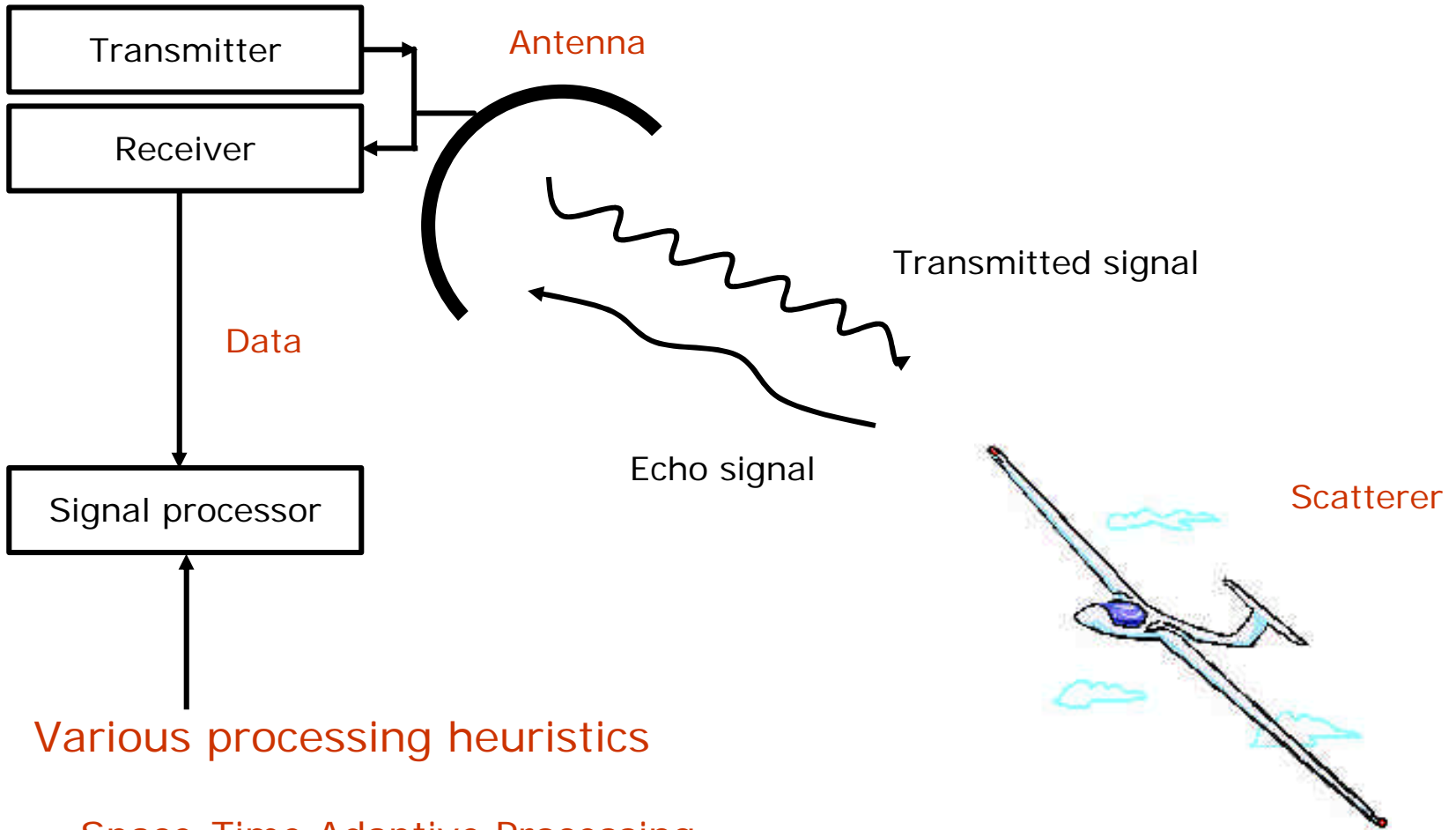


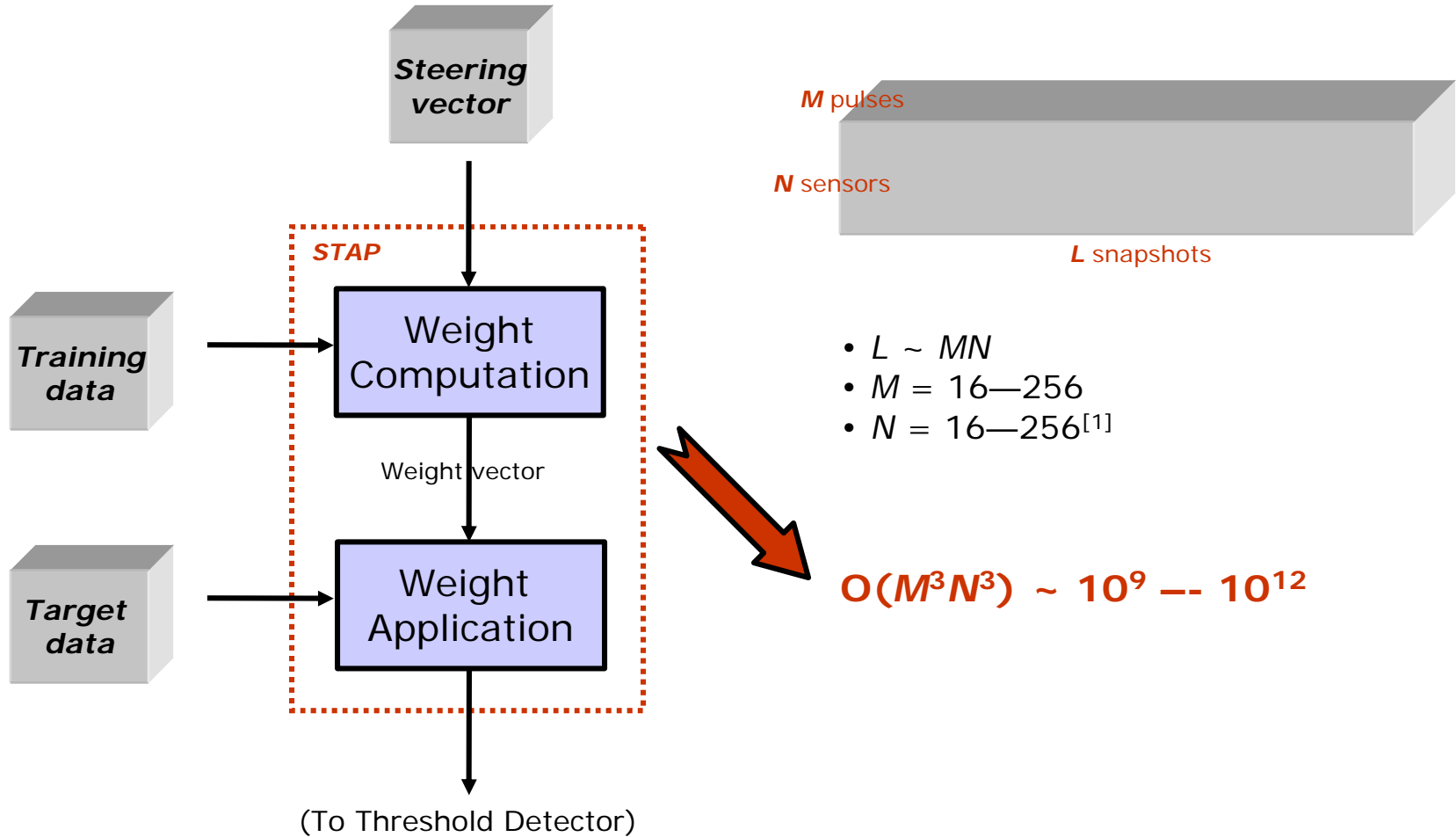
Performance Modeling and Optimization Framework for Space-Time Adaptive Processing (STAP)

