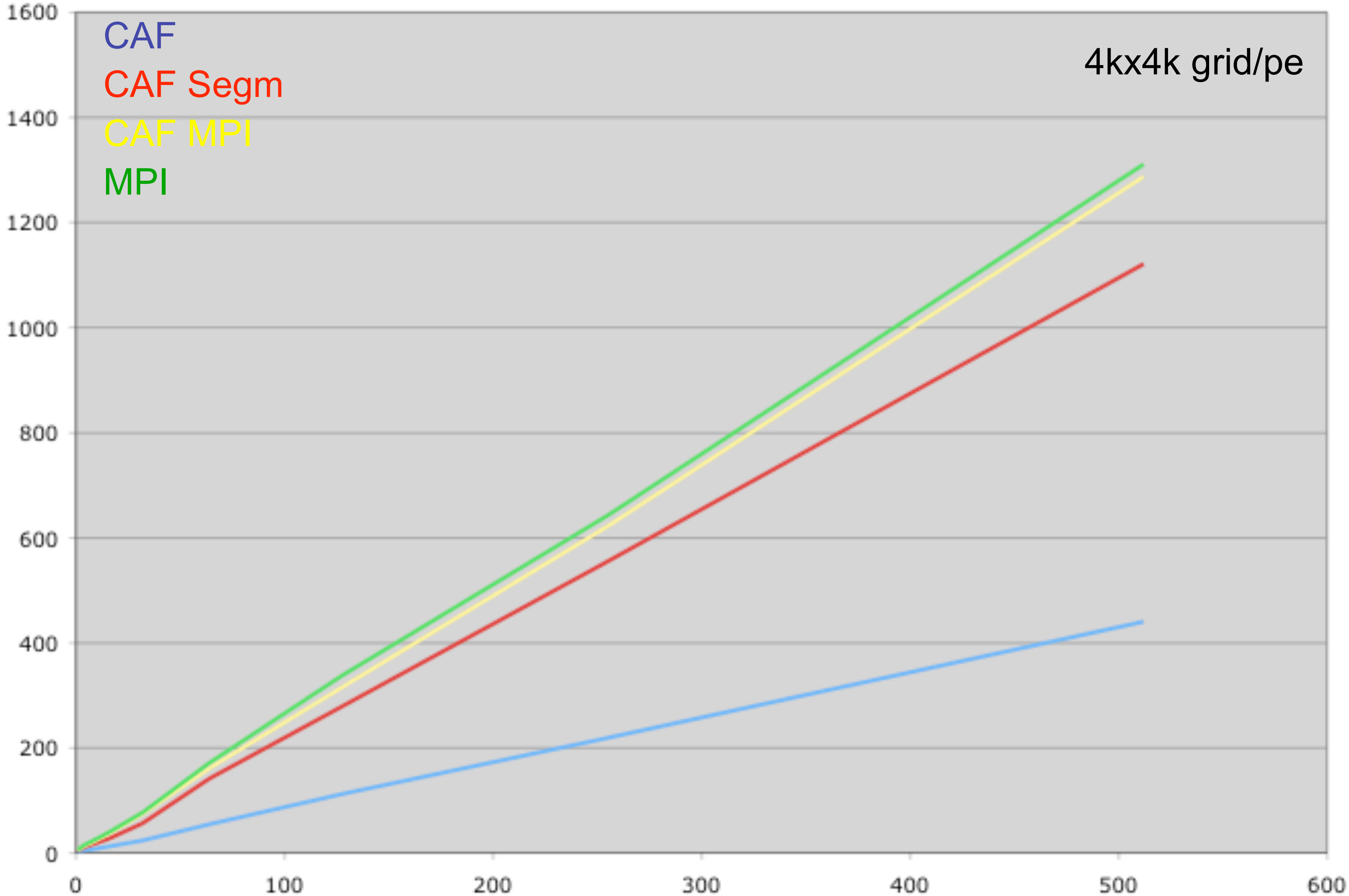
9-pt stencil; weak scaling

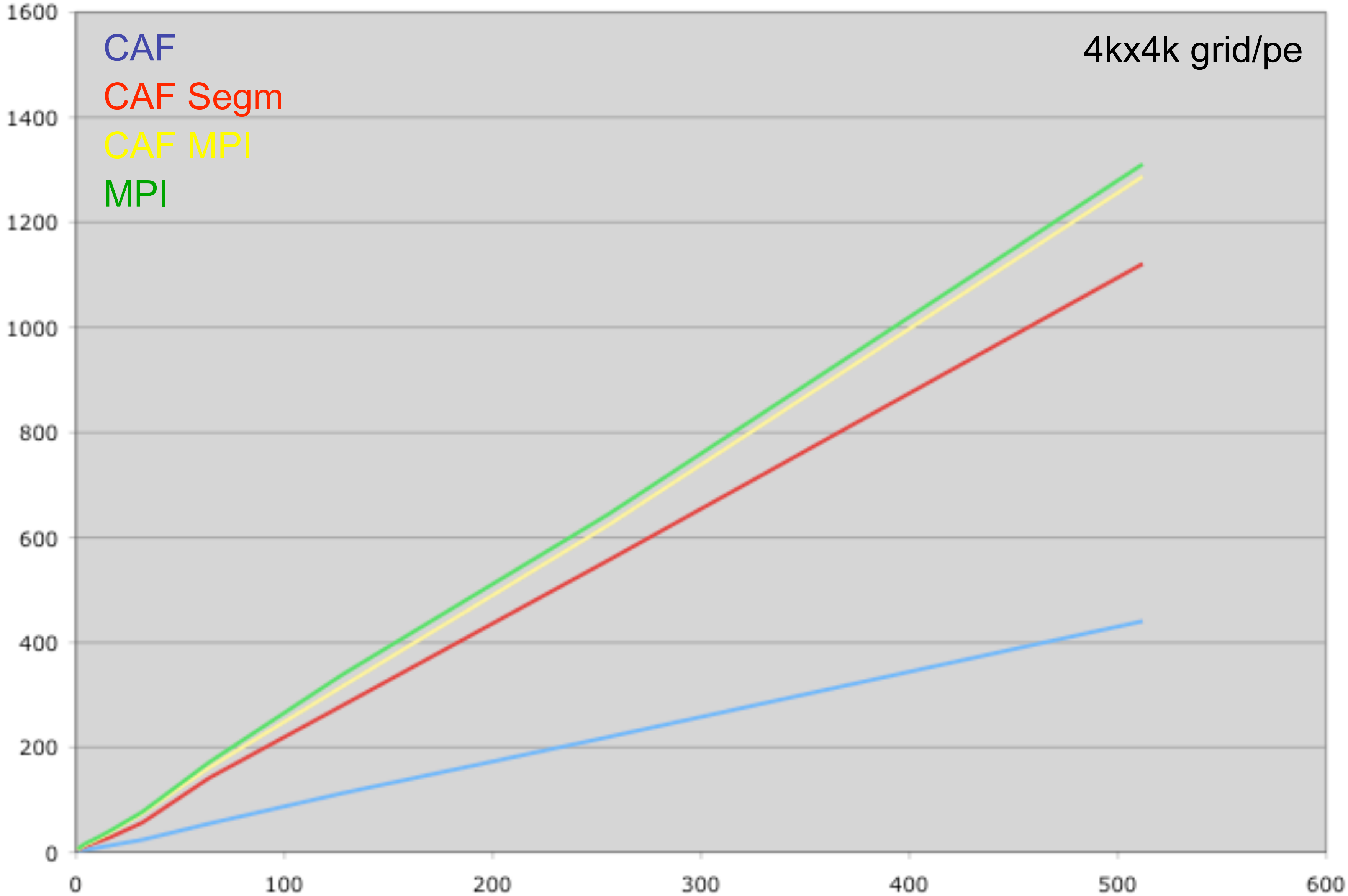


9-pt stencil; weak scaling



9-pt stencil; weak scaling

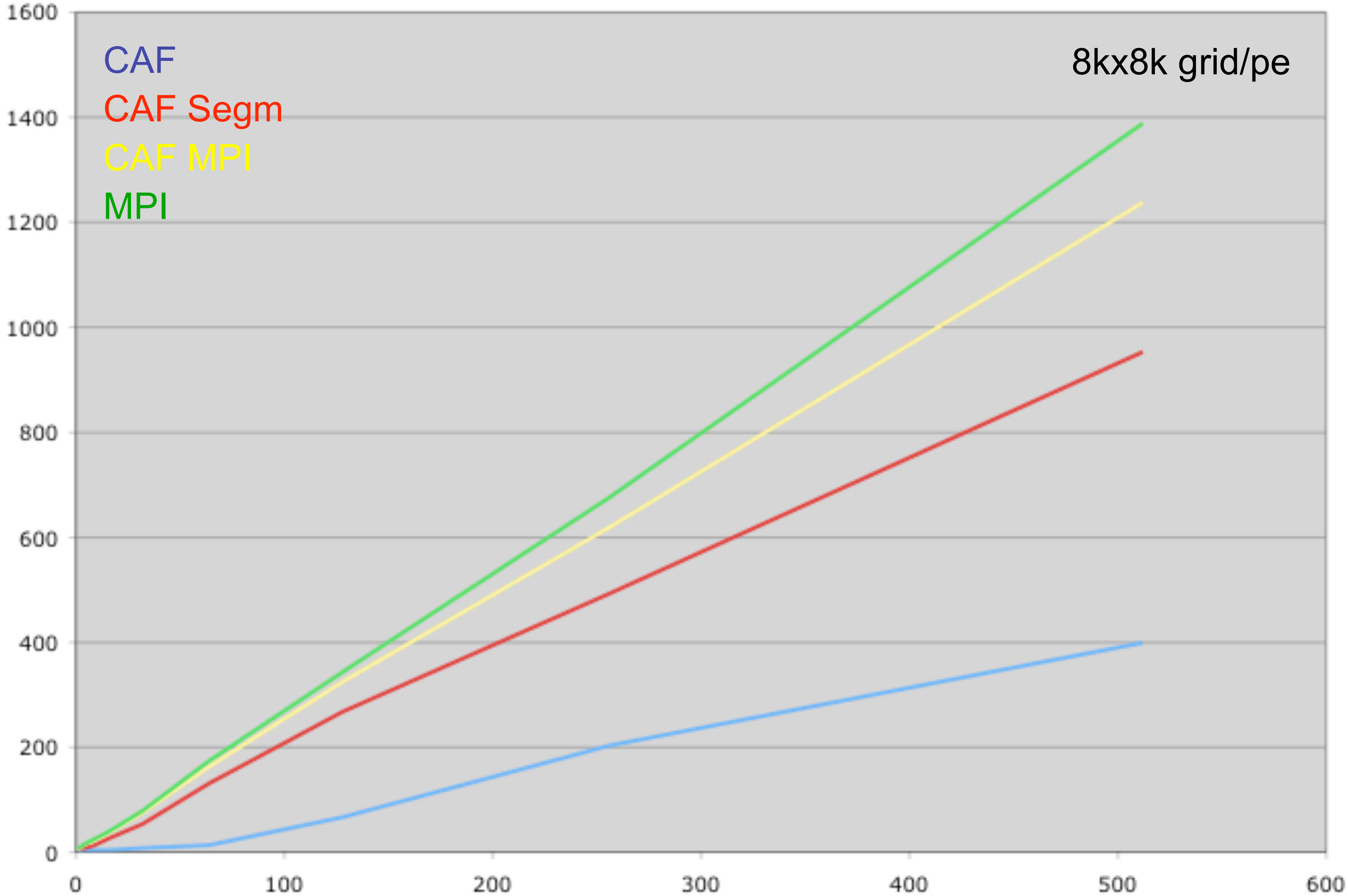




CAF
CAF Segm
CAF MPI
MPI
4kx4k grid/pe
1600
1400
1200
1000
800
600
400
200
0
0
100
200
300
400
500
600

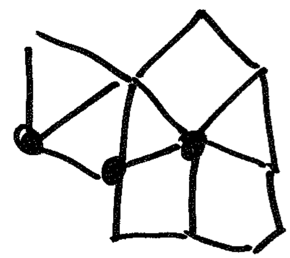
9-pt stencil; weak scaling

9-pt stencil; weak scaling



Unstruct- and semi-struct mesh Inter-process sharing requirments

UNSTRUCTURED MESH

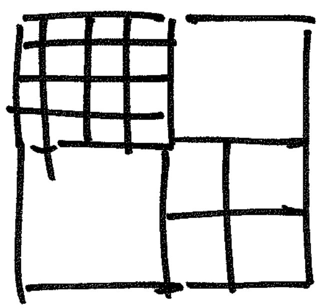


GATHER from NODES

SCATTER TO NODES

$$A(B(I)) = C(D(I))$$

SEMI STRUCTURED



$$A(I) = B(C(I))$$

Closer look at CAF

LEFT = GRID(II(I,J-1),JJ(I,J-1))[IMG_LOC(I,J-1)]

- Set IMG_LOC() = MY_IMAGE *(No comm)*

LEFT = GRID(II(I,J-1),JJ(I,J-1))[<*my_image*>]