Kyusoon Lee (and Adam W. Bojanczyk)

School of Electrical and Computer Engineering
Cornell University

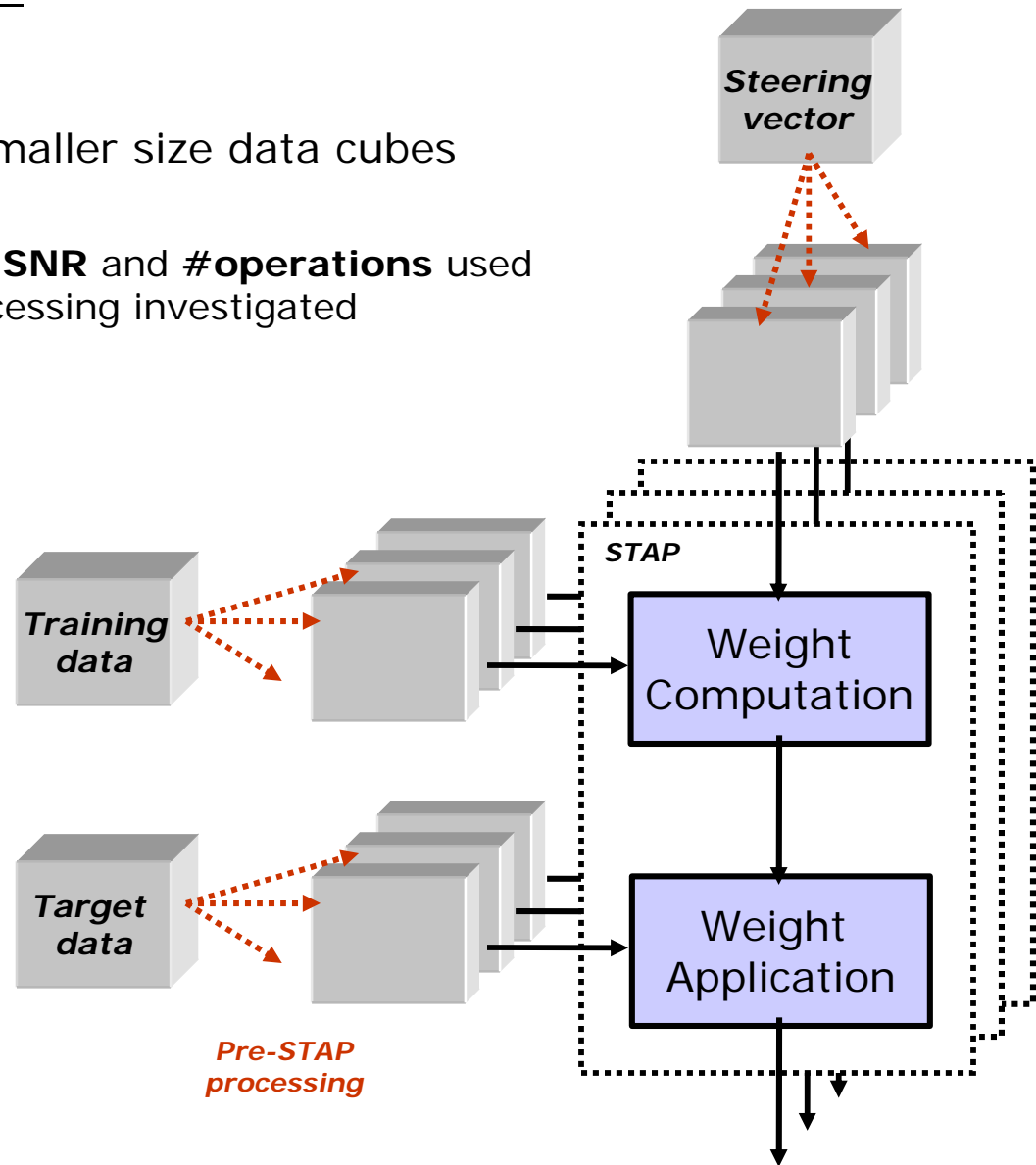


Fully-adaptive Space-Time Adaptive Processing

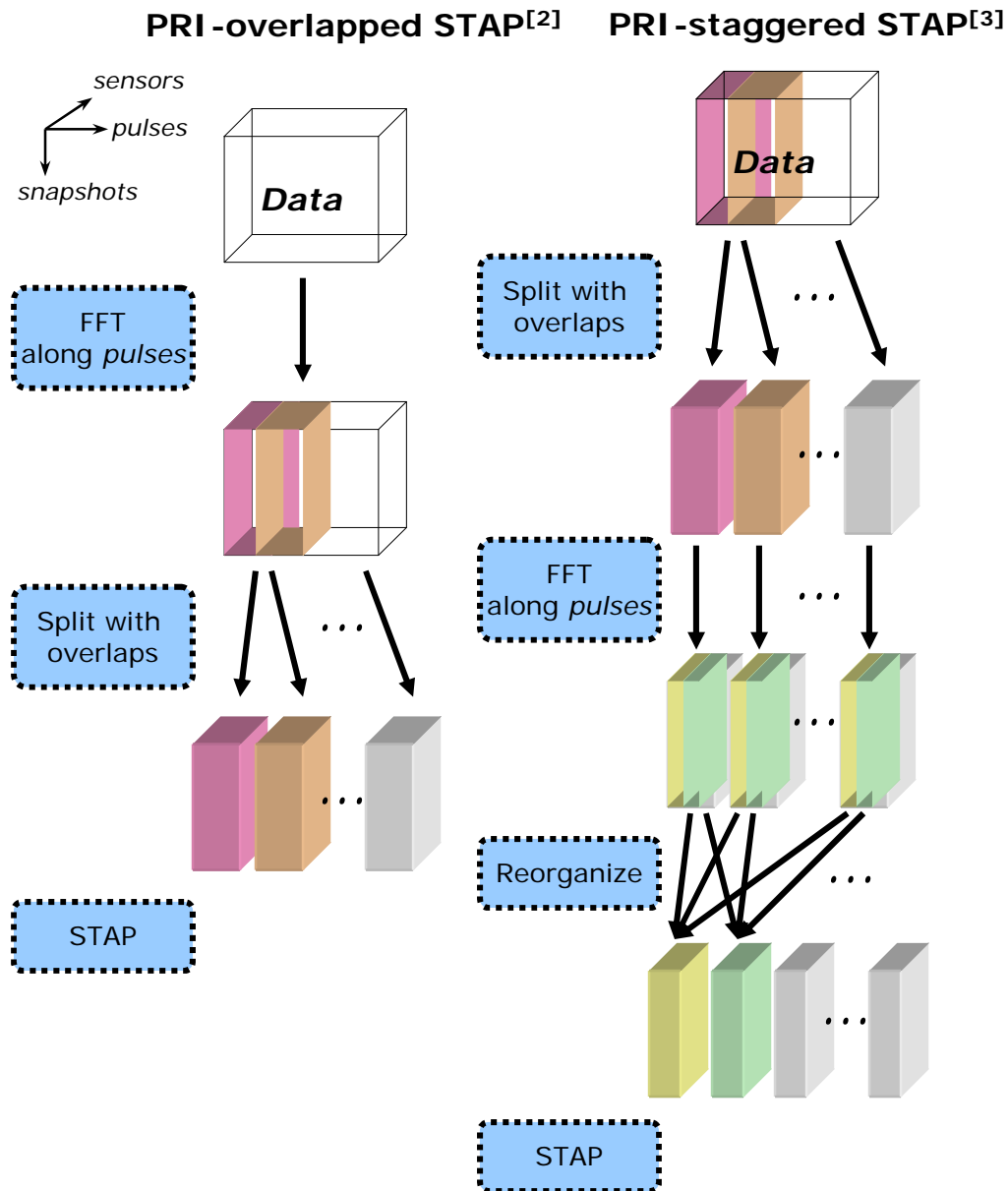


Partially-adaptive STAP

- Map data cube into smaller size data cubes
- Compromise between **SNR** and **#operations** used
- Various pre-STAP processing investigated

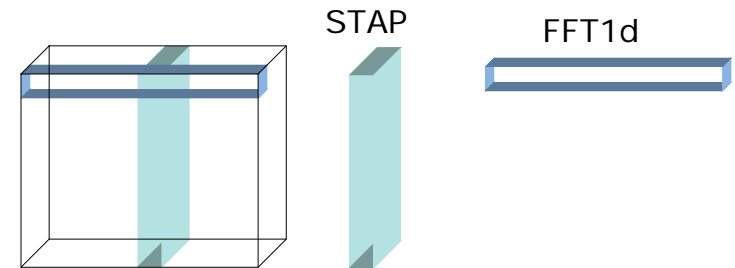


Pre-STAP Processing



- Similar operations in different order
- Repetitions

- **Basic building blocks**



- STAP on planes orthogonal to *pulses*
- FFT1d along *pulses*

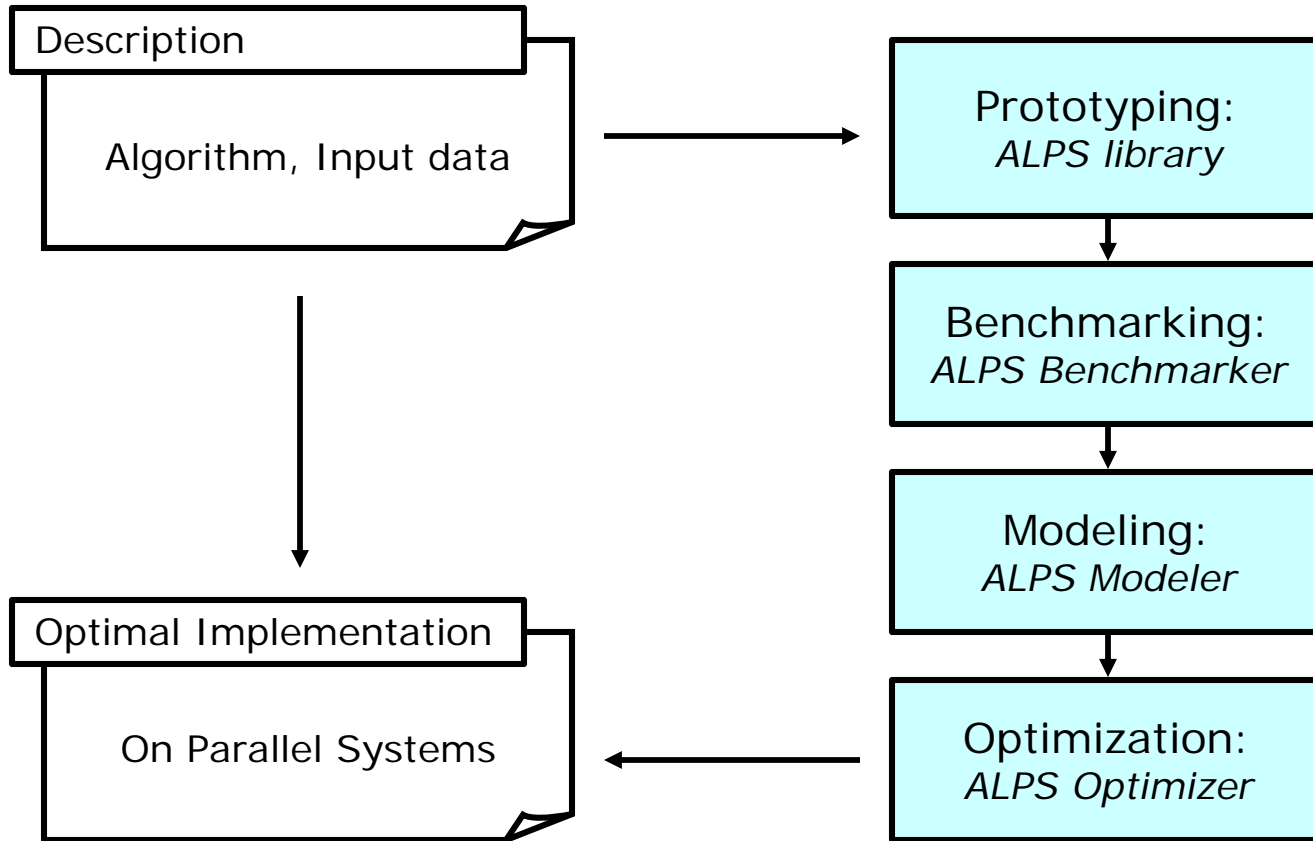
- **Different orientation**
- **Different distribution on 3D processors**

[2] R.C. DiPietro (1992)

[3] L.E. Brennan and F.M. Staudaher (1992)

[4] James Lebak and Adam Bojanczyk (2000)

Goal and Approach



Other Approaches: Lincoln Lab at MIT, USC, ...