- Remove indirection indexing *(No Indexing)*

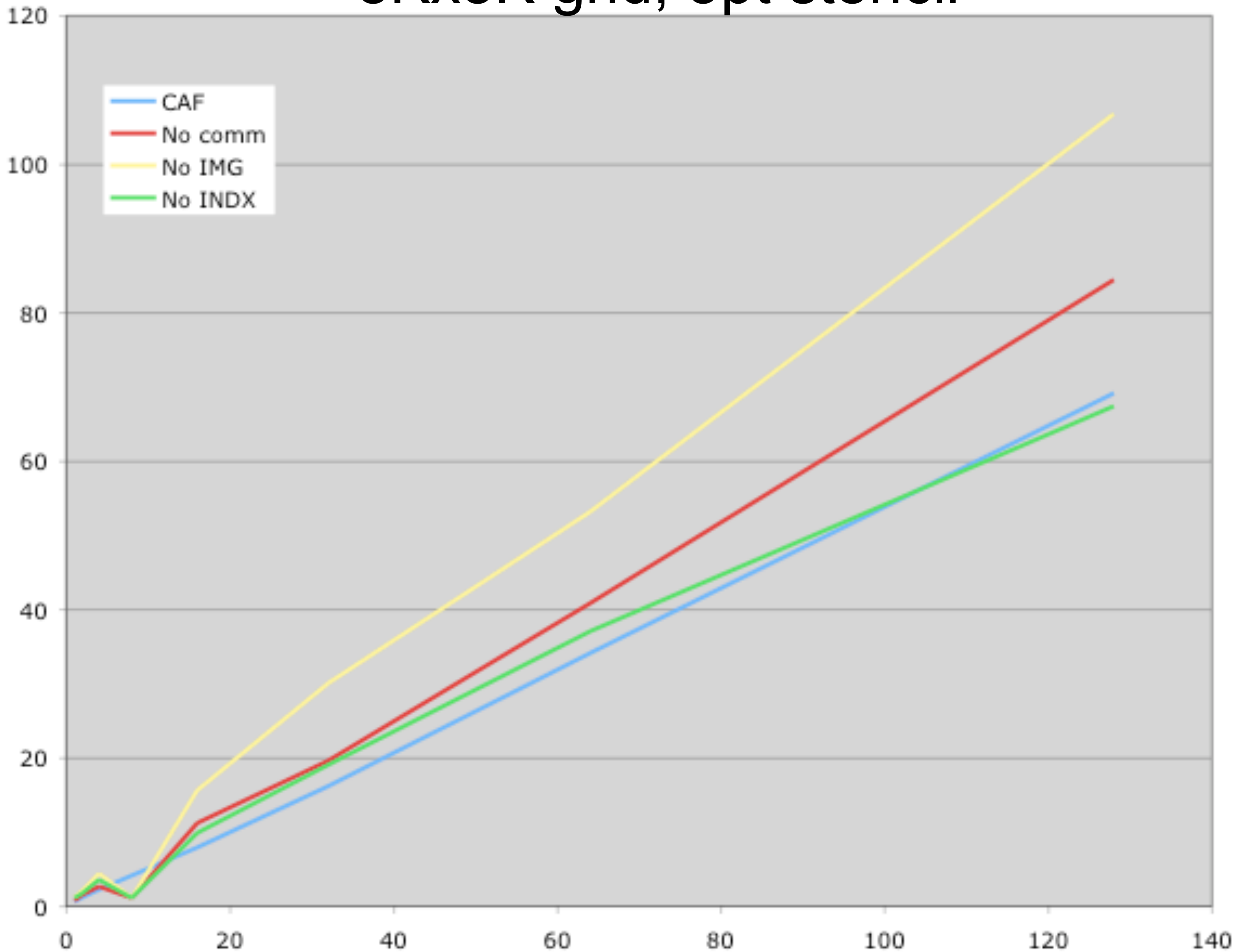
LEFT = GRID(I,J-1)[IMG_LOC(I,J-1)]

- Remove co-array notation. *(No image)*

LEFT = GRID(II(I,J-1),JJ(I,J-1))