Algorithmic Library for Parallel STAP (ALPS)

Data structures

- Handles **3D data cube** (list) on **3D processor cube**

Transformation Routines

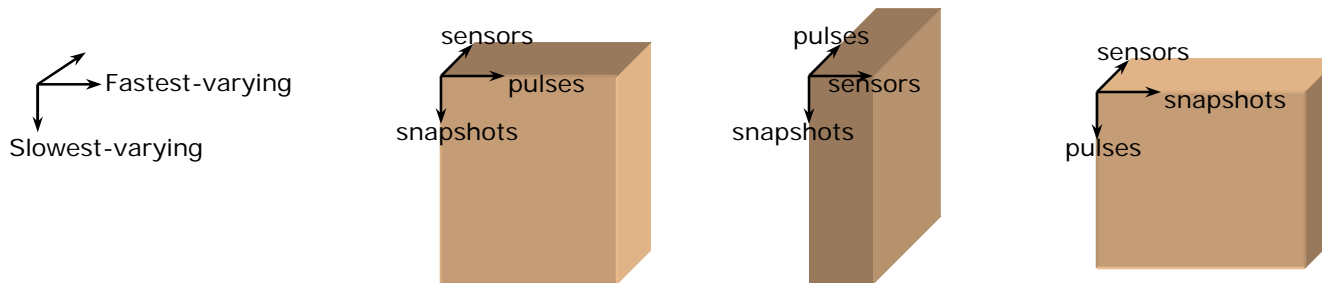
- **Splits, joins, or reorganizes a data cube** with or without overlaps

Processing Routines

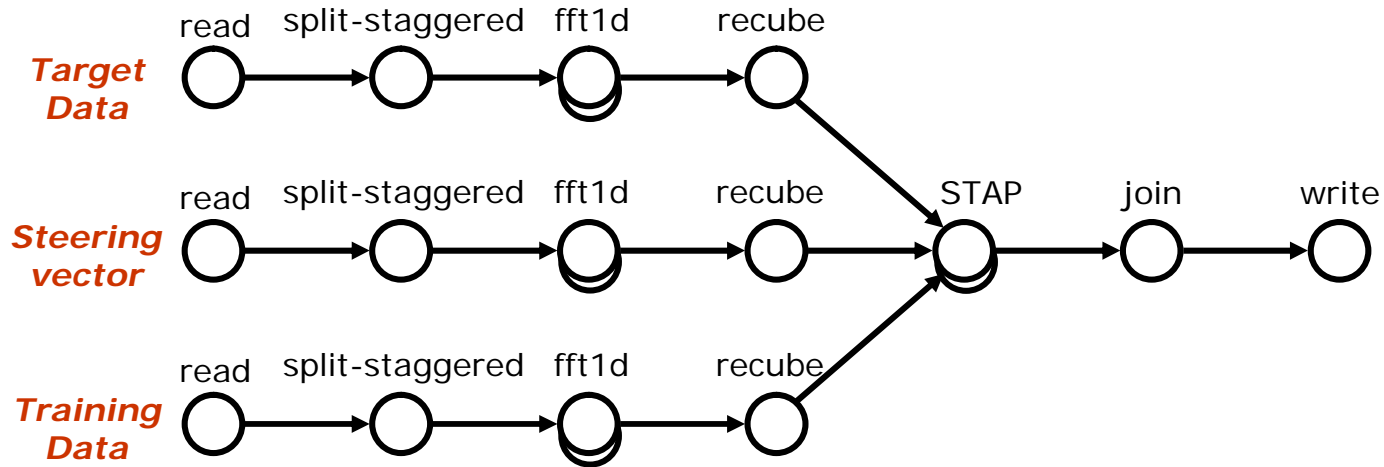
- **LAPACK, FFTW, STAP algorithms**

Communication Routines (MPI)

- Changes **the processor cube dimensions**: $(p_x, p_y, p_z) \rightarrow (p_x', p_y', p_z')$
- Changes **the orientation** of 3D data cube



Example: PRI-staggered STAP with ALPS



- Describe: *"what this algorithm does"*
 - Does not have to describe: *"how this algorithm does it"*
 - Processor cube dimensions for each task
 - Cube orientations for each task
 - Algorithm for each task
- } ALPS Optimizer

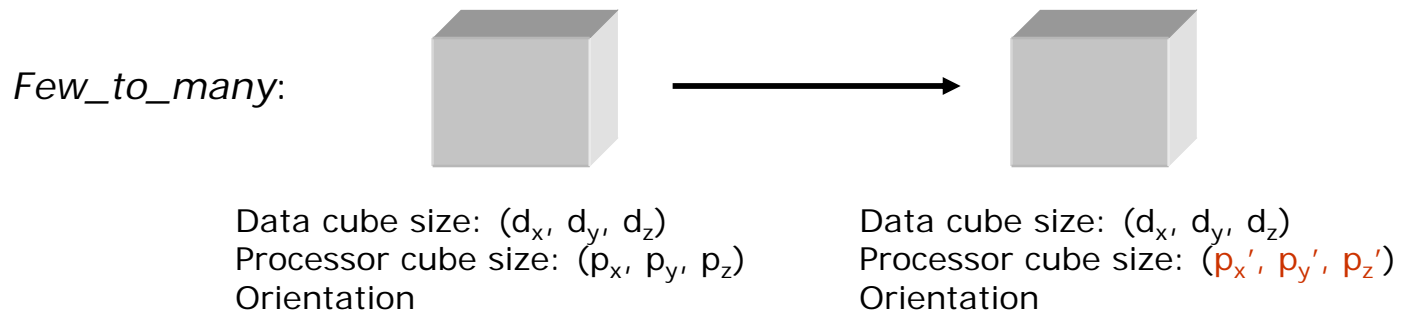
Benchmarking and Modeling: Approach

Benchmarker

- Data sizes and algorithm parameters input by users

Modeler

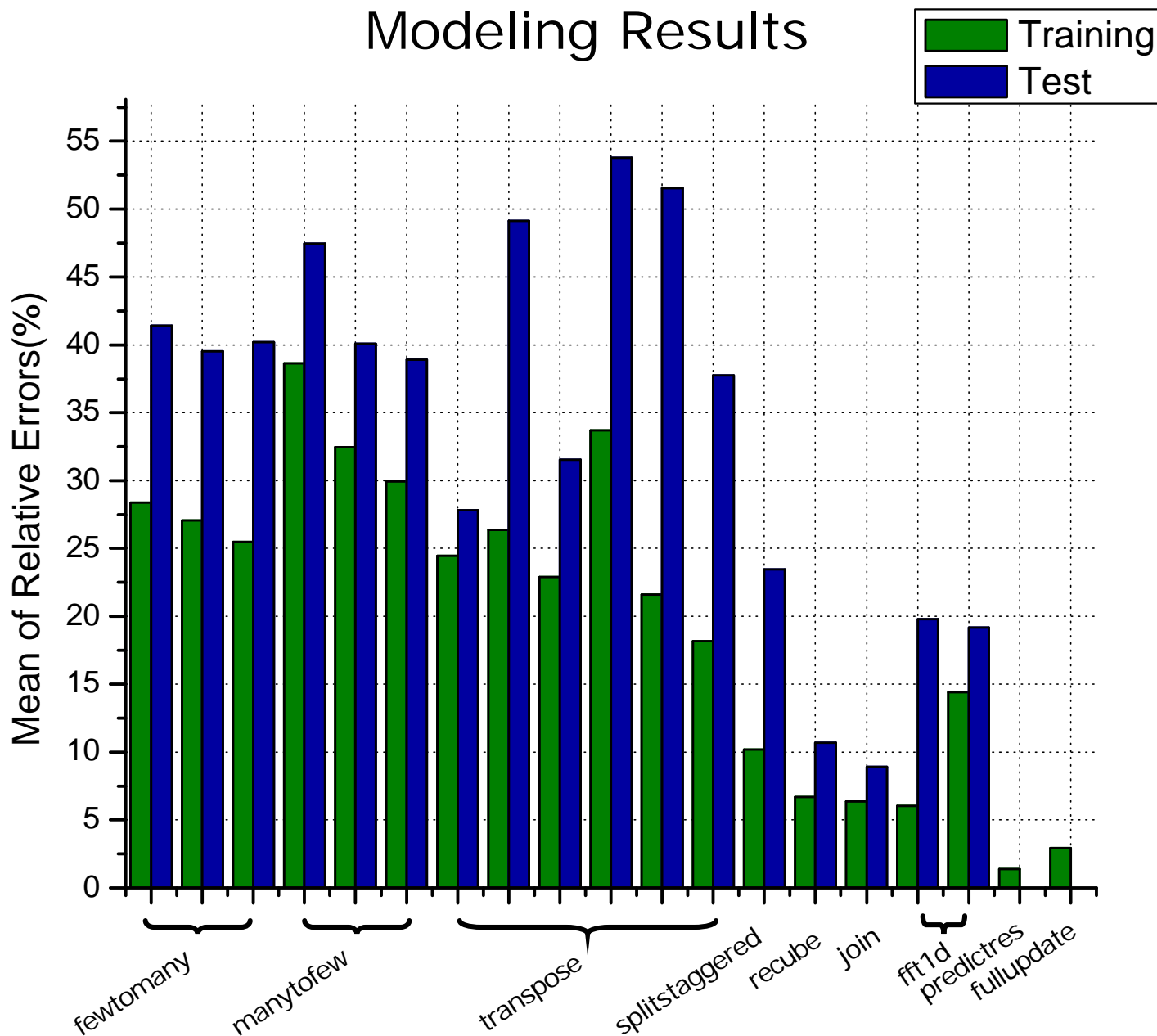
- Execution time models



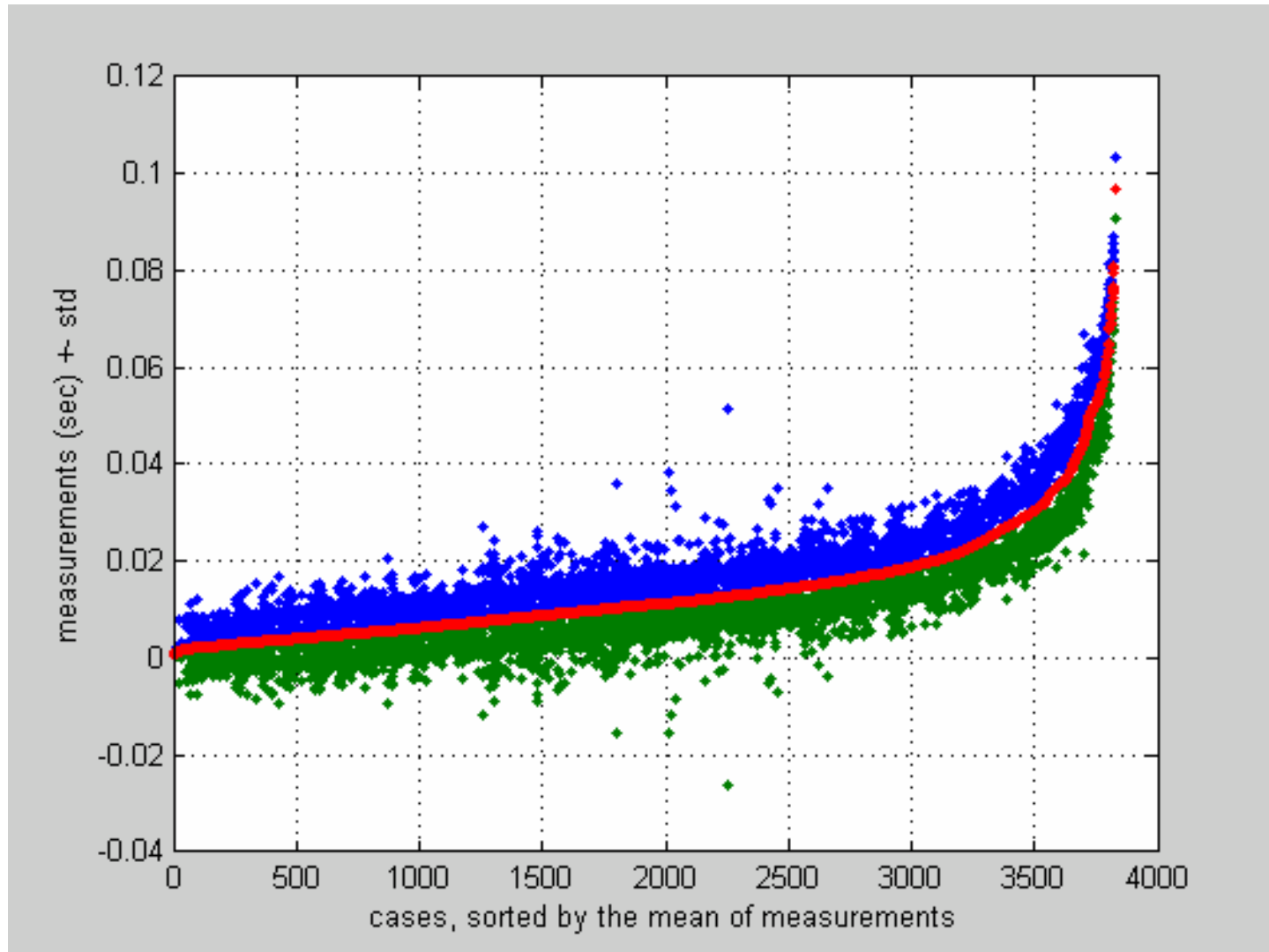
$$t_{fewtomany} = f(d_x, d_y, d_z, p_x, p_y, p_z, \text{Orientation}, p_x', p_y', p_z')$$

- Linear regression model
- Basis functions: poly, log, exp, etc.
- Incremental least-square solver