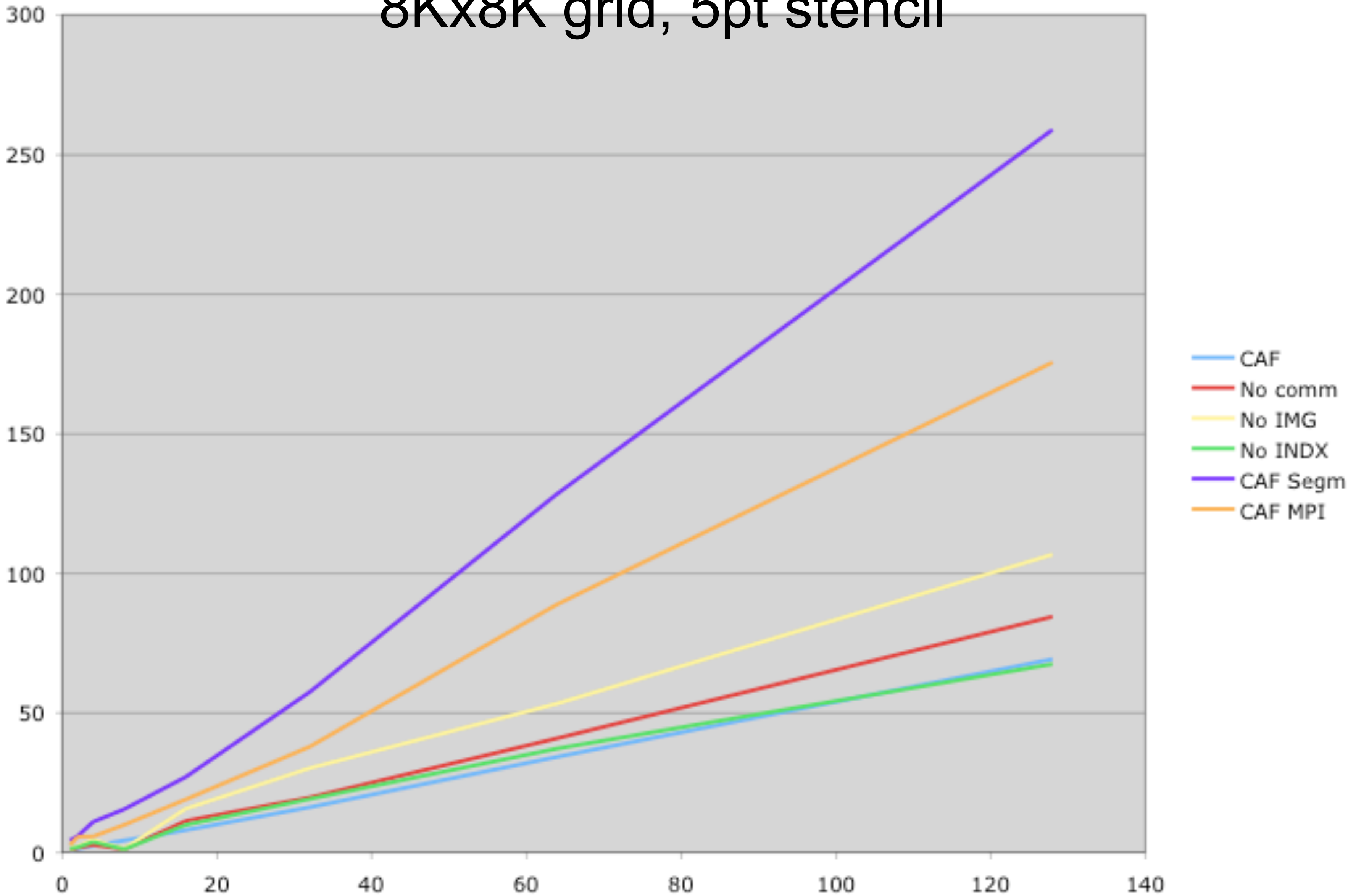
CAF Testing

8Kx8K grid, 5pt stencil



CAF Testing

8Kx8K grid, 5pt stencil



Summary

- Amazing number of views for CAF for simple algorithm.
  - Trey's response
- Coding effort no less that MPI.
- Strong potential for unstructured grids.
  - But concern over variable length arrays
- CAF esp. good with short messages.
  - Implication for strong scaling

Summary continued

- Performance improvements?
  - Persistent communication
  - Size-based protocol

- My view: CAF will succeed (in HPC) if(f):
  - (Significantly) outperforms MPI.
  - Wide availability
    - Multi-core availability *soon*.
      - Hybrid programming?

Future work

- Unstructured grids: in progress.
- Multi-core: in progress.
- Other algorithms.
- Comparisons with UPC, and other new models.

Acknowledgements

- This research was sponsored by the Office of Mathematical, Information, and Computational Sciences, Office of Science, U.S. Department of Energy under Contract No. DE-AC05-00OR22725 with UT-Battelle, LLC. Accordingly, the U.S. Government retains a non-exclusive, royalty-free license to publish or reproduce the published form of this contribution, or allow others to do so, for U.S. Government purposes.
- Cray, including Supercomputing Center of Excellence
  - Bill Long, Nathan Wichmann, Cathy Willis
- ORNL
  - Nikhil Bhatia, Mark Fahey, Trey White

PGAS 2006

- October 3-4, Washington, DC (GWU)
- CFP coming soon. (Paper submissions)
- UPC Developers workshop
- CAF Developers workshop