Modeling Results

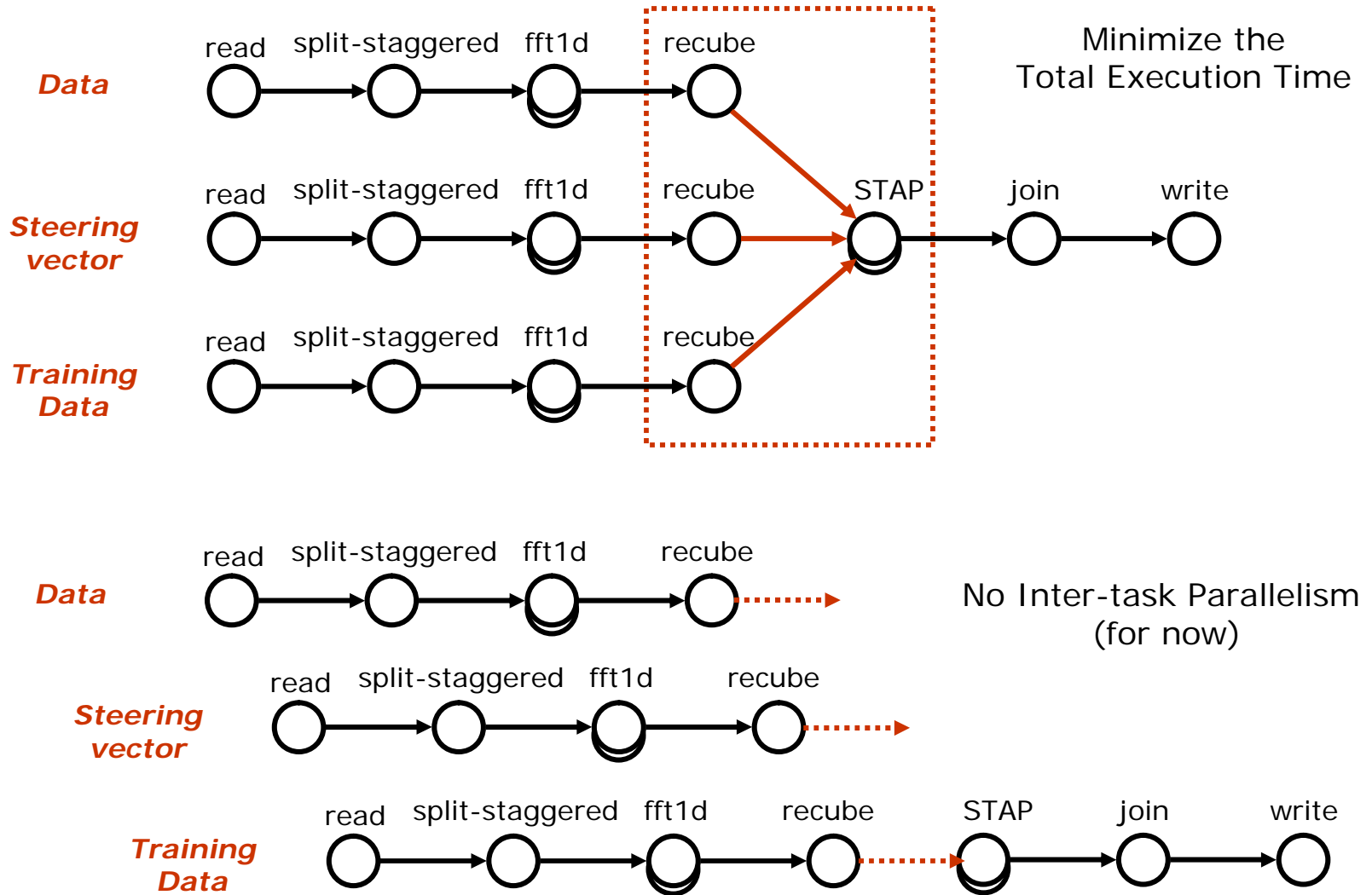


Measurements and Standard Deviation

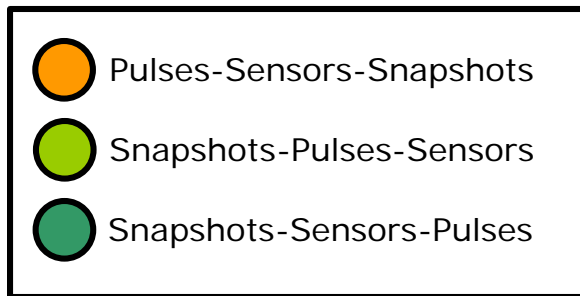
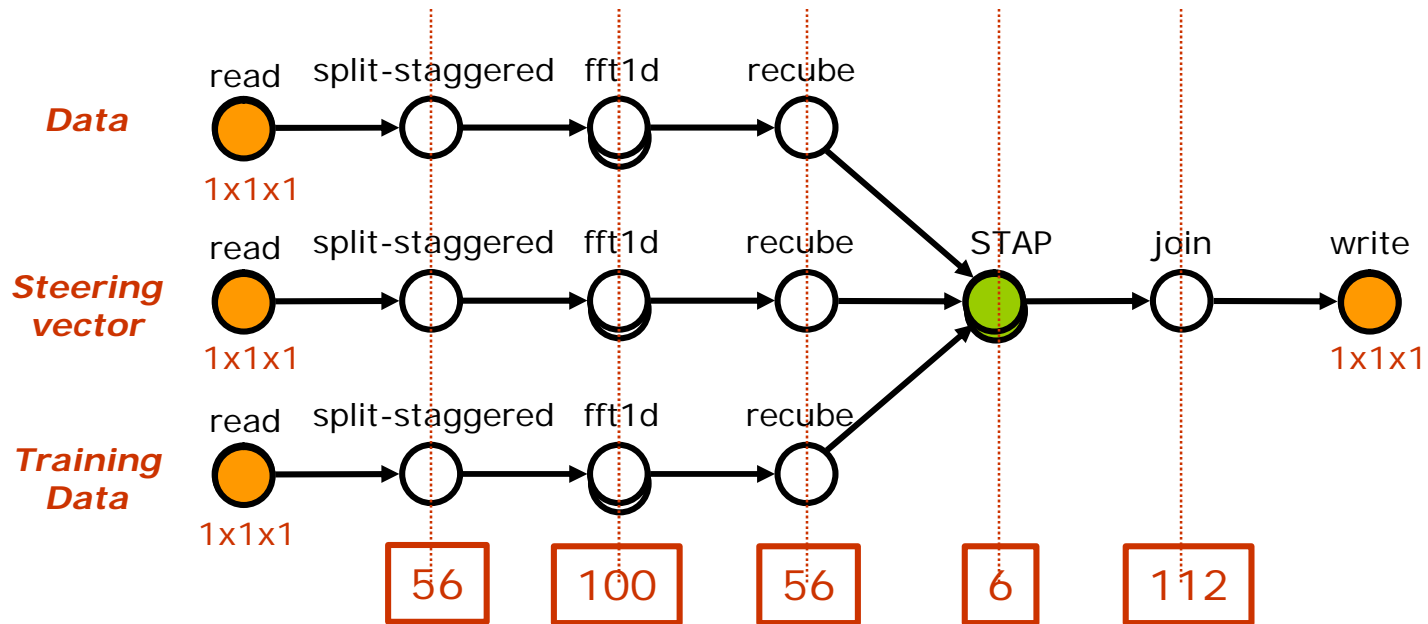


(From manytofew operation, vplus cluter at CTC)

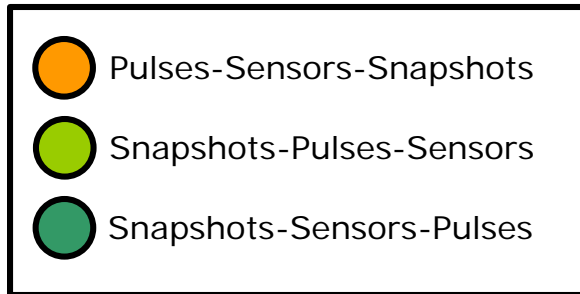
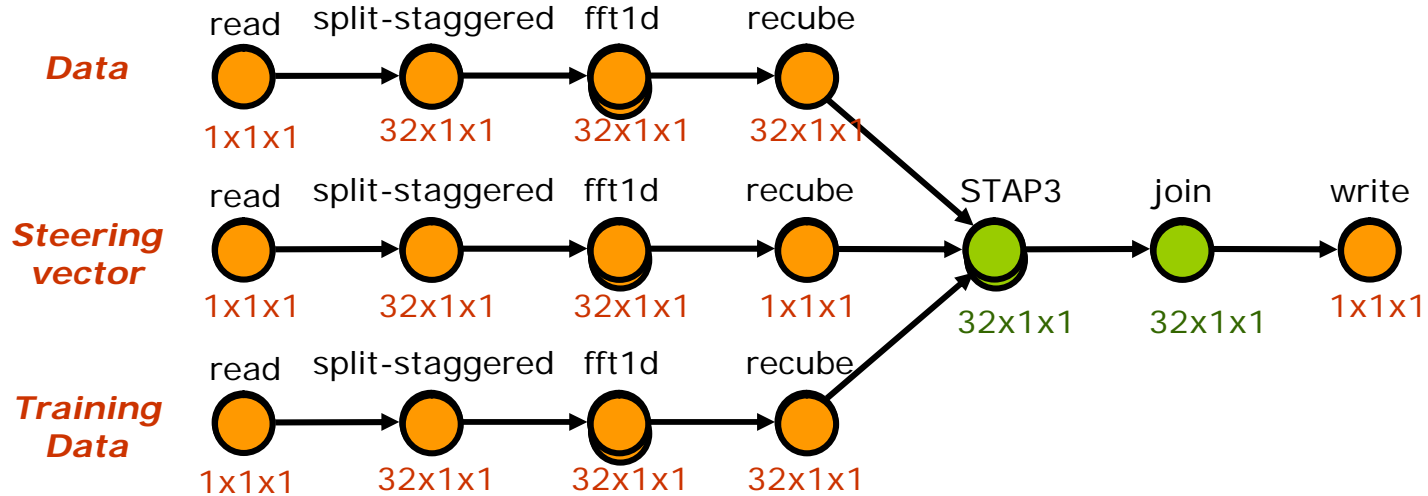
Parallel STAP Optimization: Dynamic Programming



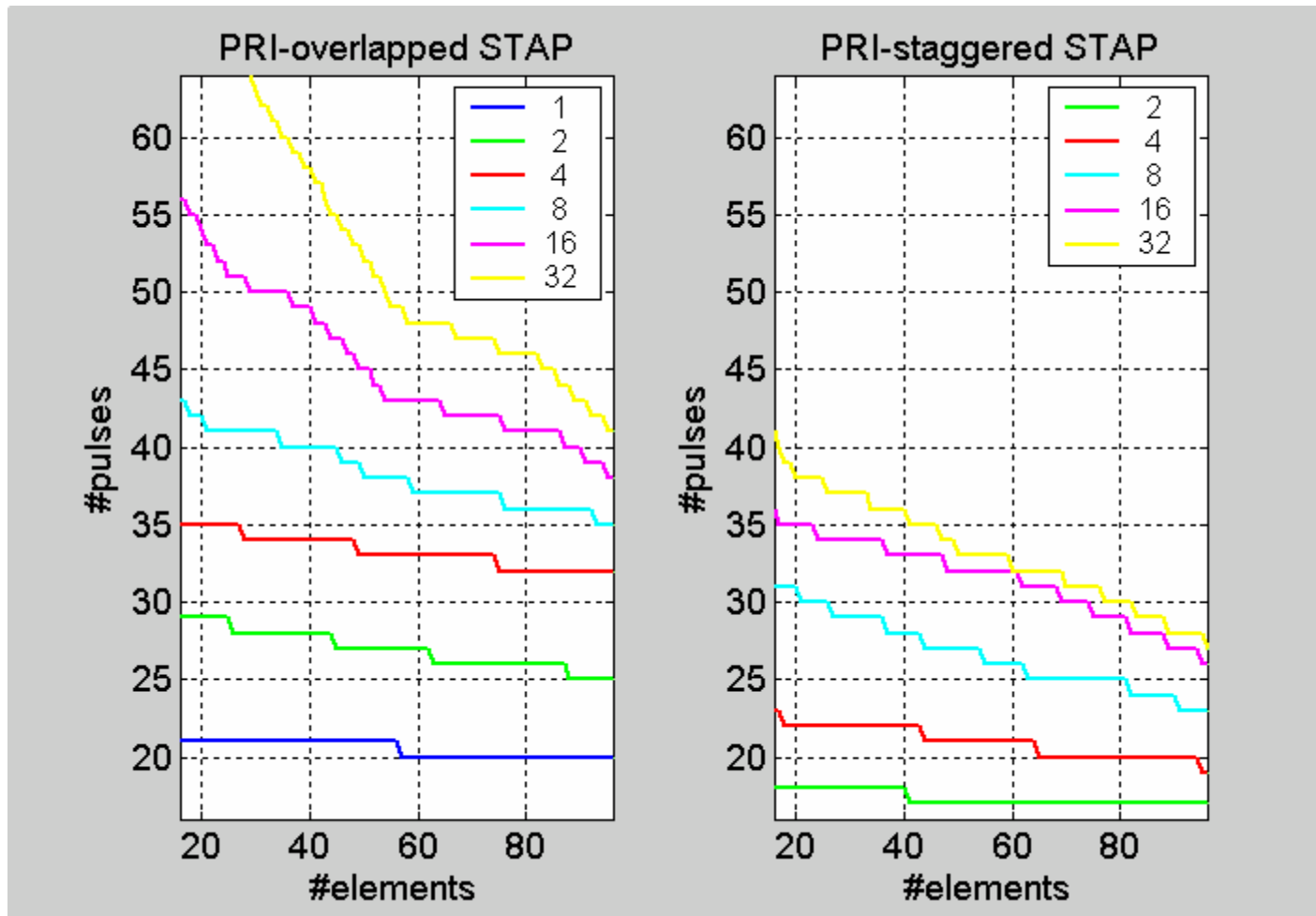
Parallel STAP Optimization: Results



Parallel STAP Optimization: Results



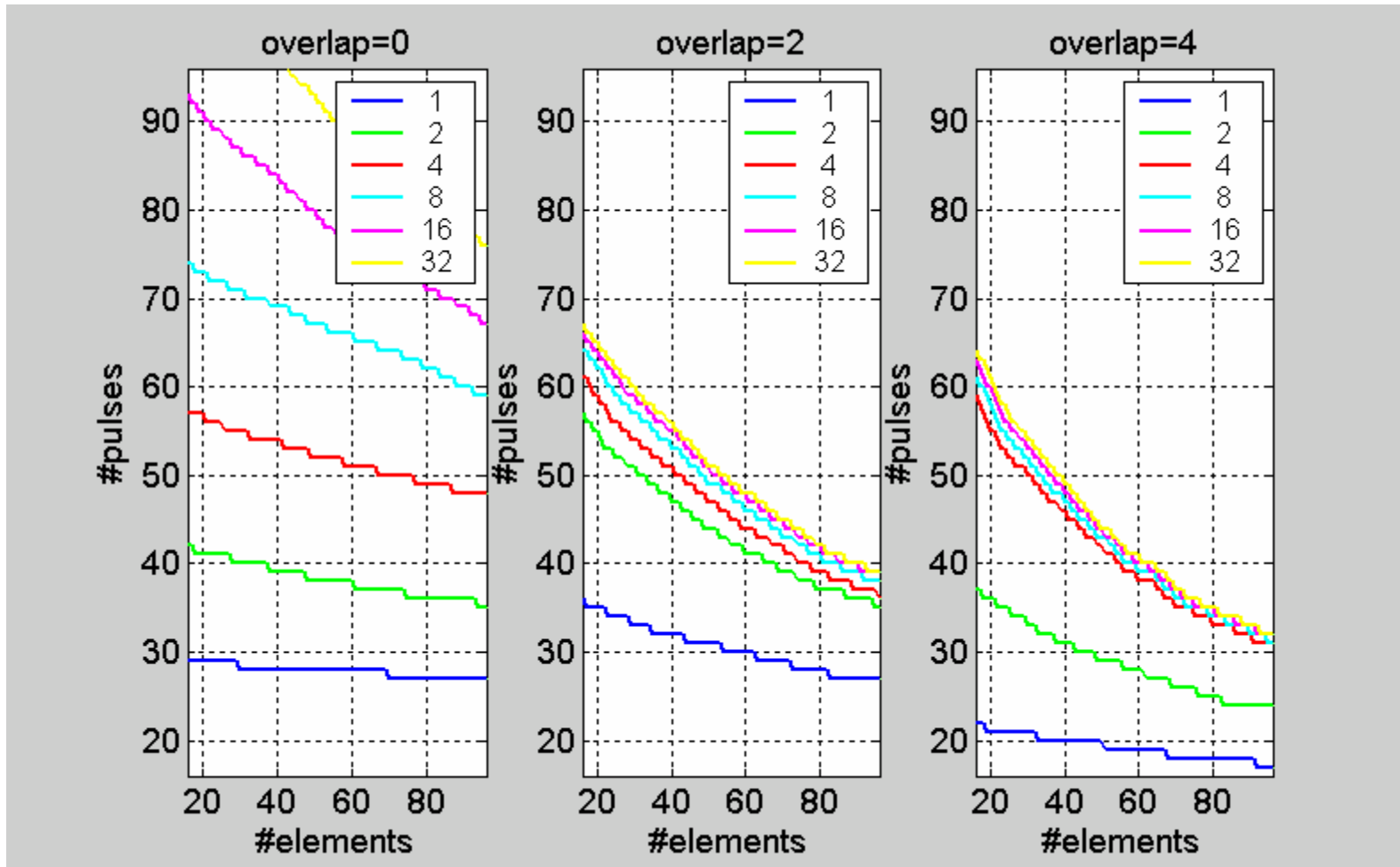
PRI-overlapped vs. PRI-staggered STAP algorithm



- #snapshots=2048
- #pulses=2*sublen
- Full wraparound

- Deadline of 100 units applied
- Legends: #processors

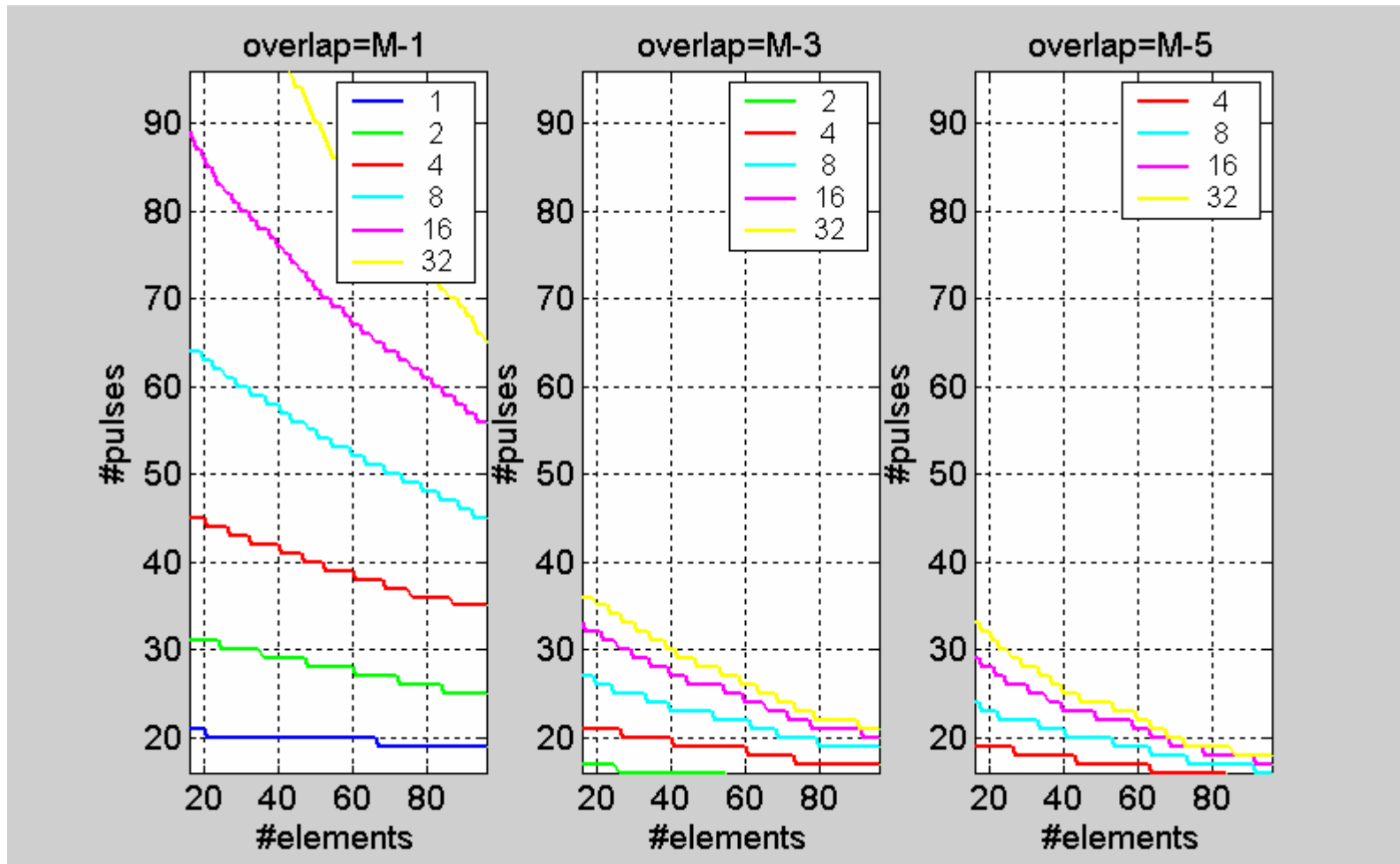
PRI-overlapped STAP algorithm w.r.t. *overlap*



- #snapshots=2048
- No wraparound

- Deadline of 20 units applied
- Legends: #processors

PRI-staggered STAP algorithm w.r.t. *overlap*



- #snapshots=2048
- No wraparound

- Deadline of 20 units applied
- Legends: #processors

Conclusion

Space-Time Adaptive Processing (STAP):

- Various heuristics
- Parallel processing
- Hard to optimize

Software framework:

- Prototyping various STAP with ALPS library
- Finding the optimal process cube sizes and optimal orientations
- Selecting the optimal implementation of each task

Work in progress:

- Allowing inter-task parallelism
- Refining models
- Deadlines vs. PFA and PD